Group 4

Milestone Report 1

SportsHub

Amateur Sport Events Platform

20.11.2021

Ali Alperen Durak
Ece Dilara Aslan
Fatih Akgöz
Mehmet Hilmi Dündar
Muhammed İrfan Bozkurt
Tolga Kerimoğlu
Yağmur Selek
Yiğit Sarıoğlu

1. Executive Summary

Starting off the semester with new team members and the task of creating a fully-fledged application on two fronts, we were happy to take on the challenge. The first few weeks were spent adapting to the new workflow, onboarding our new members, and forming respective teams for frontend, backend, and android. Deciding on the communication schedule and getting used to the new workflow, we were able to pick up the pace and make a significant improvement across all teams, which is what this document will be about.

- **What has been done?**

The first order of business was going over the deliverables from the previous semester. Revising our requirements, refining and updating our class diagrams and project plan. This was crucial for proper planning and the smooth development of our applications.

On the frontend team, unlike our other two teams, both members were new to the technologies and had to learn from scratch. Nevertheless, we were able to present a fully functioning login/register system along with event creation and event search views based on geolocation. No mock data was used and the application was properly connected to the backend. We were happy with the results and what we could manage to put together until the demo.

Having an experienced lead, the backend team started off the semester quickly and lots of core functionality were available before the demo date. Besides the geocoded search, login/register, and event creation functionalities, further functionalities such as user search by name or username, adding applying to events, and notification were also implemented but not displayed during the demo. The backend API was also fully deployed into an Amazon EC2 instance.

The android team was also able to display login/register functionality with event listing and search pages as well but the backend connections were not complete as of the demo date which was something we could definitely improve upon.

- **Plans for the next milestone**

Having found our pace we believe we'll be able to bring lots of new functionality to display for the next milestone. Among the plans are a fully functioning notification system, commenting on events and support for discussions, tags and badges for users, and following/blocking functionalities. As ambitious goals, we can list activity history tracking and an interactive map implementation for locating and searching events. Additionally., we also plan to initialize a CI/CD pipeline using Github workflows for easier development.

2. List and Status of Deliverables

	DELIVERABLE	DUE DATE	STATUS
1	**PROJECT PLAN**	21.10.2021	DELIVERED
2	**PROJECT REQUIREMENTS**	21.10.2021	DELIVERED
3	**ANDROID**		
	REGISTER PAGE	21.10.2021	PARTIALLY
	LOGIN PAGE	21.10.2021	PARTIALLY
	SEARCH EVENTS PAGE	21.10.2021	PARTIALLY
	CREATE EVENT PAGE	21.10.2021	PARTIALLY
4	**FRONTEND**		
	HOME PAGE	09.11.2021	PARTIALLY
	REGISTER PAGE	09.11.2021	DELIVERED
	LOGIN PAGE	09.11.2021	DELIVERED
	EVENT PAGE	14.11.2021	DELIVERED
	EQUIPMENT PAGE	14.11.2021	PARTIALLY
	SEARCH EVENT PAGE	15.11.2021	DELIVERED
	CONTACT PAGE	02.11.2021	DELIVERED
	ABOUT US PAGE	02.11.2021	DELIVERED
5	**BACKEND**		
	REGISTER ENDPOINTS	21.10.2021	DELIVERED
	LOGIN ENDPOINTS	21.10.2021	DELIVERED
	CRUD ENDPOINTS FOR EVENTS	02.11.2021	DELIVERED
	SEARCH ENDPOINTS FOR EVENTS AND USERS	13.11.2021	DELIVERED
	NOTIFICATIONS	14.11.2021	PARTIALLY
	COMPREHENSIVE UNIT TESTS FOR EVENTS AND AUTHENTICATION APIS	14.11.2021	DELIVERED

3. Evaluation of the status of deliverables

Project Plan

We have done the Project Plan in the CMPE 352 course. But some of our friends (team members) left us. New teammates joined us. That's why we changed the project plan and responsibility assignment matrix(RAM), appointed new friends and changed the dates. Also, the initially decided roles were changed and new tasks were added.

Requirements

Requirements have been reviewed according to discussions with the customer. The final requirements seem feasible both related to the customer needs and implementation considerations.

Communication Plan

Communication plan was also reviewed at the beginning of the semester. New members joined the team and the communication plan was updated to suit everybody's needs.

Backend Deliverables

The backend team successfully implemented functionality for login/register, event creation, event update, adding/removing participants/followers/applicants to events, geocoded event search by location and distance, user search by username, name, or keyword.

You can find the code and manual documentation here:
https://github.com/bounswe/2021SpringGroup4/tree/backend-main/code
You can find the deployed endpoints at http://ec2-3-67-188-187.eu-central-1.compute.amazonaws.com:8000/

Frontend Deliverables

- **Home Page**
  This page contains navigation bar. The user can view the homepage after logging in to the system.It can only display certain pages that have been previously determined.
- **Login Page**
  On the website's Login Page, users are prompted to give their credentials. This page is completed and all of the relevant requirements are met. After logging in, the logout button is also activated on the page.
- **Register Page**

This page allows guest users to create an account on our platform. The required information for an account is first name, email, password. There is a form validation implemented, when user enters wrong data, system warns and does not allow registration.

- **Event Page**
  Both guest and registered users can discover the listed events as preview form. When they click any event, the box which includes detailed information will appear. They can also search the events via entering the location and maximum distance.
  For registered users there is an opportunity to participate in the events using the join button.

Android Deliverables

- **Register Page**
  Register page’s ui and navigation components are successfully implemented however since android app is not integrated with backend properly it doesn’t send necessary requests and handle the responses.
- **Login Page**
  Same as Register page ui and navigation is ready but rest is not working properly yet.
- **Search Event Page**
  Search event page is designed such that page gets all of the events after login and via the search bar at the top user can search by location with predefined range, however rest connection wasn’t integrated so it shows static data for events.
- **Create Event Page**
  Create Event Page has only a Fragment as ui and has no function as of now.

4. Evaluation of Tools and Process

We have used Discord, Google Meet and Whatsapp as our communication platforms, Python as our main programming language, Django as our web framework, React JS for frontend user-interfaces, Android Studio (Kotlin JS language) for the android app, PostgreSQL as our database server, coreAPI for automatic documentation, git for version management. We will evaluate all the tools one by one.

Django Rest Framework

Django Rest Framework is a framework for Django that aids in the creation of RESTful APIs. It comes with many features supporting generic endpoint creation and it also provides facilities for customization. Overall, we have found the process relatively smooth using DRF and did not encounter any functionality we needed that could not be implemented within DRF. The learning curve can be steep but overall we are happy with our choice.

Github

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. The Github Issue System is a very useful thing for us in a way that it helps us to see the progress of the assignments/works. We have documented all the works in Github Wiki Pages. We also opened different branches for every sub team (android-backend-frontend). We have worked in different branch, this gave us the opportunity to work more easily

Docker

Docker is an open source containerization platform. It enables us to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. We also dockerize our backend ,so everyone(group members) used that docker image in their locals and easily run their codes.

Postman

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration.Postman is an application used for API testing. It is an HTTP client that tests HTTP requests, utilizing a graphical user interface, through which we obtain different types of responses that need to be subsequently validated.

Visual Studio Code

Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications.We have used Visual Studio IDE, both development on backend with django and frontend with react js. It has helped us a lot with its addons.

Android Studio

Android Studio is the official Integrated Development Environment (IDE) for

Android app development, based on IntelliJ IDEA with Google collaboration. A unified environment where we can design and develop for all types of Android devices including smart TV's and smart watches. It comes with AVD manager which is a tool for virtual android devices supporting various screen sizes and OS builds.Developers can also test and run their code in real Android devices with ease. It comes with a wide range of libraries and custom implementations which can be added to projects via gradle.

Google Meet

Google Meet, is built to let dozens of people join the same virtual meeting, and speak or share video with each other from anywhere with internet access. We have used this platform to communicate and talk to each other.

5. The requirements addressed in this milestone

- 3.1.1.1.1 User shall include his/her name, username, password, e-mail during sign-up.
- 3.1.1.3.1 User shall be able to delete his/her account.
- 3.1.1.4.2 Event creator shall provide location, date, duration, and type of an event when creating an event.
- 3.1.1.5.7 User shall be able to comment on the event page.
- 3.1.1.6.3 User shall be able to create equipment sale posts that include its description.
- 3.1.2.1.1 The system shall provide users with options to create a new account or sign-in to an existing account.
- 3.1.2.1.4 The system shall verify the entered information before creating an account for registering users. Users who provide invalid information should not be allowed to create an account.
- 3.1.2.3.1 The system shall provide information about the events such as event name, location, date, sport and the number of players.
- 3.1.2.3.4 The system shall allow users to create events by specifying event name, location, date, sport and the number of players.
- 3.2.1.1 The system shall run as a web application, available through modern web browsers & an Android application.
- 3.2.1.2 Cross-platform languages shall not be used implementing the system.
- 3.2.1.3 The system language shall be English.
- 3.2.3.3 Users shall not be asked to grant any information besides than a name & an e-mail address.
- 3.2.4.3 E-mail and password info for users shall be encrypted before storage.
- 3.2.4.4 No user shall be granted admin permissions.

6. Summary of work performed

BACKEND

TEAM MEMBER	CONTRIBUTION
Tolga Kerimoğlu	- Leading the backend team. - Created the initial Django application and integrated Django Rest Framework. - Implemented Json Web Token authentication and login/register endpoints. - Implemented a simple events API to guide other members through API creation. - Using coreapi, generated automatic documentation/interface for our endpoints. - Created custom permissions that can be easily imported and used by other developers while creating their APIs. - Created a fully-fledged events API that adheres to the W3C standards. - Implemented location servies using the GeoDjango framework initially and googlemaps later. The endpoint currently supports filtering events by their distance to a given location. - Have been intermittently developing a generic search engine that employs the complex query systems Q of Django to allow for complex filtering options throughout the website. Currently supporting user search by keyword, first name or username. - I have deployed our to an AWS EC2 instance. It - Created an quick manual documentation for our endpoints annotated with the functionality they provide, with example JSON queries, so the frontend and android teams know what to implement next and how. - Started writing unit tests that will later be integrated as part of our CI/CD pipeline. Completed the tests for the authentication system
İrfan Bozkurt	CRUD API for event comments Models for notifications Tests for events, event comments, profiles, notifications
Mehmet Hilmi Dündar	Joined the team meetings. Learned about the rest framework. Set up the environment and run the backend in local and examine the backend code developed by team

	members. Analysed the requirements for equipment part of the project and created unit tests for initial equipment api.

FRONTEND

TEAM MEMBER	CONTRIBUTION
Yiğit Sarıoğlu	Leads the frontend team Implemented the “login page”, “register page”, “event create page”, “sell equipment page”, “find equipment page”. Implemented the user authorization part on the frontend side, integrated with backend Opened a page on the wiki (github) for the React JS-Documented the React and frontend side Used the documentation prepared by the backend team to access to the API endpoints Used POSTMAN to test the API endpoints HTTP requests are sent by using fetch() method. Implemented the routing mechanism of the website using React-Router Routing part was done with the “link to” feature of react. Was also responsible for reaching these pages using routing.
Yağmur Selek	Joined in frontend meetings on Google Meet. Implemented the home page I implemented the prior version of the event create page. I used the POSTMAN and test the API Participated to customer presentation
Ali Alperen Durak	Joined in frontend meetings on Google Meet. After reviewing project documents and studying the React basics, I designed and implemented the Events page with related components and modals. I connected API endpoints for events to the frontend after testing them via Postman. Implemented the search functionality to the Events page integrated with the backend’s search endpoint. Events page related routing work was performed. Participated in the customer presentation.

ANDROID

TEAM MEMBER	CONTRIBUTION
Fatih Akgöz	Created Android project and android-main branch from scratch. Implemented the logic and User Interface for Register, Login and Search Event pages. Added bottom navigation for future use. Build the navigation graphs for the current components. Joined team meetings and integration with backend meeting.
Ece Dilara Aslan	Reviewed requirements. Studied Kotlin, Android Studio and Volley library. Created connections for login, register, list events, search event and create event backend endpoints. Designed User Interface for Create Event Page. Joined team meetings and integration with backend meeting.

7. Deliverables

3.1 Functional Requirements

3.1.1 User Requirements

3.1.1.1 Sign in/up

- 3.1.1.1.1 User shall include his/her name, username, password, e-mail during sign-up.

3.1.1.2 Profile

- 3.1.1.2.1 User shall have interests with skill level, location, age and public badges shown on profile.
- 3.1.1.2.2 User shall be able to see the private activity history.
- 3.1.1.2.3 User shall be able to follow other users.
- 3.1.1.2.4 User shall be able to get notifications of the events that the users they follow create.

3.1.1.3 Settings

- 3.1.1.3.1 User shall be able to delete his/her account.
- 3.1.1.3.2 User shall be able to customize notifications.

3.1.1.4 Creating events

- 3.1.1.4.1 Any user shall be able to create events.
- 3.1.1.4.2 Event creator shall provide location, date, duration, and type of an event when creating an event.
- 3.1.1.4.3 Type of an event shall include its name, rules, number of people needed, required equipment, skill level.
- 3.1.1.4.4 If the desired type cannot be found, user shall be able to use characteristics to define an event.
- 3.1.1.4.5 Event creator shall be able to check the appliers' badges or related information for whether to accept them or not.
- 3.1.1.4.6 Event creator shall select the attendees out of appliers.
- 3.1.1.4.7 Event creator shall be able to create and give out badges to attendees.

3.1.1.5 Searching and attending an event

- 3.1.1.5.1 User shall provide at least 3 kinds of sports and skill level for searching.
- 3.1.1.5.2 User shall be able to search by location, date, type, people, skill level.
- 3.1.1.5.3 User shall be able to apply to participate in or spectate an event.
- 3.1.1.5.4 User shall be informed on whether he/she was accepted to the event.
- 3.1.1.5.5 After the event, user shall be able to give badges defined by event creators to other people.
- 3.1.1.5.6 Event creator shall be able to contact system maintainers for desired badge inclusion.
- 3.1.1.5.7 User shall be able to comment on the event page.

3.1.1.6 Buying products and other services

- 3.1.1.6.1 User shall be able to search for equipment sales by sport and location.
- 3.1.1.6.2 User shall be able to check information of equipment sellers and read descriptions written by them.
- 3.1.1.6.3 User shall be able to create equipment sale posts that include its description.

3.1.2 System Requirements

3.1.2.1 Sign in/up

- 3.1.2.1.1 The system shall provide users with options to create a new account or sign-in to an existing.

- 3.1.2.1.2 The system shall allow users to enter information about the kinds of sports they are interested in and their skill level while they are signing up.
- 3.1.2.1.3 The system should allow users to recover their password by sending a password-recovery e-mail.
- 3.1.2.1.4 The system shall verify the entered information before creating an account for registering users. Users who provide invalid information should not be allowed to create an account.

3.1.2.2 Search

- 3.1.2.2.1 The system shall allow users to search for events, according to event name, location, sport, date.
- 3.1.2.2.2 The system shall allow users to search for other users and view their location, schedule and interests if they have provided such information.
- 3.1.2.2.3 The system shall allow users to search for other users by name, view their profile and follow other users.
- 3.1.2.2.4 The system shall allow users to search for equipment for any sport they are interested in.

3.1.2.3 Events

- 3.1.2.3.1 The system shall provide information about the events such as event name, location, date, sport and the number of players.
- 3.1.2.3.2 The system shall maintain and provide analysis information about the events.
- 3.1.2.3.3 The system shall allow users to request joining an event.
- 3.1.2.3.4 The system shall allow users to create events by specifying event name, location, date, sport and the number of players.
- 3.1.2.3.5 The system shall provide default sport types for the event creators to select from.
- 3.1.2.3.6 If no matching sports type exist, the system shall allow event creators to specify the event characteristics.
- 3.1.2.3.7 The system shall send a pop-up message after a user creates an event.
- 3.1.2.3.8 The system shall allow users to cancel events they have created.

3.1.2.4 Recommendations

- 3.1.2.4.1 The system shall recommend users events based on their location and sports they are interested in.
- 3.1.2.4.2 The system shall recommend event creators other users based on their location, interests and schedules.

- 3.1.2.4.3 The system shall recommend sport-related content(gadgets, clothing, tools, merchandise) to users for the sports they are interested in.

3.1.2.5 Notifications

- 3.1.2.5.1 The system shall send notifications to event creators when a user sends a request to one of their events.
- 3.1.2.5.2 The system shall send notifications to users who are accepted or rejected for that event.
- 3.1.2.5.3 The system shall send notifications to users when one of the users they are following creates an event.

3.1.2.6 Badges

- 3.1.2.6.1 The system shall allow users to give badges to others users who they have joined an event together with.
- 3.1.2.6.2 The system shall allow event creators to give badges to users who have participated in their events.
- 3.1.2.6.3 The system shall provide a default selection of customizable badges.
- 3.1.2.6.4 The system should allow users to make requests for new badge types they would like to see available.

3.1.2.7 Interactions

- 3.1.2.7.1 The system shall allow users to block other users.
- 3.1.2.7.2 The system shall not allow users to view other users profiles who have blocked them.
- 3.1.2.7.3 The system should allow users to leave comments on the events they have participated in.

3.2 Nonfunctional Requirements

3.2.1 Availability

- 3.2.1.1 The system shall run as a web application, available through modern web browsers & an Android application.
- 3.2.1.2 Cross-platform languages shall not be used implementing the system.
- 3.2.1.3 The system language shall be English.

