CMPE451-MILESTONE REPORT 1

20/11/2021

Group 2

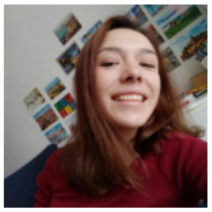
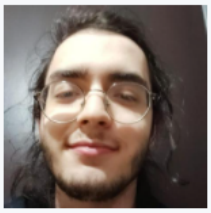
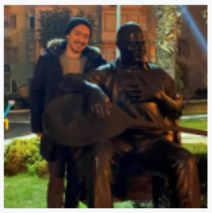
				
Doğukan Akar	Şefika Akman	Ömer Arslan	İbrahim Kağan Bayat	Berkay Döner
				
A. Necip Görgülü	Ege Can Kaya	Bengisu Özaydın	Mehmet Saraçoğlu	Kerem Zaman

Table Of Contents

Table Of Contents **2**

Executive Summary **3**

List and Status of Deliverables **4**

Evaluation of the status of deliverables **5**

Evaluation of Tools **6**
- Google Drive 6
- Discord 6
- Whatsapp 7
- GitHub 7
- Doodle 7
- React.js 8
- Material-ui 8
- Webpack 9
- Postman 9
- Git 9
- PyCharm 10
- Google Cloud Platform 10
- Docker 11
- PostgreSQL 11
- Django 11
- Slack 12
- Gmail 12
- Kotlin 13

Summary of Work Done by Each Team Member **13**

Implemented Functionalities **18**
- **Login Functionalities** **18**
- Register Functionality 21
- Forgot Password Functionality 24
- Get Event Functionalities 27
- Create Event Functionalities 30

Deliverables **34**
- Project Requirements 34
  - Glossary 34
  - 1. Functional Requirements 34
    - 1.1 User Requirements 34
    - 1.2 System Requirements 35
  - 2. Non-Functional Requirements 39
- Scenarios and Mockups 40

1. Creating an Event 40
2. User Searches for an Event 47
3. Player & Spectator Application 53
4. Event Creator Selecting from Applicants 64
5. Discussion 72
UML Diagrams 78
Class Diagram 78
Use Case Diagram 79
Sequence Diagrams 80
Project Plan&RAM&Communication Plan 91
Project Plan 91
Communication Plan 92
Responsibility Assignment Matrix 93

Executive Summary

For our first customer milestone, we have first focused on revising our project requirements, mockups, design diagrams, project plan and RAM. Then, we have focused on implementing the user management functionalities such as login, sign up, reset password and the design of related frontend and backend pages. We have also started implementing sports event functionalities.

Android Status: So far we have implemented Login, Sign up, Reset password features. We have also designed our profile page, homepage navigation bar and create event page (Although the pages are not functional right now). Existing features do not use dummy data. They are fully connected to our deployed backend server which means our application sends data to and gathers data from our database.

Android Moving Forward: We are planning to implement the “Create Event” feature next. Which is going to be followed by “View Event” and “Profile”. We have already designed some layouts for some of the features mentioned so we are going to move forward by implementing the actual features with the help of the Backend Team. We are trying to move according to our Project Plan so you can check it out to see our planned progression.

Backend status: On the back end, we started out by implementing the register and login functionalities. The login functionality uses the JWT refresh/access token system for authorization. Then we added the functionality to create event posts and view them one by one, or all together. Then, complying with the requests from the front end team, we added the functionality for resetting the password, which sends an email to the user with a token, and requires them to enter the same token to be able to change their password. Finally, we added the functionality of viewing a user’s profile.

Backend moving forward: First, we need to touch up on the user controller and add the functionality to update users. It was our intention to do this for the first milestone, however other priorities emerged and we let this go. Then, we will look into paginations for viewing events in a pagely manner. Also, there were some problems in the initial milestone where the remote deployment did not work with gunicorn, and we had to use Django's manage.py to get it working. We will first look into this issue, and then we can continue work complying with what is written on our Project Plan.

Frontend status: As the frontend team, we first focused on deciding the programming language, framework and any library we will use throughout the development process. We spent considerable time deciding these because our experience levels in frontend were very different and we wanted to use tools that allow every team member to contribute to frontend development without experiencing difficulties. We decided to use React.js as our main framework. We also first decided to use the Tailwind CSS library to improve the design of our pages. However, this library turned out to be difficult than we expected and we migrated to Material UI, which is a simpler React library. Using this library as the main design component, we have implemented login, register, forgot password, create event and homepage of our application. Unfortunately, due to miscommunication with the backend and lack of API documentation of some functionalities, we only managed to connect the login, register and forgot password pages to the deployed backend project. We also spent a lot of time trying to merge the backend and frontend to one port when deploying; however, this turned out to be much harder than expected due to the problem in connecting the React project to Gunicorn and Django and we could not deploy our frontend yet.

Frontend moving forward: Our first task is to connect the missing functionalities to the backend and deploying the frontend application itself. We also have to review our pages and communicate with the android team to ensure that the same formats are used in frontend and android pages. We also will review the input validations in the pages we implemented. Finally, we will follow the Project Plan and implement the search and filter functionalities, event application functionalities and comment&answer pages until the second customer milestone.

List and Status of Deliverables

Deliverable	Status
Project Requirements	Completed
UML Software Designs	Completed

Scenarios and Mockups	Completed
Project plan, Communication Plan, RAM	Completed
Milestone Review	Completed

Evaluation of the status of deliverables

Deliverable	Evaluation
Project Requirements	Requirements had to be changed since many concepts have been changed and we now know more about implementation. We have changed our requirements accordingly; however, we might still have some requirements that are unrealistic which we will only realize when we try to implement them.
UML Software Designs	We changed some of the sequence diagrams to include the database in them and improved the flow of the program.
Scenarios and Mockups	Since we have changed our requirements, we have updated our acceptance criteria according to the updated requirements. Also, we have updated some mockups.
Project plan, Communication Plan, RAM	As we have decided our android, backend and frontend teams, we have changed the RAM accordingly. Also, we revised our communication plan to include the PS hours. Finally, we have changed the dates in the project plan.
Milestone Review	In the milestone review, we have explained the status of all three teams and the future work we plan to do. Also, we have included the API documentation and the frontend&android pages of the functions we implemented along with the addressed

	requirements.

Evaluation of Tools

Google Drive

Google Drive is a file storage service. It has the google docs app to create and edit documents and spreadsheets.We created docs and spreadsheets for deliverables and milestone report. Drive helps us to be synchronized while working simultaneously.

Discord

We use discord for several purposes. Our main communication is on discord. We use discord to both chat and do online meetings. Also we use it for task distribution. We list the tasks and each member reacts to the task s/he will take. We created several chat channels to collect related texts on the same page like user requirements, system requirements. Also we have several voice channels for subgroups of works.

Whatsapp

Whatsapp is the easier way of communication. We just use it to reach other members quickly. Usually we don't take important decisions on whatsapp instead we use discord.

GitHub

GitHub

IT service management company

Everything we do is collected in github.
In our wiki page, we open separate pages for all works and upload what we have done. It is easy to update pages also it is easy to keep track who has made changes.
Another nice feature of github is that we can create an issue and make progress through issues. For example if we create an issue, we can add labels according to its importance, type, status etc., we can update and when it is done we can close it.

Doodle

Doodle is a tool for scheduling meetings in which we can create polls. We used it to schedule customer and team meetings.

React.js

React is a javascript framework for web apps. We used react with Material-ui toolkit. It has good components for frontend. React was a good choice for visuality but most of the team didn't know how to use it and we had difficulties on the frontend. We should do a better organization while we develop using react.

Material-ui

Mui is a react framework for some good looking and functional components. We used it for its components. We used Appbar, Card, Container, Grid and other components. Mui was a good choice. Its components are good and useful and all the team adapted to it fastly.

Webpack

Webpack is a static module bundler for modern JavaScript applications. We used webpack for creating static bundles and serving it within the django project on the server. Webpack also has a webpack dev server tool that also gives the hot-reload feature while developing pages.

Postman

Postman is a tool to test API methods with an user-friendly interface. It can be used to send both GET and POST requests to some URI's with different types of parameters. We used Postman to test both our own API methods and external API's.

Git

Git is version control system that is currently the industry standard in software development. It allows users to track and view changes made in files along with their previous versions. It is essential for coordinating work among programmers collaboratively writing code during the development of software. We used Git during every step of our development in conjunction with GitHub to keep track of our progress and to be able to view each other's codes.

PyCharm

PyCharm is an IDE by JetBrains which allows convenient tools for Python code editing. Some of our team members used it in the development and test runs of our practice application. It provides many convenience features such as syntax showing, error correction, an integrated VCS that can be used in conjunction with GitHub, a convenient GUI and an integrated terminal.

Google Cloud Platform

Google Cloud Platform (GCP) is a platform providing a collection of cloud computing services and APIs. We used GCP VM instances to deploy our application. When opening new account, GCP provides $300 for 90-days, which is very suitable for our project. We selected f1-micro instance type because of its $3.88 monthly price. It has allowed 0.2 vCPU and 0.6 GB RAM. We first thought that those features are enough for our application since our user base is very small and there is no high computational requirements, but we underestimated the requirements, deploying a few Docker container can utilize CPU 100%. As we observed, it would be better to upgrade server to an instance type allowing more CPU usage and having slightly more RAM. GCP provides an easy-to-use user interface that allows you to create an instance in minutes.

Docker

Docker is a great tool for managing different services in an efficient and modular way. We dockerized our API and frontend and used a ready-to-use PostgreSQL docker image for the database. Using the PostgreSQL image was very convenient to configure. Also, we used docker-compose to manage our different services which are two different containers. It would be very hard to deploy our application without docker. Docker saves us to manage each dependency, running our application, and configuring the database each time.

PostgreSQL

We used PostgreSQL server to build our database for the practice app, following the advice of our instructor. It is very similar to SQL so we did not have many problems using it. PostgreSQL provided the functionalities we needed for now, but we might want to change our database server to a No-SQL server, which is more flexible. PostgreSQL also comes with a convenient GUI tool called pgAdmin to create databases and check their content, which came in handy during development.

Django

,

Django is our new back-end framework for the project. Last semester, we were using Flask. The differences between the two are very easy to see. I find that Flask is a much simpler and intuitive framework, especially for someone coming from a Spring Book background. I see it as the “traditional” way to do back-end programming. However, Django is much more gimmicky and different. It does have a pretty standard structure on its own, but it takes some time getting used to it. Writing class “views” instead of just your regular API endpoint functions is one such gimmick, so is creating separate “apps” for virtually every functionality of your back end. It makes things super easy and automated, but it really takes some time to switch from a Flask mindset to a Django mindset.

Slack

Slack is our new form of more formal communication in the team. We still use Discord for the voice chats and WhatsApp for urgent messaging that everyone should be notified of, but the way Slack is designed compelled us to use it alongside those two. The channels system works better in Slack, where one user can choose to enter whichever channels they want in one server and opt out from others. This makes the separation between the back-end, front-end and Android teams easier.

Gmail

The back-end team used Gmail for the purposes of the forgot password functionality. We used it as the mail client of our app, sending password reset tokens. However, it constantly caused problems because of Gmail’s strict security checks and controls. In the end, we had to disable a lot of security measures to get Gmail to work with our app. We are definitely thinking of switching to another mail client that is actually designed not for personal use (like Gmail) but for apps and services. However, for the purposes of the first milestone, we could not find such a service that was free and did not require you to obtain a web domain. Hopefully, we will find one such mail service moving on.

Kotlin

Kotlin is a programming language for the Java Virtual Machine (JVM) and can, therefore, be used anywhere Java is used today (which is virtually anywhere). This includes server, client, web, and Android development. It's developed by JetBrains who are currently working to bring Kotlin to embedded systems and iOS as well, potentially making it a one-stop language for all application areas.

Since Kotlin allows us to write less code and is easy to adopt, we have decided to use Kotlin as our programming language for android development. Although some of us had no experience in android development Kotlin made it easy for us to start from scratch..

Summary of Work Done by Each Team Member

Doğukan Akar	- Revising Scenario #5 - Updating the RAM - Updating user requirements - Initialization of the frontend project - Creating A Profile Page - Implementing the signup page - Reviewing 5 pull requests
Şefika Akman	- Watching videos and reading documents about react - Learning MUI - Revising "Improved home page titles and event card layout #182" - Revising project plan with group

	- Revising RAM with group - Implementing navigation part of the project #166
Ömer Arslan	-Android Initialization -Creating activities and fragments + Main Activity ○ Home Fragment ○ Profile Fragment + Login Activity + SignUp Activity + Reset Password Activity -Changing app icon and app name -Updating .gitignore file -Connecting project to network via RestAPI + Creating modalz for request and response data classes + Creating RestAPIInterface (ReboundAPI) for communication with network + Adding LoggingInterceptor to RestAPIInterface for checking the responses from logcat easily + Creating endpoint methods in the interface + Connecting endpoints to the buttons -Connecting activities and fragments via buttons -Checking inputs and creating toast messages for success or error messages
İbrahim Kağan Bayat	● Initializing Django and Postgresql in my computer and connecting them. ● Watching many videos about Django to learn it.

	• Creating get event by id function in backend. • Creating get all events function in backend. • Creating post an event function in backend. • Creating serializer class for eventpost app in backend. • Learning observer pattern. • Studying activity stream. • Studying and creating serializer class in Django. • Creating a pull request for my backend app. #200 • Contributing to API documentation with my functions. Documentation • Issue for my functions #162, and other issue for contributing to API documentation. #196 • Contributing Ram. • Adjusting scenario 1's acceptance criterias and mockups. Issue: #135 • Reviewing project requirements.
Berkay Döner	• Revising project requirements. #130 • Revising Scenario&Mockup #3 #131 • Helped revising the project plan. #136 • Revising "Blocking a user" sequence diagram #138 • Revising "Following a user" sequence diagram #139 • Revising "Giving a badge to another user" sequence diagram #140 • Revising "Answering a comment on discussion page" sequence diagram #141 • Revising "Posting a comment on discussion page" sequence diagram #142 • Learning about React.js and Material UI • Implementing and designing the home page. #143 • Implementing the login page. #153

	• Implementing the create event page. #181 • Implementing the forgot password page. #179 • Opening and merging the pull request for the implemented functionalities. #170, #171, #182, #183, #184.
Ahmet Necip Görgülü	• Reworked the project plan for the semester. • Implemented Sign up Activity. • Designed the signup layout, • Designed the homepage navigation bar • Presented the Android part of our project for the Milestone Presentation. • Prepared the communication plan. • Wrote the Android part of the Executive Summary. • Updated the RAM
Ege Can Kaya	• Back-end: registration functionality • Back-end: login functionality • Back-end: forgot password functionality • Back-end: User model • Back-end: view user profile functionality • Putting the back-end implementations together by merging the separate branches and solving conflicts • Fixing the back-end file hierarchy • Attempting to merge front end and back end, having them run on the same port (did not succeed) • Contributing to remote deployment of back end • Contributing to the API documentation • Contributing to the Milestone report by adding the back-end part of the

	executive summary and the tools used by the back-end team • Contributing to the updated RAM diagram
Bengisu Özaydın	• User requirements update and revision (#130) • Revision of scenario 4 (#134) • Revision and update of RAM (#137) • Study of Django. • Creation of .gitignore file. • Initialization of backend codebase (#144) • Documentation of Django guide for developers' ease. (#145) • Study of React framework and MaterialUI for possible responsibilities on the frontend in the future.
Mehmet Saraçoğlu	• Modified the user and system requirements. Issue: #130 • Added the create-event page of Android. Issue: #195
Kerem Zaman	• Research on Django • Helping to initialize Django backend • Attending group meetings and backend meetings • Designing app logo on Canva with different resolutions • Configuring project to be used with PostgreSQL • Creating User, Post and Eventpost models in Django • Initializing docker for postgresql database and backend service with related Dockerfiles and docker-compose (Issue: #161) • Opening Google Cloud Server and starting a new instance • Configuring NGINX server as reverse proxy to server local containers to the web

	• Dockerizing all backend, eliminating problems and remote deployment (Issues: #178 , #192) • Opening and merging PRs (#185, #187, #191)

Implemented Functionalities

Login Functionalities

Login (obtain JWT token)

POST 'api/token/obtain/'

- **Author:** Ege Can Kaya
- **Definition:** Takes a set of user credentials and returns an access and refresh JSON web token pair to prove the authentication of those credentials.
- **Example Response Value:**

Example Response Value:

{
"refresh": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTYzODAzMzQ0OCwiaWF0IjoxNjM2ODIzODQ4LCJqdGkiOiJiNDkxZjIwZjU2OTY0YTJiYjI2OTlyNWJjMmZkNTc3YSIsInVzZXJfaWQiOjR9.tCGeBNzTXXBA8fTC2eqxyy2wfa1tYEN8nL0mSCzftP0",

"access": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjM2ODI0MTQ4LCJpYXQiOjE2MzY4MjM4NDgsImp0aSI6ImY0ZWIyNDY5ZWIwZDQyMDQ4NzI1NTAyNDQ5YjZlYTM2IiwidXNlcl9pZCI6NH0.HgN9BIi3F-1lnio_s32IHdsHNDUlAuNcWaE_s9A11vY"
}

Example Failure Value:

{
  "detail": "No active account found with the given credentials"
}

- **Parameters**

Parameter	Description	Parameter Type	Data Type
username	Username	Body	String
password	Password	Body	String

- **Response Messages**
  - 200: OK.
  - 401: Unauthorized.

Login (refresh JWT token)

POST 'api/token/obtain/'

- **Author:** Ege Can Kaya
- **Definition:** Takes a refresh type JSON web token and returns an access type JSON web token if the refresh token is valid.
- **Example Response Value:**

Example Response Value:

{

"access": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjM2ODI0NDUxLCJpYXQiOjE2MzY4MjM4NDgsImp0aSI6IjY3ZTRlZDE2NzQyYTQyYTZiOGY2OTBiZTFmMTUzZjQ1IiwidXNlcl9pZCI6NH0.J9WhNq04miuDSABsobZtPHS5drcXxcq0eQE6MH44-zU",
"refresh": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTYzODAzMzc1MSwiaWF0IjoxNjM2ODI0MTUxLCJqdGkiOiI0NjQ0MDFjNGY2ZjA0OTA3YjZlZTA3NDA0ZmQ0ZmI0MiIsInVzZXJfaWQiOjR9.r4q3jpcilwT6bqXupFqhSnu7pc58IzPNJXilnfwhwwM"
}

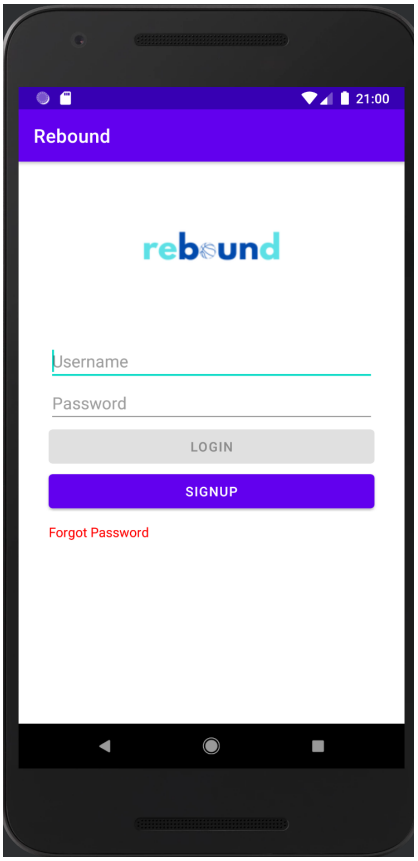
Example Failure Value:

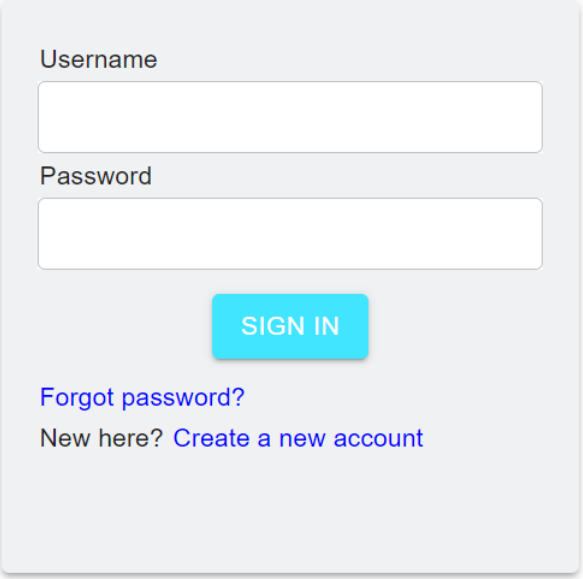
```
{
   "detail": "Token is invalid or expired",
   "code": "token_not_valid"
}
```

- **Parameters**

Parameter	Description	Parameter Type	Data Type
refresh	Refresh type JWT token	Body	String

- **Response Messages**
    - 200: OK.
    - 401: Unauthorized.





Register Functionality

Register

POST 'api/user/create/'

- **Author:** Ege Can Kaya
- **Definition:** Allows a user to register
- **Example Response Value:**

Example Response Value:

```
{
  "email": "egekaya@somemail.com",
  "username": "egekaya"
}
```

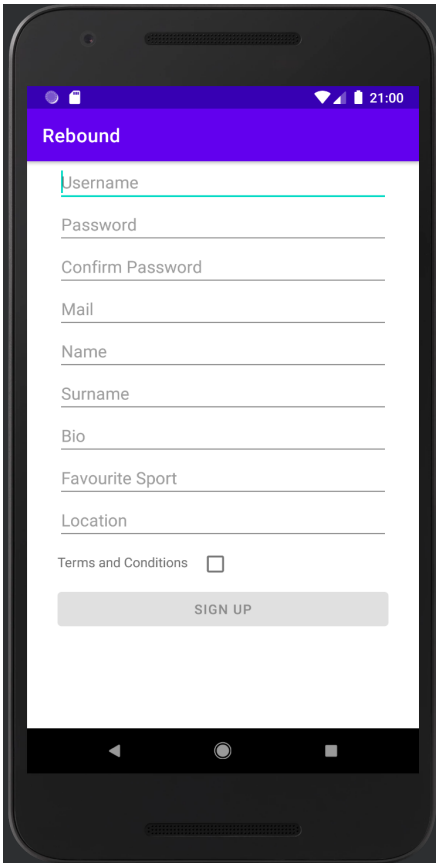
Example Failure Value:

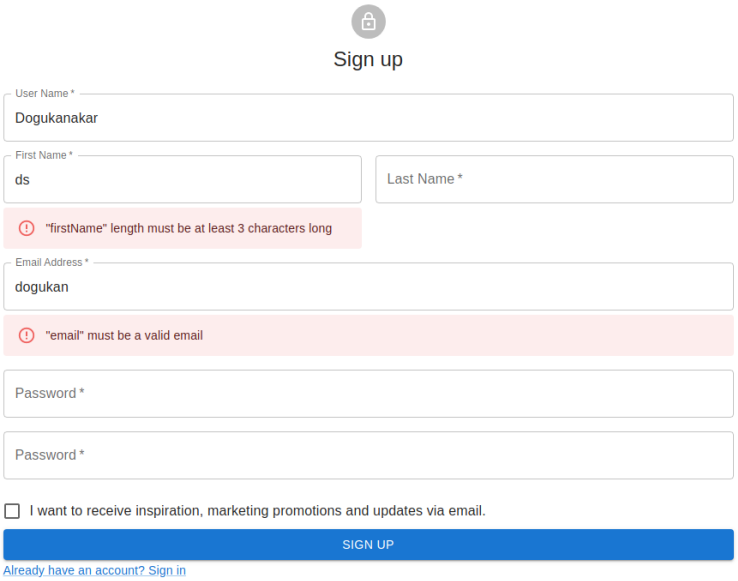
```
{
  "username": [
    "This field is required."
  ],
  "password": [
    "This field is required."
  ]
}
```


- **Parameters**

Parameter	Description	Parameter Type	Data Type
username	Username of registering user	Body	String
email	Email address of registering user	Body	String
password	Password of registering user	Body	String

- **Response Messages**
  - 201: Created.
  - 401: Bad request.


21:00
Rebound
Username
Password
Confirm Password
Mail
Name
Surname
Bio
Favourite Sport
Location
Terms and Conditions
SIGN UP


Sign up
User Name *
Dogukanakar
First Name *
ds
Last Name *
"firstName" length must be at least 3 characters long
Email Address *
dogukan
"email" must be a valid email
Password *
Password *
I want to receive inspiration, marketing promotions and updates via email.
SIGN UP
Already have an account? Sign in

Forgot Password Functionality

Reset password (send mail)

POST 'api/password/reset/'

- **Author:** Ege Can Kaya
- **Definition:** Provides a method to request a password reset token based on an e-mail address
- **Example Response Value:**

Example Response Value:

```
{
  "status": "OK"
}
```

Example Failure Value:

```
{
  "email": [
    "We couldn't find an account associated with that email. Please try a different e-mail address."
  ]
}
```

- **Parameters**

Parameter	Description	Parameter Type	Data Type
Email	Email address of user trying to reset their password	Body	String

- **Response Messages**

    - 200: OK.
    - 401: Bad request.

Reset password (confirmation)

POST 'api/password/reset/'

- **Author:** Ege Can Kaya
- **Definition:** Provides a method to reset a password based on a unique token
- **Example Response Value:**

Example Response Value:

```
{
  "status": "OK"
}
```

Example Failure Value:

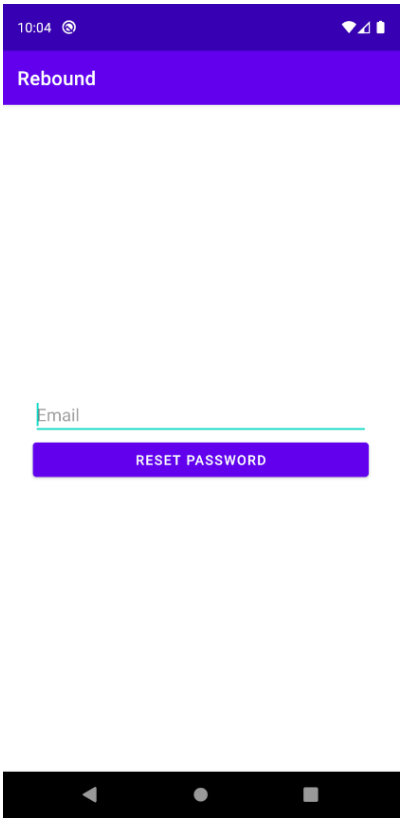
```
{
  "detail": "Not found."
}
```

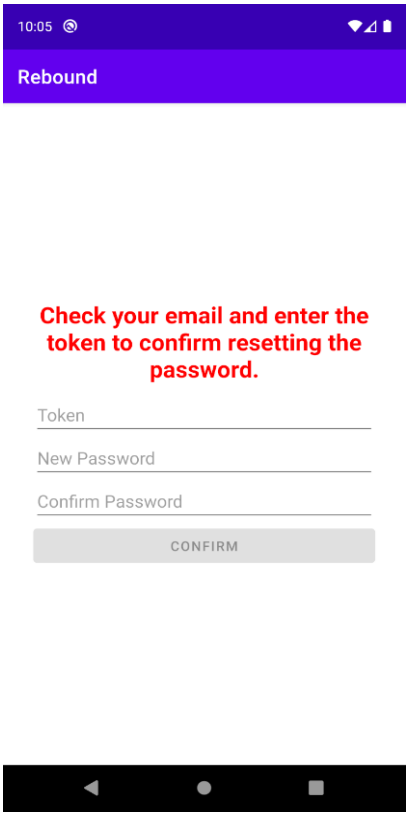
- **Parameters**

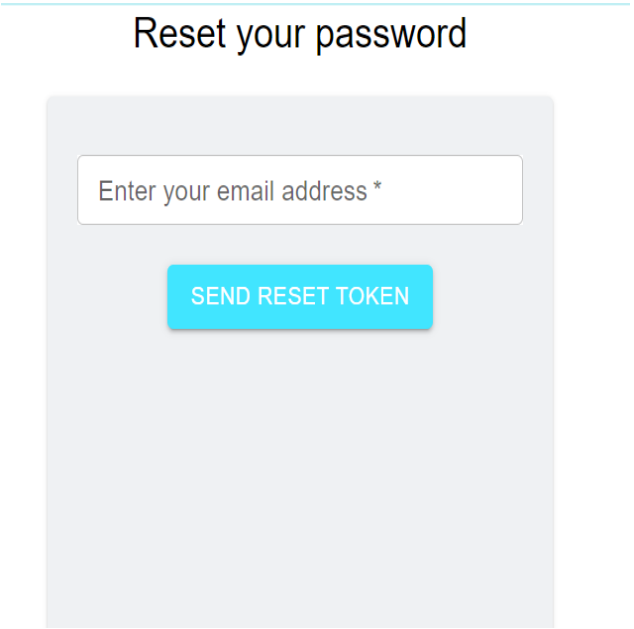
Parameter	Description	Parameter Type	Data Type
Password	New password for the user	Body	String
Token	Token received by email by the user allowing them to choose a new password	Body	String

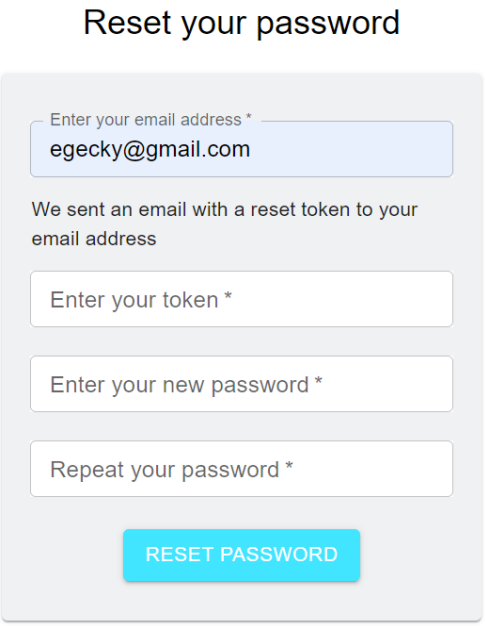
- **Response Messages**

- 200: OK.
- 401: Bad request.









