Group 6 - Milestone Report 2

Prepared by: Berk Atıl, Deniz Arda Budak, Ekrem Yusuf Ekmekci, Elif Sema Balcıoğlu, Musa Nuri İhtiyar, Ömer Faruk Süve, Salih Can Özçelik
Group number: 6
Group name: Pluto
Tag-id: https://github.com/bounswe/2021SpringGroup6/releases/tag/v.0.1

Table of Contents

Group 6 - Milestone Report 2
- Executive Summary
- Progress Based on Teamwork
  - Summary of Work
  - Progress according to requirements
  - API endpoints
  - Implementation of Core Functionality
  - User Interface/User Experience
  - User Interface/User Experience for Android
  - Standards
    - W3C Activity Stream 2.0
    - Wikidata

Executive Summary

The work we've already done until the first milestone was going over requirements and design documents such as sequence diagrams. Also we've started implementation part and completed the infrastructure related sections in general. Our next objective was focusing more on the core functionalities of the application.

Since the first milestone, we have mainly focused event related features which are the most basic building block of our application due to the fact that the primary purpose of our application was offering users a platform through which they can find people similar to them in terms of being interested in certain sports.

The main points we've focused in that context was allowing user to open a new event, view events related with the user, search from all the existing events using different criteria, viewing the profile of other users, viewing the latest activities which took place in the website (Activity Stream). We've implemented these ones in the second milestone since they were the most crucial and useful ones.

During the implementation part, the lessons we've learned from the previous milestone like making user interfaces of different pages consistent with each other. Furthermore, having an experience about what a customer presentation looks like and what the expectations of customers are were also beneficial.

For the rest of the project, we're planning to add more features to event related functionalities such as allowing users to discuss about an event. In addition, secondary features like badges and equipment ought to be integrated into the ecosystem. With these changes, our application becomes ready to be used since it addresses all the main titles in our requirements.

Progress Based on Teamwork

Summary of Work

Team Member	**Work**
Berk Atıl	• I have implemented blocking/unblocking users, changing visibility of attributes, retrieving blocked users, getting events, searching/filtering events, adding data into activity databases and getting activities in our system, deleting an event, and updating event information functionalities. Additionally, I have fixed some bugs in the endpoints that I have written before. Also, I have reviewed the Pull Requests that Elif opened and given some feedback.
Deniz Arda Budak	• I have reviewed the pull requests opened by Ömer and made comments. • I added the register functionality. • I opened the necessary and related issues.
Ekrem Yusuf Ekmekci	• Created "Event Settings"page: one tab for displaying user's events, one tab for the events that the user interacted with. • Created "Event" page: event info tab displays general information about the event including skill requirements and location, participation page lists spectators and participants. • Created "Activity Stream": picked a configurable component from alternative libraries, modified it and displayed data fetched from backend. • Created "View Profile" page: copied and

	modified user's profile page. ● Created "Modify Event" page: copied and modified create event page. ● General styling through and inside the pages.
Elif Sema Balcıoğlu	● I created the event database and implemented the create event endpoint. ● I worked on participation related endpoints. I implemented sending and deleting participation requests, adding and deleting spectating status endpoints. I also implemented accept and reject participants endpoint for event creators to select people for their events. I wrote get endpoints to obtain events participants, spectators and interested users. ● I implemented endpoints to get events user is participating, spectating or interested in. ● I worked on badges. I used Wikidata for our predefined badges. I implemented endpoints to get all badges on the system, give badges to other users and give badges to events. ● For all above functionalities, I wrote unit tests and documentations ● I deployed the front-end. ● I reviewed and merged Berk's pull requests.
İbrahim Melih Aktaş	
Musa Nuri İhtiyar	● I have prepared the opening a new event page for front-end part ● I have prepared the event search page for the front-end part. ● I have tried to deploy the front-end application. I have spent a lot of time, yet I have failed. ● I have also done classical tasks like looking at the pull requests of other people and opening some issues for other people ● I have written the executive summary for this report.
Ömer Faruk Süve	● I have implemented the user registration feature ● I have thought that we should implement our design in a more abstract way. I started to implement Base classes that later be inherited from other classes.

	• I have opened issues about features that later we need to implement, in order to see our future clearly. • I have opened several branches for different tasks. These tasks are the beginnings of some of the features like event search, blocking user operation, login token functionality, recyclerview implementation. I have progressed on these branches. • I have done research on login token functionality and I thought about the features that require us to use user tokens. • I have designed a Card View for event searching operations. This Card View includes a name, a photo related to the event, a description of the event and the date of the event. • I have redesigned the user registration page.
Salih Can Özçelik	• I have implemented the participation selection page, tried to deploy the frontend yet failed. Mostly spend time on working our team's code to get better with the environment.

Progress according to requirements

We report the status of each sub-group side by side, separating with |. Backend | Frontend | Android

- 1.1.1.1 Guests shall be able to register using username and password that they determine and weren't used by any other user before, kinds of sports they are interested in, their skill level, and email address. - **Completed | Completed | Completed**
- 1.1.1.2 Guests shall be able to provide additional information such as name, surname, location, age and gender about themselves at the registration stage. - **Completed | Completed | Completed**
- 1.1.1.3 Users shall be able to delete their accounts whenever they want. - **Completed | In Progress | In Progress**
- 1.1.2.1 Guests shall be able to login using their username and password combination. - **Completed | Completed| Completed**
- 1.1.2.2 Users shall be able to log out whenever they want. - **Completed | Completed | In progress**

- 1.1.3.1 Users shall update their username, password, location, age, gender, e-mail, skill level, and favorite sports. - **Completed | Completed | Not started**
- 1.1.3.2 Users shall be able to see the users they followed. - **Completed | Not started | Not started**
- 1.1.3.3 Users shall be able to see their event history. - **Completed | Completed | Not started**
- 1.1.3.4 Users shall be able to decide which information is visible to others. - **Completed | Not started | Not started**
- 1.1.3.5 Users shall be able to see the events they are interested in. - **Completed | In Progress | Not started**
- 1.1.3.6 Users shall be able to get a new password when they forget their passwords. - **Completed | Completed | Not started**
- 1.2.1.1 The system shall recommend the events that are for sports selected by the user. - **Not started | Not started | Not started**
- 1.2.1.2 The system prioritizes the events for users with the close skill level as the user. - **Not started | Not started | Not started**
- 1.2.1.3 The recommended events shall be automatically sorted by ascending distance. - **Not started | Not started | Not started**
- 1.2.1.4 The events created by the event creators, whom the user is following, shall be recommended. - **Not started | Not started | Not started**
- 1.2.2.1 The system shall send notifications on events, to which the user is "Interested" when the user is approved or rejected by the event creator. - **Not started | Not started | Not started**
- 1.2.2.2 The system shall send notifications when an event in which the user is "Interested" only has few available spots left. The meaning of few shall be determined for each sport separately. - **Not started | Not started | Not started**
- 1.2.2.3 The system shall send notifications when all the spots in an event to which the user is approved to participate are taken. - **Not started | Not started | Not started**
- 1.2.2.4 The system shall send notifications when an event to which the user has approved starts in a week, a day, and 3 hours. - **Not started | Not started | Not started**
- 1.2.2.5 The system shall send notifications if an event to which the user has approved has been canceled by the event creator. - **Not started | Not started | Not started**
- 1.2.2.6 The system shall send notifications if details of an event to which the user has been approved has been changed by the event creator. - **Not started | Not started | Not started**
- 1.2.3.1 The user shall be able to search events. - **Completed | Completed | In progress**
- 1.2.3.2 The user shall be able to filter the event results based on sport type, skill level and time. - **Completed | Completed | In progress**