3.2.2 Standards

- 3.2.2.1 The system shall follow W3C Activity Stream Standard 2.0 to implement subscriptions & notifications
- 3.2.2.2 The **semantic taggings should be supported with Wikidata.**

3.2.3 Privacy

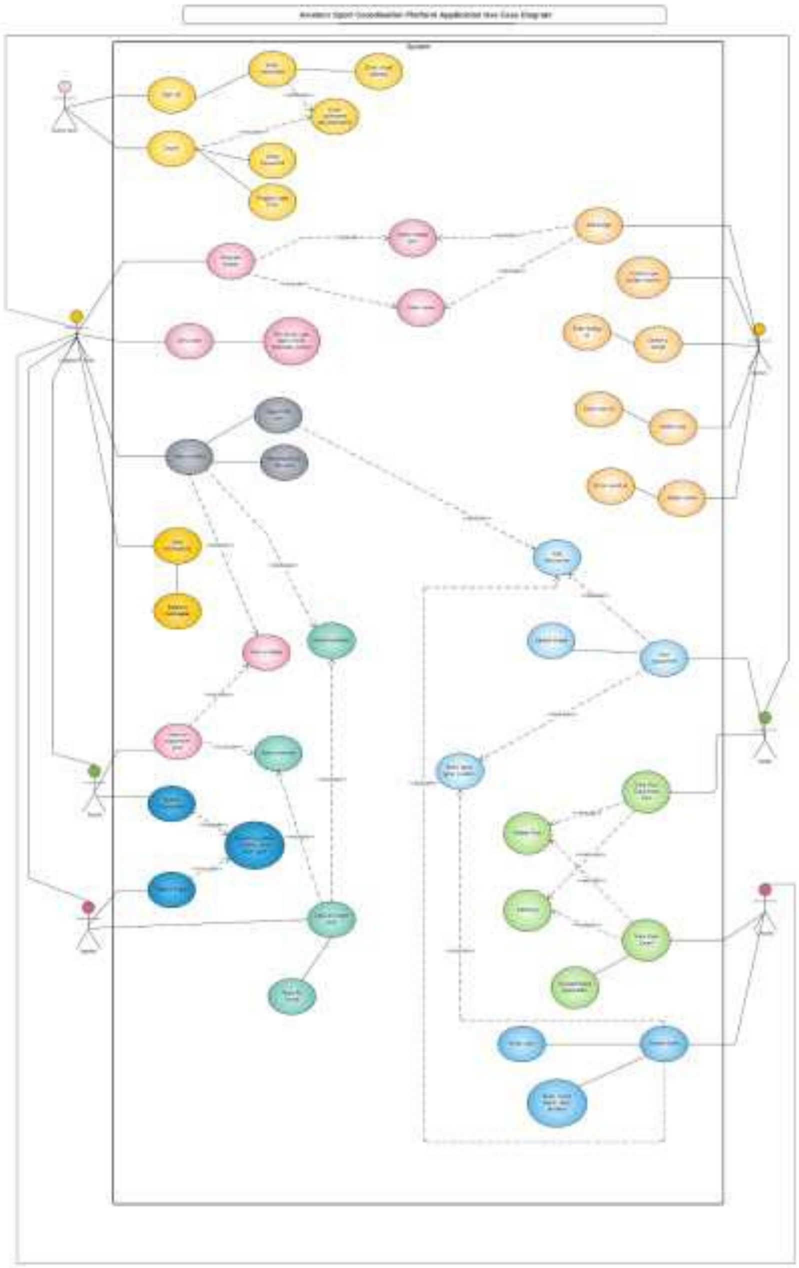
- 3.2.3.1 Users shall accept a privacy policy during a sign-up, setting permissions on personal data.
- 3.2.3.2 Users should be able to block each other, further specifying who can see their data.
- 3.2.3.3 Users shall not be asked to grant any information besides than a name & an e-mail address.
- 3.2.3.4 Data should be extracted out of what users share and allow to be visible, to analyze statistically and optimize user experience.
- 3.2.3.5 User data shall be stored and processed according to **General Data Protection Regulation(GDPR) and KVKK.**

3.2.4 Security

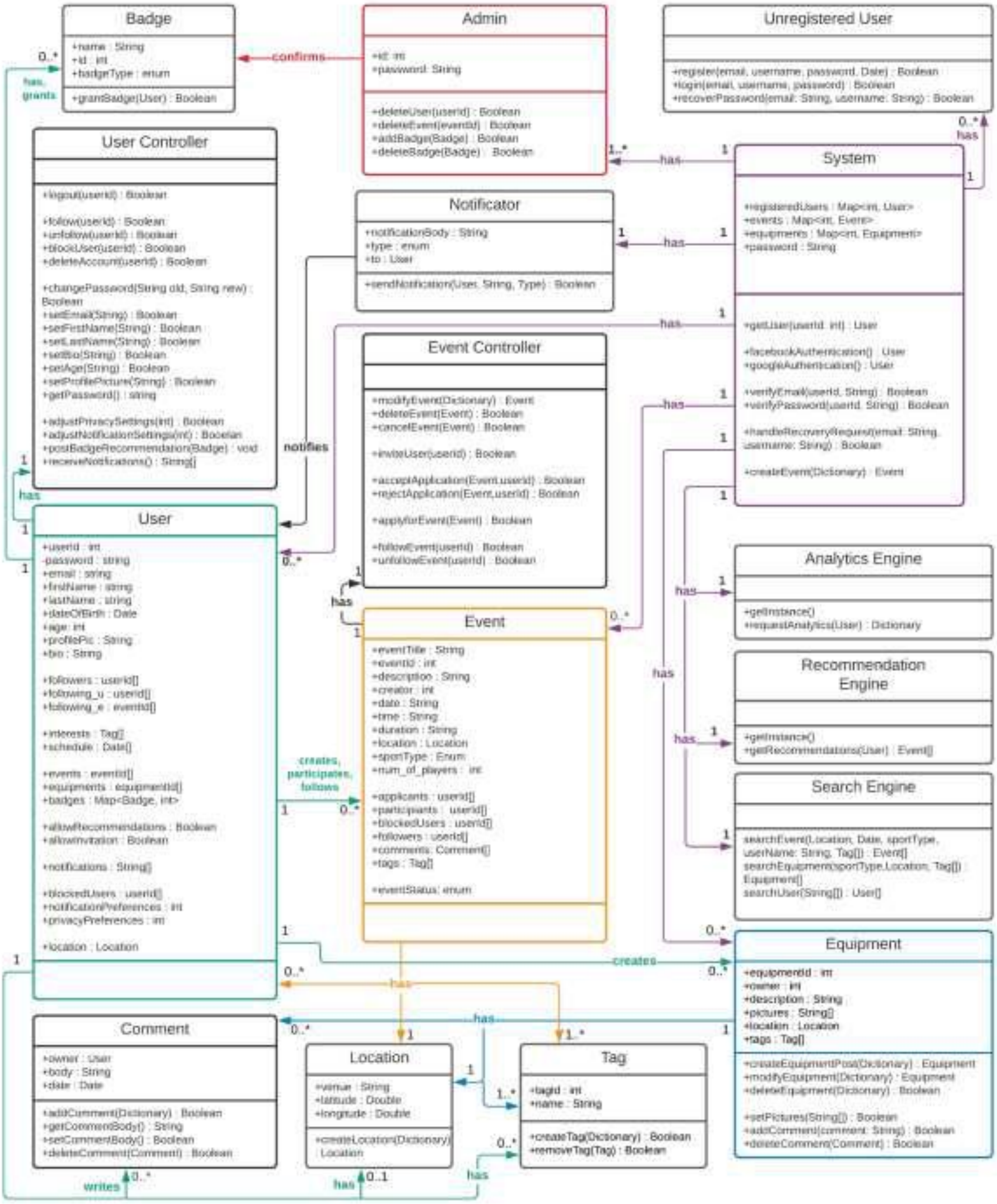
- 3.2.4.1 Users shall not be able to track transaction history of other users.
- 3.2.4.2 System shall not be accessible through old versions of browsers.
- 3.2.4.3 E-mail and password info for users shall be encrypted before storage.
- 3.2.4.4 No user shall be granted admin permissions.

UML Diagrams

Use Case Diagram


Amateur Sport Coordination Platform Application Use Case Diagram
System

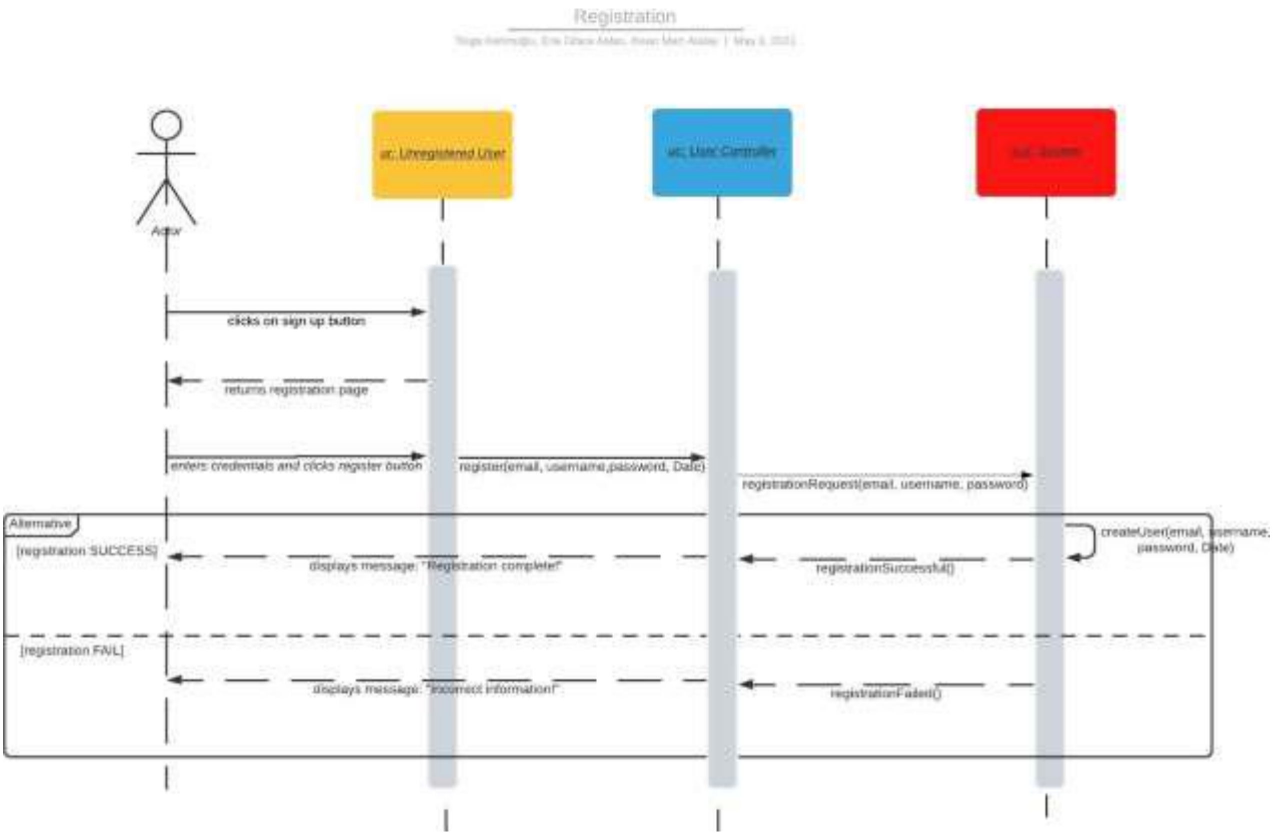
Class Diagram



Badge
+name : String
+id : int
+badgeType : enum
+grantBadge(User) : Boolean
Admin
+id: int
+password: String
+deleteUser(userId) : Boolean
+deleteEvent(eventId) : Boolean
+addBadge(Badge) : Boolean
+deleteBadge(Badge) : Boolean
confirms
Unregistered User
+register(email, username, password, Date) : Boolean
+login(email, username, password) : Boolean
+recoverPassword(email: String, username: String) : Boolean
User Controller
+logout(userId) : Boolean
+follow(userId) : Boolean
+unfollow(userId) : Boolean
+blockUser(userId) : Boolean
+deleteAccount(userId) : Boolean
+changePassword(String old, String new) : Boolean
+setEmail(String) : Boolean
+setFirstName(String) : Boolean
+setLastName(String) : Boolean
+setBio(String) : Boolean
+setAge(String) : Boolean
+setProfilePicture(String) : Boolean
+getPassword() : string
+adjustPrivacySettings(int) : Boolean
+adjustNotificationSettings(int) : Boolean
+postBadgeRecommendation(Badge) : void
+receiveNotifications() : String[]
Notificator
+notificationBody : String
+type : enum
+to : User
+sendNotification(User, String, Type) : Boolean
notifies
System
+registeredUsers : Map<int, User>
+events : Map<int, Event>
+equipments : Map<int, Equipment>
+password : String
+getUser(userId: int) : User
+facebookAuthentication() : User
+googleAuthentication() : User
+verifyEmail(userId, String) : Boolean
+verifyPassword(userId, String) : Boolean
+handleRecoveryRequest(email: String, username: String) : Boolean
+createEvent(Dictionary) : Event
Event Controller
+modifyEvent(Dictionary) : Event
+deleteEvent(Event) : Boolean
+cancelEvent(Event) : Boolean
+inviteUser(userId) : Boolean
+acceptApplication(Event,userId) : Boolean
+rejectApplication(Event,userId) : Boolean
+applyforEvent(Event) : Boolean
+followEvent(userId) : Boolean
+unfollowEvent(userId) : Boolean
User
+userId : int
-password : string
+email : string
+firstName : string
+lastName : string
+dateOfBirth : Date
+age: int
+profilePic : String
+bio : String
+followers : userId[]
+following_u : userId[]
+following_e : eventId[]
+interests : Tag[]
+schedule : Date[]
+events : eventId[]
+equipments : equipmentId[]
+badges : Map<Badge, int>
+allowRecommendations : Boolean
+allowInvitation : Boolean
+notifications : String[]
+blockedUsers : userId[]
+notificationPreferences : int
+privacyPreferences : int
+location : Location
Event
+eventTitle : String
+eventId : int
+description : String
+creator : int
+date : String
+time : String
+duration : String
+location : Location
+sportType : Enum
+num_of_players : int
+applicants : userId[]
+participants : userId[]
+blockedUsers : userId[]
+followers : userId[]
+comments: Comment[]
+tags : Tag[]
+eventStatus: enum
creates, participates, follows
Analytics Engine
+getInstance()
+requestAnalytics(User) : Dictionary
Recommendation Engine
+getInstance()
+getRecommendations(User) : Event[]
Search Engine
searchEvent(Location, Date, sportType, userName: String, Tag[]) : Event[]
searchEquipment(sportType,Location, Tag[]) : Equipment[]
searchUser(String[]) : User[]
Equipment
+equipmentId : int
+owner : int
+description : String
+pictures : String[]
+location : Location
+tags : Tag[]
+createEquipmentPost(Dictionary) : Equipment
+modifyEquipment(Dictionary) : Equipment
+deleteEquipment(Dictionary) : Boolean
+setPictures(String[]) : Boolean
+addComment(comment: String) : Boolean
+deleteComment(Comment) : Boolean
creates
Comment
+owner : User
+body : String
+date : Date
+addComment(Dictionary) : Boolean
+getCommentBody() : String
+setCommentBody() : Boolean
+deleteComment(Comment) : Boolean
Location
+venue : String
+latitude : Double
+longitude : Double
+createLocation(Dictionary) : Location
Tag
+tagId : int
+name : String
+createTag(Dictionary) : Boolean
+removeTag(Tag) : Boolean
has
has, grants
writes
0..*
0..1
1
1..*

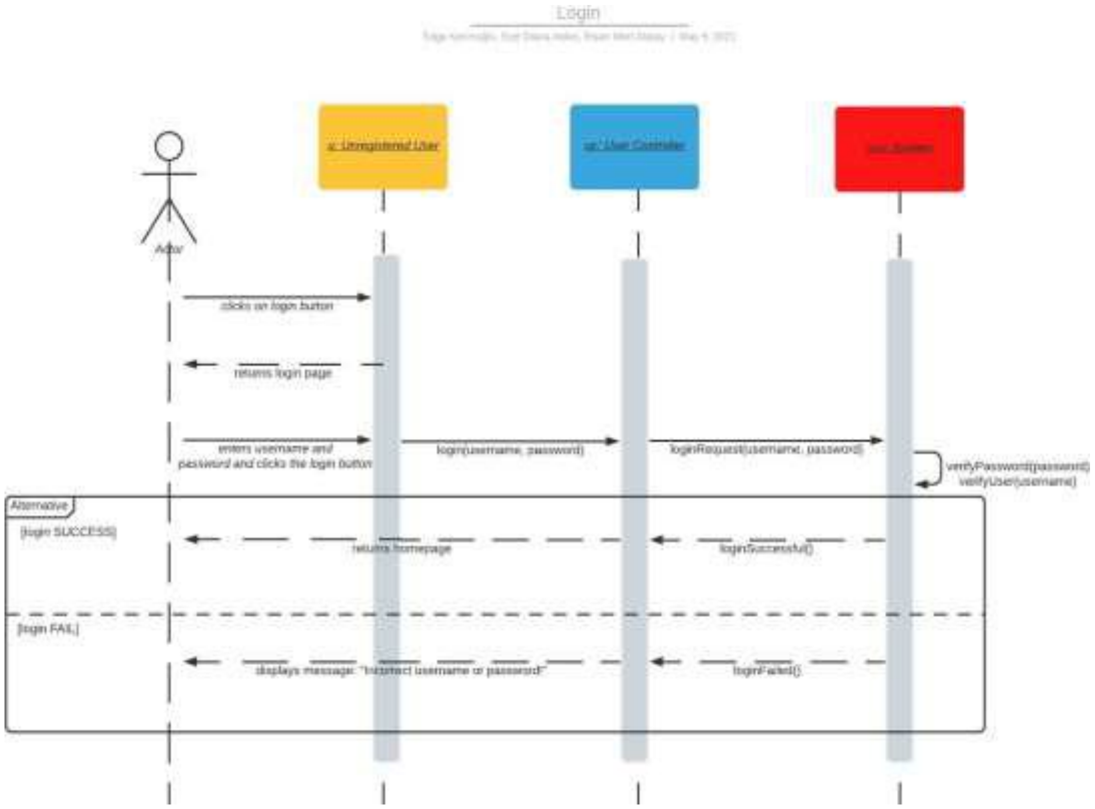
Sequence Diagrams

1. Registration
   Shows the sequence of interactions between associated objects while a user registers to the system from the homepage.