Get Event Functionalities

Get event by id

GET 'eventposts/<id of event>'

- **Author:** İbrahim Kağan Bayat
- **Definition:** Allows a user to fetch an event from database
- **Example Response Value:**

Example Response Value:


```
{
    "id": 2,
    "content": "Event123",
    "title": "Event123",
    "creation_date": "2021-11-01T19:05:38Z",
    "location": "EventLoc",
    "date": "2021-11-01T19:06:17Z",
    "duration": "19:06:33",
    "sport": "sport",
    "age_group": 8,
    "player_capacity": 5,
    "spec_capacity": 4,
    "players": 7,
    "spectators": 6,
    "latitude": 6.0,
    "longitude": 5.0,
    "owner": 1
}
```

Example Failure Value:

{

}

- **Parameters**

Parameter	Description	Parameter Type	Data Type
id	ID of the searched event	Path	int

- **Response Messages**
  - 200: OK.
  - 401: No content.

Get all events

GET 'eventposts/all'

- **Author:** İbrahim Kağan Bayat
- **Definition:** Allows a user to fetch all event from database
- **Example Response Value:**

Example Response Value:

[
{
"id": 1,
"content": "Event",
"title": "Event1",
"creation_date": "2021-11-01T19:05:38Z",
"location": "EventLoc",
"date": "2021-11-01T19:06:17Z",
"duration": "19:06:33",

"sport": "sport",
"age_group": 8,
"player_capacity": 5,
"spec_capacity": 4,
"players": 7,
"spectators": 6,
"latitude": 6.0,
"longitude": 5.0,
"owner": 1
},
{
"id": 2,
"content": "Event123",
"title": "Event123",
"creation_date": "2021-11-01T19:05:38Z",
"location": "EventLoc",
"date": "2021-11-01T19:06:17Z",
"duration": "19:06:33",
"sport": "sport",
"age_group": 8,
"player_capacity": 5,
"spec_capacity": 4,
"players": 7,
"spectators": 6,
"latitude": 6.0,
"longitude": 5.0,
"owner": 1
}
]

Example Failure Value:

{

}

- **Parameters:** None

- **Response Messages**
    - 200: OK.
    - 204: No content.

Create Event Functionalities

Create an event

POST 'eventposts/'

- **Author:** İbrahim Kağan Bayat
- **Definition:** Allows a user to create an event
- **Example Response Value:**

Example Response Value:


```
{
  "id": 1,
  "content": "Event",
  "title": "Event1",
  "creation_date": "2021-11-01T19:05:38Z",
  "location": "EventLoc",
  "date": "2021-11-01T19:06:17Z",
  "duration": "19:06:33",
  "sport": "sport",
  "age_group": 8,
  "player_capacity": 5,
  "spec_capacity": 4,
  "players": 7,
```

"spectators": 6,
"latitude": 6.0,
"longitude": 5.0,
"owner": 1
}

Example Failure Value:

{

}

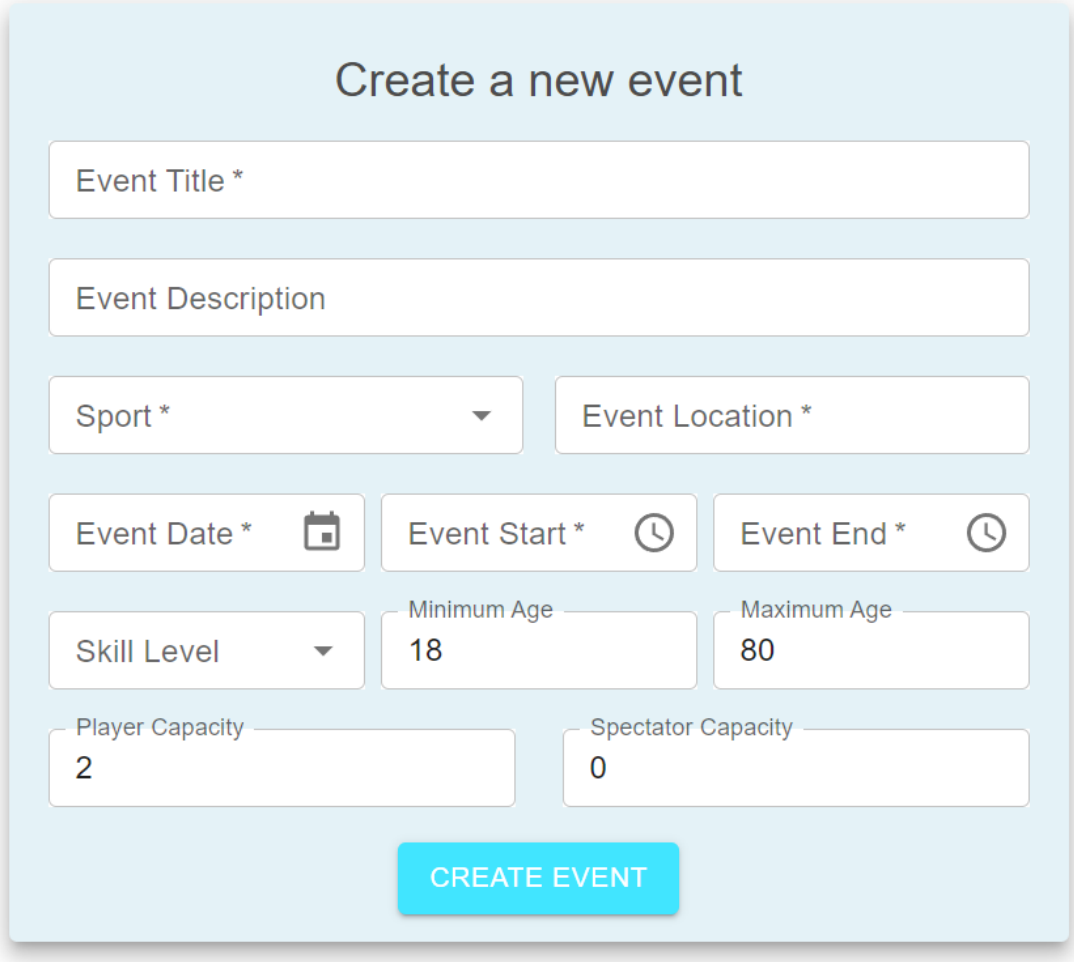
- **Parameters**

Parameter	**Description**	**Parameter Type**	**Data Type**
id	ID of the searched event	Body	Integer
content	Content of the searched event	Body	String
title	Title of the searched event	Body	String
creation_date	Creation date of the searched event	Body	DateTime
location	Location of the searched event	Body	String

date	Date of the searched event	Body	DateTime
duration	Duration of the searched event	Body	DateTime
sport	Sport type of the searched event	Body	Char
age_group	Which age group is preferred for the searched event	Body	Integer
player_capacity	Player capacity of the searched event	Body	Integer
spec_capacity	Spectator capacity of the searched event	Body	Integer
players	Players of the searched event	Body	Integer
spectators	Spectators of the searched event	Body	Integer
latitude	Latitude of the searched event	Body	Float

longitude	Longitude of the searched event	Body	Float
owner	Owner of the searched event	Body	User

- **Response Messages**
  - 200: OK.
  - 401: Bad request.



Deliverables

Project Requirements

Addressed requirements are displayed in green, if they are fully implemented and in yellow if they are implemented partially. Some addressed requirements have explanations at the end.

Glossary

1. Functional Requirements

1.1 User Requirements

1.1.1. Sign Up

1.1.1.1. Users shall register to be a registered user.
1.1.1.2. Users shall specify their email, password and username while registering.
1.1.1.3. Users shall accept the terms and conditions document to register. (Android team implemented a checkbox to accept these but we do not have such documents yet.)
1.1.1.4. Users shall fill personal information form while registering.
1.1.1.5. Users should select their interests and skills to get suggestions from the system.

1.1.2. Sign In

1.1.2.1. Registered users shall enter their email or username and password to sign in. (We have implemented a sign in functionality for username only, sign in through email should be implemented.)

1.1.3. Registered Users

1.1.3.1. Registered users shall have a personal profile page. (Connection to the backend has problems)
1.1.3.2. Registered users shall not be blocked by the user to view a user's profile page.
1.1.3.3. User should be registered in order to write comments under posts.

1.1.4. Search

1.1.4.1 Registered users shall use the search feature to find users or posts.

1.1.5. Post

1.1.5.1 Event Post

1.1.5.1.1 Registered users shall enter the values of the event to create an event post. (Connection to the backend is missing)
1.1.5.1.2 Event creators shall specify the type of sports events, manage location and type of sports events.
1.1.5.1.3 Event creators shall select the participants among the applicants.
1.1.5.1.4 Event creators shall report game results along with analytics.
1.1.5.1.5 Users shall indicate the type of sport and location in order to filter events.
1.1.5.1.6 Registered users shall specify their interests and skills to be able to get event recommendations.

1.1.5.2 Equipment Post

1.1.5.2.1 Registered users shall enter the values of the equipment to create an equipment post.
1.1.5.2.2 Registered users shall select the proper post and enter the contents of a comment to enter a comment.
1.1.5.2.3 Users shall indicate the type of sport and location in order to filter equipment.
1.1.5.2.4 Registered users shall specify their interests and skills to be able to get equipment recommendations.

1.1.6. Notification

1.1.6.1 Registered users shall select which notifications that they will see.
1.1.6.2 Registered users shall enable or disable all or a subset of notifications.

1.2 System Requirements

1.2.1 Sign Up

1.2.1.1. System shall ask for username, email and password for regular signup.

1.2.1.2. System shall send verification mail after regular signup.
1.2.1.3. System shall redirect users to profile creation page immediately after signup. (Frontend and Android pages are not the same, should be resolved.)
1.2.1.4. System shall not allow to sign up with same e-mail more than once at the same time.

1.2.2 Sign In

1.2.2.1. System shall allow users to sign in with email/username.
1.2.2.2. System shall allow users to logout by clicking a button. (Frontend part is missing.)

1.2.3 User Types

1.2.3.1. Banned User
1.2.3.2. Regular User
1.2.3.3. Administrator

1.2.4 User Profile

1.2.4.1. System shall ask user their interests and skills while creating profile
1.2.4.2. Interests and skills shall be updated according to games registered to system by admins
1.2.4.3. User profile shall display personal information, skills, badges, followers, followings and post history.
1.2.4.4. System shall allow users to edit their profile page and password.
1.2.4.5. System shall allow users to delete their profiles.

1.2.5 Events

1.2.5.1. System should recommend sports events to the users according to their interests.
1.2.5.2. System shall send emails and push notifications when a registered event approaches.
1.2.5.3. System shall allow event creators to specify event type, needed number of players, location of event, event time, skills of participants, age range of participants. (Connection to the backend is missing.)
1.2.5.4 Events with needed users registered shall be signed as full, so other people can filter them.
1.2.5.5 System shall allow guests to view the events list and individual

event pages, without participating in them.
1.2.5.6 All users shall be allowed to list all events on the events page.
1.2.5.7 Event pages shall display the event creator, event location, sports field, applicants, participants and spectators of the event. (Participants and spectators are not displayed yet.)
1.2.5.8 System shall allow all users to view the profile pages of the event creator, any of the applicants, participants and the spectators.
1.2.5.9 System shall allow only registered users to apply to events.

1.2.6 Searching

1.2.6.1 Event Search

1.2.6.1.1. Events shall be filtered by user according to event type, appropriate age range, location, date, hours, skill levels of attendees and availability (full or player needed).
1.2.6.1.2. Events shall be sorted with respect to name, time and location (e.g. from closer events to far away ones).

1.2.6.2 User Search

1.2.6.2.1. Users shall be filtered according to their ages, skills and interested sports.
1.2.6.2.2. Users shall be sorted with respect to their names and ages.

1.2.7 Sport Equipment Post

1.2.7.1. System shall recommend some websites to the users for their equipment needs.
1.2.7.2. System shall allow filtering and sorting the equipment websites according to the name of the website and name of the sport.
1.2.7.3. System shall allow equipment posts to be edited by their posters.

1.2.8 Reporting

1.2.8.1. System shall allow users to report some problems to event creators in the event.
1.2.8.2. System shall allow users to report some bugs to administrators.
1.2.8.3. System shall allow users to report some users for misusage.

1.2.9 Discussions

1.2.9.1. System shall allow users to discuss under event or equipment post by writing comments.
1.2.9.2. System shall also allow users to answer/reply discussion comments.
1.2.9.3. Users shall not be able to reply to the answers/replies, thus replies shall be the deepest level of conversations.

1.2.10 Notifications

1.2.10.1. System shall notify users when their player/spectator applications are approved/rejected.
1.2.10.2. System shall notify users for any related event happening around if they share their locations.
1.2.10.3. Users shall be notified if some people withdraw from an event after the event is signed as full. (Both registered users and users signed as want to attend but can't register because of capacity)
1.2.10.4. Users shall be notified if the event they applied/selected to be a player/spectator is delayed/cancelled.
1.2.10.5. Users shall be notified when a user follows them.
1.2.10.6. Users shall be notified when a user comments on the discussion page of an event they created.

1.2.11 Badges

1.2.11.1. There shall be badges that are given by the system after some milestones, that are given by the event creators at the end of the event and that are given by the users.
1.2.11.2. There should be badges that indicate bad behavior.
1.2.11.3 Only registered users shall be allowed to give badges to the creator of a post or to fellow players of an event.

1.2.12 User Interactions

1.2.12.1. System shall allow registered users to follow other users, so that they can be notified about the events created by the users they follow.
1.2.12.2. System shall allow users to block other users.

1.2.13 Analytics

1.2.13.1. System shall ask for and store analytics for players, event creators and sports fields after events.
1.2.13.2. System shall allow users to see the analytics for players, event creators and sports fields.

2. Non-Functional Requirements

2.1. Availability

2.1.1. System shall be available as a web application for modern browsers such as Chrome and Firefox.
2.1.2. System shall be available for Android based devices.
2.1.3. English language shall be supported.

2.2. Deployment

2.2.1. System should be containerized using Docker.(Frontend part is not dockerized.)
2.2.2. System should run on an Amazon EC2 server. (We decided to try Google Cloud and deployed our application there, we might change it to Amazon EC2 in the future.)

2.3. Performance

2.3.1. Response time for any request shall be below 5 seconds.

2.4. Privacy

2.4.1. All personal data shall be stored and processed according to KVKK and GDPR.
2.4.2. Users shall accept the privacy policy when signing up to the system.
2.4.3. Users shall be able to choose which of their personal data to be seen publicly.

2.5. Security

2.5.1. Passwords shall be at least 8 characters long and be a mixture of lowercase characters, uppercase characters and digits.
2.5.2. Passwords shall be encrypted using SHA-256 before storing.
2.5.3. Users shall verify their login information via email after 3 unsuccessful login attempts.
2.5.4. All inputs from users shall be checked to prevent SQL injections.
2.5.5. There shall be a password reset facility that sends a verification code via email or SMS.
2.5.6. System should only allow predetermined known file types to be uploaded. (e.g. PNG and JPG for images)
2.5.7. System should hide the passwords while typing on password bar. (Android sign up page does not hide passwords.)

2.6. Standards

2.6.1. System shall follow the standards introduced by W3C Activity Stream Standard 2.0.

2.6.2. The semantic taggings should be supported with Wikidata.

Scenarios and Mockups

1. Creating an Event

Persona

Eren Yiğit
22 years old
Computer Engineering student at Bogazici University
He is in summer vacation
He spends all day with playing computer games
He is a little interested in football, he can't play well.

Story

After spending a couple of weeks with online games, Eren realized that his lifestyle is very unhealthy. He never does sport, he is mentally depressed because of the attitudes of other gamers and he decided to give a break to online games. He wants to play football with his friends but he doesn't have enough friends to organize a football match. He decided to use our application to create a football event. He wants players at beginner level because they are not so good players.

Pre-conditions

Eren registered to the system.
Eren signed on to the system.
Eren created his profile and specified his skills and interests.

Goals

Eren wants to create a football event at a stadium in his city.

Actions

Eren login to our website.
He clicks on "Create event" because he and his friends want only four football players.
Eren specified sport type, date, hours and location of the football event.
Eren specified skills, count, age, gender and role of the players because they don't want many players or every type of player.

After some time, Eren has seen some notifications about applications to his events on the website.
Eren analysed applicants and approved them if he wants until the quota is full and the team are ready.

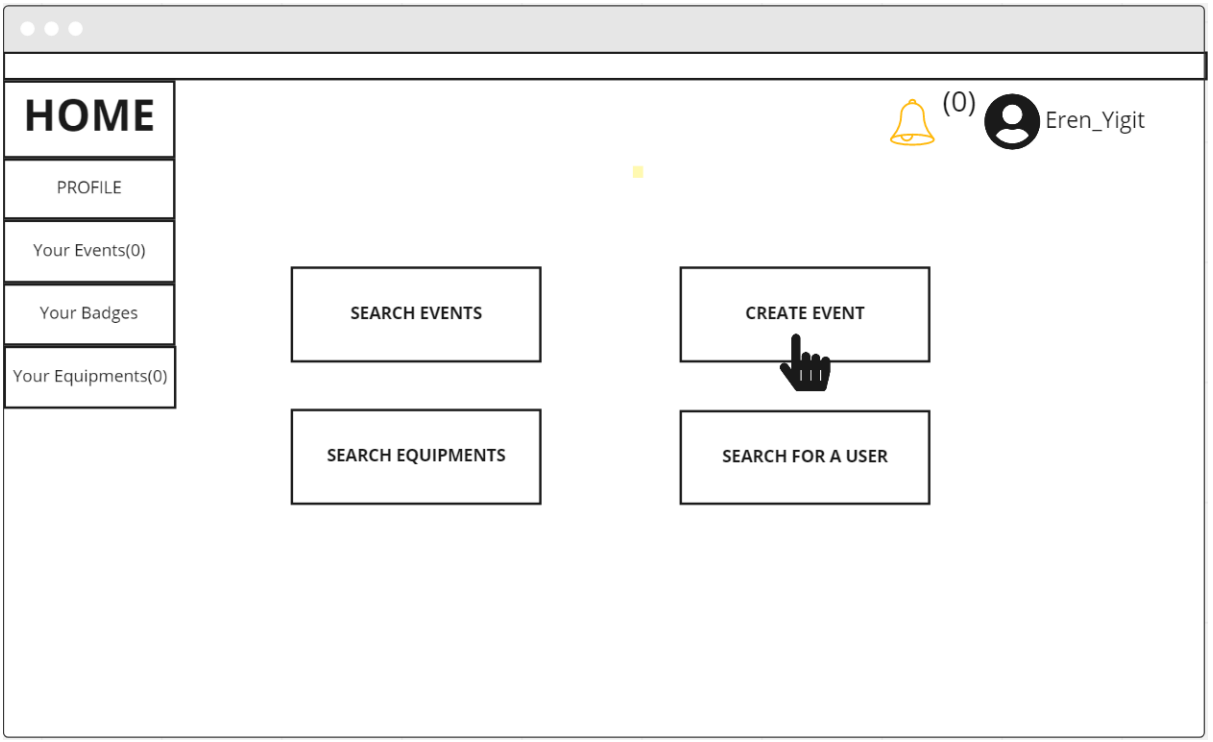
Acceptance Criteria

- (1.2.5.3.) System shall allow only event creators to create events.
- (1.2.5.4.) System shall allow event creators to specify needed number of player, location of event, event time, skills of participants, age of participants.
- (1.1.5.5.) Events with needed users registered shall be signed as full, so other people can filter them.

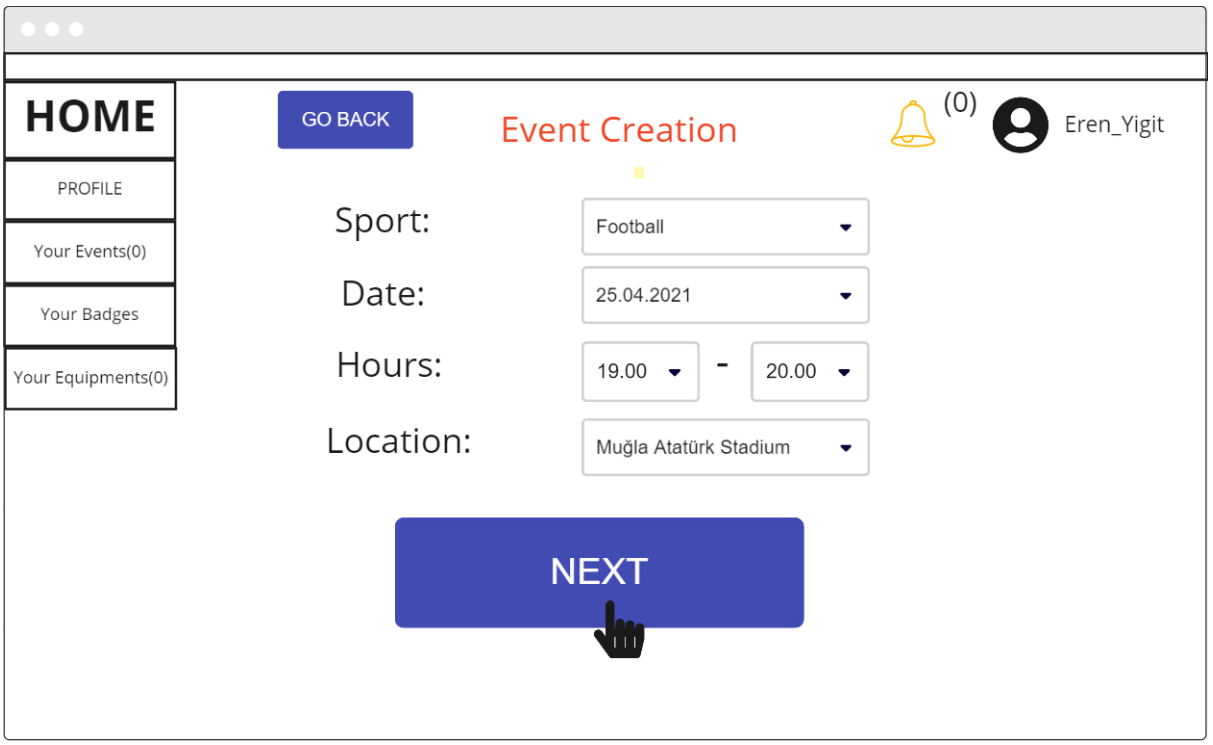
Scenario with Mockups

Web

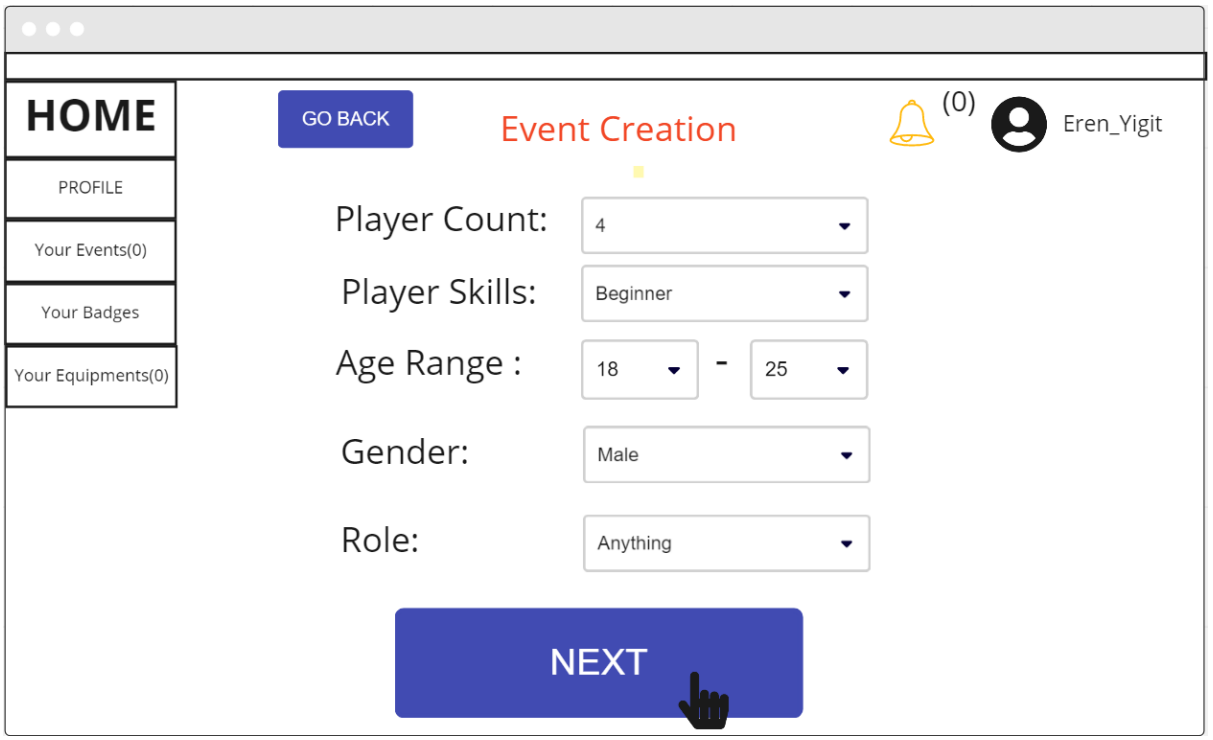
1. Eren clicked on the create event button.



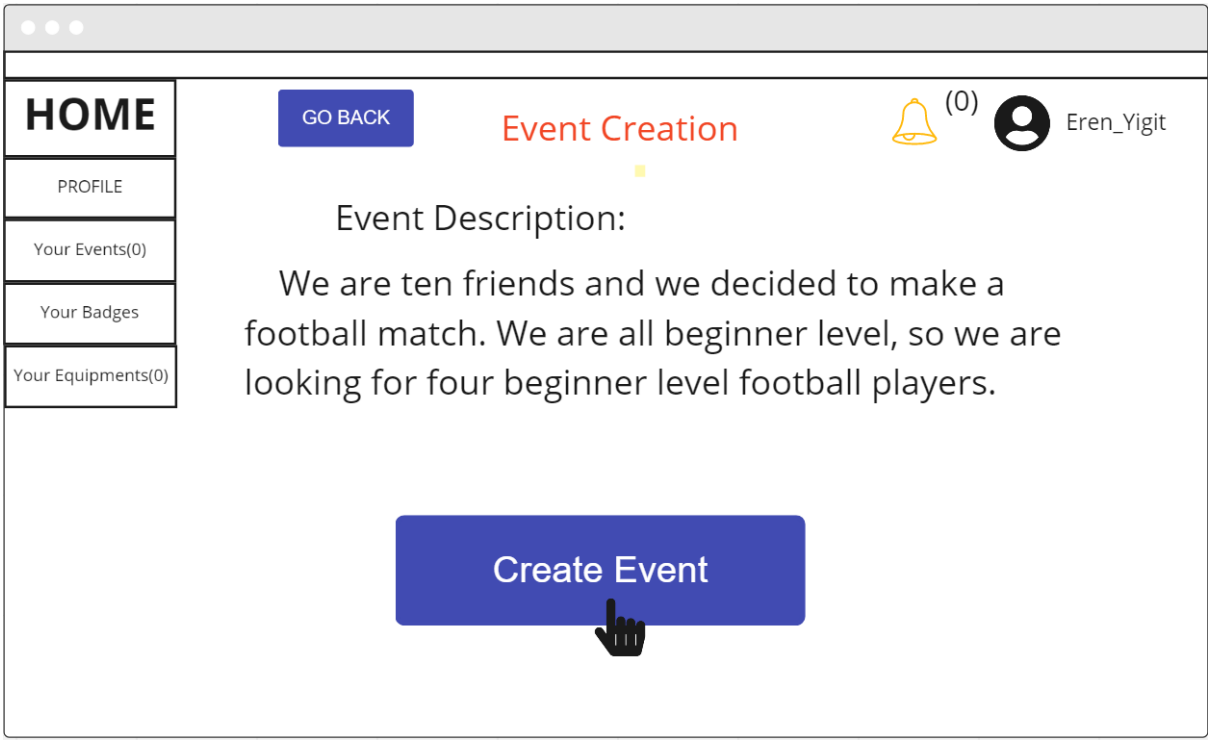
2. Eren specified sport type, date, hours and location.



