CMPE 451 FINAL MILESTONE GROUP REPORT

Course: Cmpe 451 PROJECT DEVELOPMENT IN SOFTWARE ENG
Semester: 2021 fall
Group members who have prepared this report: Gözde Ünver, Zuhal Didem Aytaç, Muhammed Furkan Atasoy, Berkay Bör, Mehmet Fatih Aksoy, Umut Gün, Salih Bedirhan Eker, Kemal Çağın Sertkaya
Name of Group: LUDO
Group Number: 5
Sub-group members:

Backend: Gözde Ünver, Mehmet Fatih Aksoy, Salih Bedirhan Eker
Frontend: Zuhal Didem Aytaç, Muhammed Furkan Atasoy, Umut Gün
Android: Berkay Bör, Kemal Çağın Sertkaya

Tag-id: v0.9 (https://github.com/bounswe/2021SpringGroup5/releases/tag/v0.9)
Date: 12.01.2022

User manual: **https://github.com/bounswe/2021SpringGroup5/wiki/User-Manual**

System manual: **https://github.com/bounswe/2021SpringGroup5/wiki/System-manual**

README.md:
https://github.com/bounswe/2021SpringGroup5/blob/master/project/README.md

Database dump:
https://github.com/bounswe/2021SpringGroup5/blob/master/melih-master.zip

Status of deliverables:

Deliverable Name:	Status:
A: Project Repository	
* Wiki pages	Completed
* Manuals:	Completed
- User Manual	Completed
- System Manual	Completed
* Software requirements specifications (SRS)	Completed
* Software design documents	Completed
* User scenarios and mockups	Completed
* Project plan	Completed
* Issues	Completed
* Milestone reports (first, second and final)	Completed
* Software related content	Completed
- API documentation	Completed
- Code documented and with unit tests	Completed
- Final release	Completed
B: Software Package	
* Final release in GitHub repository	Completed
* Documented API	Completed
* Configuration files	Completed
* Database dump	Completed
* 'Readme' file	Completed
C: Final Group Report	
* Title page	Completed
* Executive summary	Completed
- Status of deliverables	Completed
- Status of all requirements	Completed
- Final release notes.	Completed
- Final project evaluations	Completed
* Summary of work done by each team member	Completed
* W3C Activity Streams 2.0 standard and Wikidata	Completed

* API	Completed
* User manual	Completed
* System manual	Completed
* Project Plan	Completed
* Software Requirements	Completed
* Software design documents	Completed
* Group Specific Core Functionality	Completed

Status of all requirements :

Completed:

Frontend:

- Login page
- Forgot Password page
- Register page
- Profile page
    - Follow/unfollow functionality
    - Badge listing
    - Sport-skill level listing
    - Previous Event Posts - Previous Equipment Posts
- Comment Section on Event Detail Page - Equipment Detail Page
    - Comment Listing
    - Comment Posting
- Participants page
    - Different UIs for event owner - other users
    - Accept/reject functionality
    - List badge functionality
    - Send badge functionality
- Search page
    - Search event functionality
    - Search equipment functionality
    - Filter event functionality
    - Filter equipment functionality
    - Sort event functionality
    - Sort equipment functionality
- Header and Sidebar
- Homepage
    - Previous events of the followed users
    - Logo

        - Project description
        - Popular categories with images
    - Event Post
        - Creation page
        - Map functionality
        - Detail page
        - Apply Event
    - Equipment Post
        - Creation page
        - Map functionality
        - Detail page

Backend:

- Create event and equipment post
- Change event and equipment post information
- Delete event and equipment post
- Get details of event and equipment post
- Postpone an event post
- Apply and spectate to event
- Accept-reject applicants
- Return profile informations of the user
- Home page
- Comment on event/equipment posts
- Event analytics
- Signup, login and mail verification
- Follow, unfollow user
- Send badge to user
- Get all badges
- Get event and equipment posts of user

Android:

- Login screen
- Sign up flow
    - Mail verification
- Profile page
    - Badge listing
    - User created events listing
    - Log out functionality
- Main screen
- Search and sort events
- Filter events by location

- Show event details
- Apply to event
- Create event
    - Upload or use camera for event photo
    - Select event location on map

Not completed:

Frontend:

- Profile edit functionality
- Delete comment functionality
- Change event and equipment post information functionality
- Delete event and equipment post functionality
- Create event and equipment post with backend integration

Backend:

- Edit profile information
- Notifications

Android:

- Accept event participants
- Post and delete comments
- Equipment posts
- Send badges
- Follow users

Describe the changes your team has planned/implemented to improve the development process based on previous milestone reflections and how they impacted your process:

Firstly, we decided to skip the minor and less important details for the final milestone and include features with higher functionalities in terms of user experience and the importance of CMPE451 lecture. At the final milestone, we tried to focus on the important points. We put more emphasis on priority issues by ordering the importance of the things that will be done for the final milestone. Hence; backend, frontend and android teams completed their targeted tasks by ranking their importance among themselves.
In the first and second milestones, we focused on functionalities, while in the final milestone we focused on providing a complete and proper flow for the user. In addition to that, we have set earlier dates of deadlines for the completion of the tasks. In this way, we were able to change and improve the work done by checking more and more times before the delivery date.
As a result, the knowledge and experience we gained from the first two milestones helped us a lot for the final milestone and we successfully delivered the final milestone.

Final project evaluations: Reflections related to your Final Milestone Demo including lessons learned.:

We have learned several lessons during the semester. The most important thing is working together. Whole semester worked on a project as a team very effectively. In addition to working together, we have learned that managing time is necessary. Also, git is critical because whole team members worked on the same project at the same time without any conflict and follow issues. Also we learned code reviewing and testing are also essential. We noticed many bugs while testing and reviewing applications.

A summary of coding work completed by each team member (in tabular format):

Gözde Ünver	- Reviewing Mehmet Fatih's pull requests - Reviewing Salih's pull requests - Implementing home page function in the backend - Implementing return badges owned by user funtionality - Created badges from wiki data in the save badge functionality - Attending all labs and team meetings - Writing the usage of wiki data and activity stream in the group report - Implementing event post analytics function - Writing unit tests for home page, badges of a user, event analytics functions - Changing the user authentication process in the login required functions that I have implemented - Writing final release notes in the group report - Writing the status of requirements for the parts that I was responsible - Writing the api documentation of home page, badges owned by user, event analytics functions - Writing backend instructions to the system manual of our application - Creating 13 posts for the application - Updating create, change, get details of an event post functions by removing -badges of an event post-convention and updating tests according to this as well - Adding checks in the event and equipment post creations so that when the user does not type information for the required fields, error message is returned - Writing weekly efforts - Creating the final release tag of our project and mailing it - Deploying backend to Amazon - Creating user, event and equipment post for the demo and accepting an event application

Muhammed Furkan Atasoy	- Test deployed server after milestone 2 - Take Frontend Meeting Note #5 - Decide API calls for equipment search - Correct Missing Fields of Event Search Page - Implement Equipment Search Page - Implement Send Badge Functionality - Decide API Calls of Badge functionalities - Implement Map functionality and UI for Create Event and Equipment Page - Implement Home Page - Writing Test for Send Badge functionalities - Writing Test for Equipment Search Page - Writing Test for Map functionality of Create Event Page and Create Equipment Page - Writing Test for Home Page - Helping Authentication Token Issue - Implement Logout functionality - Review pull request: feature/final-fixes - Review pull request: mixed page, feature, bugs - Review pull request: bugfix/tokens - Review pull request: homepage implementation - Write individual milestone 3 report - Contribute Group Milestone: Individual Efforts - Contribute Group Milestone: System manual - Contribute Group Milestone: Create Database Instances - Contribute Group Milestone: Create Readme file for the frontend - Contribute Group Milestone: Status of requirements I completed, URI to the code in the project repository, Screenshots of all implemented Web interfaces - Contribute Group Milestone: Describe the scenario with references for search part (frontend)

Zuhal Didem Aytaç	- Implementing comment functionality in frontend with tests
	  - comment listing in event detail and equipment detail pages
	  - comment post functionality
	- Implementing participant listing in frontend with tests
	  - participant pages linked to event pages
	  - different UI's for event owner and others
	  - accept and reject application functionality
	- Implementing follow/unfollow functionality on frontend with tests
	- Changing profile page UI to show badges, skill levels, previous event and equipment posts
	- Fixing frontend bugs from Milestone 2
	  - changing error messages UI
	  - fixing register page test
	  - fixing register function object
	- Writing tests for profile page
	- Solving Token issue between frontend and backend
	- Connected new frontend functions to backend
	- Making final fixes on frontend
	  - adding navigation to posts from profile
	  - fixing bugs caused of branching
	- Reviewing PRs by Furkan and Umut
	- Editing Wiki Page of the Project, linking necessary pages
	- Writing project description
	- Preparing & performing Web app demo
	- Creating 13 realistic posts for the database dump
	- Reporting status of requirements contributed by me
	- Documenting features in the scenario contributed by me
	- Providing web app screenshots contributed by me
	- Contributing to User and System manuals frontend part

Mehmet Fatih Aksoy	-Reviewing Gözde's Pull requests -Reviewing Salih's pull requests -Reviewing Berkay's pull request -Implementing send badge, get all badge functionalities -Implementing follow, unfollow functionalitites -Attending labs and team meetings -Implementing get profile of user functionality -Implementing authorization system -Changing login required fields to make work my functions -Meeting with frontend to inform input/output format -Implementing get events and equipment posts of user functionalitites -Selecting 3 important API calls for final report -Creating 13 posts for application -Writing the api documentation for follow,unfollow, get profile of user, send badge, get all badge, get events of user, get equipments of user functions -Removing bugs from login, register functions -Deploying backend to Amazon to tests -Creating event and user for the demo -Writing weekly efforts -Learning about database dump and create database dump

Berkay Bör	- Contribute to group report:
	  - Status of all requirements, android
	  - Screenshot of all implemented Mobile interfaces
	  - Android system manual
	  - Individual efforts
	- Writing and implementing android screen functionalities:
	  - Login screen
	  - Sign up screens
	  - Home screen event listing
	  - Filter screen
	  - Create event screen
	  - Event details screen
	  - User profile screen
	- Handling navigation and callbacks
	- Handling parent child widget interactions
	- Helping to resolve authentication issues
	- Implemented authentication and header changes
	- Reviewed Kemal's pull requests
	- Attended all lab meetings
	- Implemented UI when necessary and fixed some visual issues
	- Writing weekly efforts
	- Writing final individual milestone report
	- Presenting final demo of application

Umut Gün	- Decide API calls for event post page - Connect event post detail page with backend - Decide API calls for equipment post page - Implement create equipment post page - Implement equipment post detail page - Connect equipment post detail page with backend - Implement map functionality for the equipment post detail page - Implement home page - Implement apply event feature - Writing test for appy event - Writing test for equipment detail page - Review pull request: feature/services-fixes - Review pull request: bugfix/register-page-bugfixes - Review pull request: feature/participant-listing - Review pull request: feature/comment-section - Writing individual milestone 3 report - Writing weekly efforts - Contribute Group Milestone: Individual Efforts - Contribute Group Milestone: System manual - Contribute Group Milestone: Create Database Data - Contribute Group Milestone: Status of requirements I completed, URI to the code in the project repository, Screenshots of all implemented Web interfaces - Contribute Group Milestone: Final project evaluations - Contribute Group Milestone: Project Plan (simplified version in a report-friendly format) - Contribute Group Milestone: Software Requirements Specification – Software design documents (UML diagrams, mockups) - Contribute Group Milestone: Preparing and submit the group report

Salih Bedirhan Eker	- Updates after milestone 2 - Postpone event functionality - Reject application functionality - Comment functionality - Write tests for postpone event functionality - Write tests for reject application functionality - Write tests for comment functionality - Test deployed server after milestone 2 - Review Gozde’s pull requests - Review Fatih’s pull requests - Write weekly efforts - Contribute final milestone - Updates/bug fixes on logged in user - API calls for search equipment functionality - Correct missing fields of event search page - Create database entries for demo 3 - Create database entries for milestone report - Document demo 3 individuel milestone report

Kemal Çağın Sertkaya	- Implemented user profile page UI - Implemented badges and participated events cards UI in profile page - Added necessary fields to the create event page - Added necessary fields to the filter page - Learned backend connection with http - Reviewed Berkay's Pull Requests - Attended all Labs, team and sub-team meetings - Attended meetings with backend team for input/output formats - Fixed bugs: making better UI, handling owerflowing text fields - Wrote test for login functionality - Wrote test for rounded buttons - Wrote weekly efforts - Created 13 events - Wrote Final Individual Milestone Report - Contributed Final Milestone Report: Individual Effort - Learned unit/widget tests for Flutter - Implemented send badge UI - Added app-release.apk to the repository - Contributed Final Milestone Report: Status of Requirements - Contributed Final Milestone Report: Added unit tests and file link - Contributed Final Milestone Report: User Manual - Contributed Final Milestone Report: Adding Planned/Implemented development improvement changes section. - Contribute Final Milestone Report: Adding Explanations of Some Functions and link corresponding GitHub links (Search, Filter, Profile Page) - Contributed Final Milestone Report: Add readme for Android

Answer the following questions based on the scenario that is provided for your project that is described on the next page. (Describe in detail the work that has been completed related to the features necessary to realize this scenario using your application. – Provide the associated referenced to code in GitHub. LOGIN SEARCH FOLLOW:

Frontend Search and Filter:

The functionality allows the user to search and filter the sports events. Firstly, we designed a UI that includes search bar, sort button and menu, filter button, and filter sidebar containing a separate menu for filtering sport type, capacity, start and end date, location. Then, we implemented authenticated request to the backend in order to fetch the data. The request process is repeated, when every time a user fills a field and click on the search or the filter button. The request body is filled with the data collected from the input components such as text field, date field, or a button that controls true or false values. In addition, there is a map component in the

search page which allows filtering the results according to the location determined by a marker and a circle chosen by a user.

Android Search and Filter:

Search and Filter functionalities return more accurate results about what the user is looking for. First of all, we created the UI: At the top of the home page, we put a search tab, next to that, sorting buttons which are "sort by date" and "sort by name" are located. At the end of the bottom tab, we put a filter button. Hence, it is very easy to find and use these functionalities for the users. The search function is working on the front-end. We thought that we could handle this function on the front-end. So, the back-end returns all results and we refresh the results if the user enters any input to the search bar. On the filter page, user inputs which are text fields, image fields, booleans, etc. are sent to the back-end. There is also a "map function" in the filter page. So, users may select the location and radius to show the nearest events. Later on, the back-end returns to the filtered events and we display it on the homepage.

Backend Search and Filter:

Search functionality includes the filter options. Firstly functionality take a search query and then adds filter functions. This functionality is implemented with the help of 'django queryset' functionality. Queryset functionality adds necessary sql operations to the query in order to filter and then returns an ordered list(order by is also another parameter added by queryset). Here is the related code

Frontend Profile&Follow:

The follow functionality is implemented through the profile page. When a user goes to another user's page, the user's id is extracted from the URL. The profile page is connected to the User Service and sends a request to fetch the users data, the data is shown in profile layout. Under the avatar and general profile information, there is a button. This button is written follow if the user is not yet followed, unfollow otherwise. On click, necessary requests are sent to the backend through User Service. These buttons invalidate the query that fetches user data from the backend, so that the profile information and hence the button label is changed on actions. The profile page test covers the label's dependence on the user info.

Users general information, previously created event posts, previously created equipment posts, badges and sport-skill levels are shown on profile. Users can discover each other from the Event Participants page and navigate to the profiles to learn their information.

Android Profile:

All information about the user can be accessible on the profile page. From top to bottom, the user can see "account details" which correspond to the name, surname, email; under that section, "earned badges" if the user gained a badge from the previous events and below that, "participated events" which shows the events that user participated before. All containers are

connected to the backend and whenever the page is renewed or reopened, the backend returns the data.

Frontend Participant Listing:
A user can discover the participants of events through the Event Participants page. This page is connected to Event Service and fetches the information from the backend using the event id in the URL. The event information includes accepted, waiting, spectator and rejected participants lists and also the boolean value whether the user sending the request is event owner or not. If so, all 4 lists are shown and accept-reject buttons are available for waiting participants. If not, only accepted participants are shown. Participants are shown in user cards, with avatar, name, surname and username. The clicks on the cards navigate to the corresponding user's profile.

Backend Login:

A login functionality is implemented through the login page. When user enters his/her username and password, he/she can enter the system and get the access token from JWT Authenticator.

Backend Follow:

Follow functionality is implemented through the follow page. There is a table that holds the data of follower and following users. Frontend request to backend with userId which will be followed. Follow functionality looks if user who sent request is already following that user or not. If already following, nothing happens. If not following, follow function creates an instance and saves it to the database.

Link to API: http://3.122.41.188:8000

API documentation:
https://github.com/bounswe/2021SpringGroup5/wiki/LUDO-Project-API-Documentation

Provide three examples of API calls associated with the most significant core functionality of your application. All information needed to perform the calls must be provided (i.e. required parameter values). In other words, they must be replicable to us. You may share Postman collections.: https://documenter.getpostman.com/view/16110021/UVXeqxG1

For each User Interface/User Experience design:
URI to the code in the project repository: Frontend , Mobile

Provide API calls that return JSON-LD responses

1) **Create event post (POST request)**

Input json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Gözde is creating an event post",
   "type":"Create",
   "actor":{
      "type":"Person",
      "name":"Gozde",
      "surname":"Unver",
      "username":"gozde_unver",
      "Id":32
   },
   "object":{
      "type":"EventPost",
      "post_name":"kite activity",
      "sport_category":"kite",
      "longitude":29.037842,
      "latitude":40.979566,
      "description":"I am looking for a player who can fly kites with me",
      "date_time":"2022-01-25 18:30",
      "participant_limit":3,
      "spectator_limit":2,
      "rule":"No rules",
      "equipment_requirement":"bring your own kite",
      "location_requirement":"",
      "contact_info":"05550000000",
      "skill_requirement":"beginner",
      "repeating_frequency":0
   }
}
```

Output json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Gözde is creating an event post",
   "type":"Create",
   "actor":{
      "name":"Gozde",
      "surname":"Unver",
      "username":"gozde_unver",
      "Id":32,
      "type":"Person"
   },
   "object":{
      "id":45,
      "post_name":"kite activity",
      "created_date":"2022-01-07 19:41:53",
      "description":"I am looking for a player who can fly kites with me",
      "longitude":29.037842,
      "latitude":40.979566,
      "date_time":"2022-01-25 18:30:00",
      "participant_limit":3,
      "spectator_limit":2,
      "rule":"No rules",
      "equipment_requirement":"bring your own kite",
      "status":"upcoming",
      "capacity":"open to applications",
      "location_requirement":"",
      "contact_info":"05550000000",
      "pathToEventImage":"https://nzftk20rg4.execute-api.eu-central-1.amazonaws.com/v1/lodobucket451?file=EventPost_45.jpg",
      "current_player":0,
      "current_spectator":0,
      "owner":{
         "Id":32,
         "name":"Gozde",
         "username":"gozde_unver",
         "surname":"Unver"
      },
      "sport_category":"kite",
      "skill_requirement":"beginner",
      "type":"EventPost"
   }
}
```


2) Create equipment post (POST request)

Input json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Gözde is creating an equipment post",
   "type":"Create",
   "actor":{
      "type":"Person",
      "Id":32,
      "name":"Gozde",
      "surname":"Unver",
      "username":"gozde_unver"
   },
   "object":{
      "type":"EquipmentPost",
      "post_name":"basketball ball",
      "sport_category":"basketball",
      "description":"this ball looks great you should check it out",
      "link":"https://www.adidas.com.tr/tr/all-court-2.0-basketbol-topu/GL3946.html"
   }
}
```

Output json:


```
{
  "@context":"https://www.w3.org/ns/activitystreams",
  "summary":"Gözde is creating an equipment post",
  "type":"Create",
  "actor":{
    "type":"Person",
    "Id":32,
    "name":"Gozde",
    "surname":"Unver",
    "username":"gozde_unver"
  },
  "object":{
    "id":15,
    "post_name":"basketball ball",
    "created_date":"2022-01-07 19:55:12",
    "description":"this ball looks great you should check it out",
    "longitude":null,
    "latitude":null,
    "link":"https://www.adidas.com.tr/tr/all-court-2.0-basketbol-topu/GL3946.html",
    "active":true,
    "pathToEquipmentPostImage":"https://nzftk20rg4.execute-api.eu-central-1.amazonaws.com/v1/lodobucket451?file=EquipmentPost_15.jpg",
    "owner":{
      "Id":32,
      "name":"Gozde",
      "surname":"Unver",
      "username":"gozde_unver"
    },
    "sport_category":"basketball",
    "type":"EquipmentPost"
  }
}
```


3) Change equipment post information (PATCH request)

Input json:


```
{
  "@context":"https://www.w3.org/ns/activitystreams",
  "summary":"Gozde updated an equipment post",
  "type":"Update",
  "actor":{
    "type":"Person",
    "Id":32,
    "name":"Gozde",
    "surname":"Unver",
    "username":"gozde_unver"
  },
  "object":{
    "type":"EquipmentPost",
    "post_id":21
  },
  "modifications":{
    "post_name":"adidas football ball"
  }
}
```

Output json:


```
{
  "@context":"https://www.w3.org/ns/activitystreams",
  "summary":"Gozde updated an equipment post",
  "actor":{
    "type":"Person",
    "Id":32,
    "name":"Gozde",
    "surname":"Unver",
    "username":"gozde_unver"
  },
  "type":"Update",
  "object":{
    "id":21,
    "post_name":"adidas football ball",
    "owner":{
      "Id":32,
      "name":"Gozde",
      "surname":"Unver",
      "username":"gozde_unver"
    },
    "sport_category":"football",
    "created_date":"2022-01-07 22:23:10",
    "description":"this ball is very cheap check it out",
    "longitude":null,
    "latitude":null,
    "link":"https://www.adidas.com.tr/tr/ucl-mini-real-madrid-pyrostorm-top/GU0208.html?pr=product_rr&slot=8&rec=mt",
    "active":true,
    "pathToEquipmentPostImage":"https://nzftk20rg4.execute-api.eu-central-1.amazonaws.com/v1/lodobucket451?file=EquipmentPost_21.jpg"
  }
}
```


4) **Change event post information (PATCH request)**

Input json:


```
{
  "@context":"https://www.w3.org/ns/activitystreams",
  "summary":"Gozde is updating event post",
  "type":"Update",
  "actor":{
    "type":"Person",
    "name":"Gozde",
    "surname":"Unver",
    "username":"gozde_unver",
    "Id":32
  },
  "object":{
    "type":"EventPost",
    "post_id":45
  },
  "modifications":{
    "participant_limit":5,
    "spectator_limit":5,
    "contact_info":"05553333333"
  }
}
```

Output json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Gozde is updating event post",
   "actor":{
      "type":"Person",
      "name":"Gozde",
      "surname":"Unver",
      "username":"gozde_unver",
      "Id":32
   },
   "type":"Update",
   "object":{
      "id":45,
      "post_name":"kite activity",
      "owner":{
         "Id":32,
         "name":"Gozde",
         "surname":"Unver",
         "username":"gozde_unver"
      },
      "sport_category":"kite",
      "created_date":"2022-01-07 19:41:53",
      "description":"I am looking for a player who can fly kites with me",
      "longitude":29.037842,
      "latitude":40.979566,
      "date_time":"2022-01-25 15:30:00",
      "participant_limit":5,
      "spectator_limit":5,
      "rule":"No rules",
      "equipment_requirement":"bring your own kite",
      "status":"upcoming",
      "capacity":"open to applications",
      "location_requirement":"",
      "contact_info":"05553333333",
      "pathToEventImage":"https://nzftk20rg4.execute-api.eu-central-1.amazonaws.com/v1/lodobucket451?file=EventPost_45.jpg",
      "skill_requirement":"beginner",
      "current_player":0,
      "current_spectator":0
   }
}
```


5) **Delete equipment post (DELETE request)**

Input json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Gozde deleted an event post",
   "type":"Delete",
   "actor":{
      "type":"Person",
      "Id":32,
      "name":"Gozde",
      "surname":"Unver",
      "username":"gozde_unver"
   },
   "object":{
      "type":"EquipmentPost",
      "post_id":21
   },
   "origin":{
      "type":"Collection",
      "name":"Gozde's equipment posts"
   }
}
```

Output json:

For the deletion, frontend and mobile only check the status code so the output json is not implemented in activity stream format.


```
{
   "message":"Equipment post is deleted"
}
```


and the returned status code is 200

6) **Get details of an event post (POST request)**

Input json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Gozde read an event post",
   "type":"View",
   "actor":{
      "type":"Person",
      "name":"Gozde",
      "surname":"Unver",
      "username":"gozde_unver",
      "Id":32
   },
   "object":{
      "type":"EventPost",
      "post_id":45
   }
}
```

Output json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Gozde read an event post",
   "type":"View",
   "actor":{
      "name":"Gozde",
      "surname":"Unver",
      "username":"gozde_unver",
      "Id":32,
      "type":"Person"
   },
   "object":{
      "id":45,
      "post_name":"kite activity",
      "owner":{
         "Id":32,
         "name":"Gozde",
         "surname":"Unver",
         "username":"gozde_unver"
      },
      "sport_category":"kite",
      "created_date":"2022-01-07 19:41:53",
      "description":"I am looking for a player who can fly kites with me",
      "longitude":29.037842,
      "latitude":40.979566,
      "date_time":"2022-01-25 15:30:00",
      "participant_limit":5,
      "spectator_limit":5,
      "rule":"No rules",
      "equipment_requirement":"bring your own kite",
      "status":"upcoming",
      "capacity":"open to applications",
      "location_requirement":"",
      "contact_info":"05553333333",
      "pathToEventImage":"https://nzftk20rg4.execute-api.eu-central-1.amazonaws.com/v1/lodobucket451?file=EventPost_45.jpg",
      "skill_requirement":"beginner",
      "current_player":0,
      "current_spectator":0,
      "comments":[

      ],
      "spectators":[


      ],
      "inadequate_player_applications":[

      ],
      "waiting_players":[

      ],
      "accepted_players":[

      ],
      "rejected_players":[

      ],
      "is_event_creator":true,
      "type":"EventPost"
   }
}
```

7) **Get details of an equipment post (POST request)**

Input json:


```
{
    "@context":"https://www.w3.org/ns/activitystreams",
    "summary":"Gozde read an equipment post",
    "type":"View",
    "actor":{
        "type":"Person",
        "name":"Gozde",
        "surname":"Unver",
        "username":"gozde_unver",
        "Id":32
    },
    "object":{
        "type":"EquipmentPost",
        "post_id":15
    }
}
```


Output json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Gozde read an equipment post",
   "type":"View",
   "actor":{
      "type":"Person",
      "name":"Gozde",
      "surname":"Unver",
      "username":"gozde_unver",
      "Id":32
   },
   "object":{
      "id":15,
      "post_name":"basketball ball",
      "owner":{
         "Id":32,
         "name":"Gozde",
         "surname":"Unver",
         "username":"gozde_unver"
      },
      "sport_category":"basketball",
      "created_date":"2022-01-07 19:55:12",
      "description":"this ball looks great you should check it out",
      "longitude":null,
      "latitude":null,
      "link":"https://www.adidas.com.tr/tr/all-court-2.0-basketbol-topu/GL3946.html",
      "active":true,
      "pathToEquipmentPostImage":"https://nzftk20rg4.execute-api.eu-central-1.amazonaws.com/v1/lodobucket451?file=EquipmentPost_15.jpg",
      "is_event_creator":true,
      "comments":[

      ],
      "type":"EventPost"
   }
}
```

8) **Get event post analytics (POST request)**

Input json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Gozde viewed event post analytics",
   "type":"View",
   "actor":{
      "type":"Person",
      "id":32,
      "name":"Gozde",
      "surname":"Unver",
      "username":"gozde_unver"
   },
   "object":{
      "type":"EventPost",
      "post_id":45
   }
}
```


Output json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Gozde viewed event post analytics",
   "type":"View",
   "actor":{
      "type":"Person",
      "id":32,
      "name":"Gozde",
      "surname":"Unver",
      "username":"gozde_unver"
   },
   "object":{
      "type":"EventPost",
      "post_id":45,
      "avg_accepted_event_count_of_accepted_users":0
   }
}
```

9) **Apply to event (POST request)**

Input json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Salih applied to an event",
   "type":"Application",
   "object":{
      "type":"EventPost",
      "Id":18
   }
}
```


Output json:


```
{
   "message":"Application is successfully created"
}
```


10) **Spectate to event (POST request)**

Input json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Salih spectates to an event",
   "type":"Spectate",
   "object":{
      "type":"EventPost",
      "Id":30
   }
}
```


Output json:


```
{
   "message":"Spectate application is successfully created"
}
```

11) Accept application (POST request)

Input json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Salih accepted an application",
   "type":"Accept",
   "applicant":{
      "Id":5
   },
   "object":{
      "type":"EventPost",
      "Id":2
   }
}
```


Output json:


```
{
   "message":"Application is accepted"
}
```


12) Reject application (POST request)

Input json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Salih rejected an application",
   "type":"Reject",
   "applicant":{
      "Id":5
   },
   "object":{
      "type":"EventPost",
      "Id":20
   }
}
```

Output json:


```
{
    "message":"Application is rejected"
}
```


13) Postpone event (PATCH request)

Input json:


```
{
    "@context":"https://www.w3.org/ns/activitystreams",
    "summary":"Salih postponed an event",
    "type":"Postpone",
    "object":{
        "type":"EventPost",
        "post_id":29
    },
    "new_date":"13/12/2025:10"
}
```


Output json:


```
[
    {
        "id":29,
        "post_name":"tenis maci",
        "owner_id":33,
        "created_date":"2022-01-07 01:54:20",
        "description":"teniste iddialiyimdir",
        "longitude":29.5,
        "latitude":40.0,
        "date_time":"2025-12-13 07:00:00",
        "participant_limit":1,
        "spectator_limit":0,
        "rule":"no rules extra",
        "equipment_requirement":"",
        "status":"upcoming",
        "capacity":"open to applications",
        "location_requirement":"",
        "contact_info":"054155555",
        "pathToEventImage":"https://nzftk20rg4.execute-api.eu-central-1.amazonaws.com/v1/lodobucket451?file=EventPost_29.jpg",
        "current_player":0,
        "current_spectator":0,
        "sport_name":"tennis",
        "skill_requirement":"beginner"
    }
]
```


14) Delete event post (DELETE request)

Input json:

```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Salih deleted an event post",
   "type":"Delete",
   "object":{
      "type":"EventPost",
      "post_id":29
   }
}
```


Output json:


```
{
   "message":"Event post is deleted"
}
```


15) Create event comment (POST request)

Input json:


```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Salih is creating a comment",
   "type":"Create",
   "object":{
      "type":"Comment",
      "content":"I know the place! :)",
      "post_id":5
   }
}
```


Output json:


```
{
   "message":"Comment created successfully"
}
```

16) **Create equipment comment (POST request)**

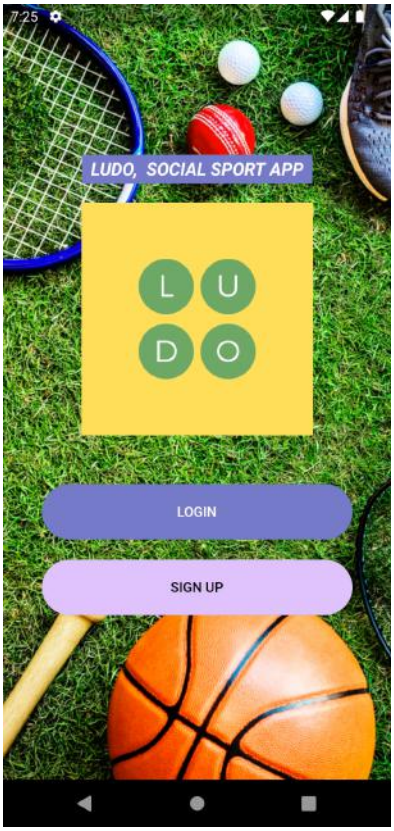
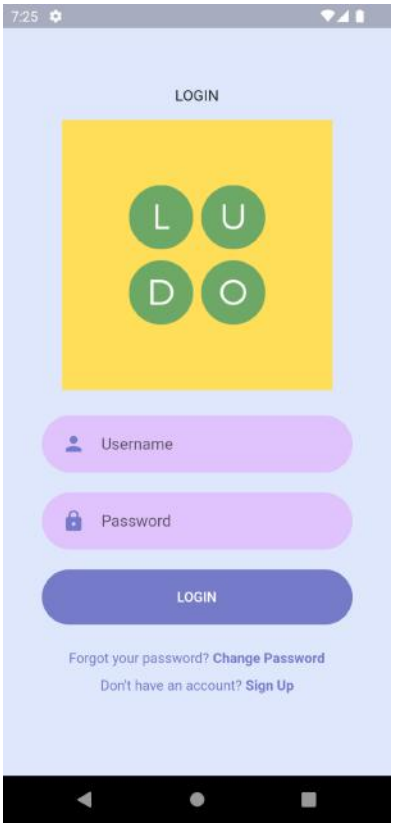
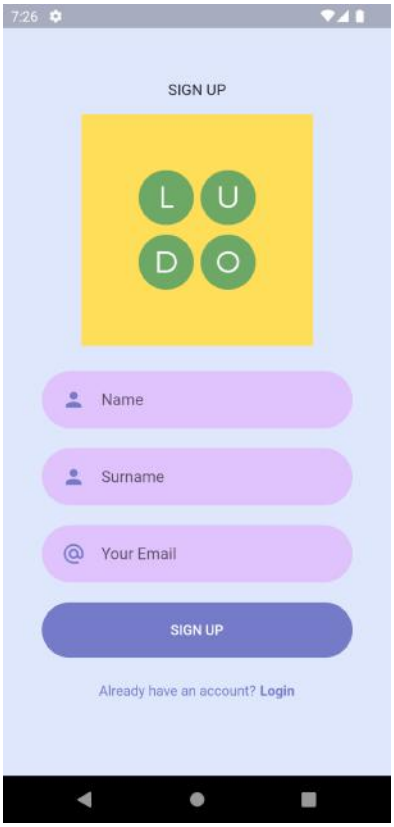
Input json:

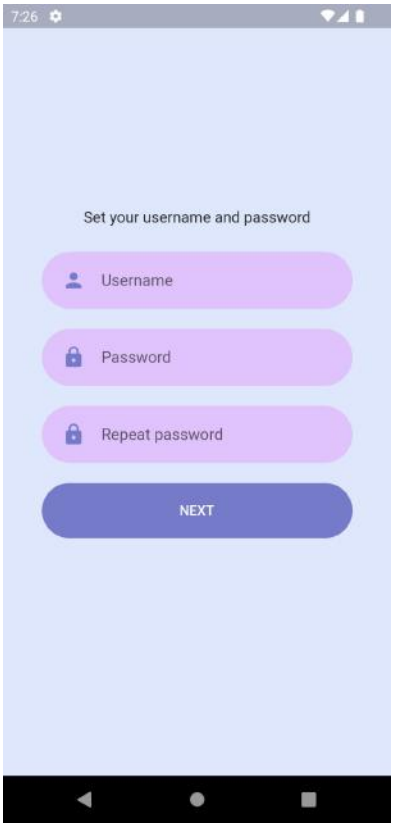
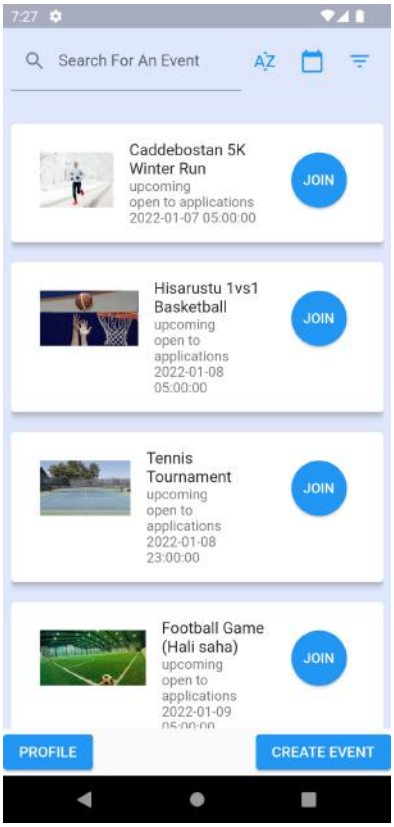

```
{
   "@context":"https://www.w3.org/ns/activitystreams",
   "summary":"Salih is creating a comment",
   "type":"Create",
   "object":{
      "type":"Comment",
      "content":"I want that racket",
      "post_id":20
   }
}
```

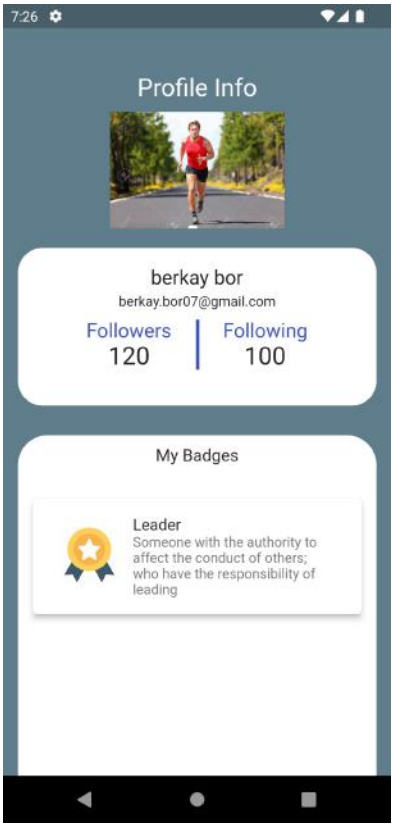
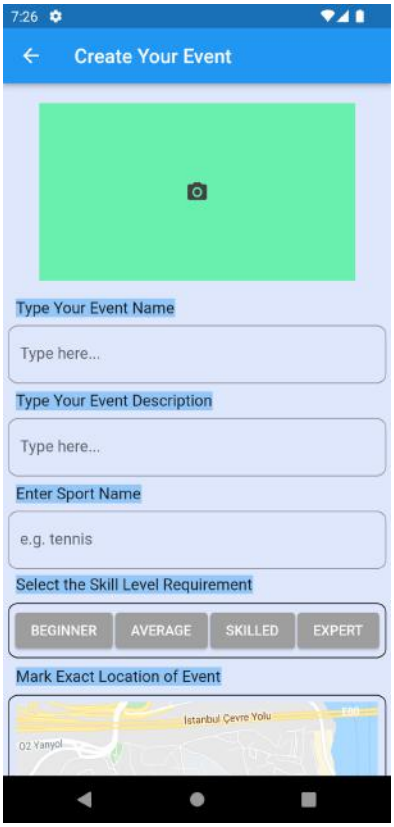
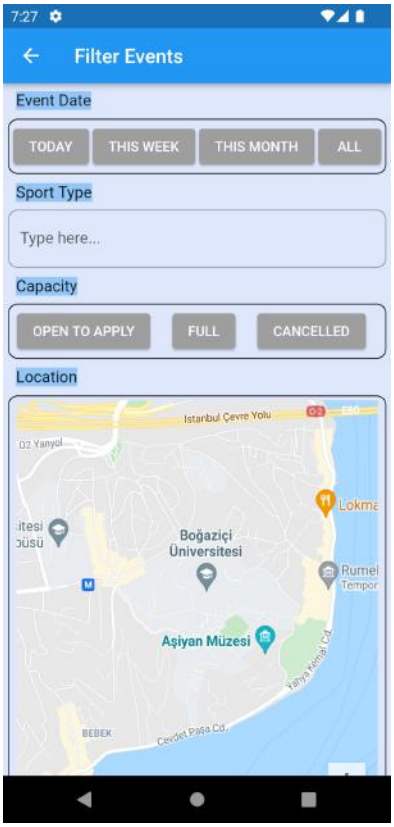

Output json:

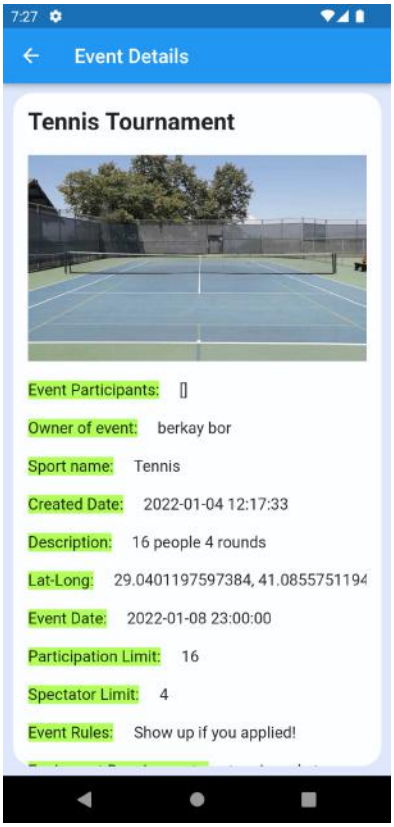
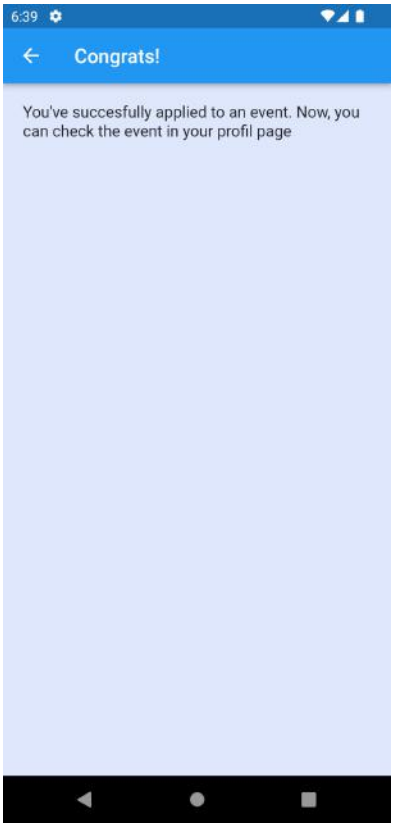
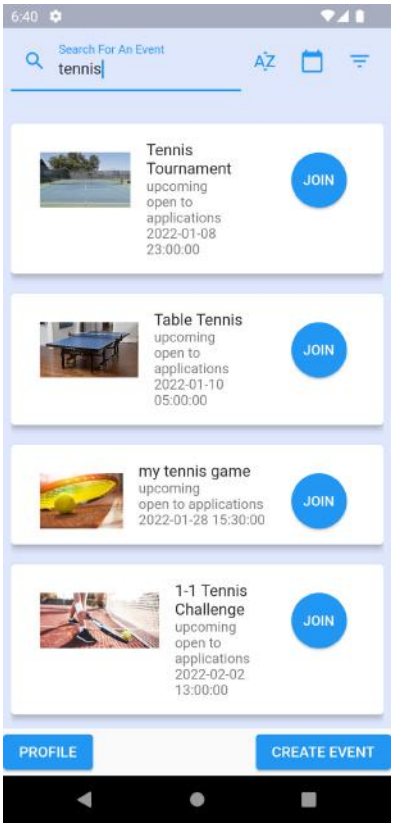

```
{
   "message":"Comment created successfully"
}
```

- Screenshot of all implemented Mobile interfaces:

MAIN SCREEN	LOGIN	SIGN UP STEP 1
		

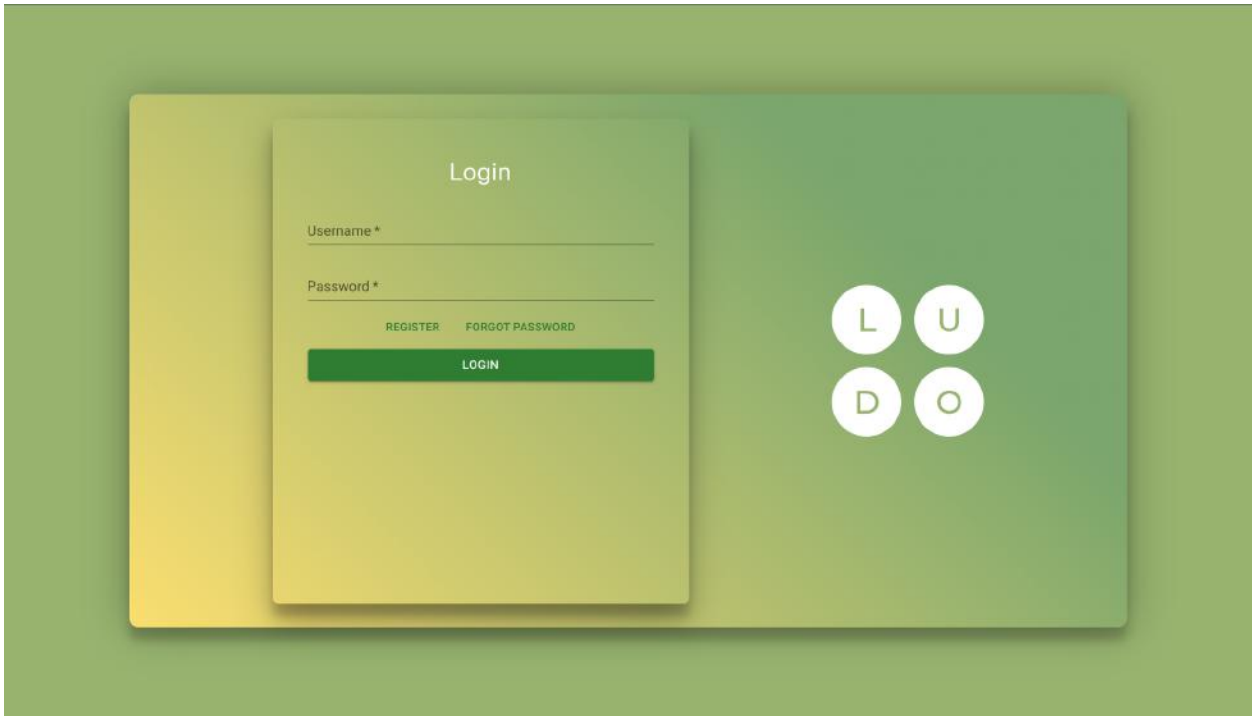
SIGN UP STEP 2	SIGN UP STEP 3	HOME SCREEN
		