- 1.2.3.3 The user shall be able to sort search results based on their distance if location data is shared, starting time, skill level for events. - **Not started | Not started | Not started**
- 1.2.3.4 The user shall be able to search users based on their name and username. - **In Progress | In Progress**
- 1.2.3.5 The user shall be able to search equipment. - **Not started | Not started | Not started**
- 1.2.3.6 The user shall be able to filter equipment based on the sport type. - **Not started | Not started | Not started**
- 1.2.3.7 The user shall be able to search field. - **Not started | Not started | Not started**
- 1.2.3.8 The user shall be able to sort field based on its distance. - **Not started | Not started | Not started**
- 1.2.3.9 The user shall be able to search events on map. - **Completed | Completed | Not started**
- 1.2.4.1 Every user shall have a profile page. - **Completed | Completed | In progress**
- 1.2.4.2 Every user's profile page is visible to other users. - **Completed | Completed | Not started**
- 1.2.4.3 Profile page of a user shall include the username. - **Completed | Completed | In progress**
- 1.2.4.4 Users shall be given the option to show and hide the full name, profile picture, age, user location, skill level for each sport, personal badges, previously attended events, previously created events. - **In Progress** | **Not started | Not started**
- 1.2.4.5 If the user is a field owner, field details, previous events that took place in the fields the user has shall be shown on the profile page of the user. - **Not started | Not started | Not started**
- 1.2.4.6 Users shall be able to block other users from their profile pages. - **Completed | In Progress | Not started**
- 1.2.4.7 Users cannot see the profile of the user who blocks them. - **In Progress | In Progress | Not started**
- 1.2.4.8 Users cannot move to the profile page of the user who blocks them when they see him/her on the event attendee list. - **In Progress | Not started | Not started**
- 1.2.4.9 Users shall be able to follow other users from their profile pages. - **Completed | In Progress | Not started**
- 1.3.1.1 Any user shall be able to create an event. After creating an event, the host is going to get the "Event Creator" status and have privileges merely for that specific event. - **Completed | Completed | Not started**
- 1.3.1.2 The creator for an event shall be able to cancel the event before it starts. In that case, anyone who stated that they are "Coming" to the event or "Interested" in the event shall receive notifications about the cancellation. - **In Progress | Not started | Not started**

- 1.3.1.3 Every event must have one host and any number of co-hosts. - **Not started | Not started | Not started**
- 1.3.2.1.1 Users can send registration request via event page. - **Completed | In Progress | Not started**
- 1.3.2.1.2 Users can declare themselves as spectator via event page. - **Completed | Completed | Not started**
- 1.3.2.1.3 Users can access information and comments/discussions about the specific event and place via the event page. - **Not started | Not started | Not started**
- 1.3.2.2.1 There will be some basic information about the event on this page like "Time", "Location", "Min and max number of participants" and "Description". - **Completed | Completed | Not started**
- 1.3.2.2.2 Also, some events might have some constraints for the users who want to participate. These "Requirements" are also shown in the "About" Tab. - **In Progress | In Progress | Not started**
- 1.3.2.2.3 There will be information about the badges to be delivered. After the event, the ones who gained the badges will be displayed. - **Completed**
- 1.3.2.2.4 Badges to be delivered can be changed by the creator. That is, "event owner to players badges" is customizable. See badges section. - **In Progress | Not started | Not started**
- 1.3.2.3.1 The event page will also include a "Discussion Page" for people to post their questions, talk about where to find the necessary equipment or offer changes in the plan. - **Not started | Not started | Not started**
- 1.3.2.3.2 Who can post or see the posts on this page can be adjusted by the event creator. - **Not started | Not started | Not started**
- 1.3.2.4 Unless otherwise stated, users shall be able to see who is "Coming" to the event, "Interested" in the event, and "Wants to Watch" the event. - **Completed | Completed | Completed**
- 1.3.2.5 Participants shall be able to share photos and videos after the event if the event creator allows, in the Event Page. - **Not started | Not started | Not started**
- 1.3.3.1 Depending on the event creator's choices, some events might require an approval from the creator before the participation. In these cases, the user shall seem to be "Interested" in the event and wait for the event creator to approve their participation. They can send an "approval request message" as to why they want to participate to obtain a higher chance of getting an approval. - **Completed | In Progress | Not started**
- 1.3.3.2 Event creator can make changes in the activity, even after its creation. - **Completed | Completed | Not started**
- 1.3.4.1 Users can ask their questions about the event in the "Discussion Page" if permitted. - **Not started | Not started | Not started**
- 1.3.4.2 For approval-required events, the users who want to participate can send an approval request and explain why they want to attend. - **Completed | In Progress | Not started**

- 1.3.5.1 The users who do not participate as player can be spectator. They also can undo it - **Completed | Completed | Not started**
- 1.3.5.2 Users can declare themselves as spectator via event page. (See 1.3.2.1.2) - **Completed | Completed | Not started**
- 1.3.5.3 If the field capacity is less than or equal to the number of spectators, new spectators will be warned about it. - **In Progress | Not started | Not started**
- 1.3.5.4 If the field capacity is less than the number of spectators, current spectators will be warned about it. - **Not started | Not started | Not started**
- 1.4.1.1 There will be certain badges already designed in the system. - **Completed | Not started | Not started**
- 1.4.1.2 Users shall be able to request new type of badges from the system. - **Not started | Not started | Not started**
- 1.4.1.3 There will also be badges indicating misbehaviour. - **Completed | Not started | Not started**
- 1.4.2.1 Users shall be able to acquire badges as event participators. **Completed | Not started | Not started**
- 1.4.2.2 Users shall be able to acquire badges as event creators. **Not started | Not started | Not started**
- 1.4.2.3 The badges shall be given by the system, event participators and event creators. - **Completed? | Not started | Not started**
- 1.5.1.1 Any user shall be able to create an equipment post. - **Not started | Not started | Not started**
- 1.5.1.2 The creator of equipment shall be able to delete the equipment from the system. - **Not started | Not started | Not started**
- 1.5.2.1.1 Users can access information and comments/discussions about the specific equipment and place via the equipment page. - **Not started | Not started | Not started**
- 1.5.2.2.1 There will be some basic information about the equipment on this page like "Location", "Related Sports" and "Description". - **Not started | Not started | Not started**
- 1.5.2.3.1 The equipment page will also include a "Discussion Page" for people to post their questions, talk about where to find the equipment. - **Not started | Not started | Not started**
- 1.5.2.3.2 Participants shall be able to share photos and videos about the equipment. - **Not started | Not started | Not started**

API endpoints

API Documentation :
https://github.com/bounswe/2021SpringGroup6/wiki/Squad-Game-API-Documentation
API URL: 3.20.232.108:8080
For the API call examples, we provide postman collections and you can reach via this link
https://documenter.getpostman.com/view/10799193/UVRAH6gW