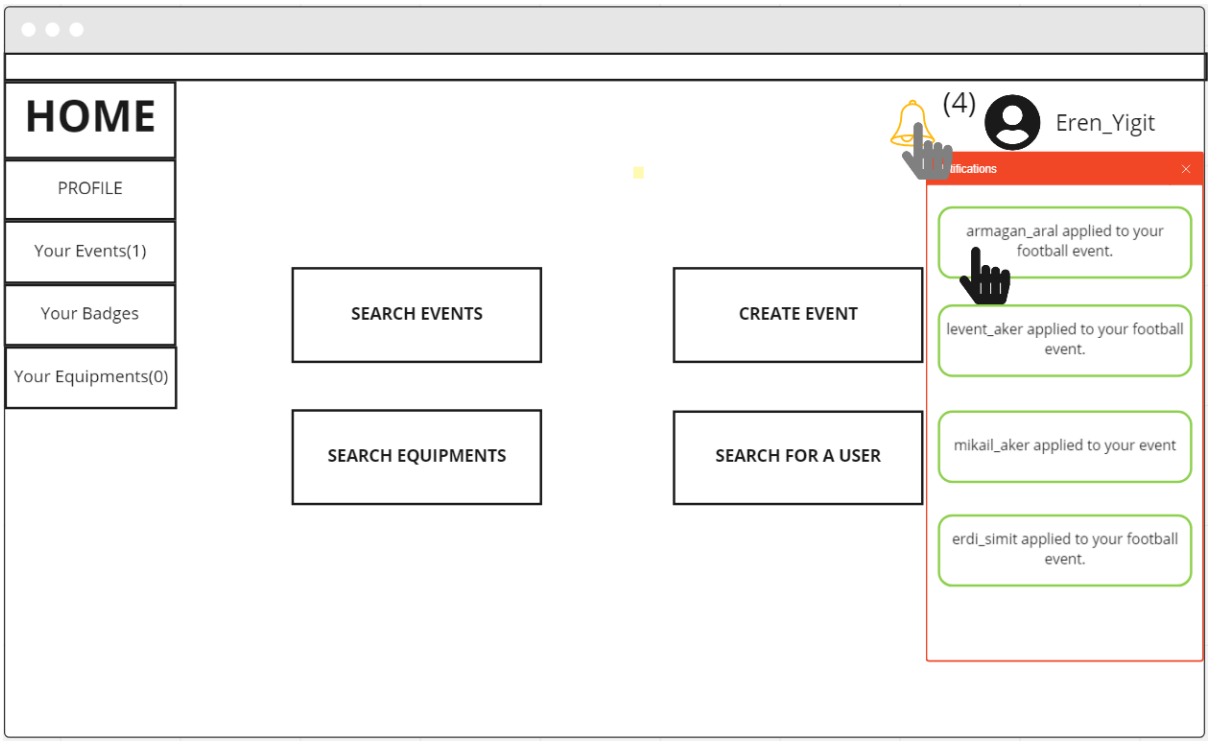
3. Eren specified skills, count, age, gender and role of the players.



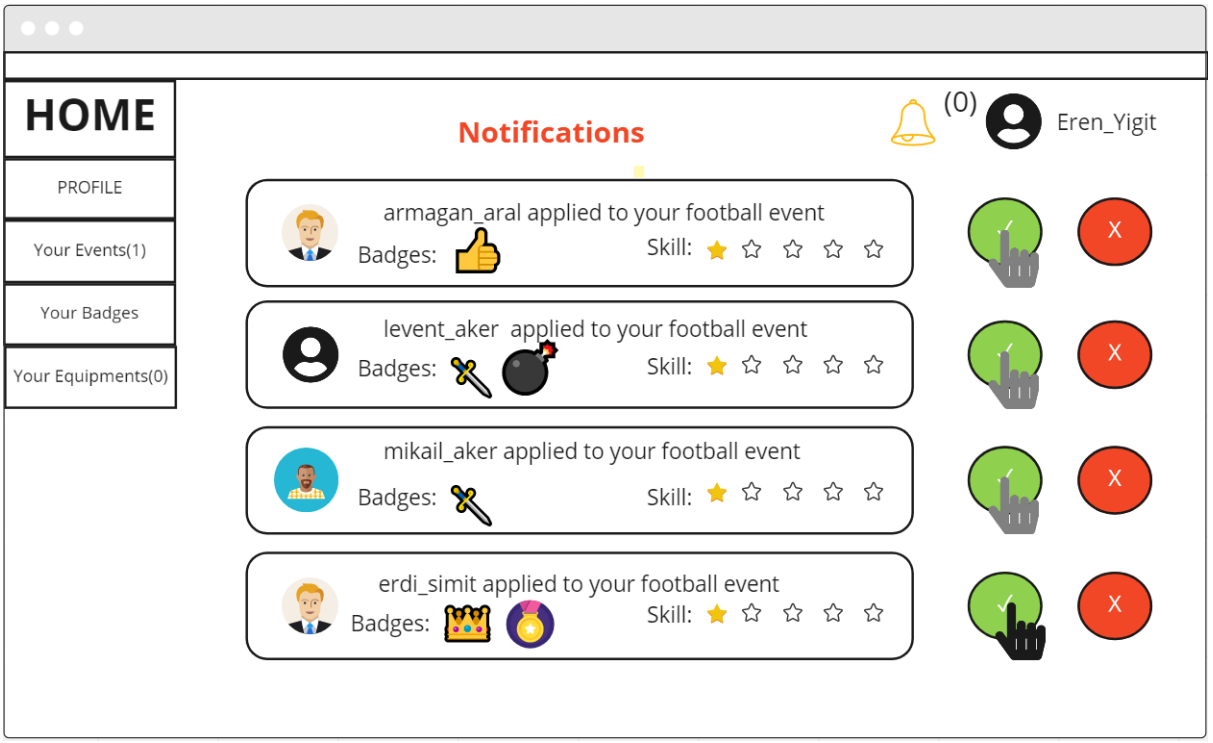
4. Eren described the event and made some comments about it.



5. After some time, Eren received some notifications and checked them.

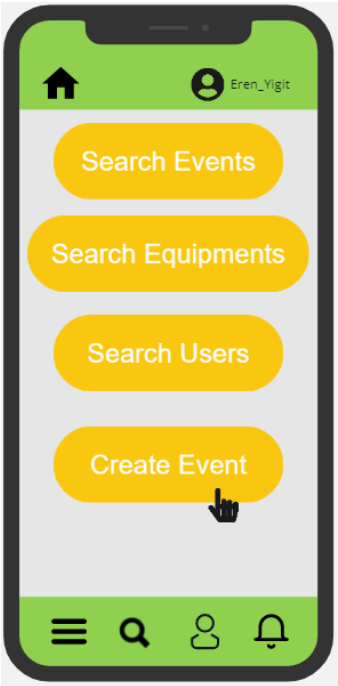


6. Eren analysed applicants and approved them if he wanted.

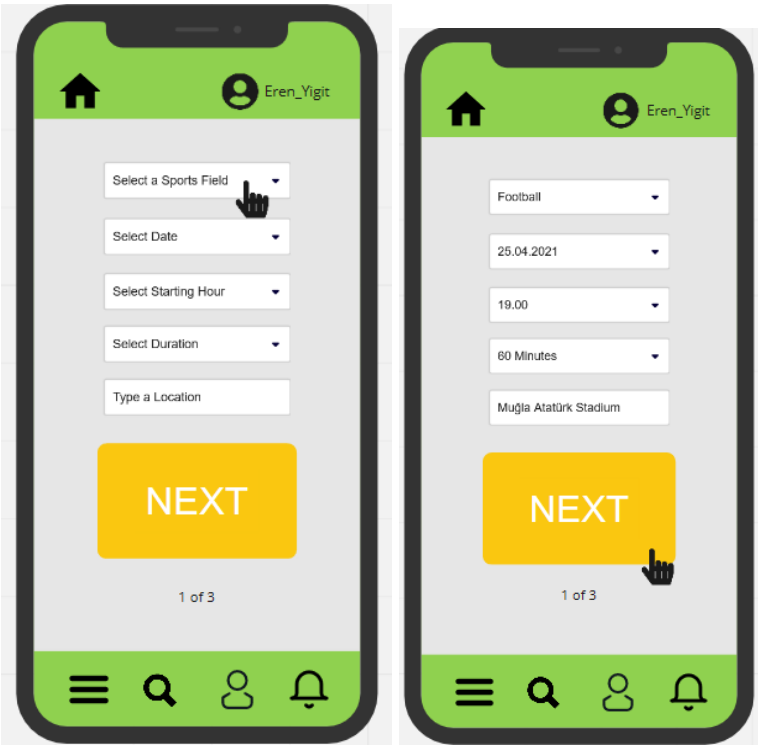


Android App

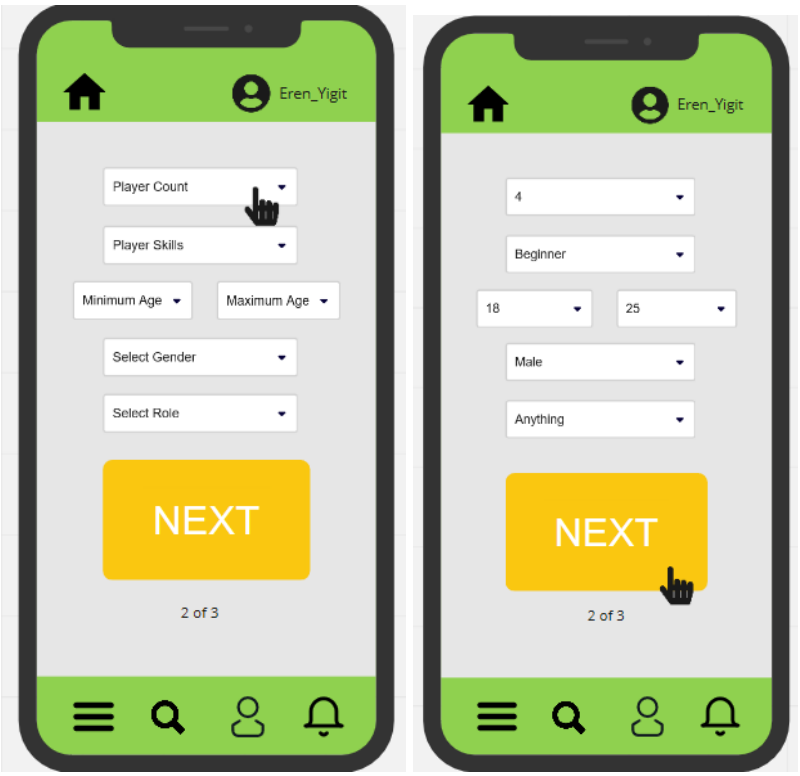
1. Eren clicked on create event button



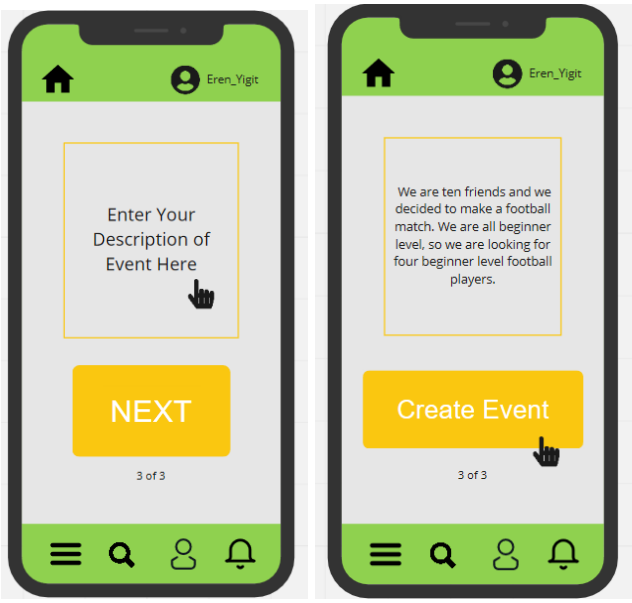
2. Eren specified sport type, date, hours and location



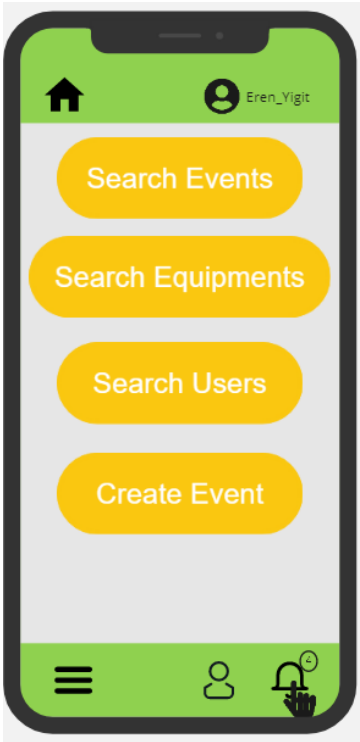
3. Eren specified skills, count, age, gender and role of the players.



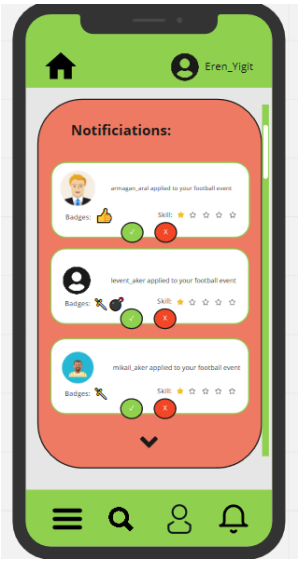
4. Eren described the event and made some comments about it.



5. After some time, Eren received some notifications and checked them.



6. Eren analysed applicants and approved them if he wanted.



2. User Searches for an Event

Persona:

Ali Topçu
40 years old
Father of two little boys
Used to work as an accountant, currently unemployed
Started gaining a little bit of weight
Galatasaray fan
Loves watching and playing football
His main role is defender

Story:
Currently unemployed and bored out of his mind, Ali wants to play football while he's still searching for a job. He thinks playing football also will help him lose some weight. Unfortunately he lost contact with his former football group, he wants to find people who love the sport as much as he does and play football with them.

Pre-conditions:
Ali is already registered as a user.
Ali has already stated his skills and interests in his profile.

Goals:

Ali wants to search for a football game he can play in.

Acceptance Criteria:

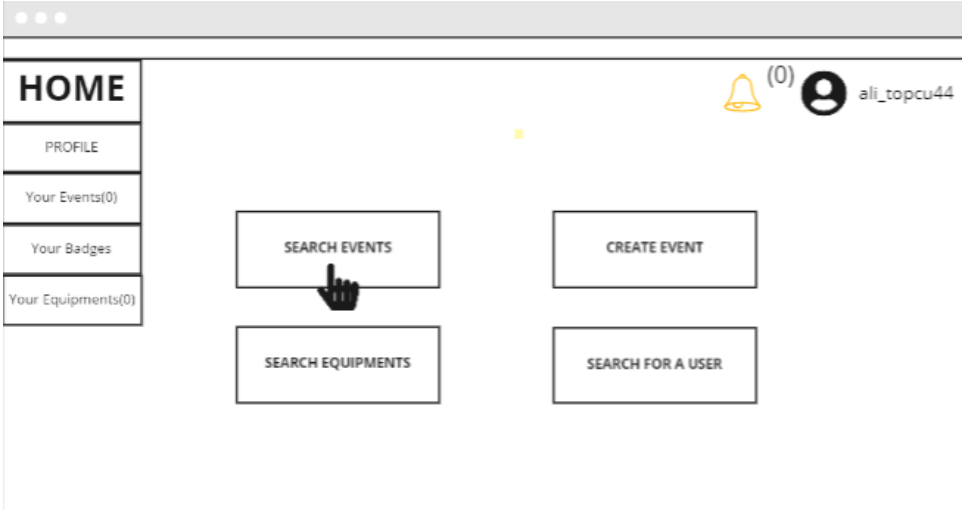
- (1.1.5.1.5) Users shall indicate the type of sport and location in order to filter events.
- (1.2.5.6.) All users shall be allowed to list all events on the events page.
- (1.2.5.7.) Event pages shall display the event creator, event location, sports field, applicants, participants and spectators of the event.
- (1.2.5.9.) System shall allow only registered users to apply to events.
- (1.2.6.1.1.) Events shall be filtered by user according to event type, appropriate age range, location, date, hours, skill levels of attendees and availability (full or player needed).
- (1.2.6.1.2.) Events shall be sorted with respect to name, time and location (e.g. from closer events to far away ones).

Scenario with Mockups:

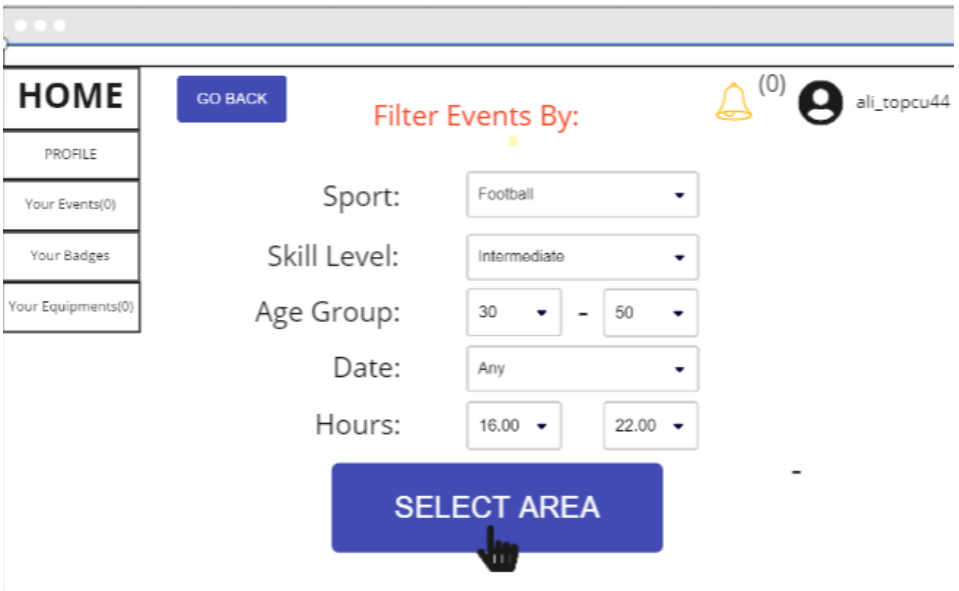
Ali wants to search for a game he can play in, therefore he clicks the "SEARCH EVENT" button.
Ali uses the filtering feature to find a game with the properties he wants.
After selecting the properties, Ali clicks "SELECT AREA" to define the area he wants to search.
He selects a circular area on the map and then clicks "SEARCH EVENTS".
After clicking "SEARCH EVENTS" he can browse the events with the properties he wants.

Web Mockup:

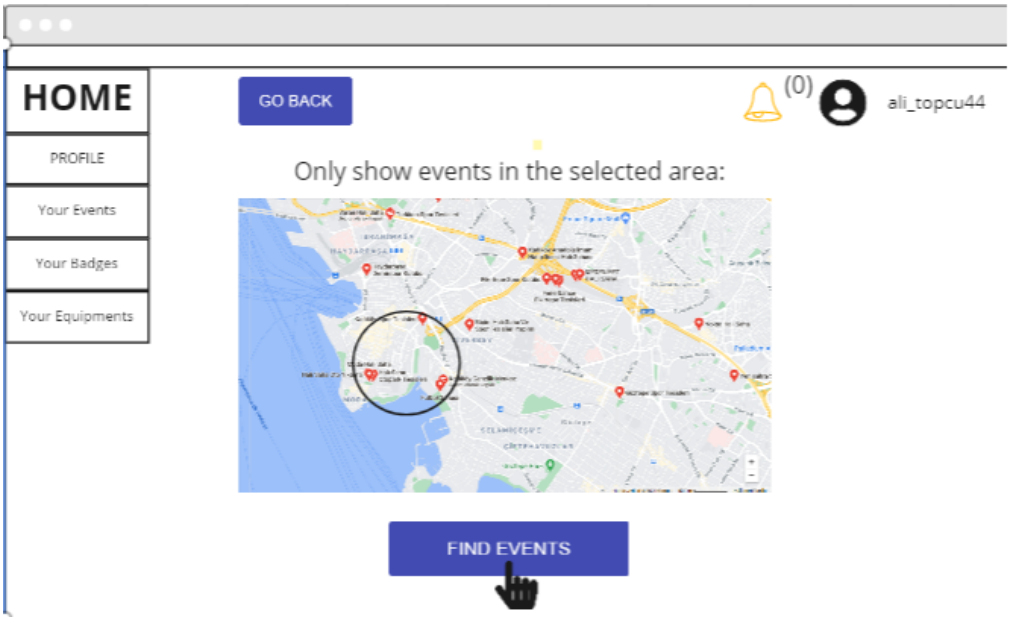
1. Home Page



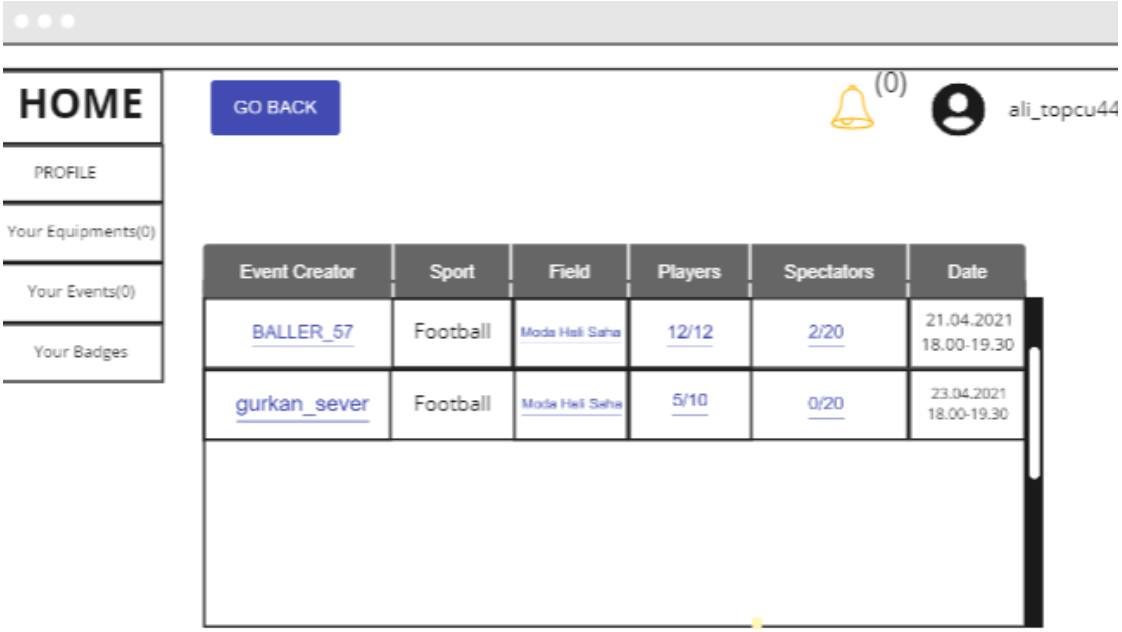
2. Filtering



3. Selecting the Area

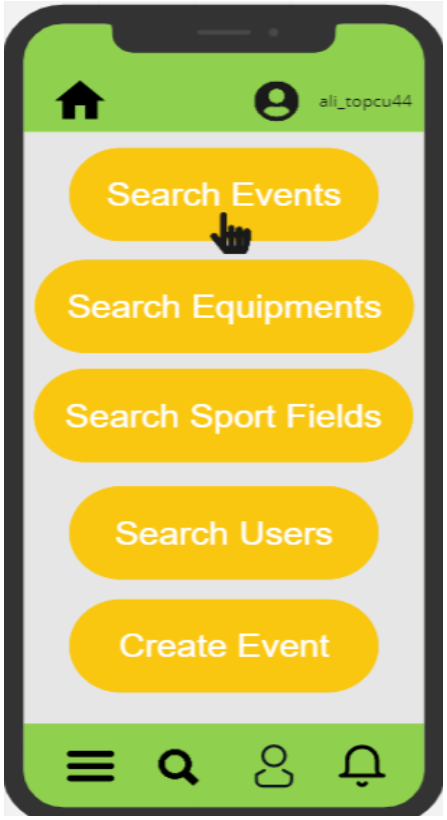


4. Search Results

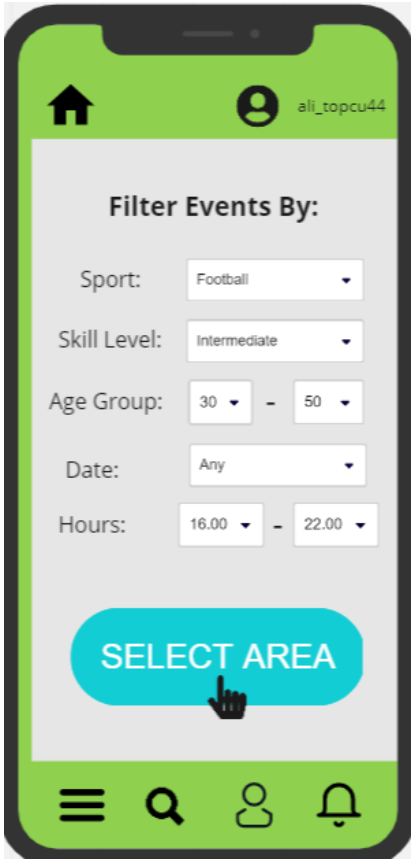


Android Mockup:

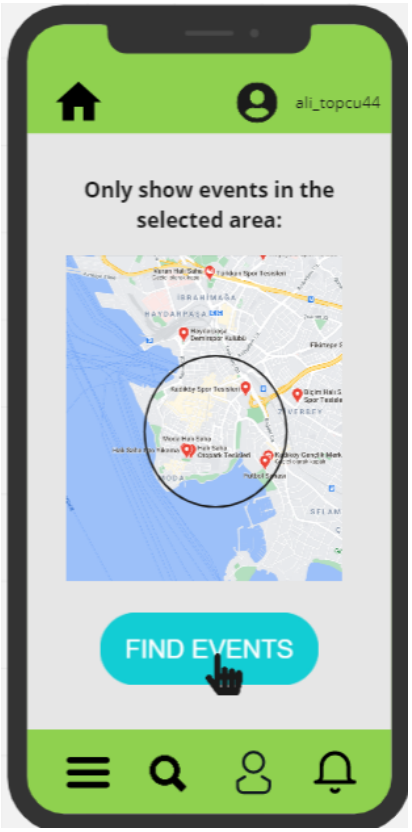
1. Home Page



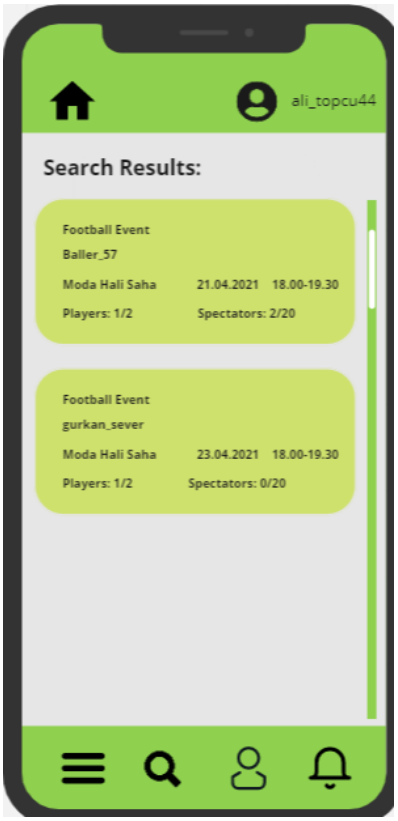
2. Filtering



3. Selecting the Area



4. Search Results



3. Player & Spectator Application

Persona:

- Melike Taşkıran
- 26 years old
- Music lover
- Has one cat and one dog
- Extrovert

Story:

Melike is an assistant brand manager at a leading company. Working remotely for more than one year, she feels unmotivated and bored. She wants to make some changes in her life by playing sports. Since her job allows her to work with a flexible schedule, Melike wants to play sports during the daytime. However she can't play with any of her friends as they are working at that time. She searches for a website that allows matchmaking with other interested people and encounters our platform. She decides to try our platform to make some new friends to play sports with. She wants to minimize the risk of being infected by avoiding team sports. So, she searches for tennis events nearby. Since she is a tennis newbie, she also wants to watch some tennis matches played by advanced players.

Pre-conditions:

- Melike is already registered as a user.
- Melike has already stated her skills and interests in her profile.
- Melike has already logged in to the system and searched for tennis events.

Goals:

- Melike wants to apply to an event to be a player.
- Melike wants to apply to an event to be a spectator.

Acceptance Criteria:

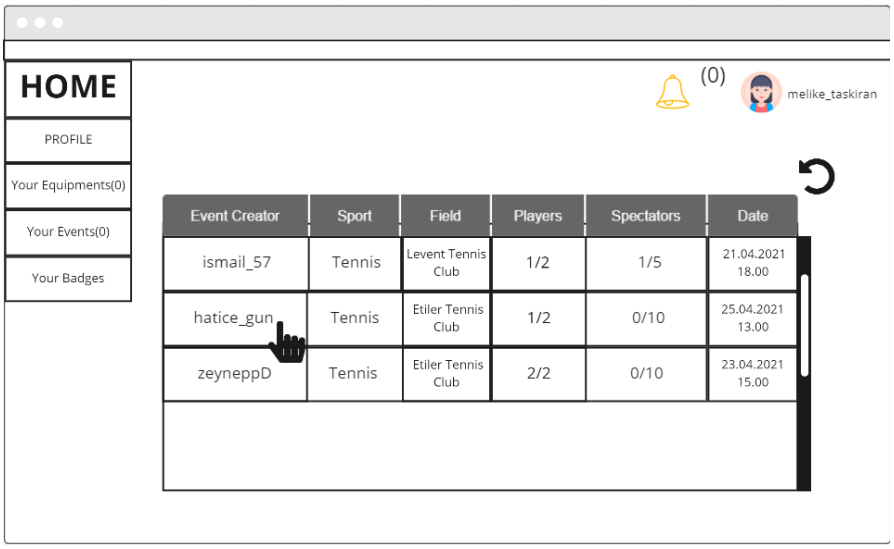
- (1.1.5.4.) All users shall be able to see the event pages that includes event creator, location, participants and badges.
- (1.1.5.5.) All users shall be able to see the profile pages of the event participants.
- (1.1.5.6.) Registered users shall be able to apply to join an event.
- (1.1.7.4.) All participants shall be informed whether he/she is selected.