PROFILE SCREEN	CREATE EVENT	FILTER EVENTS
		

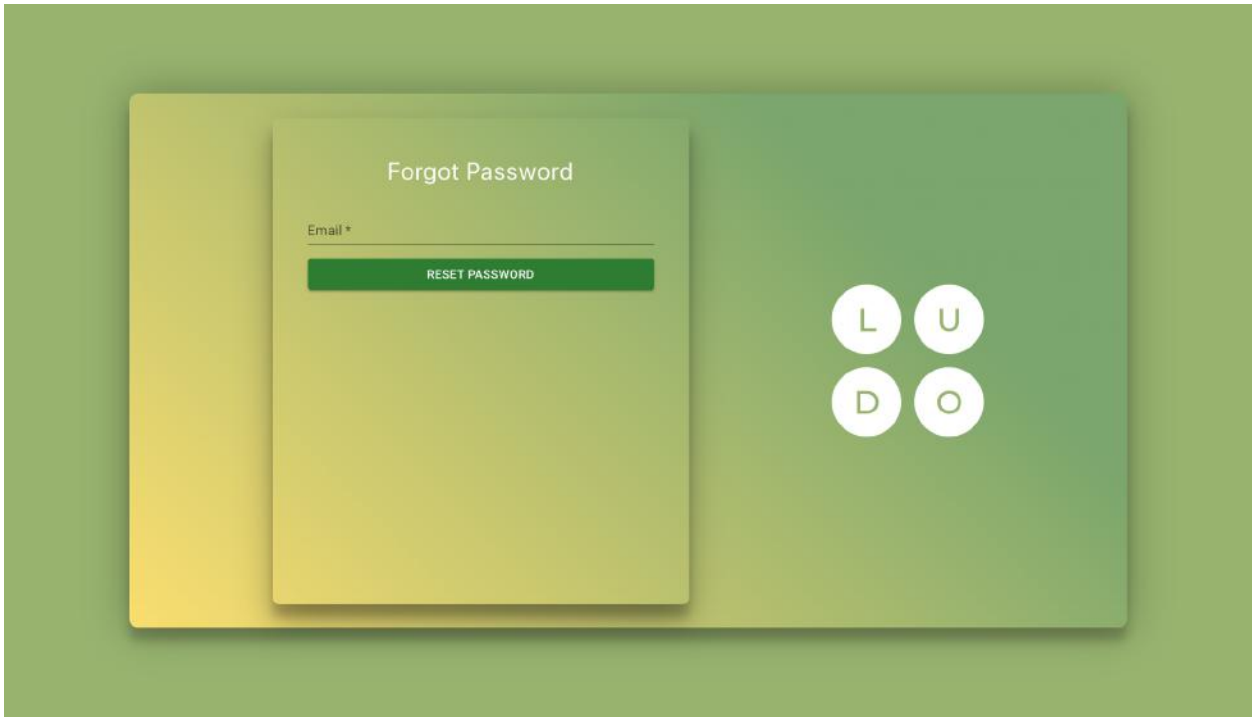
EVENT DETAILS	APPLY TO EVENT	SEARCH EVENTS
		

Screenshots of all implemented Web interfaces:

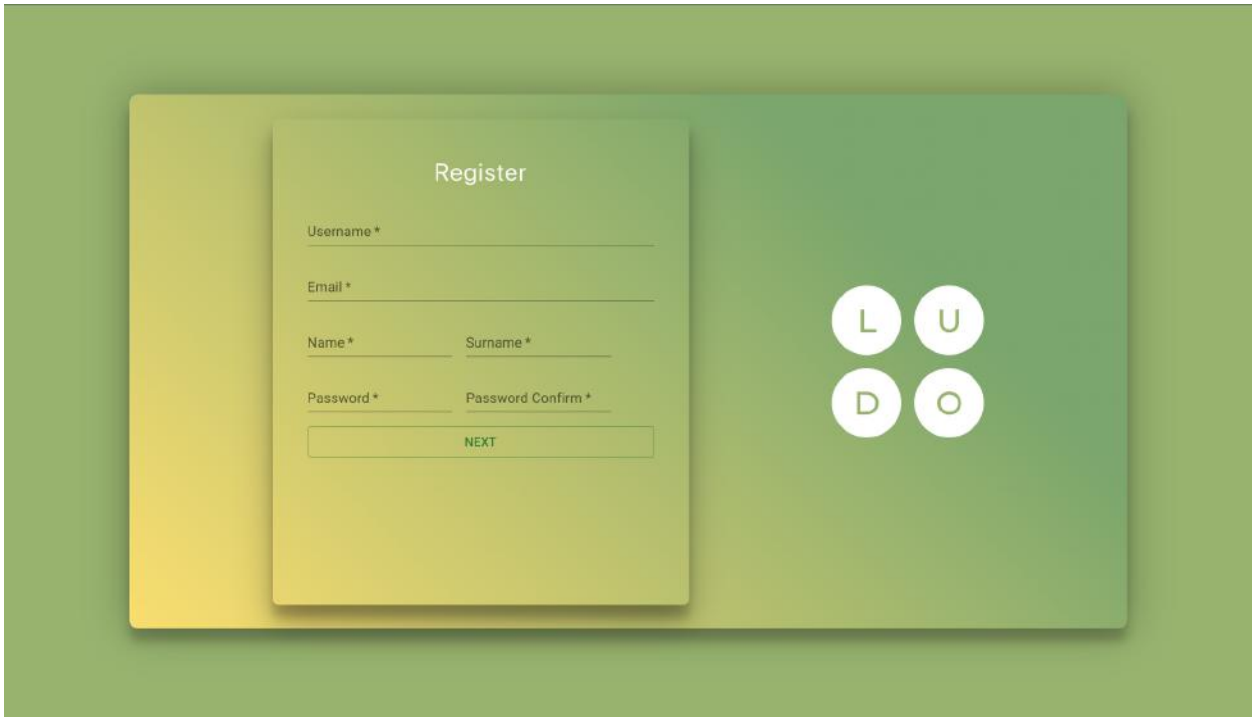
LOGIN



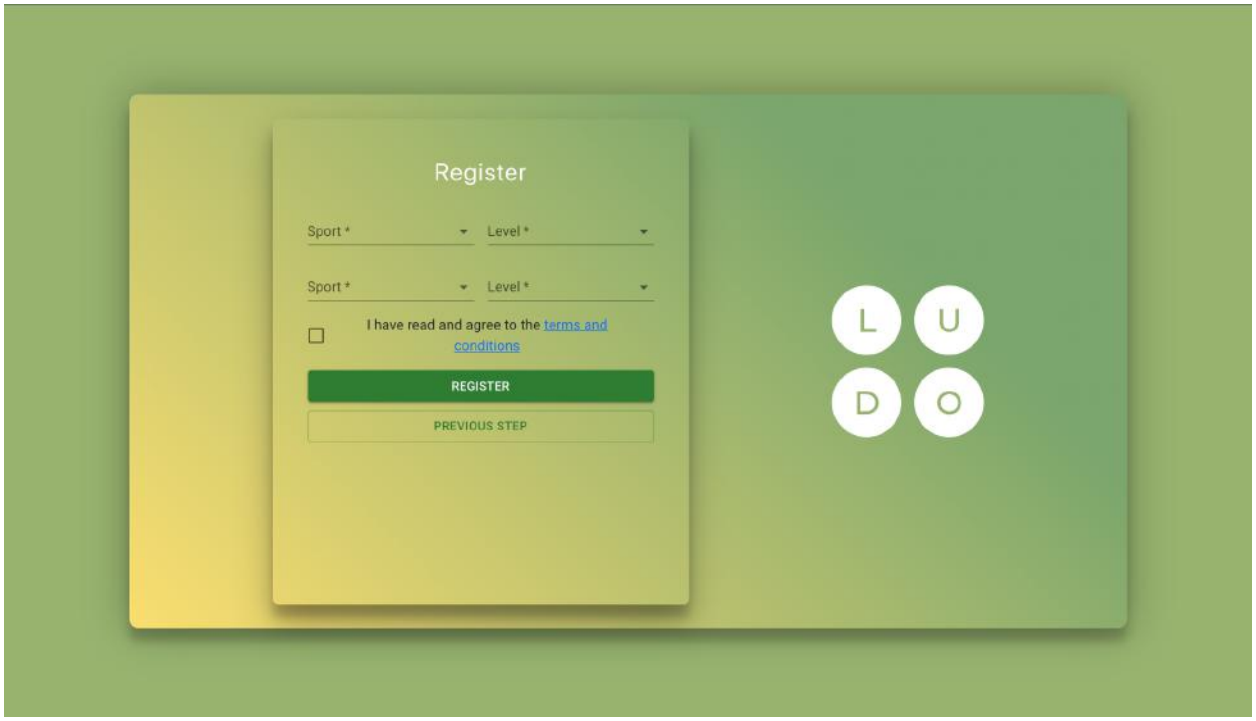
FORGOT PASSWORD



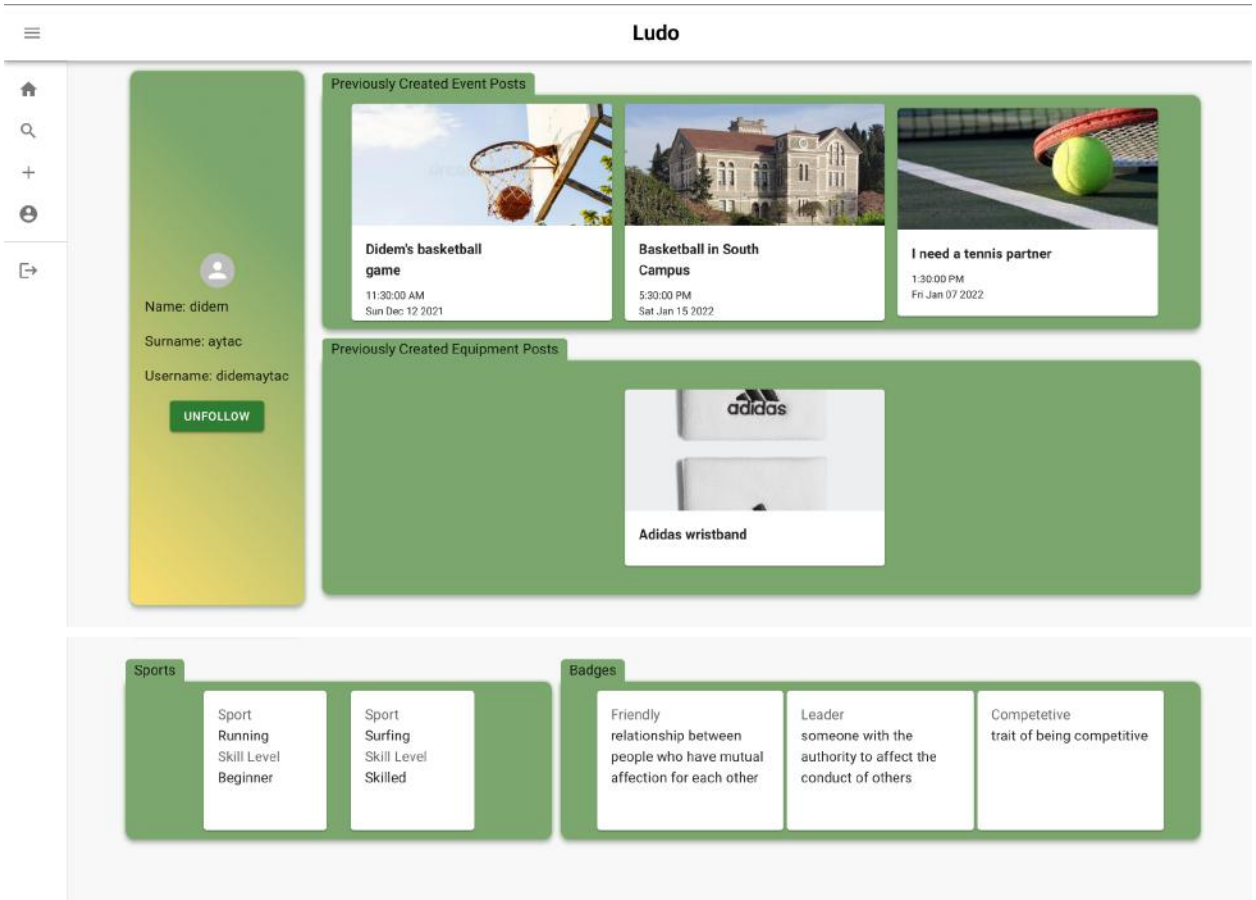
REGISTER STEP 1



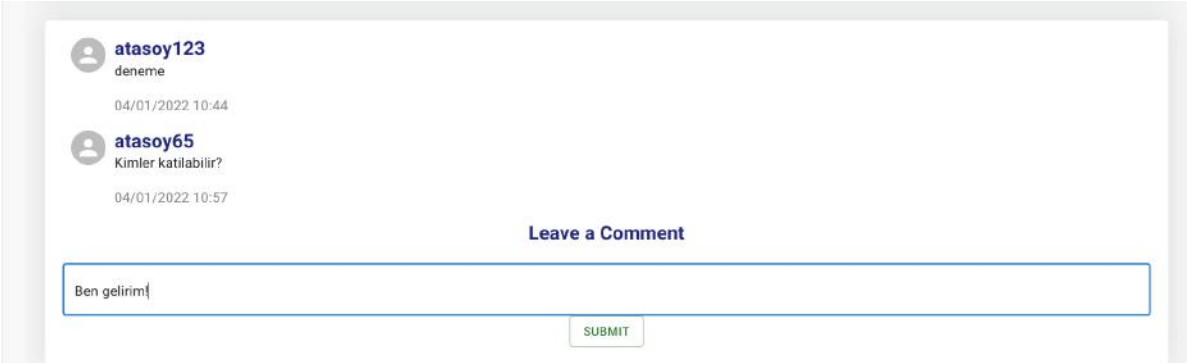
REGISTER STEP 2



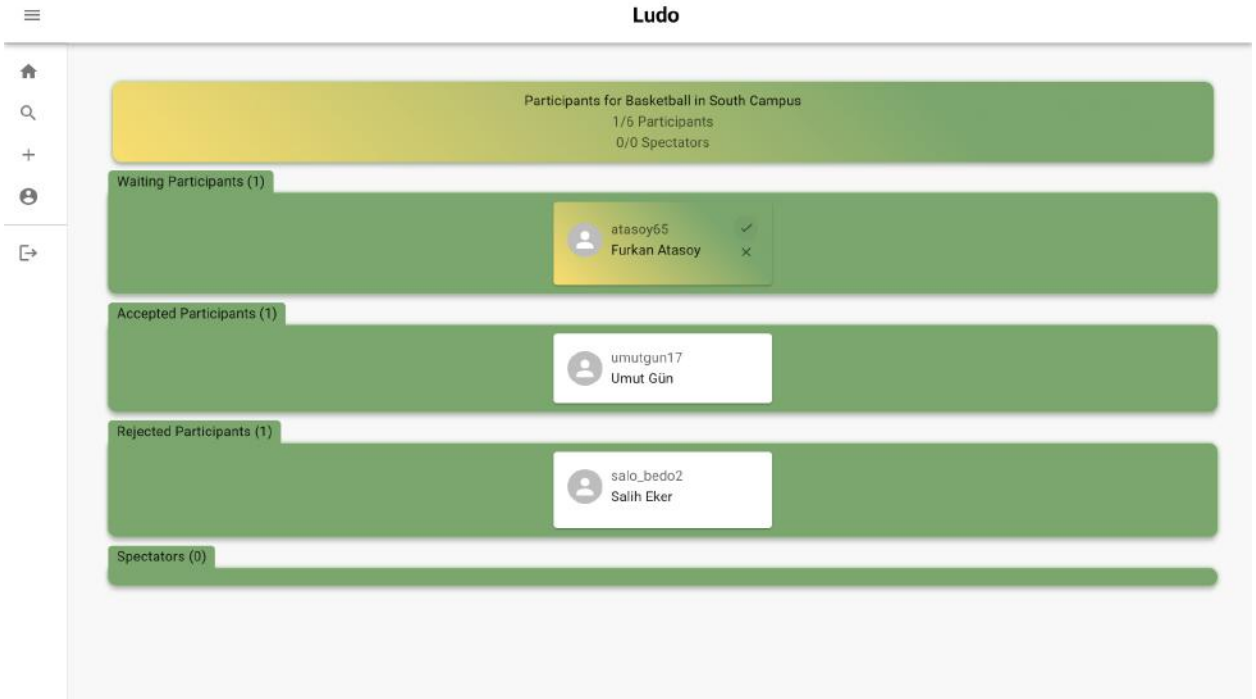
PROFILE



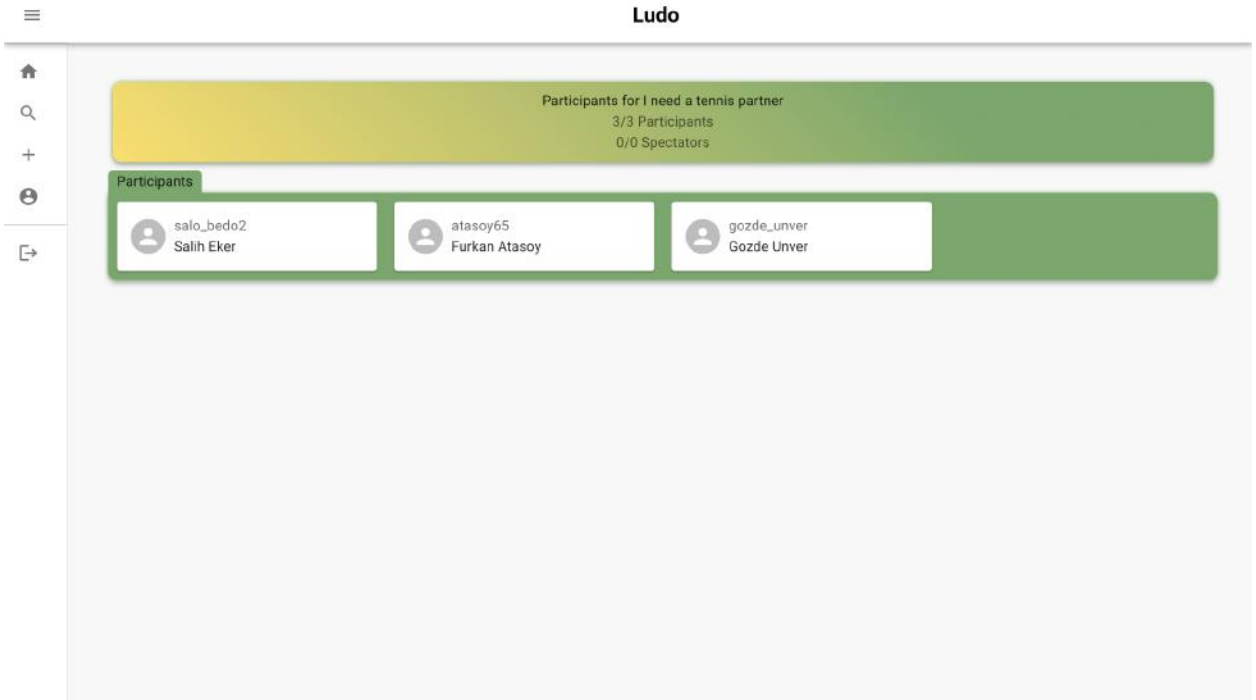
COMMENT SECTION



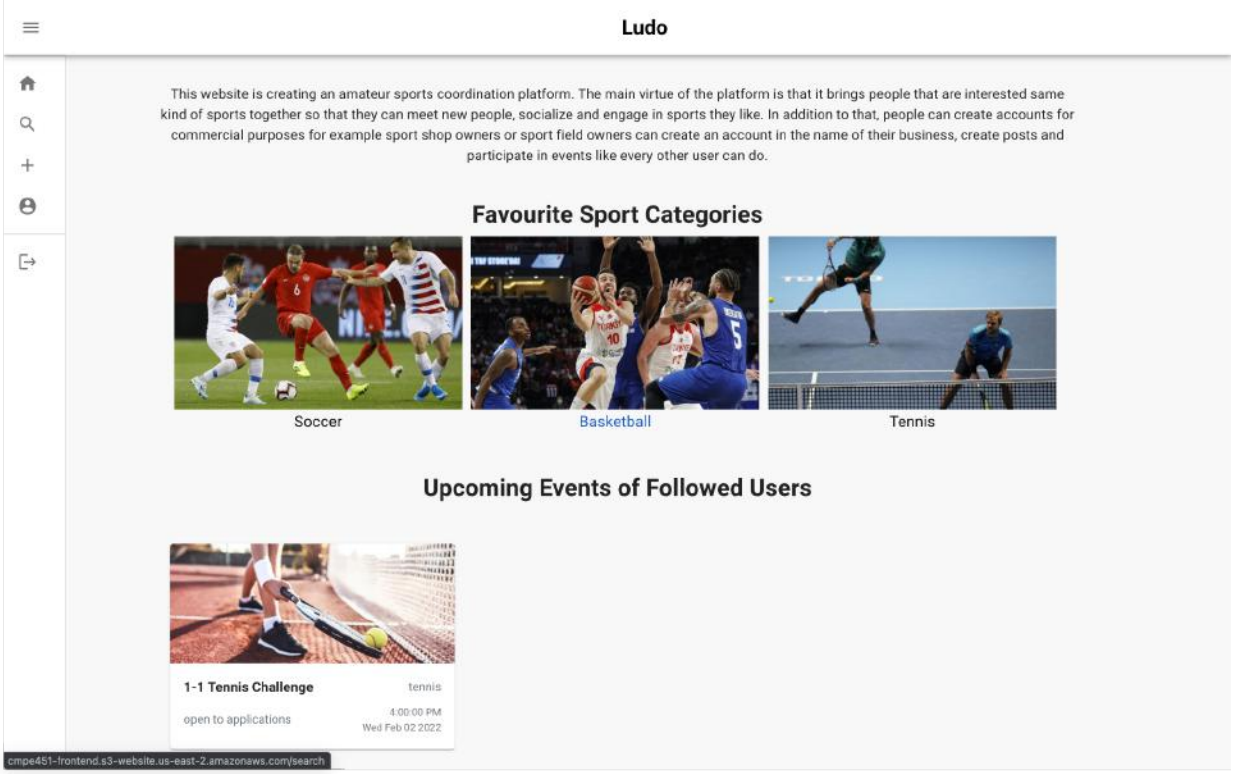
PARTICIPANTS SCREEN (EVENT OWNER)



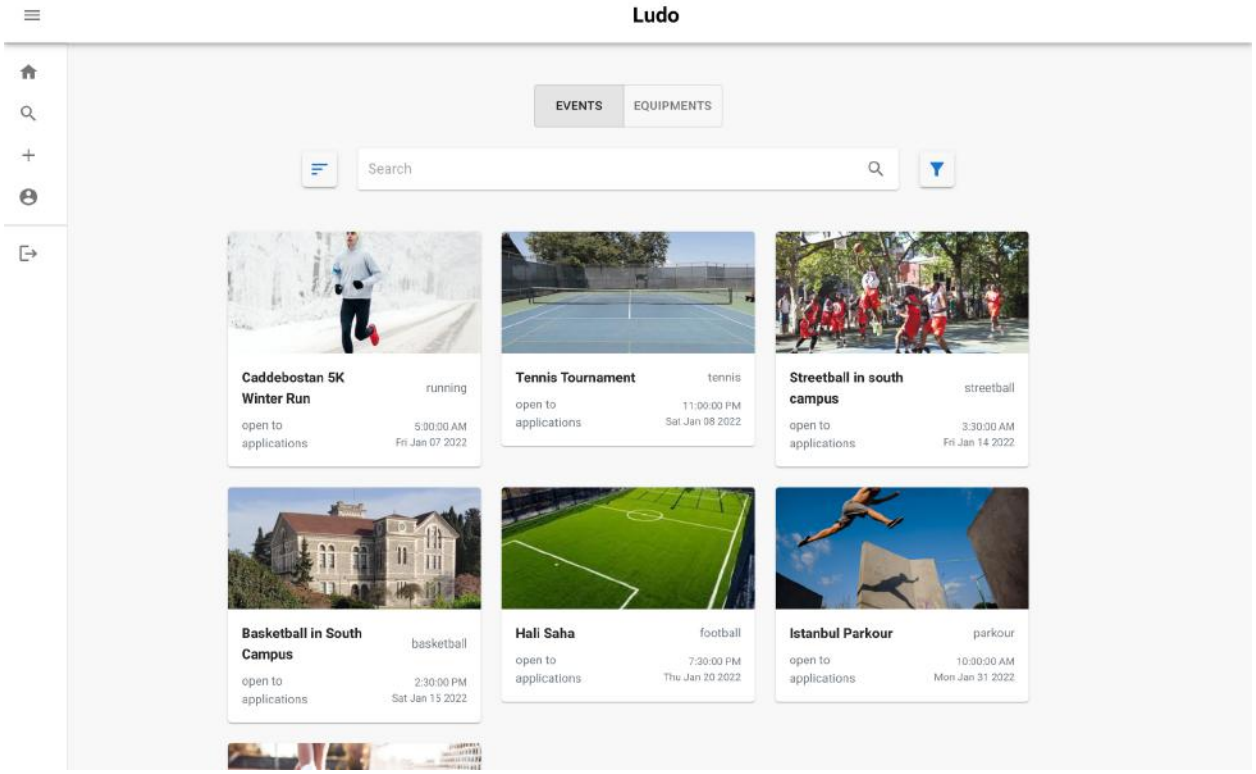
PARTICIPANTS SCREEN (NOT EVENT OWNER)



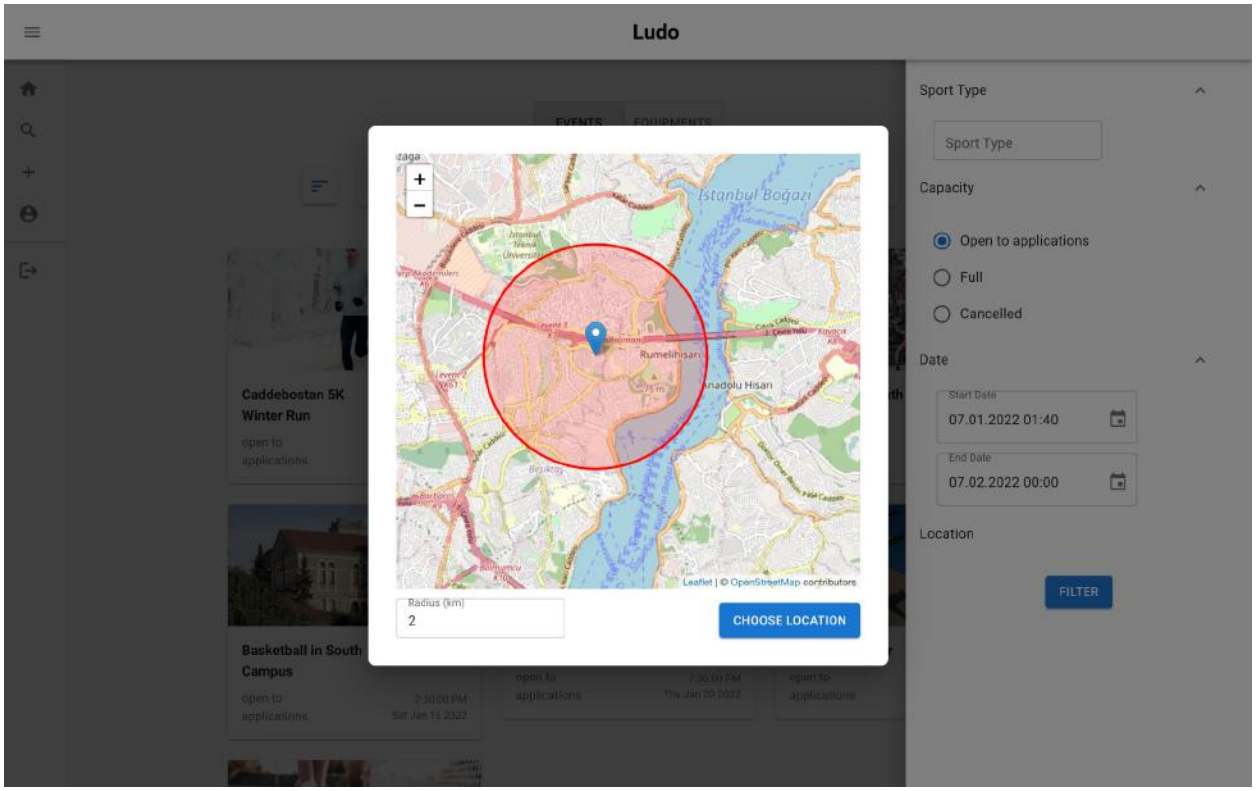
HOME SCREEN



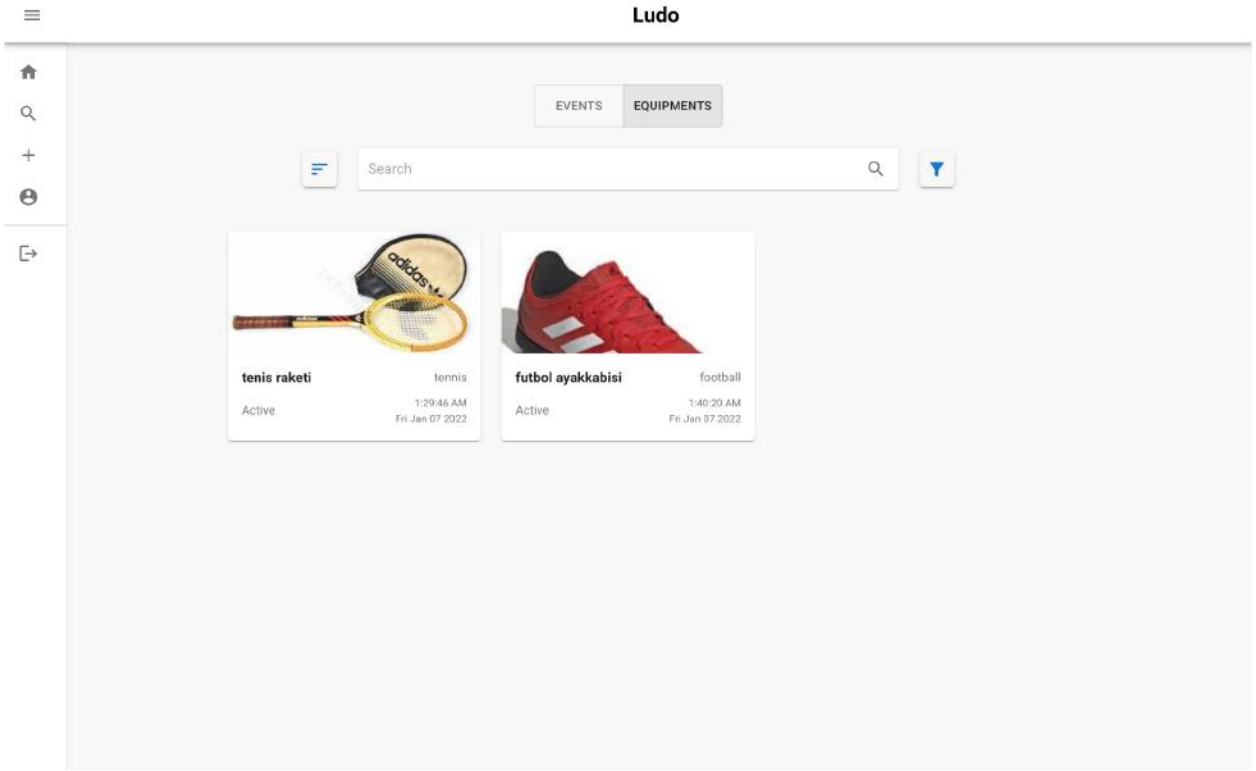
EVENT SEARCH SCREEN



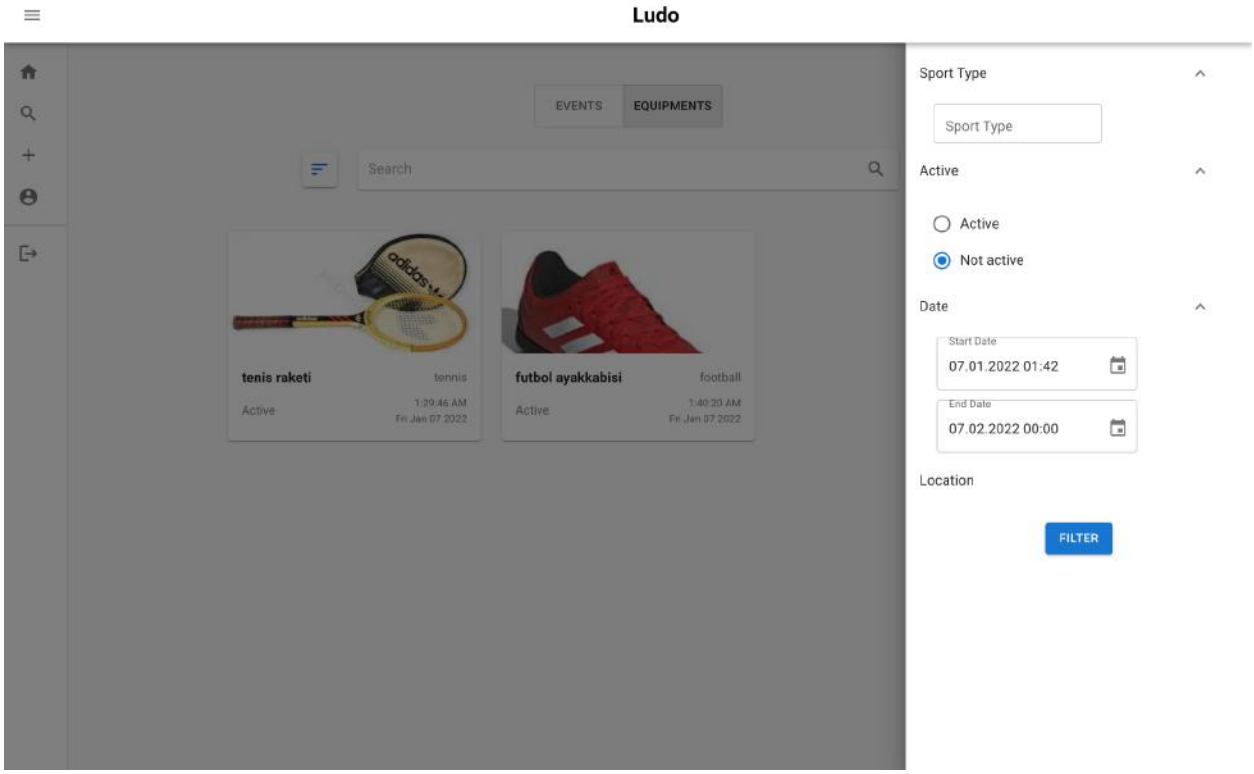
EVENT FILTER SCREEN



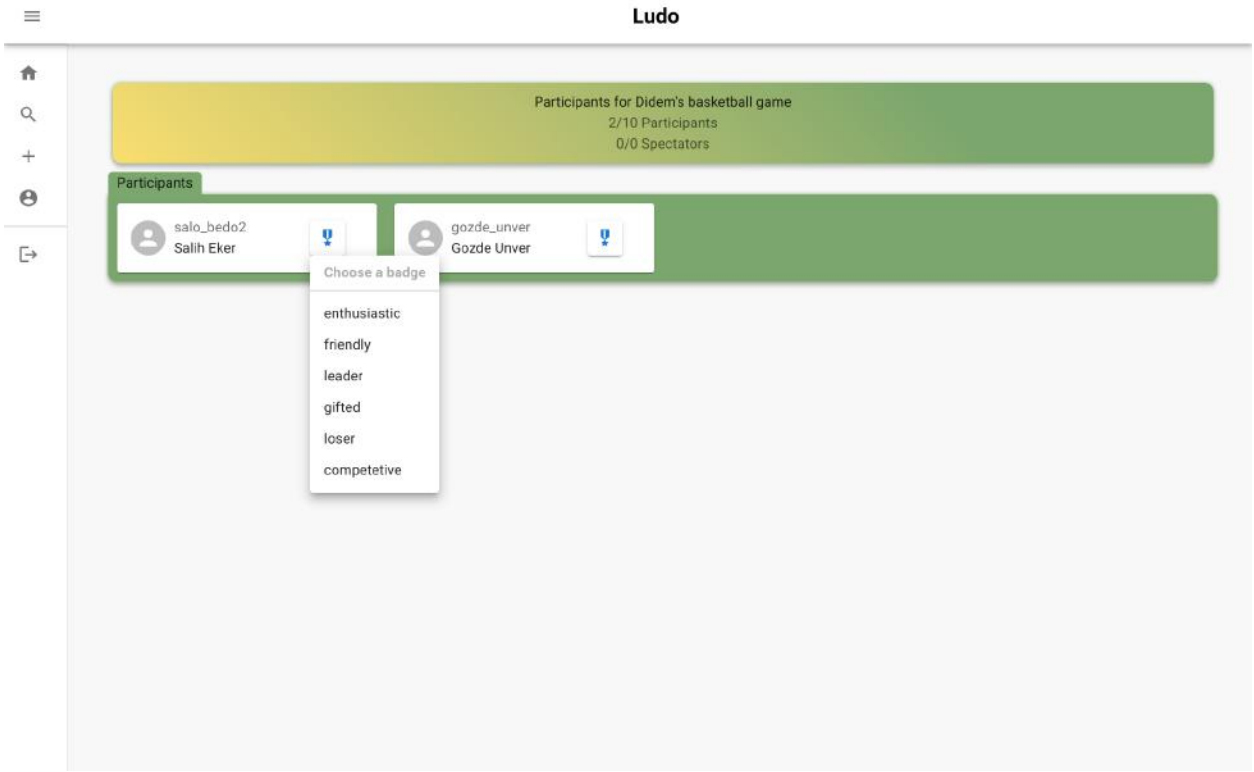
EQUIPMENT SEARCH SCREEN



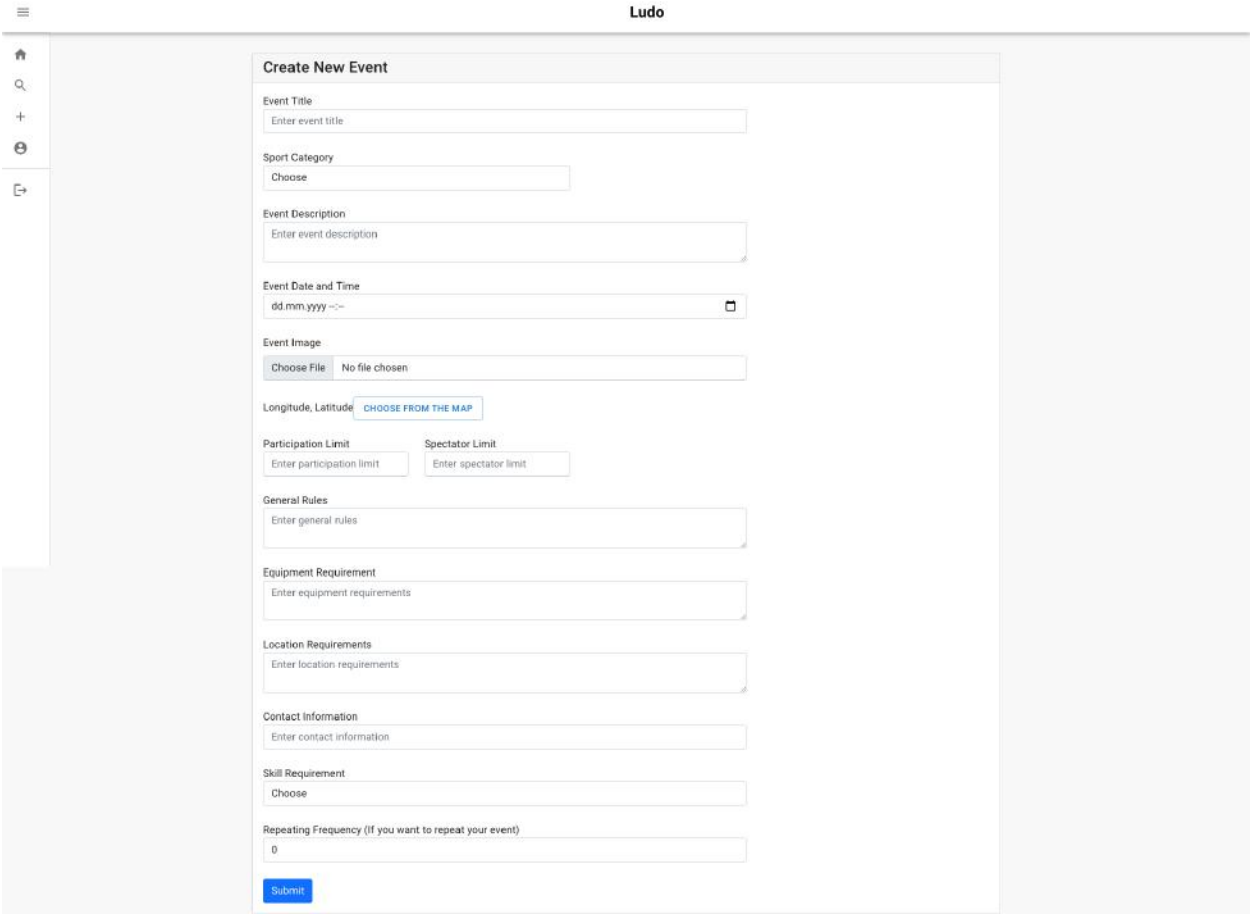
EQUIPMENT FILTER SCREEN



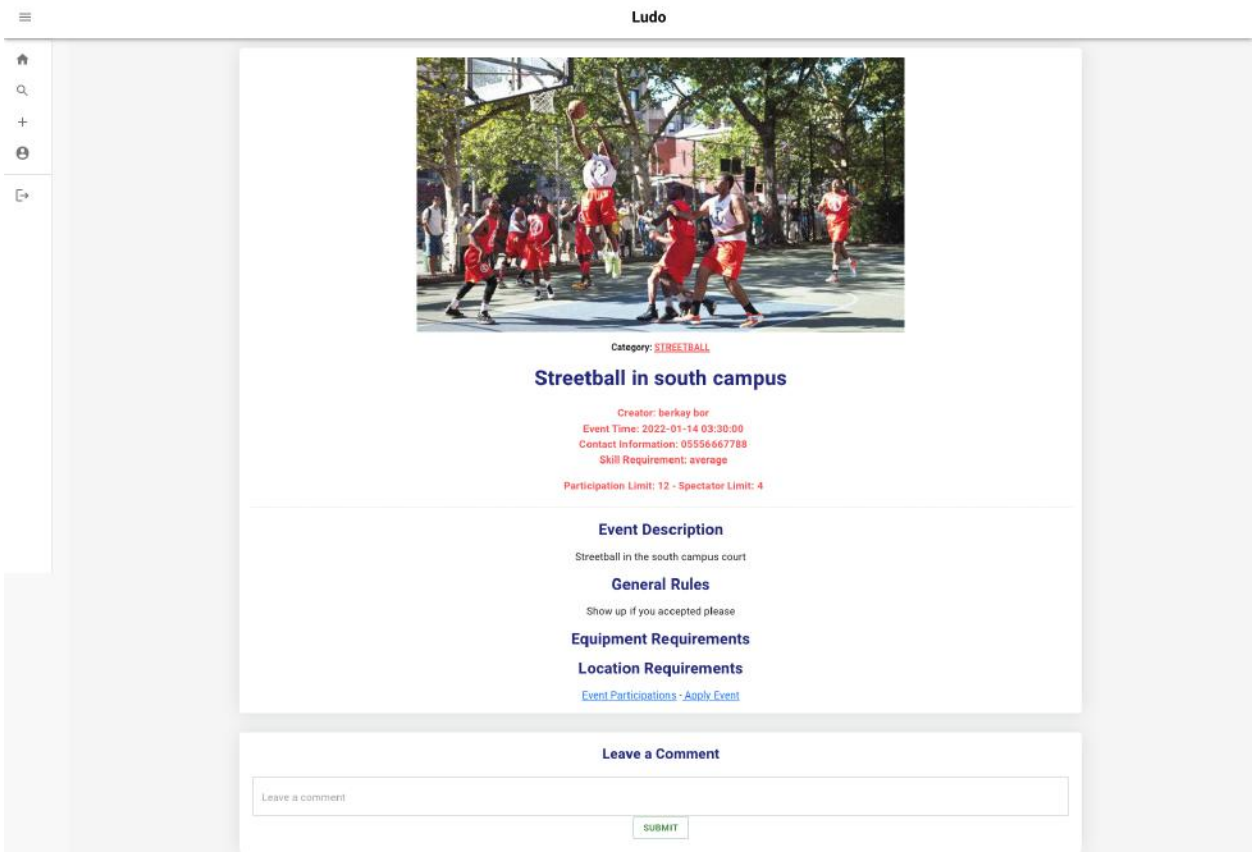
PARTICIPANT LISTING PAGE BADGE UI



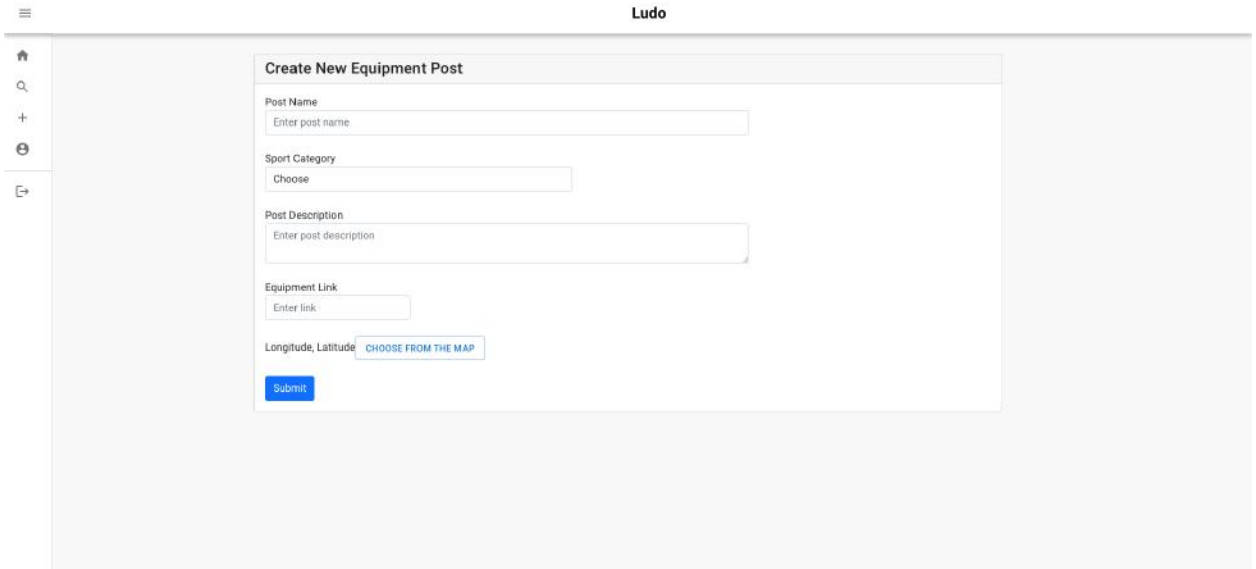
CREATION EVENT



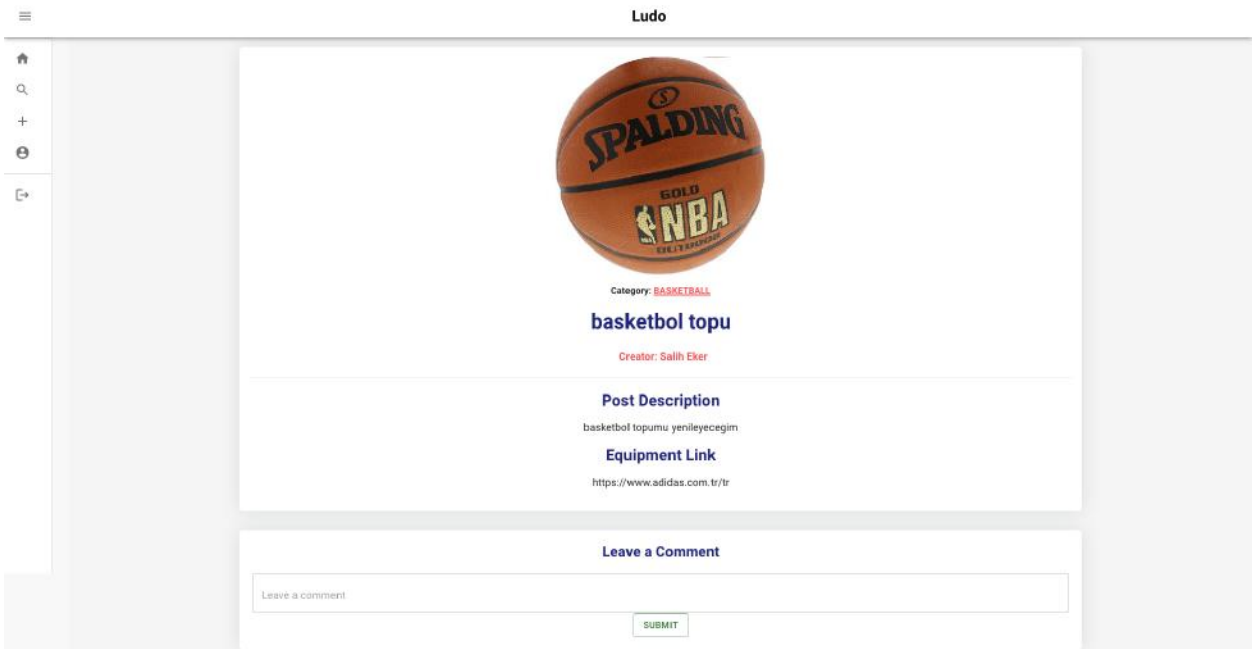
DETAIL EVENT PAGE



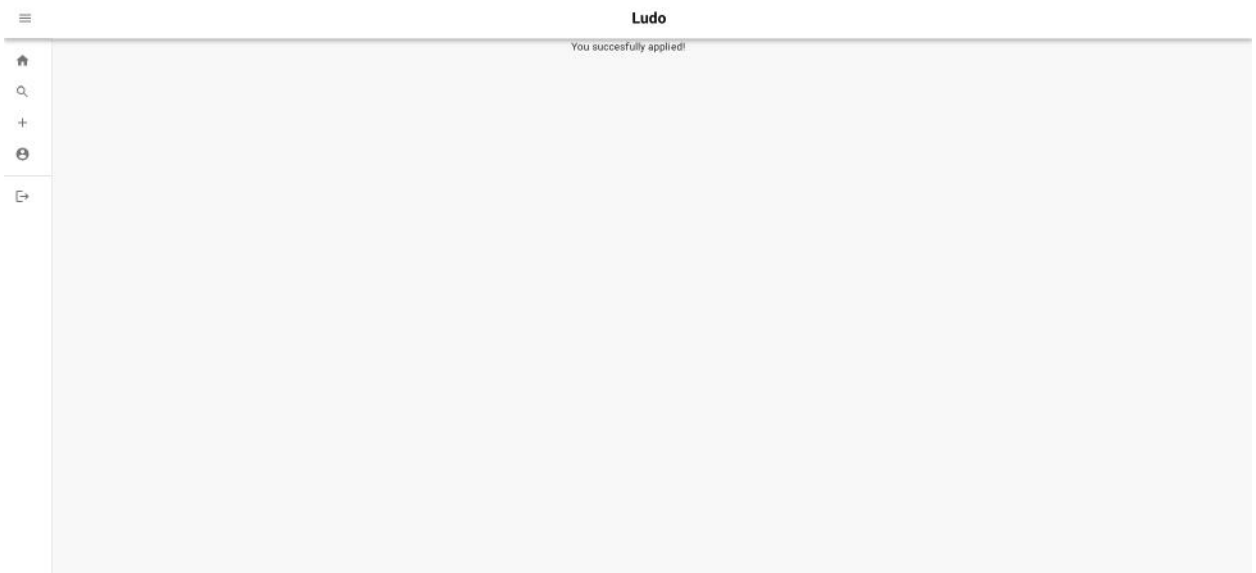
CREATION EQUIPMENT POST



DETAIL EQUIPMENT POST PAGE



APPLY EVENT



Final release notes

Even though backend implemented create event and equipment post functions, in the frontend they are not connected to backend but create event is connected to backend in android application. Frontend works on mock data for post creations. Also, spectate to event, delete event and equipment posts, change information of event and equipment post, event analytics functionalities were also implemented in the backend but they were not implemented in the

frontend and mobile. We didn't implement notification and change profile information functionalities.