Registration
Actor
uc: Unregistered User
uc: User Controller
clicks on sign up button
returns registration page
enters credentials and clicks register button
register(email, username,password, Date)
registrationRequest(email, username, password)
Alternative
[registration SUCCESS]
createUser(email, username, password, Date)
displays message: "Registration complete!"
registrationSuccessful()
[registration FAIL]
displays message: "Incorrect information!"
registrationFailed()

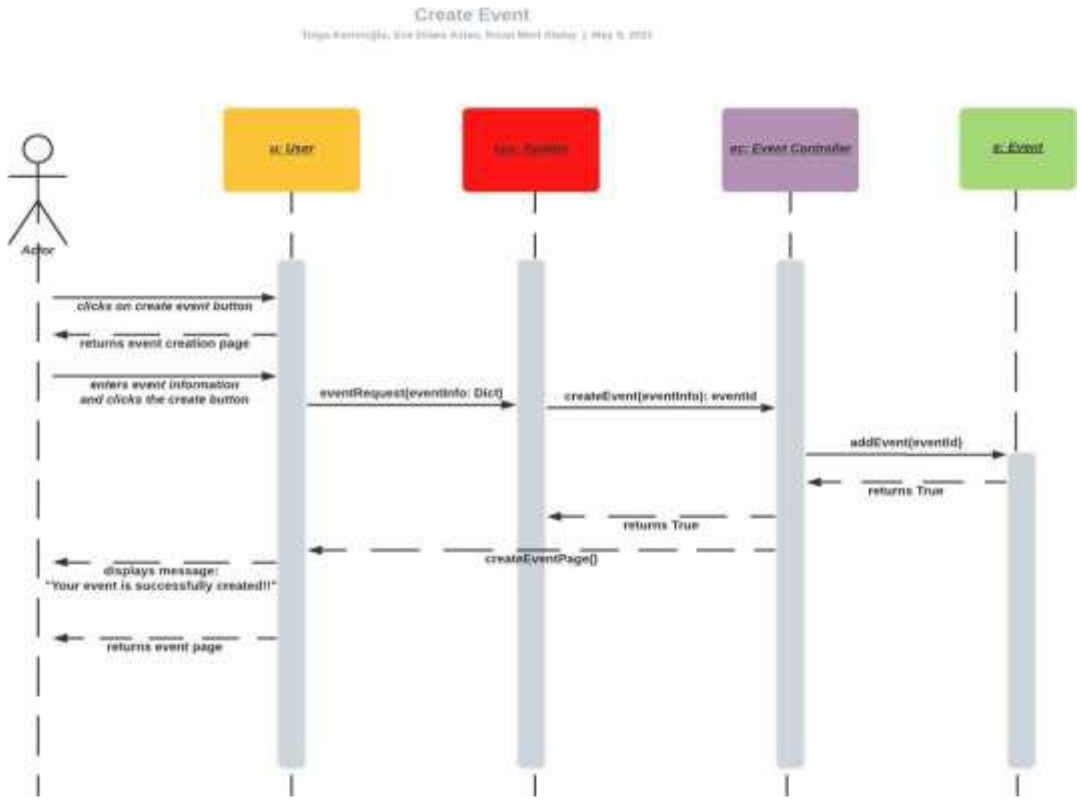
2. Login

   Shows the sequence of interactions between associated objects while a user logs in to his/her account from the homepage.



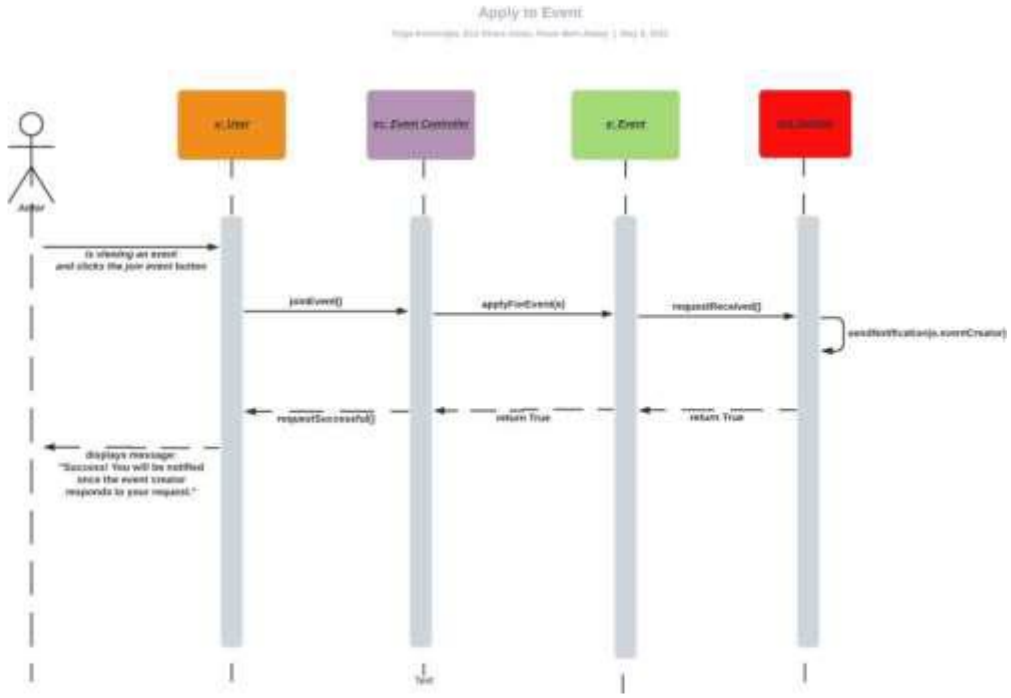
3. Create Event

   Shows the sequence of interactions between associated objects while a user creates an event where he/she had already logged in and is in the homepage.


Create Event
u: User
sys: System
ec: Event Controller
e: Event
Actor
clicks on create event button
returns event creation page
enters event information
and clicks the create button
eventRequest(eventInfo: Dict)
createEvent(eventInfo): eventId
addEvent(eventId)
returns True
returns True
createEventPage()
displays message:
"Your event is successfully created!!"
returns event page

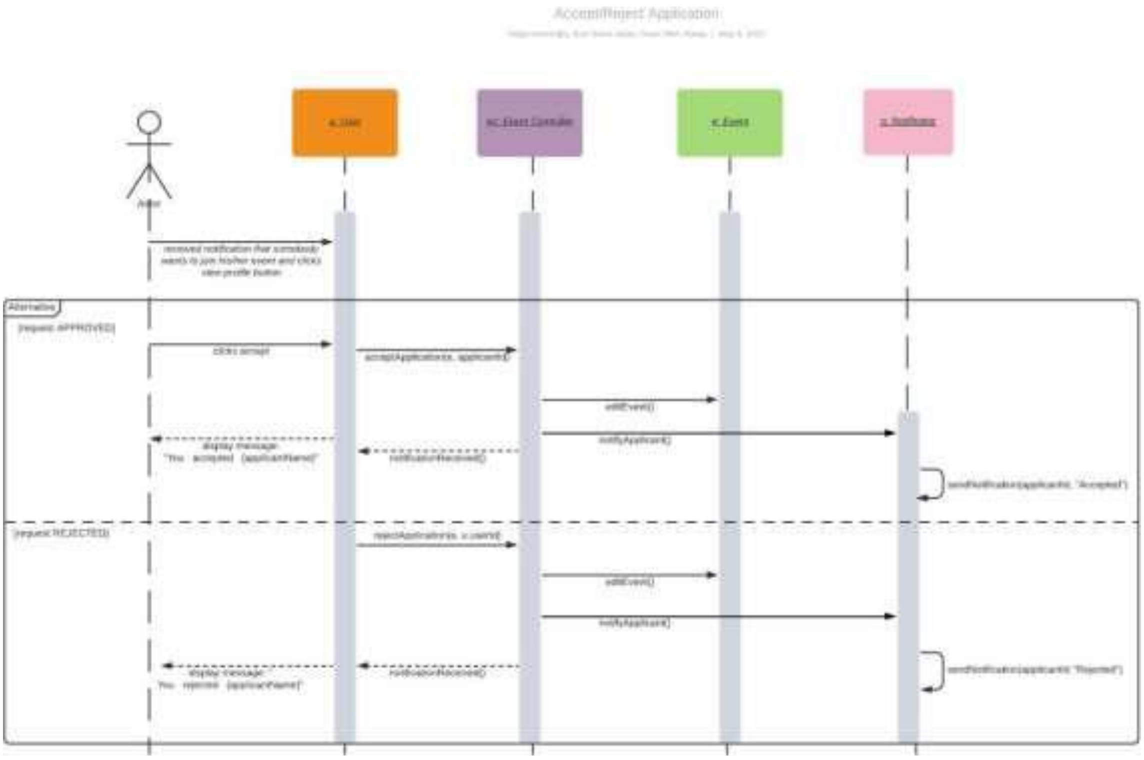
4. Apply to event

   Shows the sequence of interactions between associated objects when a user applies to join to an event.



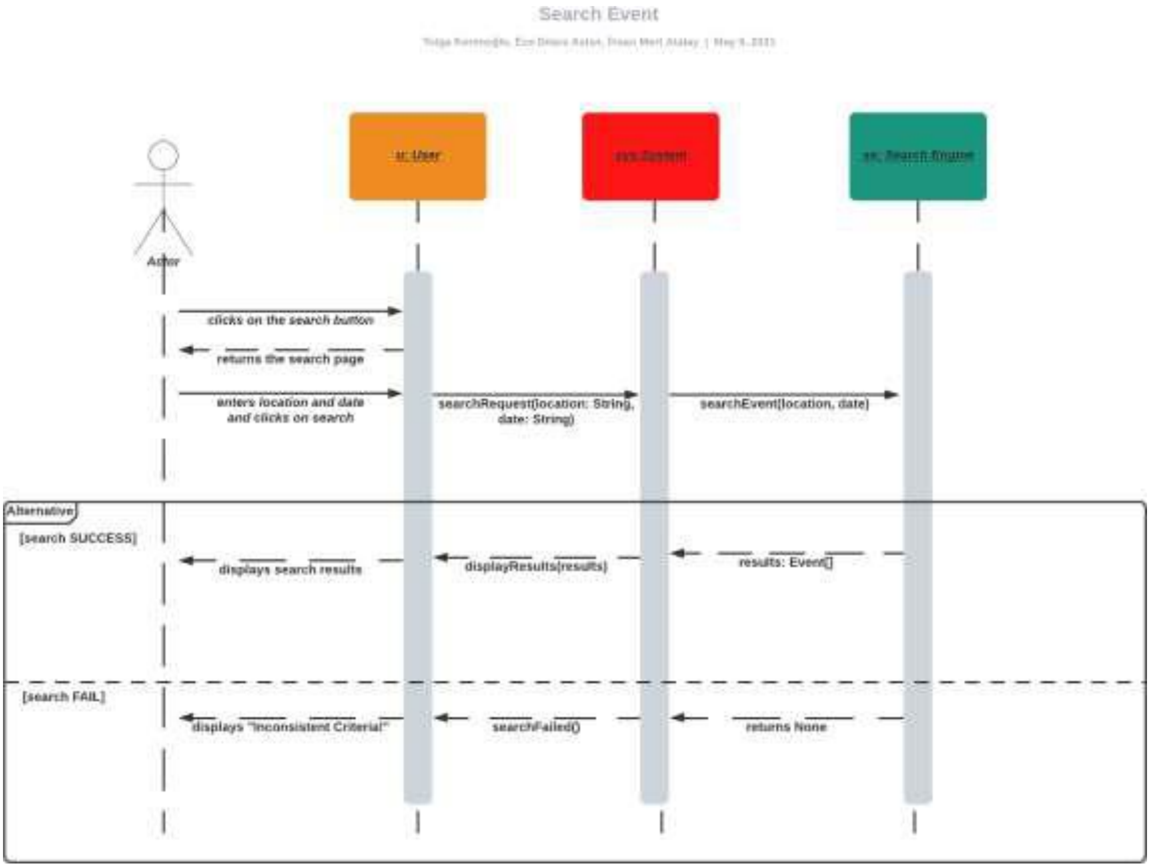
5. Accept/Reject Applicant

   Shows the sequence of interactions between associated objects when a user accepts/rejects a user to an event he/she created.


Accept/Reject Application
u: User
ec: Event Controller
e: Event
n: Notification
Actor
received notification that somebody wants to join his/her event and clicks view profile button
Alternative
[request APPROVED]
clicks accept
acceptApplication(e, applicantId)
editEvent()
notifyApplicant()
display message: "You accepted {applicantName}"
notificationReceived()
sendNotification(applicantId, "Accepted")
[request REJECTED]
rejectApplication(e, u.userId)
editEvent()
notifyApplicant()
display message: "You rejected {applicantName}"
notificationReceived()
sendNotification(applicantId "Rejected")

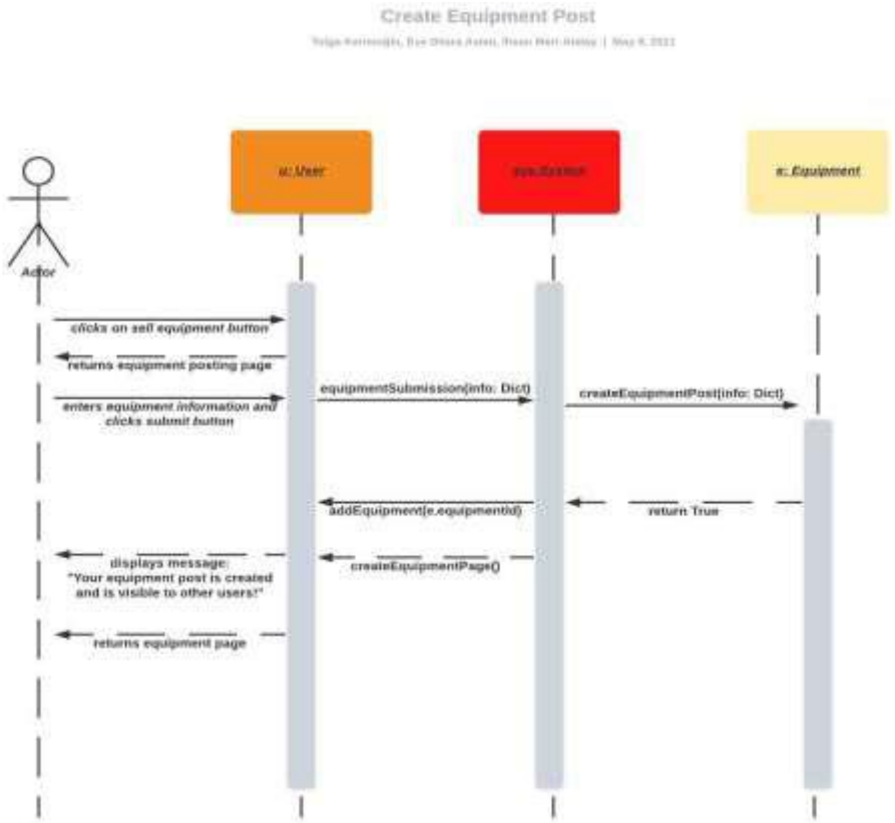
6. Search Event

Shows the sequence of interactions between associated objects while a user searches for an event by location and date after he/she has already logged in.


Search Event
Actor
clicks on the search button
returns the search page
enters location and date and clicks on search
searchRequest(location: String, date: String)
searchEvent(location, date)
Alternative
[search SUCCESS]
displays search results
displayResults(results)
results: Event[]
[search FAIL]
displays "Inconsistent Criteria!"
searchFailed()
returns None

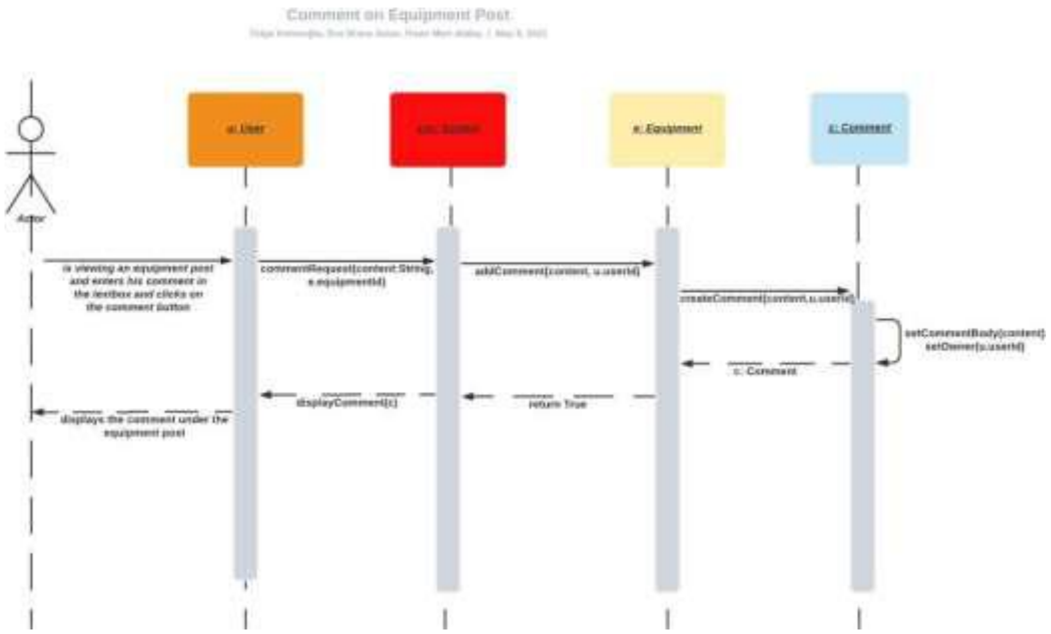
7. Create Equipment Post

   Shows the sequence of interactions between associated objects while a user creates an equipment post to sell equipment after he/she has already logged in.


Create Equipment Post
u: User
e: Equipment
Actor
clicks on sell equipment button
returns equipment posting page
enters equipment information and clicks submit button
equipmentSubmission(info: Dict)
createEquipmentPost(info: Dict)
addEquipment(e.equipmentId)
return True
displays message: "Your equipment post is created and is visible to other users!"
createEquipmentPage()
returns equipment page

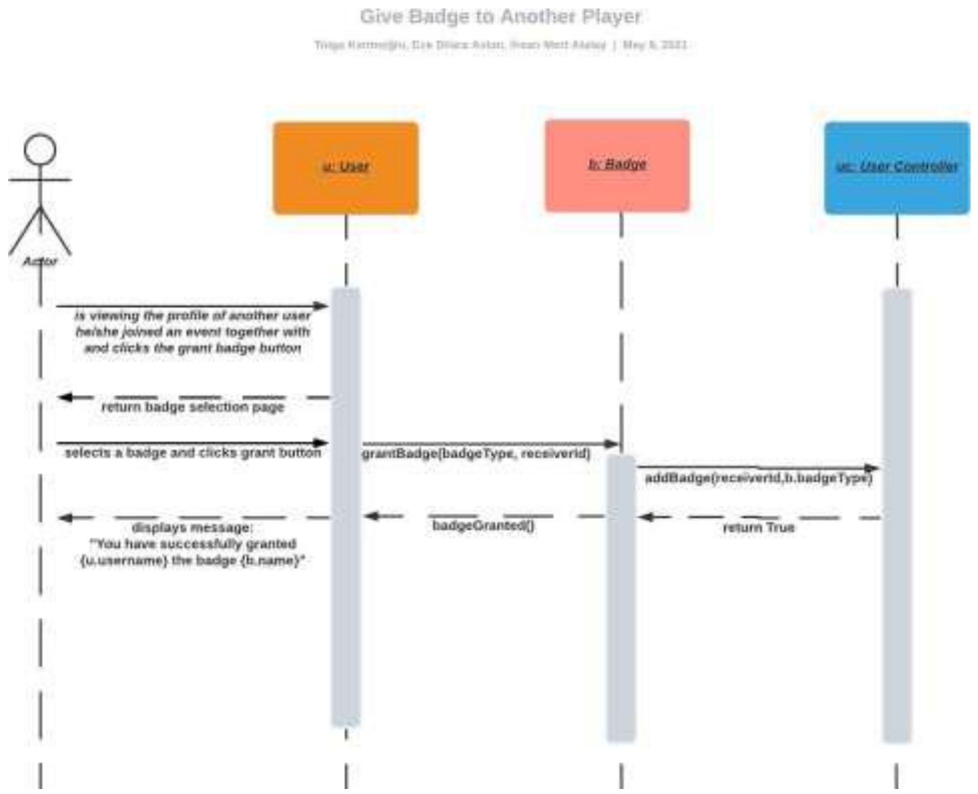
8. Comment on Equipment

   Shows the sequence of interactions between associated objects when a user submits a comment for an equipment while he/she is already viewing the equipments post.