- (1.2.4.3.) User profile shall display personal information, skills, badges, followers, followings and post history.
- (1.2.10.1.) System shall notify users when their player/spectator applications are approved/rejected.
- (2.4.3.) Users shall be able to choose which of their personal data to be seen publicly.

Scenario with Mockups:

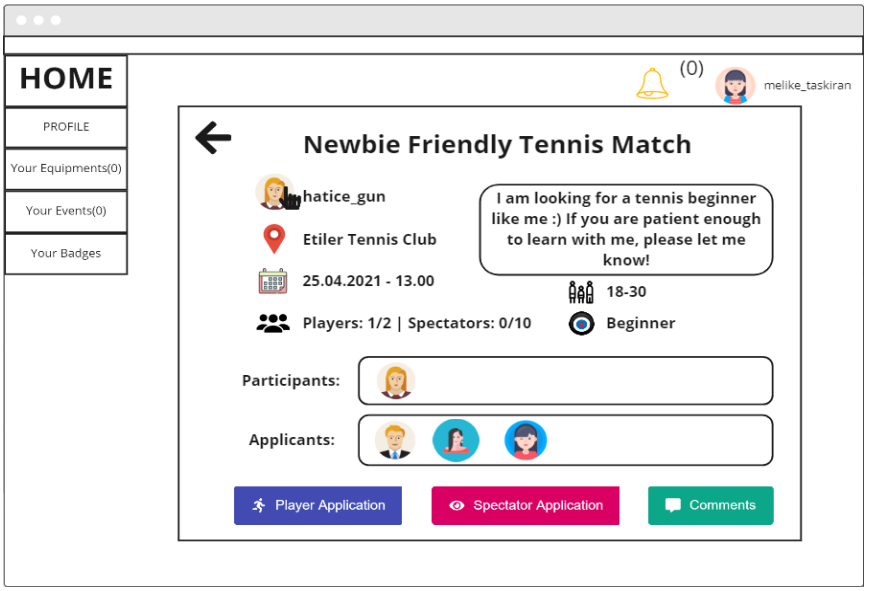
Web Application:

She has already searched for the events she is interested in and she is in the search results page. She likes the date and the location of the second event, so she clicks on the event to get to the event page.

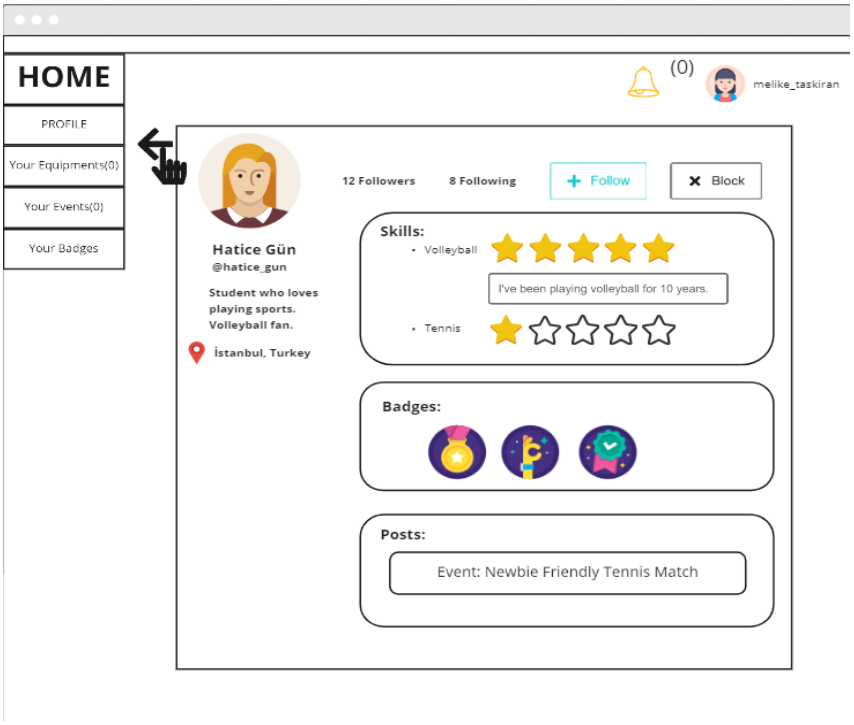


She is happy that the event creator is looking for a beginner player like her. But, she wants to learn more about the event creator before she applies, so she clicks on the

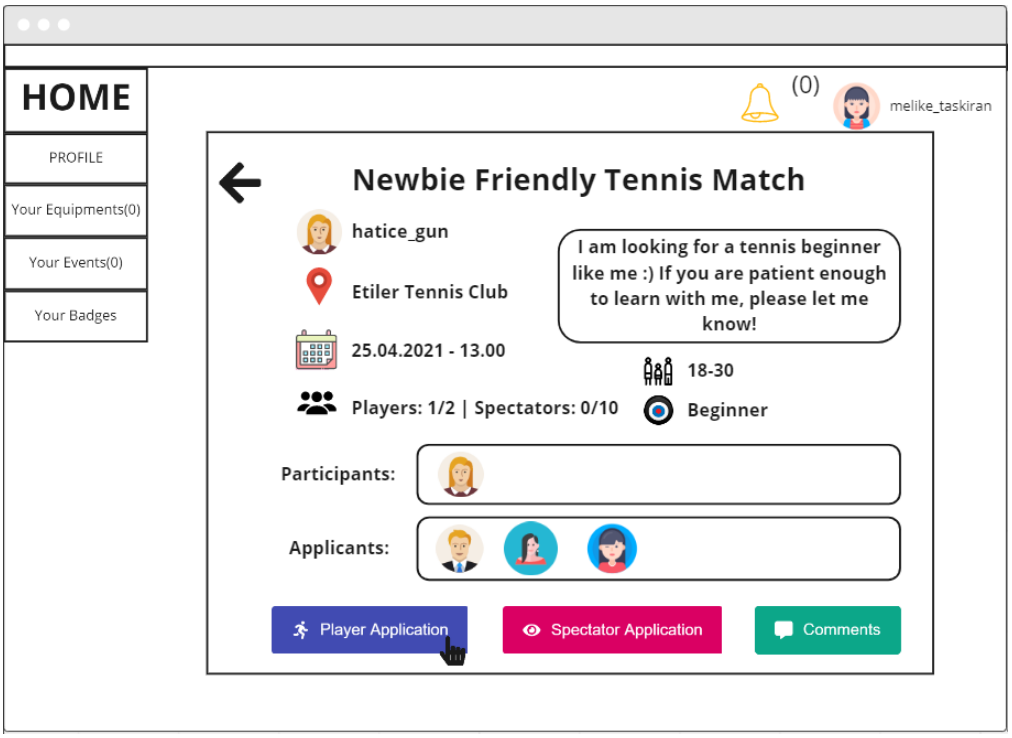
icon of the event creator to get to her profile page.



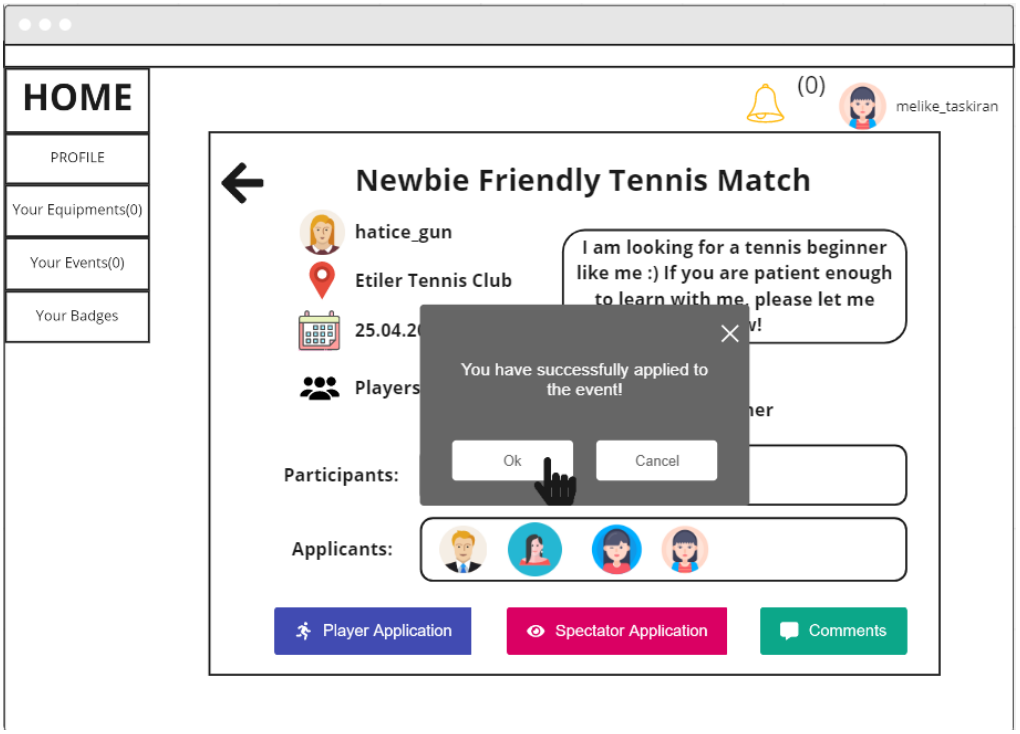
She notices that the event creator decided to hide their age so she can't get an idea from there. However, she also notices that the event creator has some good badges, therefore she thinks that it would be a good idea to apply to her event. She gets back to the event page by clicking the go back button.



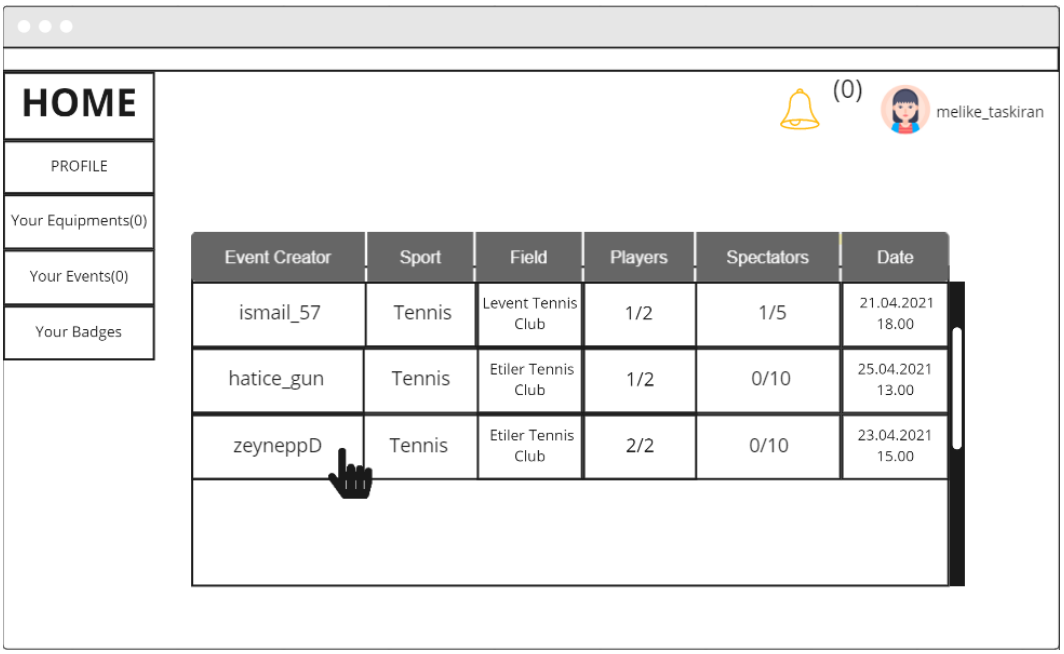
Now that she is in the event page and she is interested, she clicks on the player application button to apply.



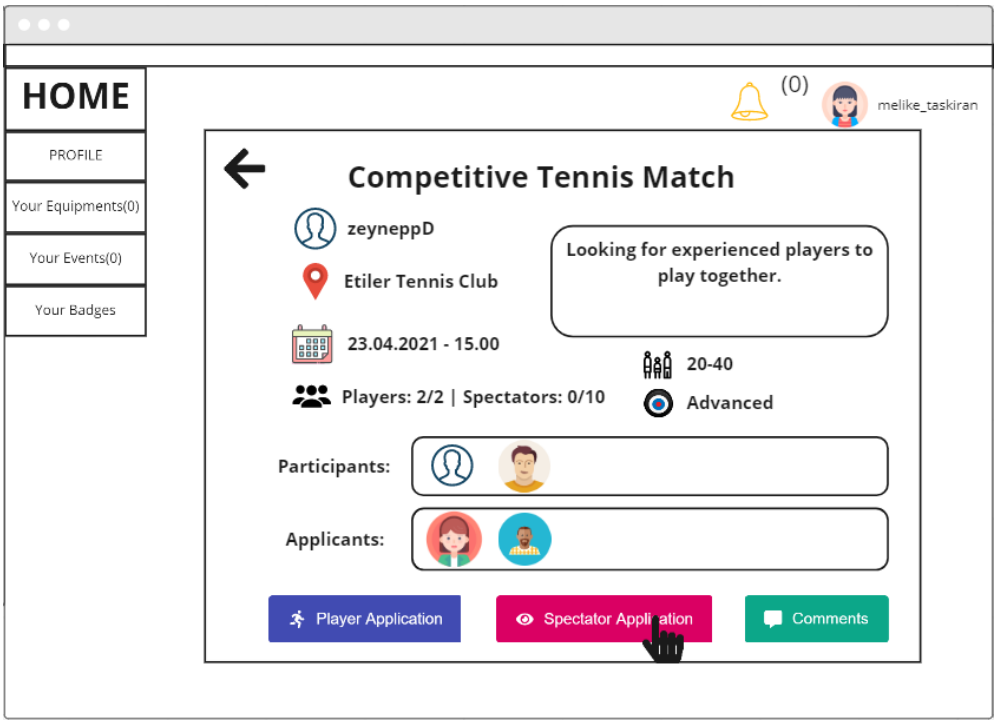
A popup page shows up to inform that she successfully applied to be a player. Also, she is added to the applicants list.



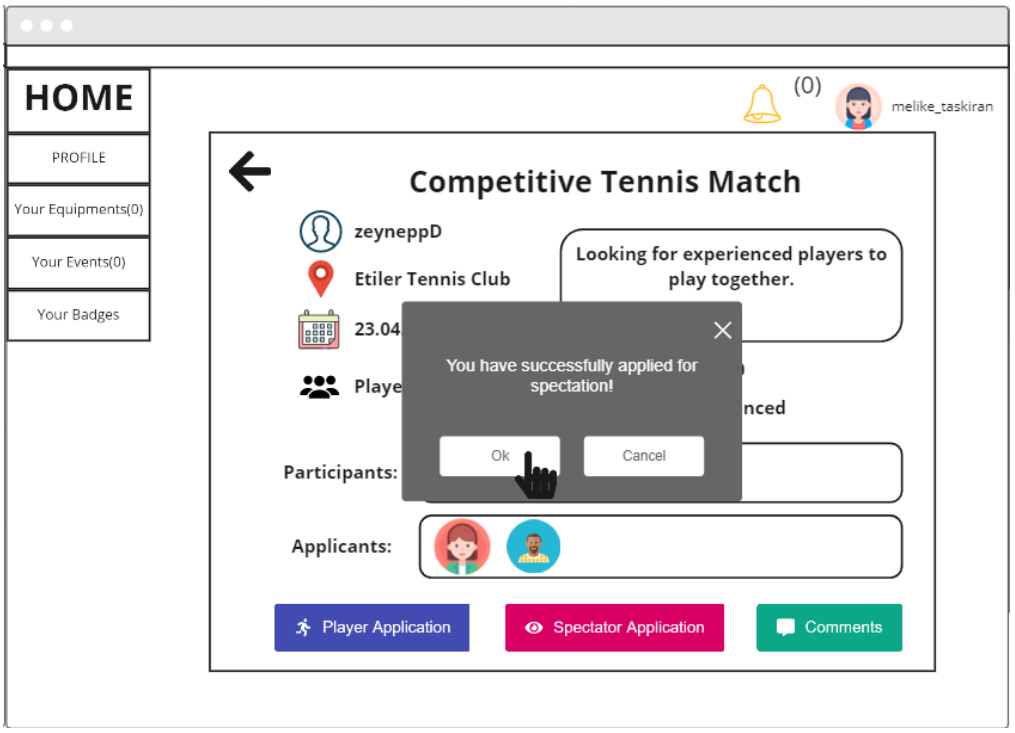
Now, Melike wants to apply to an event to be a spectator and she gets back to the search results. The last event seems reasonable to spectate as it is full. Melike clicks on the event.



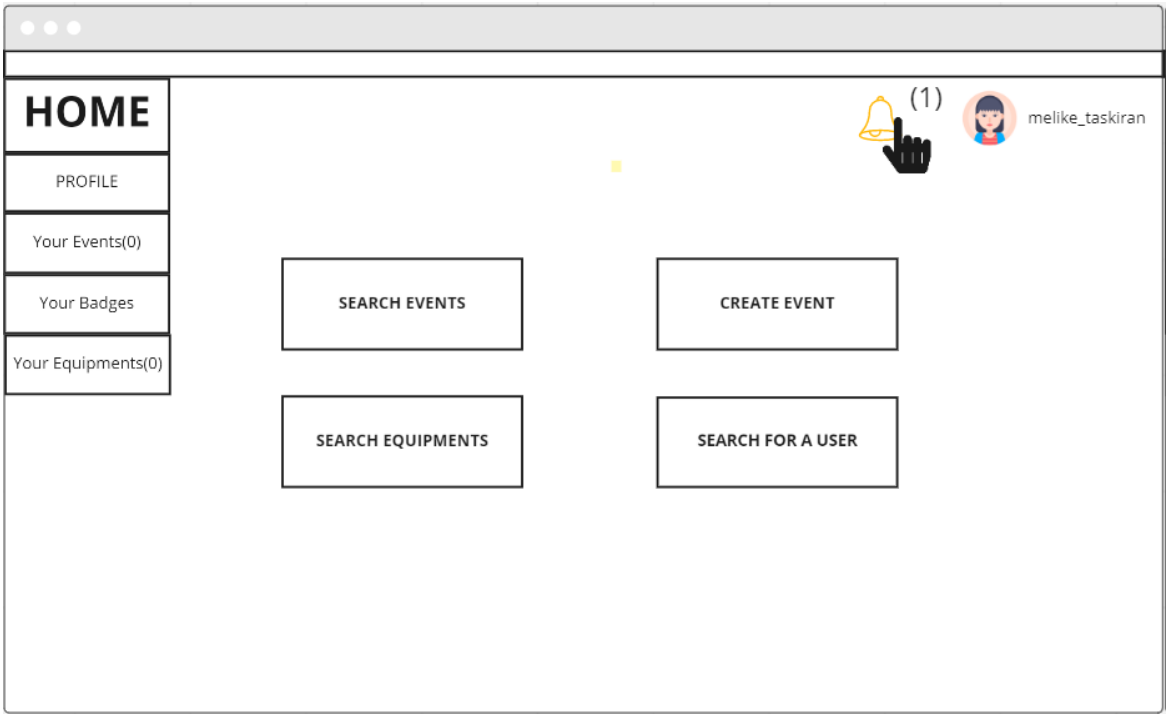
Melike reads the description and sees that the event creator is looking for advanced players. So, she thinks this match could be very useful to observe and clicks the spectator application.



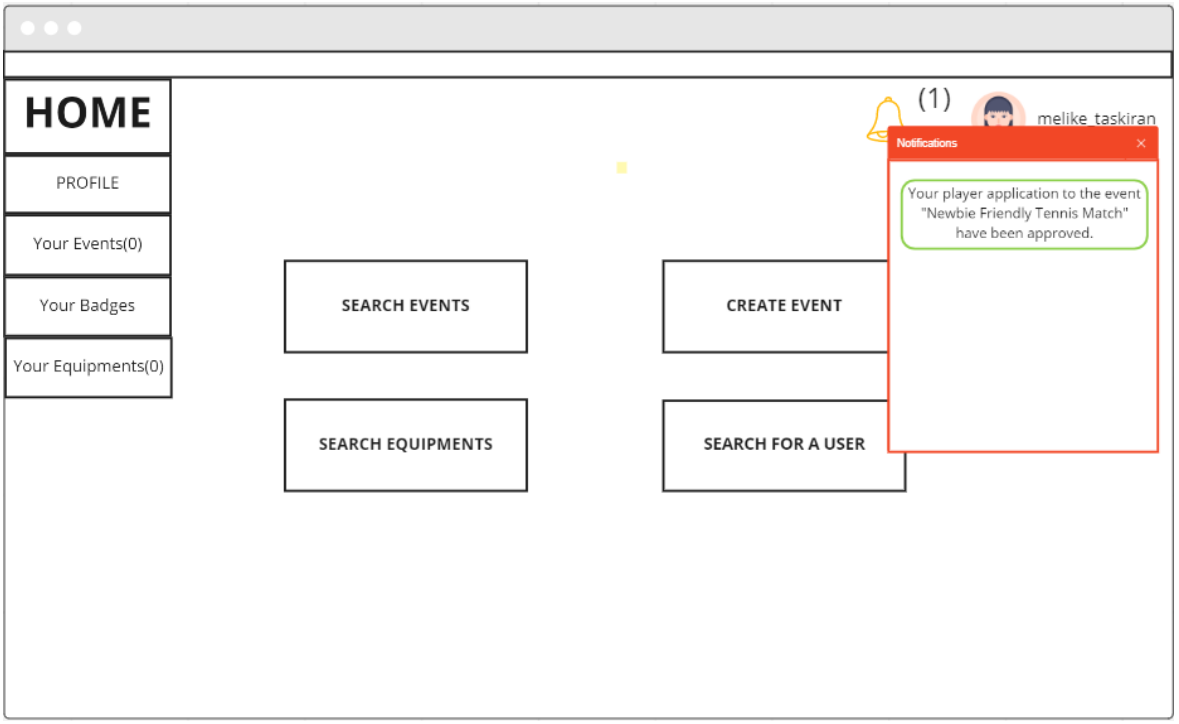
A popup page shows up to inform that she successfully applied to be a spectator.

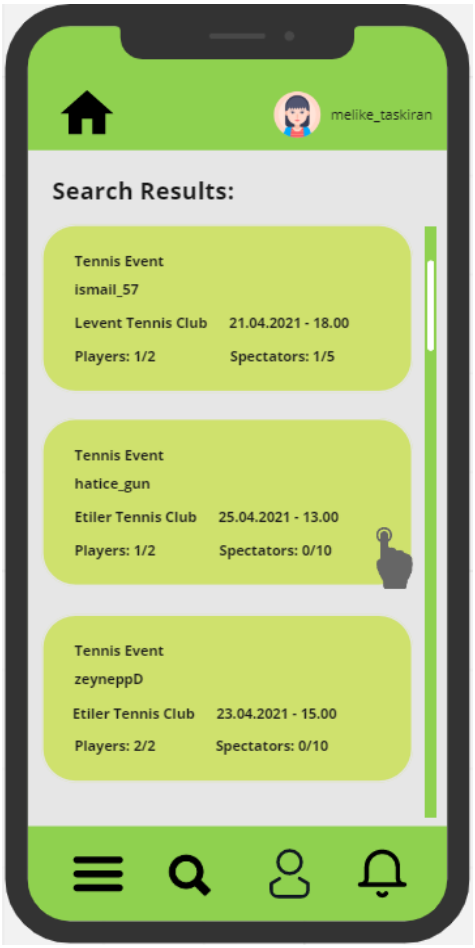


Some time later, she enters the app again and notices that she received a notification. So, she clicks the notification button.



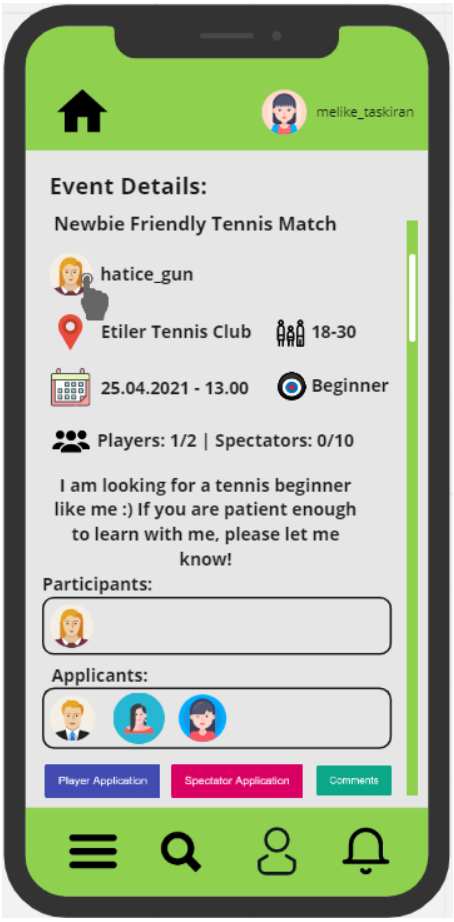
Seeing her player application approved, Melike is now waiting for her tennis match!



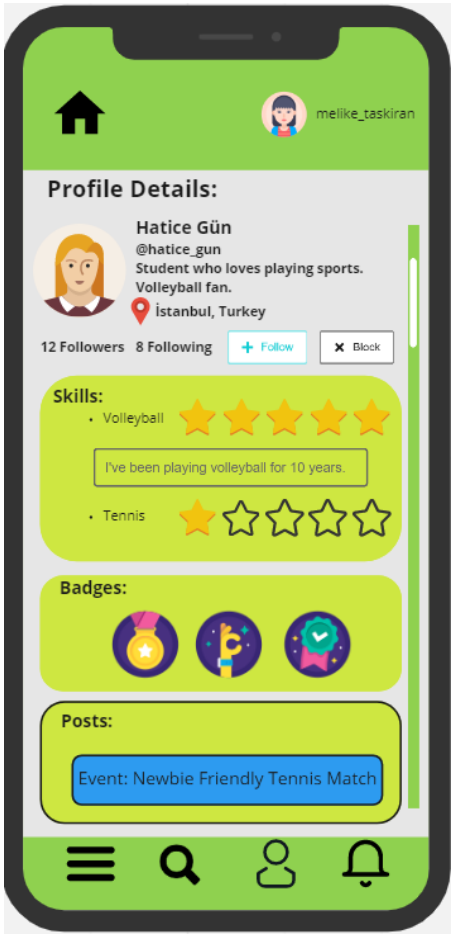


Android Application:

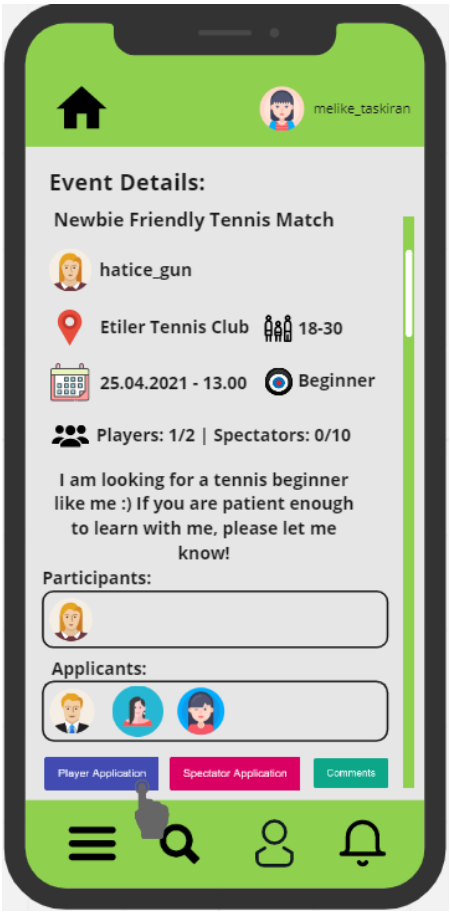
She has already searched for the events she is interested in and she is in the search results page. She likes the date and the location of the second event, so she clicks on the event to get to the event page.



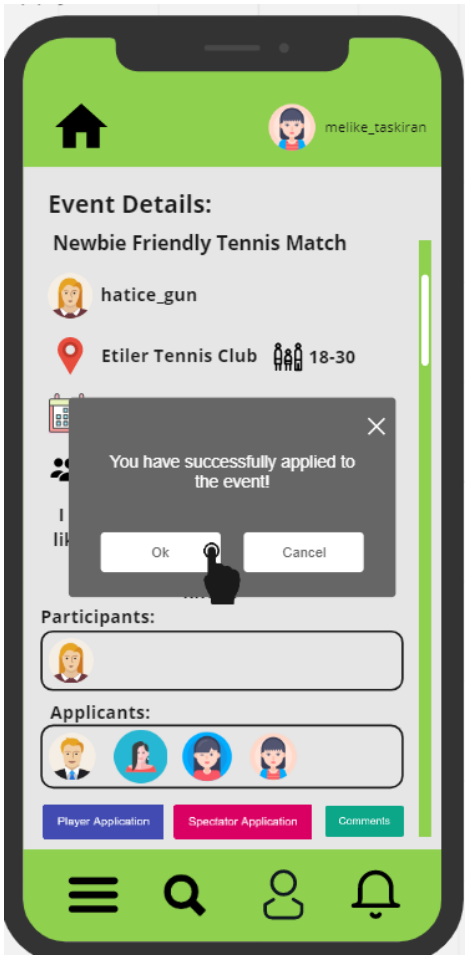
She is happy that the event creator is looking for a beginner player like her. But, she wants to learn more about the event creator before she applies, so she clicks on the icon of the event creator to get to her profile page.