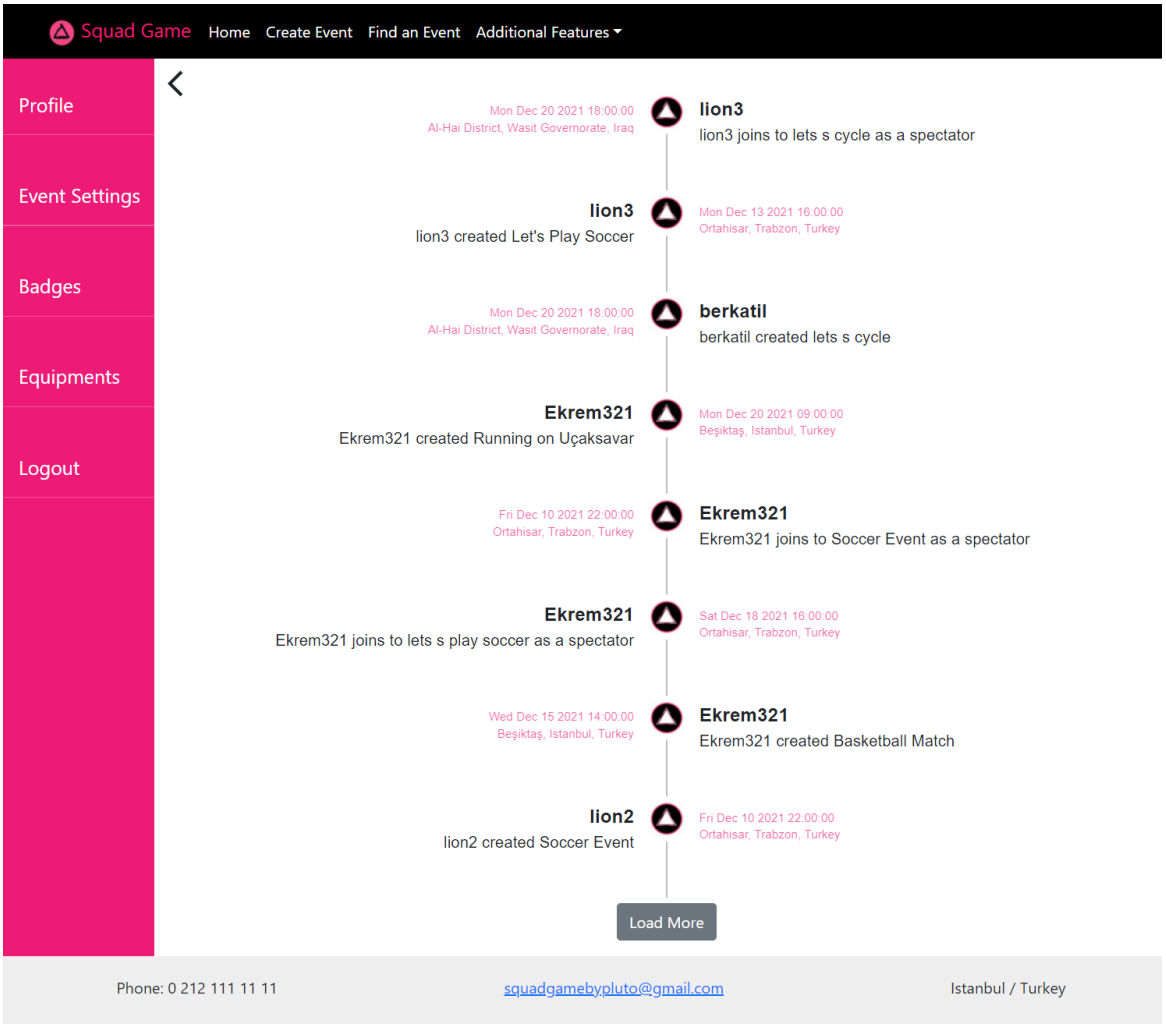
Implementation of Core Functionality

Web: Display/Select functionality is done via PiegonMap
Users can create, modify, filter events based on map functionality. Filtering is done by determining two points. These points correspond to two edges of a rectangle. Events are filtered regarding the area of the rectangle.
Event creation is done by selecting a point in the map. In the future, users may choose a field through map.
An event's location information is displayed on its page.
Field creation and sorting is not implemented. However, their relation with map functionality would be similar to the event's.

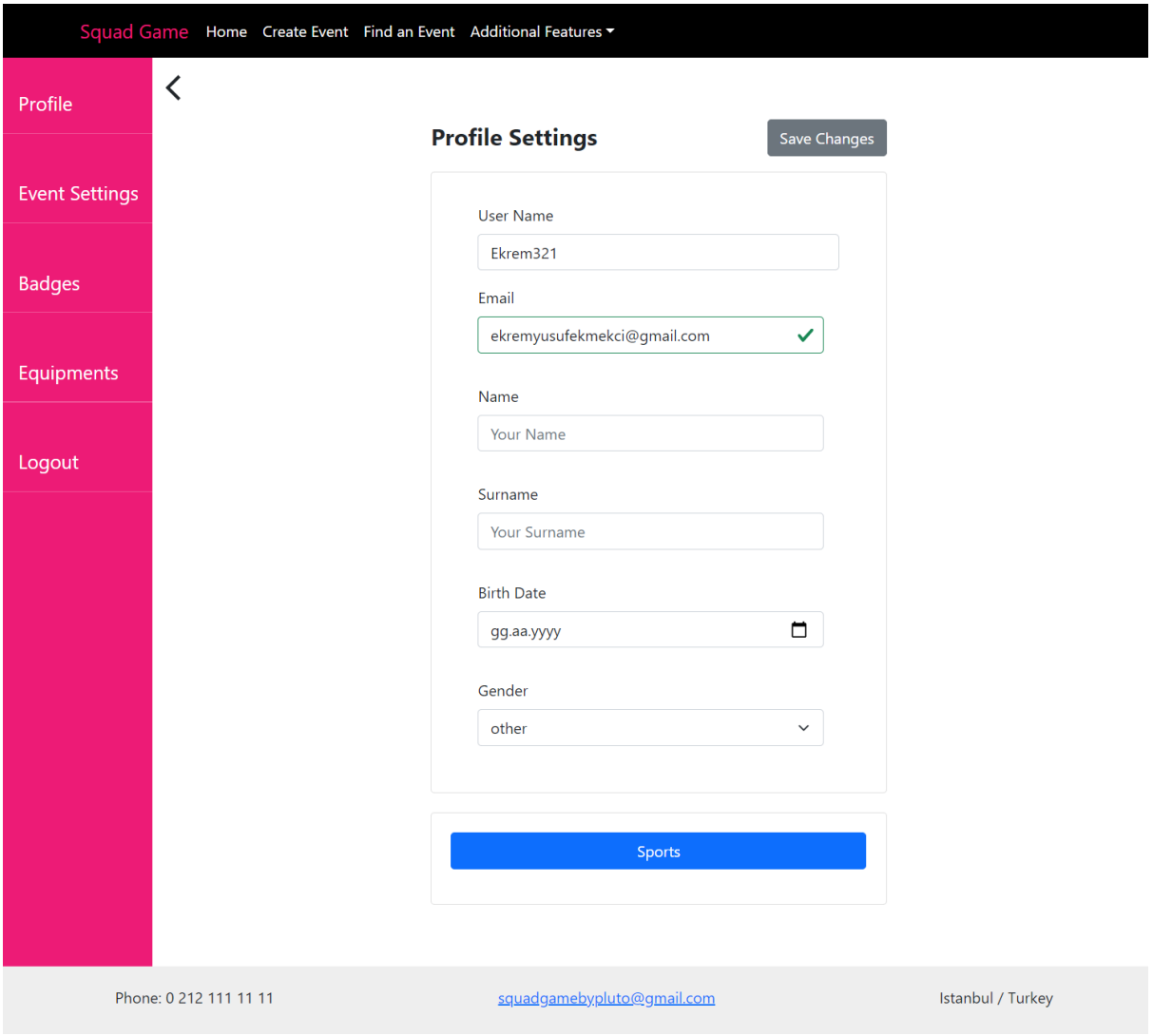
User Interface/User Experience

Web: http://3.20.232.108:3001

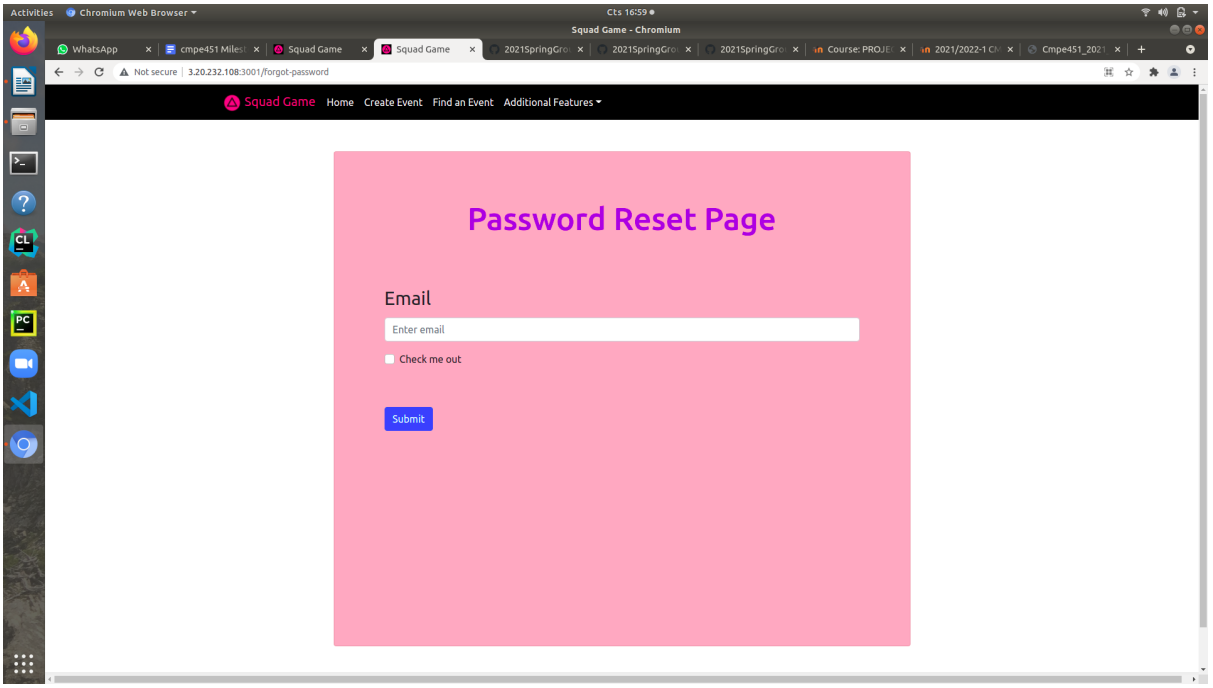
Homepage:
https://github.com/bounswe/2021SpringGroup6/tree/master/front-end/playground/src/pages/ActivityStream