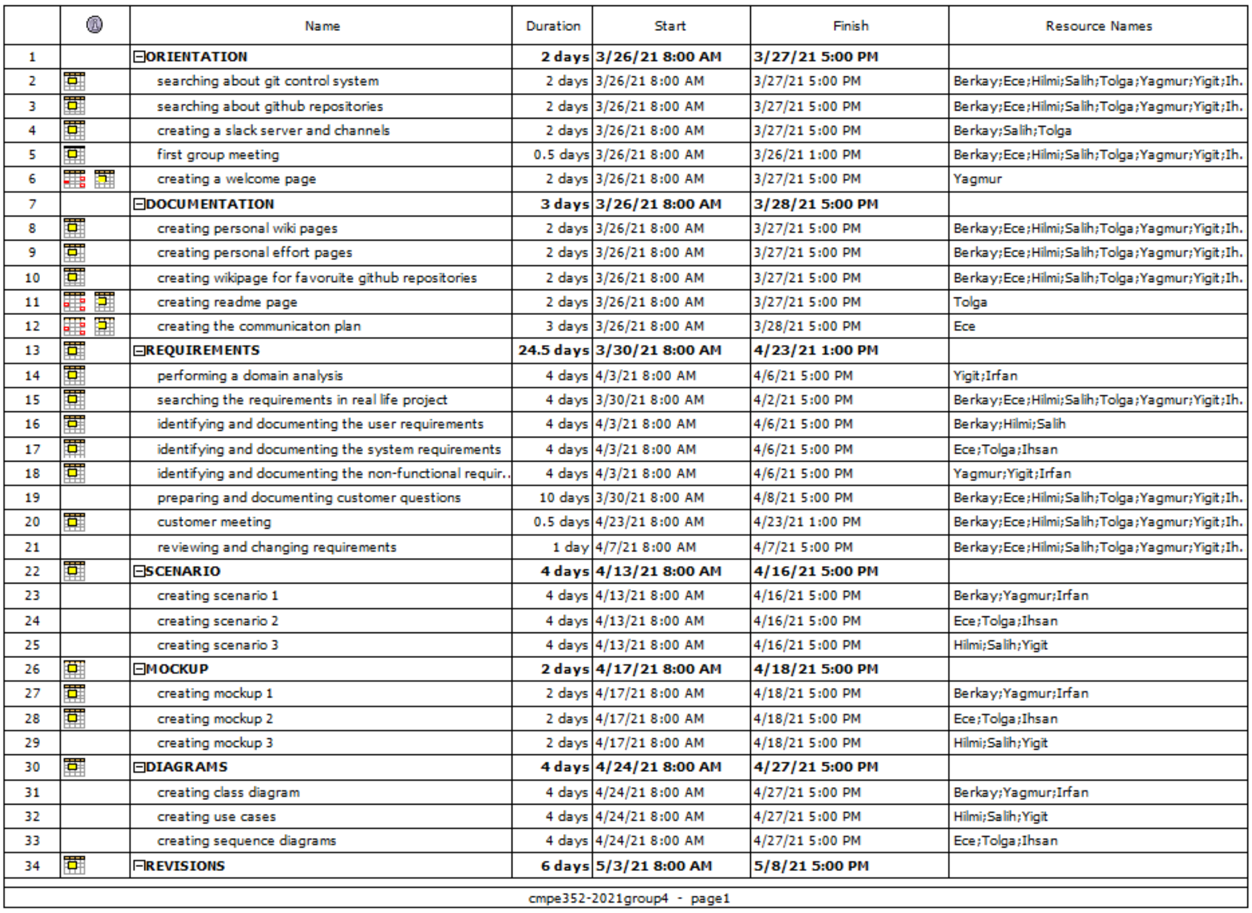

Comment on Equipment Post
Actor
u: User
e: Equipment
c: Comment
is viewing an equipment post and enters his comment in the textbox and clicks on the comment button
commentRequest(content:String, e.equipmentId)
addComment(content, u.userId)
createComment(content,u.userId)
setCommentBody(content)
setOwner(u.userId)
c: Comment
return True
displayComment(c)
displays the comment under the equipment post

9. Give Badge to Another Player

Shows the sequence of interactions between associated objects when a user gives a badge to another user he/she has joined an event together with while he/she is already viewing the users profile.

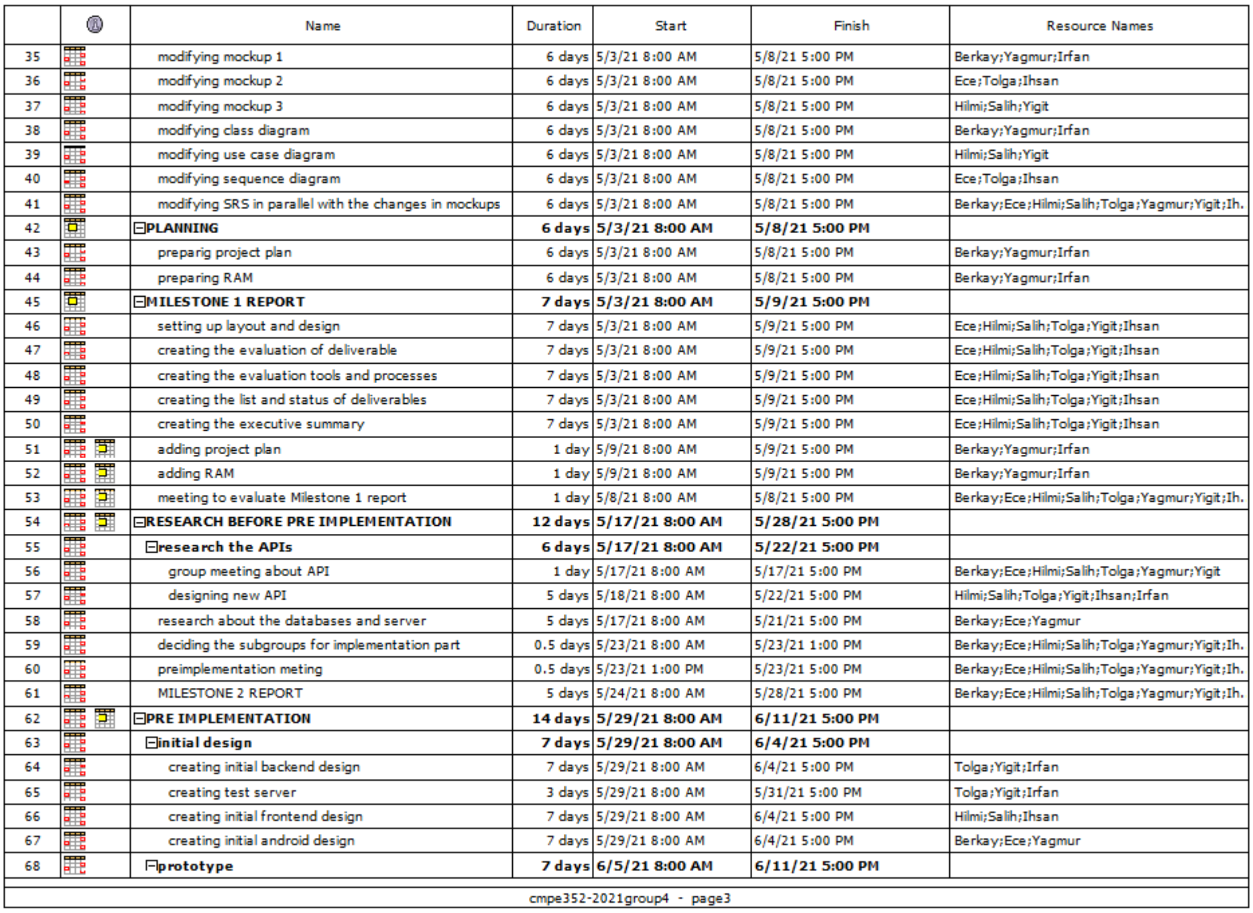

Give Badge to Another Player
Actor
u: User
b: Badge
uc: User Controller
is viewing the profile of another user he/she joined an event together with and clicks the grant badge button
return badge selection page
selects a badge and clicks grant button
grantBadge(badgeType, receiverId)
addBadge(receiverId,b.badgeType)
displays message: "You have successfully granted {u.username} the badge {b.name}"
badgeGranted()
return True

Project Plan



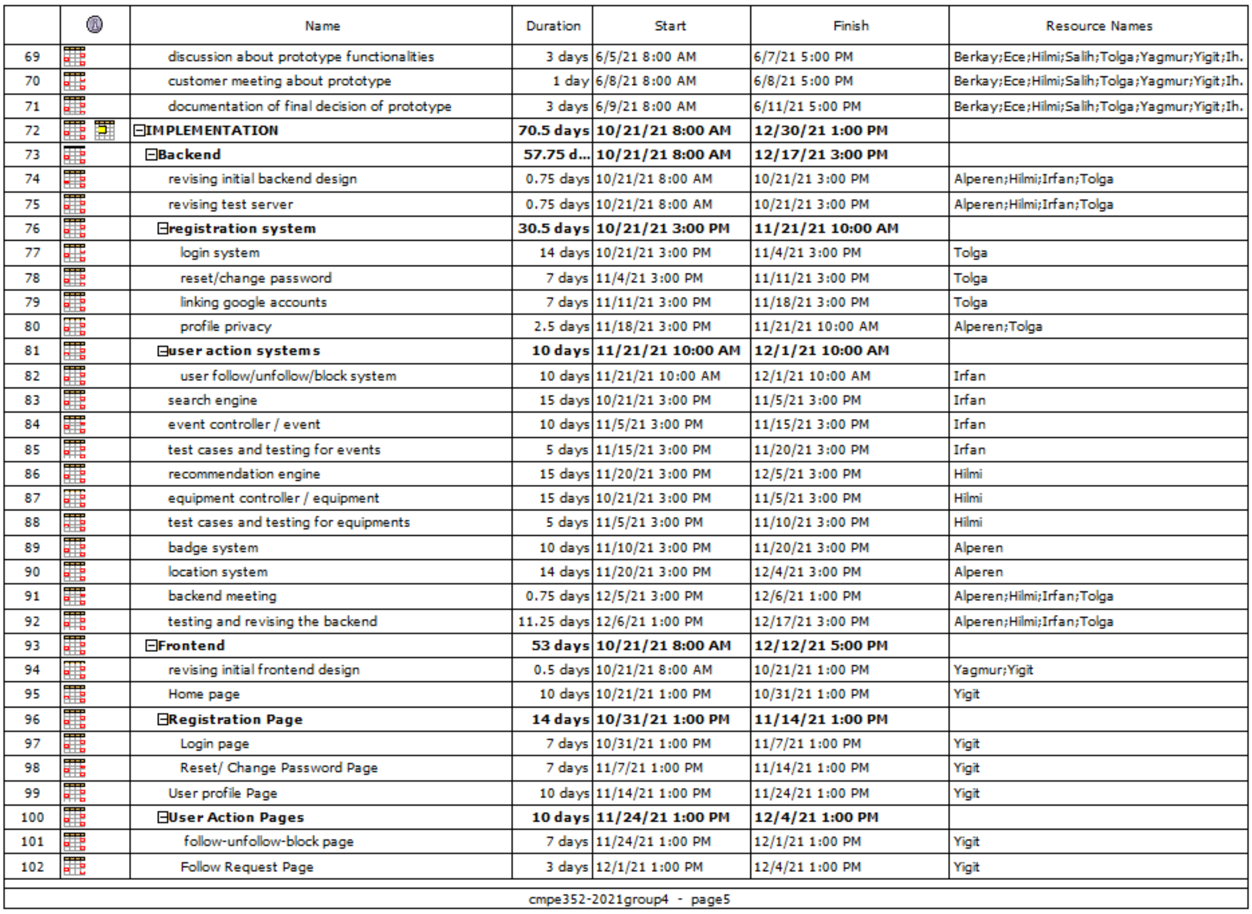
	ⓘ	Name	Duration	Start	Finish	Resource Names
1		⊟**ORIENTATION**	**2 days**	**3/26/21 8:00 AM**	**3/27/21 5:00 PM**	
2	▦	searching about git control system	2 days	3/26/21 8:00 AM	3/27/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
3	▦	searching about github repositories	2 days	3/26/21 8:00 AM	3/27/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
4	▦	creating a slack server and channels	2 days	3/26/21 8:00 AM	3/27/21 5:00 PM	Berkay;Salih;Tolga
5	▦	first group meeting	0.5 days	3/26/21 8:00 AM	3/26/21 1:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
6	▦ ▦	creating a welcome page	2 days	3/26/21 8:00 AM	3/27/21 5:00 PM	Yagmur
7		⊟**DOCUMENTATION**	**3 days**	**3/26/21 8:00 AM**	**3/28/21 5:00 PM**	
8	▦	creating personal wiki pages	2 days	3/26/21 8:00 AM	3/27/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
9	▦	creating personal effort pages	2 days	3/26/21 8:00 AM	3/27/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
10	▦	creating wikipage for favoruite github repositories	2 days	3/26/21 8:00 AM	3/27/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
11	▦ ▦	creating readme page	2 days	3/26/21 8:00 AM	3/27/21 5:00 PM	Tolga
12	▦ ▦	creating the communicaton plan	3 days	3/26/21 8:00 AM	3/28/21 5:00 PM	Ece
13	▦	⊟**REQUIREMENTS**	**24.5 days**	**3/30/21 8:00 AM**	**4/23/21 1:00 PM**	
14	▦	performing a domain analysis	4 days	4/3/21 8:00 AM	4/6/21 5:00 PM	Yigit;Irfan
15	▦	searching the requirements in real life project	4 days	3/30/21 8:00 AM	4/2/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
16	▦	identifying and documenting the user requirements	4 days	4/3/21 8:00 AM	4/6/21 5:00 PM	Berkay;Hilmi;Salih
17	▦	identifying and documenting the system requirements	4 days	4/3/21 8:00 AM	4/6/21 5:00 PM	Ece;Tolga;Ihsan
18	▦	identifying and documenting the non-functional requir..	4 days	4/3/21 8:00 AM	4/6/21 5:00 PM	Yagmur;Yigit;Irfan
19		preparing and documenting customer questions	10 days	3/30/21 8:00 AM	4/8/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
20	▦	customer meeting	0.5 days	4/23/21 8:00 AM	4/23/21 1:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
21		reviewing and changing requirements	1 day	4/7/21 8:00 AM	4/7/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
22	▦	⊟**SCENARIO**	**4 days**	**4/13/21 8:00 AM**	**4/16/21 5:00 PM**	
23		creating scenario 1	4 days	4/13/21 8:00 AM	4/16/21 5:00 PM	Berkay;Yagmur;Irfan
24		creating scenario 2	4 days	4/13/21 8:00 AM	4/16/21 5:00 PM	Ece;Tolga;Ihsan
25		creating scenario 3	4 days	4/13/21 8:00 AM	4/16/21 5:00 PM	Hilmi;Salih;Yigit
26	▦	⊟**MOCKUP**	**2 days**	**4/17/21 8:00 AM**	**4/18/21 5:00 PM**	
27	▦	creating mockup 1	2 days	4/17/21 8:00 AM	4/18/21 5:00 PM	Berkay;Yagmur;Irfan
28	▦	creating mockup 2	2 days	4/17/21 8:00 AM	4/18/21 5:00 PM	Ece;Tolga;Ihsan
29		creating mockup 3	2 days	4/17/21 8:00 AM	4/18/21 5:00 PM	Hilmi;Salih;Yigit
30	▦	⊟**DIAGRAMS**	**4 days**	**4/24/21 8:00 AM**	**4/27/21 5:00 PM**	
31		creating class diagram	4 days	4/24/21 8:00 AM	4/27/21 5:00 PM	Berkay;Yagmur;Irfan
32		creating use cases	4 days	4/24/21 8:00 AM	4/27/21 5:00 PM	Hilmi;Salih;Yigit
33		creating sequence diagrams	4 days	4/24/21 8:00 AM	4/27/21 5:00 PM	Ece;Tolga;Ihsan
34	▦	⊟**REVISIONS**	**6 days**	**5/3/21 8:00 AM**	**5/8/21 5:00 PM**	