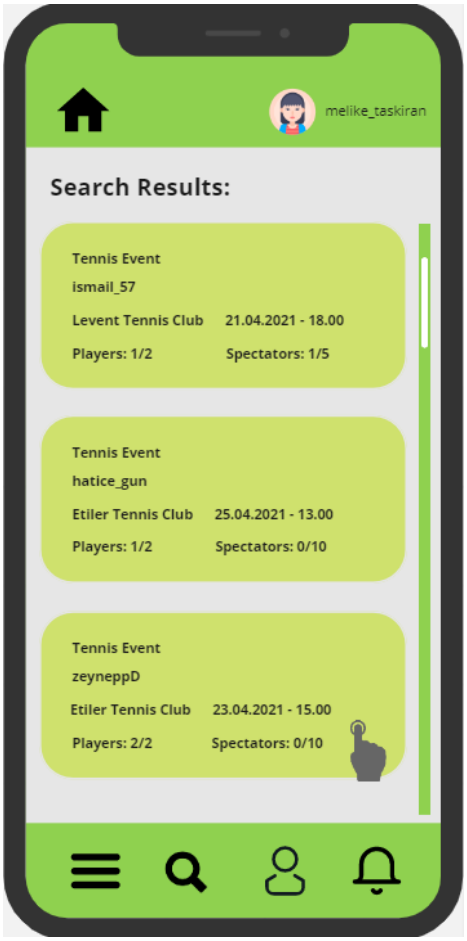
She notices that the event creator has some good badges, therefore she thinks that it would be a good idea to apply to her event. She gets back to the event page.



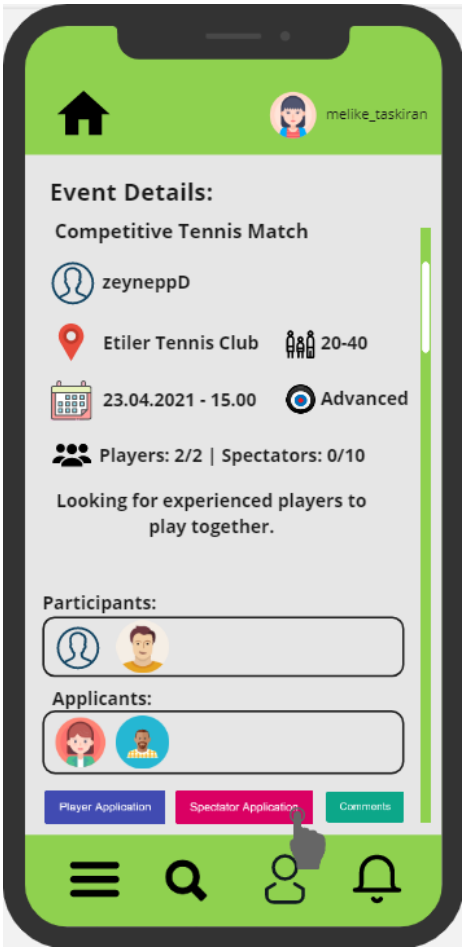
Now that she is in the event page and she is interested, she clicks on the player application button to apply.



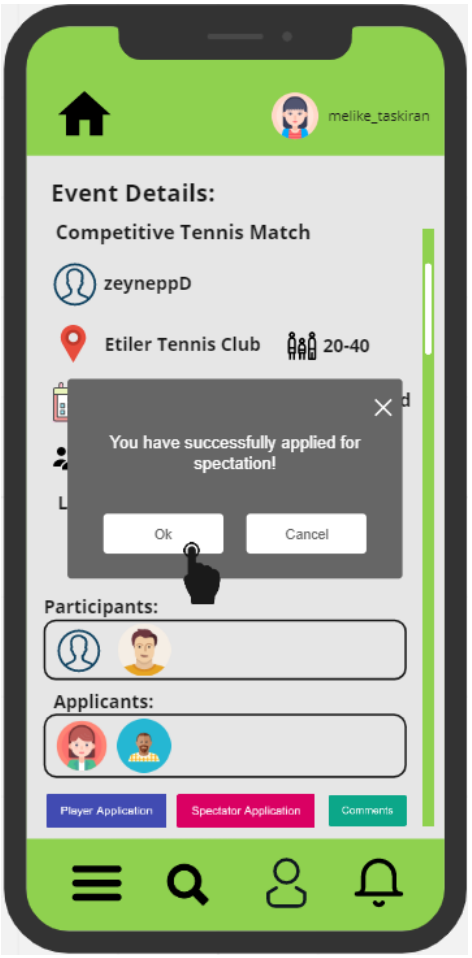
A popup page shows up to inform that she successfully applied to be a player. Also, she is added to the applicants list.



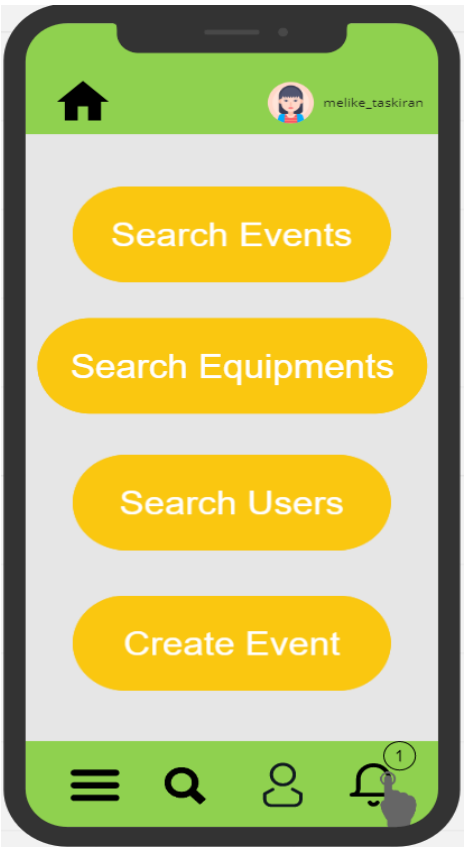
Now, Melike wants to apply to an event to be a spectator and she gets back to the search results. The last event seems reasonable to spectate as it is full. Melike clicks on the event.



Melike reads the description and sees that the event creator is looking for advanced players. So, she thinks this match could be very useful to observe and clicks the spectator application.



A popup page shows up to inform that she successfully applied to be a spectator.



Sometime later, she enters the app again and notices that she received a notification. So, she clicks the notification button.

Seeing her player application approved, Melike is now waiting for her tennis match!

4. Event Creator Selecting from Applicants

Persona

Fatih Kerim

33 years old

Grocery store owner

Loves watching amateur football games

Has a limp on left leg

Avid player of PES and NBA2K console games

Addicted to sports betting

Lives with his mother

Story

Fatih loves watching football and basketball games, both professional and amateur. He was also an amateur football player in his youth, aspiring to move on to higher professional leagues. However, his career ended due to a permanent injury on his left leg caused by a construction machinery accident he had while he was working as a construction worker to support his family. Since then, he cannot play football but loves organizing amateur matches and watching them. He now dreams of being a coach. He hears of our platform from a friend who tells him it might be of interest to him. Without much reading, he registers to the platform and creates a 6v6 football match event at a halı saha. After a day of waiting, he now has enough applications to his event to review and select 12 people.

Pre-conditions

Fatih has registered as a user.
Fatih has signed in to the platform.
Fatih has created his profile, filled in the relevant fields (skills and interests)
Fatih has created an event by specifying relevant properties (sports type, date and time, location, skill level, age, gender, role and number of players)
Fatih has received enough applications to fill the spots in his event

Goals

Fatih wants to create 2 teams of 6 by reviewing the applications he has received and approving 12 of them that he sees fit

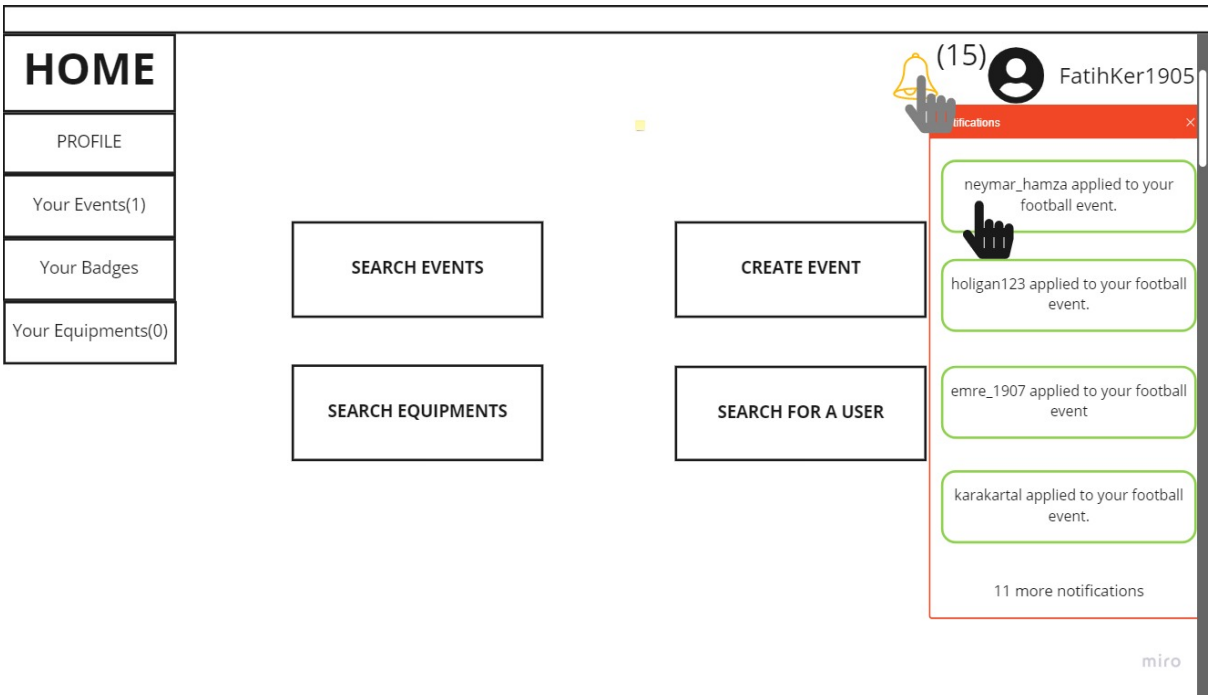
Acceptance Criteria

- (1.1.6.2.) Event creators shall select players among the applicants.
- (1.2.5.8.) System shall allow all users to view the profile pages of the event creator, any of the applicants, participants and the spectators.
- (1.2.13.2.) System shall allow users to see the analytics for players, event creators and sports fields.

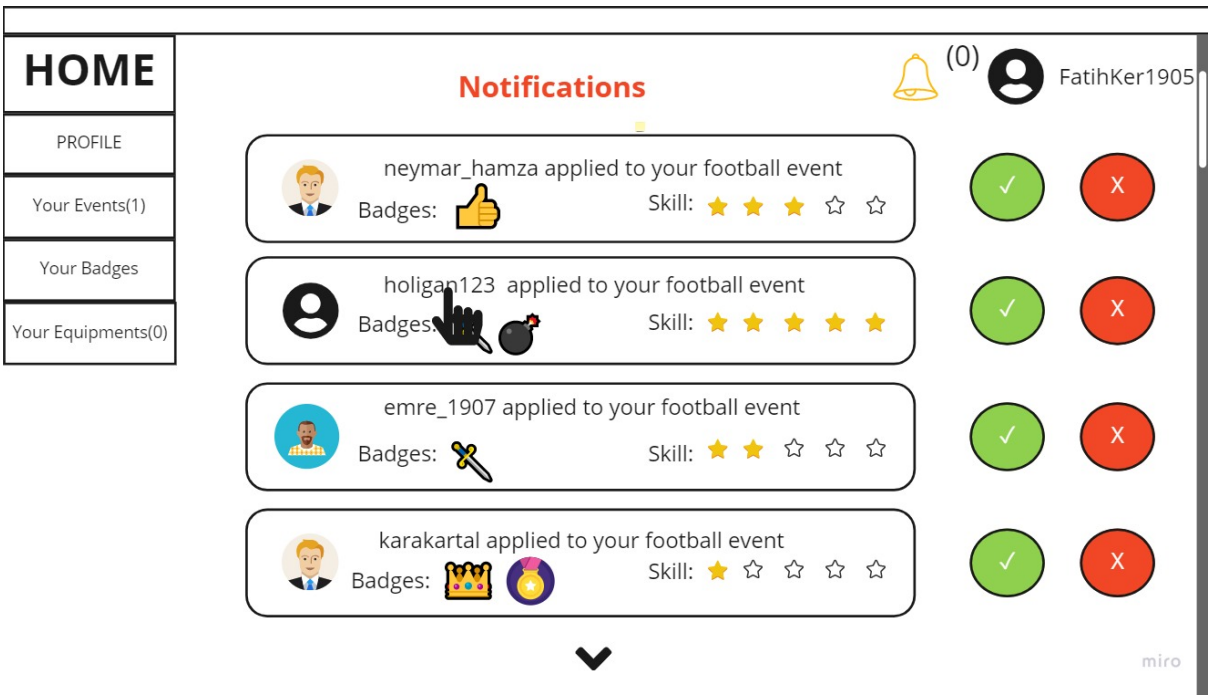
Scenario with Mockups

Web

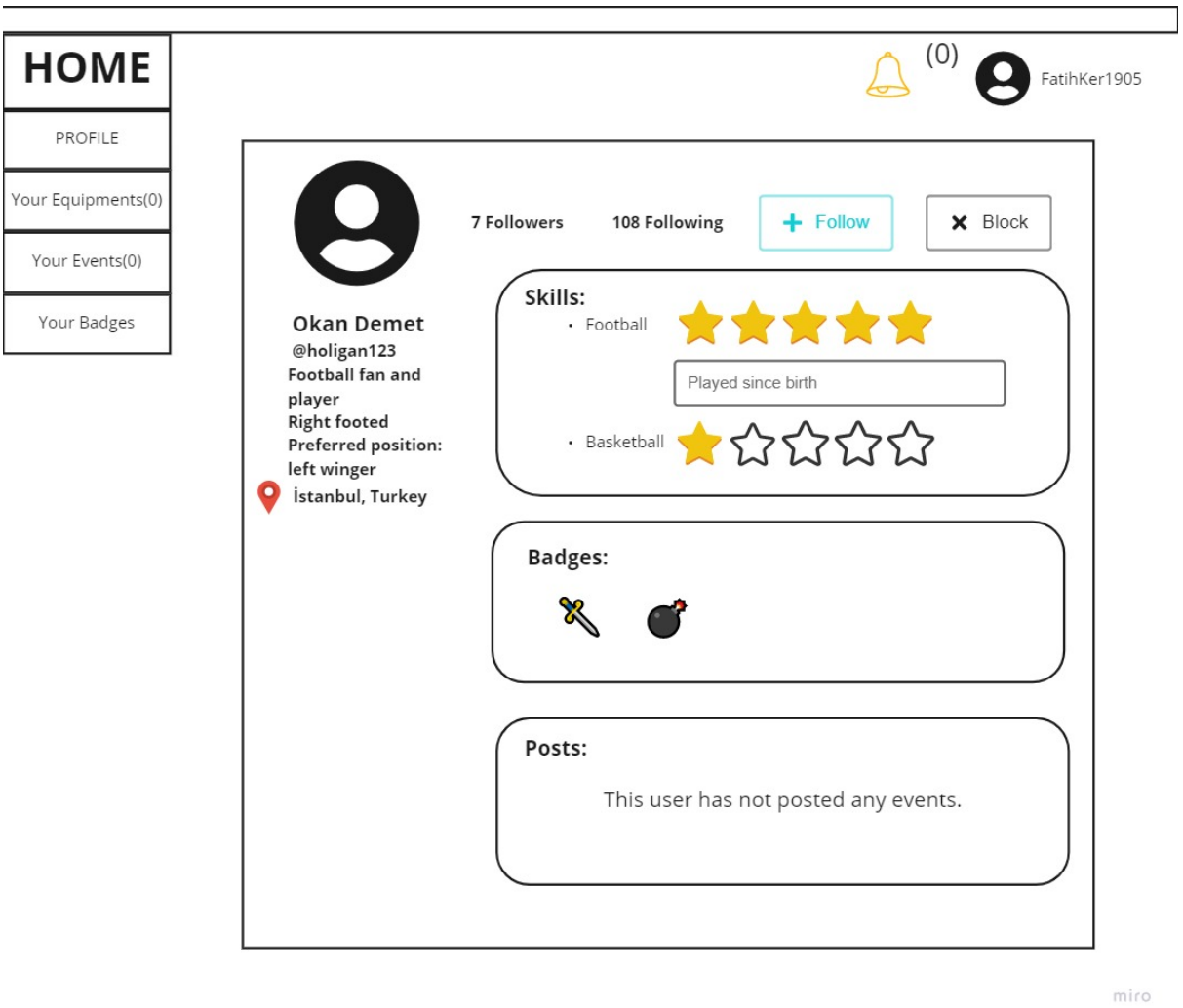
1. Checking application notification from homepage



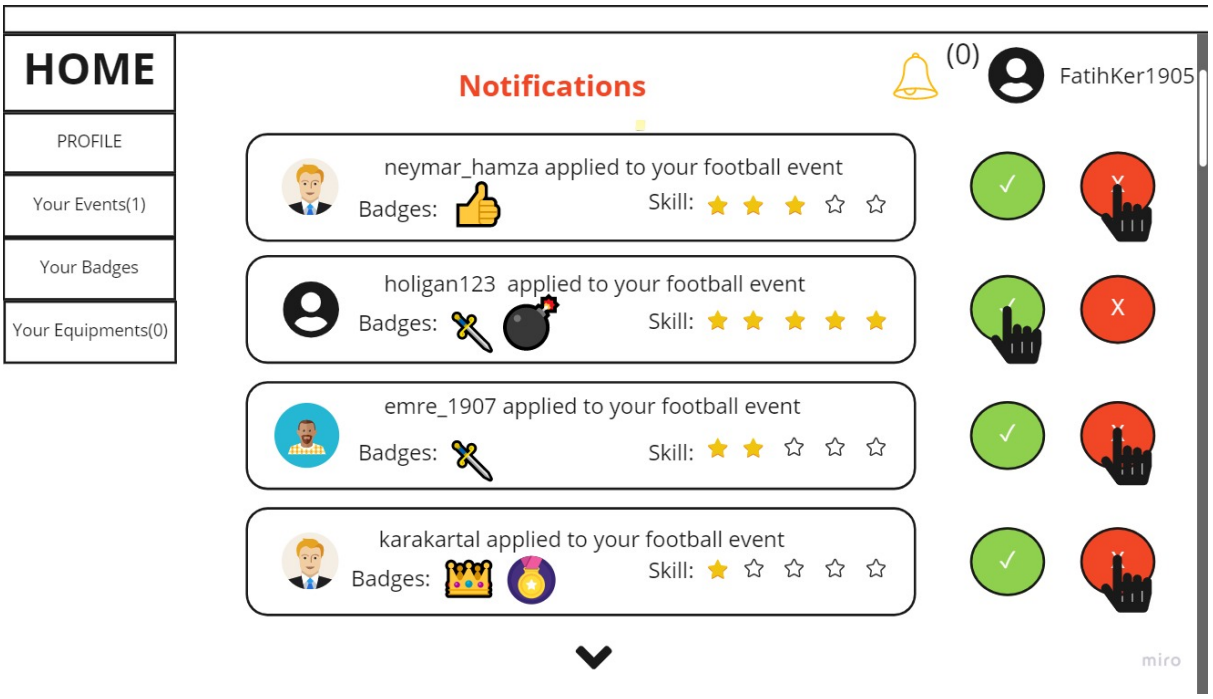
2. Seeing applicants from notifications page and clicking on applicant profile



3. Checking the profile page of applicant

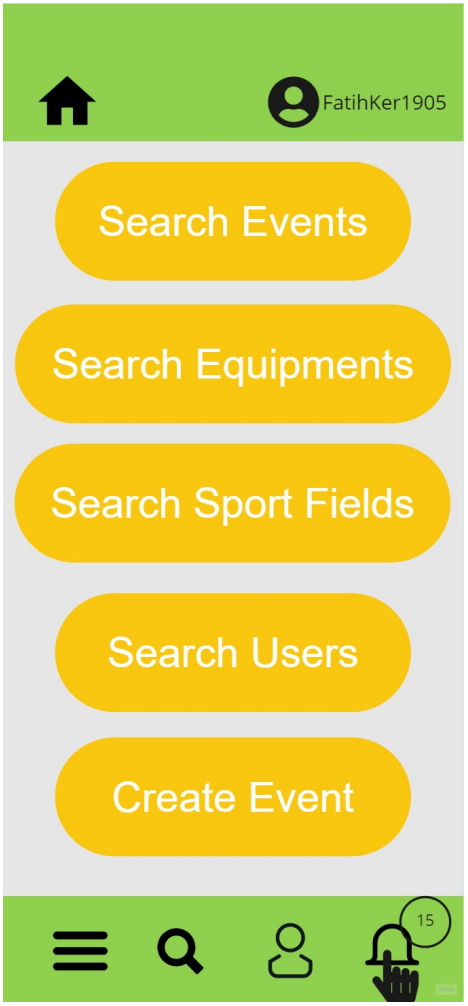


4. Accepting or declining applicants

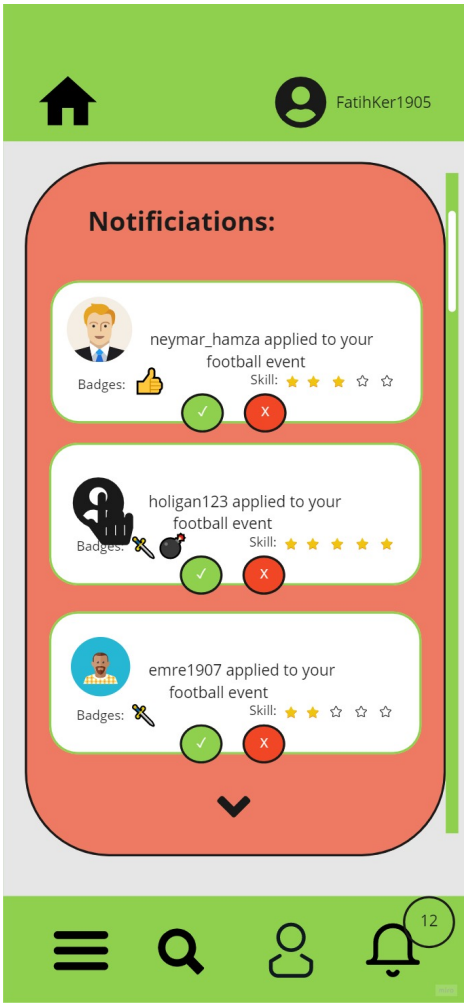


Android

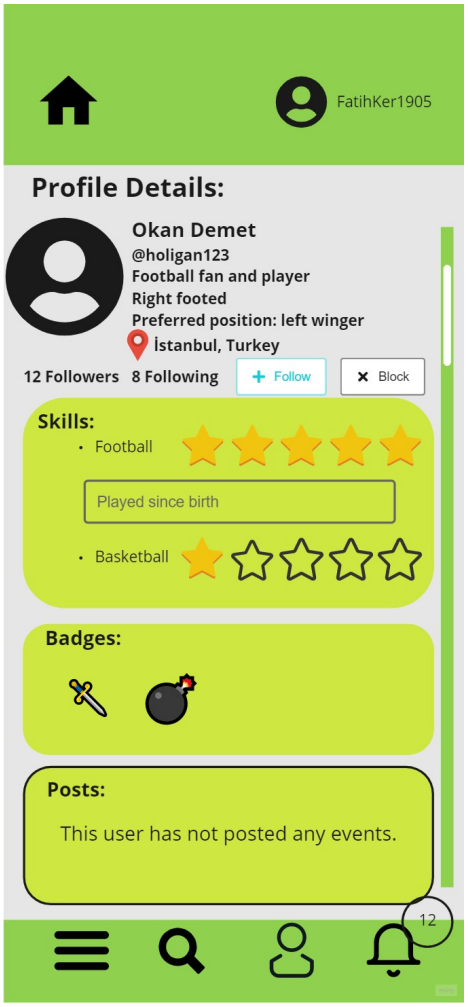
1. Checking application notification from homepage



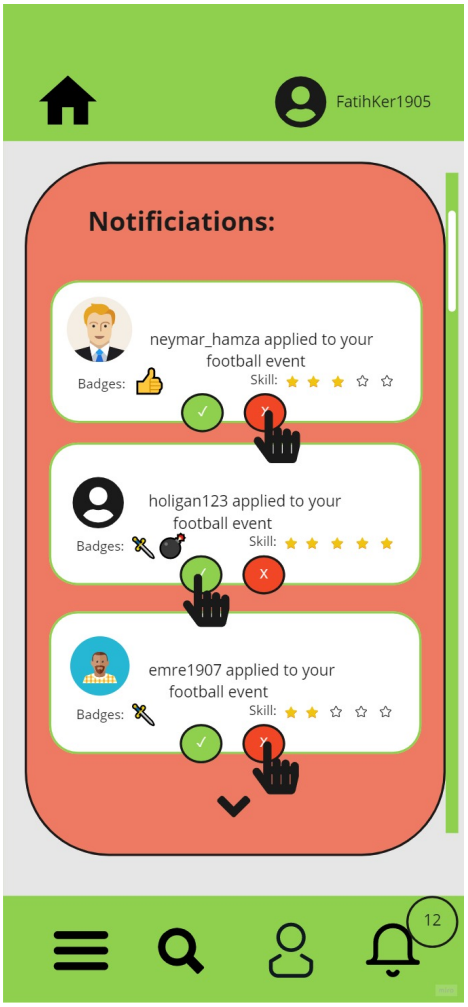
2. Seeing applicants from notifications page and clicking on applicant profile



3. Checking the profile page of applicant



4. Accepting or declining applicants



5. Discussion

Persona:
Zülfikar Tunç
20 years old
University student
Has been staying at home for one year because of the pandemic
Tired of being inactive and wants to do sports
Roger Federer fan
Likes watching tennis and wants to start playing

Story:
Online education made him really bored and he wants to join activities which he can go outside and enjoy. During quarantine, he started watching tennis games and he liked this sport. After the lockdown ended he decided to try playing tennis and found a good event online to join. He will apply for the event but he does not have a tennis racket.

Pre-conditions:

Zülfikar is already registered as a user.

Zülfikar has already found a game and opened the game page.

Goals

Zülfikar wants to ask the event creator if he can borrow a tennis racket.

Acceptance Criteria:

- (1.2.9.1.) System shall allow users to discuss under event or equipment post by writing comments.
- (1.2.9.2.) System shall also allow users to answer/reply discussion comments.
- (1.2.9.3.) Users shall not be able to reply to the answers/replies, thus replies shall be the deepest level of conversations.

Scenario with Mockups:

Zülfikar is on the event's page and he wants to ask a question.

Zülfikar opens the discussion page by clicking the discussion button.

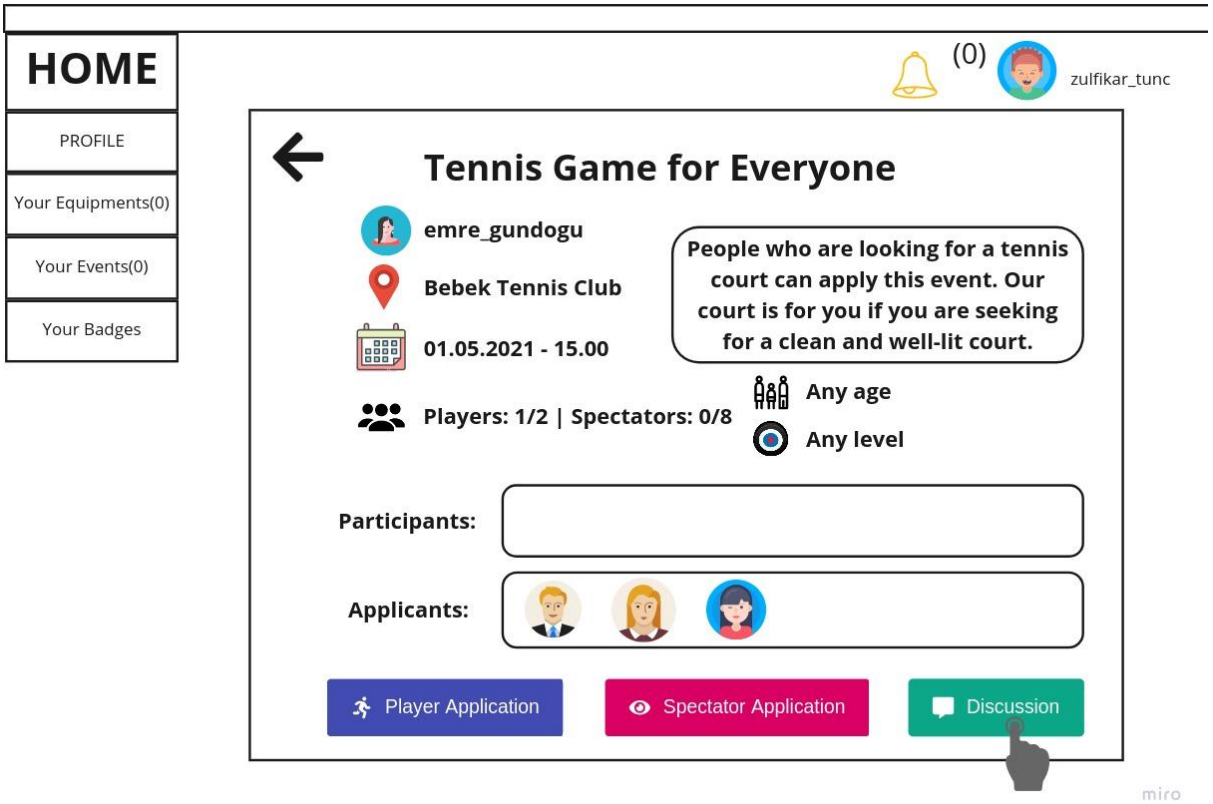
Zülfikar clicks the text field and writes there.