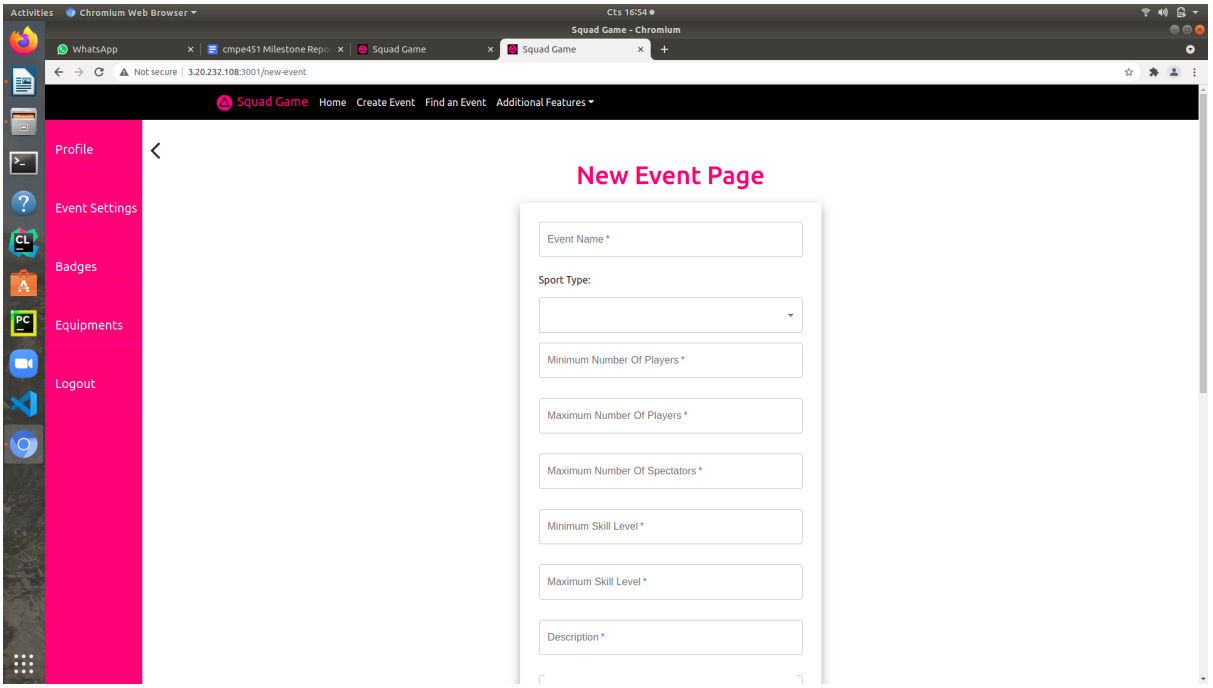
Profile Page: /profile or /profile/<id>
https://github.com/bounswe/2021SpringGroup6/tree/master/front-end/playground/src/pages/profile

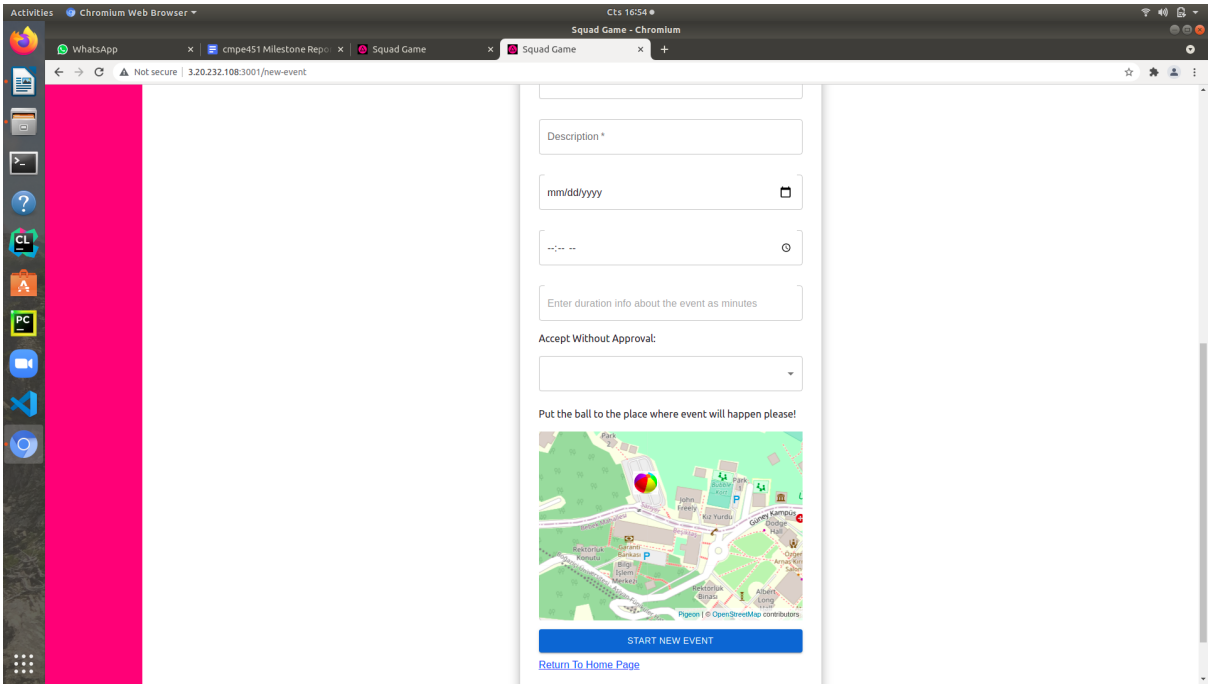


Forgot Password Page: /forgot-password
https://github.com/bounswe/2021SpringGroup6/tree/master/front-end/playground/src/pages/PasswordReset