Wiki data

We created our badges using keywords from wiki data. We created 6 badges with these names: enthusiastic (wiki data id: Q107261265), friendly (wiki data id: Q491), leader (wiki data id: Q1251441), gifted (wiki data id: Q467677), loser (wiki data id: Q20861252), competitive (wiki data id: Q107289411). These badges are used when a user wants to send a badge to the other participants of the event that he/she is participating in.

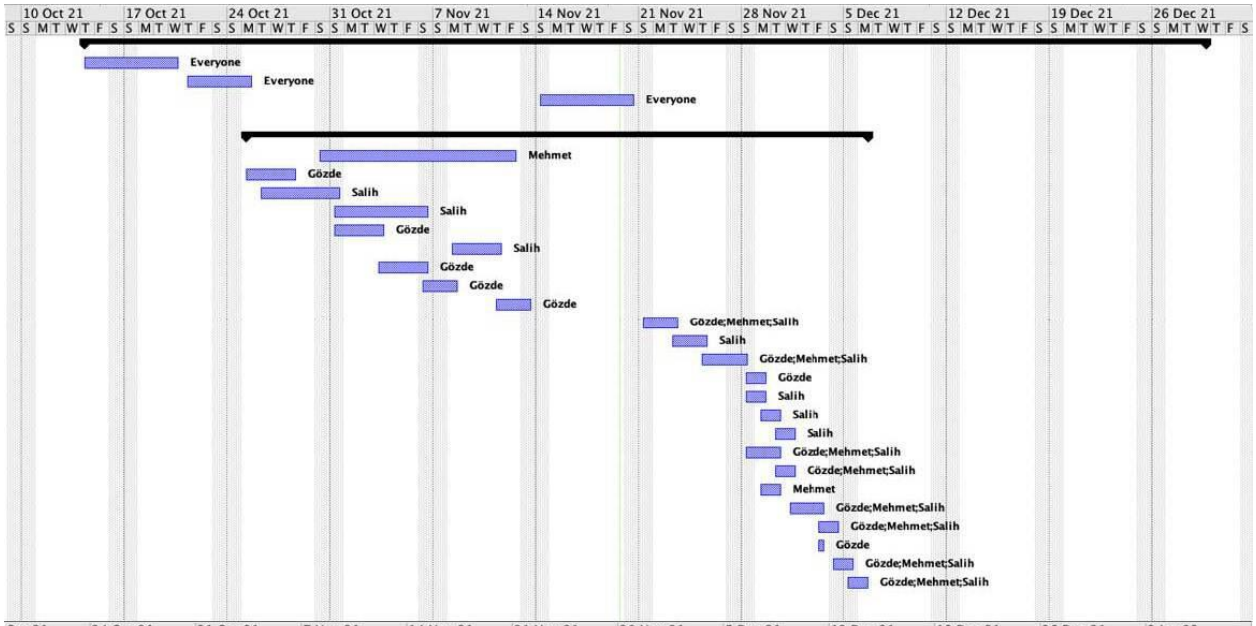
W3C Activity Streams 2.0 standard

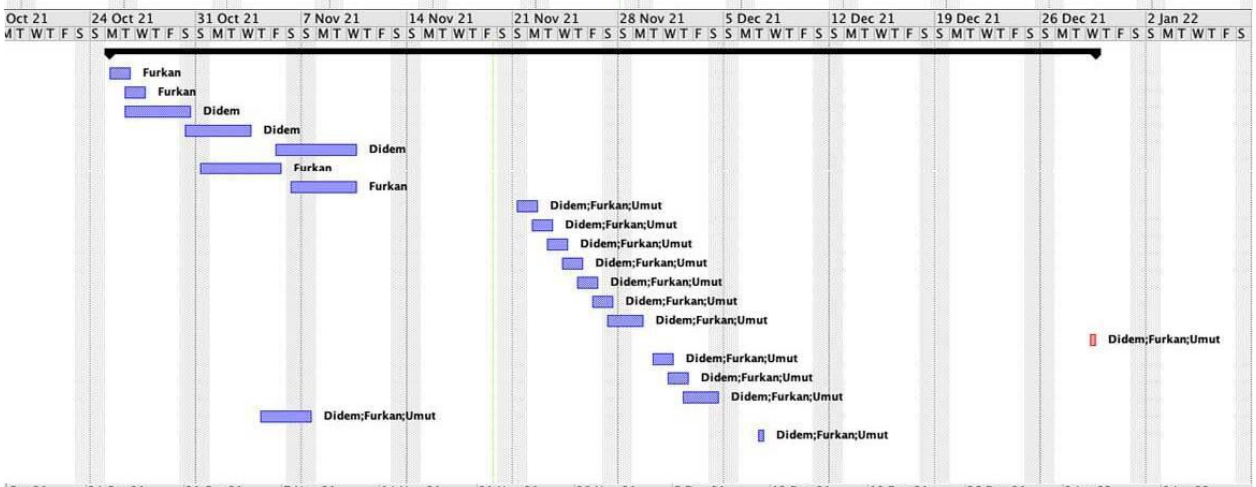
We examined the activity stream structıre and applied its json convention to the requests that are covered by the activity stream. We wrote down input and output jsons of these functions. Those functions are create event post, create equipment post, delete event post (only input json is in activity stream format), delete equipment post, delete event post, view event post details, view equipment post details, change event post information, change equipment post information, get event post analytics, apply to event, spectate to event, accept application, reject application, postpone event, create event comment, create equipment comment functions. For these functions, frontend and mobile sends requests according to the activity stream conventions. Backend retrieves these jsons, processes them and also stores them in EventPostActivityStream table, EquipmentPostActivityStream table, EventCommentActivityStream table and Equipment CommentActivityStream table. Before saving the activity stream information into the database we save the actual "object", in our case this is an event or equipment post. Then we save the activity stream information into the database referring to the object that we have just created/deleted/updated as a foreign key. In all 4 of these activity stream tables there are generally 5 fields: object, context, summary, actor and type. This information comes from jsons that are created according to the w3c activity stream convention. "Object" field refers to the comment, event or equipment object and in the models they are foreign keys to the event, equipment or comment objects. "Context" field refers to the w3c activity stream documentation url and this is always this url "https://www.w3.org/ns/activitystreams", "summary" is the summary of the activity explaining what the user has just done, "actor" is referring to the user object who made this activity and actor is stored as a foreign key to this user object and "type" is the type of the activity such as create, delete or update.
The communication between the backend and frontend/mobile is done with activity stream jsons for these activities. Frontend and mobile send requests according to this convention and backend returns the information according to this convention. But we also modified a little these jsons in this way: When the user creates an event and the information must be sent to the backend, w3c activity stream sends very little information, instead of that we list all the information of the post in the "object" part of the json. Similarly, we return all the information of the object in the "object" part of the json. For the update case, in addition to context, type, summary, object

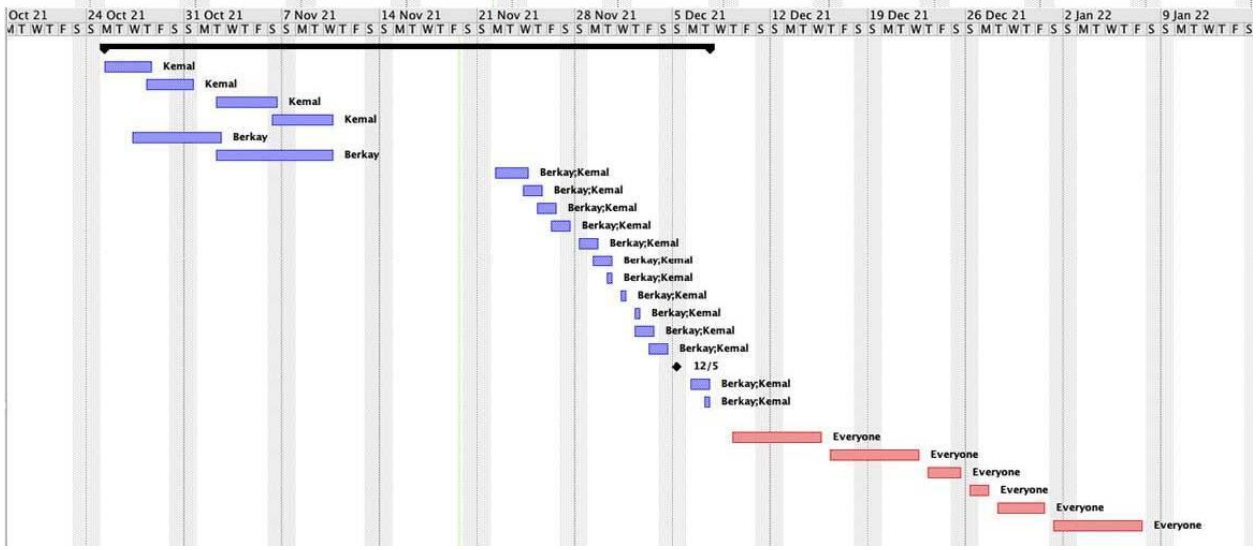
keywords we added another keyword in the json that mobile and frontend sends to backend, “modifications”. The parts where the user changed the information are sent only here so the backend updates the object using only the information in the “modifications” keyword. As the output jsons, backend sends all the information about the object in the “object” keyword of the activity stream json structure, for the delete case, only success message is returned to the frontend and mobile because they only check the status code if everything is okay and don’t display anything.

Project Plan

		Name	Duration	Start	Finish	Predecessors	Resource Names
86		**CMPE451**	55 days?	**10/14/21 8:00 AM**	**12/29/21 5:00 PM**		
87		Review CMPE352	7 days?	10/14/21 8:00 AM	10/20/21 5:00 PM		Everyone
88		Deciding to technology (for database, backend, frontend and anc	5 days?	10/21/21 8:00 AM	10/25/21 5:00 PM		Everyone
89		Milestone 1	7 days?	11/14/21 8:00 AM	11/20/21 5:00 PM		Everyone
90		**Backend**	31 days?	**10/25/21 8:00 AM**	**12/6/21 5:00 PM**		
91		login, logout, sign up, forgot password	14 days?	10/30/21 8:00 AM	11/12/21 5:00 PM		Mehmet
92		event creation	4 days?	10/25/21 8:00 AM	10/28/21 5:00 PM		Gözde
93		postpone	6 days?	10/26/21 8:00 AM	10/31/21 5:00 PM		Salih
94		delete event post	7 days?	10/31/21 8:00 AM	11/6/21 5:00 PM		Salih
95		update event post information	4 days?	10/31/21 8:00 AM	11/3/21 5:00 PM		Gözde
96		event participants (participant selection)	4 days?	11/8/21 8:00 AM	11/11/21 5:00 PM		Salih
97		equipment post creation	4 days?	11/3/21 8:00 AM	11/6/21 5:00 PM		Gözde
98		delete equipment post	3 days?	11/6/21 8:00 AM	11/8/21 5:00 PM		Gözde
99		update equipment post information	3 days?	11/11/21 8:00 AM	11/13/21 5:00 PM		Gözde
100		search, filter, sort posts	2 days?	11/21/21 8:00 AM	11/23/21 5:00 PM		Gözde;Mehmet;Salih
101		spectate	3 days?	11/23/21 8:00 AM	11/25/21 5:00 PM		Salih
102		write event post comment	2 days?	11/25/21 8:00 AM	11/28/21 11:00 AM		Gözde;Mehmet;Salih
103		view event details	2 days?	11/28/21 8:00 AM	11/29/21 5:00 PM		Gözde
104		participant analytics	2 days?	11/28/21 8:00 AM	11/29/21 5:00 PM		Salih
105		aply to event	2 days?	11/29/21 8:00 AM	11/30/21 5:00 PM		Salih
106		event analytics	2 days?	11/30/21 8:00 AM	12/1/21 5:00 PM		Salih
107		homepage	2 days?	11/28/21 8:00 AM	11/30/21 5:00 PM		Gözde;Mehmet;Salih
108		event search	2 days?	11/30/21 8:00 AM	12/1/21 5:00 PM		Gözde;Mehmet;Salih
109		update user profile information	2 days?	11/29/21 8:00 AM	11/30/21 5:00 PM		Mehmet
110		equipment search	3 days?	12/1/21 8:00 AM	12/3/21 5:00 PM		Gözde;Mehmet;Salih
111		write equipment comment	1 day?	12/3/21 8:00 AM	12/4/21 5:00 PM		Gözde;Mehmet;Salih
112		view equipment post detail	1 day?	12/3/21 8:00 AM	12/3/21 5:00 PM		Gözde
113		send badge	2 days?	12/4/21 8:00 AM	12/5/21 5:00 PM		Gözde;Mehmet;Salih
114		send notification	2 days?	12/5/21 8:00 AM	12/6/21 5:00 PM		Gözde;Mehmet;Salih
115		**Frontend**	48 days?	**10/25/21 8:00 AM**	**12/29/21 5:00 PM**		
116		header	2 days?	10/25/21 8:00 AM	10/26/21 5:00 PM		Furkan
117		sidebar	2 days?	10/26/21 8:00 AM	10/27/21 5:00 PM		Furkan
118		login	5 days?	10/26/21 8:00 AM	10/30/21 5:00 PM		Didem
119		sign-up	5 days?	10/30/21 8:00 AM	11/3/21 5:00 PM		Didem
120		forget password	6 days?	11/5/21 8:00 AM	11/10/21 5:00 PM		Didem
121		search page (event&equipment)	6 days?	10/31/21 8:00 AM	11/5/21 5:00 PM		Furkan
122		search filter	5 days?	11/6/21 8:00 AM	11/10/21 5:00 PM		Furkan
123		event page	1 day?	11/21/21 8:00 AM	11/22/21 5:00 PM		Didem;Furkan;Umut
124		event creation	2 days?	11/22/21 8:00 AM	11/23/21 5:00 PM		Didem;Furkan;Umut
125		event edit	2 days?	11/23/21 8:00 AM	11/24/21 5:00 PM		Didem;Furkan;Umut
126		event participants	2 days?	11/24/21 8:00 AM	11/25/21 5:00 PM		Didem;Furkan;Umut
127		participant analytics	2 days?	11/25/21 8:00 AM	11/26/21 5:00 PM		Didem;Furkan;Umut
128		user profile	1 day?	11/26/21 8:00 AM	11/27/21 5:00 PM		Didem;Furkan;Umut
129		user profile edit	1 day?	11/27/21 8:00 AM	11/29/21 5:00 PM		Didem;Furkan;Umut
130		comments (event&equipment)	1 day?	12/29/21 8:00 AM	12/29/21 5:00 PM		Didem;Furkan;Umut
131		equipment post page	2 days?	11/30/21 8:00 AM	12/1/21 5:00 PM		Didem;Furkan;Umut
132		equipment post creation	2 days?	12/1/21 8:00 AM	12/2/21 5:00 PM		Didem;Furkan;Umut
133		equipment post edit	2 days?	12/2/21 8:00 AM	12/4/21 5:00 PM		Didem;Furkan;Umut
134		homepage	2 days?	11/4/21 8:00 AM	11/7/21 5:00 PM		Didem;Furkan;Umut
135		static pages (about us&contact)	1 day?	12/7/21 8:00 AM	12/7/21 5:00 PM		Didem;Furkan;Umut
136		**Mobile**	32 days?	**10/25/21 8:00 AM**	**12/7/21 5:00 PM**		
137		Welcome Page	4 days?	10/25/21 8:00 AM	10/28/21 5:00 PM		Kemal
138		Login Page	4 days?	10/28/21 8:00 AM	10/31/21 5:00 PM		Kemal
139		Sign-up Page	5 days?	11/2/21 8:00 AM	11/6/21 5:00 PM		Kemal
140		Forgot Password Page	5 days?	11/6/21 8:00 AM	11/10/21 5:00 PM		Kemal
141		Favorite Sports Page	7 days?	10/27/21 8:00 AM	11/2/21 5:00 PM		Berkay
142		Email Confirm Page	9 days?	11/2/21 8:00 AM	11/10/21 5:00 PM		Berkay
143		search filter	3 days?	11/22/21 8:00 AM	11/24/21 5:00 PM		Berkay;Kemal
144		event page	2 days?	11/24/21 8:00 AM	11/25/21 5:00 PM		Berkay;Kemal
145		event creation	2 days?	11/25/21 8:00 AM	11/26/21 5:00 PM		Berkay;Kemal
146		event edit	1 day?	11/26/21 8:00 AM	11/27/21 5:00 PM		Berkay;Kemal
147		event participants	1 day?	11/28/21 8:00 AM	11/29/21 5:00 PM		Berkay;Kemal
148		participant analytics	2 days?	11/29/21 8:00 AM	11/30/21 5:00 PM		Berkay;Kemal
149		user profile	1 day?	11/30/21 8:00 AM	11/30/21 5:00 PM		Berkay;Kemal
150		user profile edit	1 day?	12/1/21 8:00 AM	12/1/21 5:00 PM		Berkay;Kemal
151		comments (event&equipment)	1 day?	12/2/21 8:00 AM	12/2/21 5:00 PM		Berkay;Kemal
152		equipment post page	2 days?	12/2/21 8:00 AM	12/3/21 5:00 PM		Berkay;Kemal
153		equipment post creation	1 day?	12/3/21 8:00 AM	12/4/21 5:00 PM		Berkay;Kemal
154		equipment post edit	0 days?	12/5/21 8:00 AM	12/5/21 5:00 PM		Berkay;Kemal
155		homepage	2 days?	12/6/21 8:00 AM	12/7/21 5:00 PM		Berkay;Kemal
156		static pages (about us&contact)	1 day?	12/7/21 8:00 AM	12/7/21 5:00 PM		Berkay;Kemal
157		Milestone 2	7 days?	12/8/21 5:00 PM	12/15/21 5:00 PM		Everyone
158		Review project	7 days?	12/15/21 5:00 PM	12/22/21 5:00 PM		Everyone
159		Fix Bugs	3 days?	12/22/21 5:00 PM	12/25/21 5:00 PM		Everyone
160		Debugging	2 days?	12/25/21 5:00 PM	12/27/21 5:00 PM		Everyone
161		Testing	4 days?	12/27/21 5:00 PM	12/31/21 5:00 PM		Everyone
162		Final Milestone	7 days?	1/1/22 8:00 AM	1/7/22 5:00 PM		Everyone


10 Oct 21
17 Oct 21
24 Oct 21
31 Oct 21
7 Nov 21
14 Nov 21
21 Nov 21
28 Nov 21
5 Dec 21
12 Dec 21
19 Dec 21
26 Dec 21
Everyone
Everyone
Everyone
Mehmet
Gözde
Salih
Salih
Gözde
Salih
Gözde
Gözde
Gözde
Gözde;Mehmet;Salih
Salih
Gözde;Mehmet;Salih
Gözde
Salih
Salih
Salih
Gözde;Mehmet;Salih
Gözde;Mehmet;Salih
Mehmet
Gözde;Mehmet;Salih
Gözde;Mehmet;Salih
Gözde
Gözde;Mehmet;Salih
Gözde;Mehmet;Salih


Oct 21
24 Oct 21
31 Oct 21
7 Nov 21
14 Nov 21
21 Nov 21
28 Nov 21
5 Dec 21
12 Dec 21
19 Dec 21
26 Dec 21
2 Jan 22
Furkan
Furkan
Didem
Didem
Didem
Furkan
Furkan
Didem;Furkan;Umut
Didem;Furkan;Umut
Didem;Furkan;Umut
Didem;Furkan;Umut
Didem;Furkan;Umut
Didem;Furkan;Umut
Didem;Furkan;Umut
Didem;Furkan;Umut
Didem;Furkan;Umut
Didem;Furkan;Umut
Didem;Furkan;Umut
Didem;Furkan;Umut
Didem;Furkan;Umut


Oct 21
24 Oct 21
31 Oct 21
7 Nov 21
14 Nov 21
21 Nov 21
28 Nov 21
5 Dec 21
12 Dec 21
19 Dec 21
26 Dec 21
2 Jan 22
9 Jan 22
Kemal
Kemal
Kemal
Kemal
Berkay
Berkay
Berkay;Kemal
Berkay;Kemal
Berkay;Kemal
Berkay;Kemal
Berkay;Kemal
Berkay;Kemal
Berkay;Kemal
Berkay;Kemal
Berkay;Kemal
Berkay;Kemal
12/5
Berkay;Kemal
Berkay;Kemal
Everyone
Everyone
Everyone
Everyone
Everyone
Everyone

Glossary

- Subscription: Users who sign up and login to app
- Applicants to an event: Users who want to be participants to an event as players thus they apply to the event post. They may or may not be chosen as participants to the event by the event creator.
- Event Creator: User that creates sports events on the application so that users can apply to that event.
- Event: An amateur sport incident, created on the platform by an event-creator that has a specified time&location
- Spectator: Users that don't play in events but like to participate as a spectator and follow events, receive notifications and recommendations from the system.
- Badge: An emblem received by users from other users, system or event creator.
- Notification: Remarks/messages about the application that users receive automatically. They can happen when the user is both in or out of the application.
- Recommendation: User is recommended about events, users and sports equipments by the system according to the information that the user provided. Recommendations are made to users according to their interests.
- Blocking: When a user blocks another user, they don't see the posts or the events that the other user participates or creates.
- Report: Users are able to report posts or other users when they post or write inappropriately. System administrators may punish these accounts.
- Follow: A user can follow another users to be informed of the events they create.
- Encrypted: Encryption is the process of encoding information. This process converts the original representation of the information, known as plaintext, into an alternative form known as ciphertext.
- SSL: is Secure Sockets Layer which is a protocol for establishing authenticated and encrypted links between networked computers.
- Direct message: Private message channel or private communication channel between users.
- GDPR: is General Data Protection Regulation which is a regulation in EU law on data protection and privacy in the European Union and the European Economic Area.
- KVKK: The Turkish Data Protection Authority is a government organization in Turkey which provides the protection of personal data and works.
- SHA: is Secure Hash Algorithm which is a cryptographic hash function which takes an input and produces a 160-bit (20-byte) hash value.

- W3C Activity Stream: A paradigm to display recent activity in a context (ex: UI paradigm)
- Profile page: A page accessible to the public, which provides personal information about the user.

1.Functional Requirements

1.1.User Requirements

1.1.1 User profile

- *1.1.1.1* **Users shall have a profile page.**
- *1.1.1.2* **Users shall be able to make selected information private or public. These information are all information other than username.**
- *1.1.1.3* **Users shall be able to add profile picture.**
- *1.1.1.4* **There is a default profile picture.**
- *1.1.1.5* **Users shall be able to show or hide badges that other users sent to them.**
- *1.1.1.6* **Users will be able to show the sports they are interested in.**
- *1.1.1.7* **Users shall be able to add/modify their skill levels in sports that they are interested in.**
- *1.1.1.8* **Users shall be able to show the events they previously joined.**
- *1.1.1.9* **Users who created an event before shall be able to show the events they previously created.**
- *1.1.1.10* **Users shall be able to change the informations given at sign up.**
- *1.1.1.11* **Upcoming events the user created or will participate shall be shown on the profile page.**
- **1.1.1.12 Users shall be able to see the badges they have received**

1.1.2 Users

- *1.1.2.1* **User shall sign up to the system.**
- **1.1.2.2 All users including the user that created the event shall be able to participate in events.**
- **1.1.2.3 Users shall receive badges.**

    - **1.1.2.3.1 Users that participate in events shall be able to receive badges from event creators.**

    - **1.1.2.3.2 Users shall be able to receive badges from the system after participating number of events.**
    - **1.1.2.3.3 Users shall be able to receive badges from other users that participated in the same event.**
- **1.1.2.4 Users shall be able to select badges from the options that system provides.**
- *1.1.2.5* **Users shall discover potential sport events of interest.**

    - *1.1.2.5.1* **This discovery shall be done by stating the name of the event, the type of sports, time of the event and location that are specified by the user.**
    - *1.1.2.5.2* **Users shall be able to do such discoveries with the information that is stated in 1.1.2.5.1 directly by searching the keywords and/or indirectly by the system recommendations.**
- *1.1.2.6* **Users shall be able to discover other users. The user can view the participant list of the event that they participate.**
- **1.1.2.7 Users shall be able to search sports equipments.**
- **1.1.2.8 Users shall be able to post the link to a sports equipment which is useful/ has a good price etc. so that other users may benefit from them.**
- *1.1.2.9* **Users shall receive notifications about status of the join request to event and changes on the date of the event.**
- *1.1.2.10* **Users can turn notifications on/off.**
- *1.1.2.11* **Users shall be able to tag any item with semantic tags (from Wikidata).**
- *1.1.2.12* **Users shall be able to see the limit to the number of people that can participate in an event, name of the event, equipment required for an event, date and time, detailed location information, rules, skill requirement, status, badges that might be given at the end of the event and description of an event. Event creators can edit these information.**
- *1.1.2.13* **Users shall be able to communicate through posts.**
- *1.1.2.14* **Event creators shall choose among the applicants as participants of the event. Event creator shall be free to choose between the candidates.**
- *1.1.2.15* **When an event is being created, event creator shall choose among the types of sports, if there is no such event recognized by the system, event creator shall describe the type of the event in terms of its characteristics.**
- **1.1.2.16 Event creators shall be able to see the analytics of the event they created: The average number of events that the accepted participants of this event have been accepted to.**
- **1.1.2.17 Users shall be able to block other users.**

- **1.1.2.19 Users shall be able to follow each other.**
- **1.1.2.20 Post creator shall be able to delete comments of the post he/she opened.**
- **1.1.2.21 Users can create an event and an equipment post.**
- **1.1.2.22 Users can update the information of an event and an equipment post.**
- **1.1.2.23 Users can delete an event and an equipment post.**
- **1.1.2.24 Users can apply to events but their applications are either accepted or rejected by the event creator.**
- **1.1.2.25 Users can apply to events as spectators.**

1.1.3 Sign Up

- *1.1.3.1* **Users shall input their name, surname, email, username and password.**
- *1.1.3.2* **Users shall accept the terms & conditions.**
- *1.1.3.3* **After successful sign up, users shall be able to provide detailed information that are age, gender, location, favorite sports on their profile page.**

1.1.4 Sign In

- *1.1.4.1* **Users shall be able to sign in with their username (or email) and password.**
- *1.1.4.2* **There shall be an option for forgotten passwords. User could change the password in such a case.**
- *1.1.4.3* **There shall be a “remember me” option on the sign in page.**

1.2.System Requirements

1.2.1 Authentication

1.2.1.1 Sign Up

- *1.2.1.1.1* **Sign up shall require the user to enter their e-mail address, name and surname.**
- *1.2.1.1.2* **E-mail shall be verified via a code.**
- *1.2.1.1.3* **After successful sign up, the system shall go on the next step that the user specifies a password, a username and agrees the terms & conditions contracts.**

- *1.2.1.1.4* **The username shall be at least 4 characters long and containing only alphanumeric characters.**
- *1.2.1.1.5* **The username shall be non-taken**
- *1.2.1.1.6* **Password shall be at least 6 characters long, containing at least one number and one letter and containing only alphanumeric characters.**
- *1.2.1.1.7* **Password shall be asked twice**
- *1.2.1.1.8* **After registration steps are completed, users shall enter multiple kinds of sports they are interested in and their skill levels in those sports.**

1.2.1.2 Sign In

- *1.2.1.2.1* **Sign In shall require a e-mail or username and password.**
- *1.2.1.2.2* **System shall block users who entered password wrong 3 consecutive times and shall ask the user to reset their password using their verified e-mail address.**
- *1.2.1.2.3* **System shall provide reset password functionality using the verified e-mail address, and shall send a new randomly generated password to e-mail address.**

1.2.2 Sports Constraints

- *1.2.2.1* **The system shall support different kinds of amateur spots.**
- *1.2.2.2* **Each sport shall have some specifications, which are:**

    - *1.2.2.2.1* **number of people who can participate (can be limitless or tentative)**
    - *1.2.2.2.2* **general rules**
    - *1.2.2.2.3* **equipment requirement**
    - *1.2.2.2.4* **special location requirement**

1.2.3 Event posts

- *1.2.3.1.1* **Events shall be discoverable by users that are of their interests and that are nearby.**
- *1.2.3.1.2* **Events shall be applicable by users that satisfy the skill set constraint.**
- *1.2.3.1.3* **Events shall be shown to the users via the search functionality.**
- *1.2.3.1.4* **The following information about the event shall be shown:**

    - *1.2.3.1.4.1* **event's date and time**
    - *1.2.3.1.4.2* **location**

        - *1.2.3.1.4.3* **sport type**
        - *1.2.3.1.4.4* **skill level requirement**
        - *1.2.3.1.4.5* **quota of the sport**
        - *1.2.3.1.4.6* **participant/spectator count**
        - *1.2.3.1.4.7* **status (upcoming, live or passed / full or open to application)**
        - *1.2.3.1.4.8* **name of the event**
        - *1.2.3.1.4.9* **comment section**
        - **1.2.3.1.4.10 required equipment(s) if any**
        - **1.2.3.1.4.11 rules**
        - **1.2.3.1.4.12 badges that might be given at the end of the event**
        - **1.2.3.1.4.13 description of the event**
    - *1.2.3.1.5* **Event shall be postponed if necessary participant count could not be satisfied.**
    - **1.2.3.1.6 There shall be section on the event posts where users can interact via commenting.**
    - **1.2.3.1.7 If an event is requested to be repeated for consequent weeks on the same day of the week and hour, while the event is being created there shall be an option for such repetition that a user might create multiples of the same event simultaneously.**
    - **1.2.3.1.8 If there is no default type in the system for an event that a user wants to create, there shall be an "other" option as the type of the sport.**

1.2.4 Sports equipment posts

- 1.2.4.1 In the sports equipment posts, there shall be a section to write the types of sports whose equipments are offered, description of the sports equipments and why they are suggested. These informations are required
- 1.2.4.2 In the posts about sports equipments, there shall be a place to put a link to a site where the sport equipment can be found. When the link is clicked, a new page to the site shall open. This information is optional.
- 1.2.4.3 In the posts about sports equipments, there shall be a place to specify the location of the shop if the shop is not only online and also has a store. This information is optional.
- 1.2.4.4 In the posts about sports equipments, there shall be comment section where users are able to ask questions about the post.

1.2.5 Search

1.2.5.1 Game/Event

- 1.2.5.1.1 The system shall allow users to be able to search for events according to a name, location or type of a sport.
- 1.2.5.1.2 Event search can be done by typing the keywords on a search bar
- 1.2.5.1.3 Event feed shall be filterable according to the specifications stated in date, location, game rate, incomplete number of participants in event.
- 1.2.5.1.4 Default feed sorting shall be changeable by the user to different options (such as but not limited to: newest event, nearest location, nearest date or increasing fee).
- 1.2.5.1.5 When user wants to filter the event posts near a location, he/she shall be able to change the radius of the location circle so that events in the circle are presented.

1.2.5.2 Sports equipments

- 1.2.5.2.1 On the page to discover user posts about sports equipments, posts shall be searched according to the type of sports, location information that is stated in the post.
- 1.2.5.2.3 Posts that don't have location information shall not be listed when user filters the results according to a location.
- 1.2.5.2.4 When user wants to filter the sport equipment posts near a location, he/she shall be able to change the radius of the location circle so that events in the circle are presented.

1.2.6 Recommendation

1.2.6.2 By Systems

- *1.2.6.2.1* **System shall recommend related events, other users, sports gears, tools, merchandise to users automatically according to a users' interests.**

1.2.7 Notifications

- *1.2.7.1* **People who are selected and not selected for an event shall be notified by the event creator.**
- *1.2.7.2* **Users shall be able to turn on/off the notifications**
- **1.2.7.3 When a user's request to participate in an event is accepted or rejected by the event creator, system shall send a notification to these users.**

1.2.8 Location Services

- *1.2.8.1* **Users shall be able to see sports events near a specified region by searching.**

2.Non-Functional Requirements

2.1. Security

- *2.1.1* **Passwords shall be encrypted.?**
- *2.1.2* **Users should be sent a verification mail when they want to change their passwords. Until the users click the link in the mail, their passwords should not be changed.?**
- *2.1.3* **The website interactions shall use SSL.**

2.2. Availability

- *2.2.1* **The system shall have a web application.**
- *2.2.2* **The system shall have an android application.**
- *2.2.3* **The system shall support English language.**

2.3. Privacy

- *2.3.1* **The system shall respect the user privacy laws.**
- *2.3.2* **System shall follow rules defined by GPDR and KVKK.**

2.4. Deployment

- *2.4.1* **Backend and frontend will be deployed to Amazon.**

2.5. Protocol

- *2.5.1* **The system shall use the W3C Activity Stream standard 2.0 standard. Subscriptions and notifications shall follow this standard.**
- *2.5.2* **The semantic taggings shall be supported with Wikidata.org.**

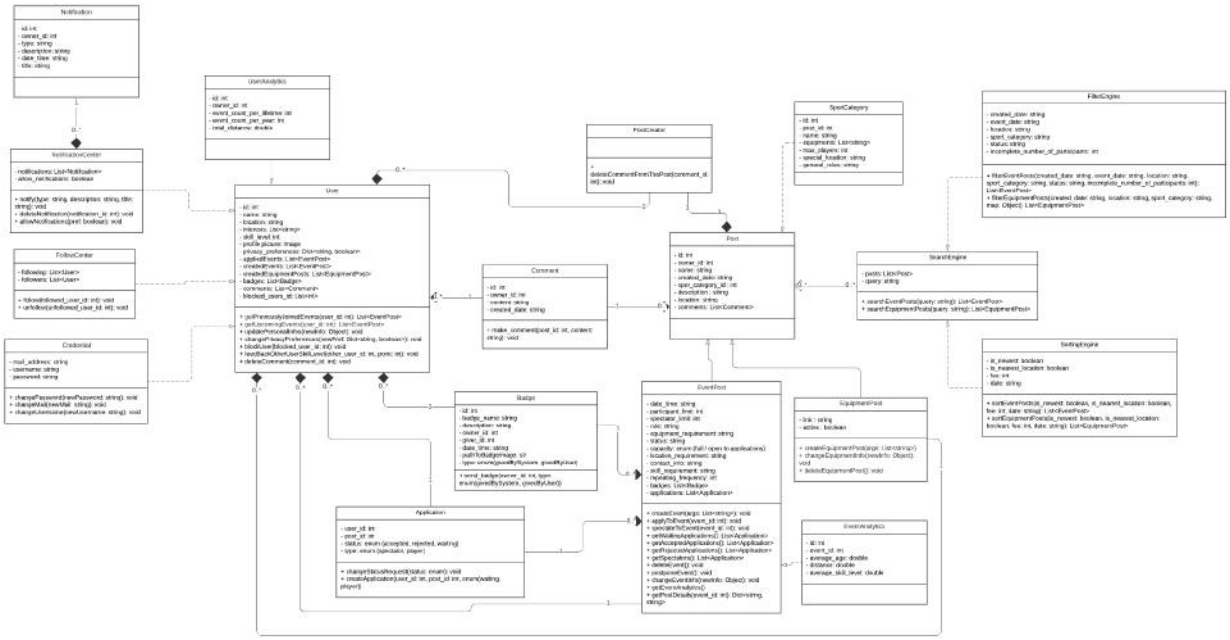
2.6 Language

- **For the backend, python will be used and django will be used as framework. For the frontend react will be used and for the mobile flutter will be used.**

Software design documents (UML diagrams, mockups)

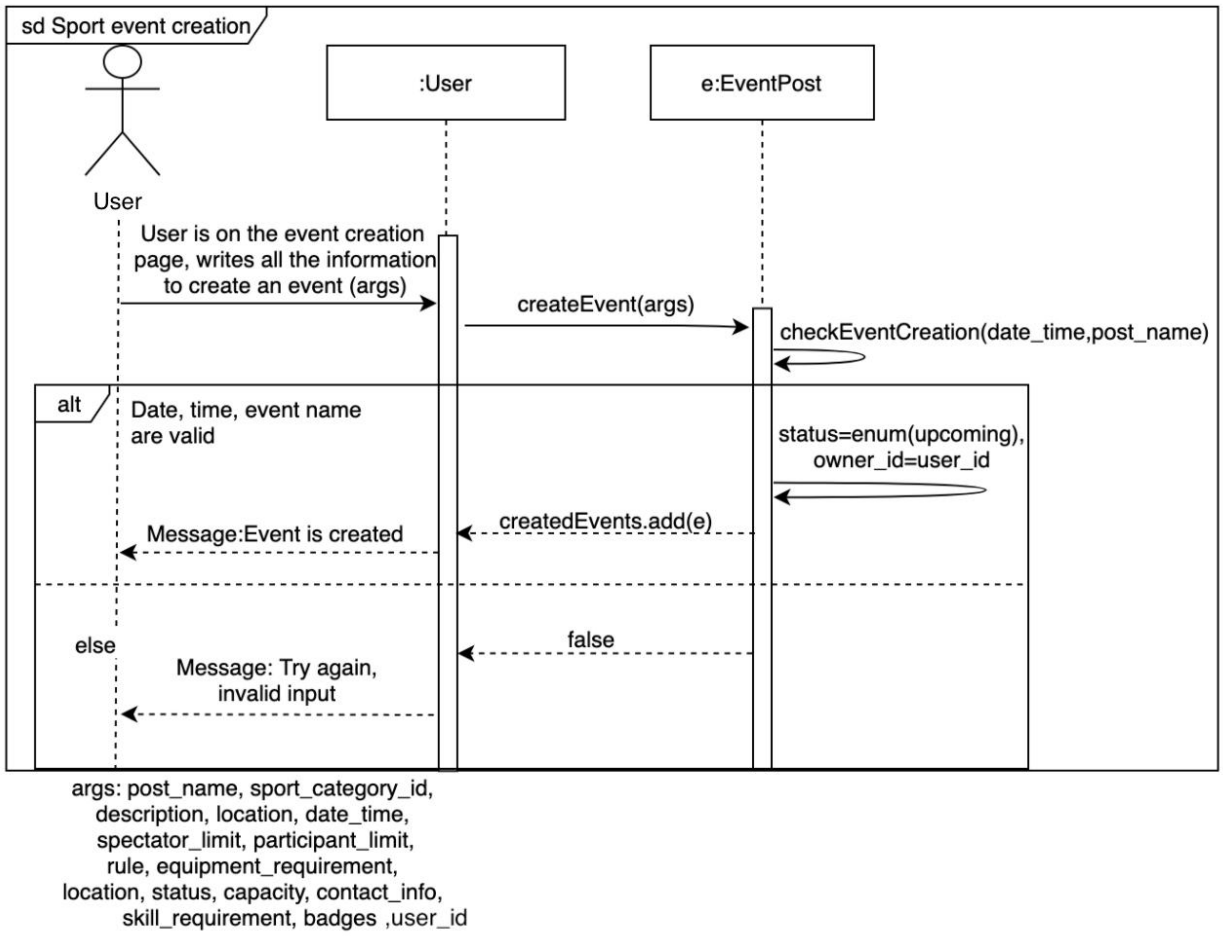
UML Diagrams

Class Diagram



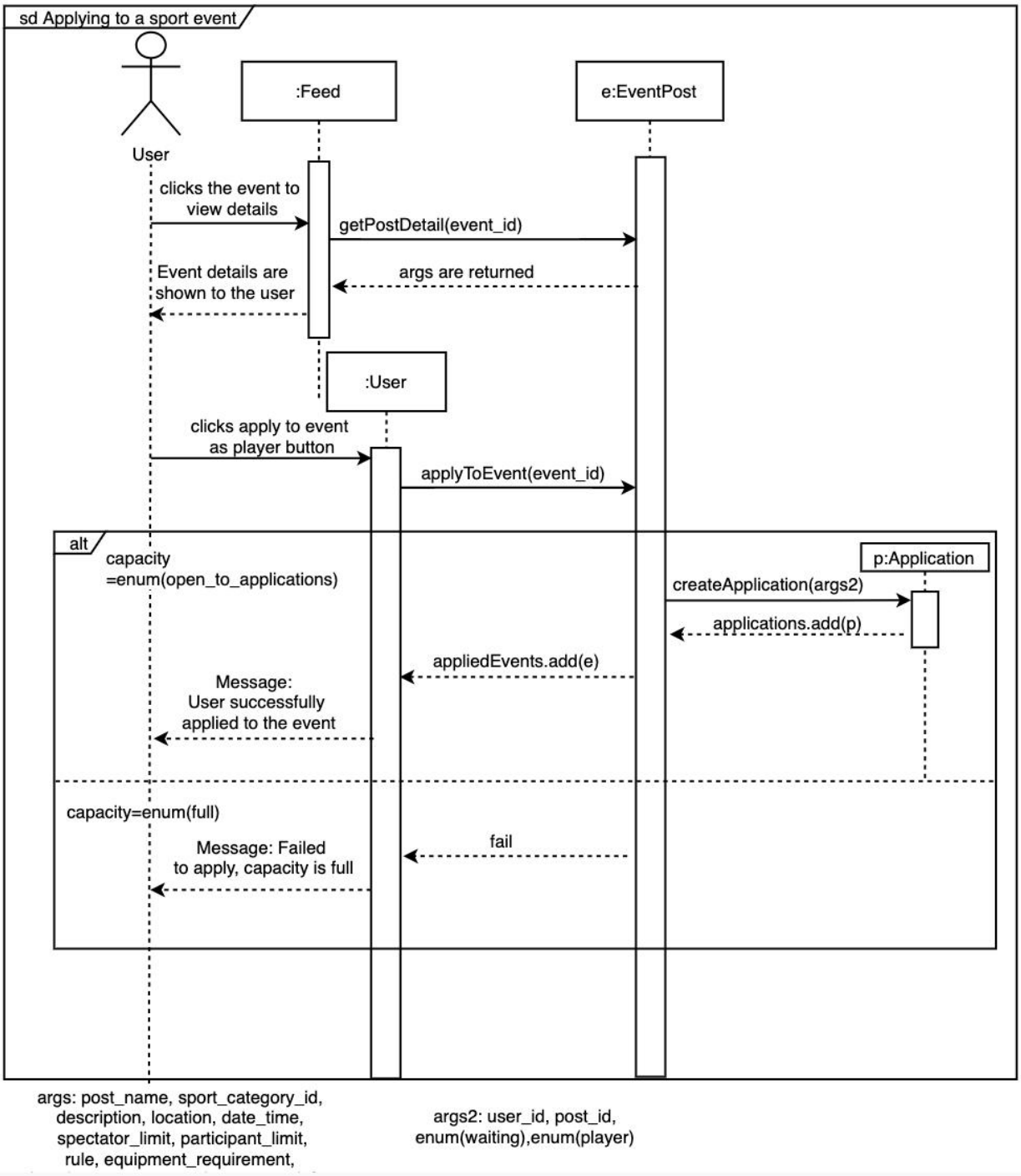
Sequence Diagrams

Sport event creation

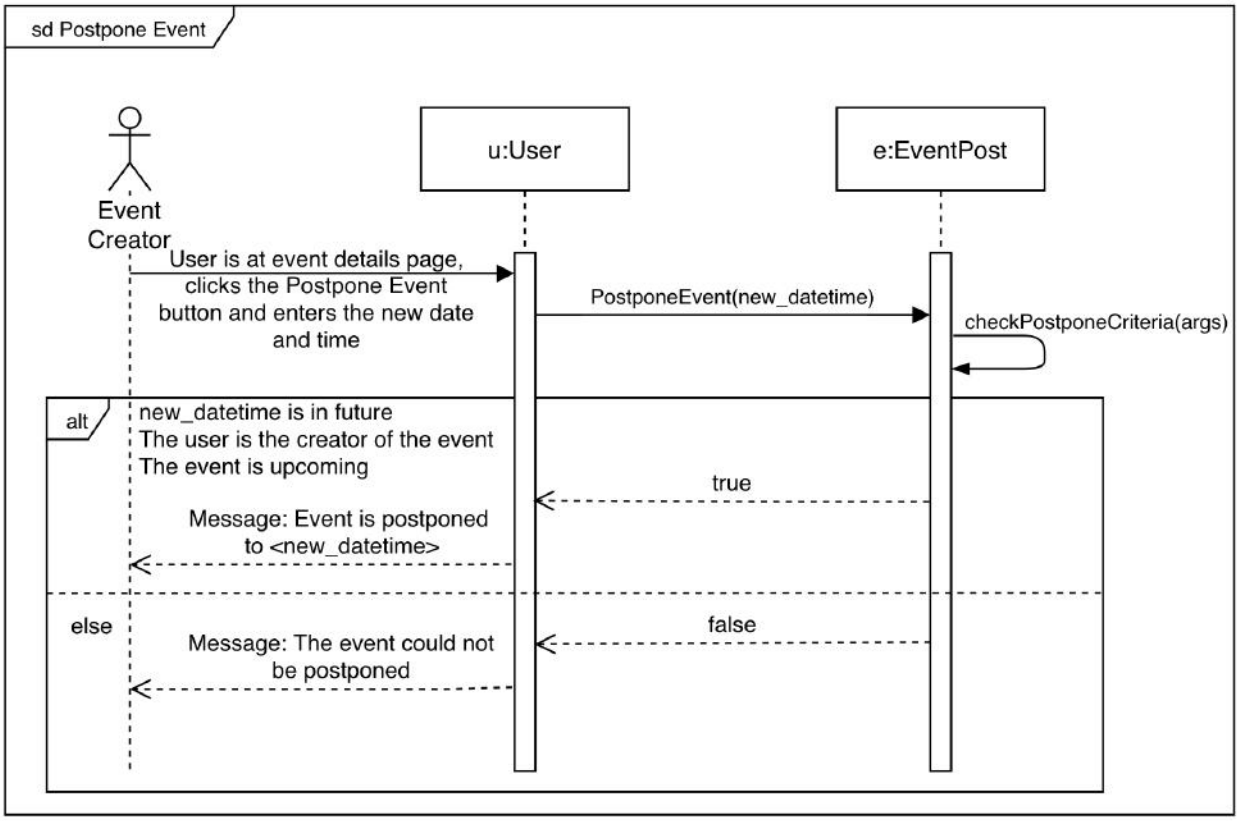


Apply to a sport event

User views the details of the event then clicks apply to event button and waits to be accepted as a participant.