Zülfikar clicks the send button.

The message is posted.

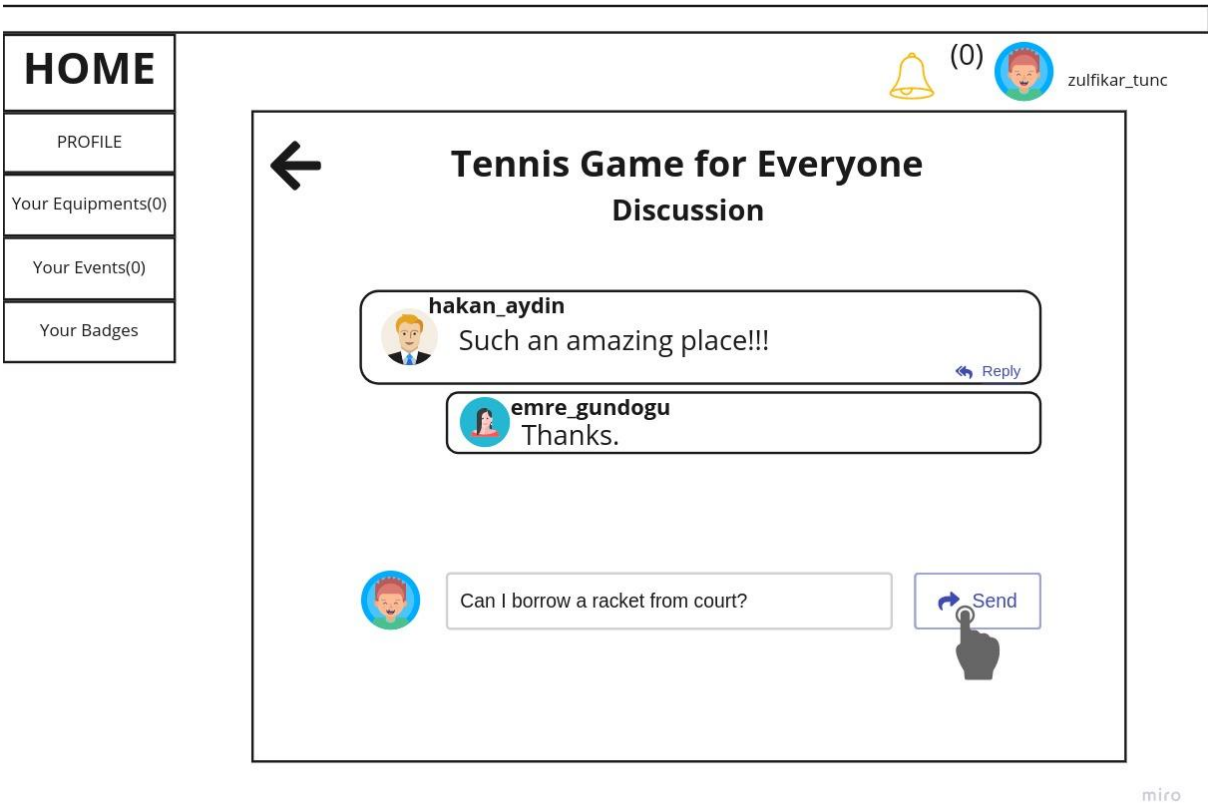
Web Mockup

1. Event Page

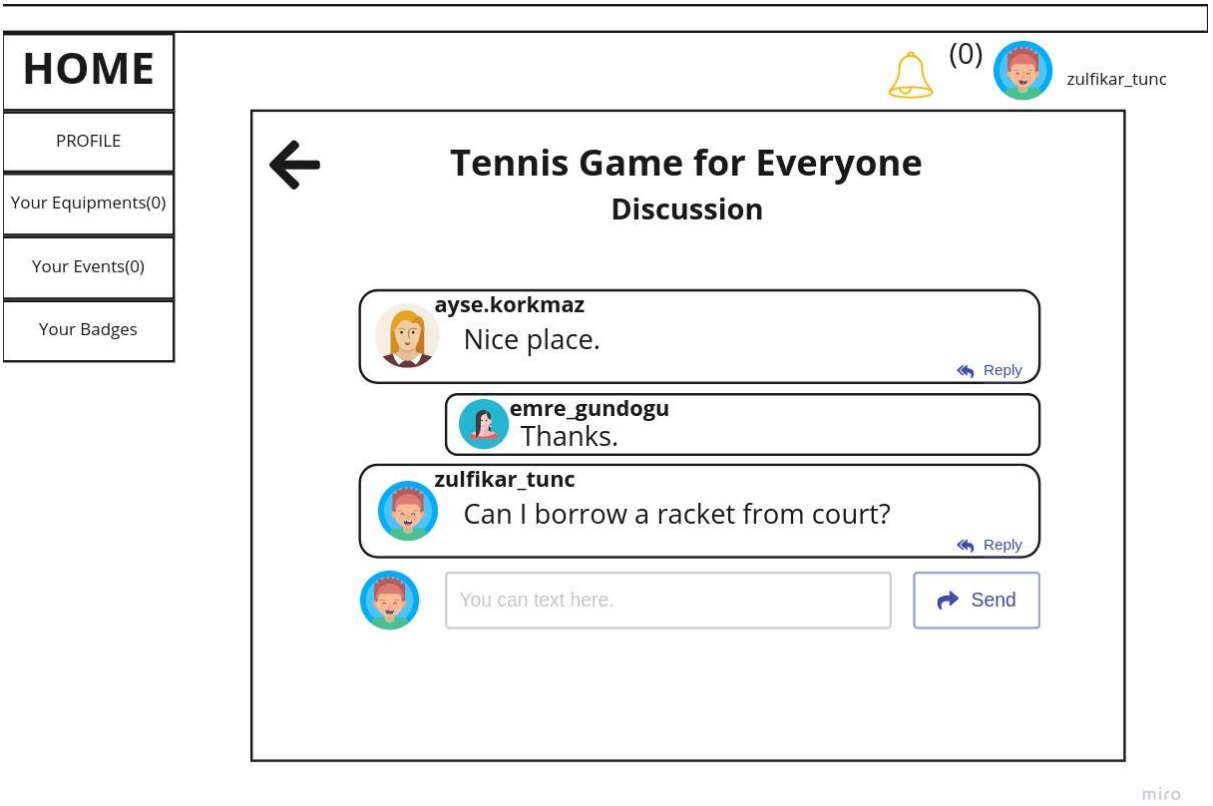


2. Discussion Page

3. Sending Message

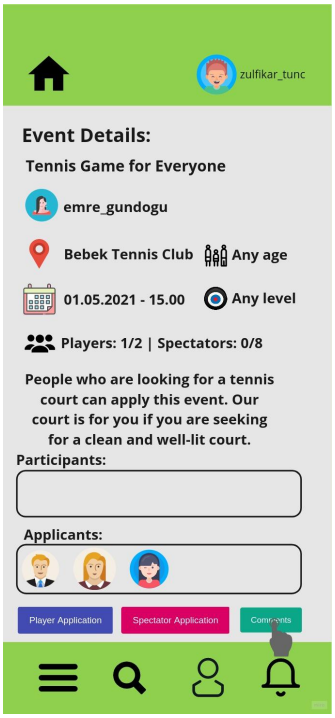


4. Discussion Page (Message is sent.)

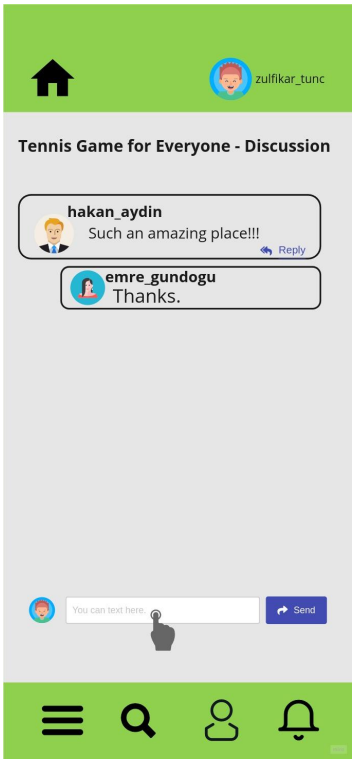


Android Mockup

1. Event Page



2. Discussion Page



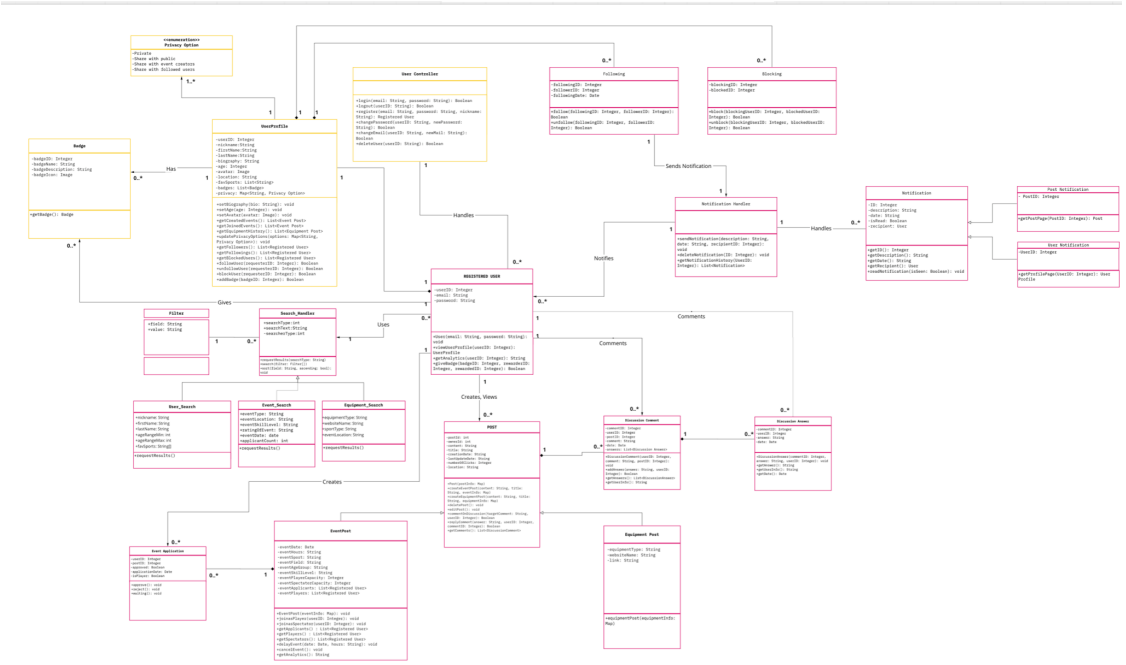
3. Sending Message

4. Discussion Page (Message is sent.)

UML Diagrams

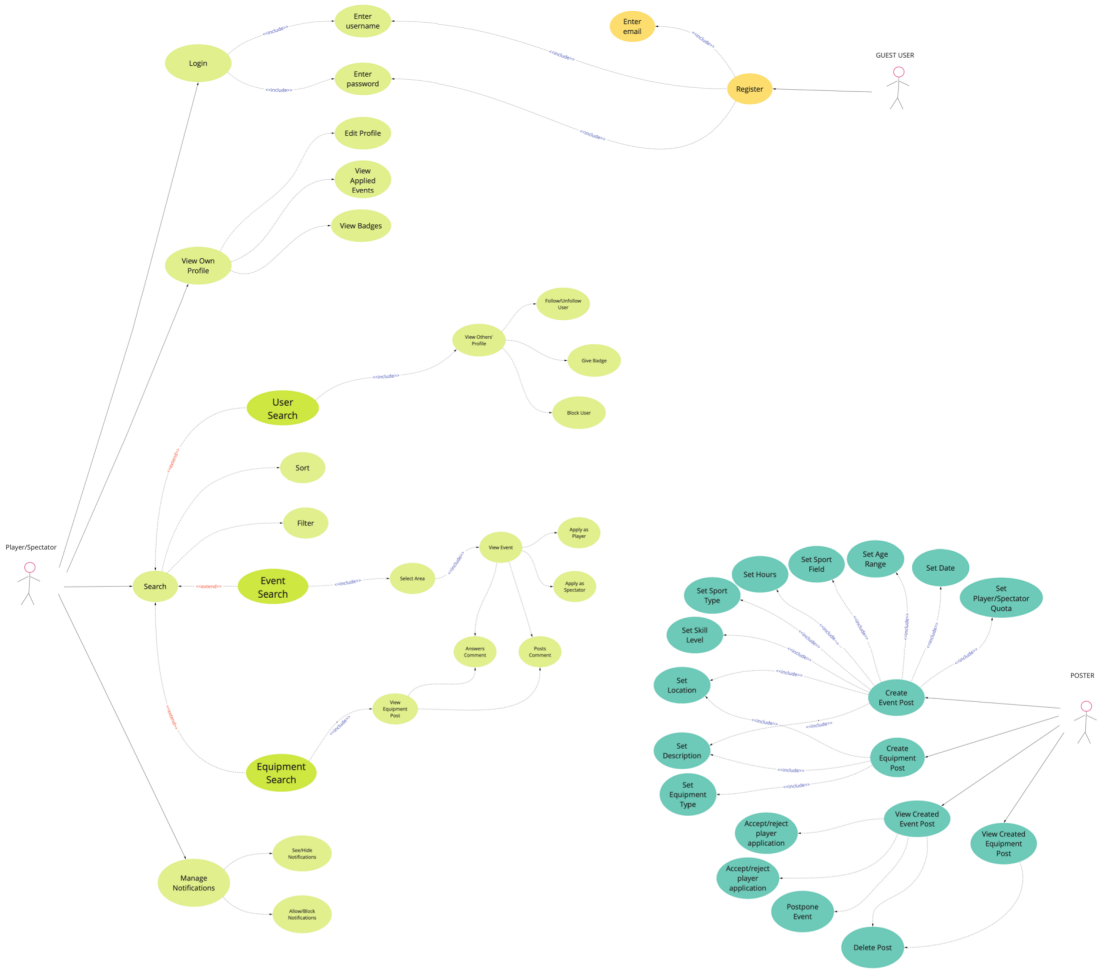
Class Diagram

Click here for high resolution.



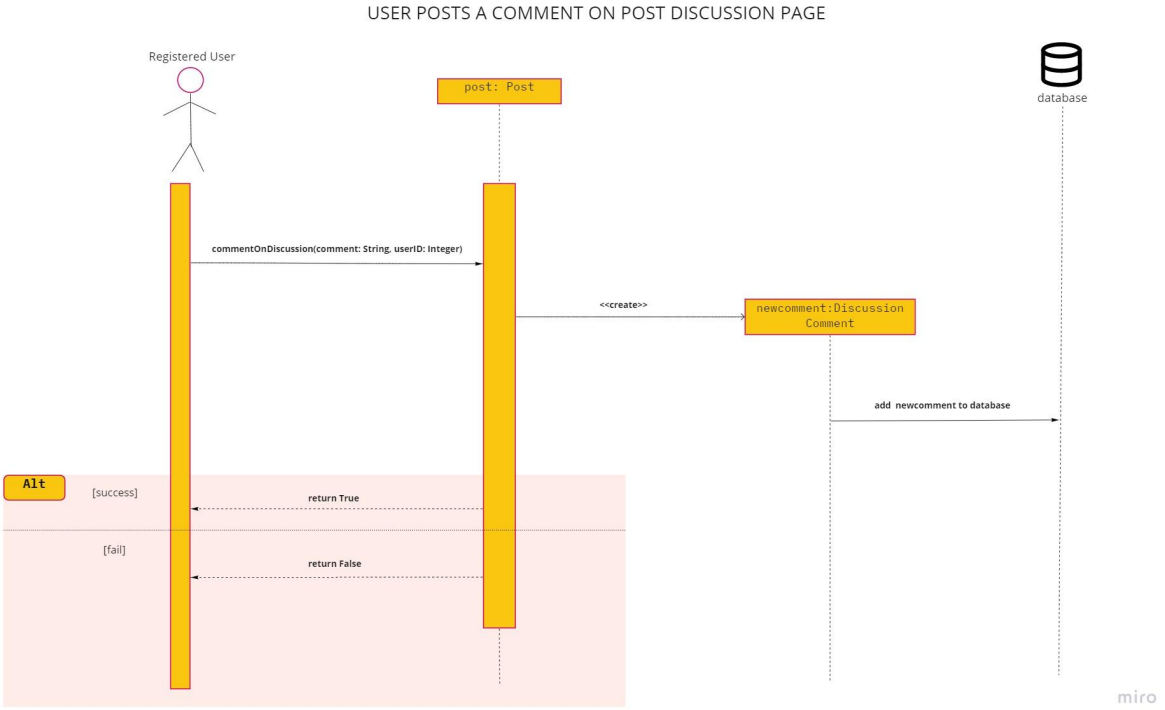
Use Case Diagram

Click here for high resolution.