cmpe352-2021group4 - page1



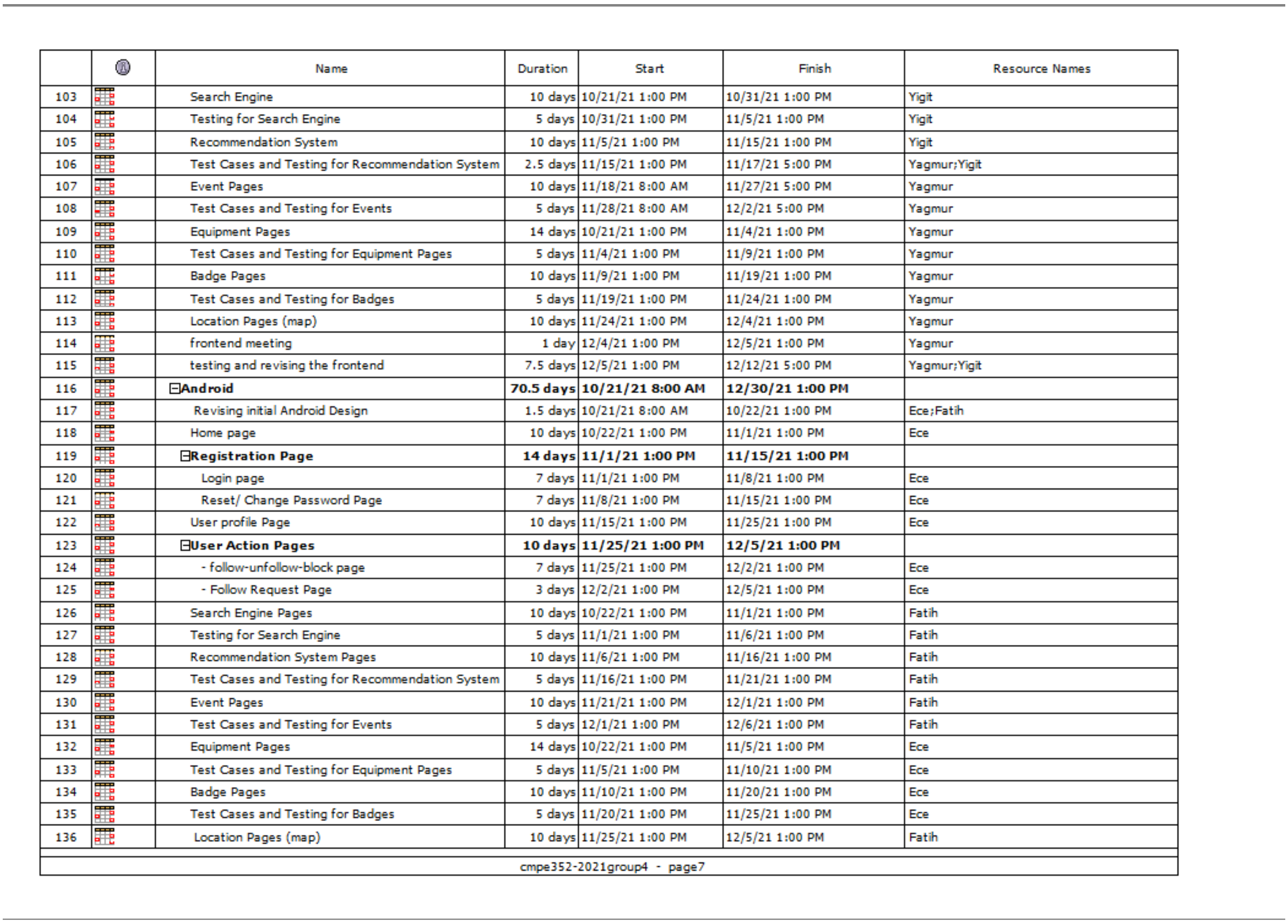
		Name	Duration	Start	Finish	Resource Names
35		modifying mockup 1	6 days	5/3/21 8:00 AM	5/8/21 5:00 PM	Berkay;Yagmur;Irfan
36		modifying mockup 2	6 days	5/3/21 8:00 AM	5/8/21 5:00 PM	Ece;Tolga;Ihsan
37		modifying mockup 3	6 days	5/3/21 8:00 AM	5/8/21 5:00 PM	Hilmi;Salih;Yigit
38		modifying class diagram	6 days	5/3/21 8:00 AM	5/8/21 5:00 PM	Berkay;Yagmur;Irfan
39		modifying use case diagram	6 days	5/3/21 8:00 AM	5/8/21 5:00 PM	Hilmi;Salih;Yigit
40		modifying sequence diagram	6 days	5/3/21 8:00 AM	5/8/21 5:00 PM	Ece;Tolga;Ihsan
41		modifying SRS in parallel with the changes in mockups	6 days	5/3/21 8:00 AM	5/8/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
42		**⊟PLANNING**	**6 days**	**5/3/21 8:00 AM**	**5/8/21 5:00 PM**	
43		preparig project plan	6 days	5/3/21 8:00 AM	5/8/21 5:00 PM	Berkay;Yagmur;Irfan
44		preparing RAM	6 days	5/3/21 8:00 AM	5/8/21 5:00 PM	Berkay;Yagmur;Irfan
45		**⊟MILESTONE 1 REPORT**	**7 days**	**5/3/21 8:00 AM**	**5/9/21 5:00 PM**	
46		setting up layout and design	7 days	5/3/21 8:00 AM	5/9/21 5:00 PM	Ece;Hilmi;Salih;Tolga;Yigit;Ihsan
47		creating the evaluation of deliverable	7 days	5/3/21 8:00 AM	5/9/21 5:00 PM	Ece;Hilmi;Salih;Tolga;Yigit;Ihsan
48		creating the evaluation tools and processes	7 days	5/3/21 8:00 AM	5/9/21 5:00 PM	Ece;Hilmi;Salih;Tolga;Yigit;Ihsan
49		creating the list and status of deliverables	7 days	5/3/21 8:00 AM	5/9/21 5:00 PM	Ece;Hilmi;Salih;Tolga;Yigit;Ihsan
50		creating the executive summary	7 days	5/3/21 8:00 AM	5/9/21 5:00 PM	Ece;Hilmi;Salih;Tolga;Yigit;Ihsan
51		adding project plan	1 day	5/9/21 8:00 AM	5/9/21 5:00 PM	Berkay;Yagmur;Irfan
52		adding RAM	1 day	5/9/21 8:00 AM	5/9/21 5:00 PM	Berkay;Yagmur;Irfan
53		meeting to evaluate Milestone 1 report	1 day	5/8/21 8:00 AM	5/8/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
54		**⊟RESEARCH BEFORE PRE IMPLEMENTATION**	**12 days**	**5/17/21 8:00 AM**	**5/28/21 5:00 PM**	
55		**⊟research the APIs**	**6 days**	**5/17/21 8:00 AM**	**5/22/21 5:00 PM**	
56		group meeting about API	1 day	5/17/21 8:00 AM	5/17/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit
57		designing new API	5 days	5/18/21 8:00 AM	5/22/21 5:00 PM	Hilmi;Salih;Tolga;Yigit;Ihsan;Irfan
58		research about the databases and server	5 days	5/17/21 8:00 AM	5/21/21 5:00 PM	Berkay;Ece;Yagmur
59		deciding the subgroups for implementation part	0.5 days	5/23/21 8:00 AM	5/23/21 1:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
60		preimplementation meting	0.5 days	5/23/21 1:00 PM	5/23/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
61		MILESTONE 2 REPORT	5 days	5/24/21 8:00 AM	5/28/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
62		**⊟PRE IMPLEMENTATION**	**14 days**	**5/29/21 8:00 AM**	**6/11/21 5:00 PM**	
63		**⊟initial design**	**7 days**	**5/29/21 8:00 AM**	**6/4/21 5:00 PM**	
64		creating initial backend design	7 days	5/29/21 8:00 AM	6/4/21 5:00 PM	Tolga;Yigit;Irfan
65		creating test server	3 days	5/29/21 8:00 AM	5/31/21 5:00 PM	Tolga;Yigit;Irfan
66		creating initial frontend design	7 days	5/29/21 8:00 AM	6/4/21 5:00 PM	Hilmi;Salih;Ihsan
67		creating initial android design	7 days	5/29/21 8:00 AM	6/4/21 5:00 PM	Berkay;Ece;Yagmur
68		**⊟prototype**	**7 days**	**6/5/21 8:00 AM**	**6/11/21 5:00 PM**	

cmpe352-2021group4 - page3

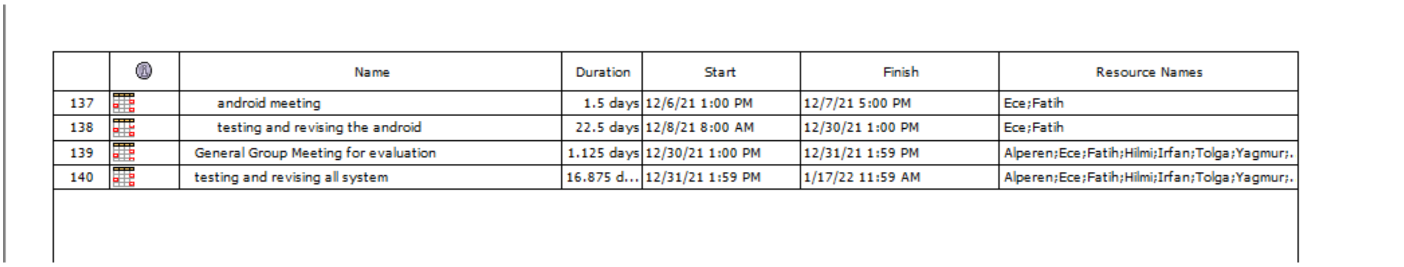


		Name	Duration	Start	Finish	Resource Names
69		discussion about prototype functionalities	3 days	6/5/21 8:00 AM	6/7/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
70		customer meeting about prototype	1 day	6/8/21 8:00 AM	6/8/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
71		documentation of final decision of prototype	3 days	6/9/21 8:00 AM	6/11/21 5:00 PM	Berkay;Ece;Hilmi;Salih;Tolga;Yagmur;Yigit;Ih.
72		**⊟IMPLEMENTATION**	**70.5 days**	**10/21/21 8:00 AM**	**12/30/21 1:00 PM**	
73		**⊟Backend**	**57.75 d...**	**10/21/21 8:00 AM**	**12/17/21 3:00 PM**	
74		revising initial backend design	0.75 days	10/21/21 8:00 AM	10/21/21 3:00 PM	Alperen;Hilmi;Irfan;Tolga
75		revising test server	0.75 days	10/21/21 8:00 AM	10/21/21 3:00 PM	Alperen;Hilmi;Irfan;Tolga
76		**⊟registration system**	**30.5 days**	**10/21/21 3:00 PM**	**11/21/21 10:00 AM**	
77		login system	14 days	10/21/21 3:00 PM	11/4/21 3:00 PM	Tolga
78		reset/change password	7 days	11/4/21 3:00 PM	11/11/21 3:00 PM	Tolga
79		linking google accounts	7 days	11/11/21 3:00 PM	11/18/21 3:00 PM	Tolga
80		profile privacy	2.5 days	11/18/21 3:00 PM	11/21/21 10:00 AM	Alperen;Tolga
81		**⊟user action systems**	**10 days**	**11/21/21 10:00 AM**	**12/1/21 10:00 AM**	
82		user follow/unfollow/block system	10 days	11/21/21 10:00 AM	12/1/21 10:00 AM	Irfan
83		search engine	15 days	10/21/21 3:00 PM	11/5/21 3:00 PM	Irfan
84		event controller / event	10 days	11/5/21 3:00 PM	11/15/21 3:00 PM	Irfan
85		test cases and testing for events	5 days	11/15/21 3:00 PM	11/20/21 3:00 PM	Irfan
86		recommendation engine	15 days	11/20/21 3:00 PM	12/5/21 3:00 PM	Hilmi
87		equipment controller / equipment	15 days	10/21/21 3:00 PM	11/5/21 3:00 PM	Hilmi
88		test cases and testing for equipments	5 days	11/5/21 3:00 PM	11/10/21 3:00 PM	Hilmi
89		badge system	10 days	11/10/21 3:00 PM	11/20/21 3:00 PM	Alperen
90		location system	14 days	11/20/21 3:00 PM	12/4/21 3:00 PM	Alperen
91		backend meeting	0.75 days	12/5/21 3:00 PM	12/6/21 1:00 PM	Alperen;Hilmi;Irfan;Tolga
92		testing and revising the backend	11.25 days	12/6/21 1:00 PM	12/17/21 3:00 PM	Alperen;Hilmi;Irfan;Tolga
93		**⊟Frontend**	**53 days**	**10/21/21 8:00 AM**	**12/12/21 5:00 PM**	
94		revising initial frontend design	0.5 days	10/21/21 8:00 AM	10/21/21 1:00 PM	Yagmur;Yigit
95		Home page	10 days	10/21/21 1:00 PM	10/31/21 1:00 PM	Yigit
96		**⊟Registration Page**	**14 days**	**10/31/21 1:00 PM**	**11/14/21 1:00 PM**	
97		Login page	7 days	10/31/21 1:00 PM	11/7/21 1:00 PM	Yigit
98		Reset/ Change Password Page	7 days	11/7/21 1:00 PM	11/14/21 1:00 PM	Yigit
99		User profile Page	10 days	11/14/21 1:00 PM	11/24/21 1:00 PM	Yigit
100		**⊟User Action Pages**	**10 days**	**11/24/21 1:00 PM**	**12/4/21 1:00 PM**	
101		follow-unfollow-block page	7 days	11/24/21 1:00 PM	12/1/21 1:00 PM	Yigit
102		Follow Request Page	3 days	12/1/21 1:00 PM	12/4/21 1:00 PM	Yigit

cmpe352-2021group4 - page5



		Name	Duration	Start	Finish	Resource Names
103		Search Engine	10 days	10/21/21 1:00 PM	10/31/21 1:00 PM	Yigit
104		Testing for Search Engine	5 days	10/31/21 1:00 PM	11/5/21 1:00 PM	Yigit
105		Recommendation System	10 days	11/5/21 1:00 PM	11/15/21 1:00 PM	Yigit
106		Test Cases and Testing for Recommendation System	2.5 days	11/15/21 1:00 PM	11/17/21 5:00 PM	Yagmur;Yigit
107		Event Pages	10 days	11/18/21 8:00 AM	11/27/21 5:00 PM	Yagmur
108		Test Cases and Testing for Events	5 days	11/28/21 8:00 AM	12/2/21 5:00 PM	Yagmur
109		Equipment Pages	14 days	10/21/21 1:00 PM	11/4/21 1:00 PM	Yagmur
110		Test Cases and Testing for Equipment Pages	5 days	11/4/21 1:00 PM	11/9/21 1:00 PM	Yagmur
111		Badge Pages	10 days	11/9/21 1:00 PM	11/19/21 1:00 PM	Yagmur
112		Test Cases and Testing for Badges	5 days	11/19/21 1:00 PM	11/24/21 1:00 PM	Yagmur
113		Location Pages (map)	10 days	11/24/21 1:00 PM	12/4/21 1:00 PM	Yagmur
114		frontend meeting	1 day	12/4/21 1:00 PM	12/5/21 1:00 PM	Yagmur
115		testing and revising the frontend	7.5 days	12/5/21 1:00 PM	12/12/21 5:00 PM	Yagmur;Yigit
116		**⊟Android**	**70.5 days**	**10/21/21 8:00 AM**	**12/30/21 1:00 PM**	
117		Revising initial Android Design	1.5 days	10/21/21 8:00 AM	10/22/21 1:00 PM	Ece;Fatih
118		Home page	10 days	10/22/21 1:00 PM	11/1/21 1:00 PM	Ece
119		**⊟Registration Page**	**14 days**	**11/1/21 1:00 PM**	**11/15/21 1:00 PM**	
120		Login page	7 days	11/1/21 1:00 PM	11/8/21 1:00 PM	Ece
121		Reset/ Change Password Page	7 days	11/8/21 1:00 PM	11/15/21 1:00 PM	Ece
122		User profile Page	10 days	11/15/21 1:00 PM	11/25/21 1:00 PM	Ece
123		**⊟User Action Pages**	**10 days**	**11/25/21 1:00 PM**	**12/5/21 1:00 PM**	
124		- follow-unfollow-block page	7 days	11/25/21 1:00 PM	12/2/21 1:00 PM	Ece
125		- Follow Request Page	3 days	12/2/21 1:00 PM	12/5/21 1:00 PM	Ece
126		Search Engine Pages	10 days	10/22/21 1:00 PM	11/1/21 1:00 PM	Fatih
127		Testing for Search Engine	5 days	11/1/21 1:00 PM	11/6/21 1:00 PM	Fatih
128		Recommendation System Pages	10 days	11/6/21 1:00 PM	11/16/21 1:00 PM	Fatih
129		Test Cases and Testing for Recommendation System	5 days	11/16/21 1:00 PM	11/21/21 1:00 PM	Fatih
130		Event Pages	10 days	11/21/21 1:00 PM	12/1/21 1:00 PM	Fatih
131		Test Cases and Testing for Events	5 days	12/1/21 1:00 PM	12/6/21 1:00 PM	Fatih
132		Equipment Pages	14 days	10/22/21 1:00 PM	11/5/21 1:00 PM	Ece
133		Test Cases and Testing for Equipment Pages	5 days	11/5/21 1:00 PM	11/10/21 1:00 PM	Ece
134		Badge Pages	10 days	11/10/21 1:00 PM	11/20/21 1:00 PM	Ece
135		Test Cases and Testing for Badges	5 days	11/20/21 1:00 PM	11/25/21 1:00 PM	Ece
136		Location Pages (map)	10 days	11/25/21 1:00 PM	12/5/21 1:00 PM	Fatih



		Name	Duration	Start	Finish	Resource Names
137		android meeting	1.5 days	12/6/21 1:00 PM	12/7/21 5:00 PM	Ece;Fatih
138		testing and revising the android	22.5 days	12/8/21 8:00 AM	12/30/21 1:00 PM	Ece;Fatih
139		General Group Meeting for evaluation	1.125 days	12/30/21 1:00 PM	12/31/21 1:59 PM	Alperen;Ece;Fatih;Hilmi;Irfan;Tolga;Yagmur;.
140		testing and revising all system	16.875 d...	12/31/21 1:59 PM	1/17/22 11:59 AM	Alperen;Ece;Fatih;Hilmi;Irfan;Tolga;Yagmur;.