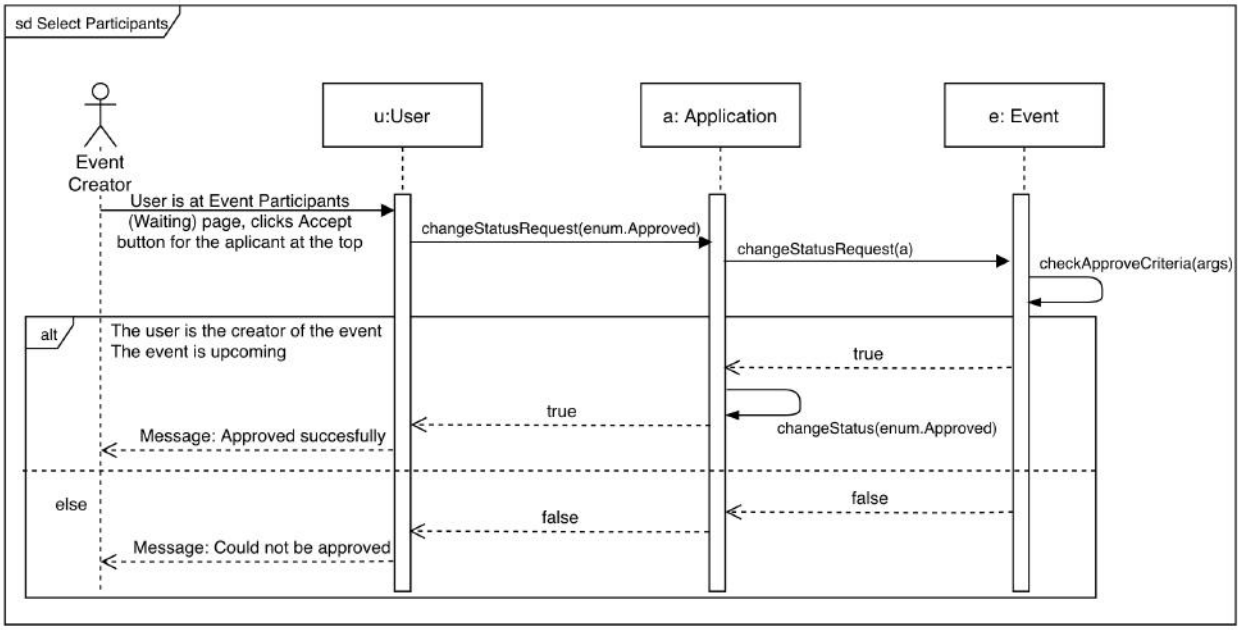
Postpone a sport event



args: event_status, new_datetime, event_date, event_time, event_creator_id, user_id

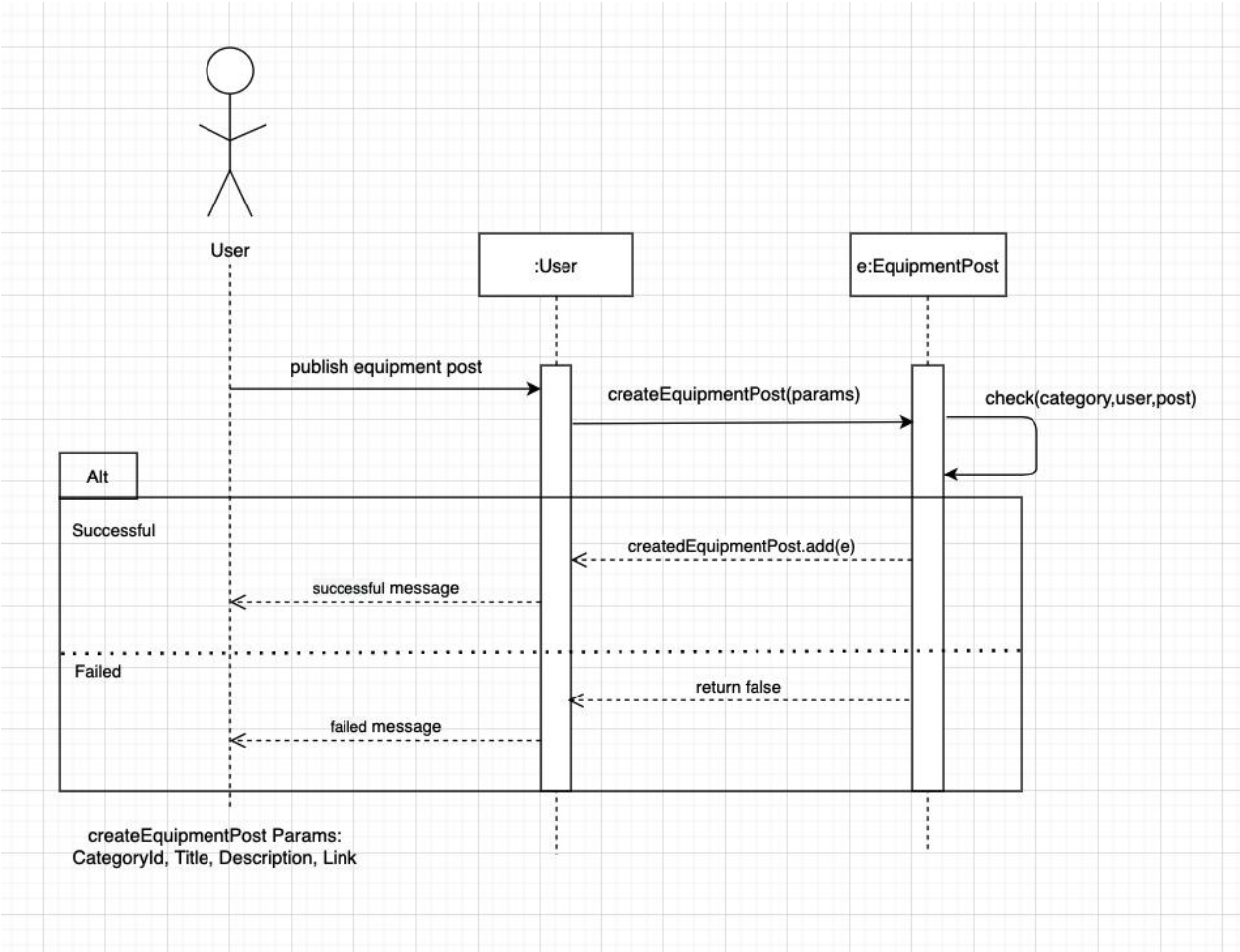
---

Approve a participant

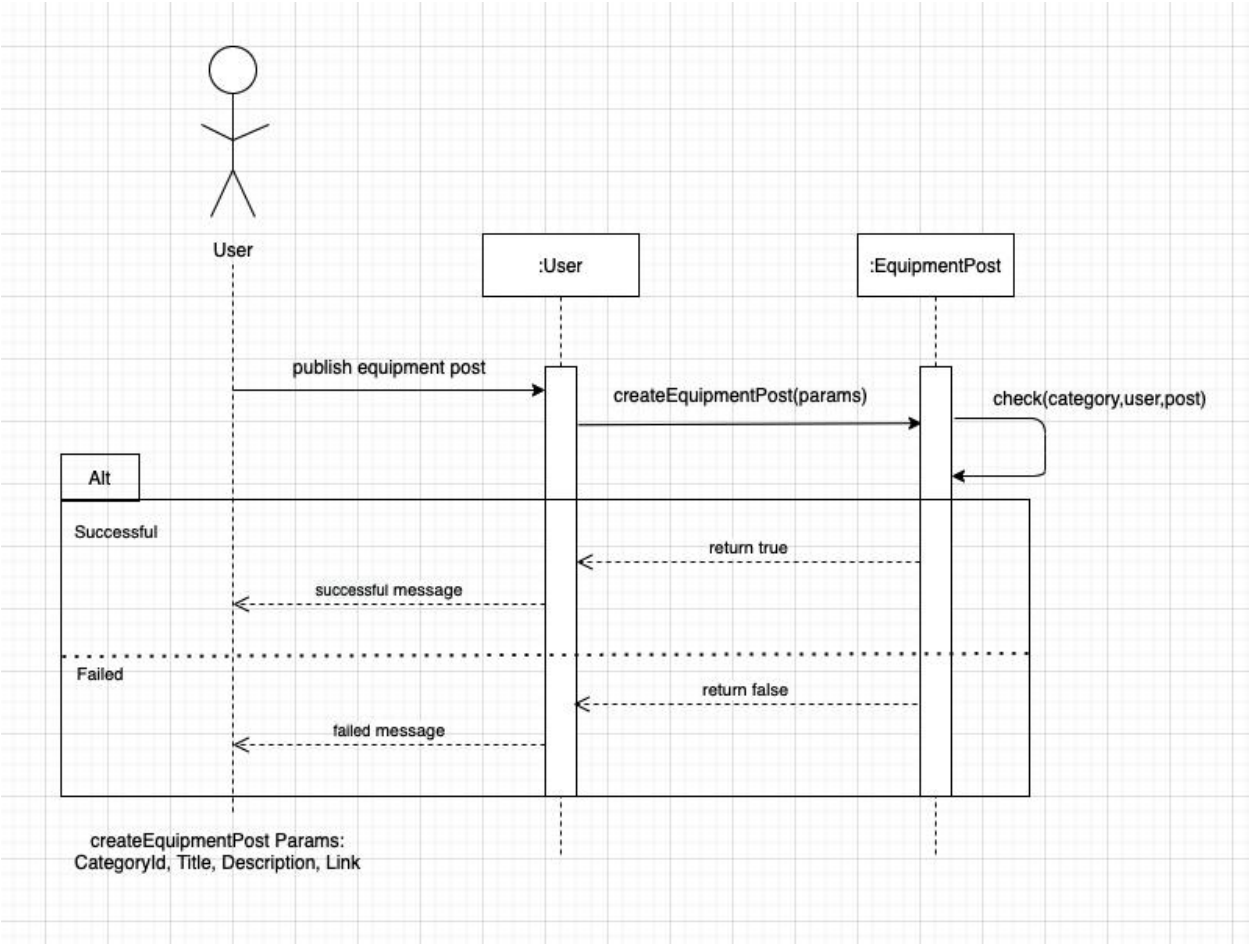


args: event_status, event_creator_id, user_id

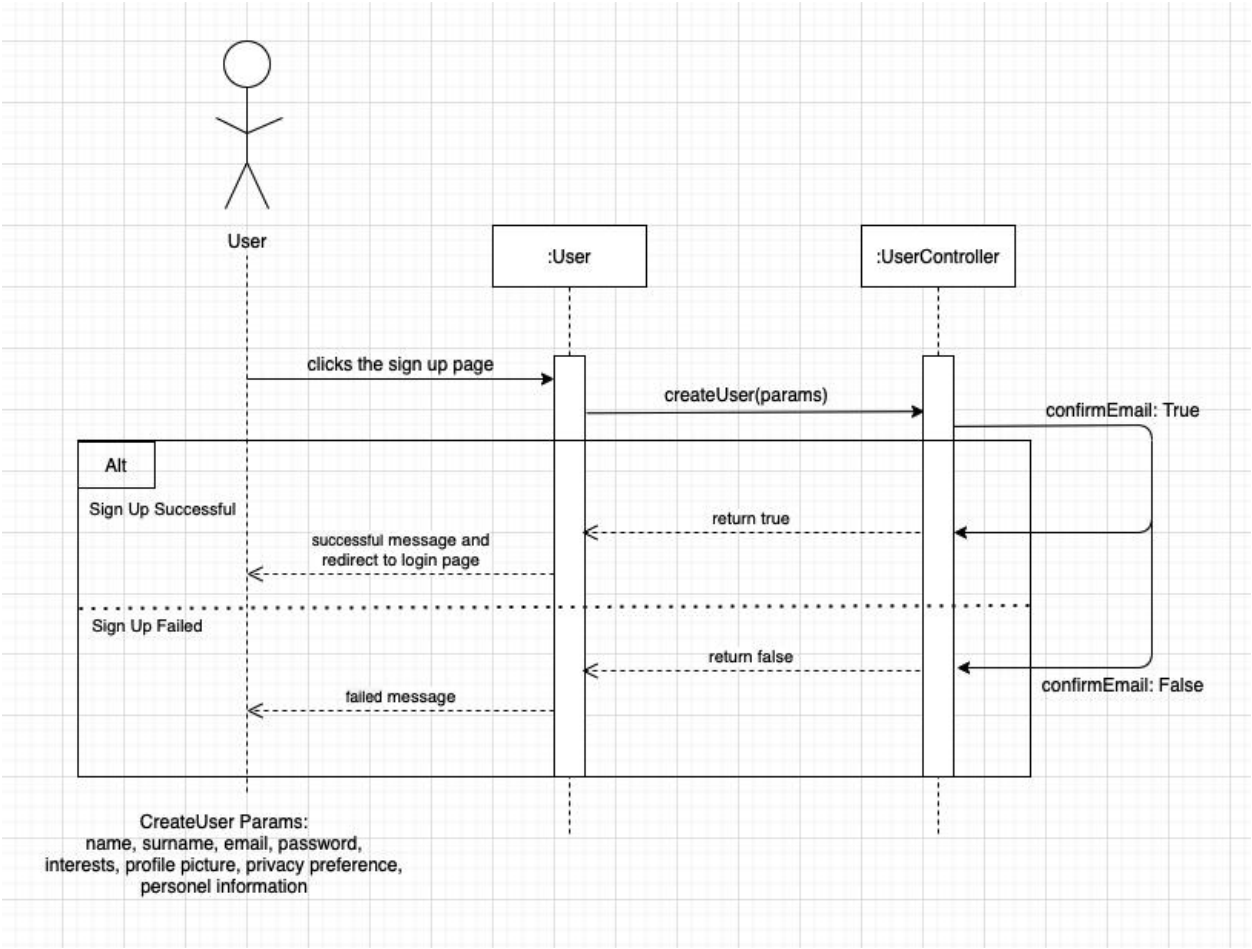
Equipment post creation



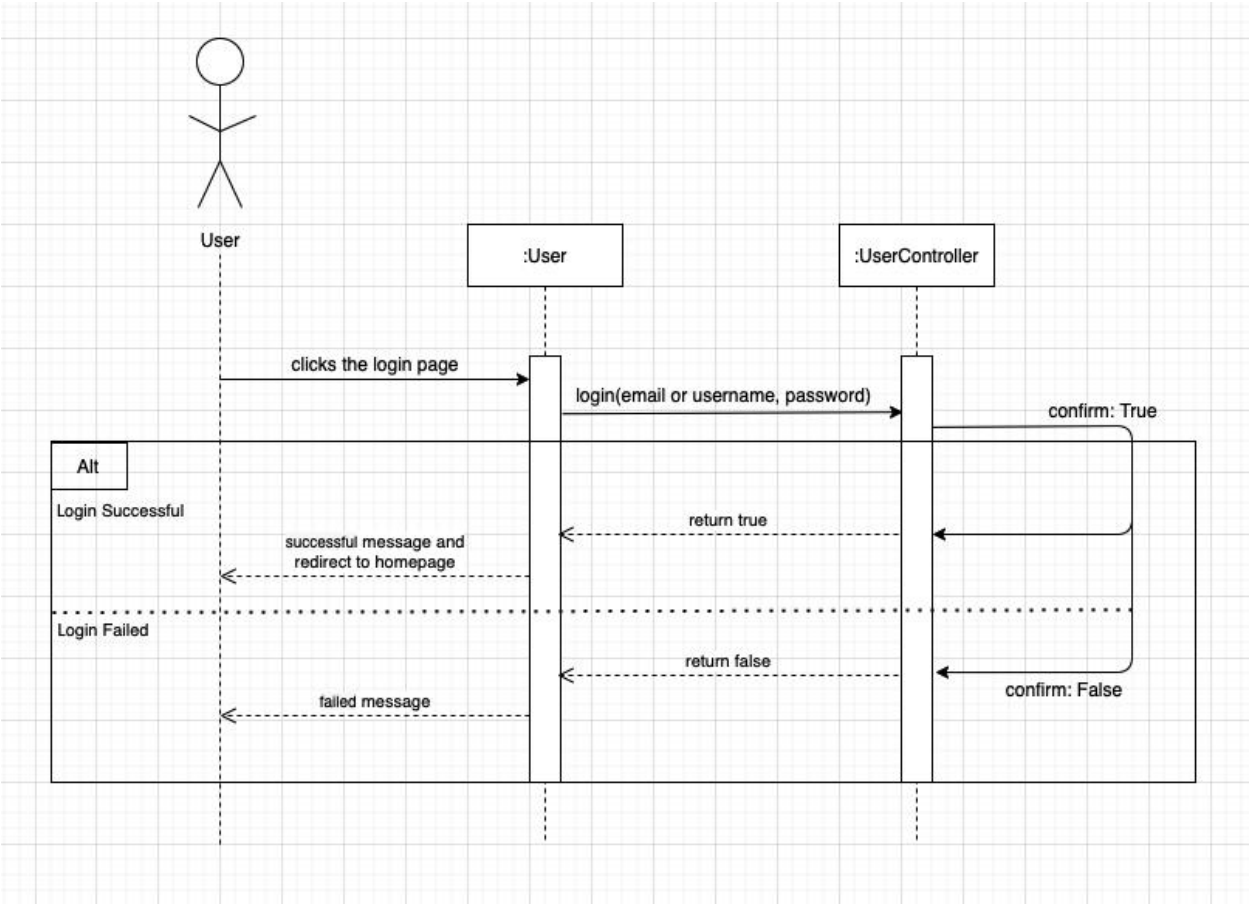
Equipment post comment


User
:User
:EquipmentPost
publish equipment post
createEquipmentPost(params)
check(category,user,post)
Alt
Successful
return true
successful message
Failed
return false
failed message
createEquipmentPost Params:
CategoryId, Title, Description, Link

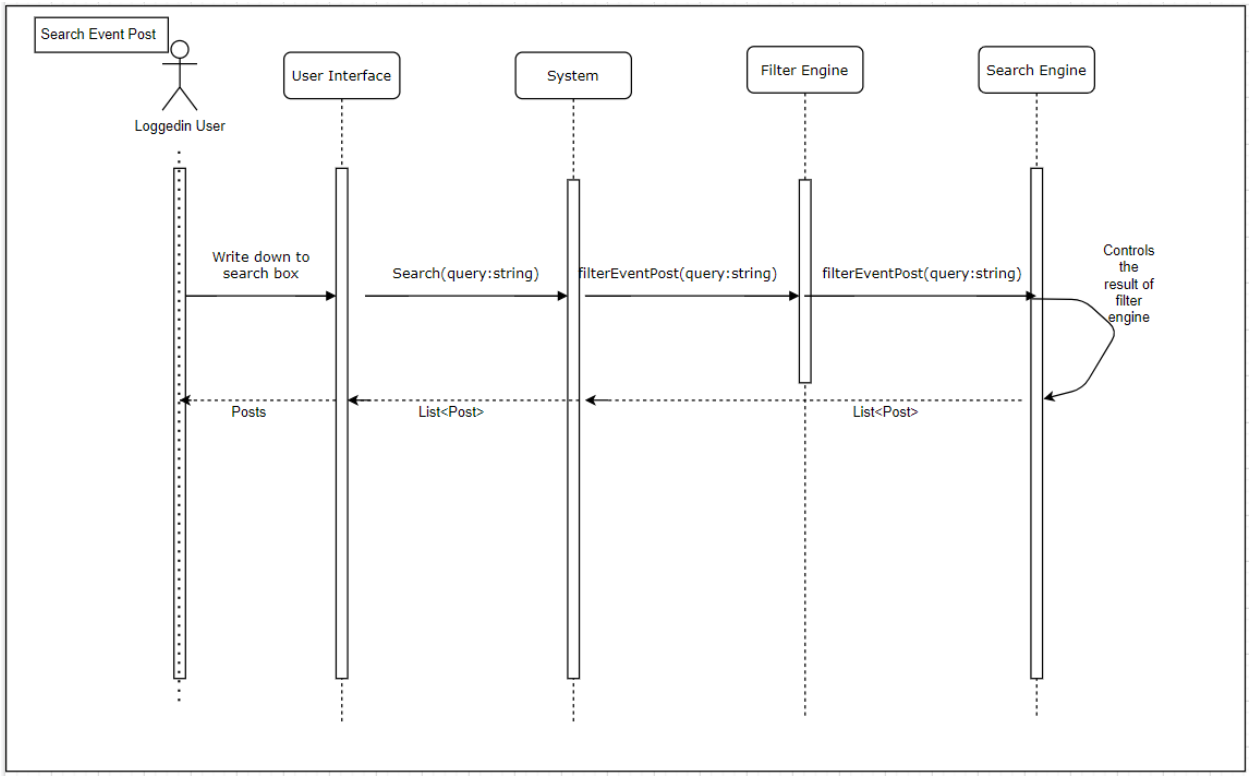
Sign Up


User
:User
:UserController
clicks the sign up page
createUser(params)
confirmEmail: True
Alt
Sign Up Successful
return true
successful message and
redirect to login page
Sign Up Failed
return false
confirmEmail: False
failed message
CreateUser Params:
name, surname, email, password,
interests, profile picture, privacy preference,
personel information

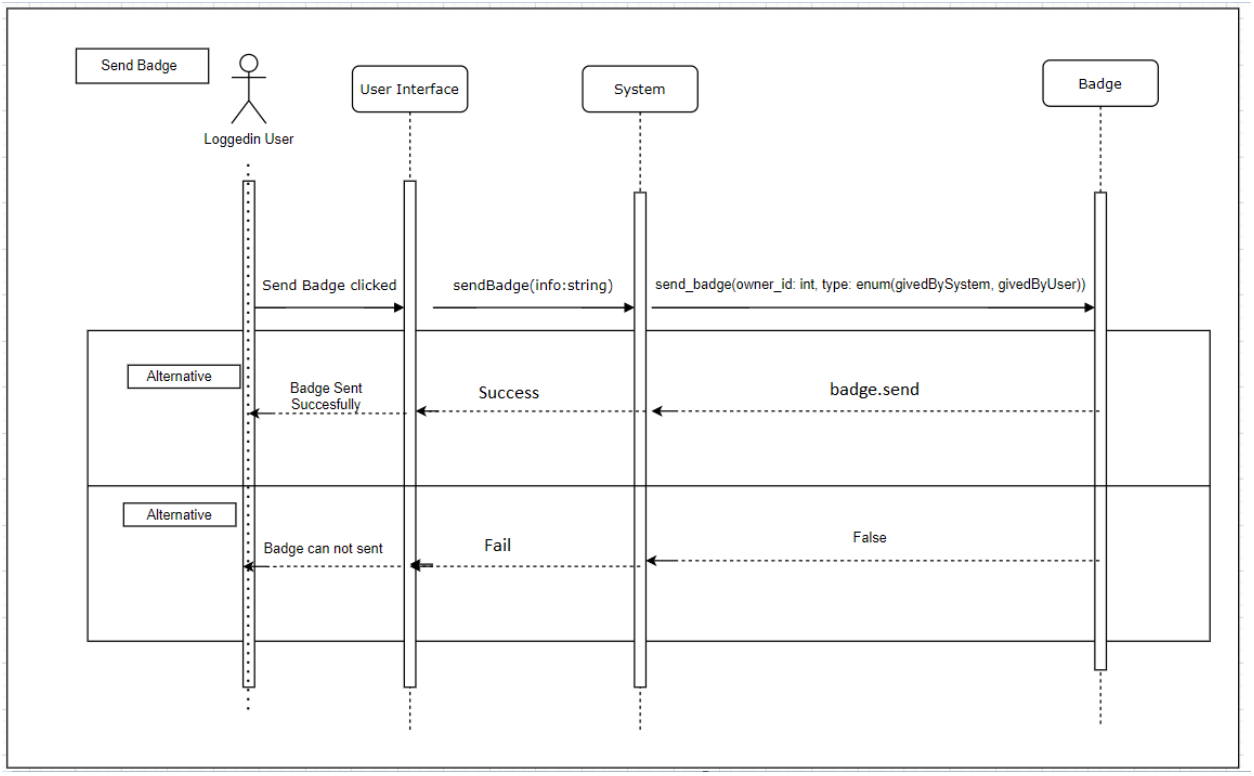
Login



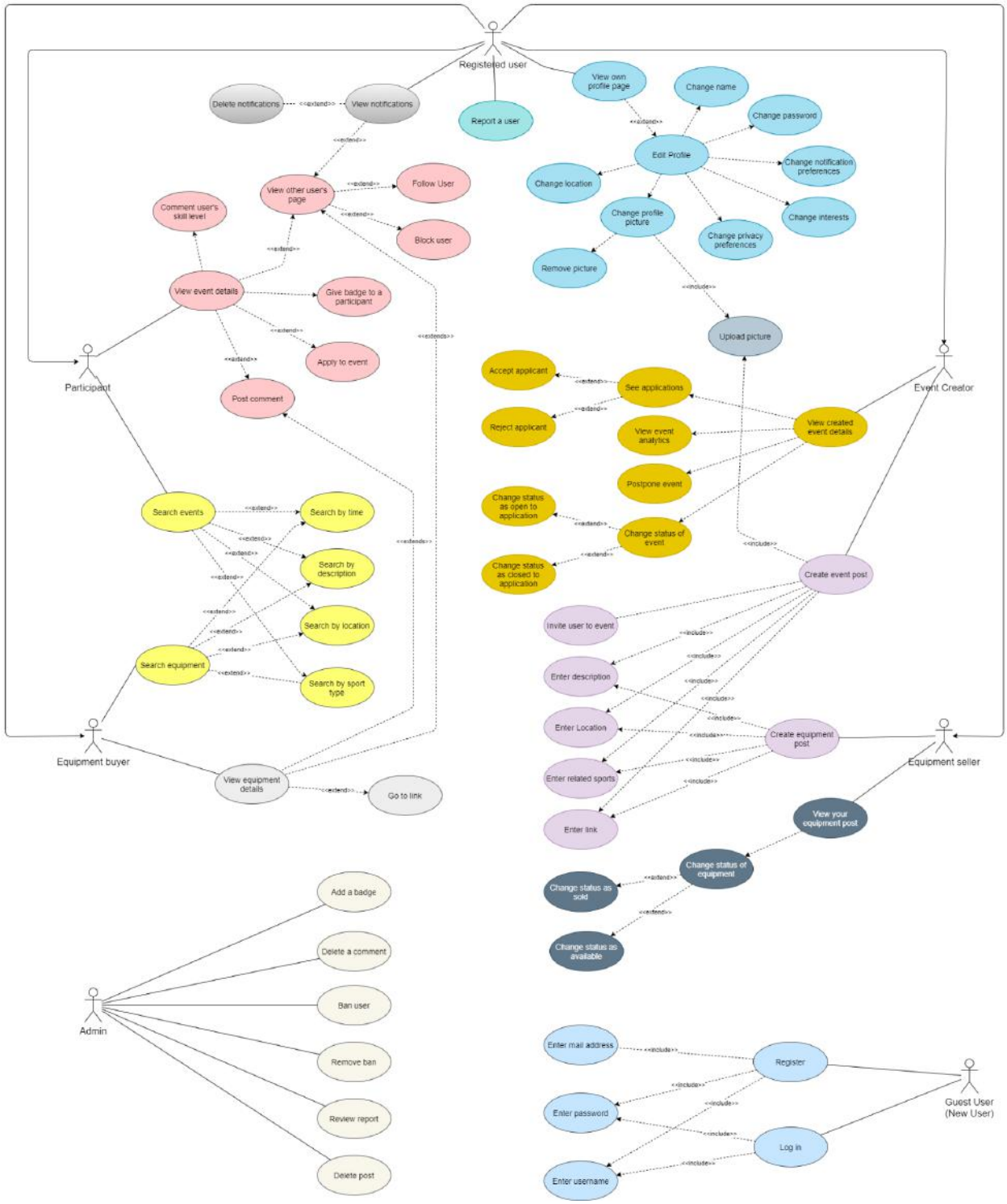
Search Event Post


Search Event Post
Loggedin User
User Interface
System
Filter Engine
Search Engine
Write down to search box
Search(query:string)
filterEventPost(query:string)
filterEventPost(query:string)
Controls the result of filter engine
Posts
List<Post>
List<Post>

Send Badge



Use Case Diagram



Scenarios and Mockups

Scenario 1 & Mockup

Kadir Hallederoğlu

Persona

- **Kadir Hallederoğlu**
- **40 years old, male**
- **Football field and a football shop owner in Ataşehir/İstanbul**
- **Very ambitious, greedy**

User Story

Kadir Hallederoğlu is a 40 years old, very greedy, a football field and a football equipments shop owner. He owns a sentetic football field called "Yeşil Halı Saha" and next to the field there is a shop that sport equipments are sold. There is a website of this shop where people can also buy equipments online. This place resides in Ataşehir/İstanbul. Kadir is very ambitious to promote his business. He discovered this web site and wants to create an account so that users in this

application can see and visit his field and maybe also buy some equipments from his shop's website.

Goals

- **He wants to sign up the application using his business' name Yeşil Halı Saha.**
- **He wants to customize his business' profile.**
- **He wants to post links to his sports equipments that he sells in his shop's website.**

Pre-Conditions

- **He doesn't already have an account.**

Scenario

1. **He enters system. The home page is shown.**
2. **He clicks the sign up button then the sign up page shows up.**
3. **He types his name, surname, email to the corresponding blanks.**
4. **A verification code is send to his email. He copies the numbers to the authentication section and submits.**
5. **Site is automatically refreshed. A page that asks username, password and agreement on terms& conditions opens. He fills these information. He writes the name of his field "Yeşil Halı Saha" as a username.**
6. **Site asks the user to enter the kinds of sports he is interested in and his skill levels in those sports.**
7. **The site shows his profile page.**
8. **He completes his profile information: He writes the location of Yeşil Halı Saha. He puts a photo of his field as a profile picture. He states his sports interest as football.**
9. **He creates a post about a football ball that he normally sells on his shop's website. He fills the type of the sport that the equipment is related to as football. He writes the description of the ball and why people should buy it. He adds the link to the page that ball is sold in his shop's website.**

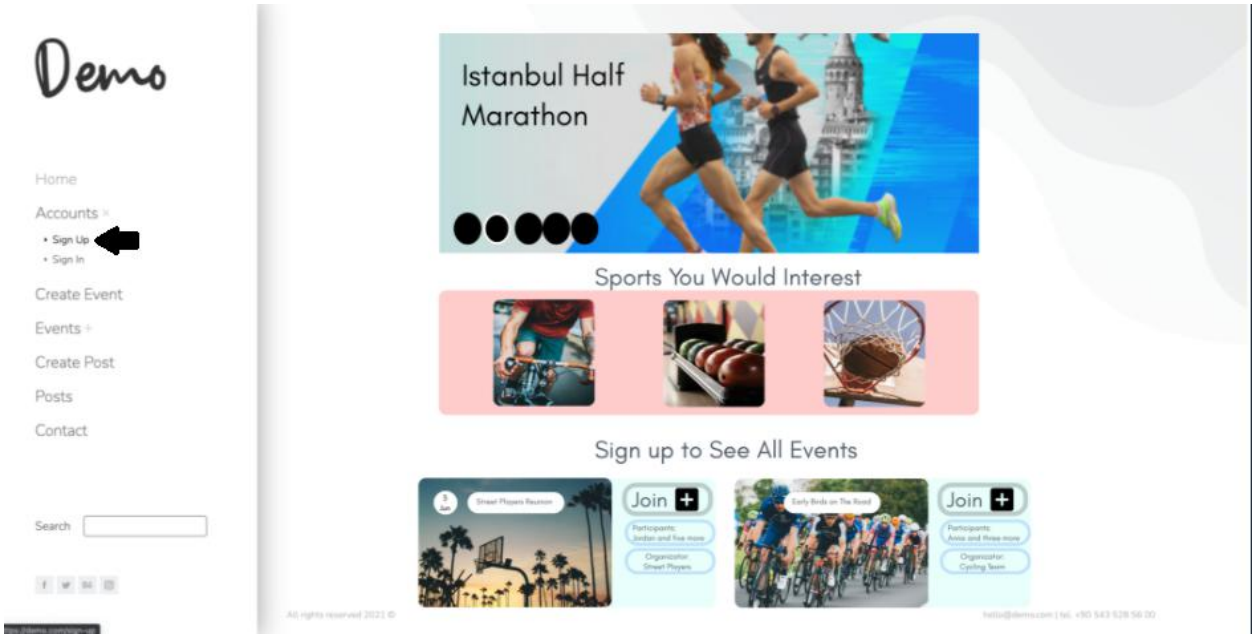
Acceptance Criteria

- *1.2.1.1.1* **Sign up shall require the user to enter their e-mail address, name and surname.**
- *1.2.1.1.2* **E-mail shall be verified via a code.**
- *1.2.1.1.3* **After successful sign up, the system shall go on the next step that the user specifies a password, a username and agrees the terms & conditions contracts.**
- *1.2.1.1.4* **The username shall be at least 4 characters long and containing only alphanumeric characters.**
- *1.2.1.1.5* **The username shall be non-taken**
- *1.2.1.1.6* **Password shall be at least 6 characters long, containing at least one number and one letter and containing only alphanumeric characters.**
- *1.2.1.1.7* **Password shall be asked twice**
- *1.2.1.1.8* **After registration steps are completed, users shall enter the kinds of sports they are interested in and their skill levels in those sports.**
- *1.1.1.1* **Users shall have a profile page.**
- *1.1.1.3* **Users shall be able to add profile picture.**
- **1.1.1.4 There is a default profile picture.**
- *1.1.1.6* **Users will be able to show the sports they are interested in.**
- **1.1.2.8 Users shall be able to post the link to a sports equipment which is useful/ has a good price etc. so that other users may benefit from them.**
- **1.2.4.1 In the sports equipment posts, there shall be a section to write the types of sports whose equipments are offered, description of the sports equipments and why they are suggested. These informations are required**
- **1.2.4.2 In the posts about sports equipments, there shall be a place to put a link to a site where the sport equipment can be found. When the link is clicked, a new page to the site shall open. This information is optional.**
- **1.2.4.3 In the posts about sports equipments, there shall be a place to specify the location of the shop if the shop is not only online and also has a store. This information is optional.**

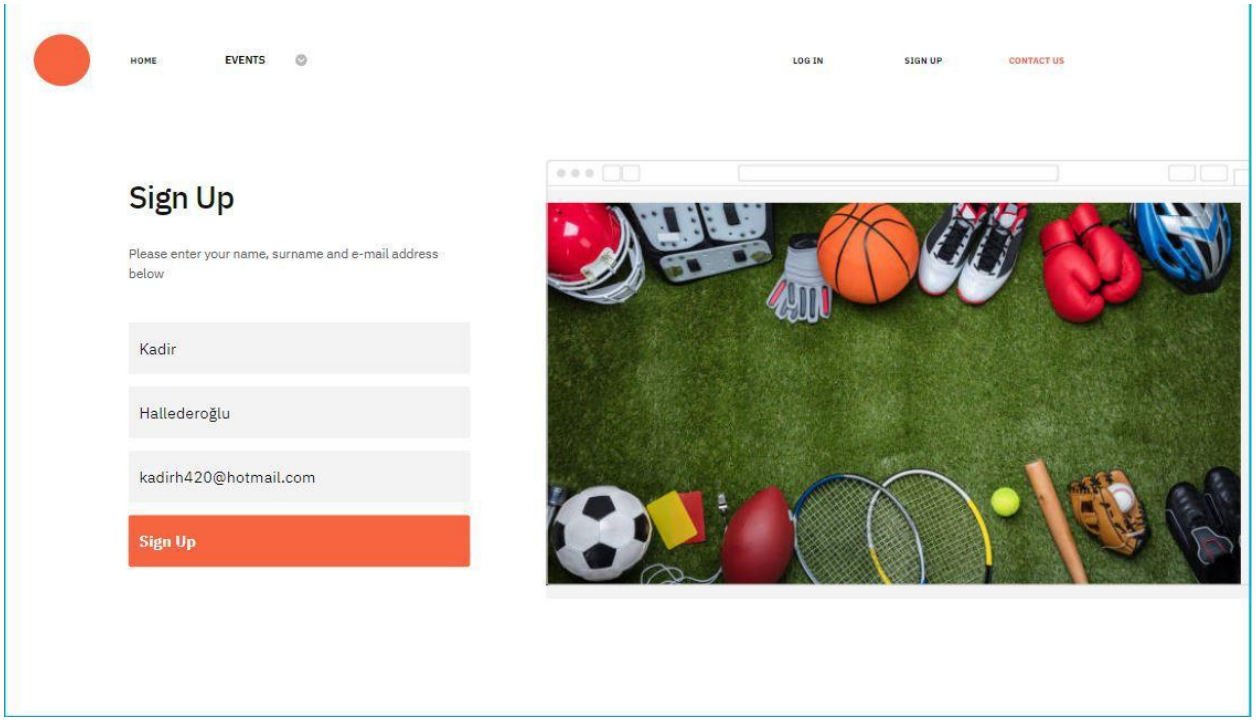
Mockup

Web

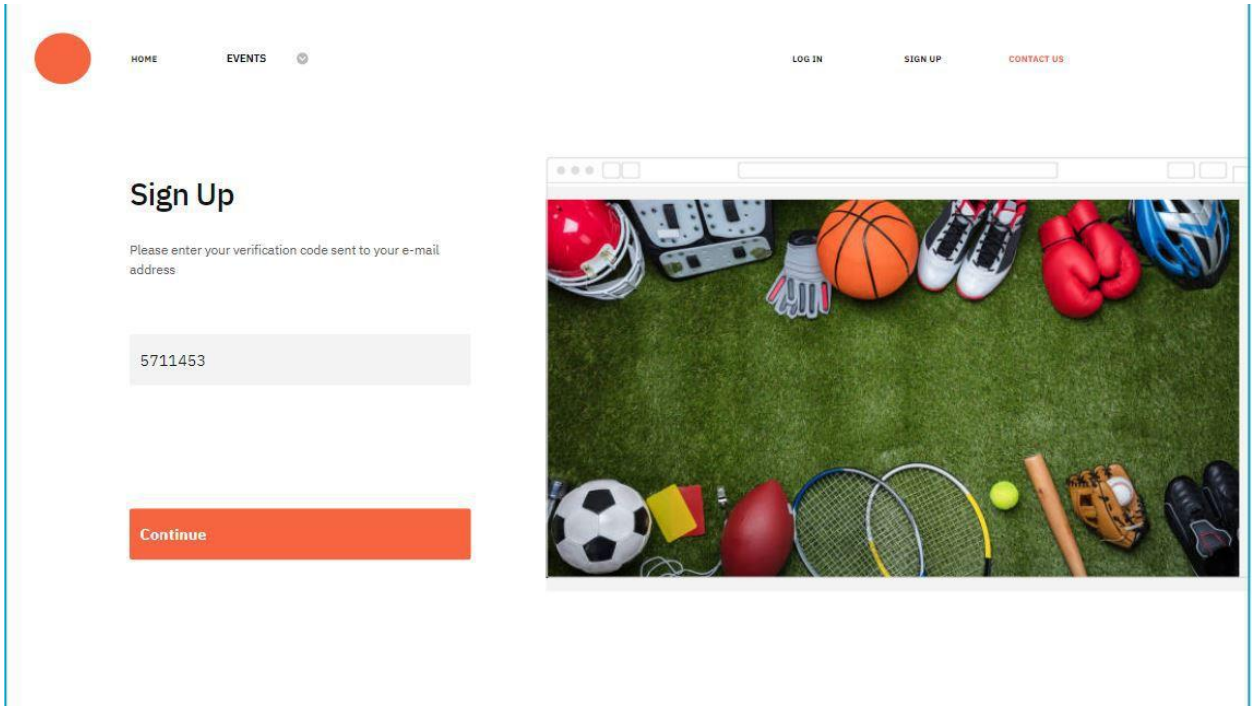
1. He enters system. The home page is shown.
2. He clicks the sign up button then the sign up page shows up.



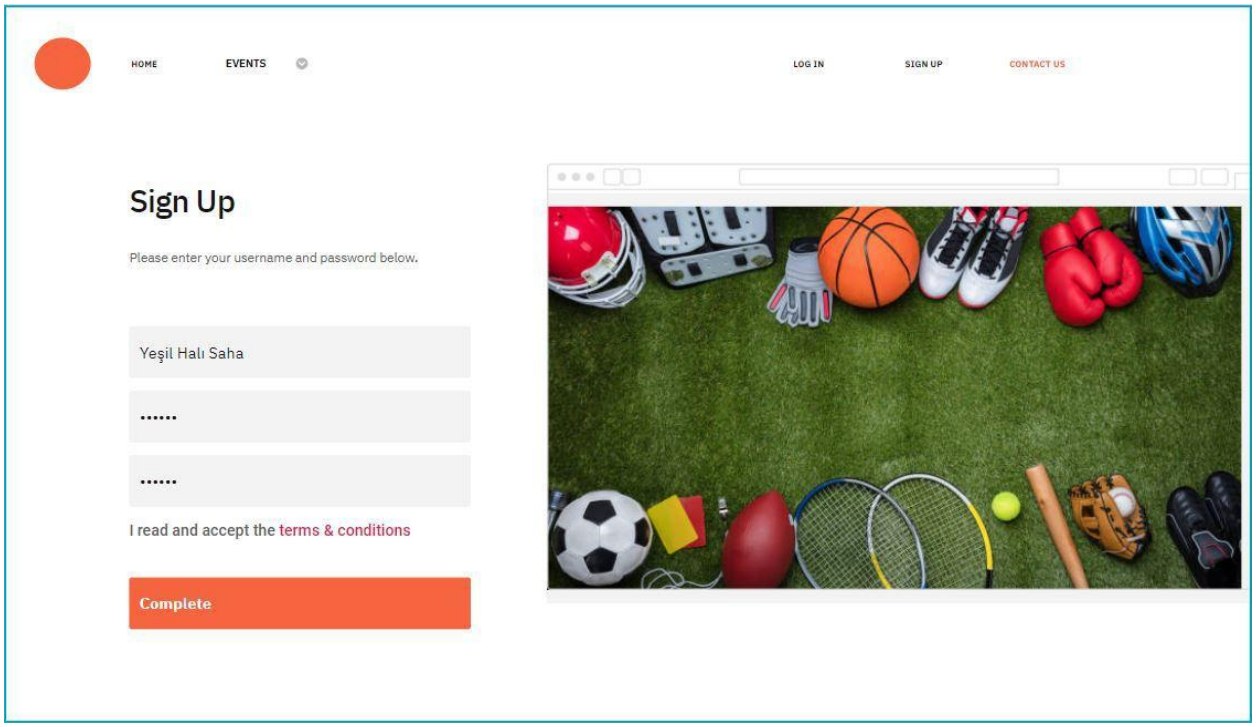
3. He types his name, surname, email to the corresponding blanks.



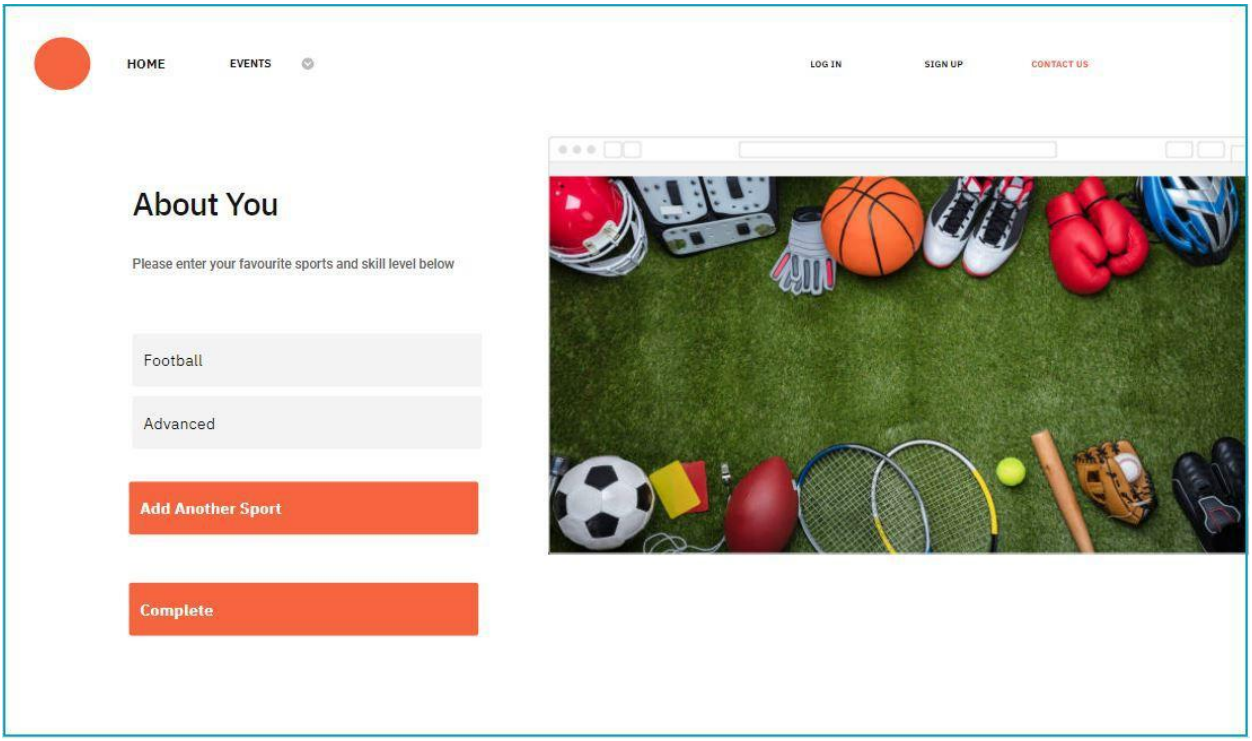
4. A verification code is send to his email. He copies the numbers to the authentication section and submits.