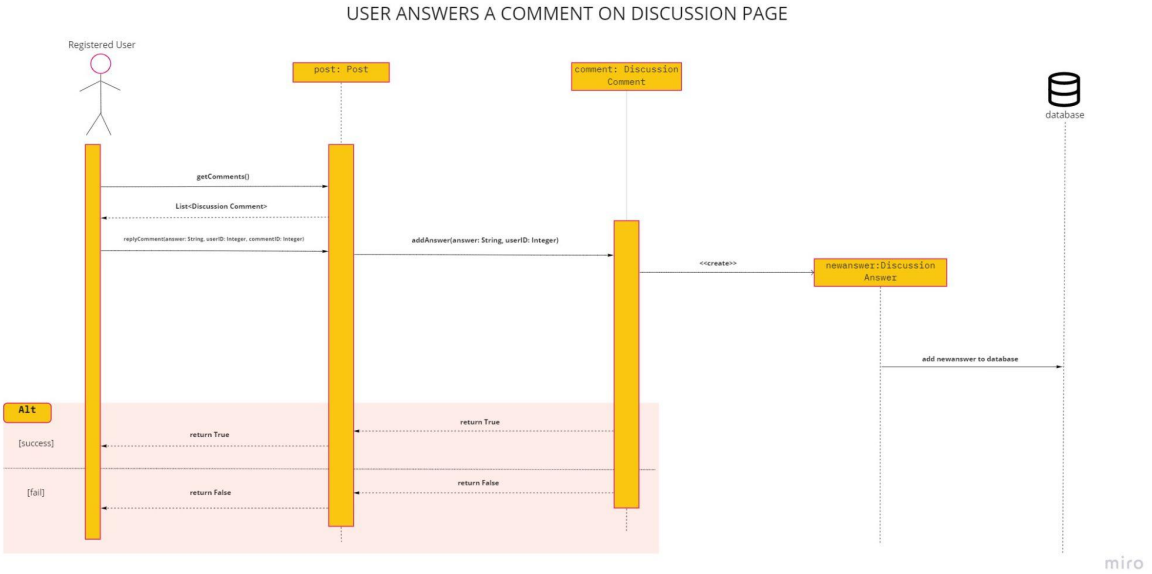


Sequence Diagrams

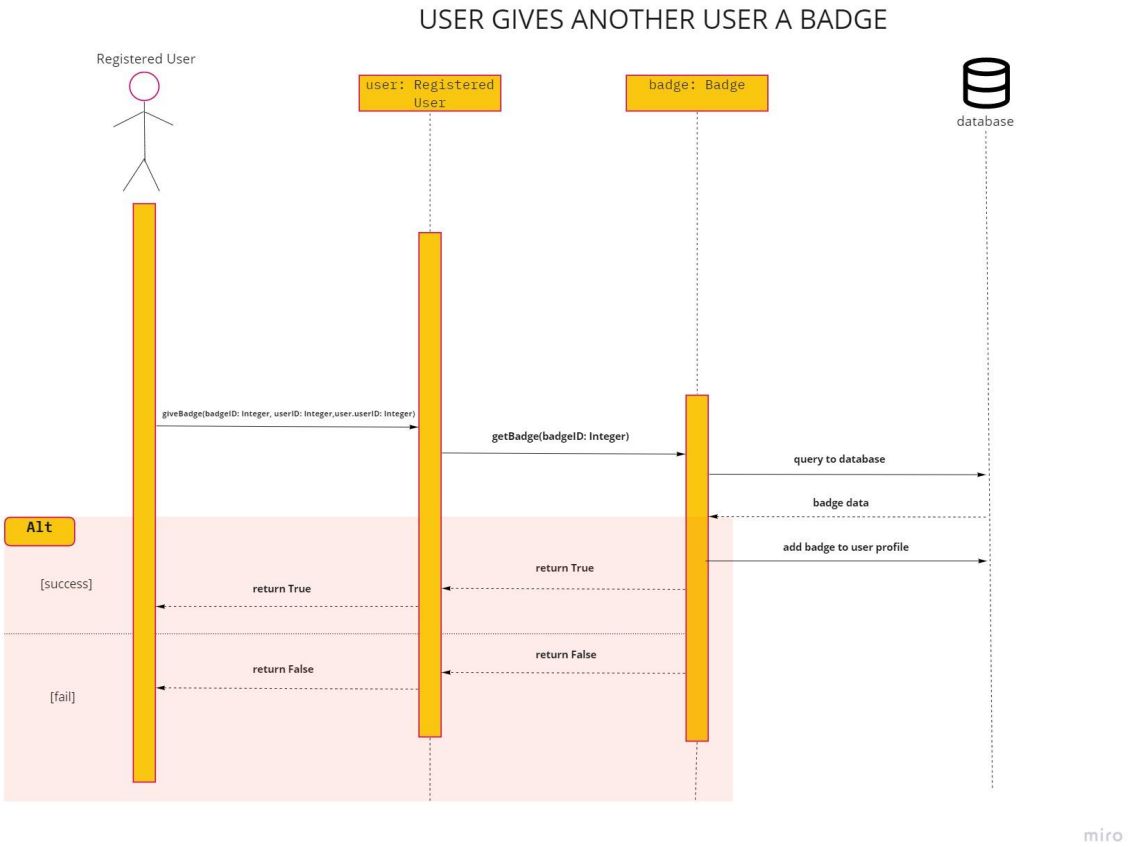
Posting a comment on discussion page



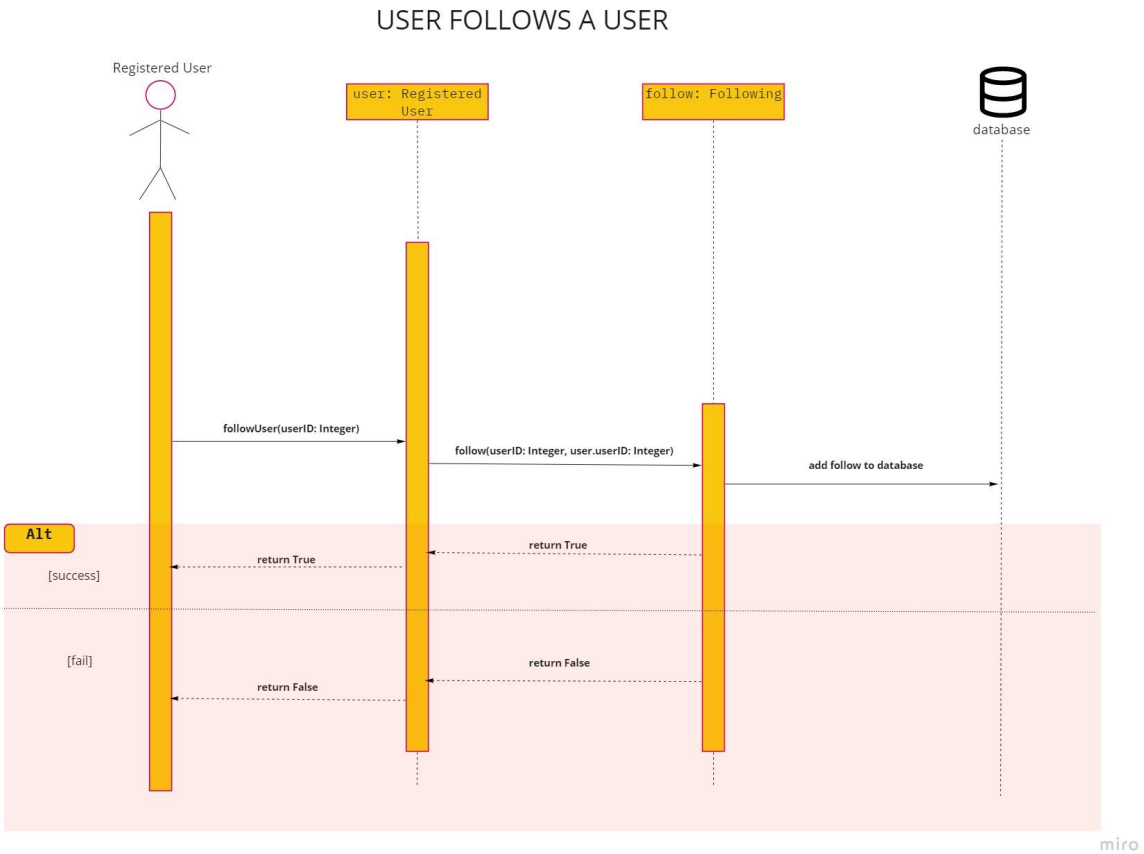
Answering a comment on discussion page



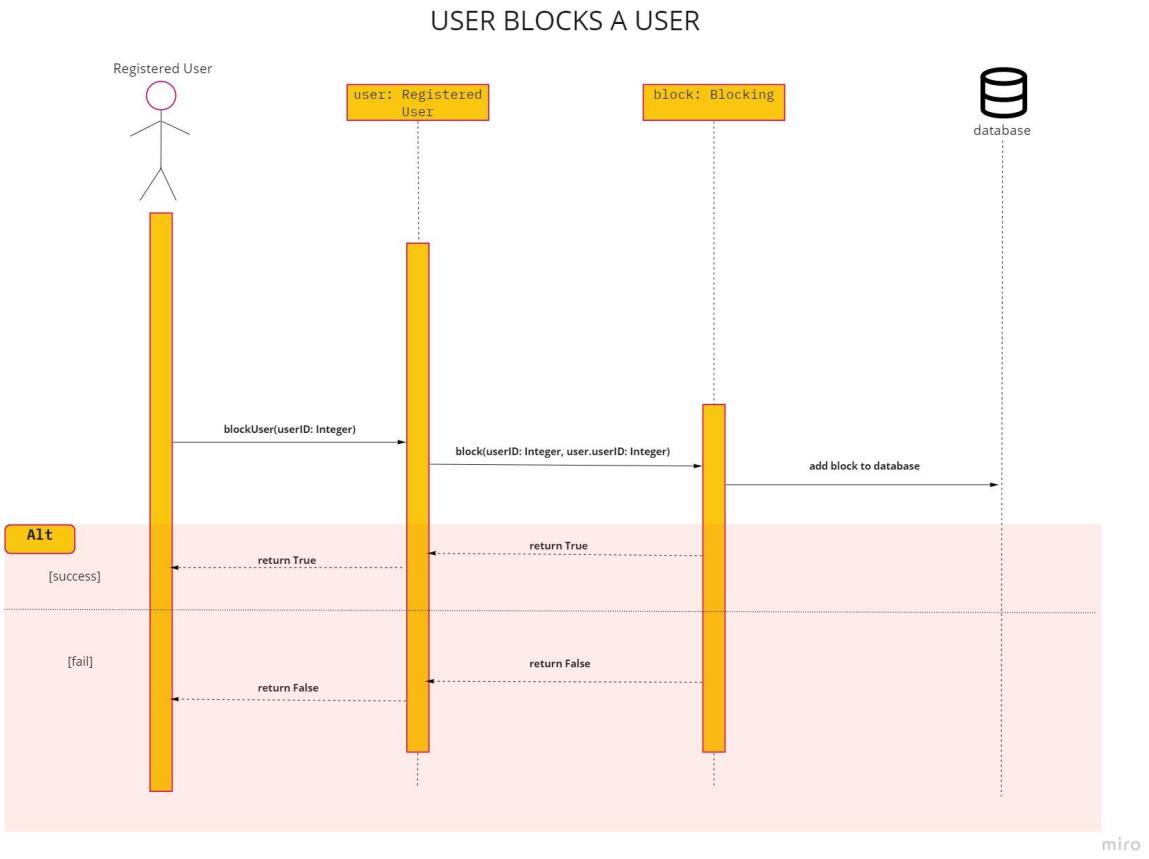
Giving a badge to another user



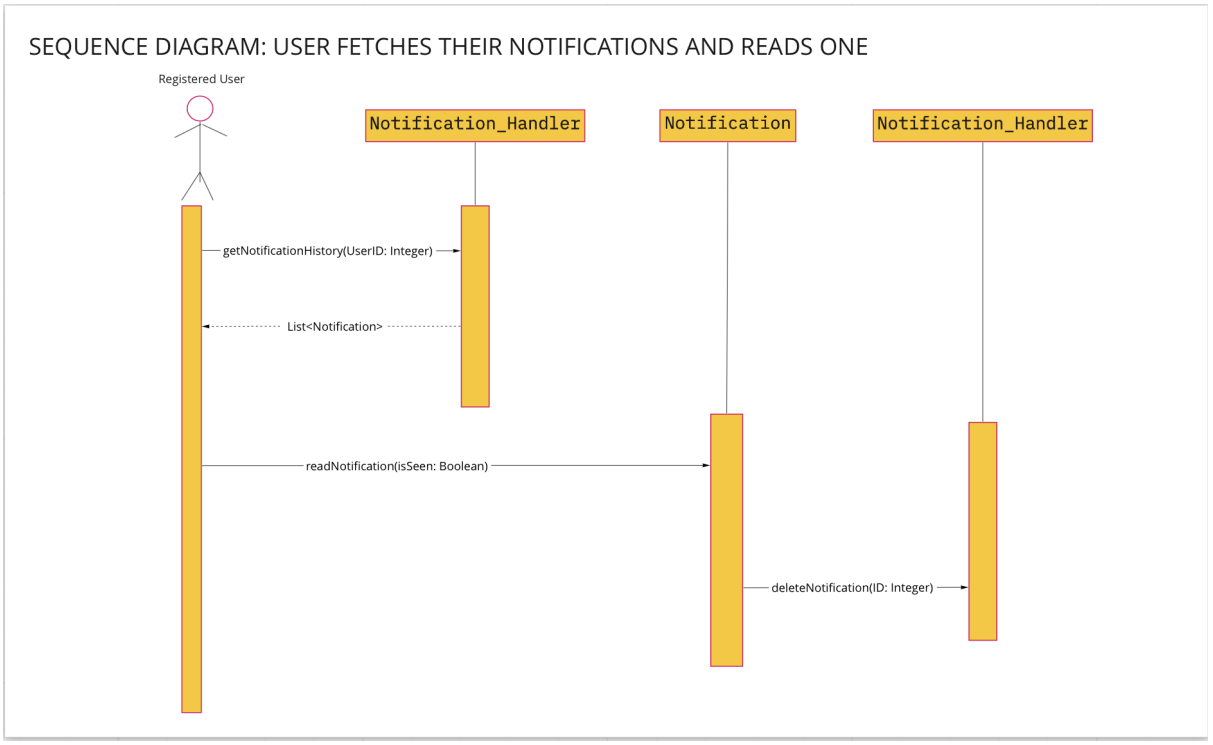
Following a user



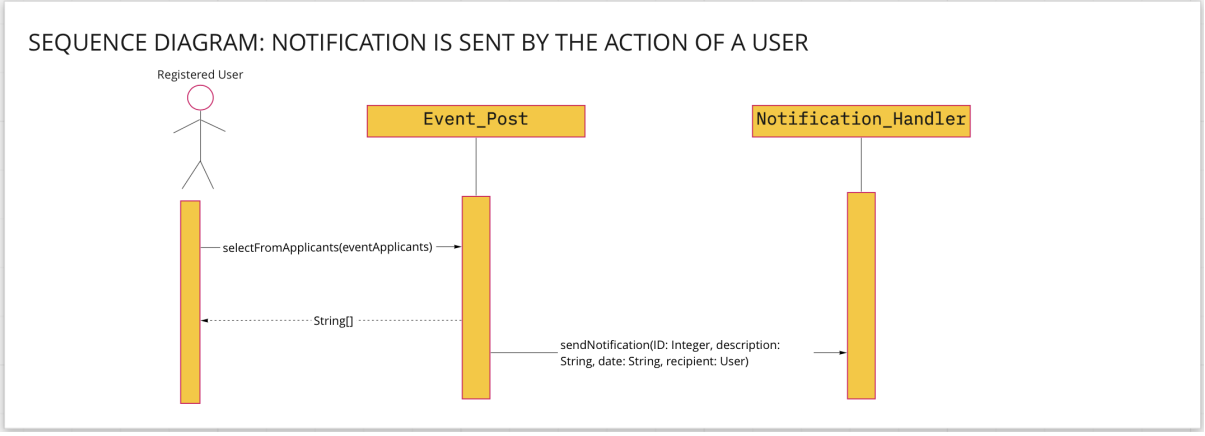
Blocking a user



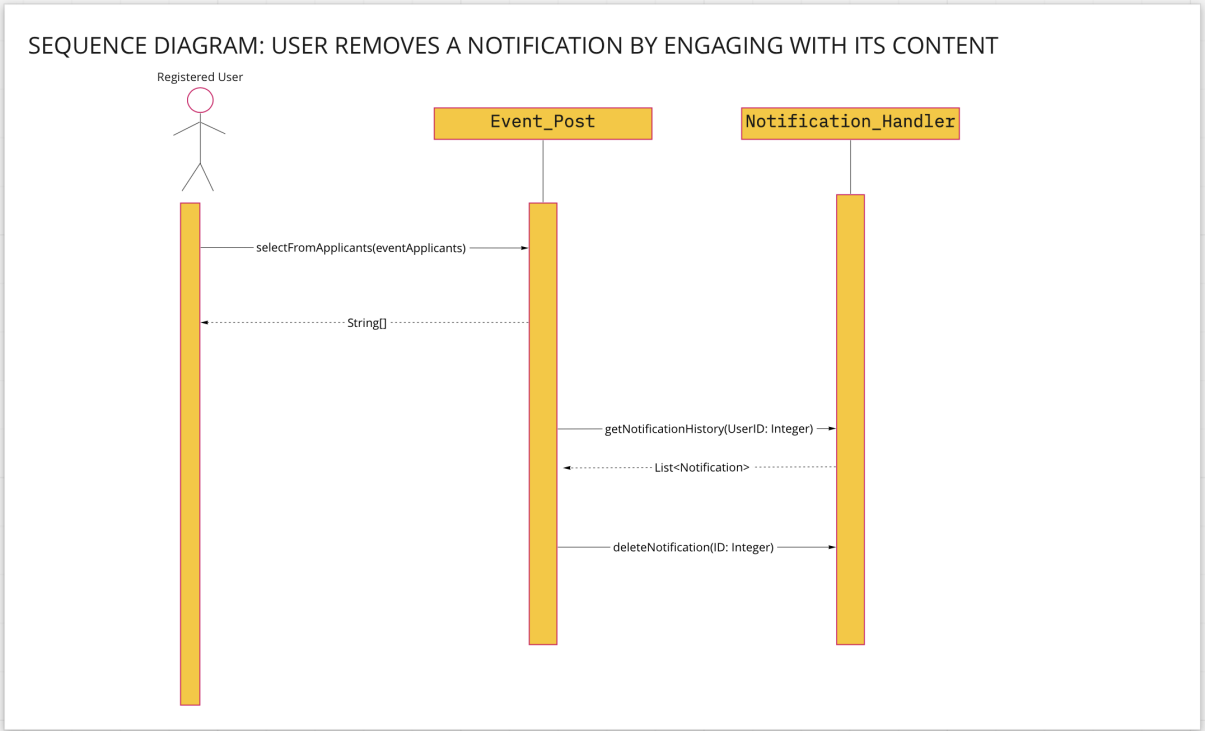
Fetching notifications and reading one



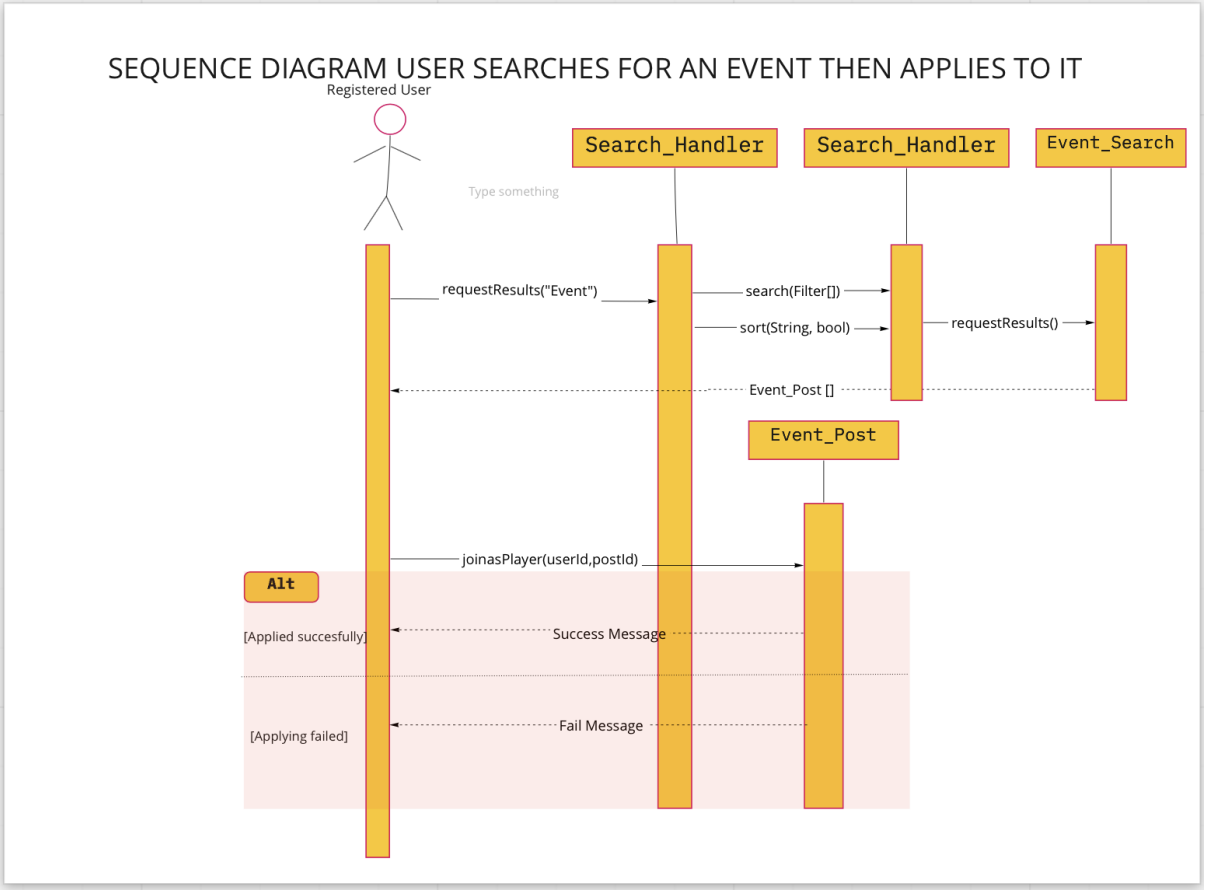
A notification is sent as a result of a user action



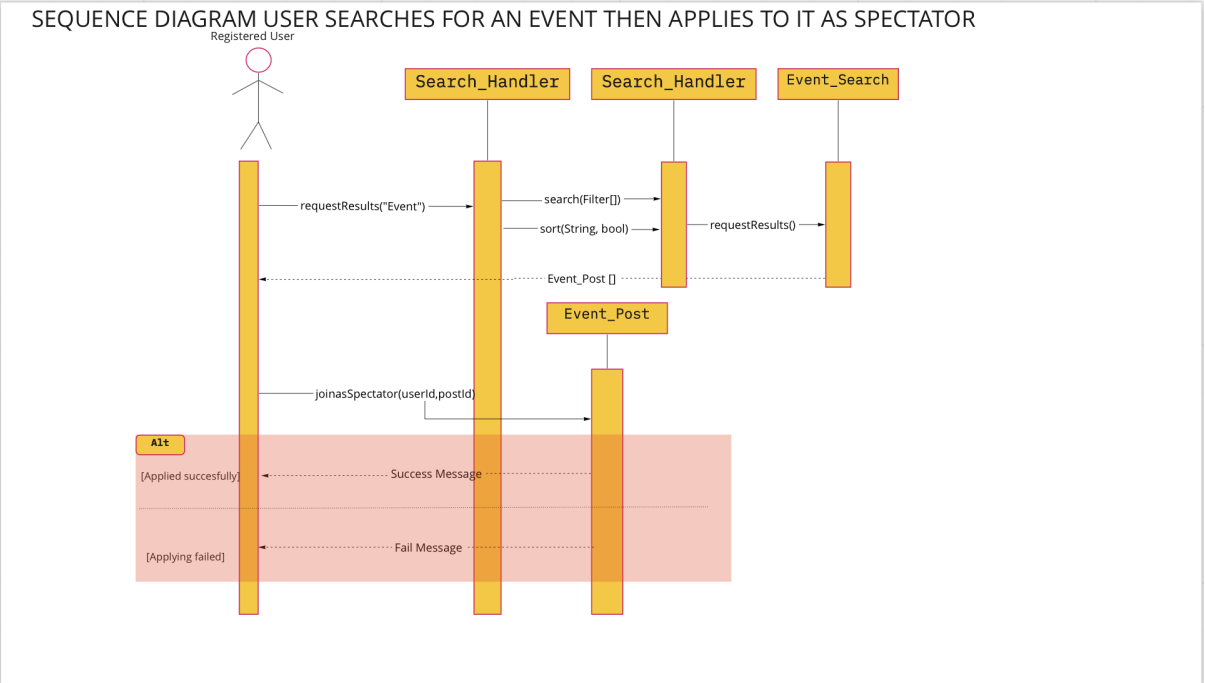
A notification is removed when a user engages with the content of that notification



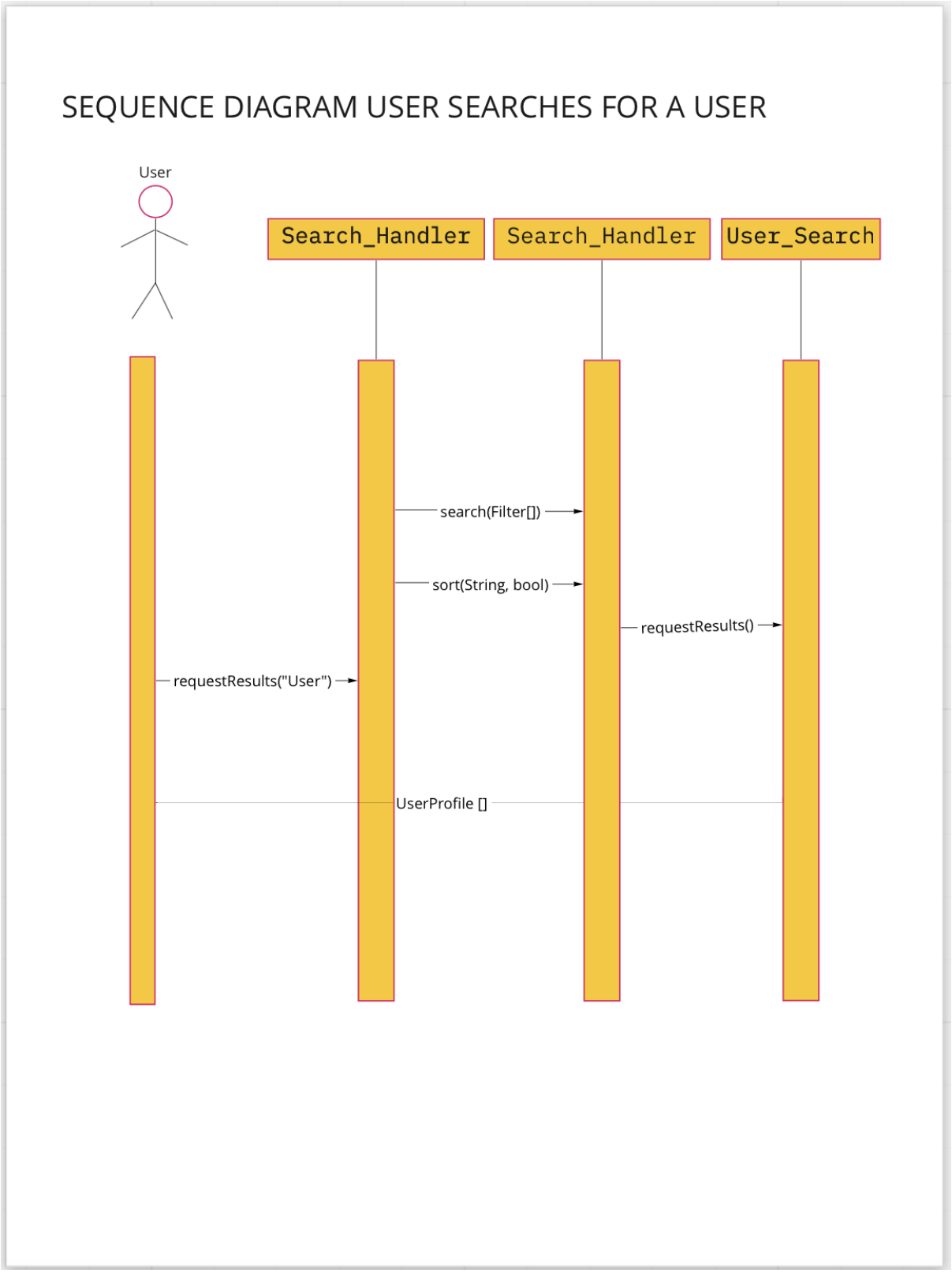
A user finds an event and applies to it



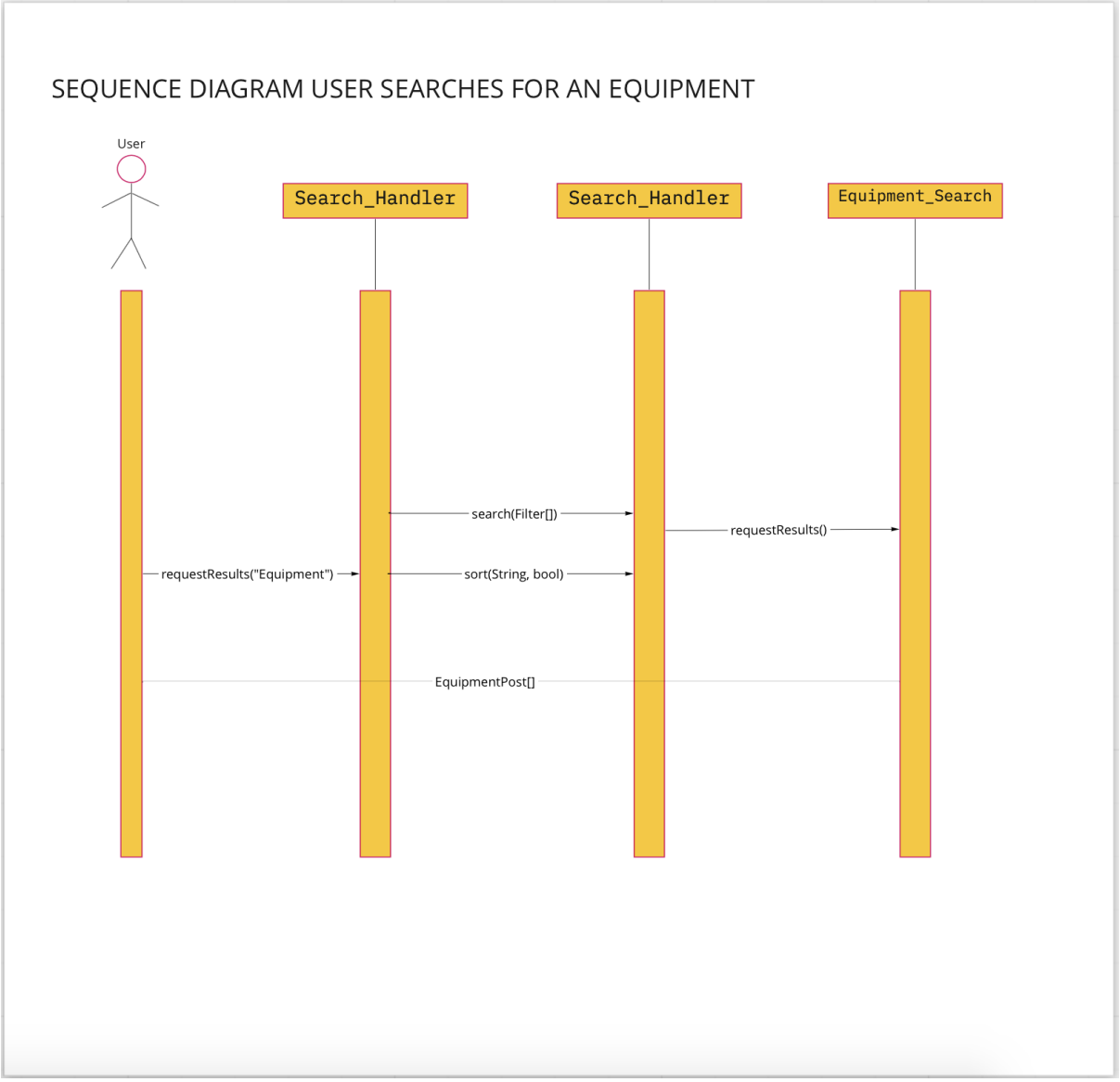
A user finds an event and applies to it as spectator