RAM

Approval(A)
Lead(L)
Secondary(S)
Contributor(C)
Reviewer(R)
None(N)
	Ece Dilara Aslan	Fatih Akgöz	Mehmet Hilmi Dündar	Muhammed İrfan Bozkurt	Tolga Kerimoğlu	Ali Alperen Durak	Yağmur Selek	Yiğit Sarıoğlu	Berkay Gümüş	İhsan Mert Atalay	Salih Furkan Akkurt

ORIENTATION											
searching about git control system	C		C	C	C		C	C	C	C	C
searching about github repositories	C		C	C	C		C	C	C	C	C
creating a slack server and channels	R		R	C	C		R	R	C	R	C
first group meeting	C		C	C	C		C	C	C	C	C
creating a welcome page	N		N	R	R		C	R	N	N	N
DOCUMENTATION											
creating personal wiki pages	C		C	C	C		C	C	C	C	C
creating personal effort pages	C		C	C	C		C	C	C	C	C
creating wikipage for favoruite github repositories	C		C	C	C		C	C	C	C	C
creating readme page	N		N	N	C		N	R	N	N	N
creating the communicaton plan	C		N	N	N		S	S	N	N	N
REQUIREMENTS											
performing a domain analysis	R		N	C	R		N	C	N	R	N
searching the requirements in real life project	C		C	C	C		C	C	C	C	C
identifying and documenting the user requirements	R		C	R	N		R	R	C	R	C
identifying and documenting the system requirements	C		R	R	C		R	R	R	C	R
identifying and documenting the non-functional requirements	R		R	C	R		C	C	N	R	R
preparing and documenting customer questions	R		R	R	R		R	C	C	R	C
customer meeting	C		C	C	L		C	C	C	C	C
reviewing and changing requirements	C		C	C	C		S	C	C	C	C
SCENARIO											
creating scenario 1	R		R	C	R		C	R	C	R	R
creating scenario 2	C		R	R	C		R	R	N	C	R
creating scenario 3	R		C	R	R		R	C	N	R	C
MOCKUP											
creating mockup 1	R		R	C	R		C	R	C	R	R
creating mockup 2	C		R	R	C		R	R	N	C	R
creating mockup 3	R		C	R	R		R	C	N	R	C
DIAGRAMS											
creating class diagram	R		R	C	R		C	R	C	R	R
creating use diagram	R		C	R	R		R	C	N	R	C
creating sequence diagrams	C		R	R	C		R	R	N	C	R
REVISIONS											
modifying mockup 1	R		R	L	R		S	R	C	R	R

Approval(A)
Lead(L)
Secondary(S)
Contributor(C)
Reviewer(R)
None(N)
	Ece Dilara Aslan	Fatih Akgöz	Mehmet Hilmi Dündar	Muhammed İrfan Bozkurt	Tolga Kerimoğlu	Ali Alperen Durak	Yağmur Selek	Yiğit Sarıoğlu	Berkay Gümüş	İhsan Mert Atalay	Salih Furkan Akkurt

modifying mockup 2	C		R	R	C		N	R	N	C	R
modifying mockup 3	R		C	R	R		N	C	N	R	C
modifying class diagram	R		R	L	R		C	R	C	R	R
modifying use case diagram	R		C	N	R		N	C	N	R	C
modifying sequence diagram	C		R	N	C		N	R	N	C	R
modifying SRS in parallel with the changes in mockups	C		N	C	C, L		C	C	C	C	C
PLANNING											
preparing project plan	R		R	S	A		C	R	L	R	R
preparing RAM	R		R	S	A, N		C	R	C	R	R
MILESTONE 1 REPORT											
setting up layout and design	C		R	R	L		R	C	R	R	C
creating the evaluation of deliverables	L		N	N	R		N	C	R	N	N
creating the evaluation tools and processes	R		N	N	R		N	N	R	N	N
creating the list and status of deliverables	R		N	N	R		N	C	R	N	N
creating the executive summary	R		N	N	L		R	N	R	N	N
adding project plan	R		N	S	A		S	N	C	N	N
adding RAM	R		N	S	A		S	N	C	N	N
meeting to evaluate Milestone 1 report	C		C	C	L		C	N	C	C	C
RESEARCH BEFORE PRE IMPLEMENTATION											
research the APIs	C		C	C	C		N	C	N	C	C
designing new API	R		C	C	C, L		N	C	N	C	C
research about the databases and server	C		R	R	R		C	C	C	C	C
deciding the subgroups for implementation part	C		C	C	A		A	C	C	C	C
preimplementation meeting	C		C	C	L		C	C	C	C	C
PRE IMPLEMENTATION											
creating initial backend design	R		R	C	C		R	C	N	R	C
creating test server	R		R	C	C		R	N	N	C	C
creating initial frontend design	R		C	R	R		R	N	N	C	R
creating initial android design	C		R	R	R		C	C	C	R	R
prototype	C		C	C	C		C	C	C	C	C
discussion about prototype functionalities	C		C	C	C		C	L	C	C	C
customer meeting about prototype	C		C	C	L		S	R	C	C	C
documentation of final decision of prototype	C		C	C, A	C, A		C	R	C	C	C
IMPLEMENTATION											
Backend											

Approval(A)
Lead(L)
Secondary(S)
Contributor(C)
Reviewer(R)
None(N)
	Ece Dilara Aslan	Fatih Akgöz	Mehmet Hilmi Dündar	Muhammed İrfan Bozkurt	Tolga Kerimoğlu	Ali Alperen Durak	Yağmur Selek	Yiğit Sarıoğlu	Berkay Gümüş	İhsan Mert Atalay	Salih Furkan Akkurt

revising initial backend design	R	R	C	C	C	C	R	R			
revising test server	R	R	C	C	C	C	R	R			
registration system	R	R	C	C	L	C	R	R			
login system	R	R	C	C	L	C	R	R			
reset/change password	R	R	C	C	L	C	R	R			
linking google accounts	R	R	C	C	L	C	R	R			
profile privacy	R	R	C	C	L	L	R	R			
user action systems	R	R	C	L	C	C	R	R			
user follow/unfollow/block system	R	R	C	L	C	C	R	R			
search engine	R	R	C	L	C	C	R	R			
event controller / event	R	R	C	L	C	C	R	R			
test cases and testing for events	R	R	C	L	C	C	R	R			
recommendation engine	R	R	L	C	S	C	R	R			
equipment controller / equipment	R	R	L	R	A	C	R	R			
test cases and testing for equipments	R	R	L	R	R	C	R	R			
badge system	R	R	R	R	R	L	R	R			
location system	R	R	R	R	R	L	R	R			
backend meeting	N	N	C	C	L	C	N	N			
testing and revising the backend	N	N	C	C	C	C	N	N			
Frontend											
revising initial frontend design	R	N	R	R	R	R	L	L			
Home page	R	N	R	R	R	R	S	L			
Registration Page	R	N	R	R	R	R	S	L			
Login page	R	N	R	R	R	R	S	L			
Reset/ Change Password Page	R	N	N	N	N	N	S	L			
User profile Page	R	N	N	N	N	N	S	L			
User Action Pages	R	N	N	N	N	N	S	L			
follow-unfollow-block page	R	N	N	N	N	N	S	L			
Follow Request Page	R	N	N	N	N	N	S	L			
Search Engine	R	N	N	N	N	N	S	L			
Testing for Search Engine	R	N	N	N	N	N	S	L			
Recommendation System Page	R	N	R	R	R	R	S	L			
Test Cases and Testing for Recommendation System	R	N	R	N	R	R	C	L			
Event Pages	R	N	R	A	R	R	L	S			
Test Cases and Testing for Events	R	N	R	N	R	R	L	S			

Approval(A)
Lead(L)
Secondary(S)
Contributor(C)
Reviewer(R)
None(N)
	Ece Dilara Aslan	Fatih Akgöz	Mehmet Hilmi Dündar	Muhammed İrfan Bozkurt	Tolga Kerimoğlu	Ali Alperen Durak	Yağmur Selek	Yiğit Sarıoğlu	Berkay Gümüş	İhsan Mert Atalay	Salih Furkan Akkurt

Equipment Pages	R	N	A	N	A	A	L	S			
Test Cases and Testing for Equipment Pages	R	N	A	N	A	A	L	S			
Badge Pages	R	N	A, R	N	A, R	A, R	L	S			
Test Cases and Testing for Badges	R	N	R	N	R	R	L	S			
Location Pages (map)	R	N	R	R	R	R	L	S			
frontend meeting	R	N	R	R	R	R	C	L			
testing and revising the frontend	N	N	N	N	N	N	C	C			
Android											
Revising initial Android Design	C	C	R	N	R	R	R	R			
Home page	L	S	R	N	R	R	R	R			
Registration Page	L	S	R	N	R	R	R	R			
Login page	L	S	R	N	R	R	R	R			
Reset/ Change Password Page	L	S	N	N	N	N	N	N			
User profile Page	L	S	N	N	N	N	N	N			
User Action Pages	L	S	N	N	N	N	N	N			
follow-unfollow-block page	L	S	N	N	N	N	N	N			
follow Request page	L	S	N	N	N	N	N	N			
Search Engine Pages	S	L	N	N	N	N	N	N			
Testing for Search Engine	S	L	N	N	N	N	N	N			
Recommendation System Pages	S	L	N	N	N	N	N	N			
Test Cases and Testing for Recommendation System	S	L	N	N	N	N	N	N			
Event Pages	S	L	N	N	N	N	N	N			
Test Cases and Testing for Events	S	L	N	N	N	N	N	N			
Equipment Pages	L	S	R	R	R	R	R	R			
Test Cases and Testing for Equipment Pages	L	S	R	R	R	R	R	R			
Badge Pages	L	S	N	N	N	N	N	N			
Test Cases and Testing for Badges	L	S	N	N	N	N	N	N			
Location Pages (map)	L	S	R	R	R	R	R	R			
android meeting	C	C	R	R	R	R	R	R			
testing and revising the android	C	C	N	N	N	N	N	N			
General Group Meeting for evaluation	C	C	C	C	C	C	C	C			
testing and revising all system	C	C	C	C	C	C	C	C			

Scenarios and Mockups

Scenario 1 & Mockup

User and Their Persona

- Ayşen Gruda
- 28 years old
- Lives in Etiler, Istanbul
- Currently an economist working for an investment company.
- Has been playing tennis for a while.

Story

Ayşen serves as an accountant for big firms to direct their investments. She's been playing tennis for some time, but all her friends are busy like her and they can share no common time to enjoy group activities. That's why she downloaded our application a while ago, and now she intends to find an opponent to play a 1v1 tennis match.

Preconditions

- Ayşen is a registered user in the system.
- Ayşen is already logged in.

Goals

- Ayşen generally wants a consistent supply of tennis opponents.
- Ayşen initially wants to join a person for a tennis match.

Actions

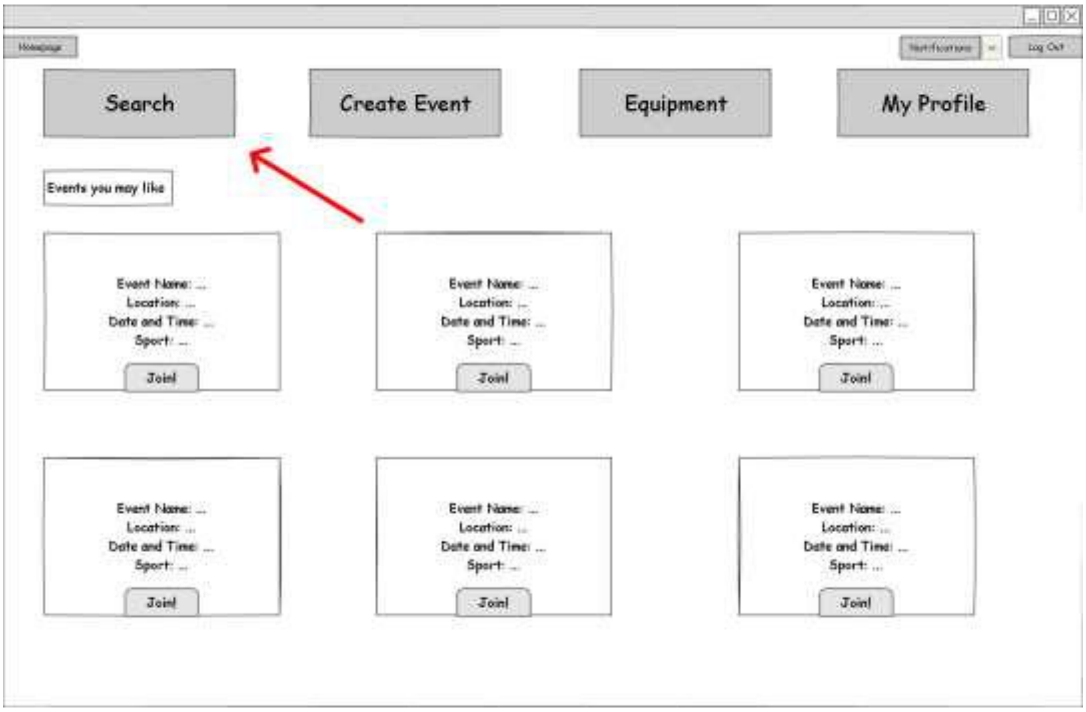
1. She opens the app and clicks 'Search' on the homepage.
2. She selects the sport type "tennis" from the options provided.
3. She enters the location, and specifies a range of date and time in the form displayed. Clicks "find".
4. She receives a list of tennis players organising a match, corresponding dates and locations.
5. She picks a tennis match she likes and visits the event page. Likes the event and clicks "apply".
6. After some time, she receives notifications informing their opponent accepted their application or the event time coming close.

Acceptance Criteria

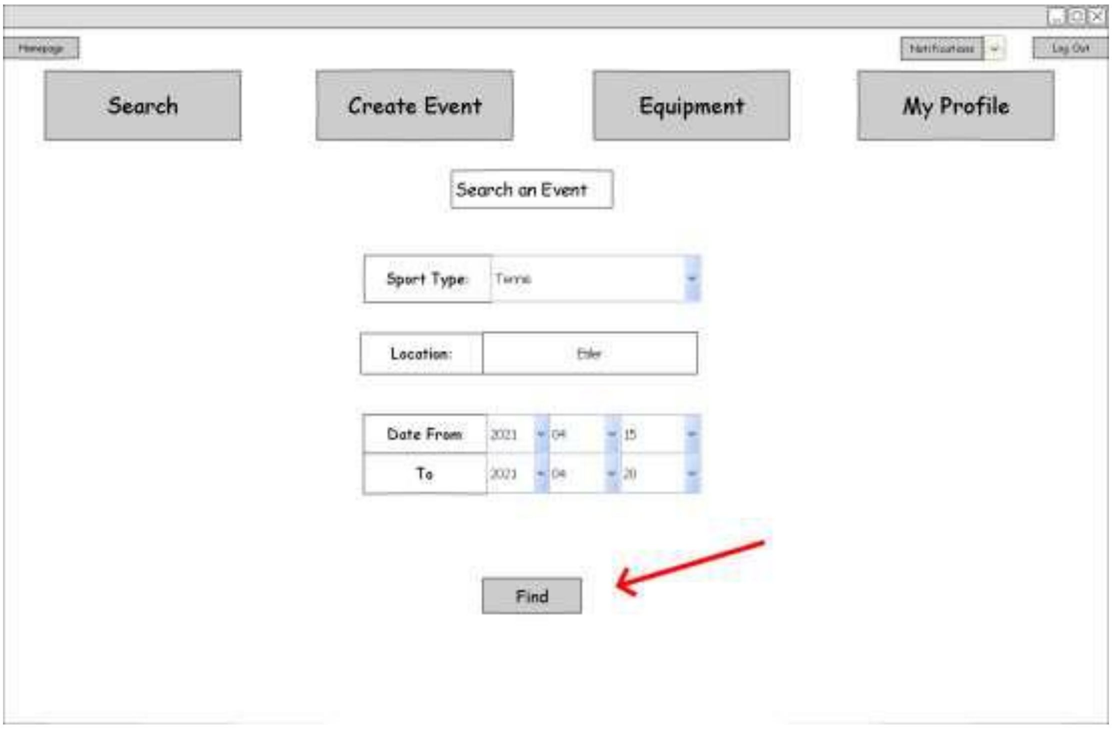
- 3.1.2.3.1 The system shall provide information about the events such as event name, location, date, sport and the number of players.
- 3.1.1.5.3 User shall be able to apply to participate in or spectate an event.
- 3.1.2.3.3 The system shall allow users to request joining an event.
- 3.1.1.5.4 User shall be informed on whether he/she was accepted to the event.
- 3.1.2.5.2 The system shall send notifications to users who are selected or not selected for a specific event.

Web Mockup

1. She opens the app and clicks 'Search' on the homepage.



2. She selects the sport type "tennis" from the options provided.
3. She enters the location, and specifies a range of date and time in the form displayed. Clicks "find".



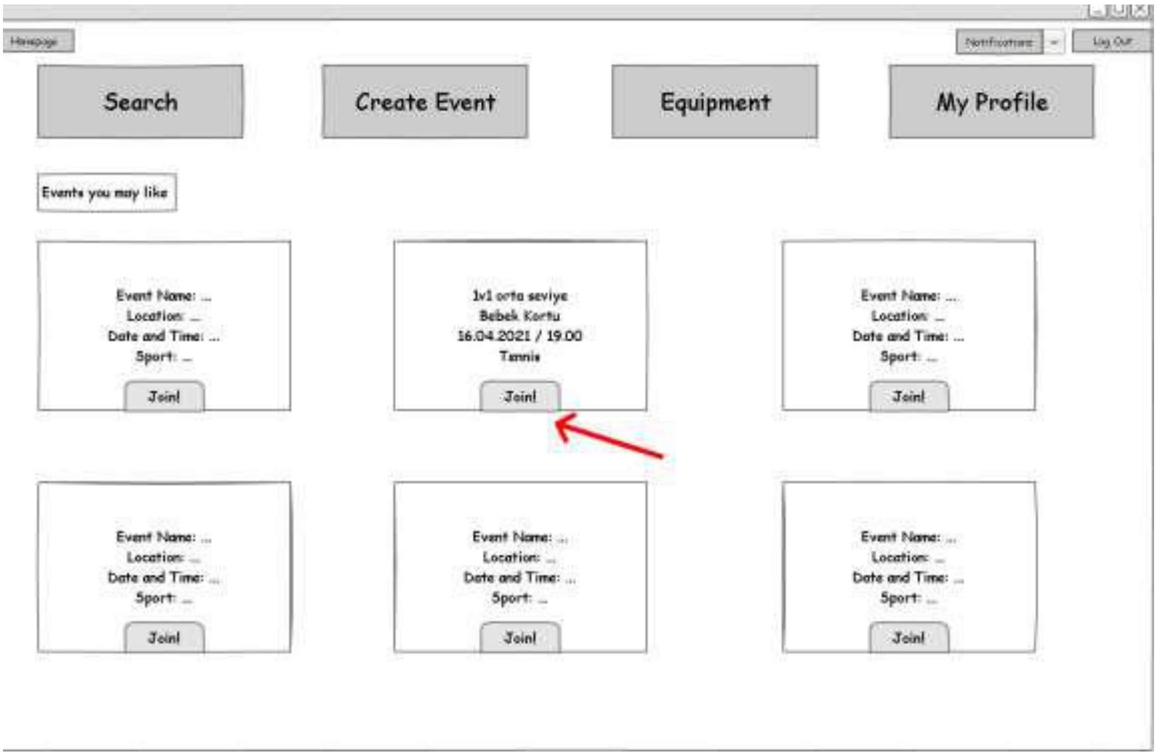
Search an Event

Sport Type: Tennis

Location: Exeter

Date From	2021	04	15
To	2021	04	20

4. She receives a list of tennis players organising a match, corresponding dates and locations.



Event Name: ...
Location: ...
Date and Time: ...
Sport: ...