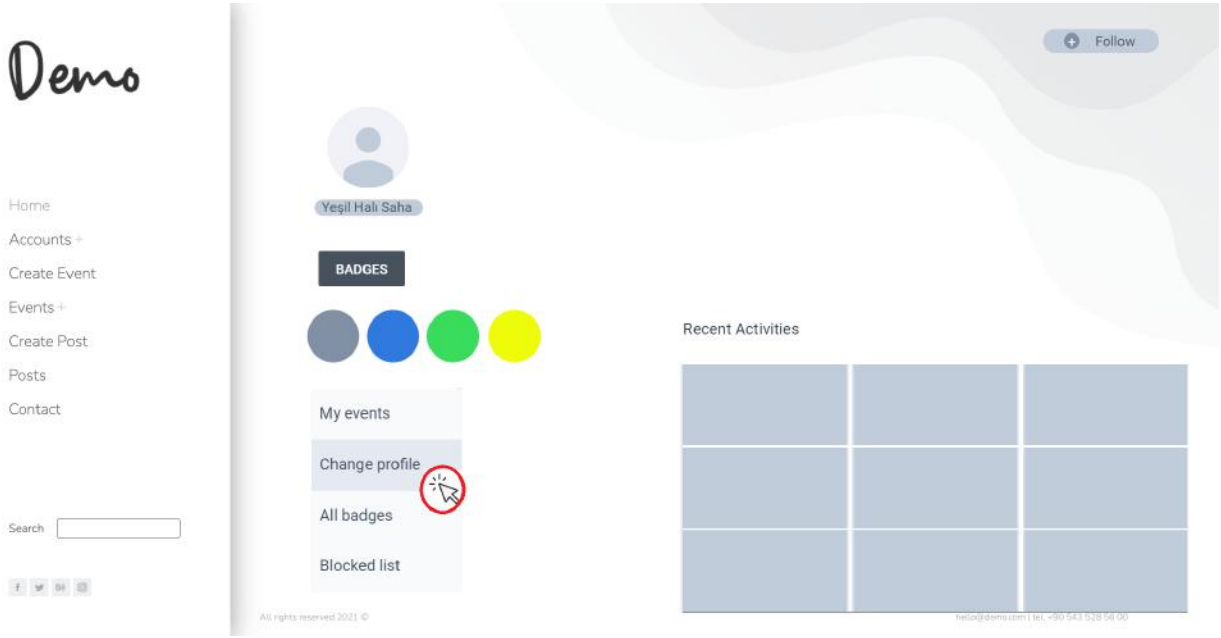
5. Site is automatically refreshed. A page that asks username, password and agreement on terms& conditions opens. He fills these information. He writes the name of his field "Yeşil Halı Saha" as a username.



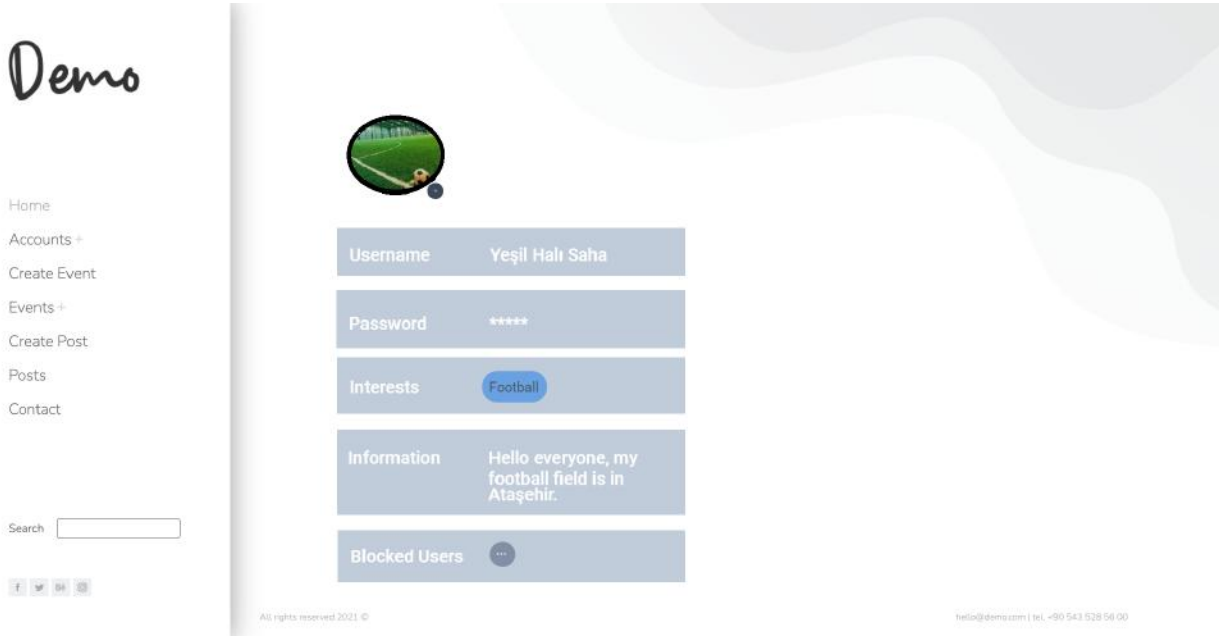
6. Site asks the user to enter the kinds of sports he is interested in and his skill levels in those sports.



7. The site shows his profile page.



8. He completes his profile information: He writes the location of Yeşil Halı Saha. He puts a photo of his field as a profile picture. He states his sports interest as football.

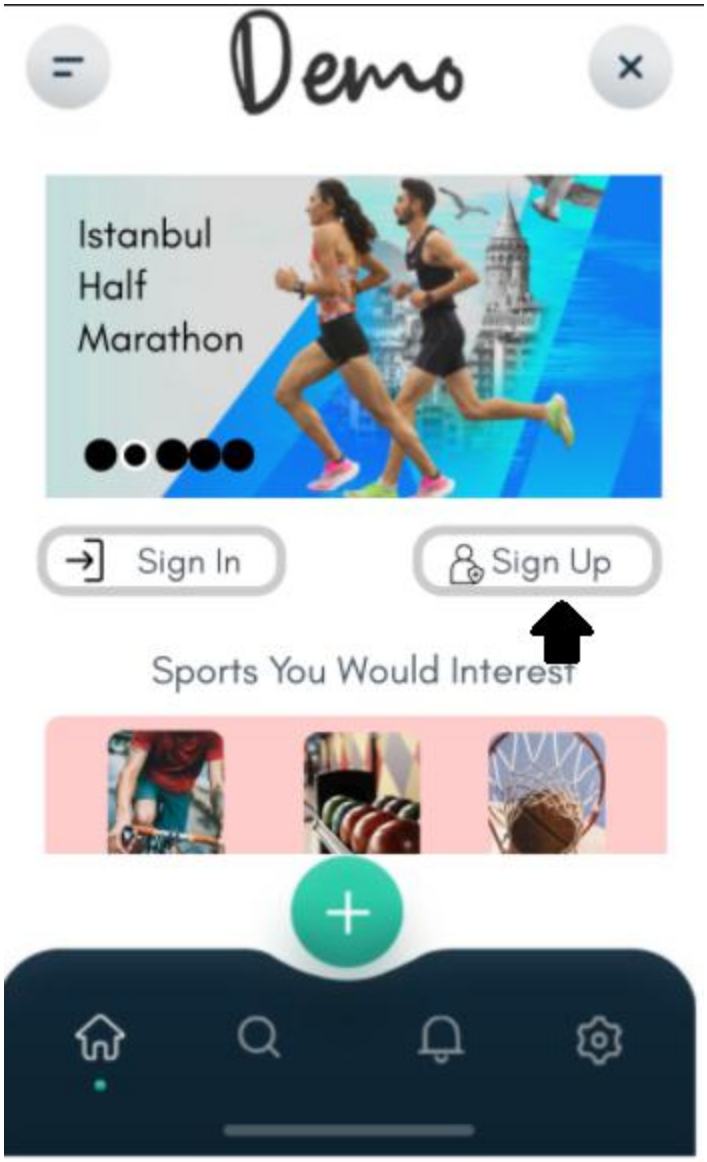


9. He creates a post about a football ball that he normally sells on his shop's website. He fills the type of the sport that the equipment is related to as football.

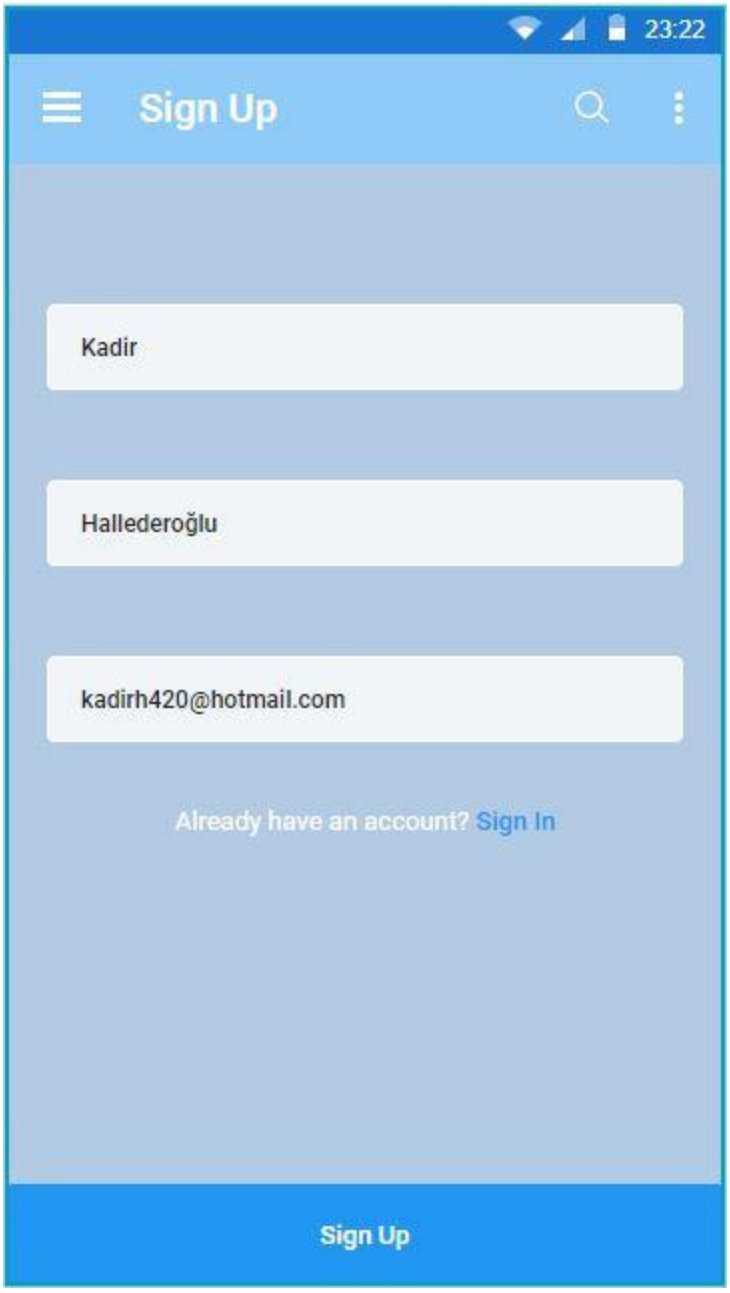
He writes the description of the ball and why people should buy it. He adds the link to the page that ball is sold in his shop's website.

Android

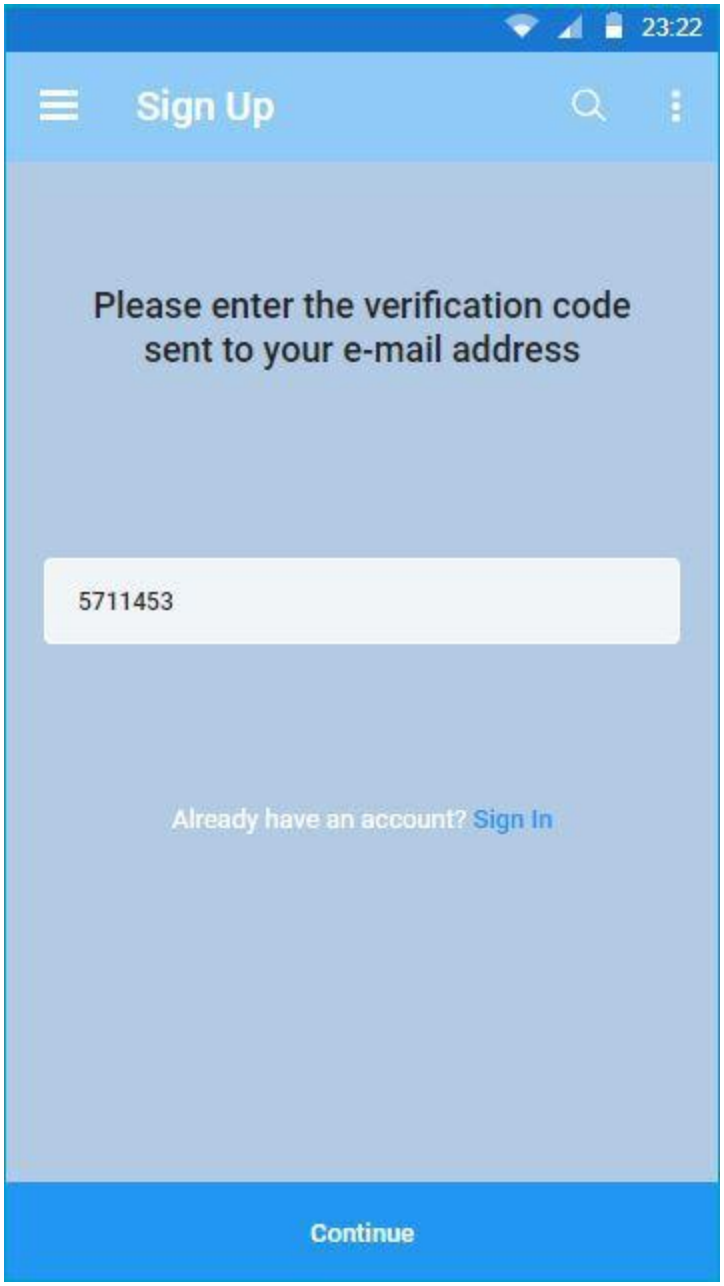
1. He enters system. The home page is shown.
2. He clicks the sign up button then the sign up page shows up.



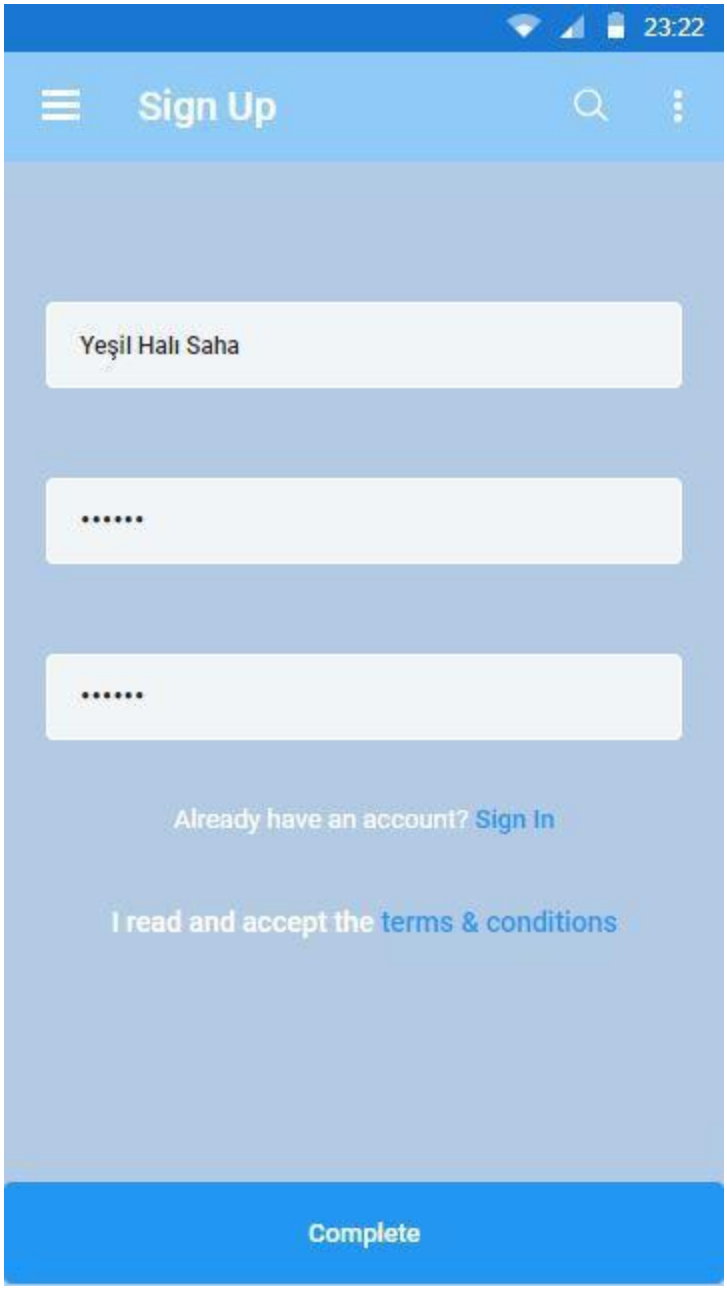
3. He types his name, surname, email to the corresponding blanks.



4. A verification code is send to his email. He copies the numbers to the authentication section and submits.

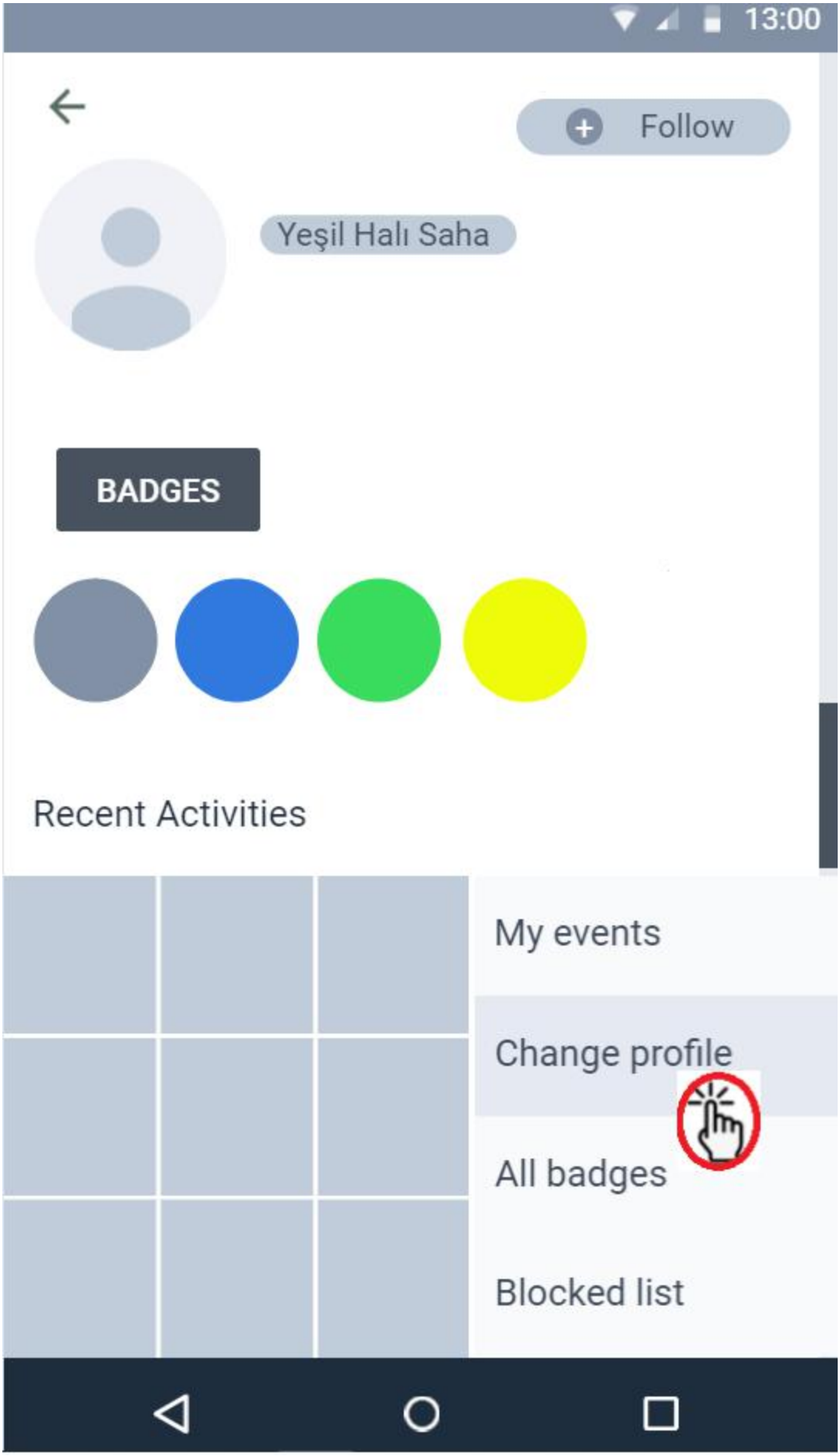


5. Site is automatically refreshed. A page that asks username, password and agreement on terms& conditions opens. He fills these information. He writes the name of his field "Yeşil Halı Saha" as a username.

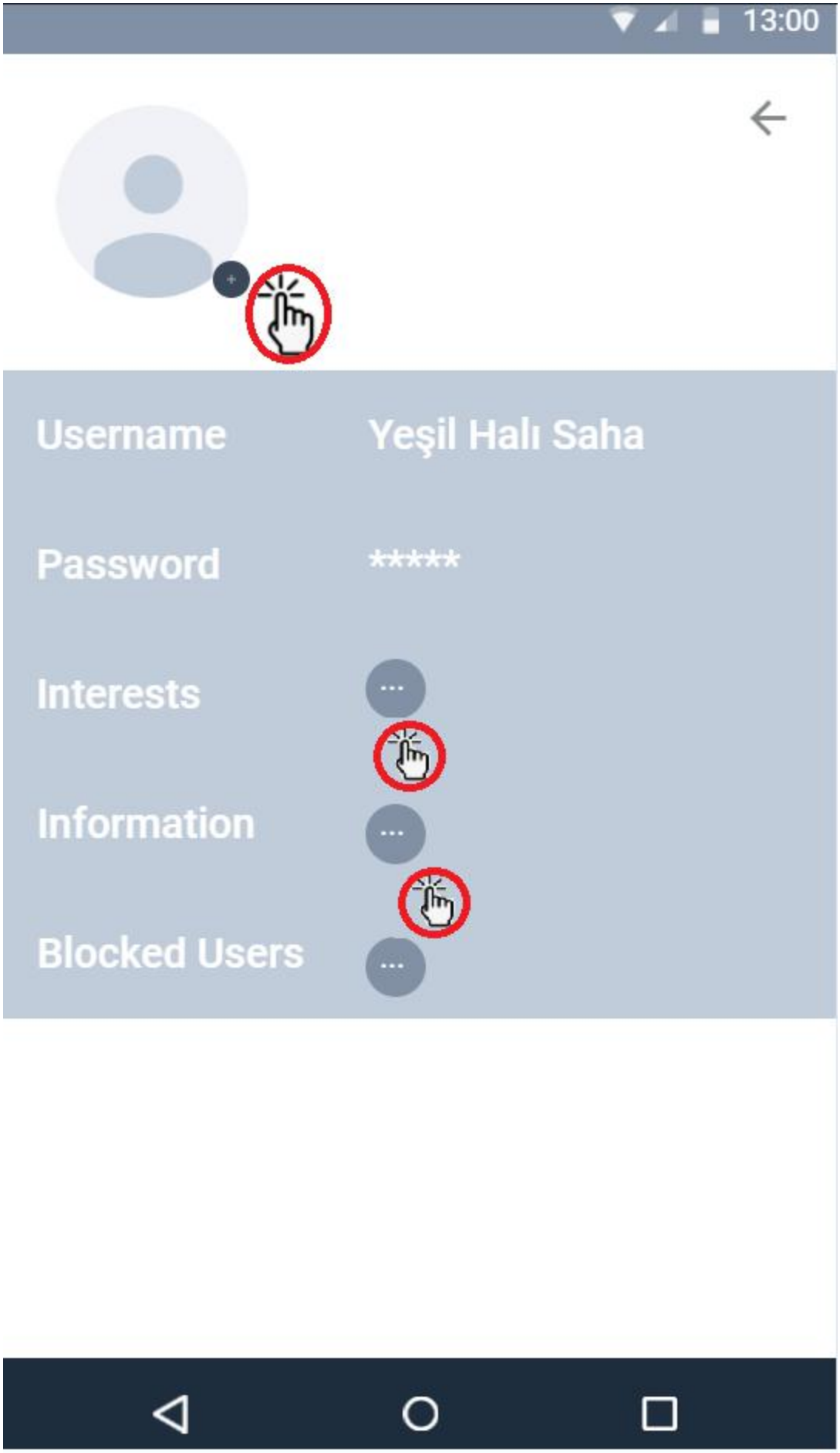


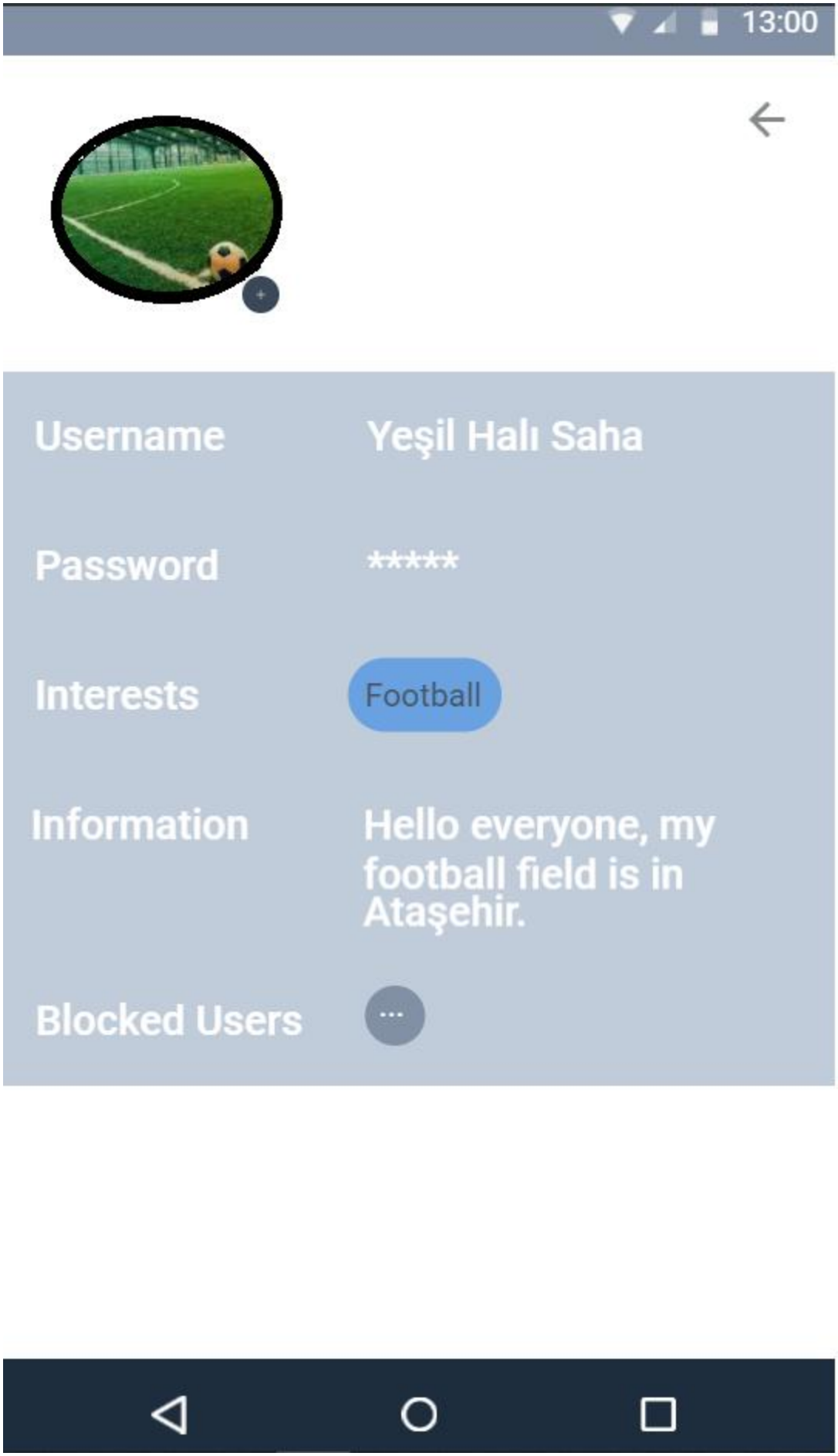
6. Site asks the user to enter the kinds of sports he is interested in and his skill levels in those sports.

7. The site shows his profile page.


13:00
Follow
Yeşil Halı Saha
BADGES
Recent Activities
My events
Change profile
All badges
Blocked list

8. He completes his profile information: He writes the location of Yeşil Halı Saha. He puts a photo of his field as a profile picture. He states his sports interest as football.


13:00
Username
Yeşil Halı Saha
Password
*****
Interests
Information
Blocked Users


13:00
Username
Yeşil Halı Saha
Password
*****
Interests
Football
Information
Hello everyone, my football field is in Ataşehir.
Blocked Users

9. He creates a post about a football ball that he normally sells on his shop's website. He fills the type of the sport that the equipment is related to as football. He writes the description of the ball and why people should buy it. He adds the link to the page that ball is sold in his shop's website.

9:41

Demo

Create an Equipment Post

Equipment Category

- Tennis
- Soccer
- Basketball
- Other

Equipment Title

Brand New Football Ball

Equipment Description

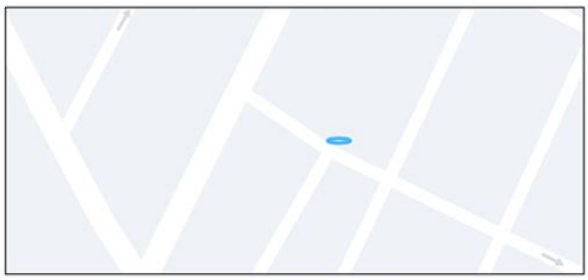
I want to sell my football ball. I bought it 2 weeks ago.

110/120

Equipment Link

https://www.satis.com/brand-new-football-ball

Choose Location



Scenario 2 & Mockup

Persona

- **John Gül**
- **25 years old, male**
- **Basketball player, freelance worker**
- **Likes to meet new people**

User Story

John is an avid fan of street basketball and used the app to organize a few events in the past. He wants to create a new basketball event open to everyone to join.

Goals

- **He wants to log in.**

- **He wants to create a sports event.**

Pre-Conditions

- **He already has an account.**

Scenario

1. **He opens the app.**
2. **He logs in to her account by entering e-mail and password.**
3. **On android when "+" button on the footer of the homepage is clicked and on web when "create event" on the side bar is clicked, event creation screen shows up.**
4. **He enters event's name, date and time, location, sport type, skill level requirement, quota, participant count, required equipment, badges, sets the status as open to application and writes a caption for the event and submits.**

Acceptance Criteria

- *1.1.4.1* **Users shall be able to sign in with their username (or email) and password.**
- **1.1.2.4 Users shall be able to select badges from the options that system provides.**
- *1.1.2.15* **When an event is being created, event creator shall choose among the types of sports, if there is no such event recognized by the system, event creator shall describe the type of the event in terms of its characteristics.**
- *1.2.3.1.1* **Events shall be discoverable by users that are of their interests and that are nearby.**
- *1.2.3.1.2* **Events shall be applicable by users that satisfy the skill set constraint.**
- *1.2.3.1.4* **The following information about the event shall be shown:**

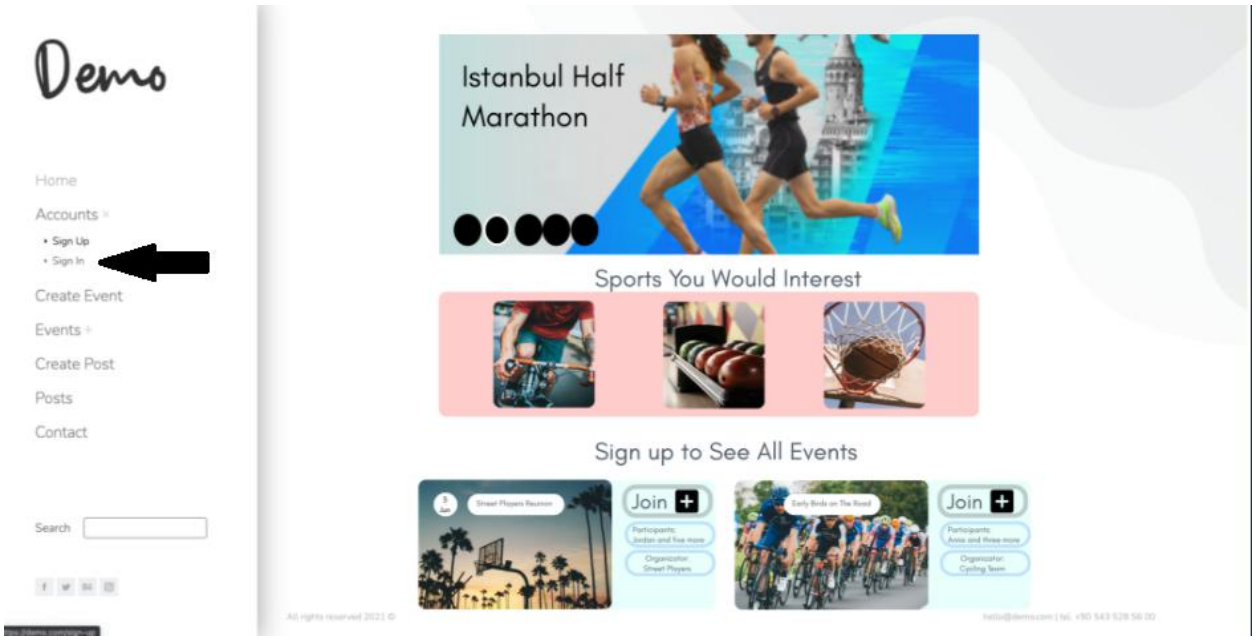
    - *1.2.3.1.4.1* **event's date and time**
    - *1.2.3.1.4.2* **location**
    - *1.2.3.1.4.3* **sport type**
    - *1.2.3.1.4.4* **skill level requirement**
    - *1.2.3.1.4.5* **quota of the sport**
    - *1.2.3.1.4.6* **participant count**

    - *1.2.3.1.4.8* **name of the event**
    - **1.2.3.1.4.10 required equipment**
    - **1.2.3.1.4.11 rules**
    - **1.2.3.1.4.12 badges that might be given at the end of the event**
    - **1.2.3.1.4.13 description of the event**
- **1.2.3.1.7 If an event is requested to be repeated for consequent weeks on the same day of the week and hour, while the event is being created there shall be an option for such repetition that a user might create multiples of the same event simultaneously.**
- **1.2.3.1.8 If there is no default type in the system for an event that a user wants to create, there shall be an "other" option as the type of the sport.**
- **1.2.9.1 System shall provide a number of badge options that users are able to choose one of them and semantically tag them.**

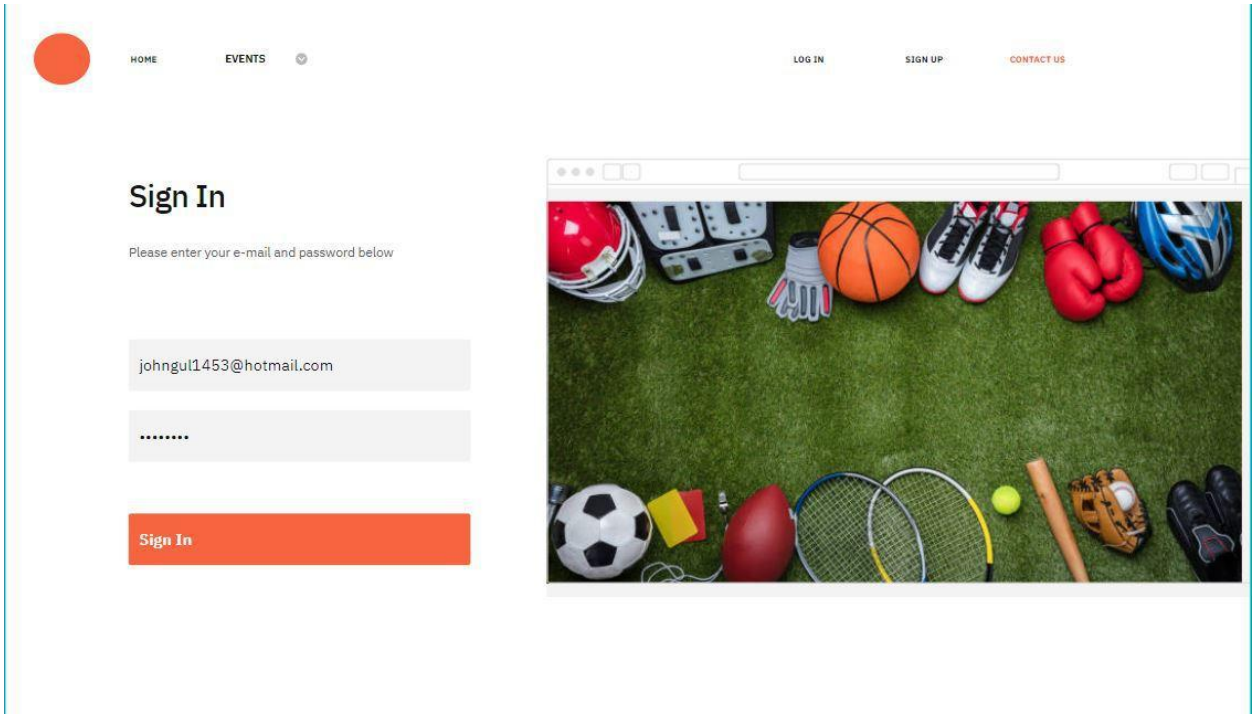
Mockup

Web

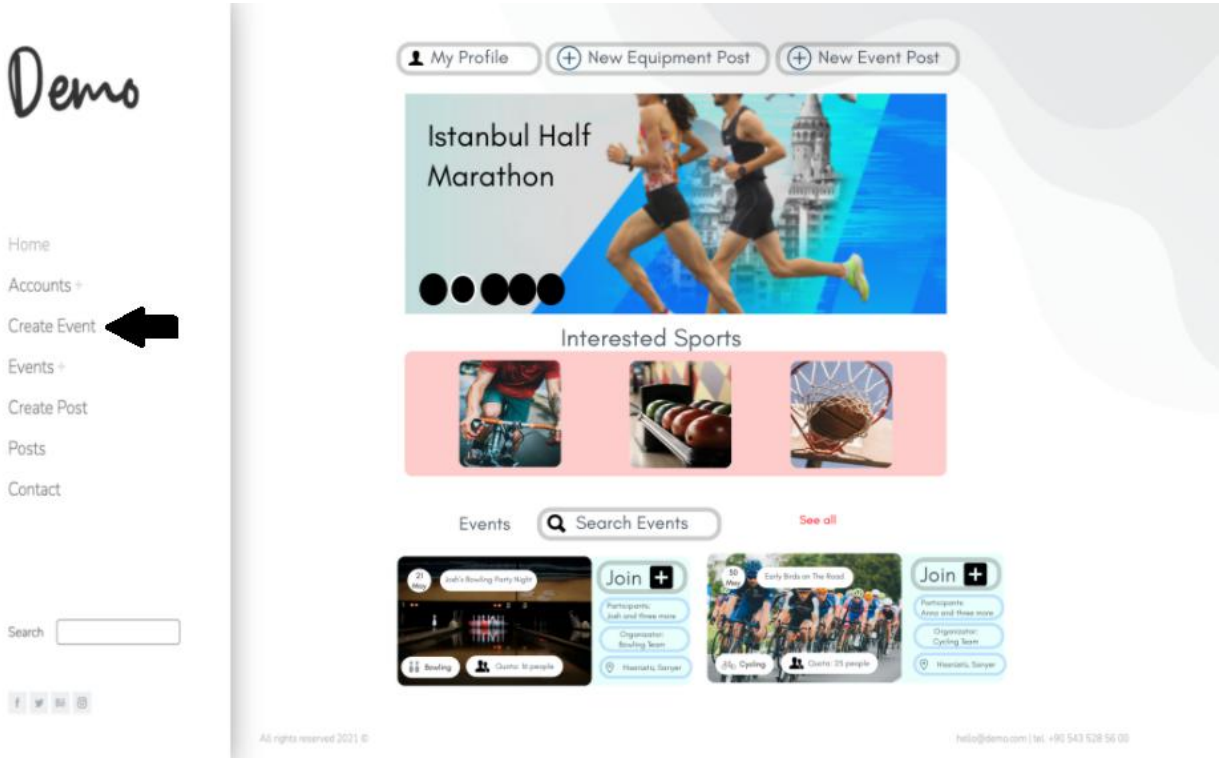
1. He opens the app.



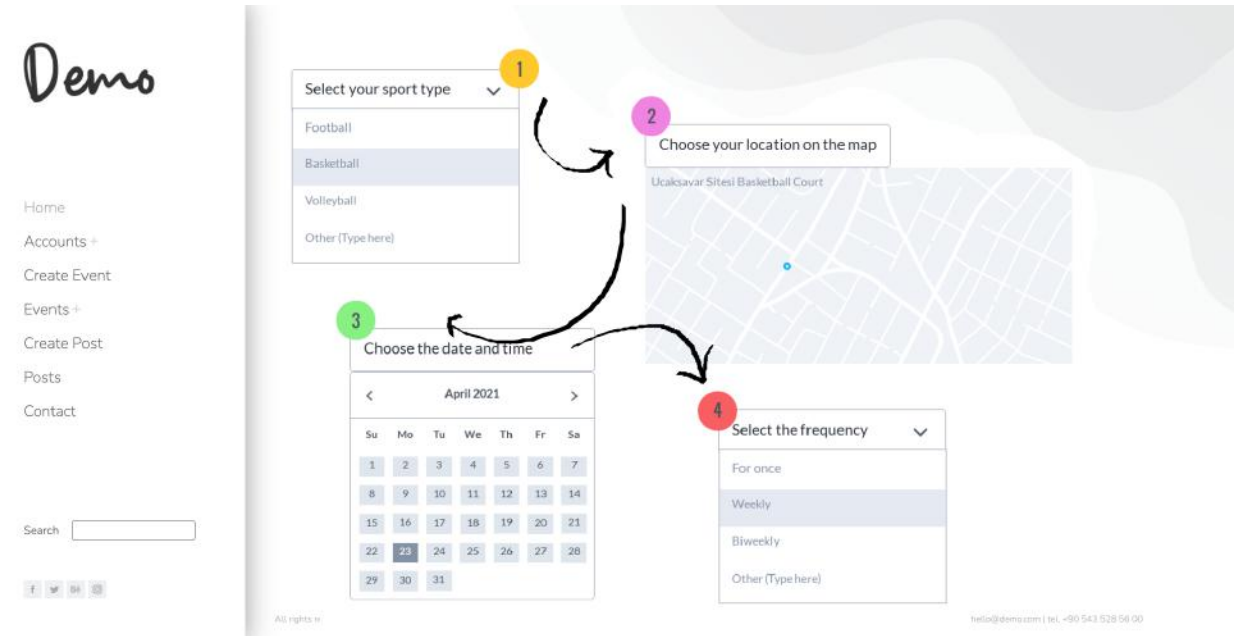
2. He logs in to her account by entering e-mail and password.



3. On android when "+" button on the footer of the homepage is clicked and on web when "create event" on the side bar is clicked, event creation screen shows up.



4. Now, He starts to create her event.
   4.1 Selects the kind of the sport she wish to create. If she cannot find what she is looking for, she can also type below. He selects Basketball.
   4.2 Writes the location down and pins the exact point on the map.
   4.3 Chooses event's date and time.
   4.4 He wants to play basketball every week and he doesn't want to create the same event every week. So, he chooses the event's frequency weekly.
   4.5 Selects the number of players as 10.



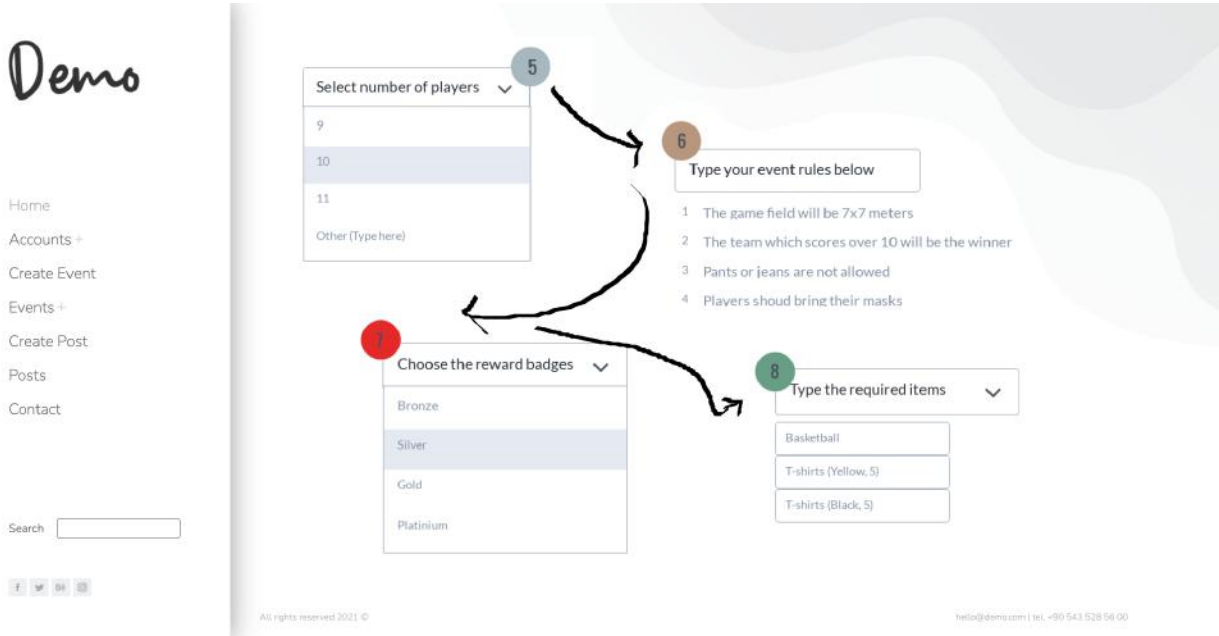
4.5 He types the event rules.