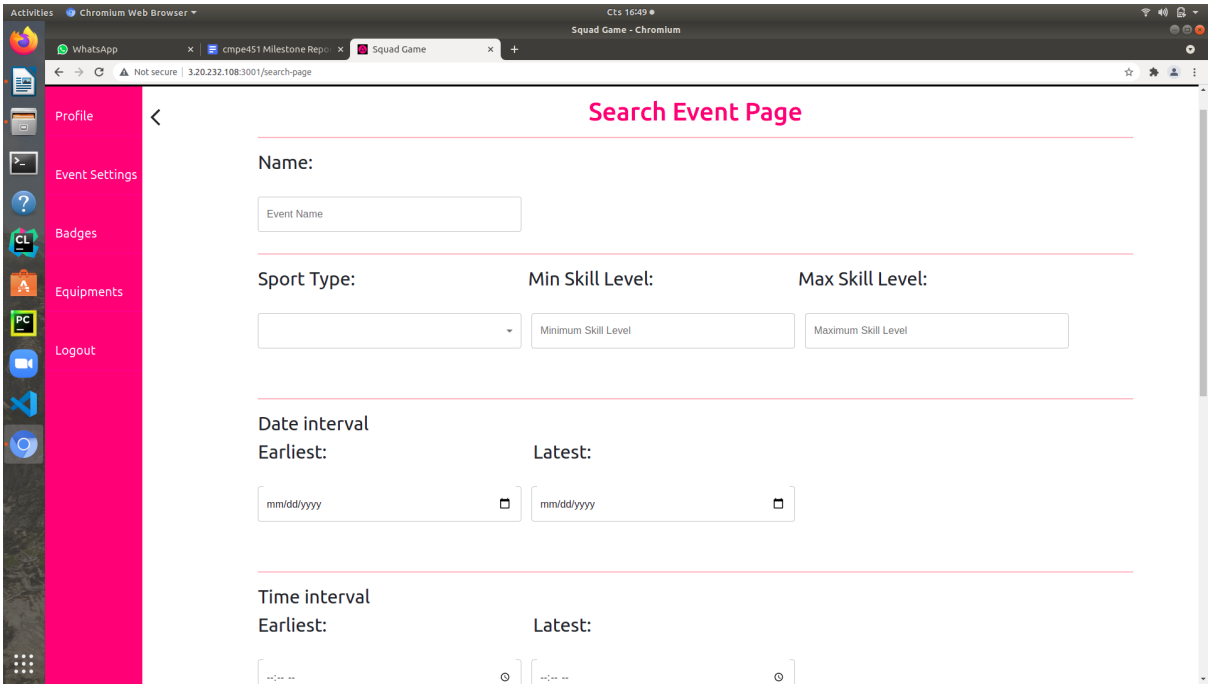


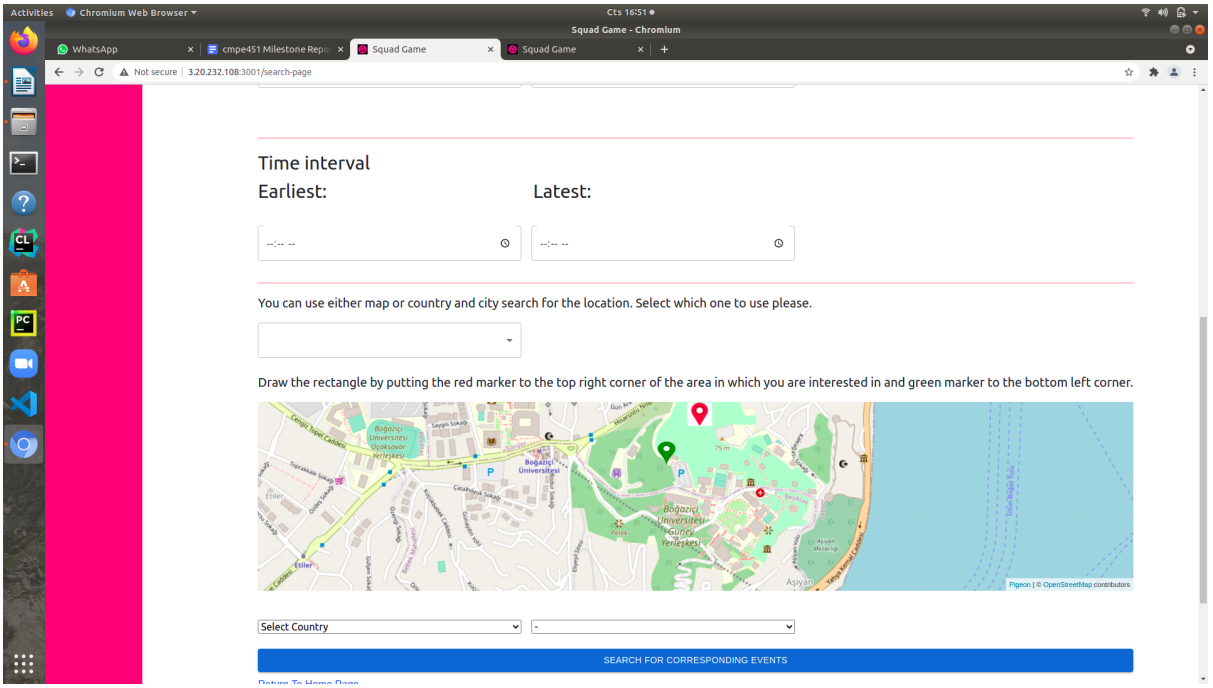
Create Event: /new-event
https://github.com/bounswe/2021SpringGroup6/tree/master/front-end/playground/src/pages/NewEvent



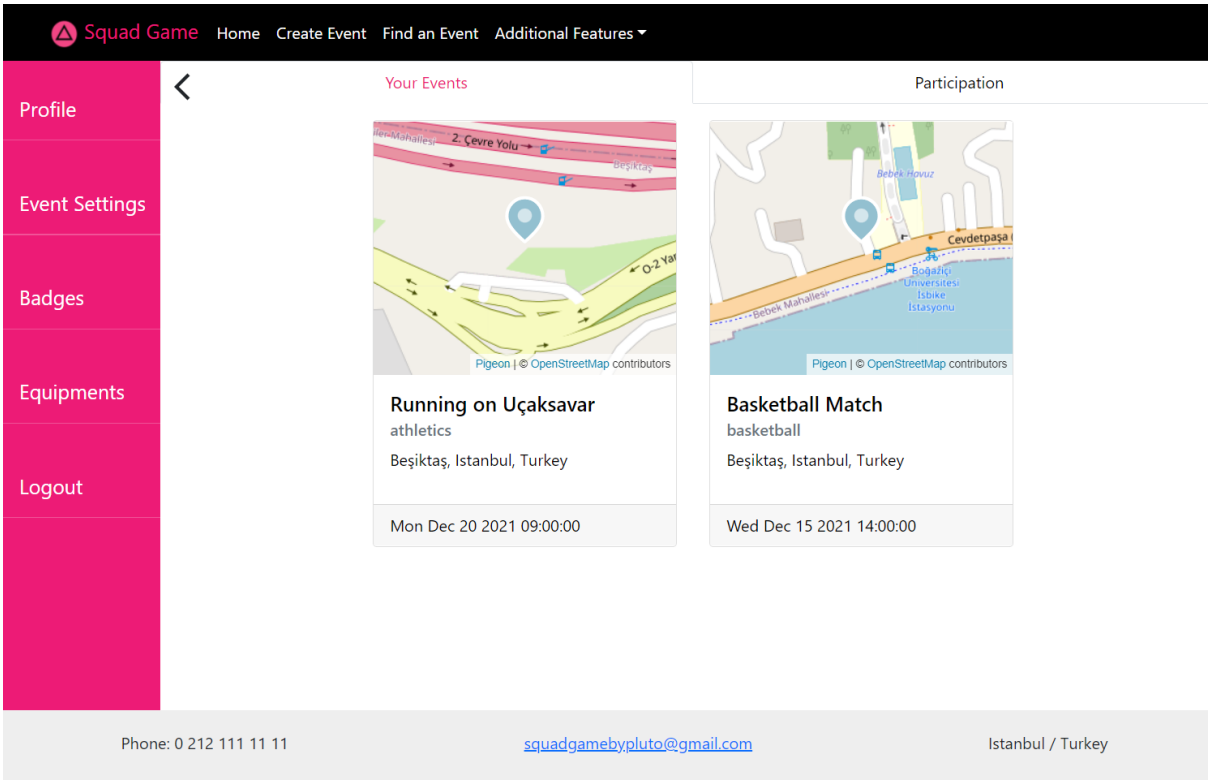


Search Event: /search-page
https://github.com/bounswe/2021SpringGroup6/tree/master/front-end/playground/src/pages/SearchPage