1v1 orta seviye
Bebek Kortu
16.04.2021 / 19.00
Tennis

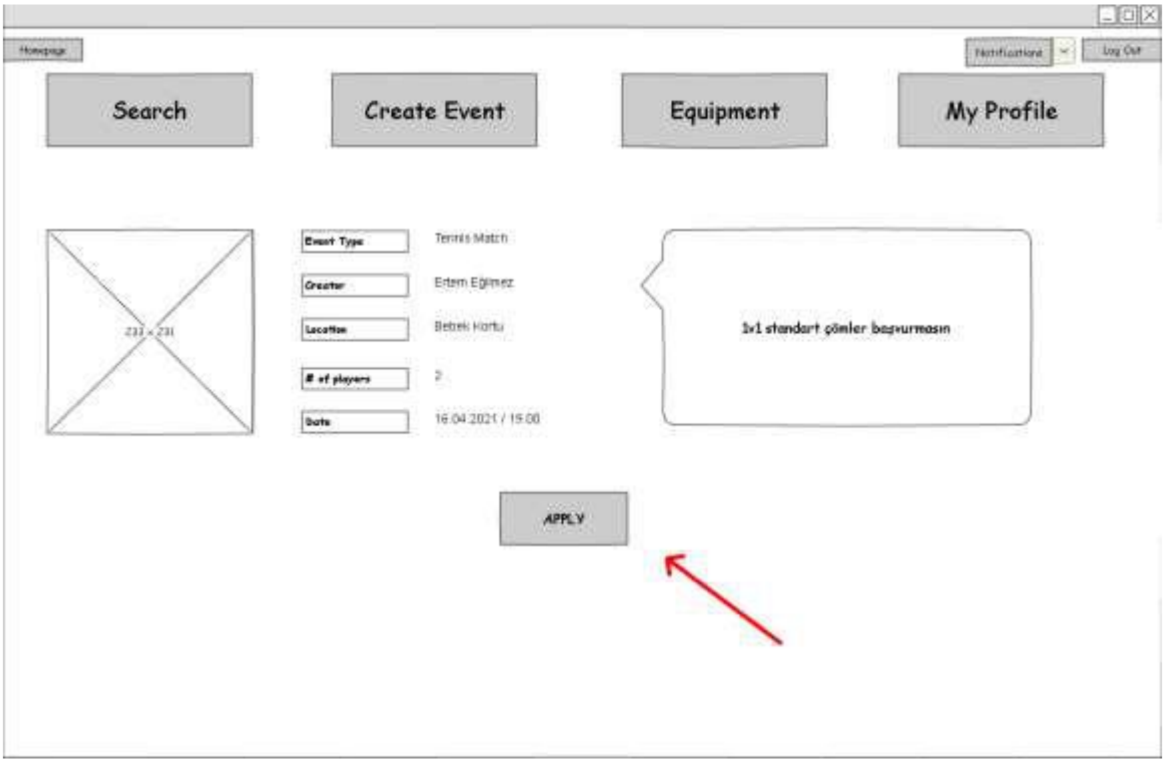
Event Name: ...
Location: ...
Date and Time: ...
Sport: ...

Event Name: ...
Location: ...
Date and Time: ...
Sport: ...

Event Name: ...
Location: ...
Date and Time: ...
Sport: ...

Event Name: ...
Location: ...
Date and Time: ...
Sport: ...

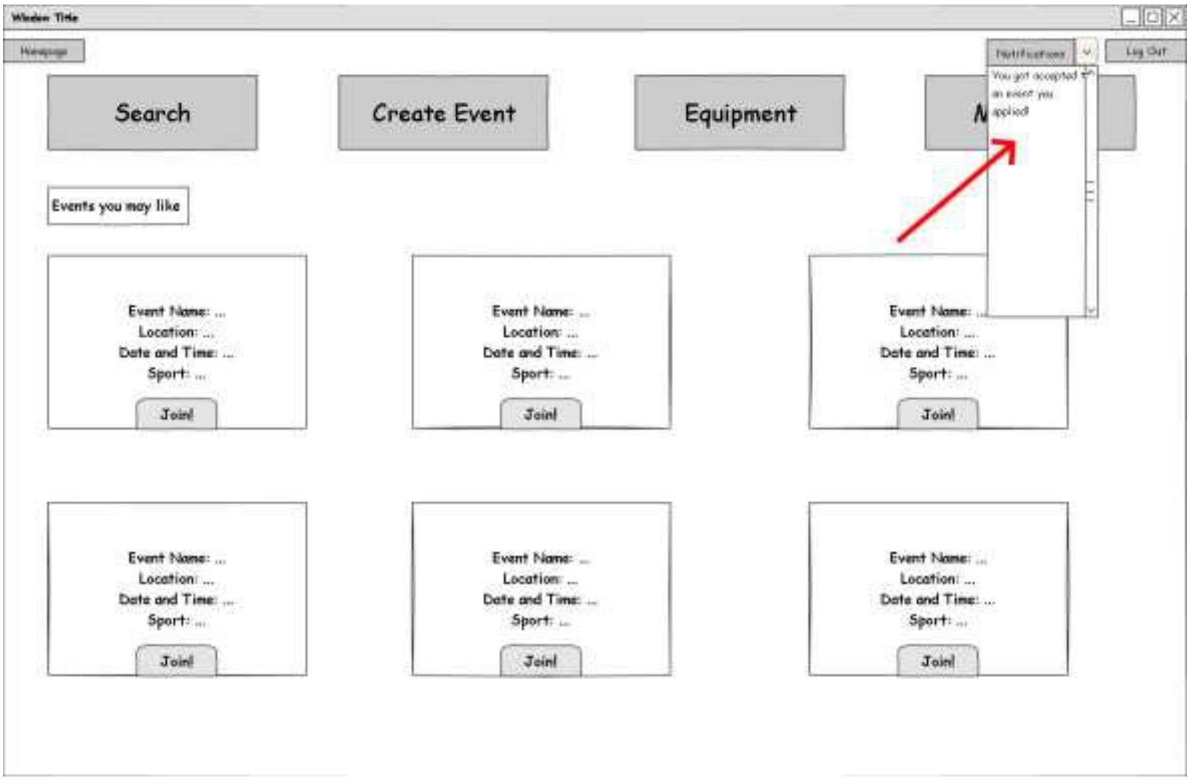
5. She picks a tennis match she likes and visits the event page. Likes the event and clicks "apply"



Event Type Tennis Match
Creator Erdem Eğilmez
Location Bebek Kortu
# of players 2
Date 16.04.2021 / 19.00

1v1 standart çömler başvurmasın

6. After some time, she receives notifications informing their opponent accepted their application or the event time coming close.



You got accepted t an event you applied!

Event Name: ...
Location: ...
Date and Time: ...
Sport: ...

Event Name: ...
Location: ...
Date and Time: ...
Sport: ...

Event Name: ...
Location: ...
Date and Time: ...
Sport: ...

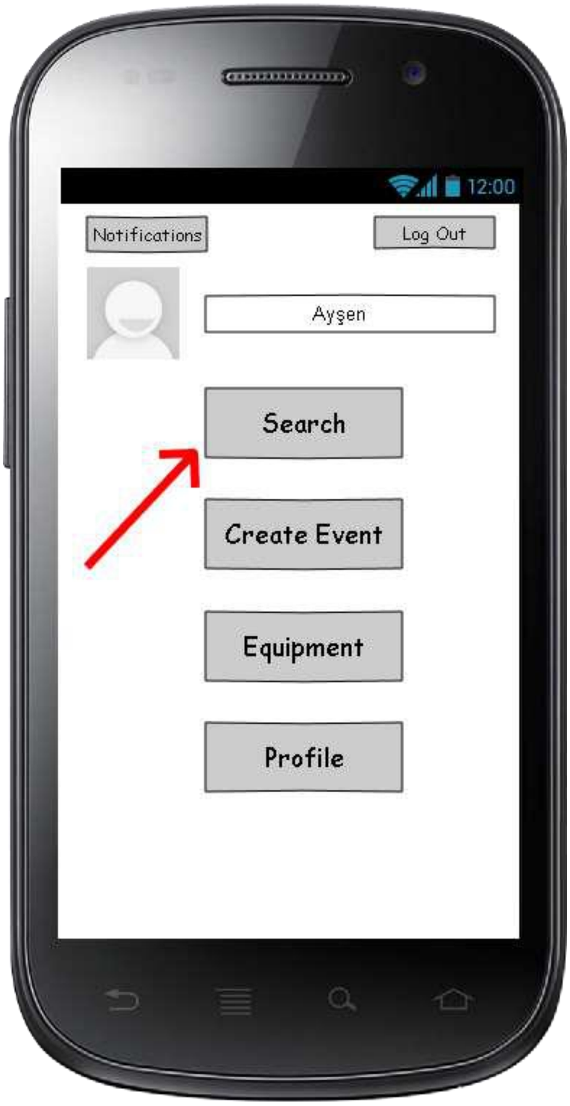
Event Name: ...
Location: ...
Date and Time: ...
Sport: ...

Event Name: ...
Location: ...
Date and Time: ...
Sport: ...

Event Name: ...
Location: ...
Date and Time: ...
Sport: ...

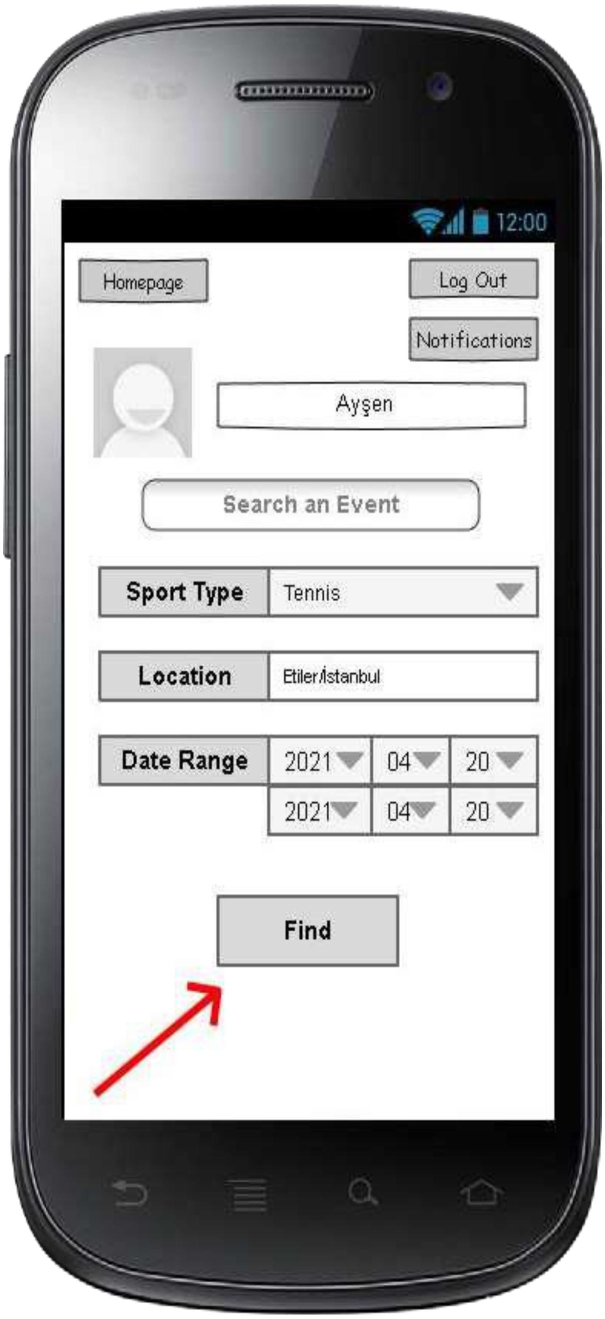
Android Mockup

1. She opens the app and clicks 'Search' on the homepage.



2. She selects the sport type "tennis" from the options provided.

3. She enters the location, and specifies a range of date and time in the form displayed. Clicks "find".



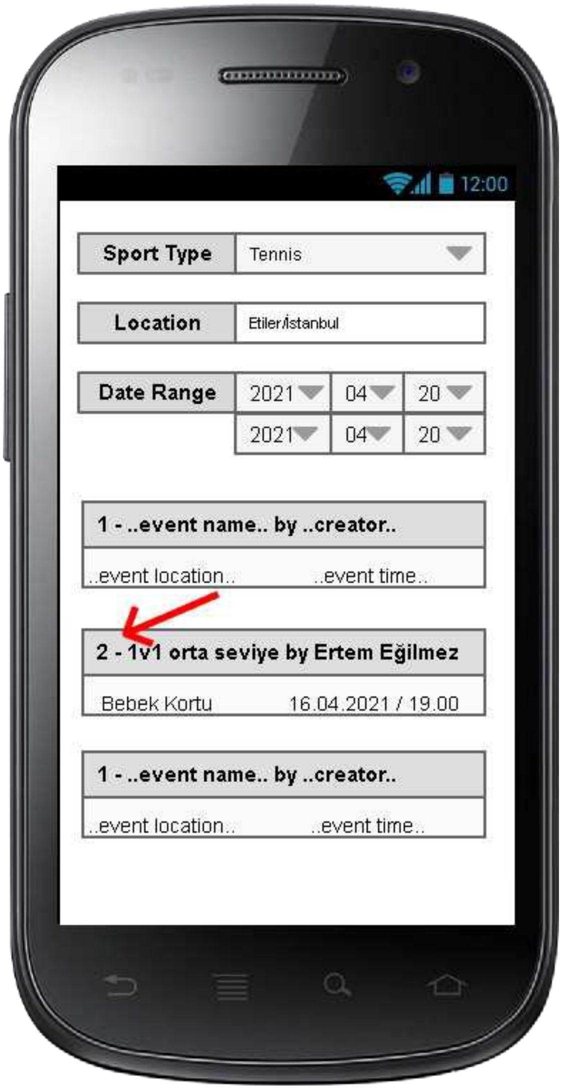
Ayşen

Sport Type: Tennis

Location: Etiler/İstanbul

Date Range: 2021 04 20
2021 04 20

4. She receives a list of tennis players organising a match, corresponding dates and locations.



Sport Type	Tennis		
Location	Etiler/İstanbul		
Date Range	2021	04	20
	2021	04	20

1 - ..event name.. by ..creator..

..event location.. ..event time..

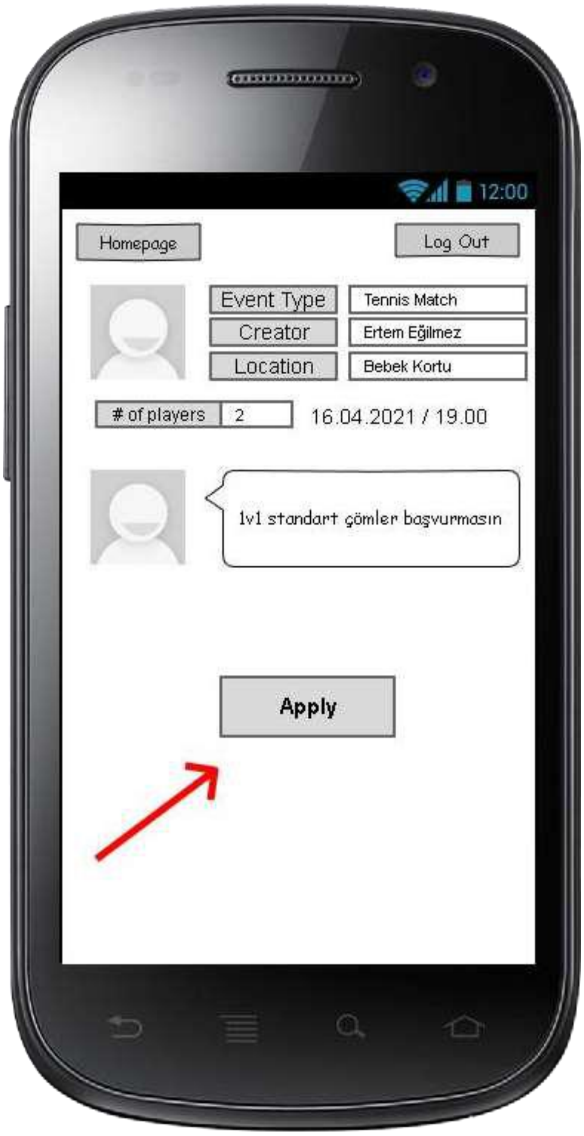
2 - 1v1 orta seviye by Ertem Eğilmez

Bebek Kortu 16.04.2021 / 19.00

1 - ..event name.. by ..creator..

..event location.. ..event time..

5. She picks a tennis match she likes and visits the event page. Likes the event and clicks "apply".

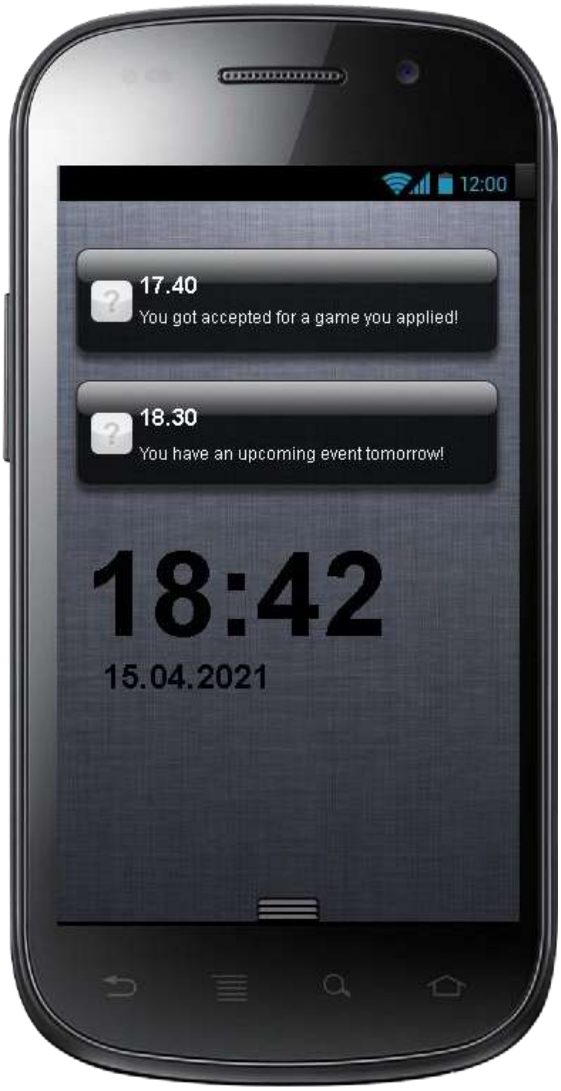


Event Type	Tennis Match
Creator	Ertem Eğilmez
Location	Bebek Kortu
# of players	2

16.04.2021 / 19.00

1v1 standart çömler başvurmasın

6. After some time, she receives notifications informing their opponent accepted their application or the event time coming close.