4.6 Chooses the badges which will be given to players after the game ends.

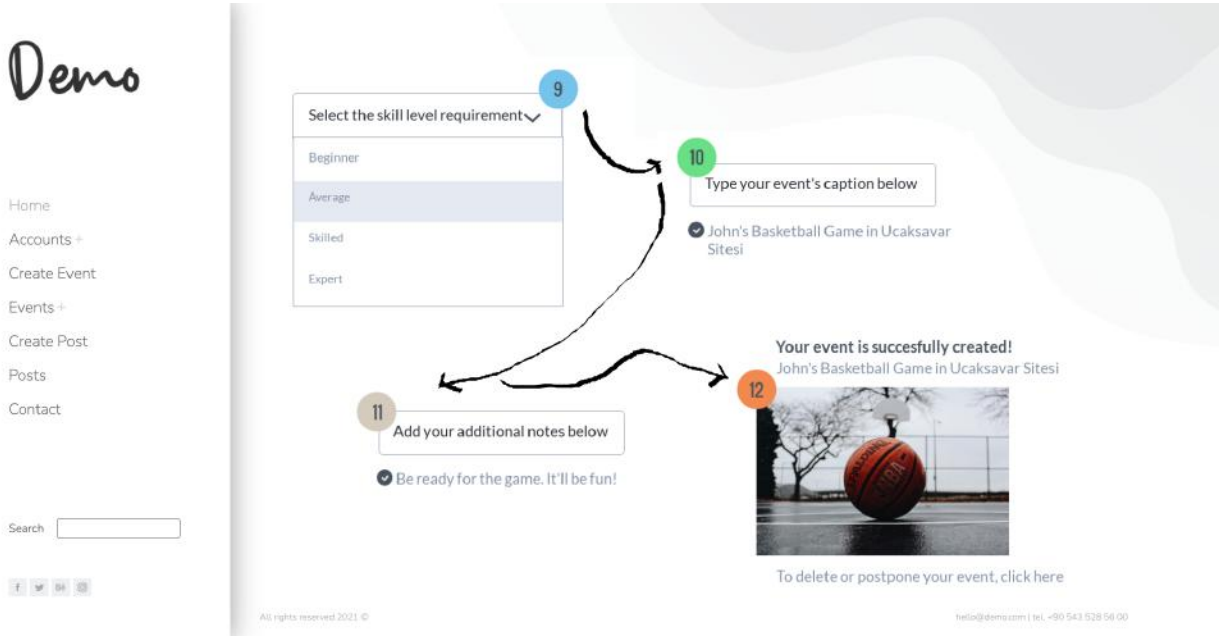
4.7 Types required items.

4.8 Selects the skill requirements for this game. He selects average for this game.

4.9 Lastly, He adds her event's caption and additional notes which will be visible by other users.

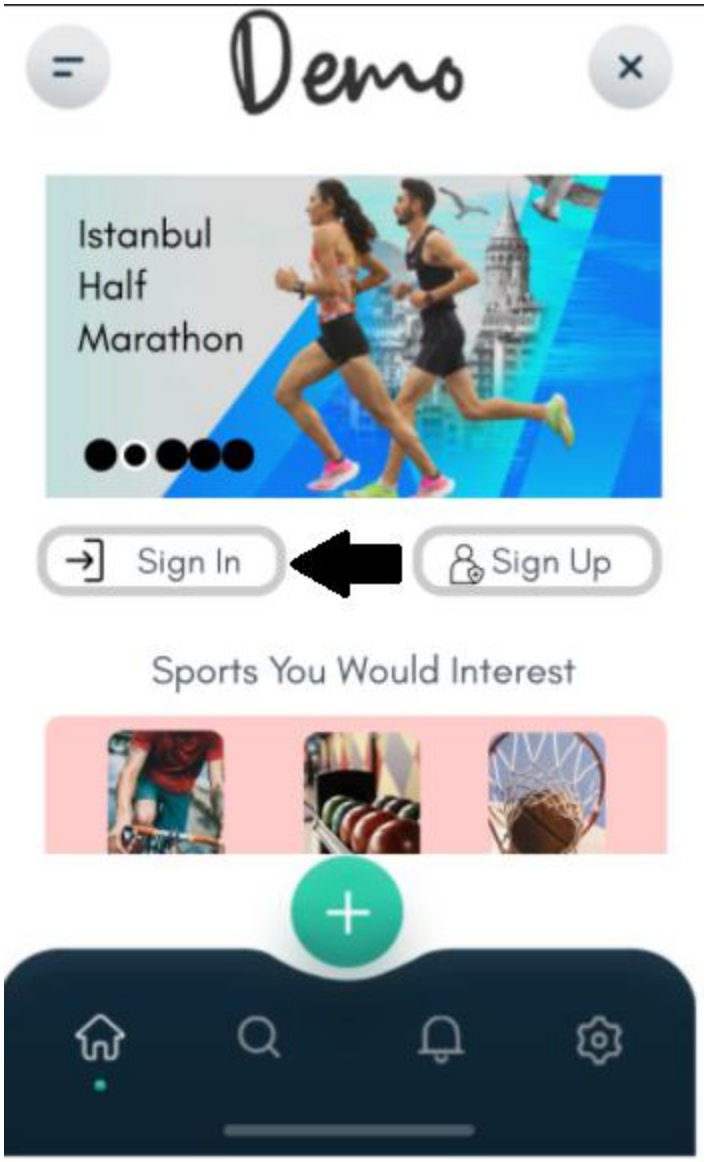


5. Now he sees the screen of "Your event is successfully created."

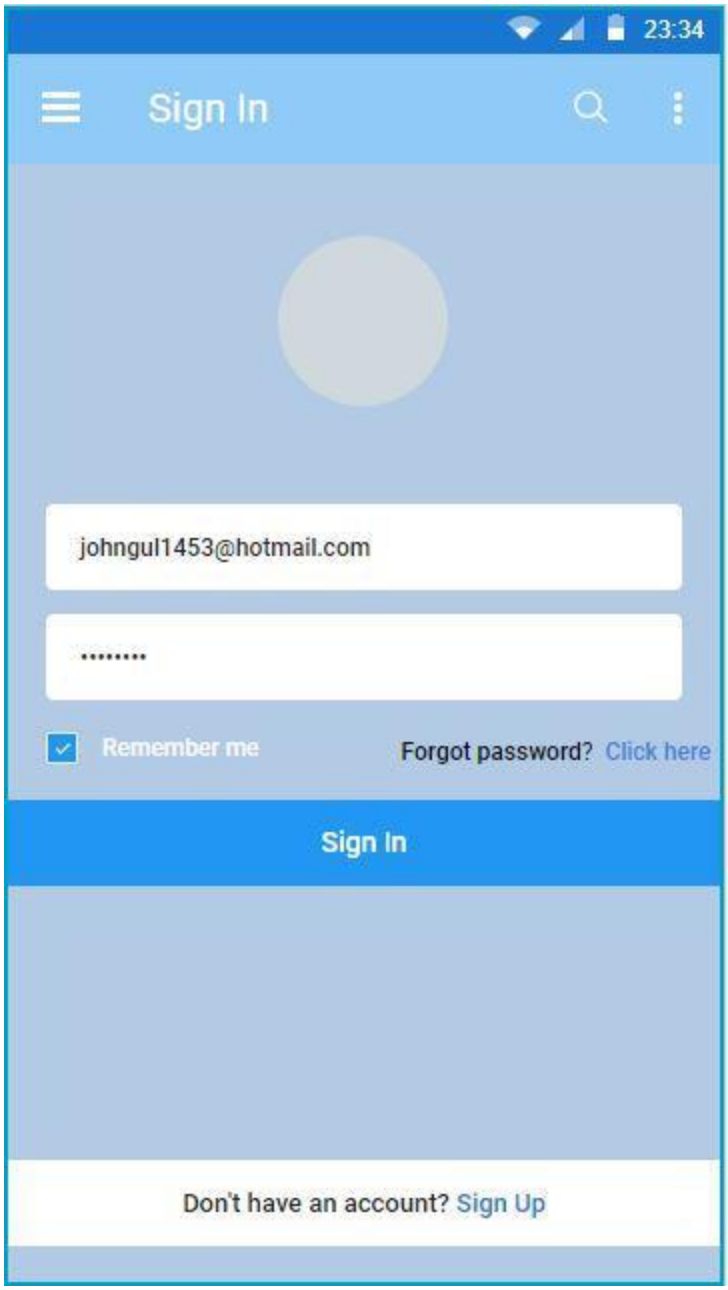


Android

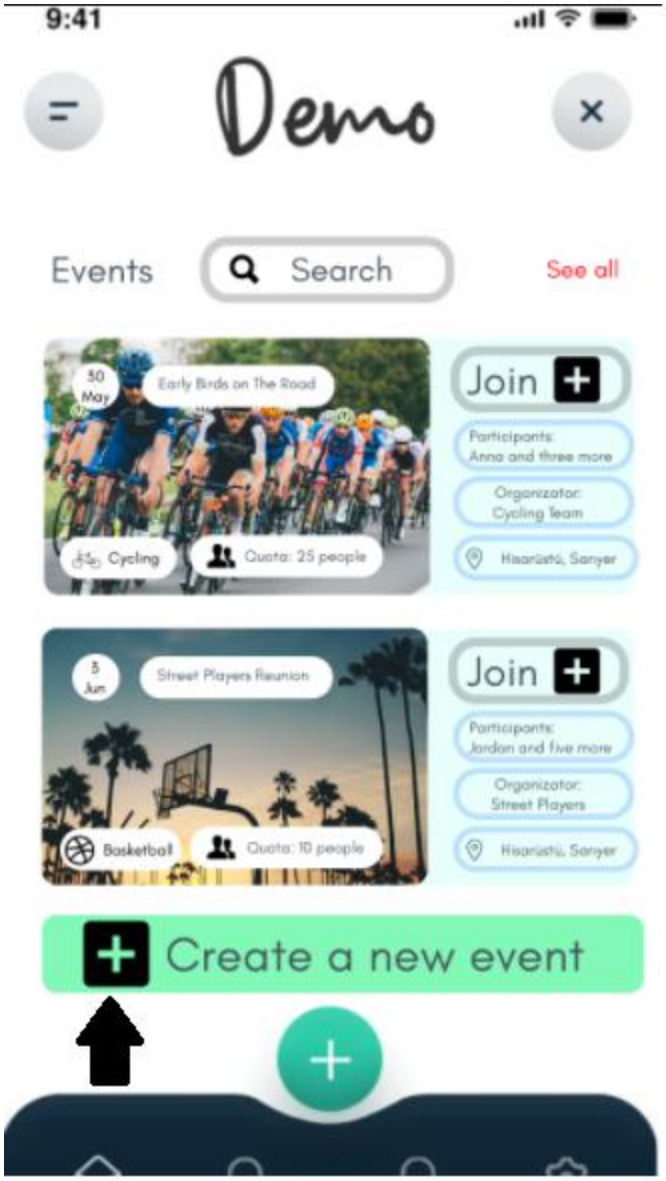
1. He opens the app.



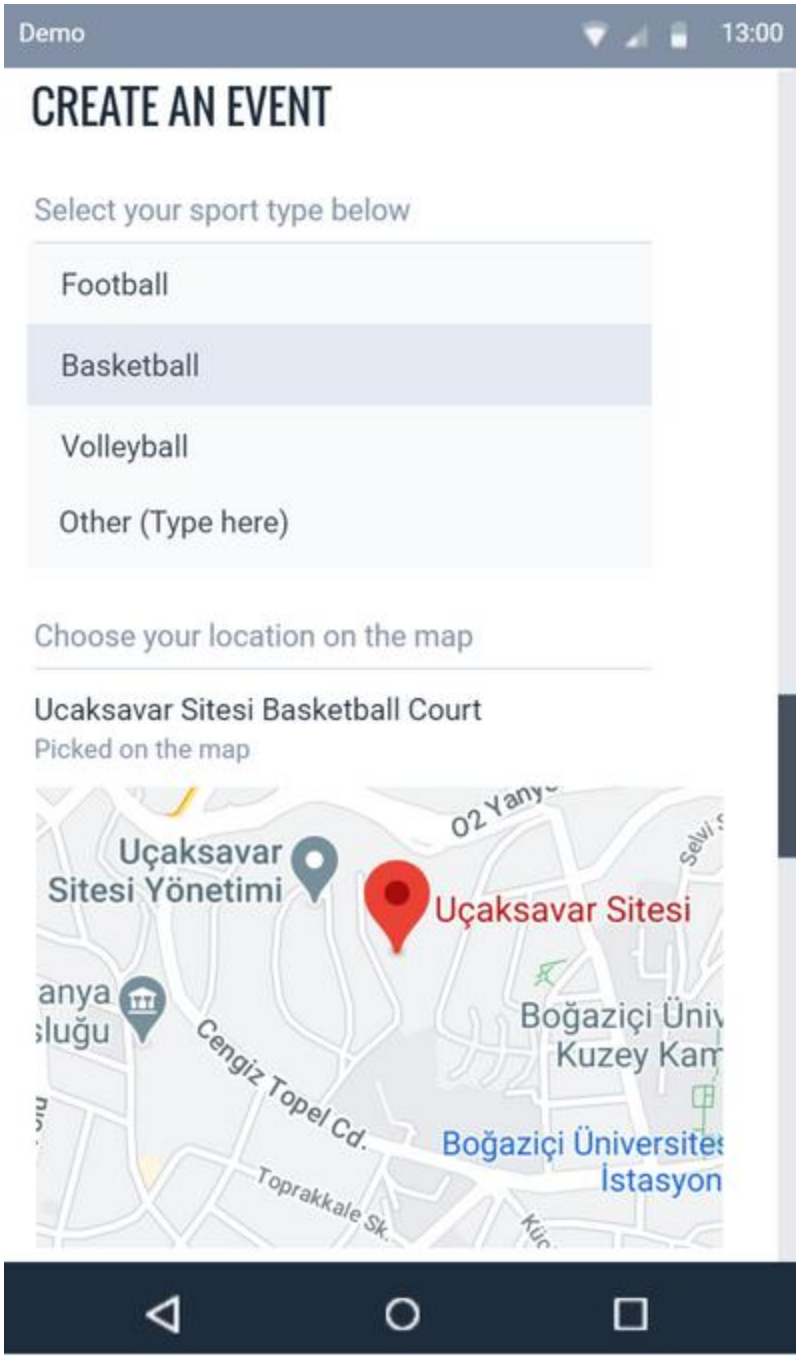
2. He logs in to her account by entering e-mail and password.

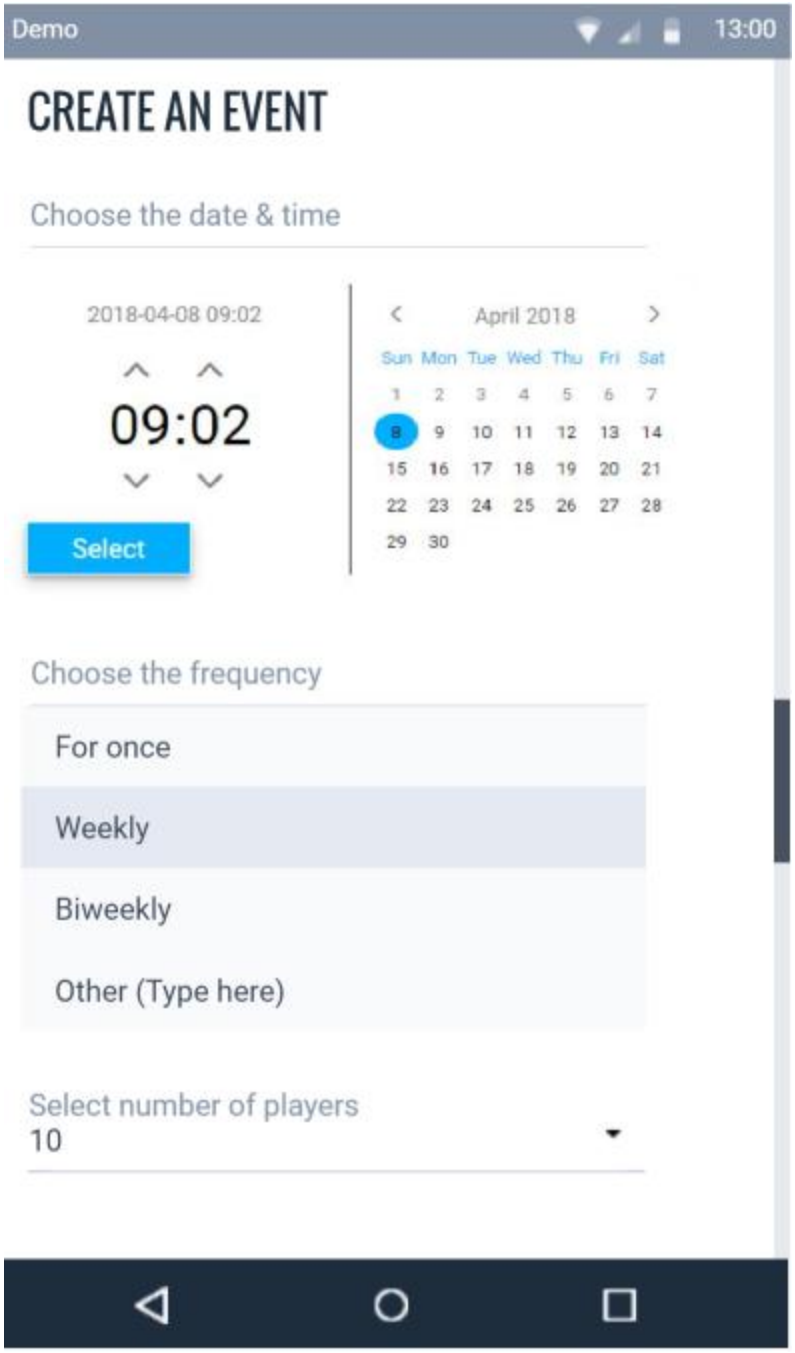


3. On android when "+" button on the footer of the homepage is clicked and on web when "create event" on the side bar is clicked, event creation screen shows up.



4. Now, He starts to create her event.
4.1 Selects the kind of the sport she wish to create. If she cannot find what she is looking for, she can also type below. He selects Basketball.
4.2 Writes the location down and pins the exact point on the map.
4.3 Chooses event's date and time.
4.4 He wants to play basketball every week and he doesn't want to create the same event every week. So, he chooses the event's frequency weekly.
4.5 Selects the number of players as 10.


Demo
13:00
CREATE AN EVENT
Select your sport type below
Football
Basketball
Volleyball
Other (Type here)
Choose your location on the map
Ucaksavar Sitesi Basketball Court
Picked on the map
Uçaksavar Sitesi Yönetimi
Uçaksavar Sitesi
Cengiz Topel Cd.
Toprakkale Sk.
Boğaziçi Üniversitesi İstasyon



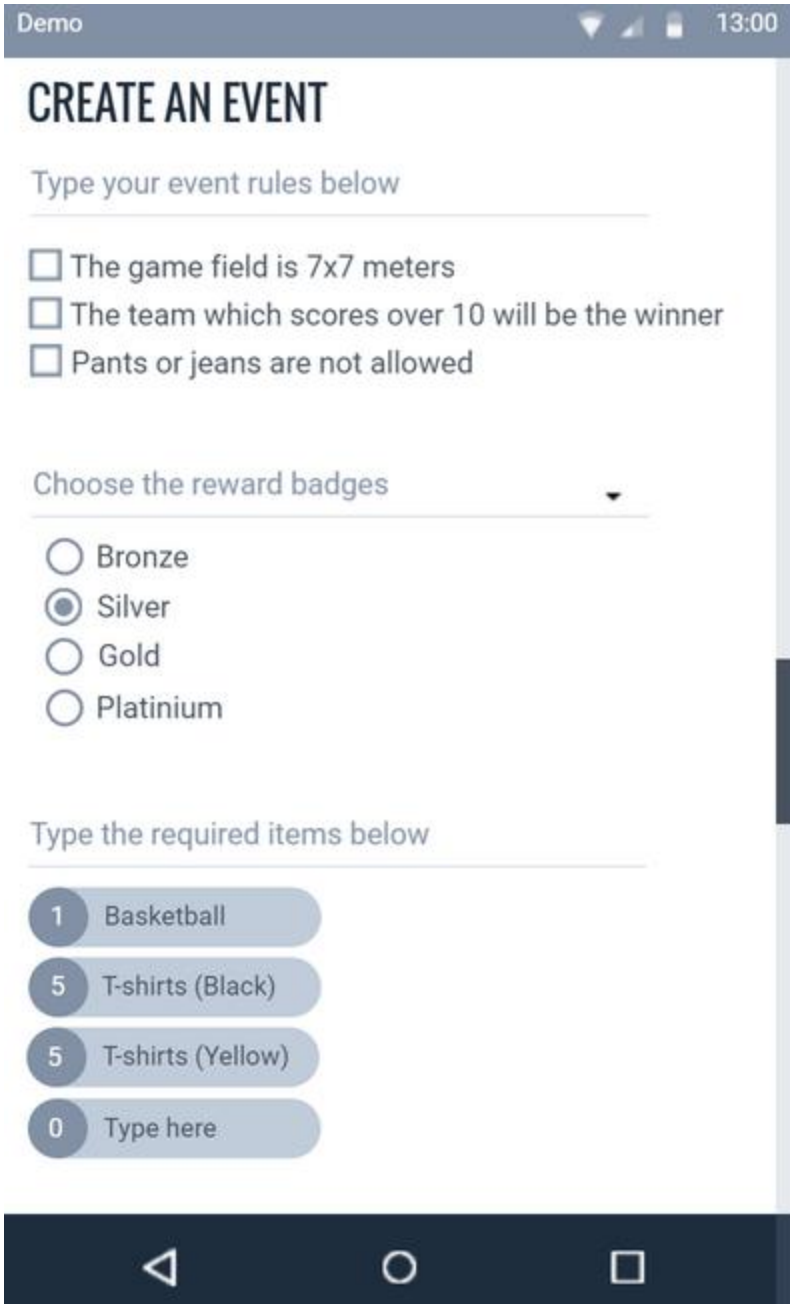
4.5 He types the event rules.

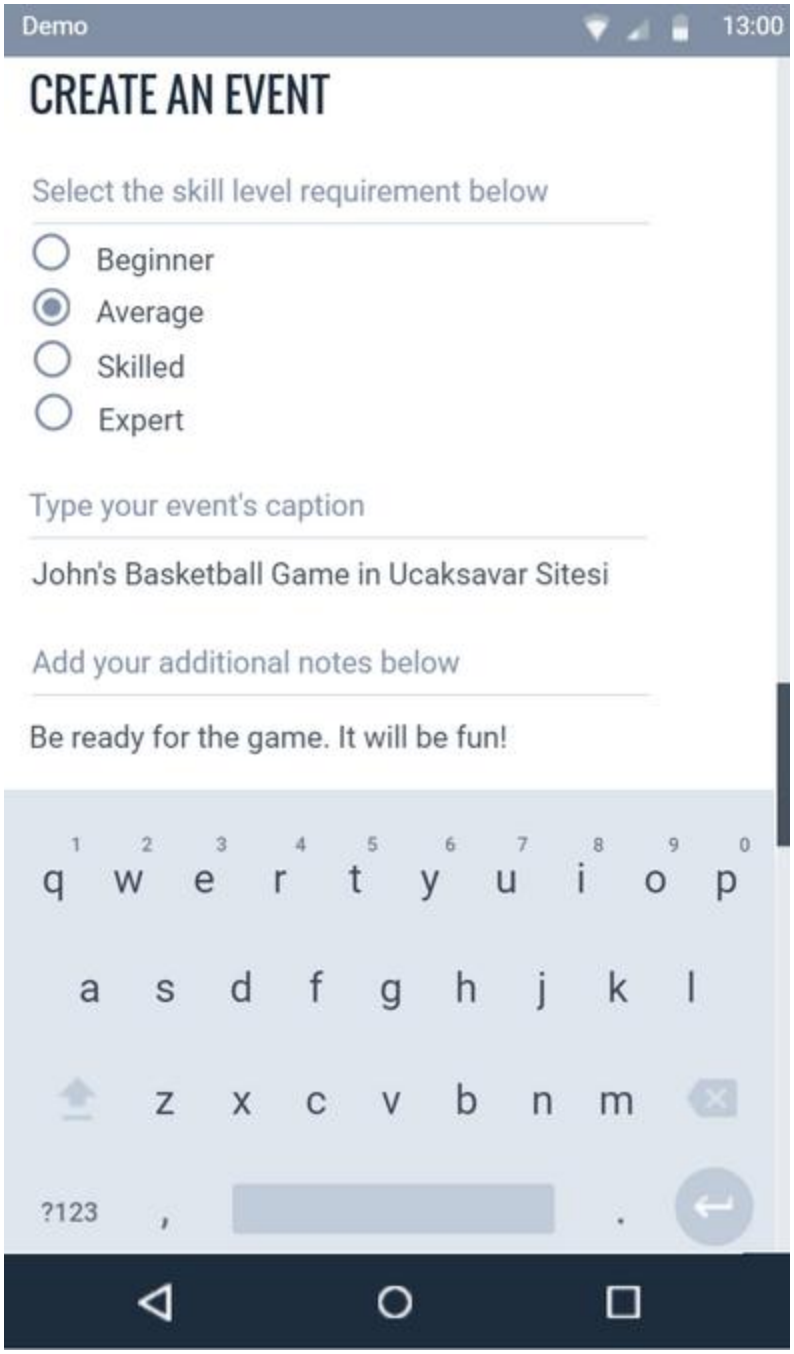
4.6 Chooses the badges which will be given to players after the game ends.

4.7 Types required items.

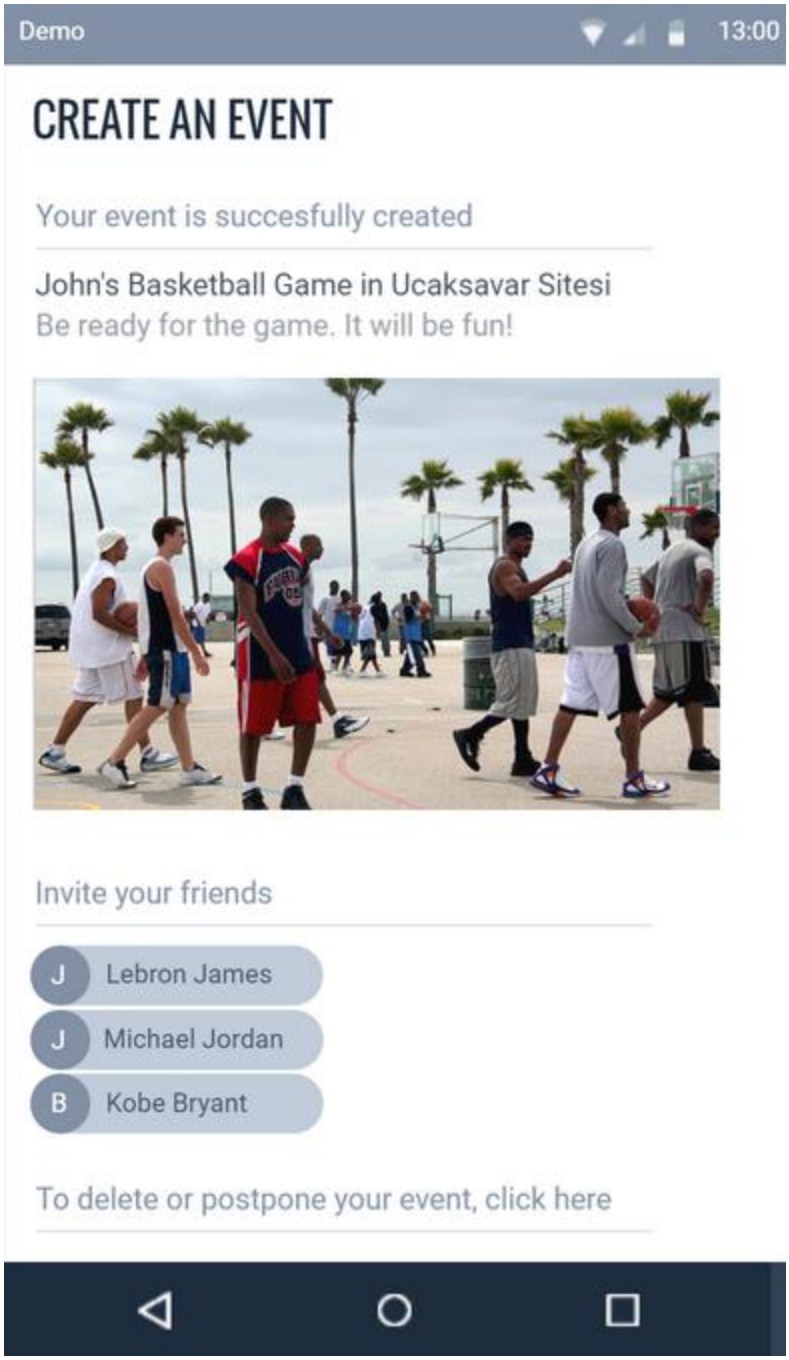
4.8 Selects the skill requirements for this game. He selects average for this game.

4.9 Lastly, He adds her event's caption and additional notes which will be visible by other users.


Demo
13:00
CREATE AN EVENT
Type your event rules below
The game field is 7x7 meters
The team which scores over 10 will be the winner
Pants or jeans are not allowed
Choose the reward badges
Bronze
Silver
Gold
Platinium
Type the required items below
1 Basketball
5 T-shirts (Black)
5 T-shirts (Yellow)
0 Type here



5. Now he sees the screen of "Your event is successfully created."



Scenario 3 & Mockup

Ali Öztürk

Persona

- *Ali Öztürk*
- *19 years old, male*
- *Freshman in computer engineering, recently moved to İstanbul from Antalya due to his university*
- *Wants to make new friends in this foreign city and continue participating sport events of his interest*

User Story

Ali Öztürk is a 19 years old freshman in computer engineering. He is very well aware of the benefits that technology provides to humans. He likes to meet new people from this app where people with similar sport interest can meet and engage in a fun activity with just couple of clicks. He is not professionally practicing a sport but likes playing basketball, football, hiking and mountain climbing occasionally. He has been using this

app for a year in Antalya to expand his social circle and to do something healthy. He recently moved to İstanbul to pursue his higher education. He is now in a city where he doesn't know anyone thus feels kind of alone. He decides to use this application to get used to this city by meeting new people of similar age and interest and also doing something fun. He has always wanted to see Kadıköy and he believes that it would be easy to find a basketball game so he prefers a basketball game that will take place in Kadıköy.

Goals

- *Discovering a sports event of his interest, in Kadıköy so that he will be able to engage in the sport he would like with new friends who are also interested in the same type of sport.*

Preconditions

- *He already has an account in the application. He has been participating in sports events using this application back in Antalya to meet new people and have a healthy habit.*
- *He has already signed in to the application so when he clicks the application, the home page shows up.*

Scenario

1. *Ali clicks the application then the home page is shown on the screen.*
2. *He clicks the event search page to discover events.*
3. *On the search page, there is a button to go to map to search events and also a search bar to type the keywords of a requested event. Ali uses search bar and types basketball.*
4. *List of basketball events from random locations are listed.*
5. *He filters the posts to find better results. He wants to find a basketball game that will occur in Kadıköy at this weekend.*
6. *Filtered results are listed on the page. He clicks the first event post that occurs on the page.*

Acceptance Criteria

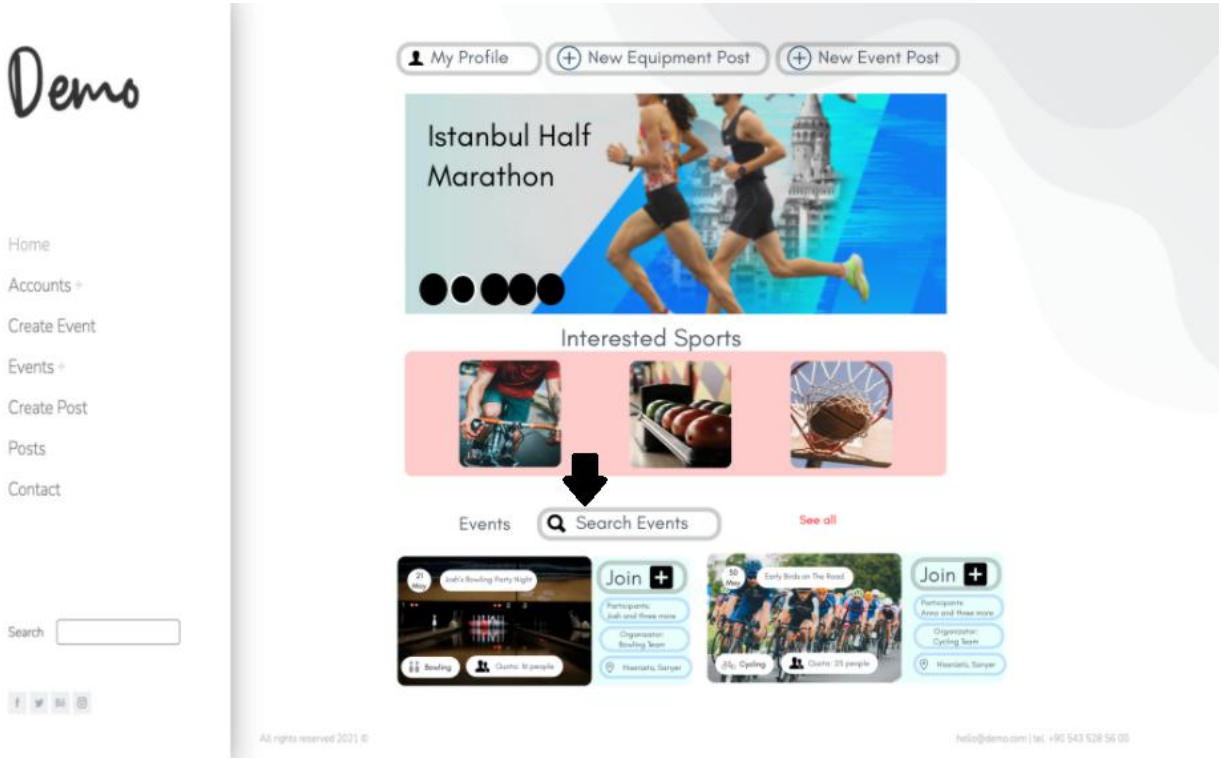
- *1.2.5.1.1 The system shall allow users to be able to search for events according to a name, location or type of a sport.*
- *1.2.5.1.2 Event search can be done by typing the keywords on a search bar.*
- *1.2.5.1.3 Event feed shall be filterable according to the specifications stated in date, location, game rate, incomplete number of participants in event.*
- *1.2.3.1.4 The following information about the event shall be shown:*

    - *1.2.3.1.4.1 event's date and time*
    - *1.2.3.1.4.2 location*
    - *1.2.3.1.4.3 sport type*
    - *1.2.3.1.4.7 status (upcoming, live or passed / full or open to application)*
    - *1.2.3.1.4.8 name of the event*

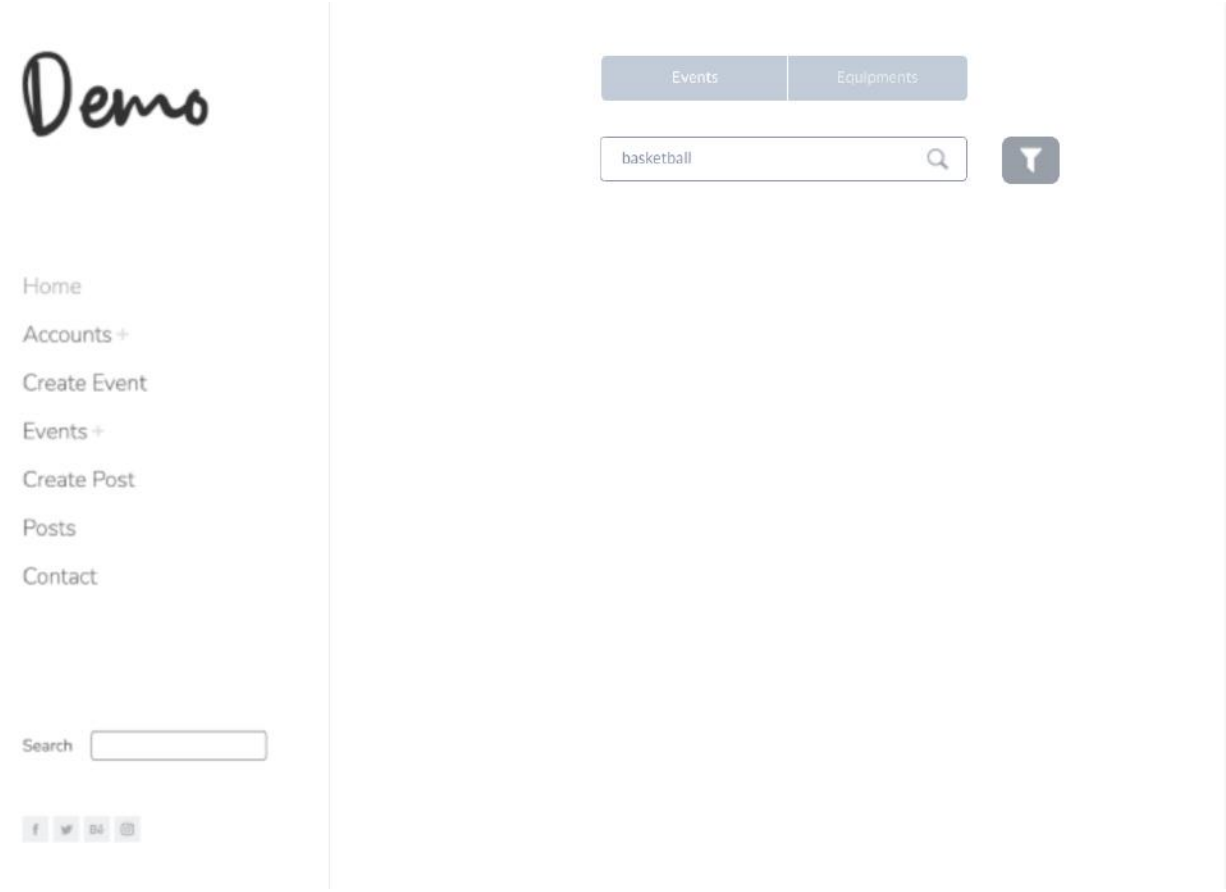
Mockup

Web

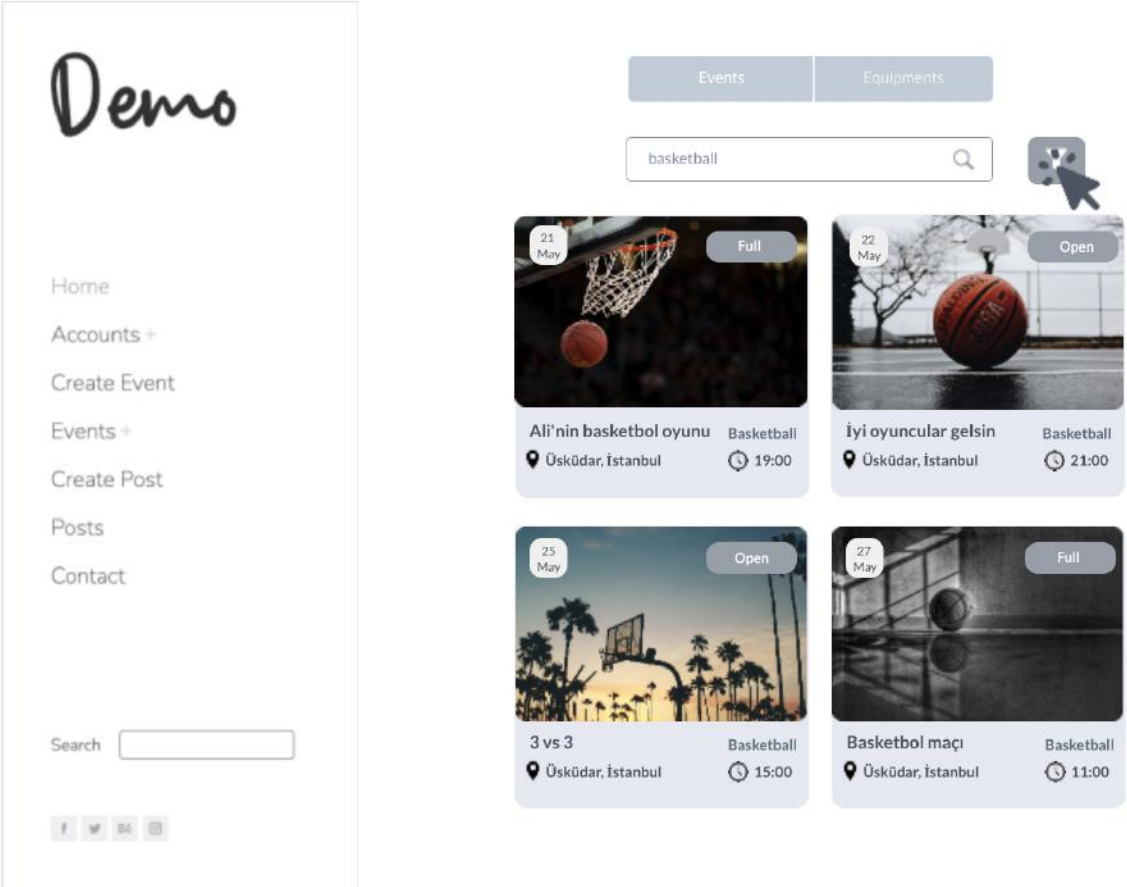
1. *Ali clicks the application then the home page is shown on the screen.*
2. *He clicks the event search page to discover events.*



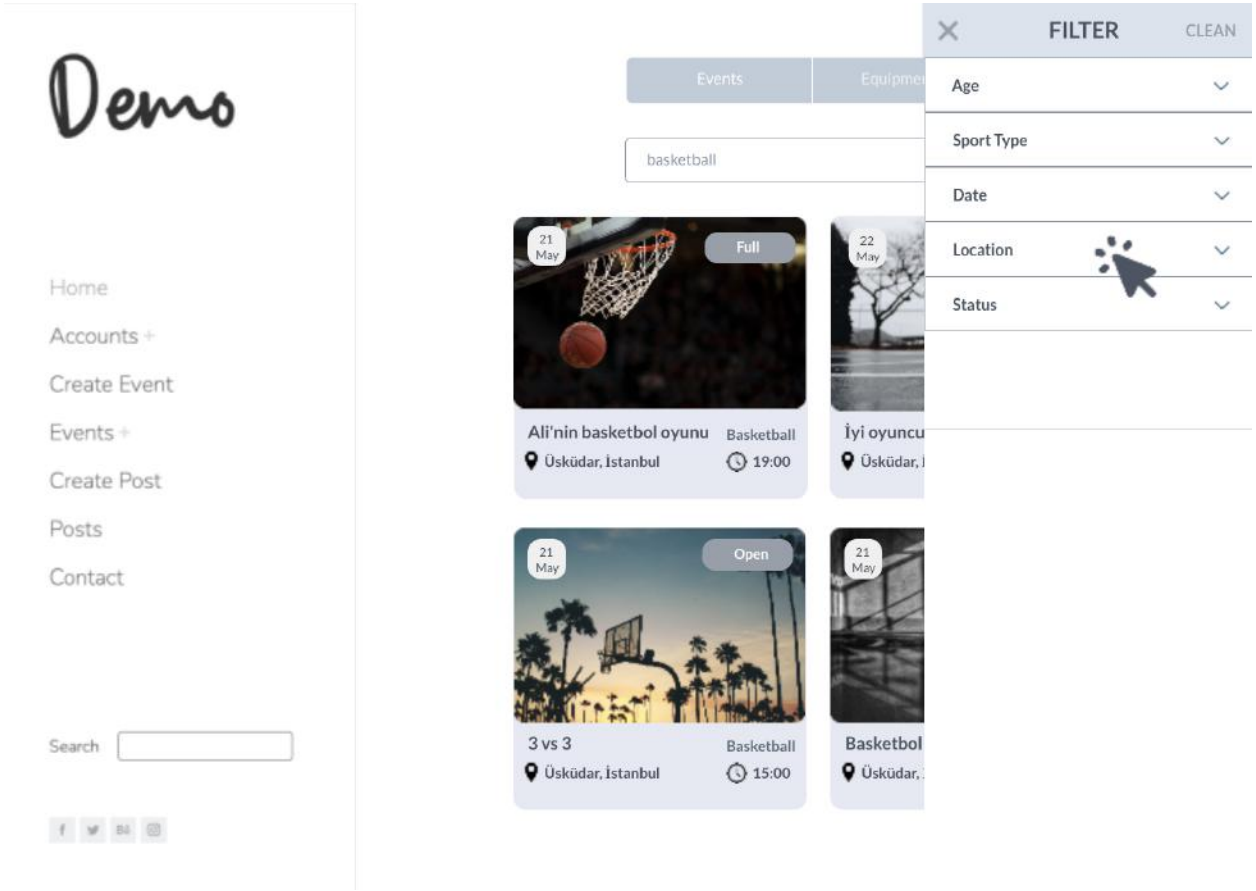
3. *On the search page, there is a button to go to map to search events and also a search bar to type the keywords of a requested event. Ali uses search bar and types basketball.*