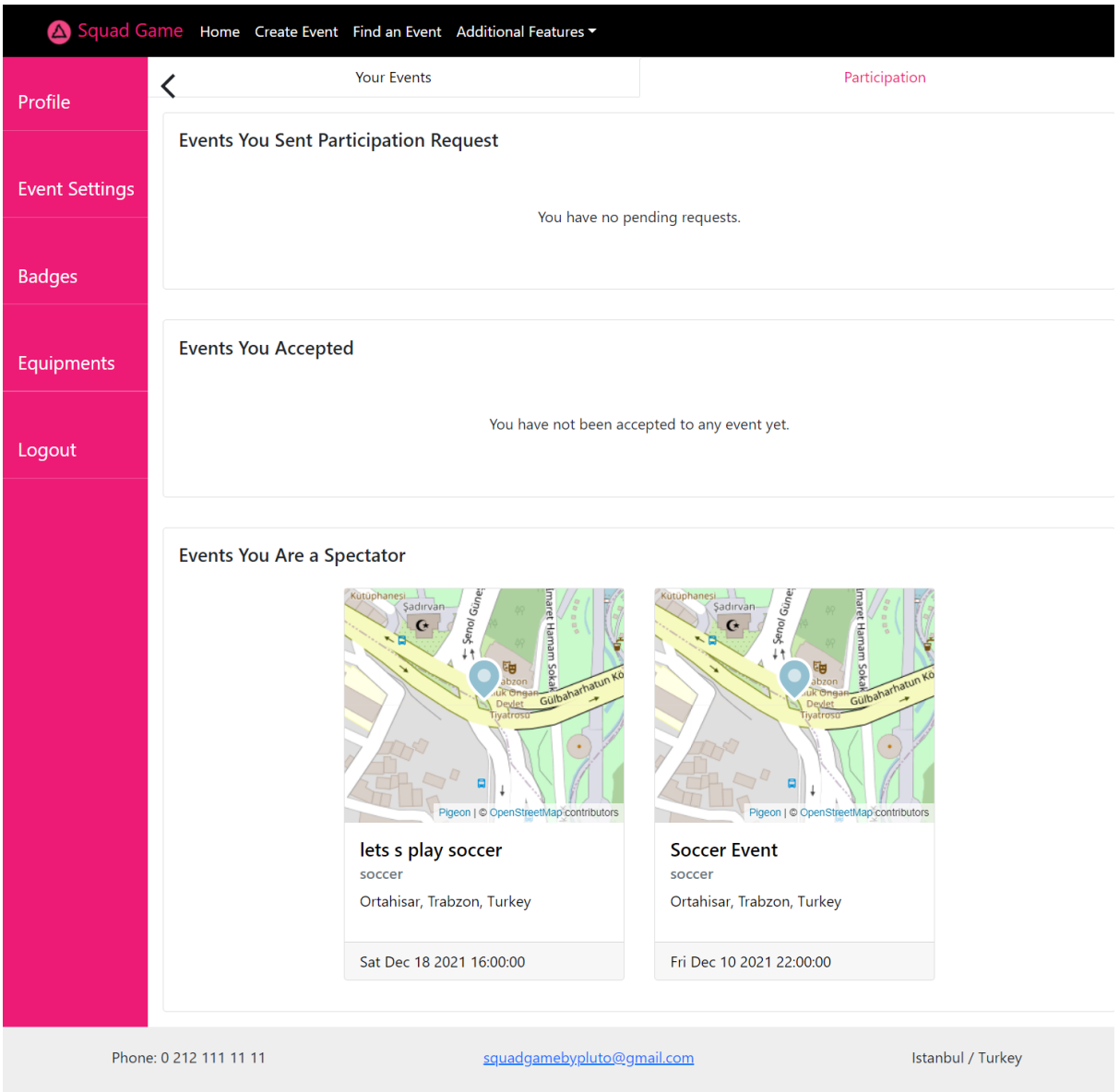




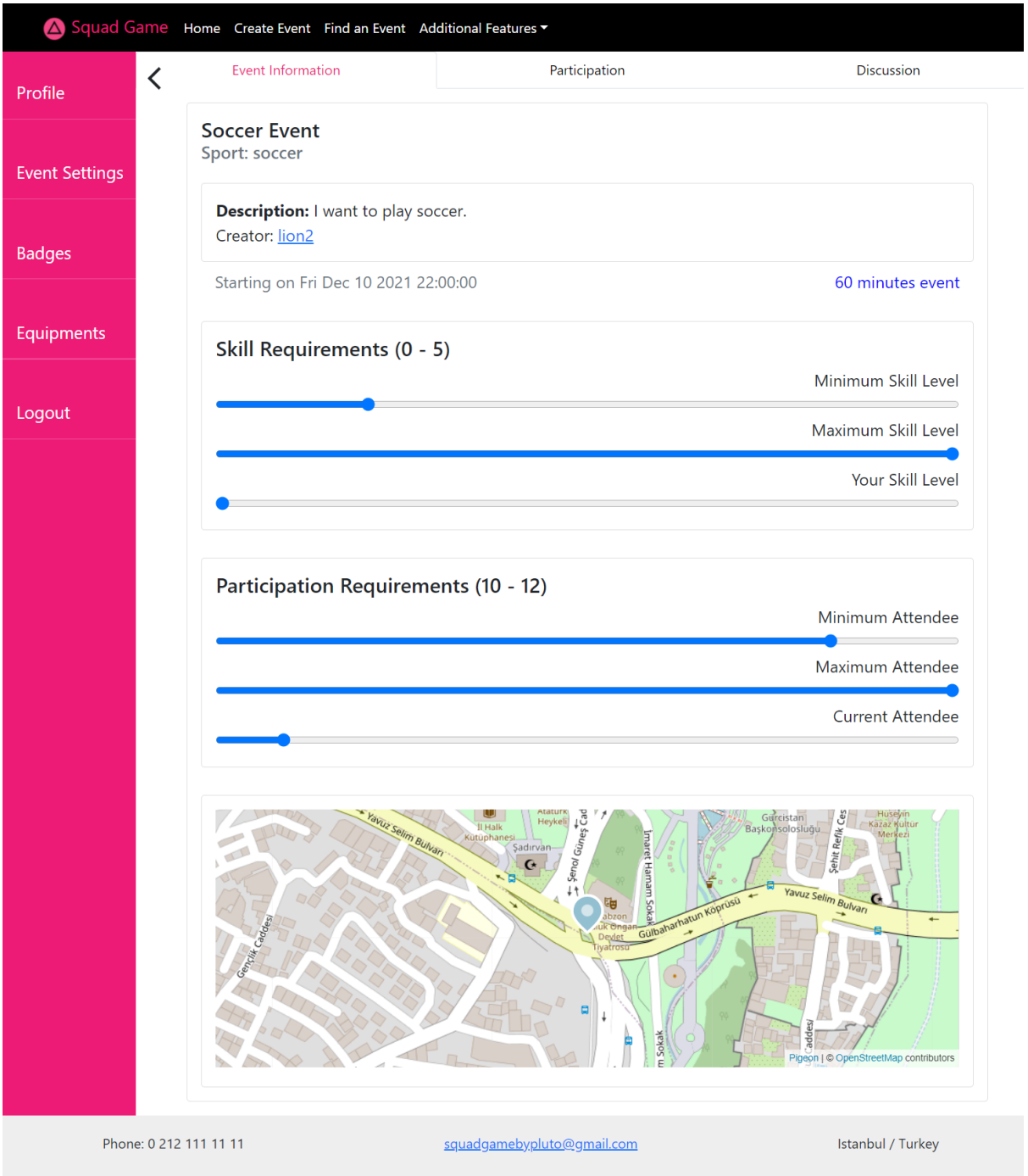
User’s Events: /event-settings
https://github.com/bounswe/2021SpringGroup6/blob/master/front-end/playground/src/pages/Event/PersonalEvents/CreatedEvents.js