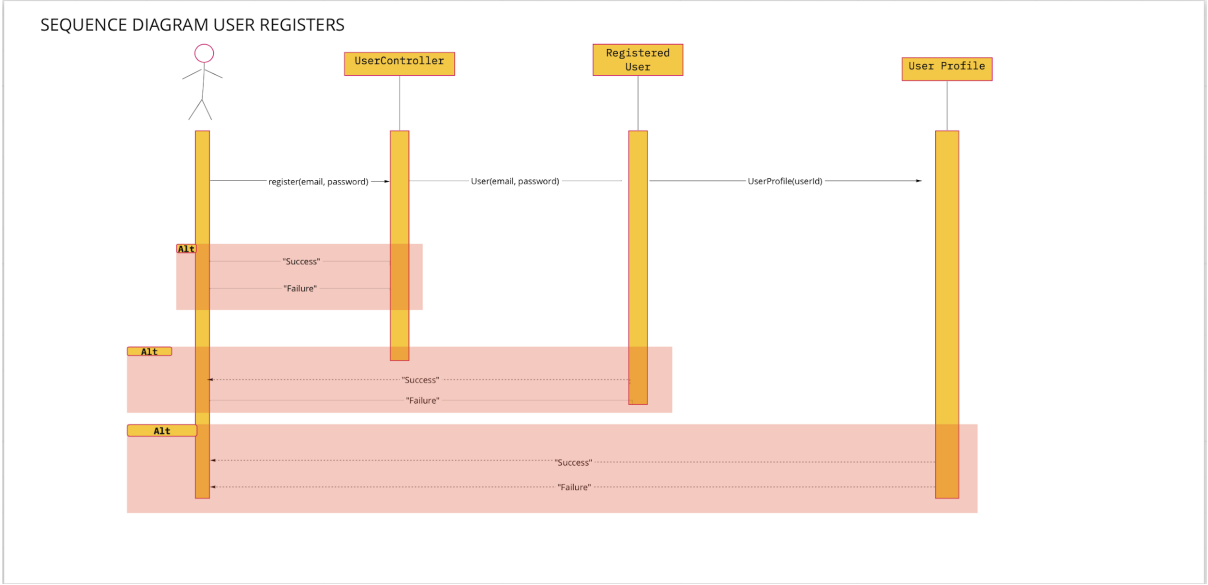
A user searches for a user



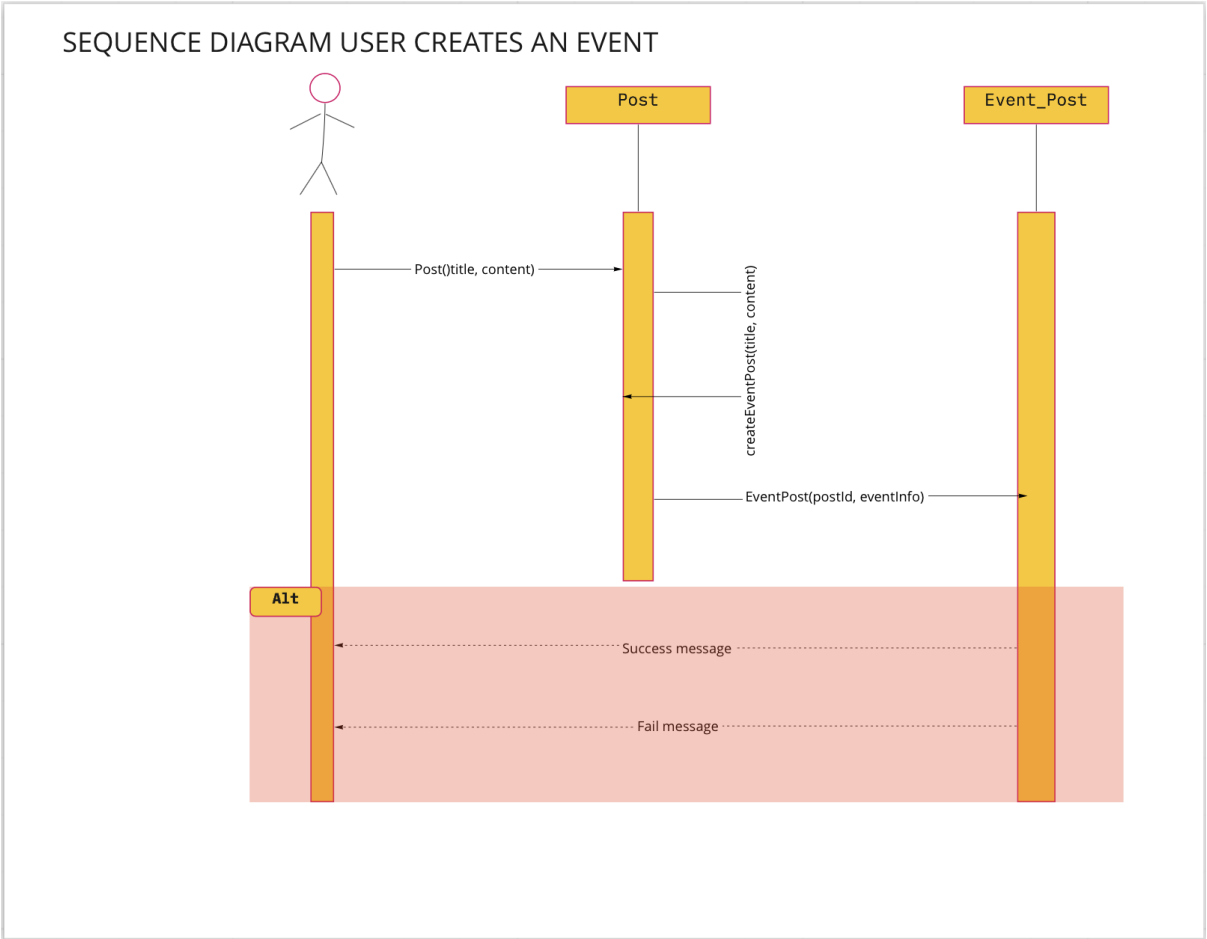
A user searches for an equipment



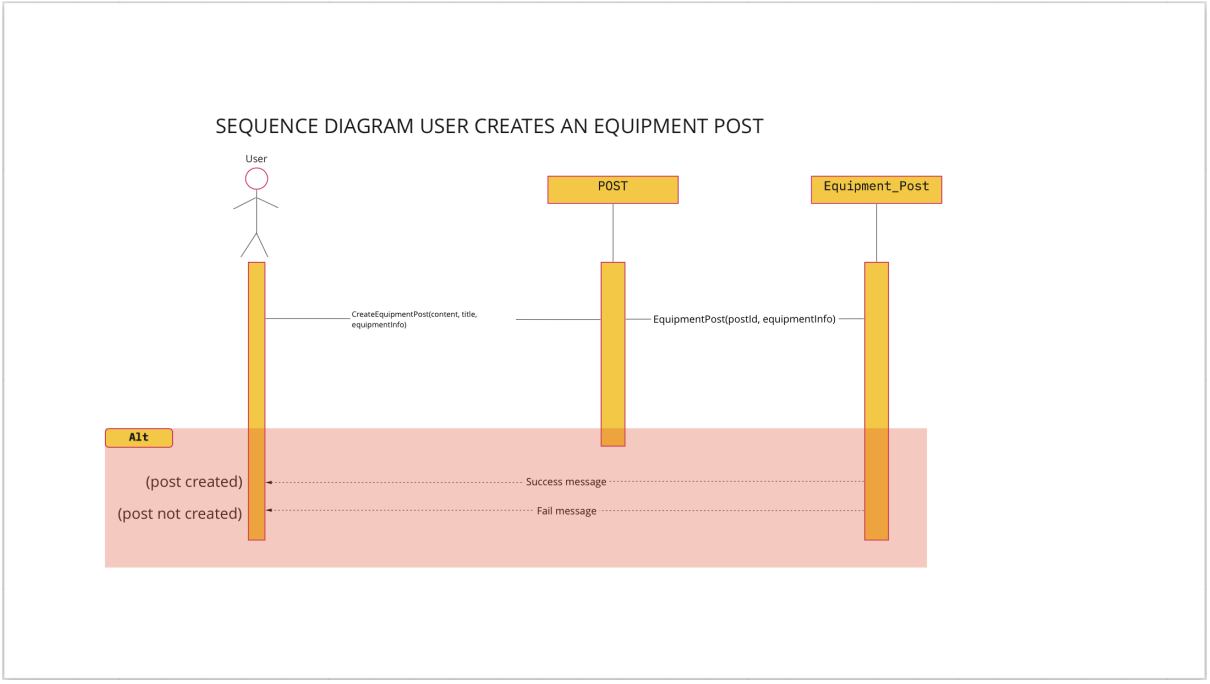
A user registers



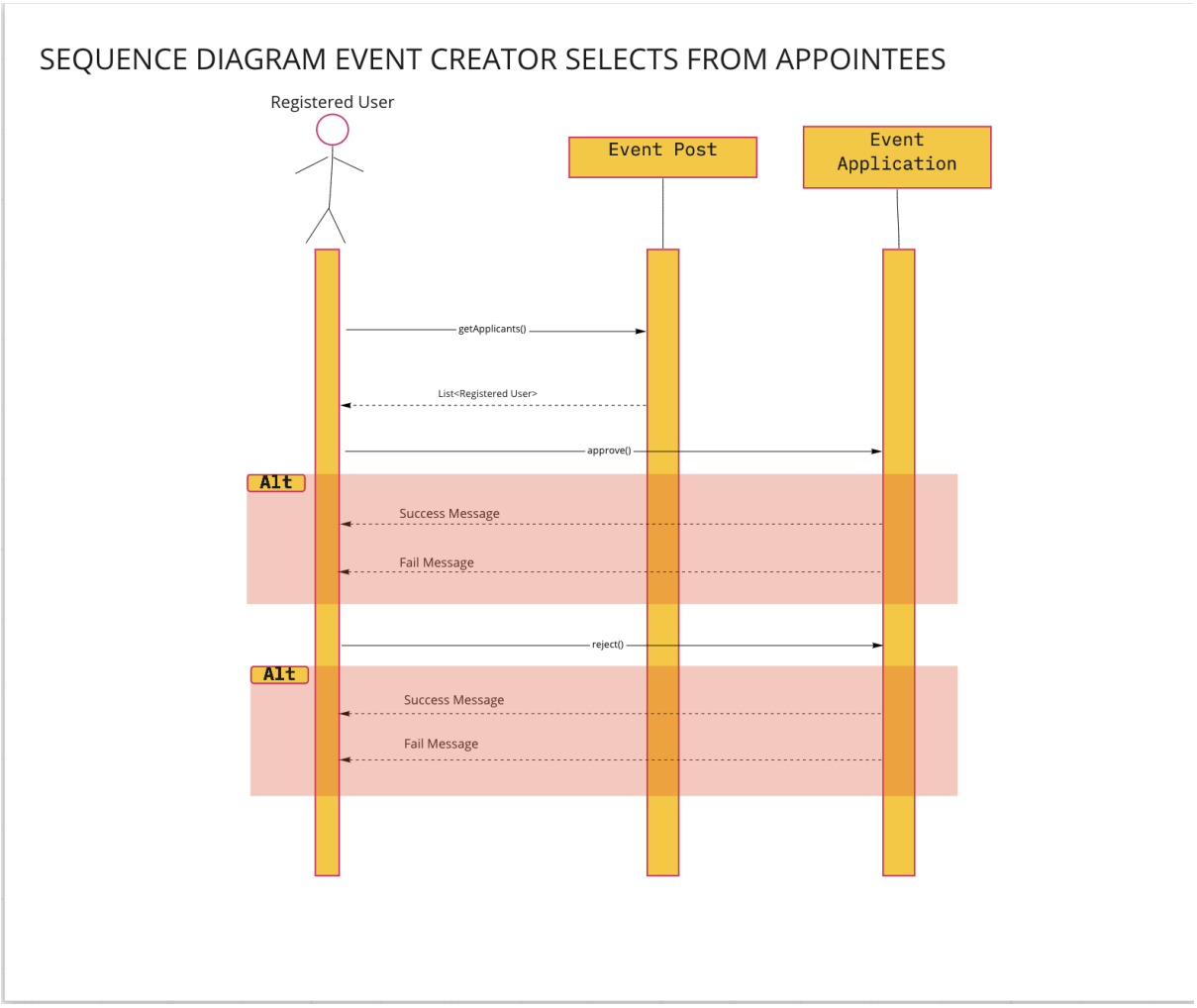
A user creates an event



A user creates an equipment post



An event creator selects from appointees



Project Plan&RAM&Communication Plan

Project Plan

	Name	Duration	Start	Finish	Predecessors	Resource Names
1	**Infrastructure Setup**	10 days?	**3/23/21 8:00 AM**	**4/1/21 5:00 PM**		**ALL**
2	Discord Server Initialization	8 days?	3/23/21 8:00 AM	4/1/21 5:00 PM		Mehmet Saraçoğlu
3	Group First Meeting	8 days?	3/23/21 8:00 AM	4/1/21 5:00 PM		ALL
4	Creating Wiki Pages	8 days?	3/23/21 8:00 AM	4/1/21 5:00 PM		ALL
5	Communication Plan	8 days?	3/23/21 8:00 AM	4/1/21 5:00 PM		Ahmet Necip Görgülü
6	Git Study	8 days?	3/23/21 8:00 AM	4/1/21 5:00 PM		ALL
7	Repository Research	8 days?	3/23/21 8:00 AM	4/1/21 5:00 PM		ALL
8	README	8 days?	3/23/21 8:00 AM	4/1/21 5:00 PM		Berkay Döner
9	Creating Issue Labels	8 days?	3/23/21 8:00 AM	4/1/21 5:00 PM		Bengisu Özaydın
10	Creating First Issues	8 days?	3/23/21 8:00 AM	4/1/21 5:00 PM		ALL
11	**Determining Requirements**	10 days?	**3/31/21 8:00 AM**	**4/9/21 5:00 PM**		**ALL**
12	Glossary	10 days?	3/31/21 8:00 AM	4/9/21 5:00 PM		Doğukan Akar;Kerem Zama...
13	**Functional Requirements**	10 days?	**3/31/21 8:00 AM**	**4/9/21 5:00 PM**		**ALL**
14	User Requirements	10 days?	3/31/21 8:00 AM	4/9/21 5:00 PM		Doğukan Akar;Şefika Akman
15	System Requirements	10 days?	3/31/21 8:00 AM	4/9/21 5:00 PM		Kerem Zaman;Mehmet Sara...
16	**Non-functional Requirements**	10 days?	**3/31/21 8:00 AM**	**4/9/21 5:00 PM**		**ALL**
17	Availability	8 days?	3/31/21 8:00 AM	4/9/21 5:00 PM		Berkay Döner
18	Deployment	8 days?	3/31/21 8:00 AM	4/9/21 5:00 PM		Berkay Döner
19	Performance	8 days?	3/31/21 8:00 AM	4/9/21 5:00 PM		Berkay Döner
20	Privacy	8 days?	3/31/21 8:00 AM	4/9/21 5:00 PM		Berkay Döner
21	Security	10 days?	3/31/21 8:00 AM	4/9/21 5:00 PM		Berkay Döner;Kerem Zaman...
22	Standards	8 days?	3/31/21 8:00 AM	4/9/21 5:00 PM		Berkay Döner
23	**Scenarios and Mock-ups**	7 days?	**4/13/21 8:00 AM**	**4/19/21 5:00 PM**	**13;16**	**ALL**
24	**User Personas and Their Stories**	7 days?	**4/13/21 8:00 AM**	**4/19/21 5:00 PM**		**ALL**
25	Person #1 Eren Yiğit	7 days?	4/13/21 8:00 AM	4/19/21 5:00 PM		Kerem Zaman;Mehmet Sara...
26	Person #2 Ali Topçu	5 days?	4/13/21 8:00 AM	4/19/21 5:00 PM		Ahmet Necip Görgülü
27	Person #3 Melike Taşkıran	5 days?	4/13/21 8:00 AM	4/19/21 5:00 PM		Berkay Döner
28	Person #4 Fatih Kerim	5 days?	4/13/21 8:00 AM	4/19/21 5:00 PM		Ege Can Kaya
29	Person #5 Zülfikar Tunç	7 days?	4/13/21 8:00 AM	4/19/21 5:00 PM		Doğukan Akar;Şefika Akman
30	**Scenarios and Mock-ups**	7 days?	**4/13/21 8:00 AM**	**4/19/21 5:00 PM**		**ALL**
31	Scenario #1: Creating an Event	7 days?	4/13/21 8:00 AM	4/19/21 5:00 PM		Kerem Zaman;Mehmet Sara...
32	Scenario #2: User Searches for an Event	5 days?	4/13/21 8:00 AM	4/19/21 5:00 PM		Ahmet Necip Görgülü
33	Scenario #3: User Applies for an Event to Be	5 days?	4/13/21 8:00 AM	4/19/21 5:00 PM		Berkay Döner
34	Scenario #4: Event Creator Selects from App	5 days?	4/13/21 8:00 AM	4/19/21 5:00 PM		Ege Can Kaya
35	Scenario #5: Discussion	7 days?	4/13/21 8:00 AM	4/19/21 5:00 PM		Doğukan Akar;Şefika Akman
36	**Design Diagrams**	11 days?	**4/20/21 8:00 AM**	**4/30/21 5:00 PM**	**24;30**	**ALL**
37	Class Diagram	9 days?	4/20/21 8:00 AM	4/30/21 5:00 PM		ALL
38	Use-Case Diagram	9 days?	4/20/21 8:00 AM	4/30/21 5:00 PM		ALL
39	Sequence Diagram	9 days?	4/20/21 8:00 AM	4/30/21 5:00 PM		ALL
40	Milestone #1	12 days?	4/28/21 8:00 AM	5/13/21 5:00 PM	36	ALL
41	Research and development	16 days?	5/18/21 8:00 AM	6/8/21 5:00 PM		ALL
42	Code versioning, Code review	6 days?	5/18/21 8:00 AM	5/25/21 5:00 PM		ALL
43	**Designing the Project Structure**	8 days?	**5/25/21 8:00 AM**	**6/1/21 5:00 PM**	**40**	**ALL**
44	Designing Frontend Part	6 days?	5/25/21 8:00 AM	6/1/21 5:00 PM		ALL
45	Designing Android Part	6 days?	5/25/21 8:00 AM	6/1/21 5:00 PM		ALL
46	Designing Backend Part	6 days?	5/25/21 8:00 AM	6/1/21 5:00 PM		ALL
47	**Verification and Validation**	8 days?	**6/2/21 8:00 AM**	**6/9/21 5:00 PM**	**43**	**ALL**
48	**Testing**	8 days?	**6/2/21 8:00 AM**	**6/9/21 5:00 PM**		**ALL**
49	Unit Tests	6 days?	6/2/21 8:00 AM	6/9/21 5:00 PM		ALL
50	User Tests	6 days?	6/2/21 8:00 AM	6/9/21 5:00 PM		ALL
51	Debugging	6 days?	6/2/21 8:00 AM	6/9/21 5:00 PM		ALL
52	Configuration Management	6 days?	6/8/21 8:00 AM	6/15/21 5:00 PM		ALL
53	Deployment	6 days?	6/8/21 8:00 AM	6/15/21 5:00 PM		ALL
54	Dockerization	6 days?	6/8/21 8:00 AM	6/15/21 5:00 PM		ALL
55	Lessons Learned	6 days?	6/8/21 8:00 AM	6/15/21 5:00 PM		ALL
56	**Milestone2**	11 days?	**6/15/21 8:00 AM**	**6/25/21 5:00 PM**	**47**	**ALL**
57	Executive Summary	9 days?	6/15/21 8:00 AM	6/25/21 5:00 PM		ALL
58	List and Status of Deliverables	9 days?	6/15/21 8:00 AM	6/25/21 5:00 PM		ALL
59	Evaluation of the status of deliverables	9 days?	6/15/21 8:00 AM	6/25/21 5:00 PM		ALL
60	Evaluation of tools and processes	9 days?	6/15/21 8:00 AM	6/25/21 5:00 PM		ALL
61	Summary of Work Done By Each Member	9 days?	6/15/21 8:00 AM	6/25/21 5:00 PM		ALL
62	Project Plan	9 days?	6/15/21 8:00 AM	6/25/21 5:00 PM		ALL
63	RAM	9 days?	6/15/21 8:00 AM	6/25/21 5:00 PM		ALL
64	Review of CMPE352 and the project status	14 days?	10/6/21 8:00 AM	10/25/21 5:00 PM		ALL
65	**Implementation Phase 1**	30 days?	**10/26/21 8:00 AM**	**11/24/21 5:00 PM**	**56;64**	
66	**Backend**	23 days?	**10/26/21 8:00 AM**	**11/17/21 5:00 PM**		
67	Initialization	2 days?	10/26/21 8:00 AM	10/27/21 5:00 PM		Bengisu Özaydın;Ege Can K...
68	Database design and API documentation	5 days?	10/26/21 8:00 AM	11/1/21 5:00 PM		ALL
69	Register Implementation	8 days?	11/2/21 8:00 AM	11/9/21 5:00 PM	67;68	Bengisu Özaydın;Ege Can K...
70	Login implementation	8 days?	11/2/21 8:00 AM	11/9/21 5:00 PM	67;68	Bengisu Özaydın;Ege Can K...
71	User Profiles	9 days?	11/9/21 8:00 AM	11/17/21 5:00 PM	67;68	Bengisu Özaydın;Ege Can K...
72	Post Creation	9 days?	11/9/21 8:00 AM	11/17/21 5:00 PM		Bengisu Özaydın;Ege Can K...
73	Post View	9 days?	11/9/21 8:00 AM	11/17/21 5:00 PM		Bengisu Özaydın;Ege Can K...
74	**Frontend**	23 days?	**10/26/21 8:00 AM**	**11/17/21 5:00 PM**		
75	Initialization	8 days	10/26/21 8:00 AM	11/2/21 5:00 PM		Berkay Döner;Doğukan Aka...
76	Homepage	7 days?	10/26/21 8:00 AM	11/1/21 5:00 PM		Berkay Döner;Doğukan Aka...
77	Navigation Bar	7 days?	10/26/21 8:00 AM	11/1/21 5:00 PM		Berkay Döner;Doğukan Aka...
78	Register Page	8 days?	11/2/21 8:00 AM	11/9/21 5:00 PM		Berkay Döner;Doğukan Aka...
79	Login Page	8 days?	11/2/21 8:00 AM	11/9/21 5:00 PM		Berkay Döner;Doğukan Aka...
80	Profile Page	8 days?	11/9/21 5:00 PM	11/17/21 5:00 PM		Berkay Döner;Doğukan Aka...
81	Post Creation	8 days?	11/9/21 5:00 PM	11/17/21 5:00 PM		Berkay Döner;Doğukan Aka...
82	Post View	8 days?	11/9/21 5:00 PM	11/17/21 5:00 PM		Berkay Döner;Doğukan Aka...
83	**Android**	23 days?	**10/26/21 8:00 AM**	**11/17/21 5:00 PM**		
84	Initialization	8 days?	10/26/21 8:00 AM	11/2/21 5:00 PM		Ahmet Necip Görgülü;Mehm...
85	Homepage	8 days	10/26/21 8:00 AM	11/2/21 5:00 PM		Ahmet Necip Görgülü;Mehm...

86	Navigation Bar	8 days	10/26/21 8:00 AM	11/2/21 5:00 PM		Ahmet Necip Görgülü;Mehm...
87	Register Page	7 days	11/2/21 5:00 PM	11/9/21 5:00 PM		Ahmet Necip Görgülü;Mehm...
88	Login Page	7 days	11/2/21 5:00 PM	11/9/21 5:00 PM		Ahmet Necip Görgülü;Mehm...
89	Profile Page	8 days	11/9/21 5:00 PM	11/17/21 5:00 PM		Ahmet Necip Görgülü;Mehm...
90	Post Creation	8 days?	11/9/21 5:00 PM	11/17/21 5:00 PM		Ahmet Necip Görgülü;Mehm...
91	Post View	8 days?	11/9/21 5:00 PM	11/17/21 5:00 PM		Ahmet Necip Görgülü;Mehm...
92	Test and Review	5 days	11/17/21 5:00 PM	11/24/21 5:00 PM		ALL
93	Milestone 1 Presentation	5 days?	11/18/21 8:00 AM	11/24/21 5:00 PM	66;74;83	ALL
94	**Implementation Phase 2**	**26 days?**	**11/25/21 8:00 AM**	**12/20/21 5:00 PM**	**66;74;83**	
95	**Backend**	**14 days?**	**11/25/21 8:00 AM**	**12/8/21 5:00 PM**		
96	Search System	14 days	11/25/21 8:00 AM	12/8/21 5:00 PM		Bengisu Özaydın;Ege Can K...
97	Search Filtering/Sorting	14 days?	11/25/21 8:00 AM	12/8/21 5:00 PM		Bengisu Özaydın;Ege Can K...
98	Event Application	14 days	11/25/21 8:00 AM	12/8/21 5:00 PM		Bengisu Özaydın;Ege Can K...
99	Comments/Answers	14 days	11/25/21 8:00 AM	12/8/21 5:00 PM		Bengisu Özaydın;Ege Can K...
100	**Frontend**	**14 days?**	**11/25/21 8:00 AM**	**12/8/21 5:00 PM**		
101	Search System	14 days?	11/25/21 8:00 AM	12/8/21 5:00 PM		Berkay Döner;Doğukan Aka...
102	Search Filtering/Sorting	14 days?	11/25/21 8:00 AM	12/8/21 5:00 PM		Berkay Döner;Doğukan Aka...
103	Event Application	14 days	11/25/21 8:00 AM	12/8/21 5:00 PM		Berkay Döner;Doğukan Aka...
104	Comments/Answers	14 days	11/25/21 8:00 AM	12/8/21 5:00 PM		Berkay Döner;Doğukan Aka...
105	**Android**	**14 days?**	**11/25/21 8:00 AM**	**12/8/21 5:00 PM**		
106	Search System	14 days?	11/25/21 8:00 AM	12/8/21 5:00 PM		Ahmet Necip Görgülü;Mehm...
107	Search Filtering/Sorting	14 days?	11/25/21 8:00 AM	12/8/21 5:00 PM		Ahmet Necip Görgülü;Mehm...
108	Event Application	14 days	11/25/21 8:00 AM	12/8/21 5:00 PM		Ahmet Necip Görgülü;Mehm...
109	Comments/Answers	14 days	11/25/21 8:00 AM	12/8/21 5:00 PM		Ahmet Necip Görgülü;Mehm...
110	Test and Review	8 days?	12/8/21 5:00 PM	12/20/21 5:00 PM		ALL
111	Milestone 2 Presentation	8 days?	12/9/21 8:00 AM	12/20/21 5:00 PM	95;100;105	ALL
112	**Implementation Phase 3**	**16 days?**	**12/20/21 5:00 PM**	**1/5/22 5:00 PM**	**95;100;105**	
113	**Backend**	**16 days**	**12/20/21 5:00 PM**	**1/5/22 5:00 PM**		
114	Follow/Block Functionalities	16 days	12/20/21 5:00 PM	1/5/22 5:00 PM		Bengisu Özaydın;Ege Can K...
115	Badge	16 days	12/20/21 5:00 PM	1/5/22 5:00 PM		Bengisu Özaydın;Ege Can K...
116	Recommendation System	16 days	12/20/21 5:00 PM	1/5/22 5:00 PM		Bengisu Özaydın;Ege Can K...
117	Notification System	16 days	12/20/21 5:00 PM	1/5/22 5:00 PM		Bengisu Özaydın;Ege Can K...
118	**Frontend**	**16 days?**	**12/21/21 8:00 AM**	**1/5/22 5:00 PM**		
119	Follow/Block Functionalities	16 days	12/21/21 8:00 AM	1/5/22 5:00 PM		Berkay Döner;Doğukan Aka...
120	Badge	16 days	12/21/21 8:00 AM	1/5/22 5:00 PM		Berkay Döner;Doğukan Aka...
121	Recommendation System	16 days	12/21/21 8:00 AM	1/5/22 5:00 PM		Berkay Döner;Doğukan Aka...
122	Notification System	16 days?	12/21/21 8:00 AM	1/5/22 5:00 PM		Berkay Döner;Doğukan Aka...
123	**Android**	**16 days?**	**12/21/21 8:00 AM**	**1/5/22 5:00 PM**		
124	Follow/Block Functionalities	16 days	12/21/21 8:00 AM	1/5/22 5:00 PM		Ahmet Necip Görgülü;Mehm...
125	Badge	16 days	12/21/21 8:00 AM	1/5/22 5:00 PM		Ahmet Necip Görgülü;Mehm...
126	Recommendation System	16 days	12/21/21 8:00 AM	1/5/22 5:00 PM		Ahmet Necip Görgülü;Mehm...
127	Notification System	16 days?	12/21/21 8:00 AM	1/5/22 5:00 PM		Ahmet Necip Görgülü;Mehm...
128	Milestone 3 and Final Presentation	5 days?	1/6/22 8:00 AM	1/12/22 5:00 PM	113;118;123	ALL

Communication Plan

Audience	Purpose	Where	When
All Team Members	General Discussion	Whatsapp	Anytime
All Team Members	Initial meeting to review the weekly Assignment	Zoom	Tuesdays at 15.00-17.00
All Team Members	Teams are going to arrange separate meetings	Discord-Face to Face	Will be decided by teams separately

Responsibility Assignment Matrix

	Doğukan Akar	Şefika Akman	Ömer Arslan	İbrahim Kağan Bayat	Berkay Döner	Ahmet Necip Görgülü	Ege Can Kaya	Bengisu Özaydın	Mehmet Saraçoğlu	Kerem Zaman
Project Repository Page										
Wiki Home Page&Sidebar	L	C	C	C	A	C	C	C	C	C
Personal Wiki Pages	C	C	C	C	C	C	C	C	C	C
Communication Plan	N	N	N	N	A	L	N	N	N	N
Labels	C	C	R	R	C	C	R	L	C	A
Repository Home Page	C	C	N	N	L	A	C	C	C	N
Git Research	N	N	N	N	L	N	A	N	N	N
Repository Research	C	C	C	C	C	C	C	C	C	C
Meeting Notes	N	N	N	N	N	L	C	N	N	N
TA Meeting Notes	N	N	N	N	L	N	N	N	N	N
Project Requirements										
User Requirements	C	L	R	R	A	R	R	R	R	R
System Requirements	N	N	R	C	R	A	R	A	C	C
Non-functional Requirements	N	N	N	R	L	N	A	N	N	N
Glossary	L	C	N	N	A	N	R	N	N	N
Similar Project Research	N	N	A	N	N	N	L	C	N	N
Question Preparation	N	N	N	N	L	C	C	N	N	N
Customer Meeting Attendance	C	C	C	C	C	L	C	C	C	C
Customer Meeting Notes	N	N	N	N	L	N	C	N	N	N
Scenarios&Mockups										
Event Creation Scenario	R	R	N	L	N	R	N	N	C	C
Event Creation Mockup	R	R	N	L	N	R	N	N	C	R
Event Search Scenario	R	R	R	N	N	L	N	R	N	N
Event Search Mockup	R	R	N	N	R	L	N	R	N	N
Event Application Scenario	R	R	N	N	L	R	R	N	N	N
Event Application Mockup	R	R	N	N	L	R	R	N	N	N
Player Selection Scenario	R	R	N	N	N	N	L	N	N	N
Player Selection Mockup	R	R	N	N	N	N	L	N	N	N
Discussion Scenario	C	L	N	N	N	N	N	N	N	N
Discussion Mockup	C	L	N	N	R	N	N	N	N	N

UML Diagrams										
Use Case Diagram	C	C	C	C	C	C	C	C	C	C
Class Diagram	C	C	C	C	C	C	C	C	C	C
Sequence Diagram: Posting Comment	N	N	N	N	L	C	N	R	N	N
Sequence Diagram: Answering Comment	N	N	N	N	L	R	N	R	N	N
Sequence Diagram: Giving Badge	N	N	N	N	L	N	N	C	N	N
Sequence Diagram: Following User	N	N	N	N	L	N	N	C	N	N
Sequence Diagram: Blocking User	N	N	N	N	L	N	N	C	N	N
Sequence Diagram: Fetching Notifications	N	N	N	N	N	N	L	N	N	N
Sequence Diagram: Sending Notification	N	N	N	N	N	N	L	N	N	N
Sequence Diagram: Removing Notification	N	N	N	N	N	N	L	N	N	N
Sequence Diagram: Player Application	N	N	N	N	N	L	N	C	N	N
Sequence Diagram: Spectator Application	N	N	N	L	N	N	N	C	N	N
Sequence Diagram: User Search	N	N	N	N	N	R	N	C	N	L
Sequence Diagram: Equipment Search	N	N	N	N	N	R	N	C	N	L
Sequence Diagram: User Registration	R	L	N	N	N	N	N	N	N	N
Sequence Diagram: Event Creation	L	R	N	N	N	N	N	N	N	N
Sequence Diagram: Equipment Post Creation	N	N	N	N	N	N	N	C	L	N
Sequence Diagram: Selecting Applicants	N	N	L	N	N	N	N	C	N	N
Project Plan										
Revision of Requirements	L	C	N	N	C	N	R	N	N	N
Revision of Scenarios&Mockups	C	C	N	N	C	C	R	N	N	N
Revision of Diagrams	C	C	N	C	C	C	C	L	N	C
Project Plan: Work We've Done So Far	N	N	L	N	R	N	R	N	L	N
Project Plan: Future Tasks for CMPE352	N	N	N	L	R	N	R	N	N	R
Project Plan: Future Tasks for CMPE451	N	N	N	N	C	N	L	N	N	C
RAM	C	C	C	C	L	C	C	C	C	C
Milestone Report	C	C	C	C	C	L	C	C	C	C
CMPE352 - Milestone #1										
API Implementation										
API Research	C	C	C	C	C	C	C	C	C	C
Implementation	C	C	C	C	C	C	C	A	C	C
Unit Testing	C	C	C	C	C	A	C	L	C	C
Acceptance Testing	C	C	C	C	L	A	C	C	C	N
Milestone Report	C	C	C	C	C	L	C	C	C	A
CMPE352 - Milestone #2										
Backend										
Initialization	N	N	N	C	N	N	C	C	N	C
Database Design	C	C	C	C	C	C	C	L	C	C
Registration Implementation	N	N	N	C	R	N	A	C	N	L
Login Implementation	R	N	N	C	N	N	A	L	N	C
User Profiles	N	R	N	C	N	N	C	A	N	C
Profile Privacy Options	N	N	N	C	R	N	C	A	N	C
Post Creation	R	N	N	C	N	R	C	A	N	L
Post View	N	R	N	C	N	N	L	A	N	C
Frontend										

Initialization	L	C	N	N	C	N	N	C	N	N
Homepage	C	C	N	N	L	N	N	R	N	N
Navigation Bar	C	L	N	N	C	R	N	R	N	N
Registration Page	L	C	N	N	C	N	N	C	N	R
Login Page	C	C	N	N	L	N	N	C	N	R
Profile Page	L	C	N	N	C	N	N	N	N	N
Post Creation	C	C	N	N	L	N	N	N	N	N
Post View	C	L	N	N	C	N	N	N	N	N
Android										
Initialization	N	N	L	C	N	C	N	N	C	N
Homepage	N	R	N	L	N	A	N	N	C	N
Navigation Bar	N	R	N	L	N	L	N	N	C	N
Registration Page	R	N	N	A	N	A	N	R	L	R
Login Page	R	N	N	A	N	A	N	R	L	R
Profile Page	N	N	L	A	N	A	N	R	C	N
Post Creation	N	N	N	N	R	N	N	N	C	N
Post View	N	N	N	N	R	N	N	N	C	N
CMPE451 - Milestone #1										
Backend										
Search System	R	N	N	C	N	N	C	C		C
Search Filtering/Sorting	R	N	N	C	N	R	C	C		C
Event Application	N	R	N	C	N	N	C	L	N	C
Comments/Answers	N	N	N	C	R	N	C	L	N	C
Frontend										
Search System	L	C	R		C	R	N	C		N
Search Filtering/Sorting	C	L	N		C	N	N	C		N
Event Application	C	C	N	N	L	N	N	R	N	R
Comments/Answers	C	C	R	N	L	N	N	R	N	N
Android										
Search System	R	N	L		N	A	N	R		N
Search Filtering/Sorting	R	N	A		N	L	N	R		N
Event Application	N	R	L	A	N	A	N	N	C	N
Comments/Answers	N	N	L	A	R	C	N	N	C	N

CMPE451 - Milestone #2										
Backend										
Follow/Block Functionalities	R	N	N	C	N	N	C	L	N	C
Badge	N	N	N	C	R	N	C	L	N	C
Recommendation System	N	R	N	C	N	N	C	C	N	C
Notification System	N	N	N	C	R	N	L	C	N	C
Frontend										
Follow/Block Functionalities	L	C	N	N	C	N	N	R	N	N
Badge	A	L	N	N	C	N	N	R	N	N
Recommendation System	C	C	N	N	L	N	N	C	N	N
Notification System	C	L	N	N	A	N	N	C	N	N
Android										
Follow/Block Functionalities	R	N	L	N	N	A	N	N	C	N
Badge	R	N	L	A	N	L	N	N	C	N
Recommendation System	N	R	C	A	N	N	N	R	L	R
Notification System	N	N	A	L	R	N	N	R	C	N
CMPE451 - Milestone #3										