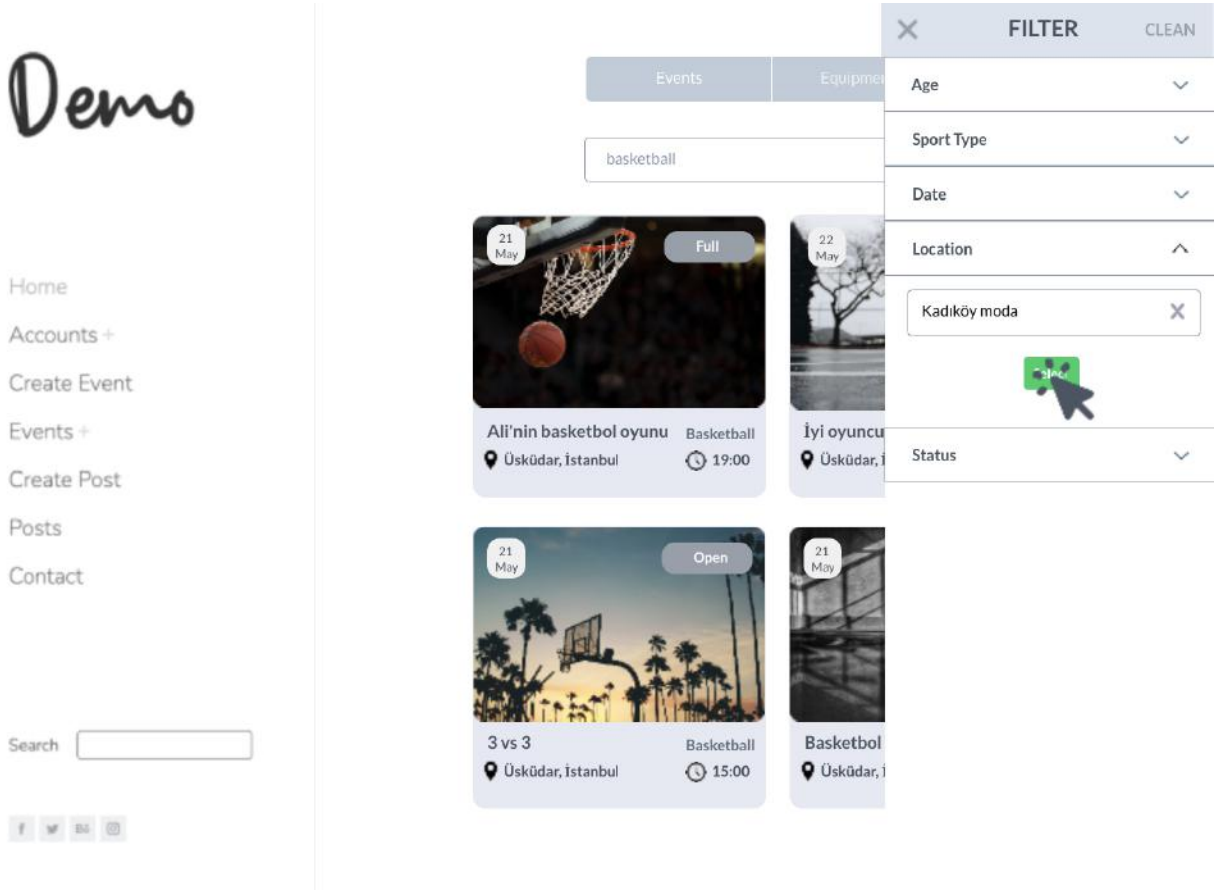


4. *List of basketball events from random locations are listed.*

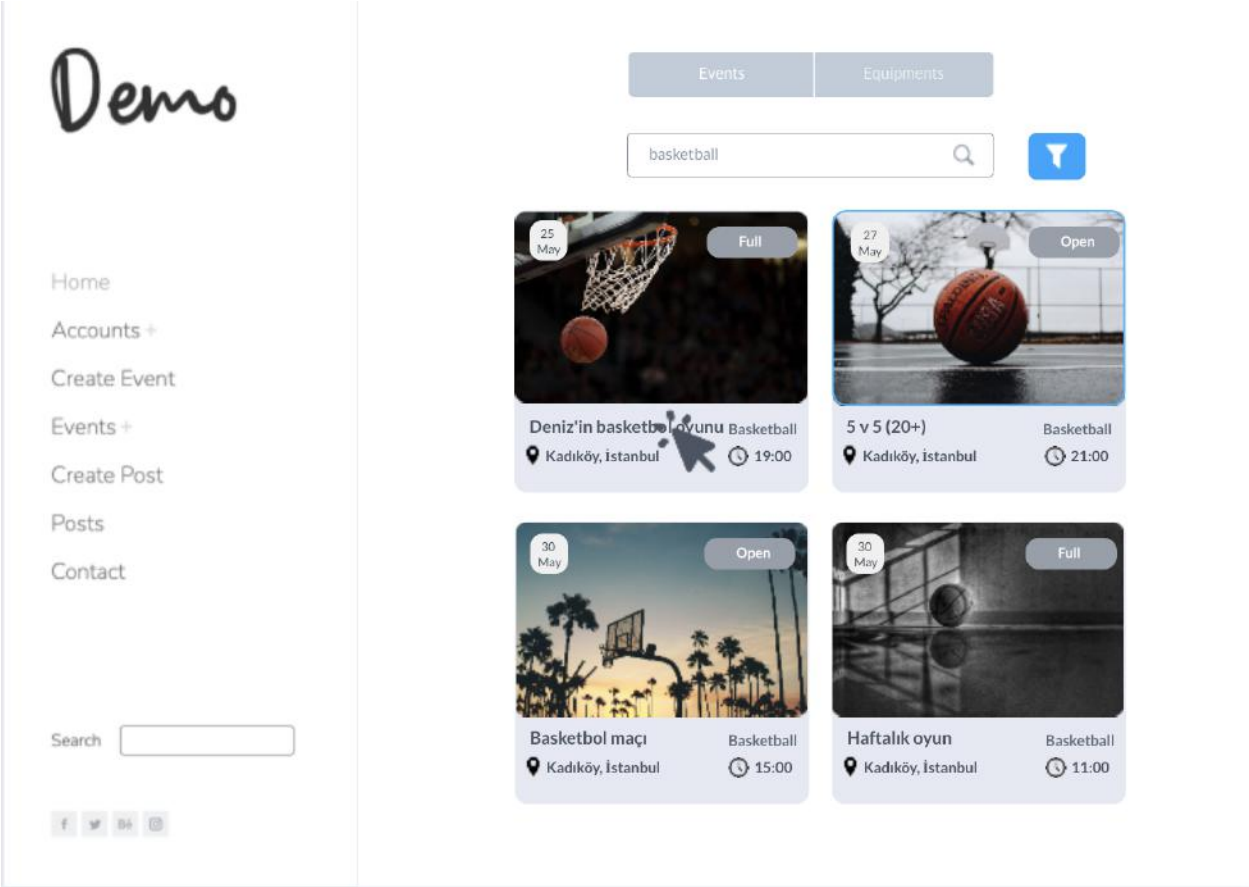


5. *He filters the posts to find better results. He wants to find a basketball game that will occur in Kadıköy at this weekend.*


Demo
Home
Accounts
Create Event
Events
Create Post
Posts
Contact
Search
Events
basketball
FILTER
CLEAN
Age
Sport Type
Date
Location
Status
21 May
Full
Ali'nin basketbol oyunu
Basketball
Üsküdar, İstanbul
19:00
22 May
İyi oyuncu
21 May
Open
3 vs 3
Basketball
Üsküdar, İstanbul
15:00
21 May
Basketbol

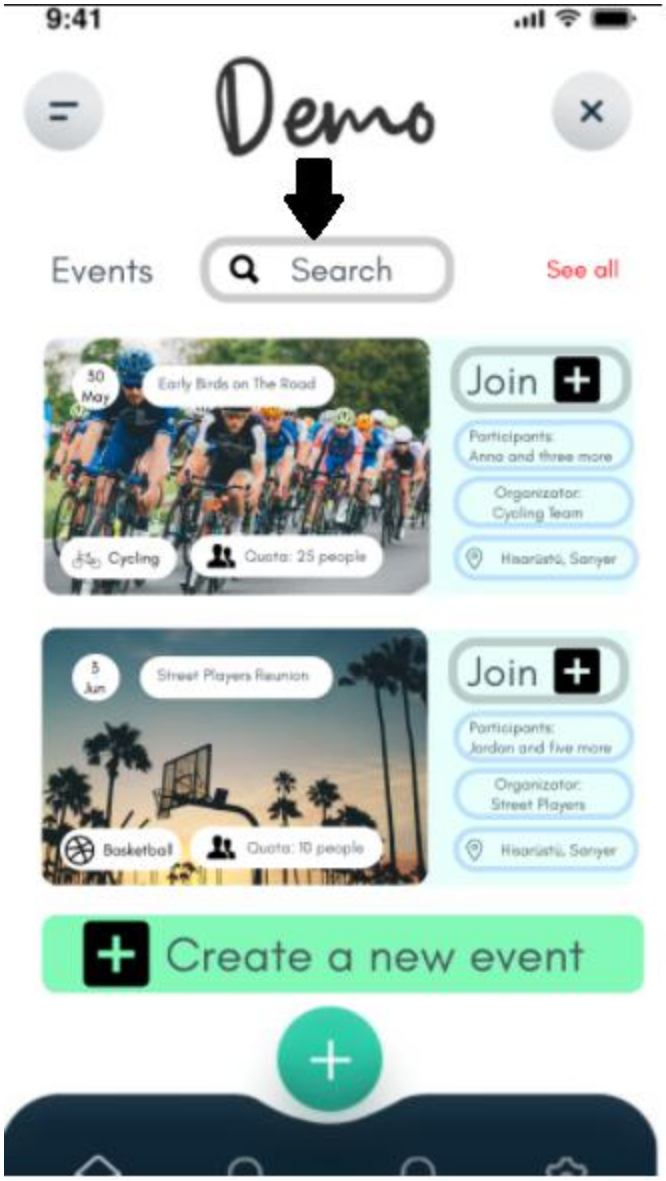


6. *Filtered results are listed on the page. He clicks the first event post that occurs on the page.*

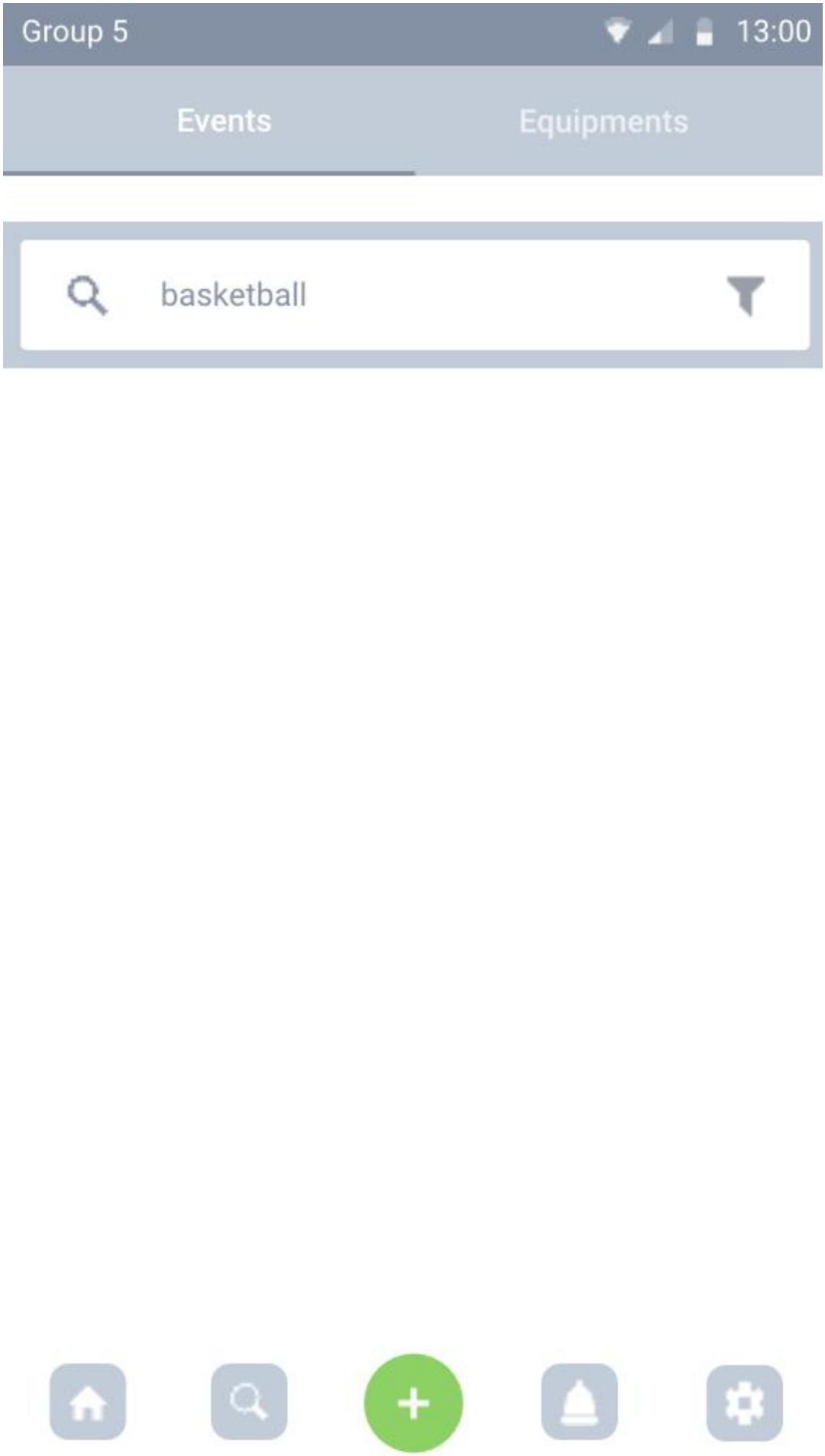


Android

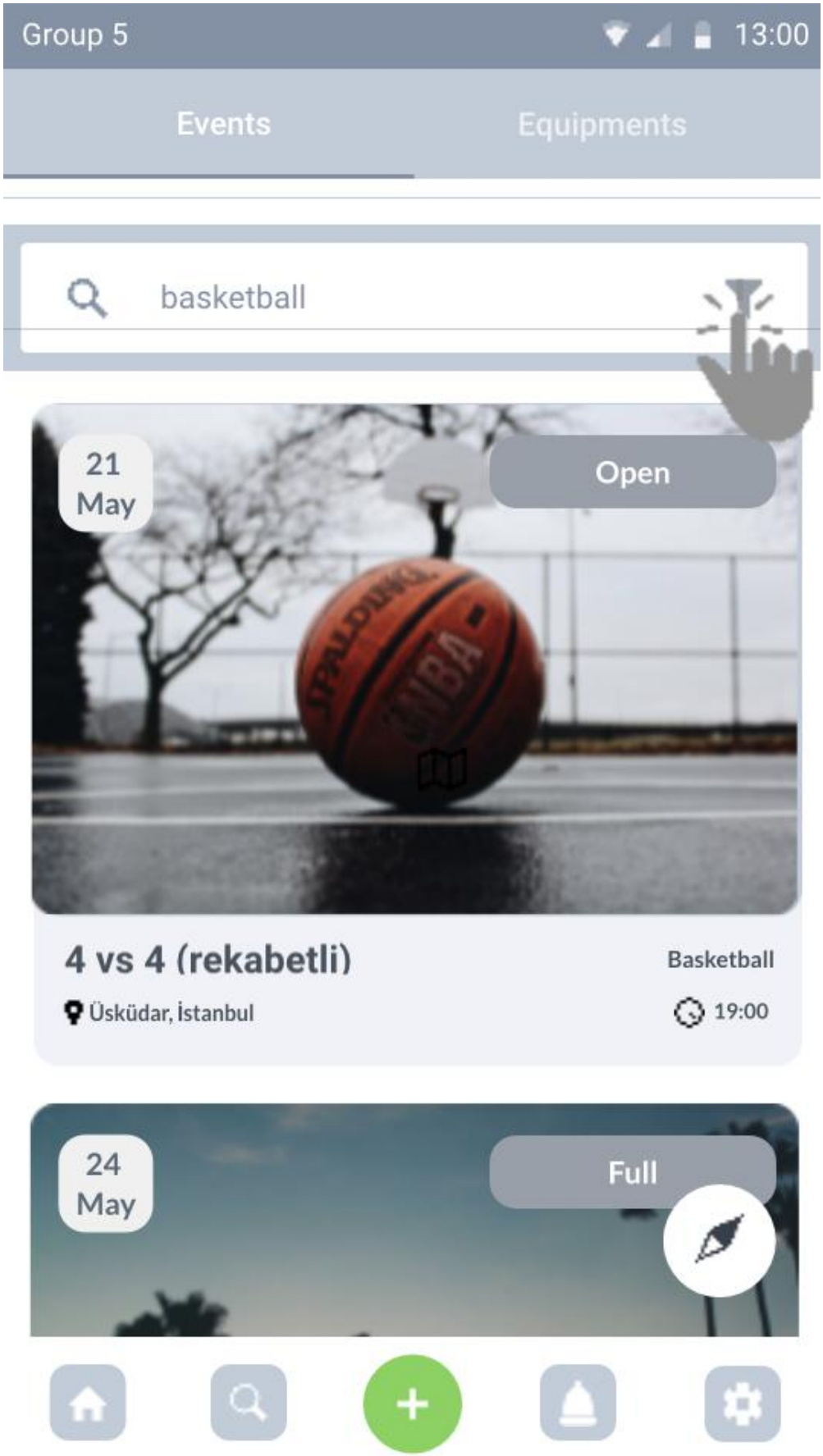
1. *Ali clicks the application then the home page is shown on the screen.*
2. *He clicks the event search page to discover events.*



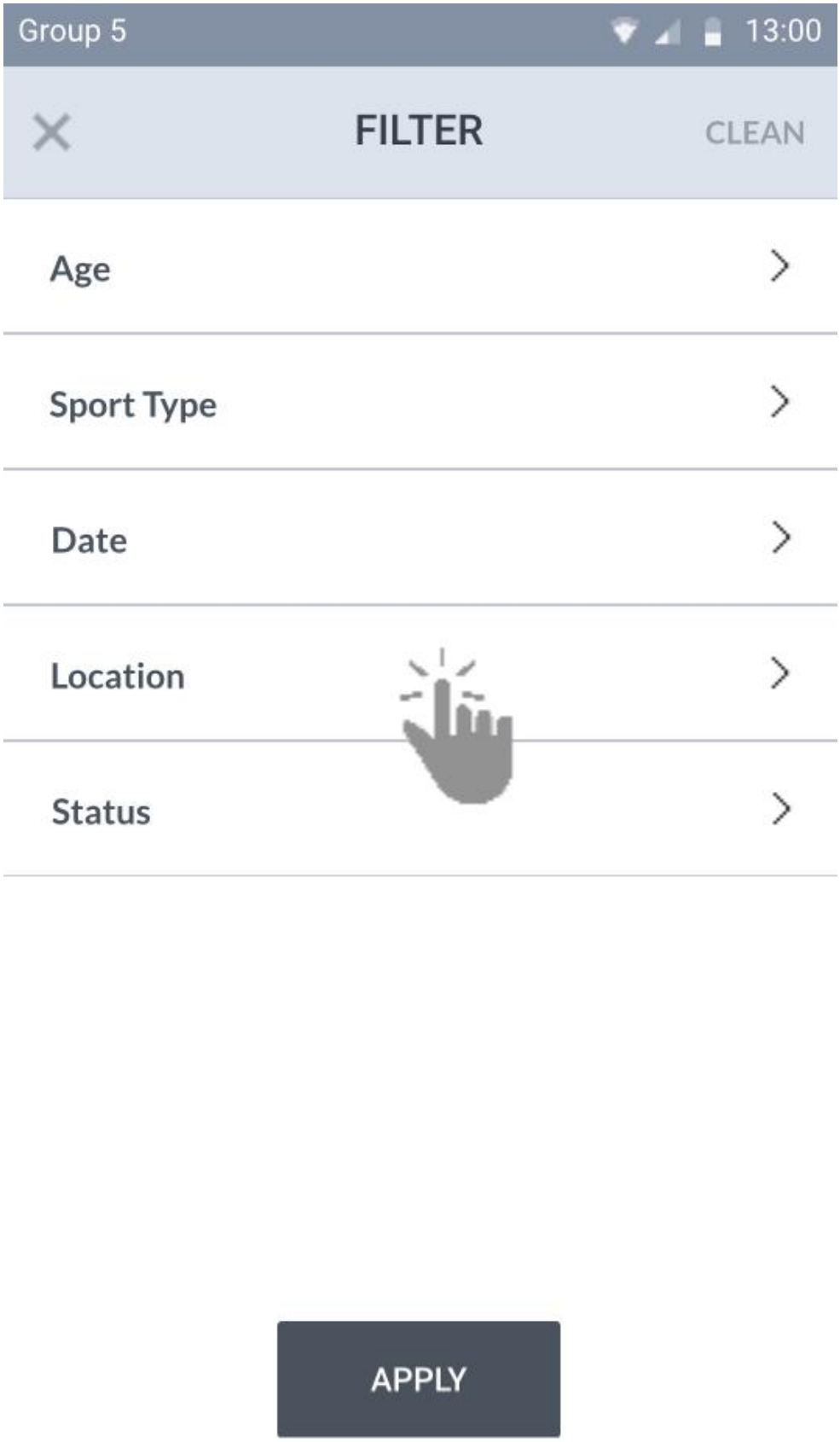
3. *On the search page, there is a button to go to map to search events and also a search bar to type the keywords of a requested event. Ali uses search bar and types basketball.*

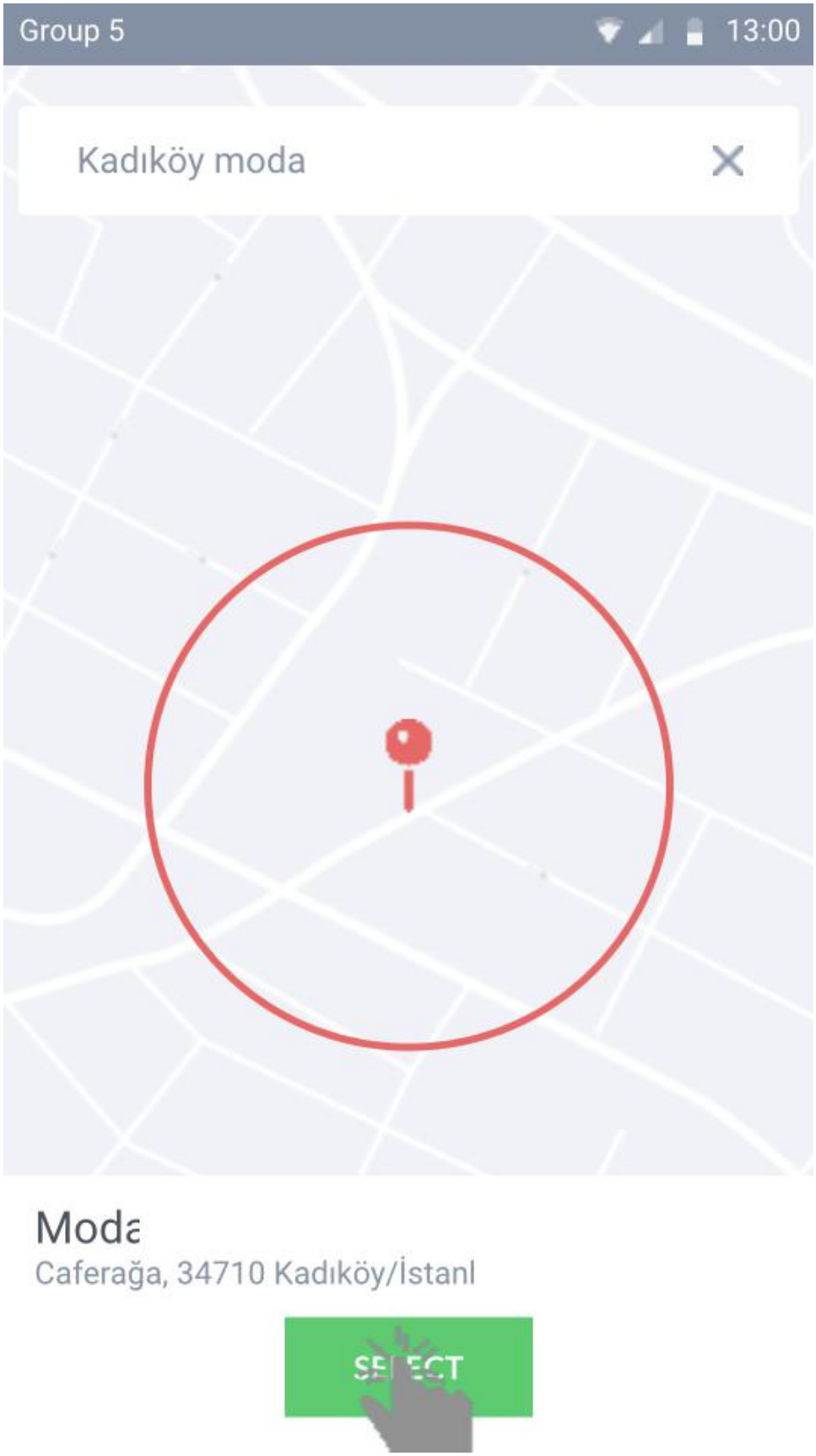

Group 5
13:00
Events
Equipments
basketball

4. *List of basketball events from random locations are listed.*


Group 5
13:00
Events
Equipments
basketball
21
May
Open
4 vs 4 (rekabetli)
Basketball
Üsküdar, İstanbul
19:00
24
May
Full

5. *He filters the posts to find better results. He wants to find a basketball game that will occur in Kadıköy at this weekend.*


Group 5
13:00
FILTER
CLEAN
Age
Sport Type
Date
Location
Status
APPLY


Group 5
13:00
Kadıköy moda
Moda
Caferağa, 34710 Kadıköy/İstanl
SELECT

Group 5 13:00

FILTER

CLEAN

Age

Sport Type

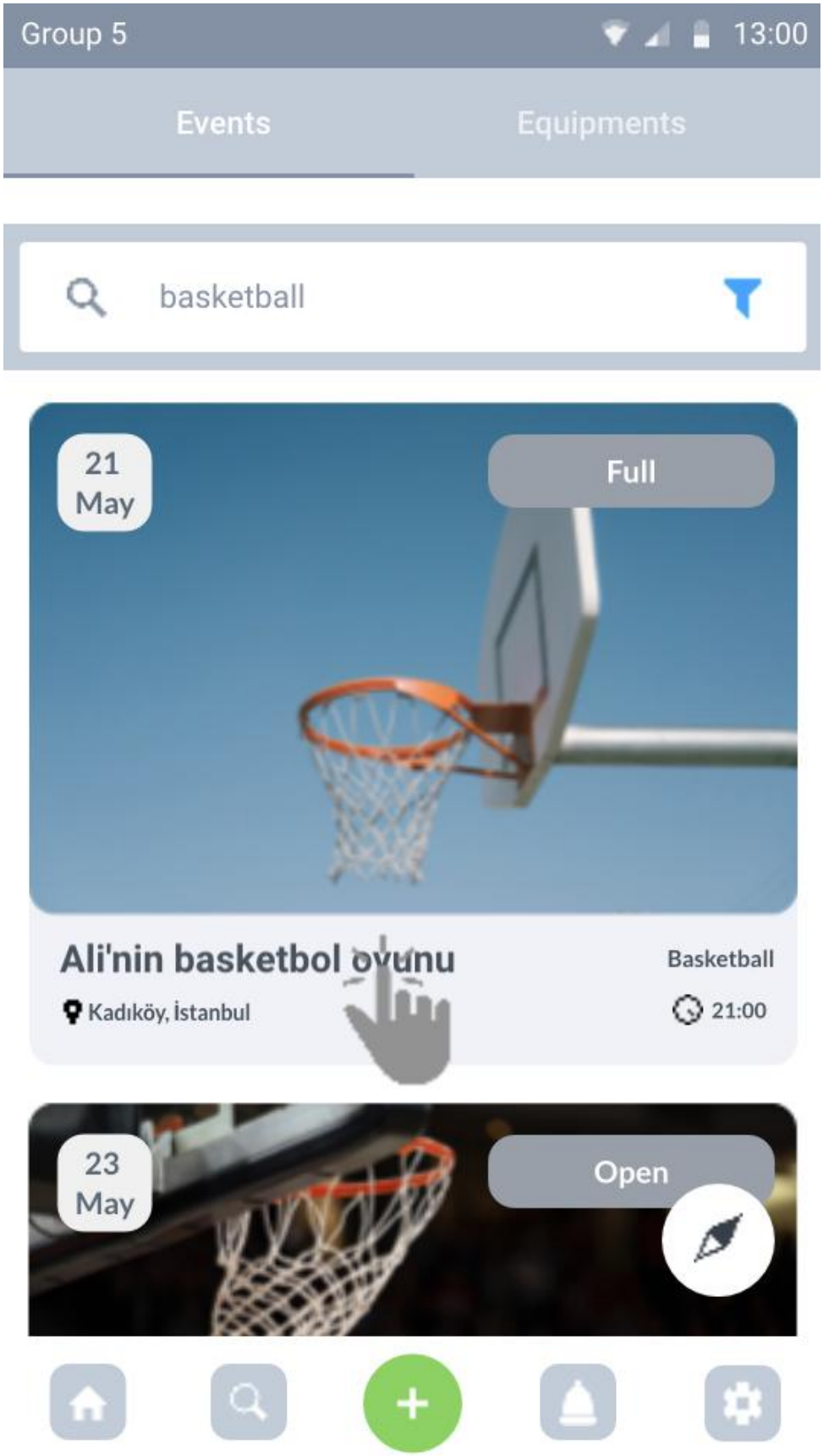
Date

Location (Kadıköy)

Status

APPLY

6. *Filtered results are listed on the page. He clicks the first event post that occurs on the page.*


Group 5
13:00
Events
Equipments
basketball
21
May
Full
Ali'nin basketbol oyunu
Basketball
Kadıköy, İstanbul
21:00
23
May
Open

Scenario 4 & Mockup

Uncle Drew

Persona

- *Uncle Drew*
- *50 years old, male*
- *Street ball legend*

User Story

Uncle Drew is a 50 years old street ball legend. He was a star when he was young, then quit playing. Now, he wants to play again and has created an event for that.

Goals

- *He wants to see the events he has created.*
- *He wants to select the participants for the basketball event.*
- *He wants to view the analytics about the participants*

Pre-Conditions

- *He already has an account.*
- *He has already signed in to the system.*
- *He has already created an event.*
- *There are some applicants to the event.*

Scenario

1. *From his profile page, he goes to My Events tab. He can view the events he has created so far in this page.*
2. *He clicks the View Details button of the event called "Drew's Streetball".*
3. *He views the event details here. There is nothing he wishes to change but he scrolls down to check if there are any problems. (There is no scroll for web, all details are shown in a single page)*
4. *He wants to see the participants. So he clicks the View Participants button.*
5. *He is in the Participants page. There are no selected participants yet and there are 3 applicants. The applicants are listed with their profile pictures, names and usernames. He sees that all 3 applicants are his old friends. He starts approving.*
6. *As he approves the applicants, the participant counts in the above bar change. Also, the accepted participants profile page border turns green and their event participation status below their username is updated.*

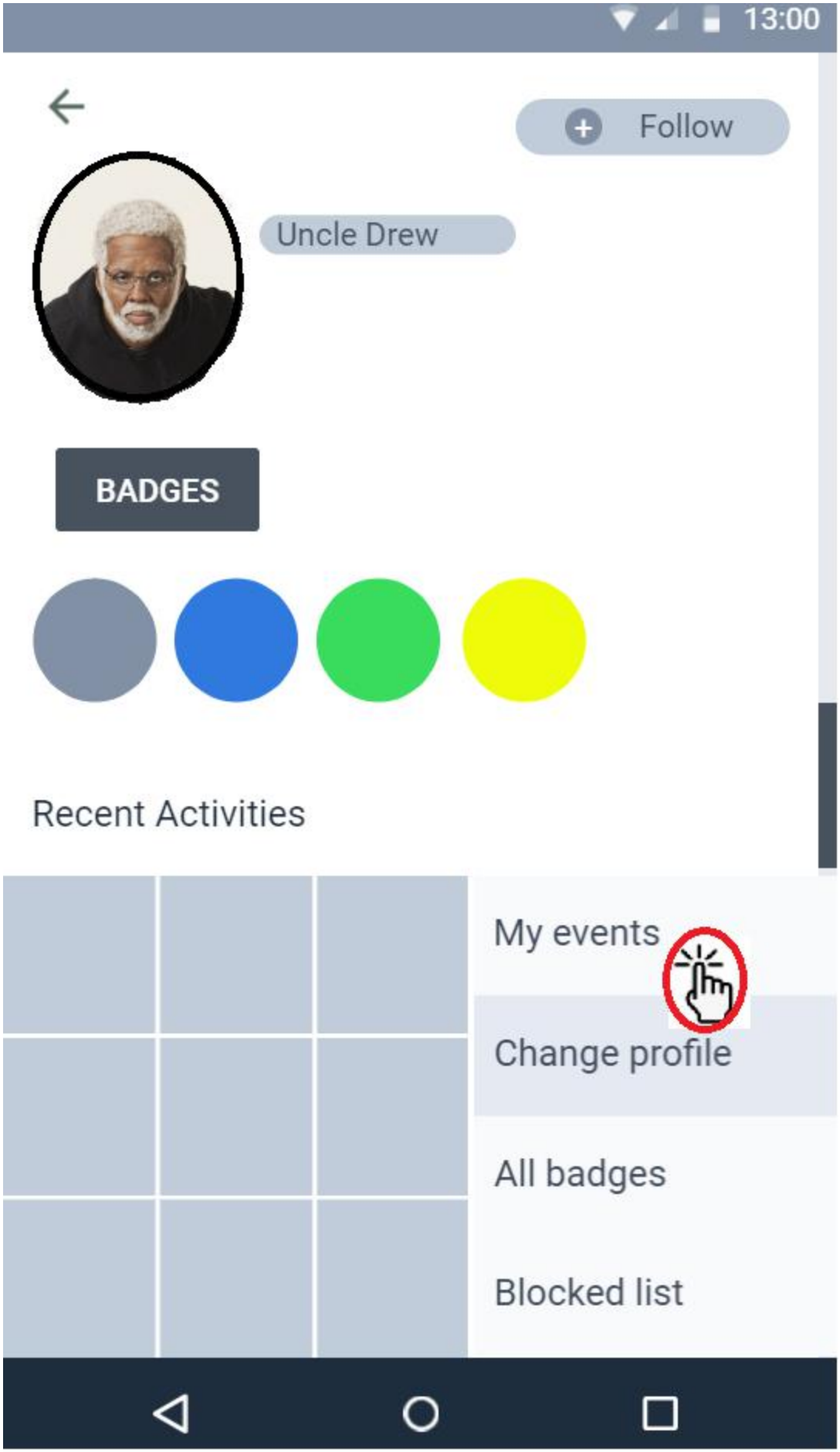
7. *He continues to approve. For the approved applicants, the approve button changes to withdraw, which allows the event creator to take the user back to waiting state.*
8. *He finishes approving all his 3 old friends' requests. He wonders the participant analytics as he haven't heard from them for a while. He clicks the View Participant Analytics button.*
9. *He views the statistics about the participants of the event in this page.*
10. *With the joy of reuniting with his old street ball gang, he quits the app (or the web page).*
11. *He goes to the Play Store and gives 5 starts, leaves a thankful comment.*

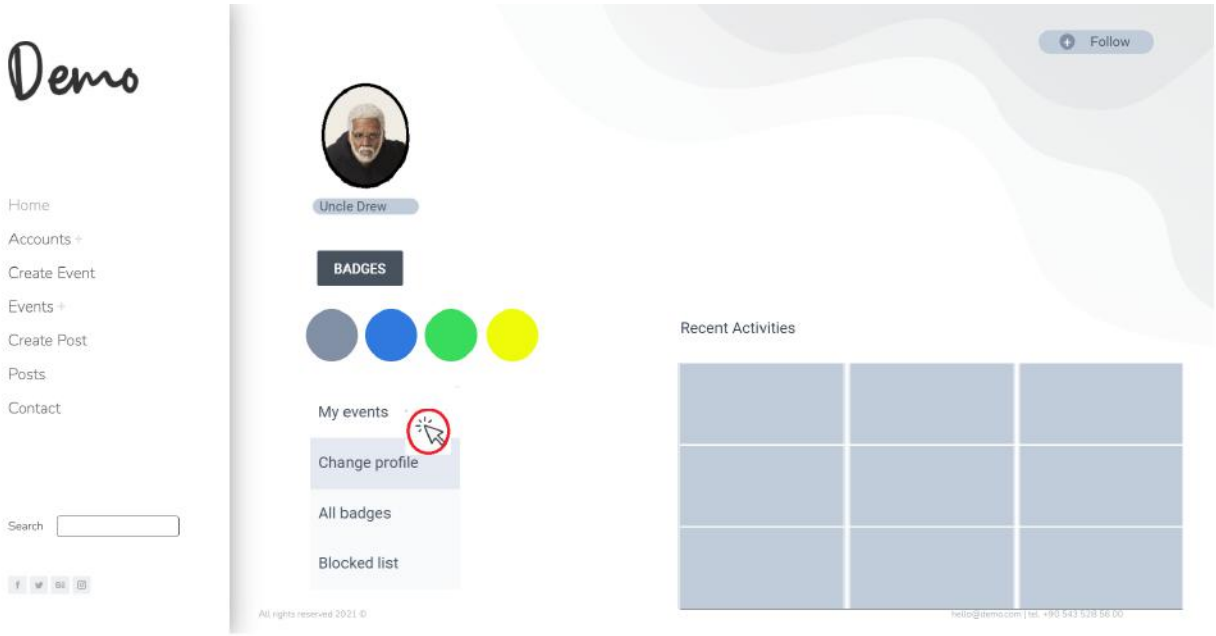
Acceptance Criteria

- *1.1.1.11 Coming events the user created or will be participated shall be seen on the profile page.*
- *1.1.2.12 Users shall be able to see the limit to the number of people that can participate in an event, name of the event, equipment required for an event, day-time, detailed location information, rules, skill requirement, status, badges that might be given at the end of the event and description of an event. Event creators can edit these information.*
- *1.1.2.14 Event creators shall choose among the applicants as participants of the event. Event creator shall be free to choose between the candidates.*
- *1.1.2.17 Event creators shall be able to see the analytics of the event they created: Average age, distance, skill level, event count per year and event count per lifetime of the participants shall be shown.*

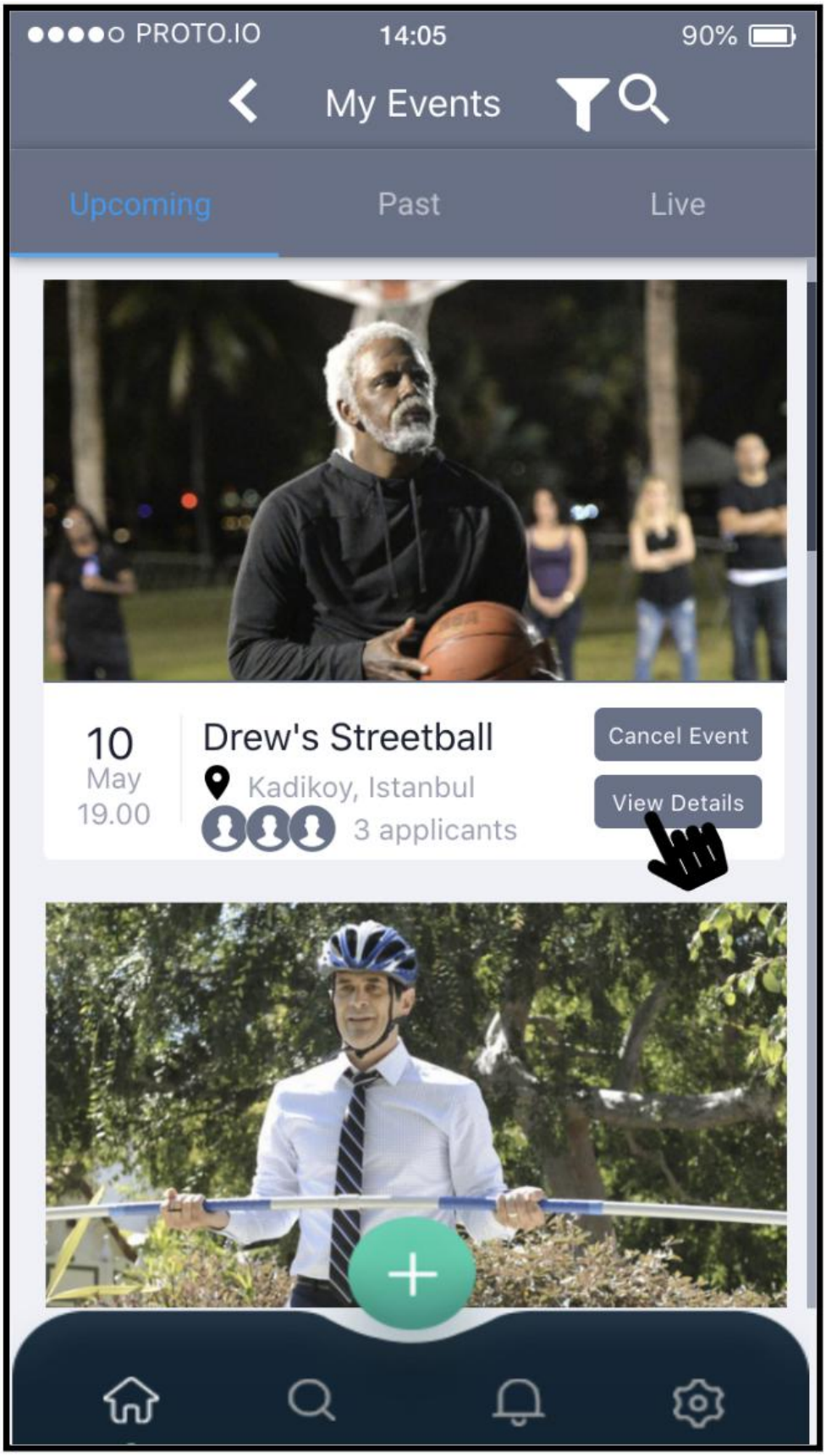
Mockup

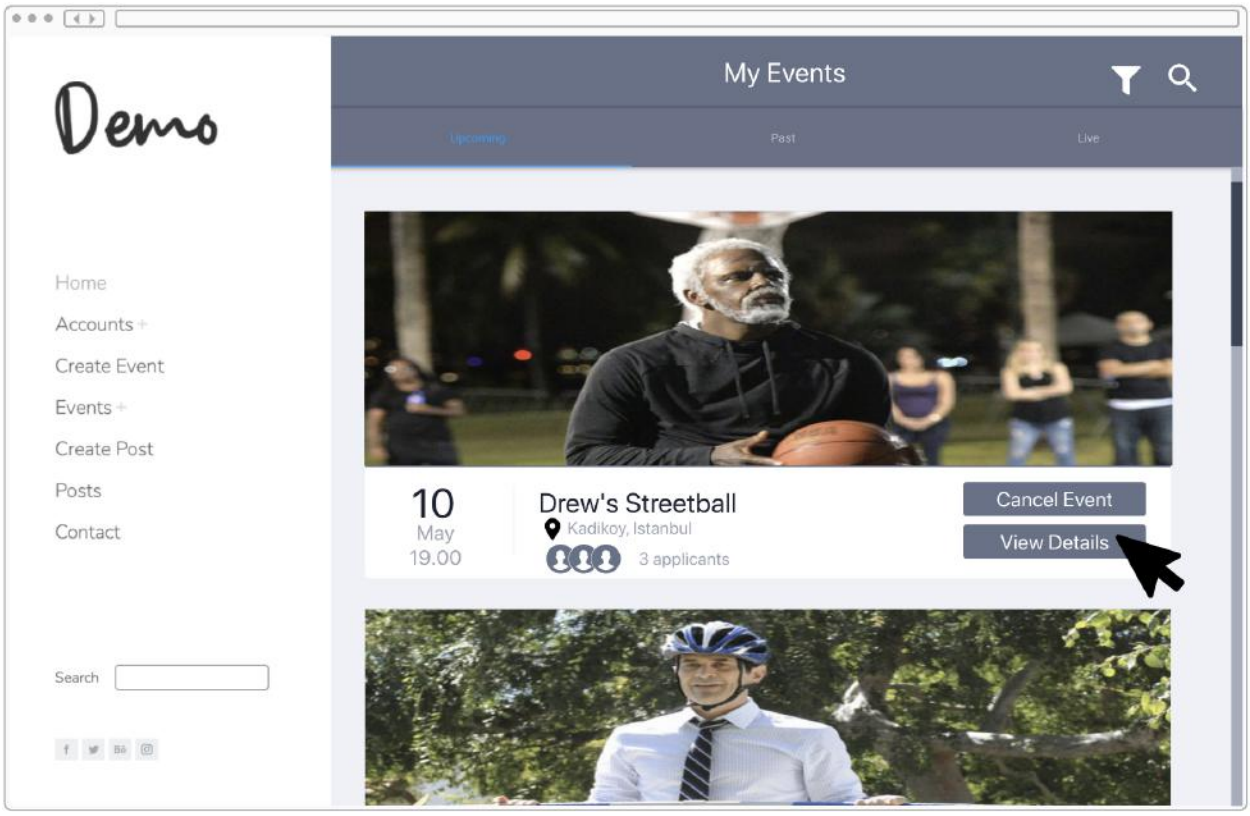
1. *From his profile page, he goes to My Events tab. He can view the events he has created so far in this page.*


13:00
Follow
Uncle Drew
BADGES
Recent Activities
My events
Change profile
All badges
Blocked list

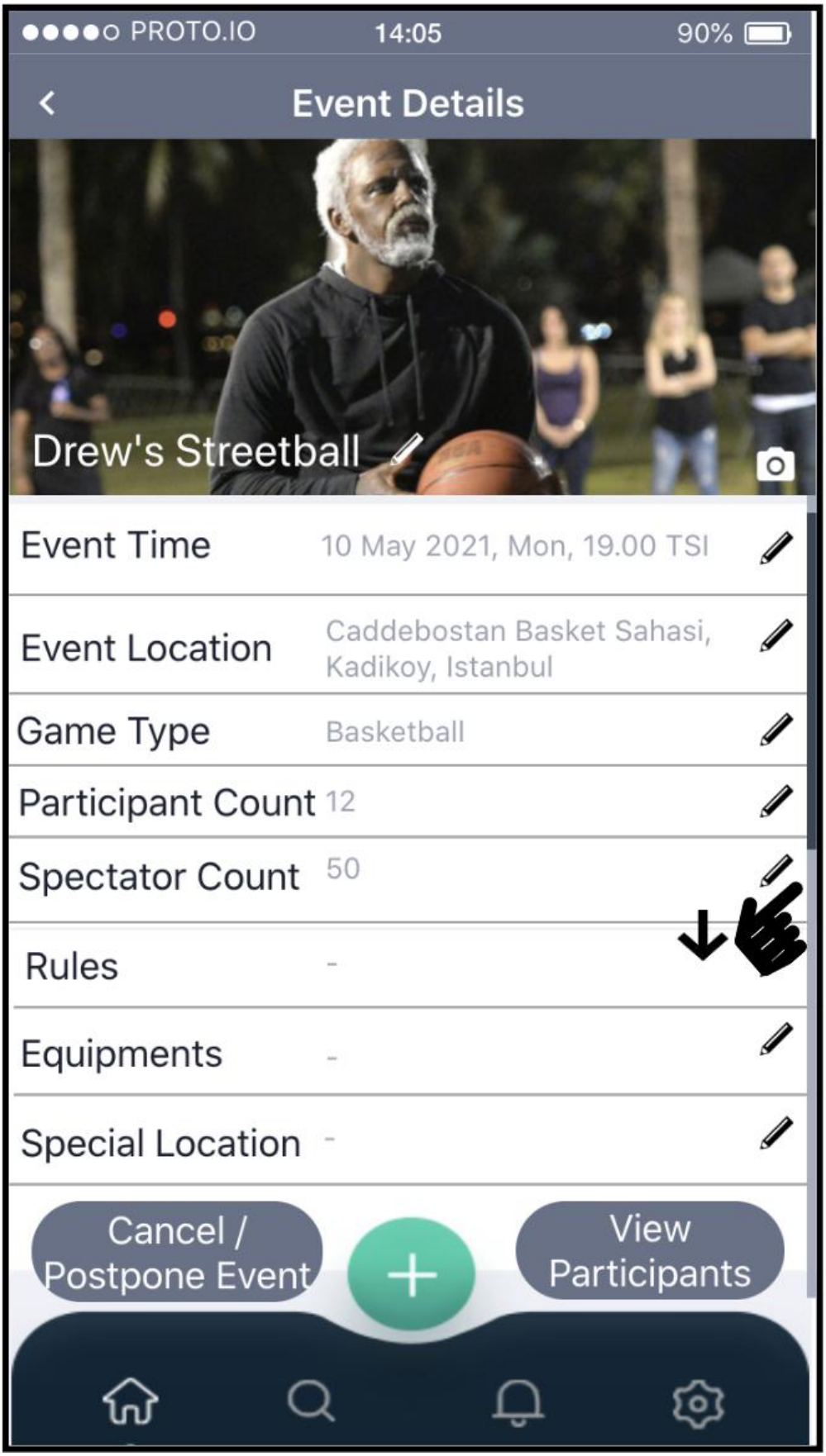


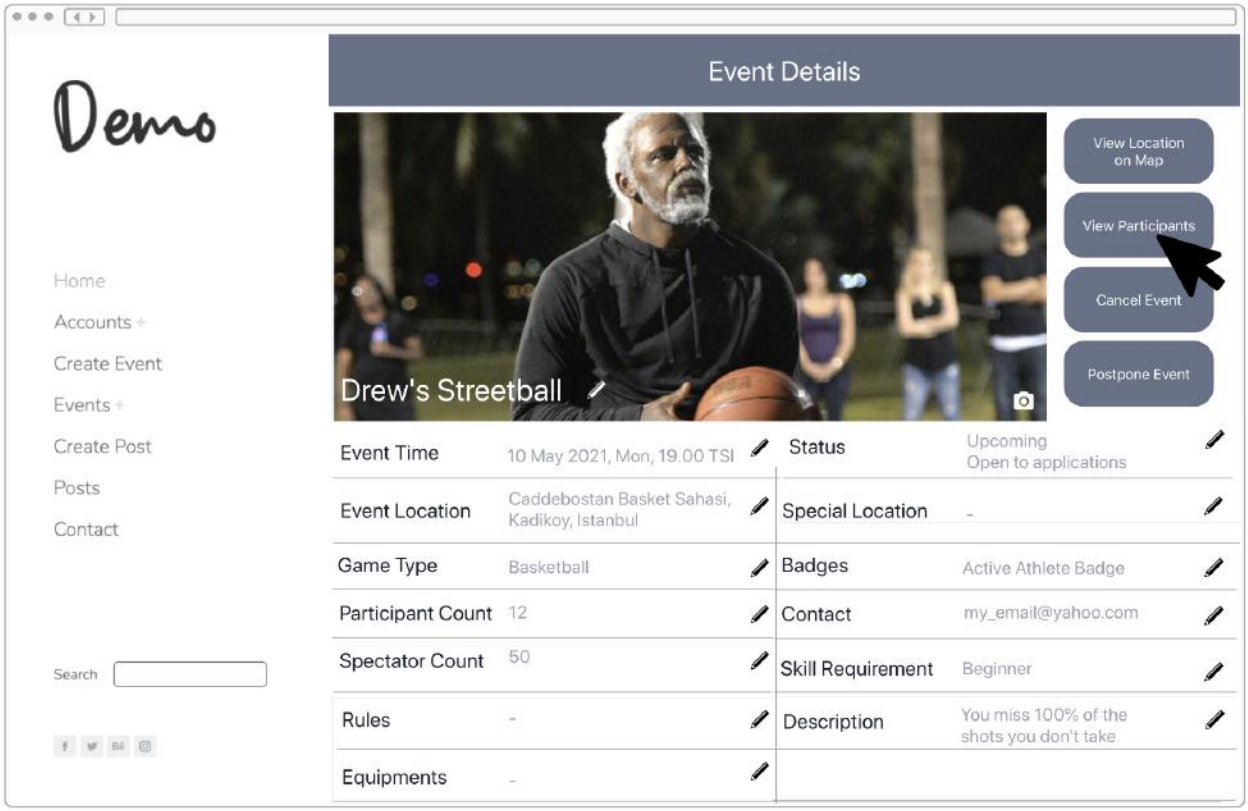
2. *He clicks the View Details button of the event called "Drew's Streetball".*