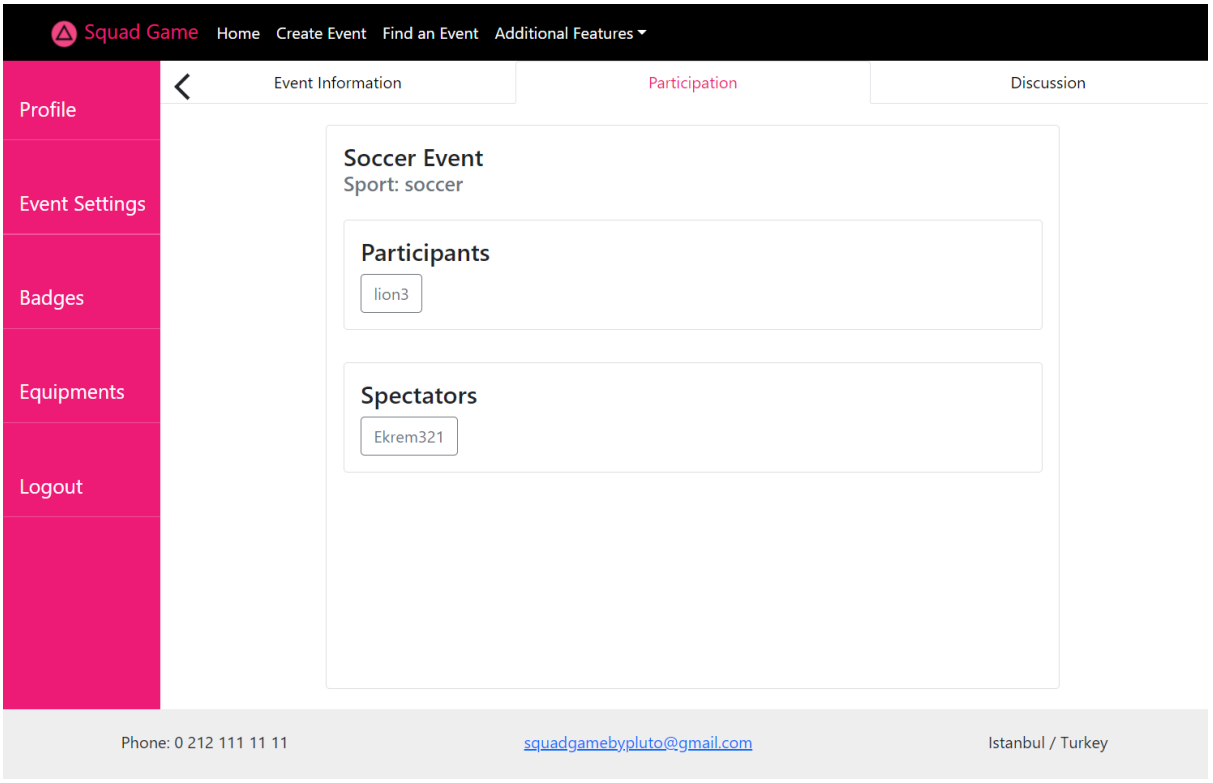
User's Participation/Related Events: /event-settings (other tab)
https://github.com/bounswe/2021SpringGroup6/blob/master/front-end/playground/src/pages/Event/PersonalEvents/ParticipationPage.js



Event Information Page: /event/<id>
https://github.com/bounswe/2021SpringGroup6/tree/master/front-end/playground/src/pages/Event/EventInformation

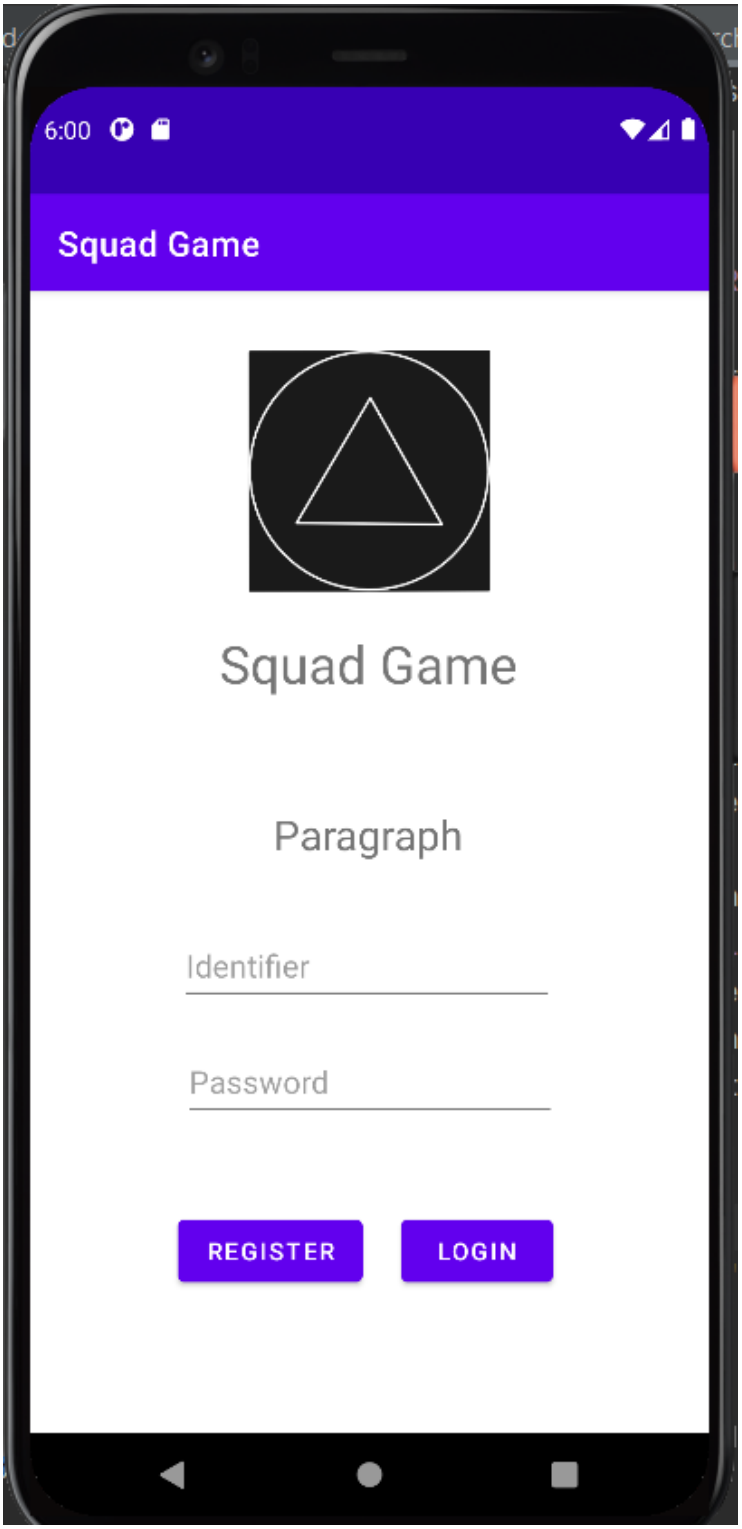


Event Participation Page: /event/<id> (other tab)
https://github.com/bounswe/2021SpringGroup6/tree/master/front-end/playground/src/pages/Event/EventParticipationInfoPage