PROTO.IO
14:05
90%
My Events
Upcoming
Past
Live
10
May
19.00
Drew's Streetball
Kadikoy, Istanbul
3 applicants
Cancel Event
View Details

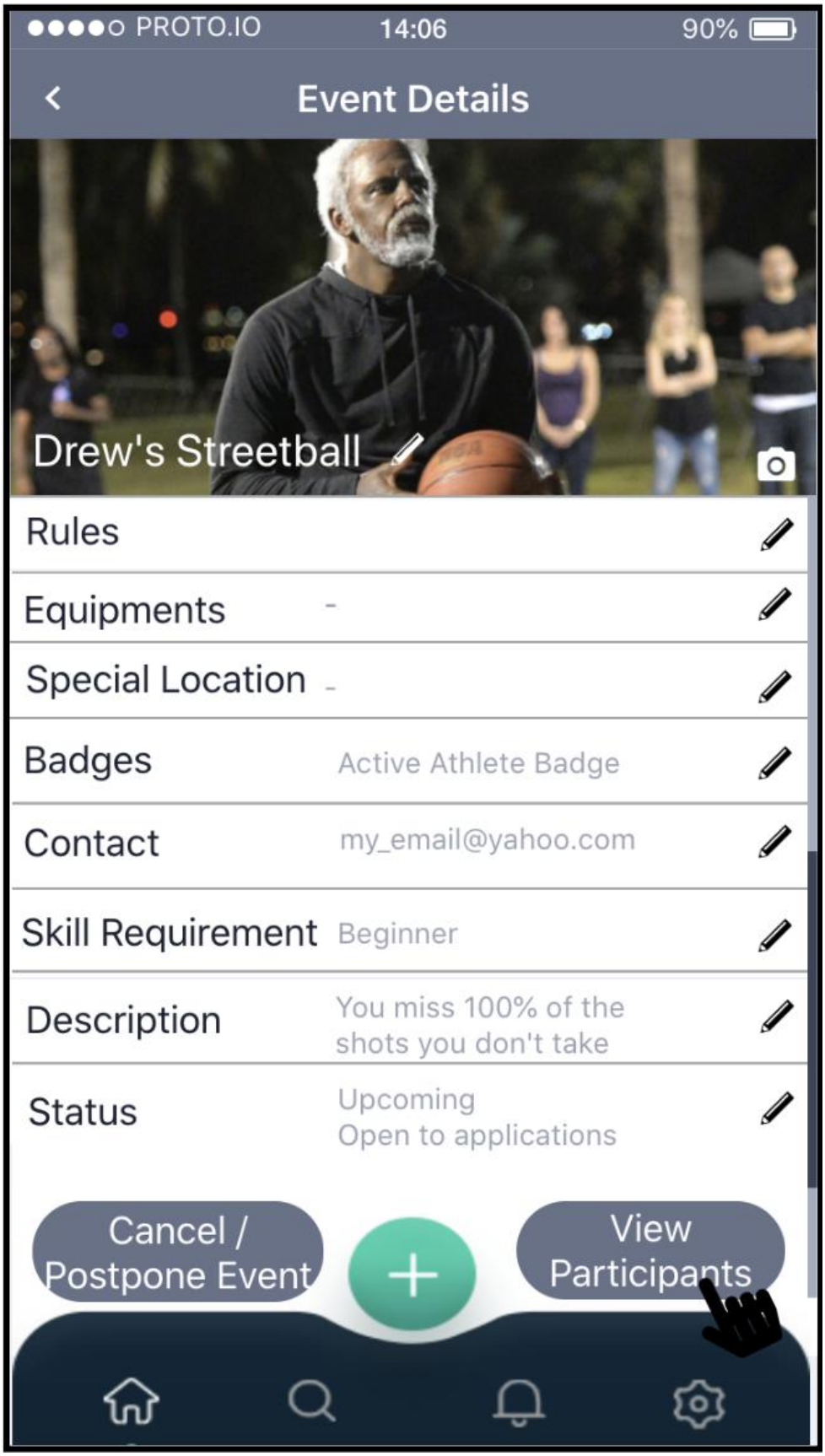


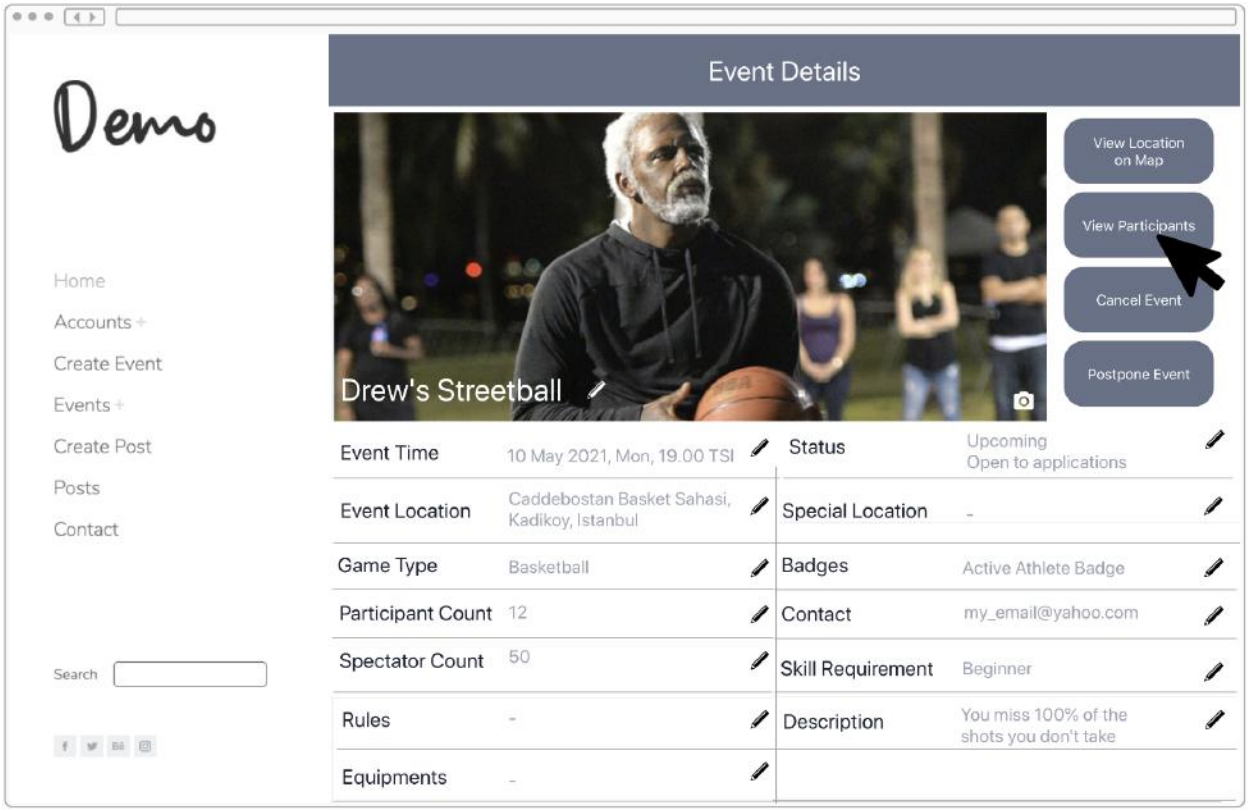
3. *He views the event details here. There is nothing he wishes to change but he scrolls down to check if there are any problems. (There is no scroll for web, all details are shown in a single page)*


PROTO.IO
14:05
90%
Event Details
Drew's Streetball
Event Time
10 May 2021, Mon, 19.00 TSI
Event Location
Caddebostan Basket Sahasi, Kadikoy, Istanbul
Game Type
Basketball
Participant Count
12
Spectator Count
50
Rules
-
Equipments
-
Special Location
-
Cancel / Postpone Event
View Participants

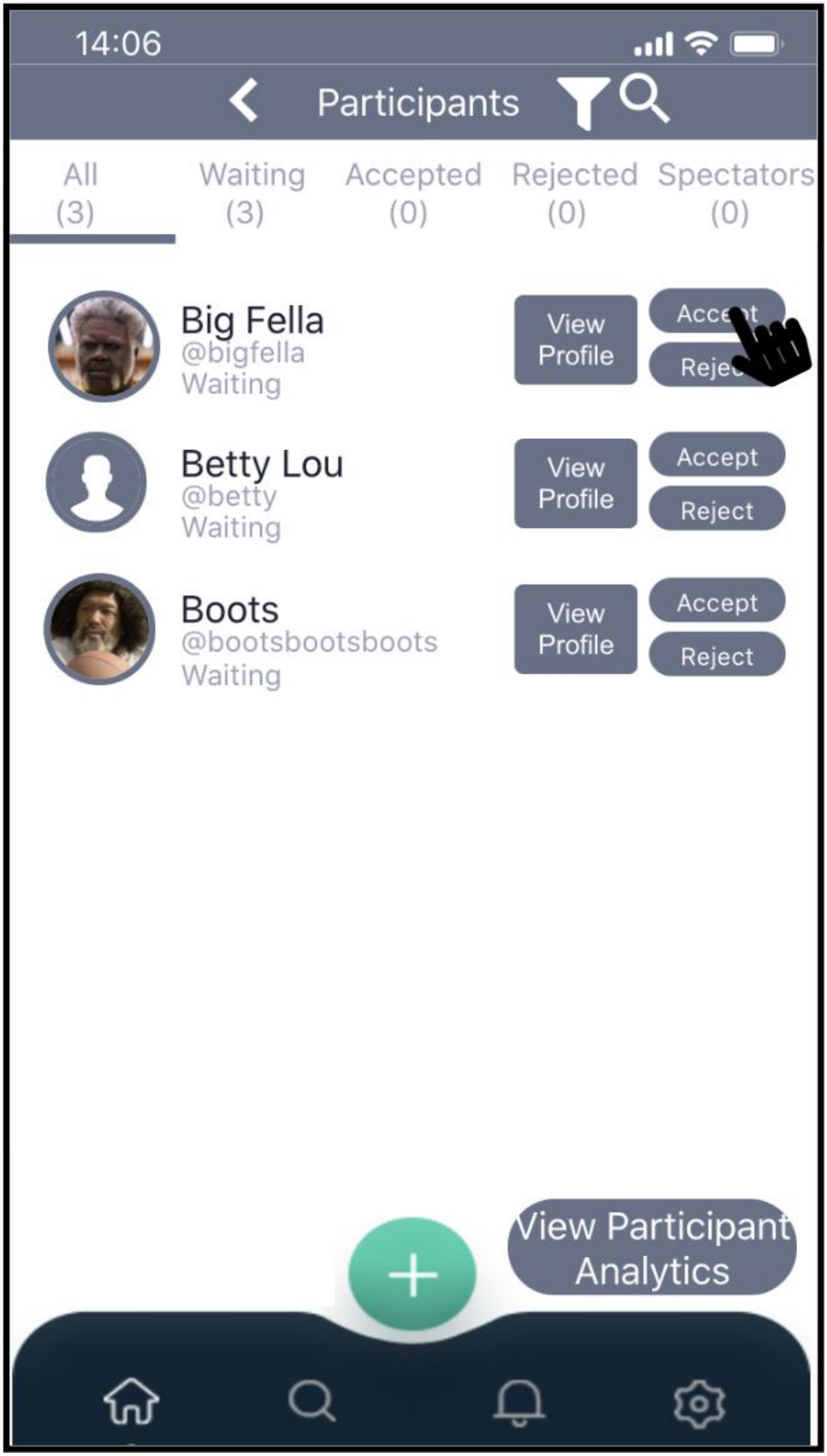


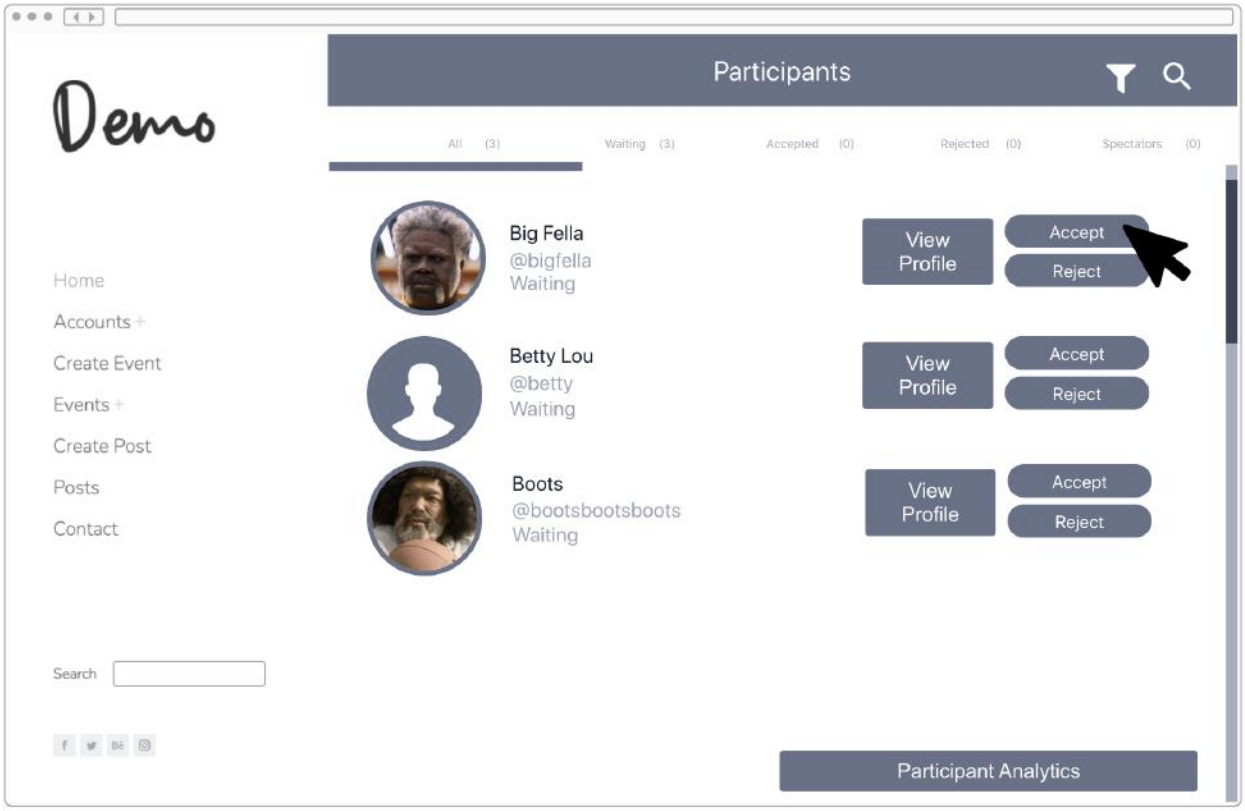
4. *He wants to see the participants. So he clicks the View Participants button.*


PROTO.IO
14:06
90%
Event Details
Drew's Streetball
Rules
Equipments
Special Location
Badges
Active Athlete Badge
Contact
my_email@yahoo.com
Skill Requirement
Beginner
Description
You miss 100% of the shots you don't take
Status
Upcoming
Open to applications
Cancel / Postpone Event
View Participants

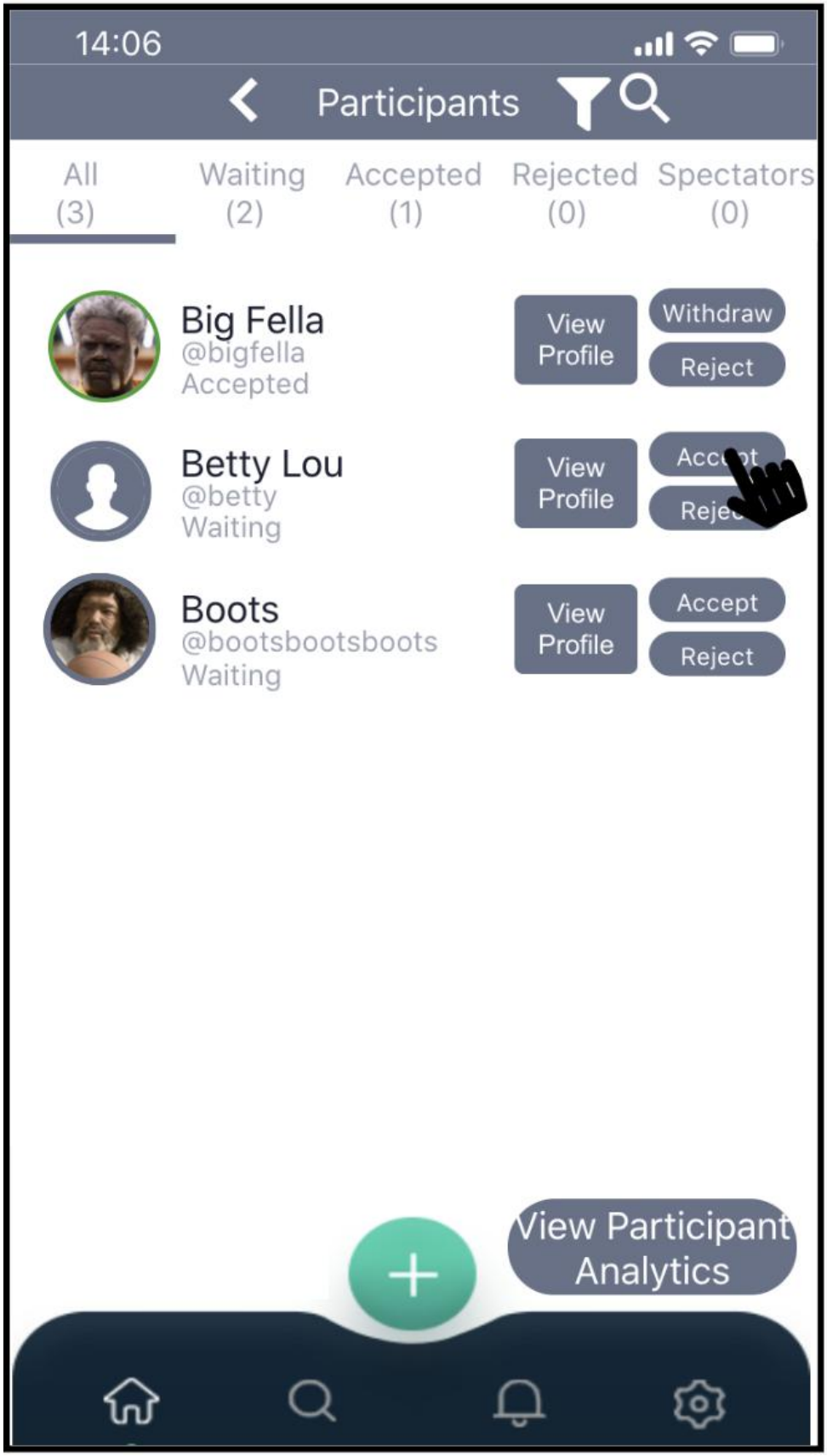


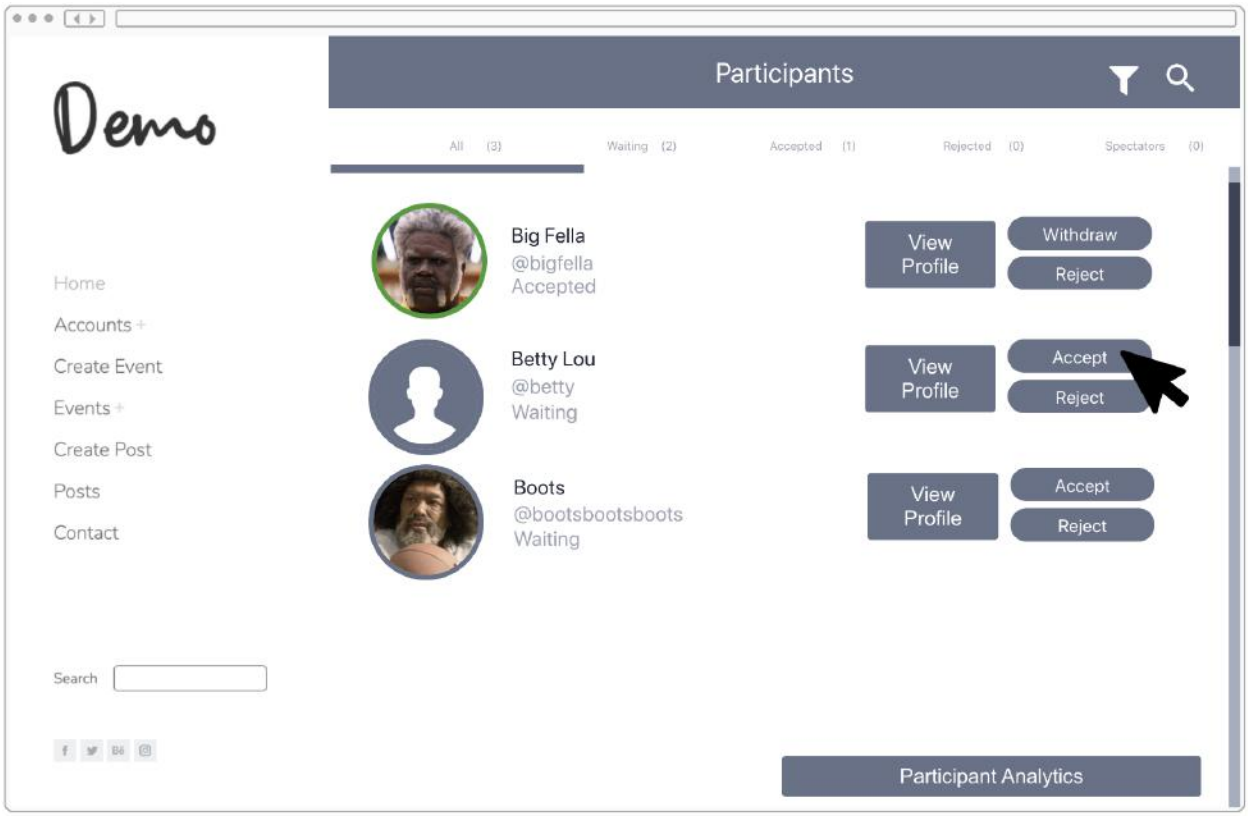
5. *He is in the Participants page. There are no selected participants yet and there are 3 applicants. The applicants are listed with their profile pictures, names and usernames. He sees that all 3 applicants are his old friends. He starts approving.*


14:06
Participants
All (3)
Waiting (3)
Accepted (0)
Rejected (0)
Spectators (0)
Big Fella
@bigfella
Waiting
View Profile
Accept
Reject
Betty Lou
@betty
Waiting
View Profile
Accept
Reject
Boots
@bootsbootsboots
Waiting
View Profile
Accept
Reject
View Participant Analytics

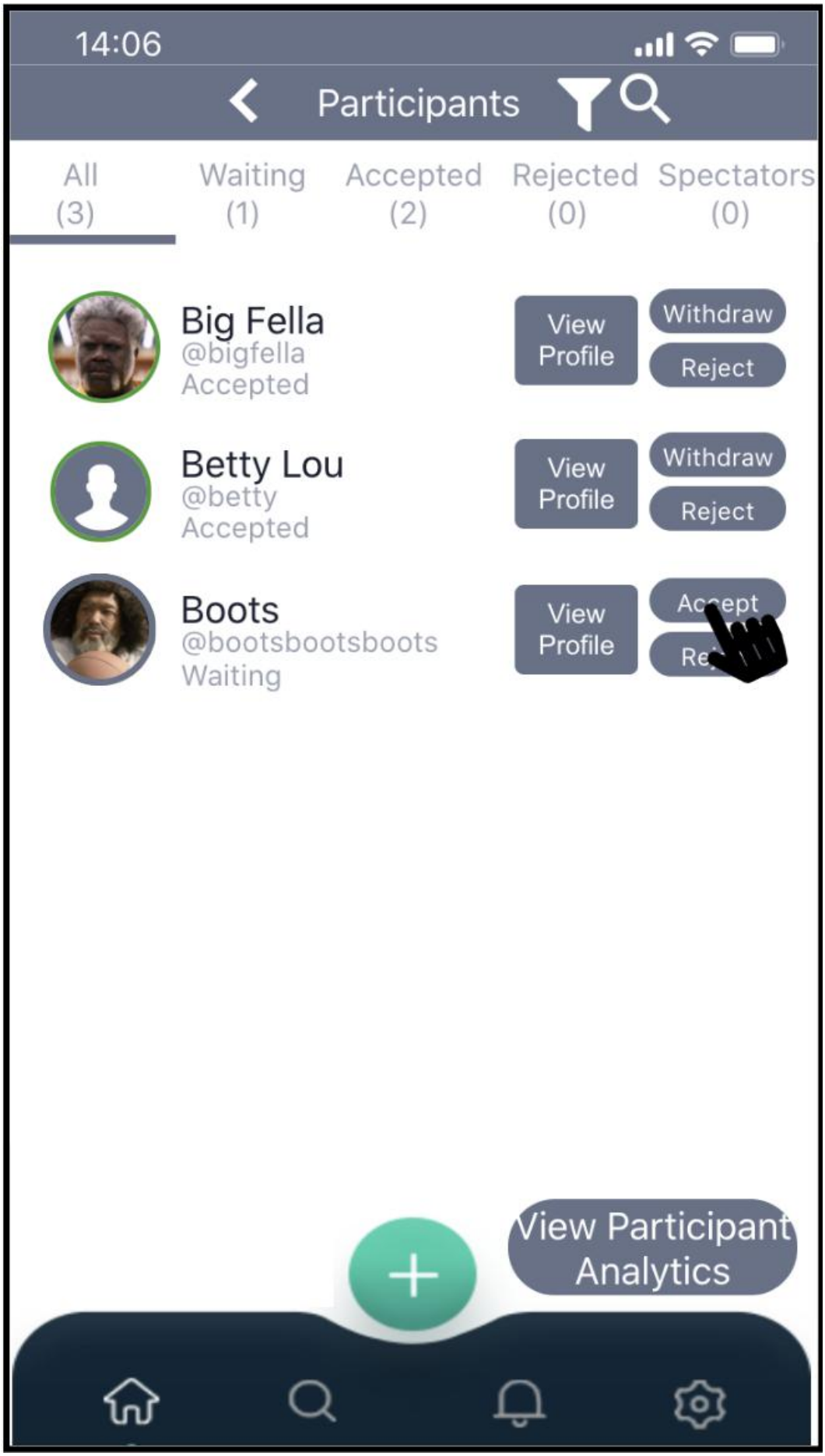


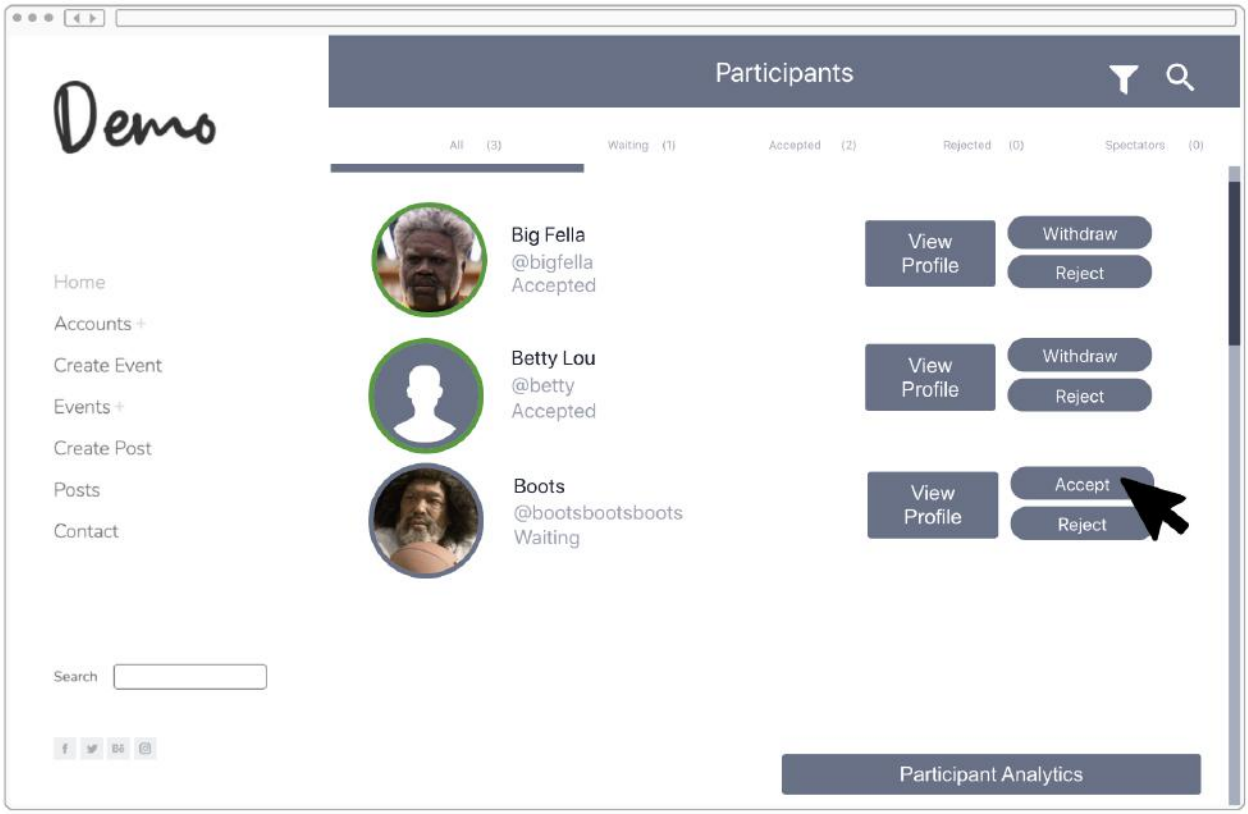
6. *As he approves the applicants, the participant counts in the above bar change. Also, the accepted participants profile page border turns green and their event participation status below their username is updated.*


14:06
Participants
All (3)
Waiting (2)
Accepted (1)
Rejected (0)
Spectators (0)
Big Fella
@bigfella
Accepted
View Profile
Withdraw
Reject
Betty Lou
@betty
Waiting
View Profile
Boots
@bootsbootsboots
Waiting
View Profile
Accept
Reject
View Participant Analytics

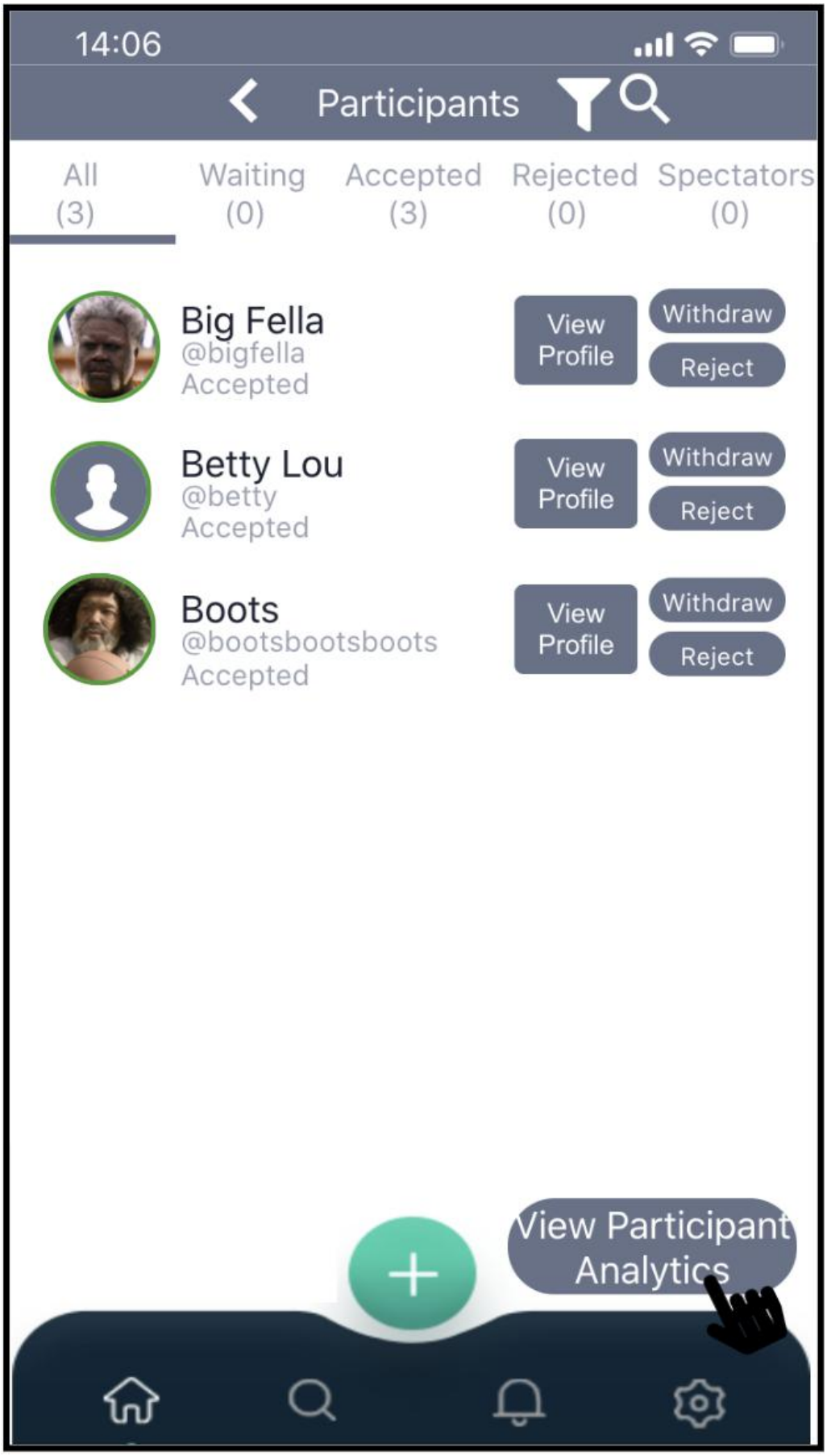


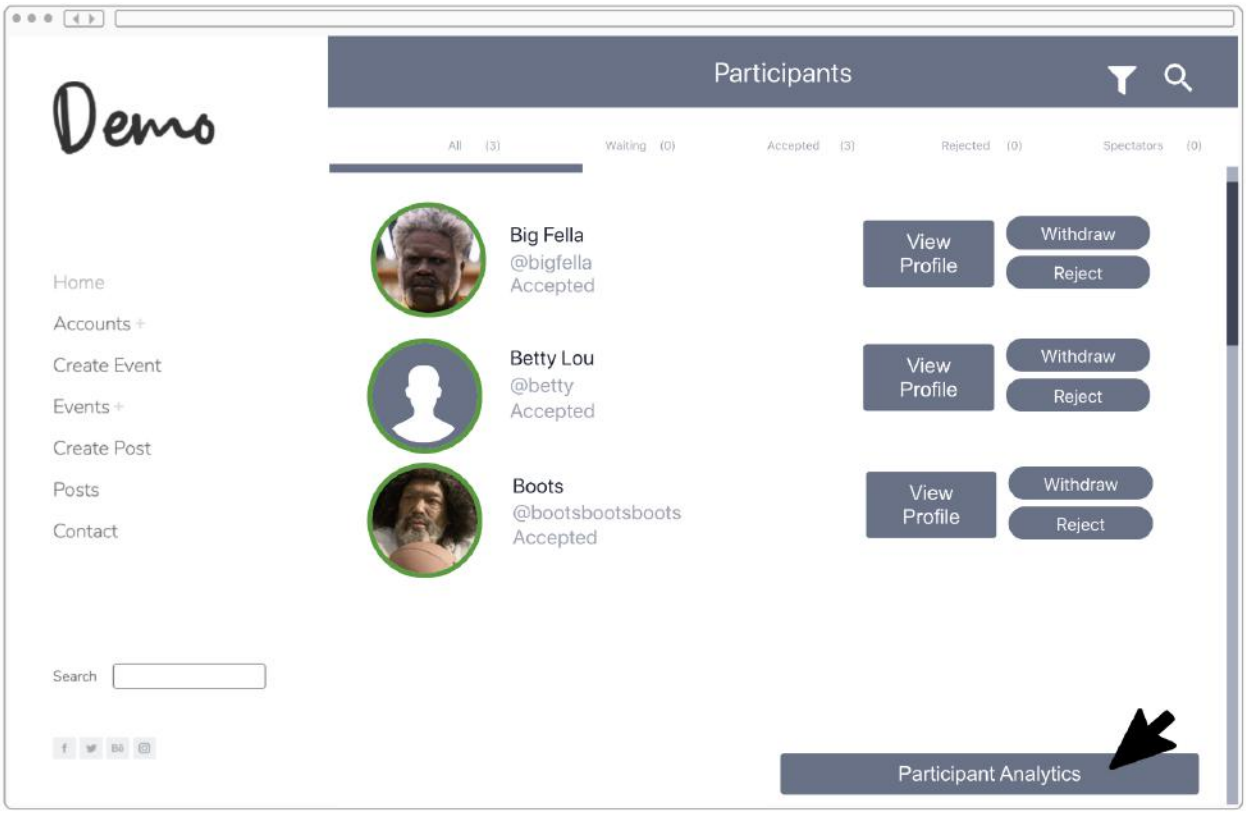
7. *He continues to approve. For the approved applicants, the approve button changes to withdraw, which allows the event creator to take the user back to waiting state.*


14:06
Participants
All (3)
Waiting (1)
Accepted (2)
Rejected (0)
Spectators (0)
Big Fella
@bigfella
Accepted
View Profile
Withdraw
Reject
Betty Lou
@betty
Accepted
View Profile
Withdraw
Reject
Boots
@bootsbootsboots
Waiting
View Profile
Accept
View Participant Analytics

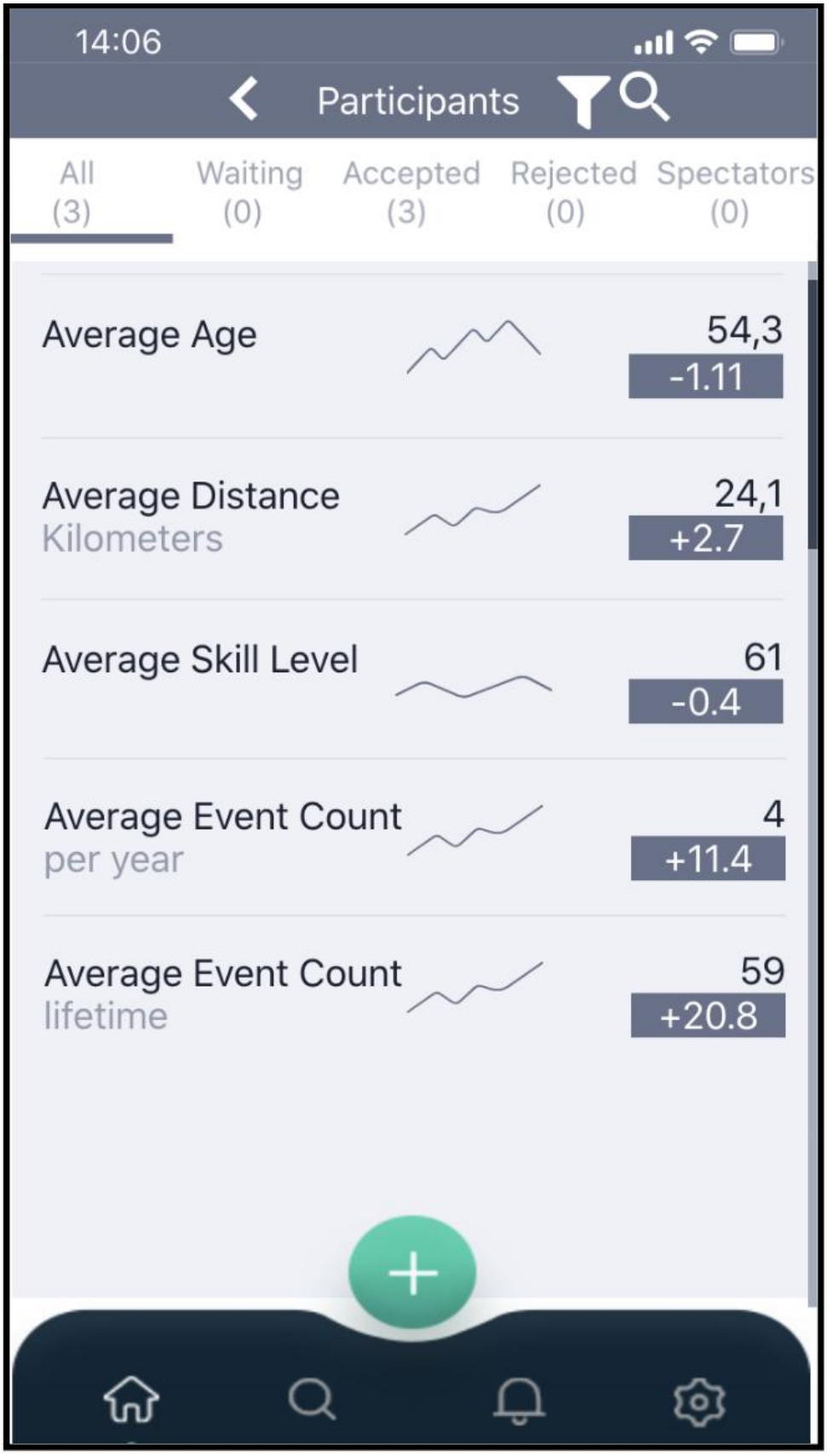


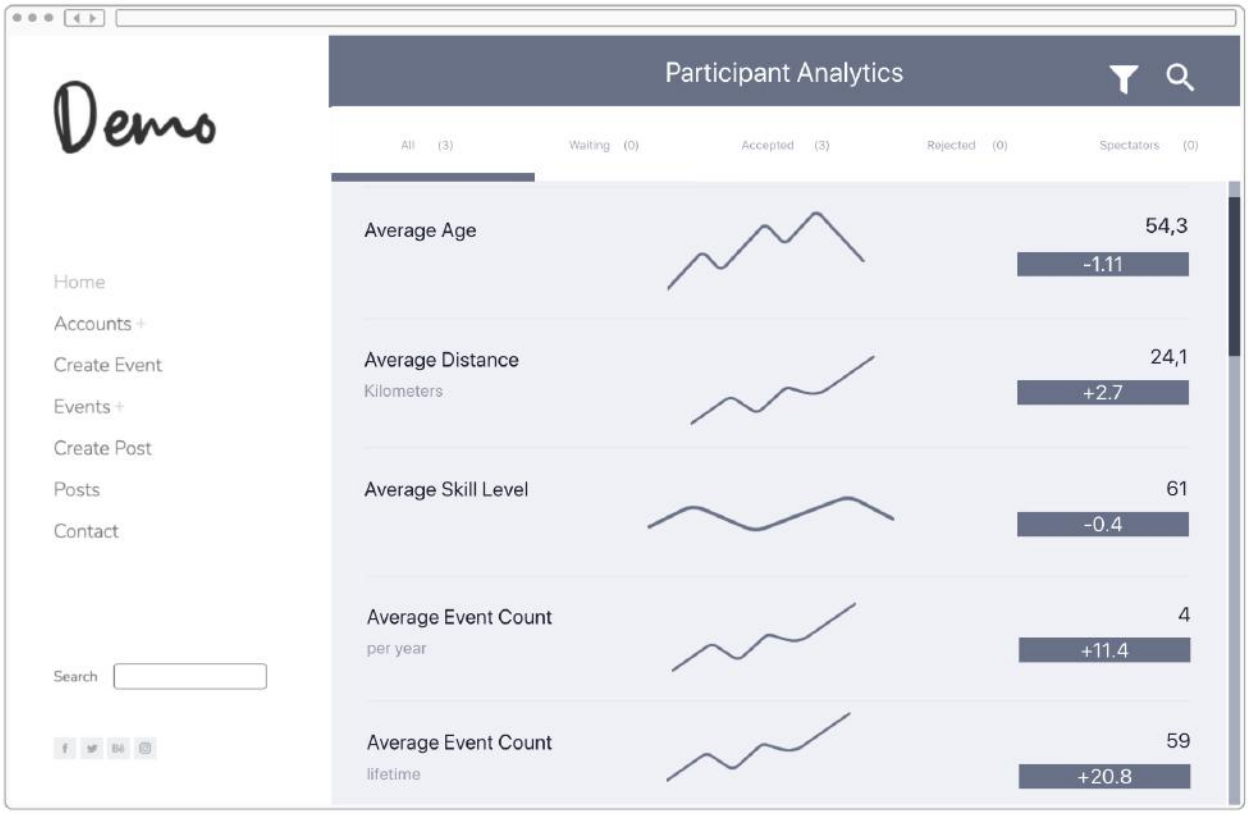
8. *He finishes approving all his 3 old friends' requests. He wonders the participant analytics as he haven't heard from them for a while. He clicks the View Participant Analytics button.*


14:06
Participants
All (3)
Waiting (0)
Accepted (3)
Rejected (0)
Spectators (0)
Big Fella
@bigfella
Accepted
View Profile
Withdraw
Reject
Betty Lou
@betty
Accepted
View Profile
Withdraw
Reject
Boots
@bootsbootsboots
Accepted
View Profile
Withdraw
Reject
View Participant Analytics



9. *He views the statistics about the participants of the event in this page.*


14:06
Participants
All (3)
Waiting (0)
Accepted (3)
Rejected (0)
Spectators (0)
Average Age
54,3
-1.11
Average Distance
Kilometers
24,1
+2.7
Average Skill Level
61
-0.4
Average Event Count
per year
4
+11.4
Average Event Count
lifetime
59
+20.8



10. *With the joy of reuniting with his old street ball gang, he quits the app (or the web page).*
11. *He goes to the Play Store and gives 5 starts, leaves a thankful comment.*