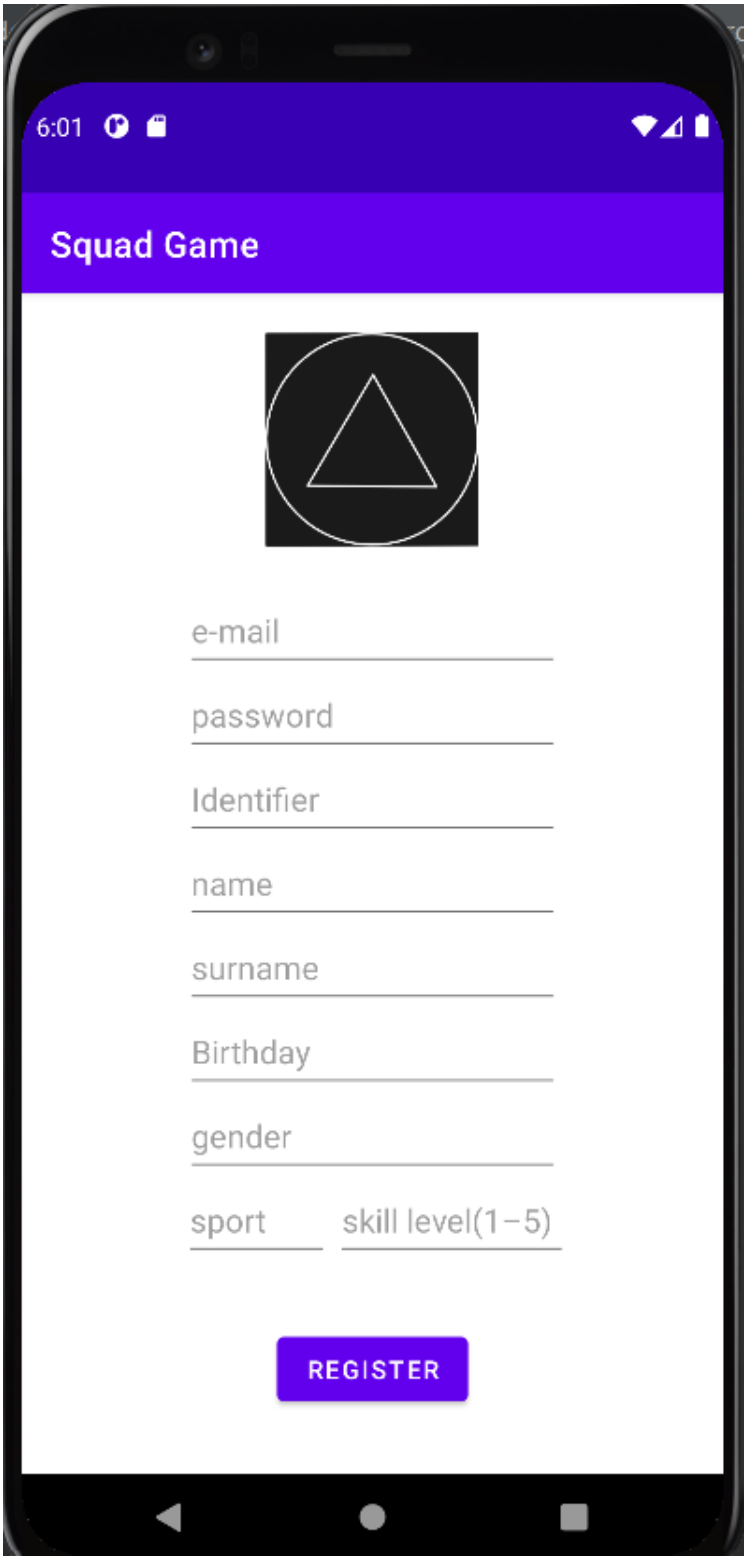


User Interface/User Experience for Android

Opening Page:
https://github.com/bounswe/2021SpringGroup6/blob/android/android/AndroidSportsPlatform/app/src/main/res/layout/activity_main.xml

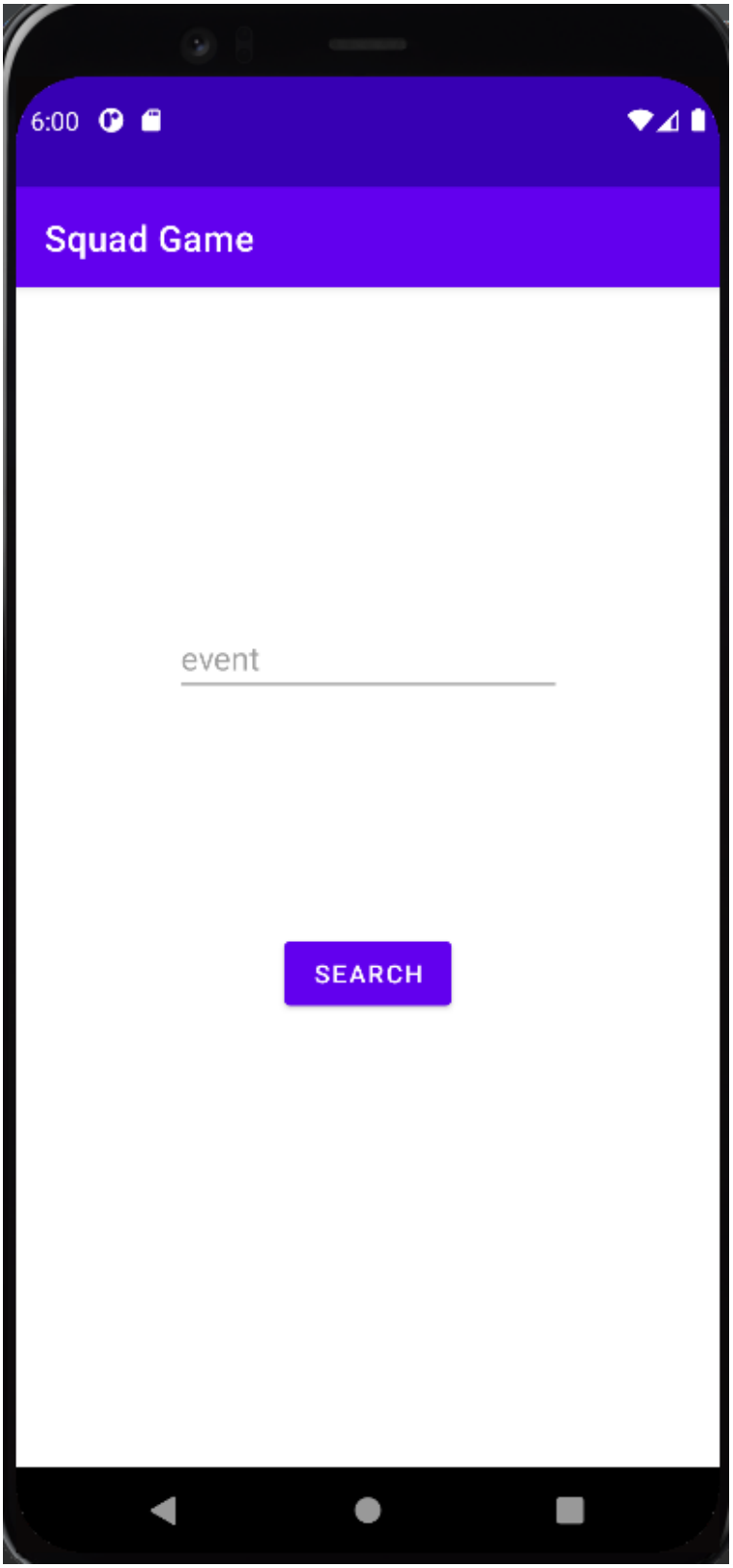


Registration Page:
https://github.com/bounswe/2021SpringGroup6/blob/android/android/AndroidSportsPlatform/app/src/main/res/layout/activity_register.xml



Homepage:
https://github.com/bounswe/2021SpringGroup6/blob/android/android/AndroidSportsPlatform/app/src/main/res/layout/activity_profile.xml

Search Page:
https://github.com/bounswe/2021SpringGroup6/blob/android/android/AndroidSportsPlatform/app/src/main/res/layout/activity_search_operations.xml



Standards

W3C Activity Stream 2.0

Back-end

Activity Stream is used as a feed of activities. To store the feed on the back-end, Activity Stream database is implemented. We selected the core activities for our application, specifically user block operation, user following operation, event creation, spectating an event activity, and accepting an interested user for an event by the event organizer operation as our stream activities. All these activities performed by the users are added to the database. These activities can then be obtained through the /activitystream endpoint. This endpoint takes two parameters, limit, which is the number of activities to return and offset, which is the id of the newest activity. Then an OrderedCollection from the W3C Activity Stream 2.0 Specification is returned.

Currently we do not make any filtering based on the user requesting the stream. In the future we plan to filter based on followed users and sports.

Front-end

Activity stream is displayed on the web application's home page. It contains links to corresponding subjects of the specified activities.

Wikidata

Back-end

We used Wikidata on the badges we add to our system. We chose our badges from the Wikidata system. At the start of the database, we add the selected badges. Not all badges are from the Wikidata system. After implementing the equipment system, we plan to use Wikidata to connect equipment to sports.