12:00

17.40

You got accepted for a game you applied!

18.30

You have an upcoming event tomorrow!

18:42

15.04.2021

Scenario 2 & Mockup

User and Their Persona

- Hasan
- 32 years old
- Lives in Etiler, Istanbul
- Currently a software engineer working at a big finance firm
- Likes to play soccer

Story

Hasan is a software engineer, waiting for his next promotion. He's a fairly social person, likes to have fun and exercise to get rid of the stress of his worklife. He has discovered our website/application a few months back when one of his friends recommended. He has joined a couple soccer games in his neighborhood, and now he wants to organize his own game.

Preconditions

- Hasan is a registered user in the system.
- Hasan is already logged in.

Goals

- He wants to organize a soccer game in the local soccer field near his house.
- He wants to have a 5 vs. 5 game, so he needs to find 9 other players.

Actions

1. Hasan is in the application homepage and clicks the button for event creation.
2. In the event creation page he types in the event name.
3. He selects the sport type as soccer from the default options provided.
4. He enters the location, number of players, date and time in the form displayed.
5. He clicks the "complete" button.
6. He receives a pop-up which states that event creation is successful, he leaves the website/application.
7. He checks his notifications from time to time to see who have applied to join the game.
8. He views the profile of the user and either accepts or rejects the user to his game by clicking on the corresponding buttons.
9. He does 7-8 until there are enough players for the game.

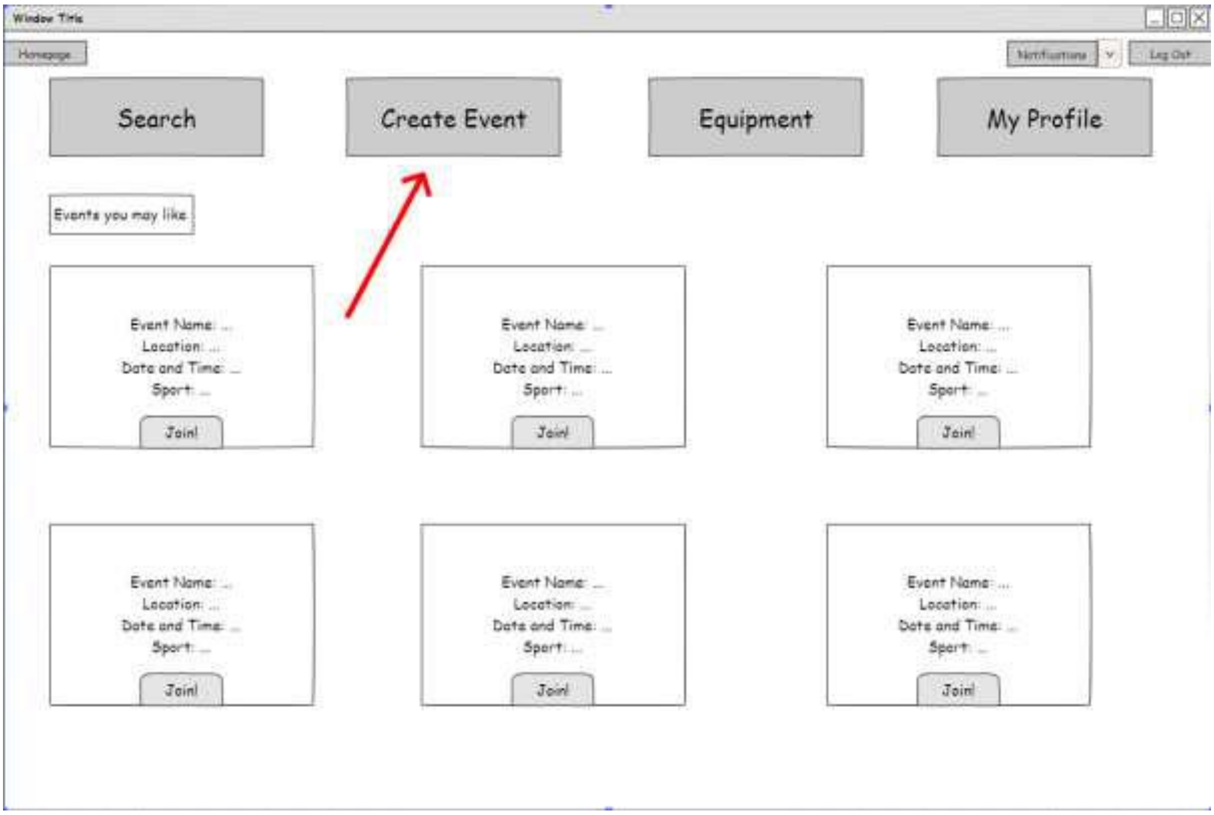
Acceptance Criteria

- 3.1.1.4.1 Any user shall be able to create events.
- 3.1.2.3.4 The system shall allow users to create events by specifying event name, location, date and sport and the number of players.
- 3.1.2.3.5 The system shall provide default sport types for the event creators to select from.
- 3.1.2.3.7 The system shall send a pop-up message after a user creates an event.

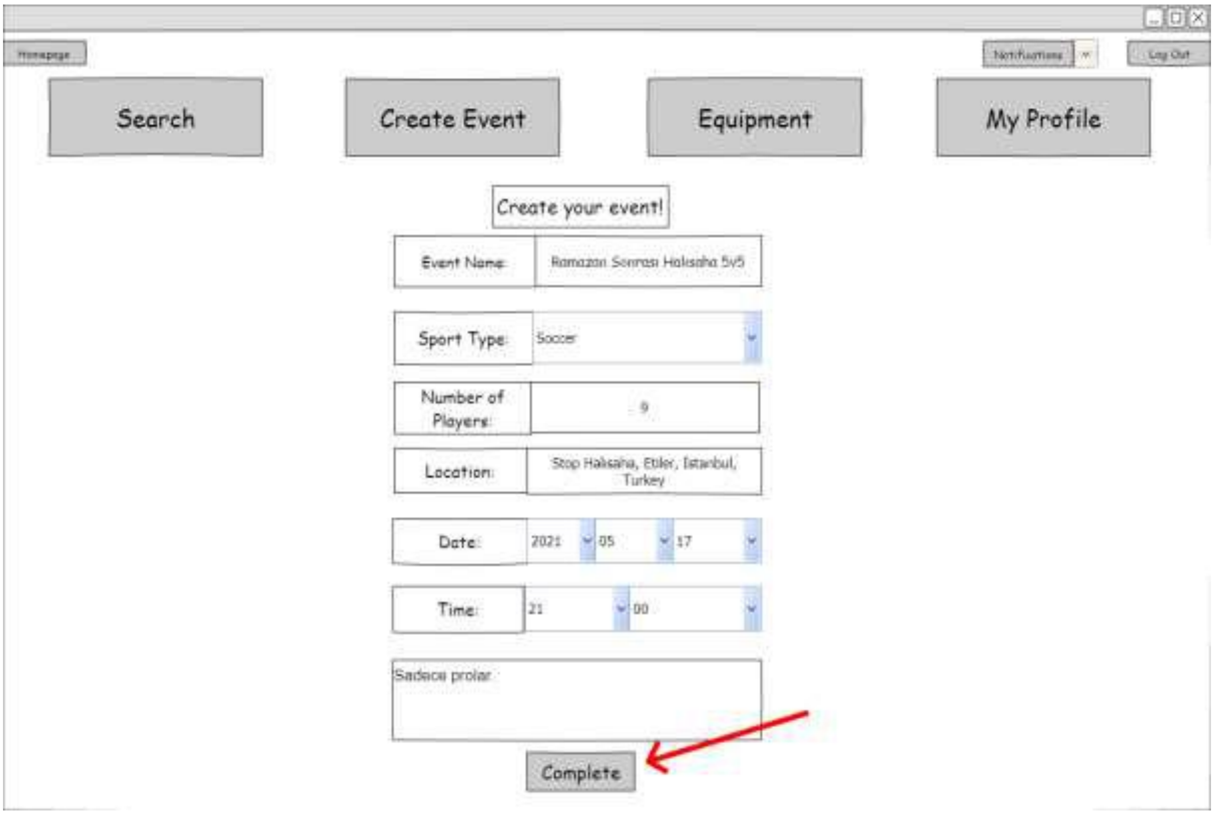
- 3.1.2.5.1 The system shall send notifications to event creators when a user sends a request to one of their events.
- 3.1.1.4.5 Event creator shall be able to check the appliers' badges or related information for whether to accept them or not.

Web Mockup

1. Hasan is in the application homepage and clicks the button for event creation.



2. In the event creation page he types in the event name.
3. He selects the sport type as soccer from the default options provided.
4. He enters the number of players, location, date and time in the form displayed.
5. He clicks the "complete" button.

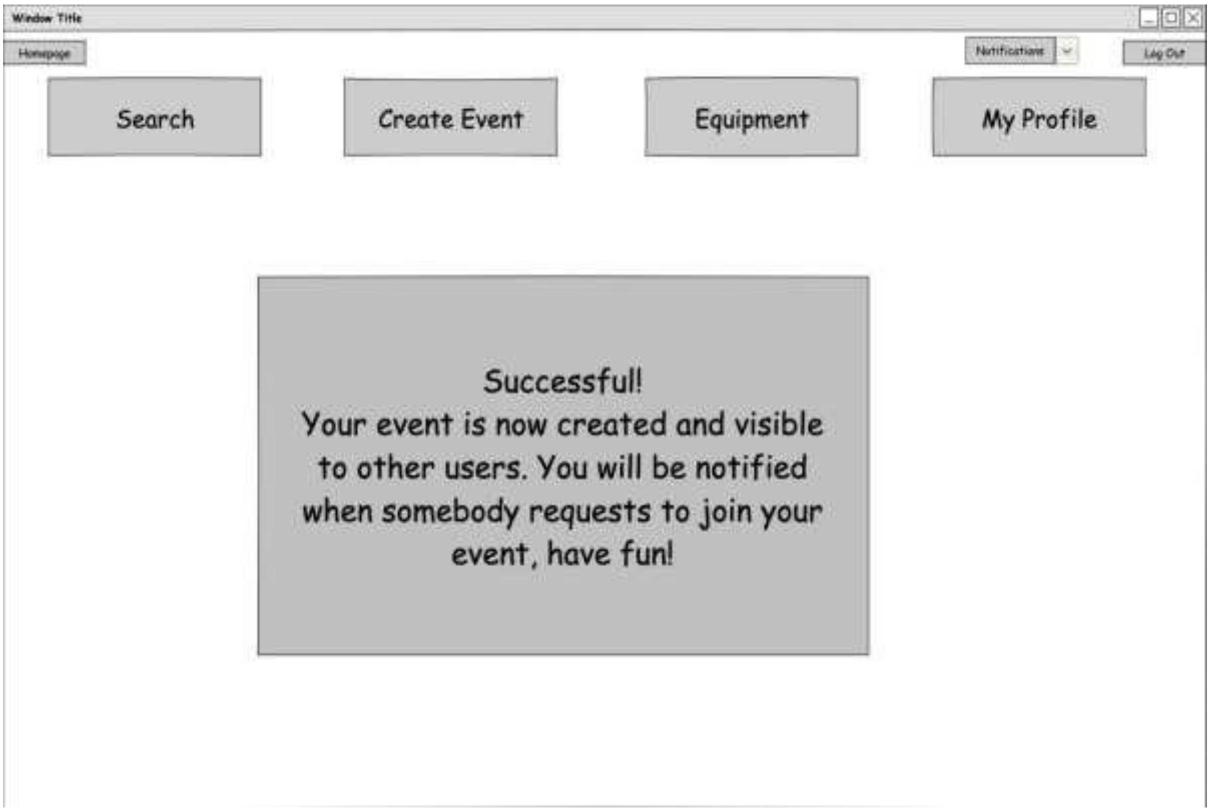


Create your event!

Event Name:	Ramazan Sonrası Halısaha 5v5
Sport Type:	Soccer
Number of Players:	9
Location:	Stop Halısaha, Etiler, İstanbul, Turkey
Date:	2021 05 17
Time:	21 00

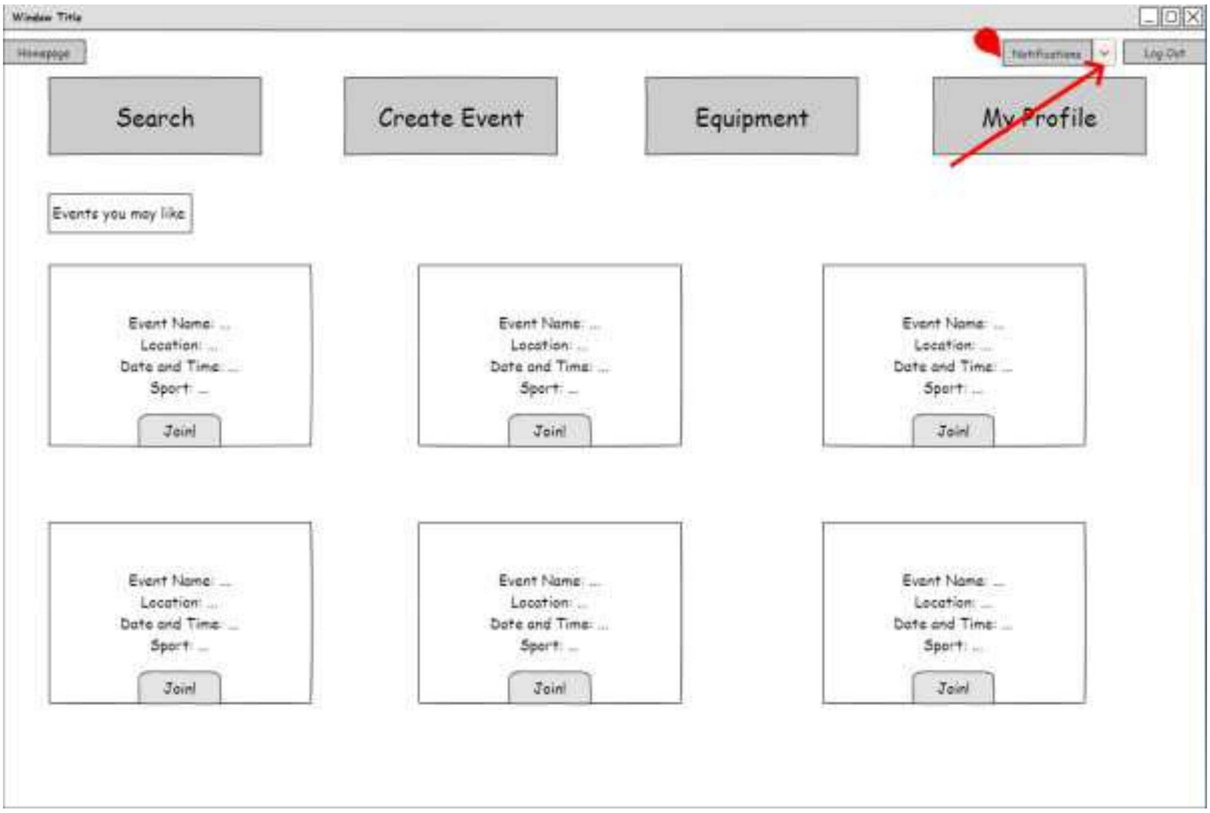
Sadece prolar

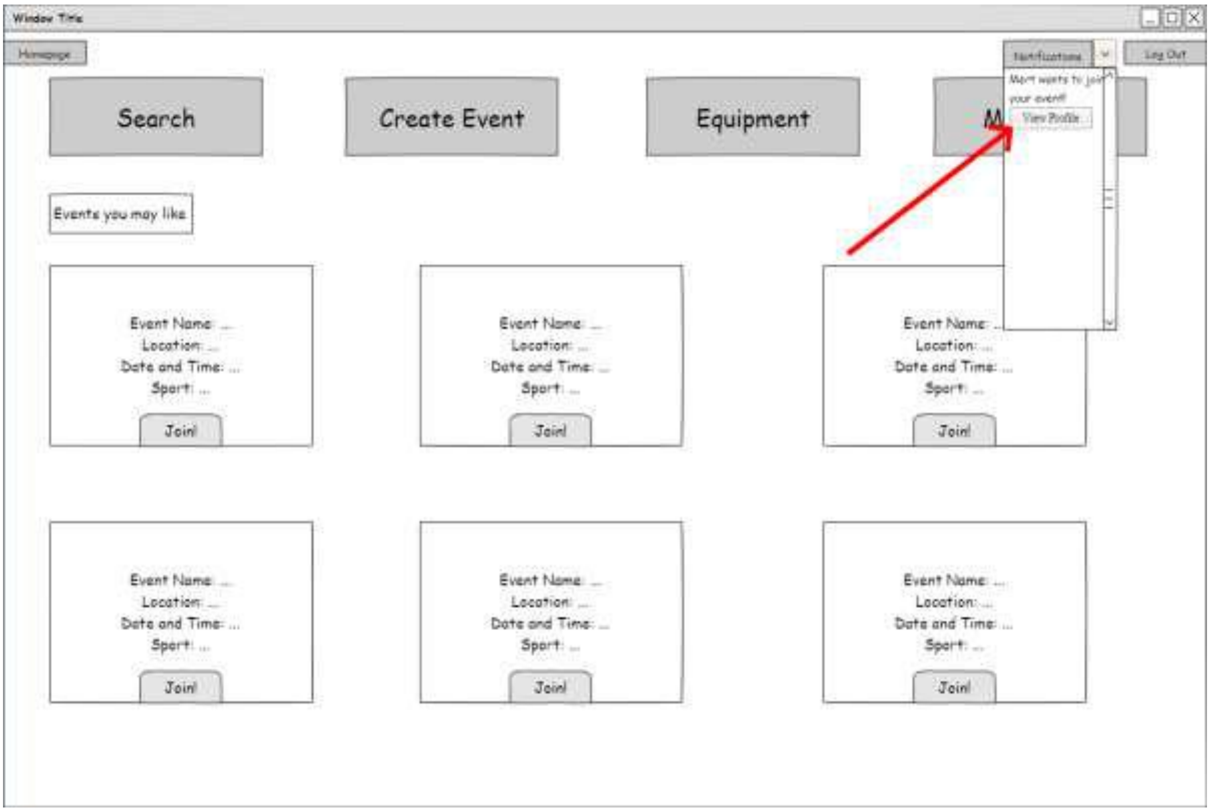
6. He receives a pop-up which states that event creation is successful, he leaves the website/application.



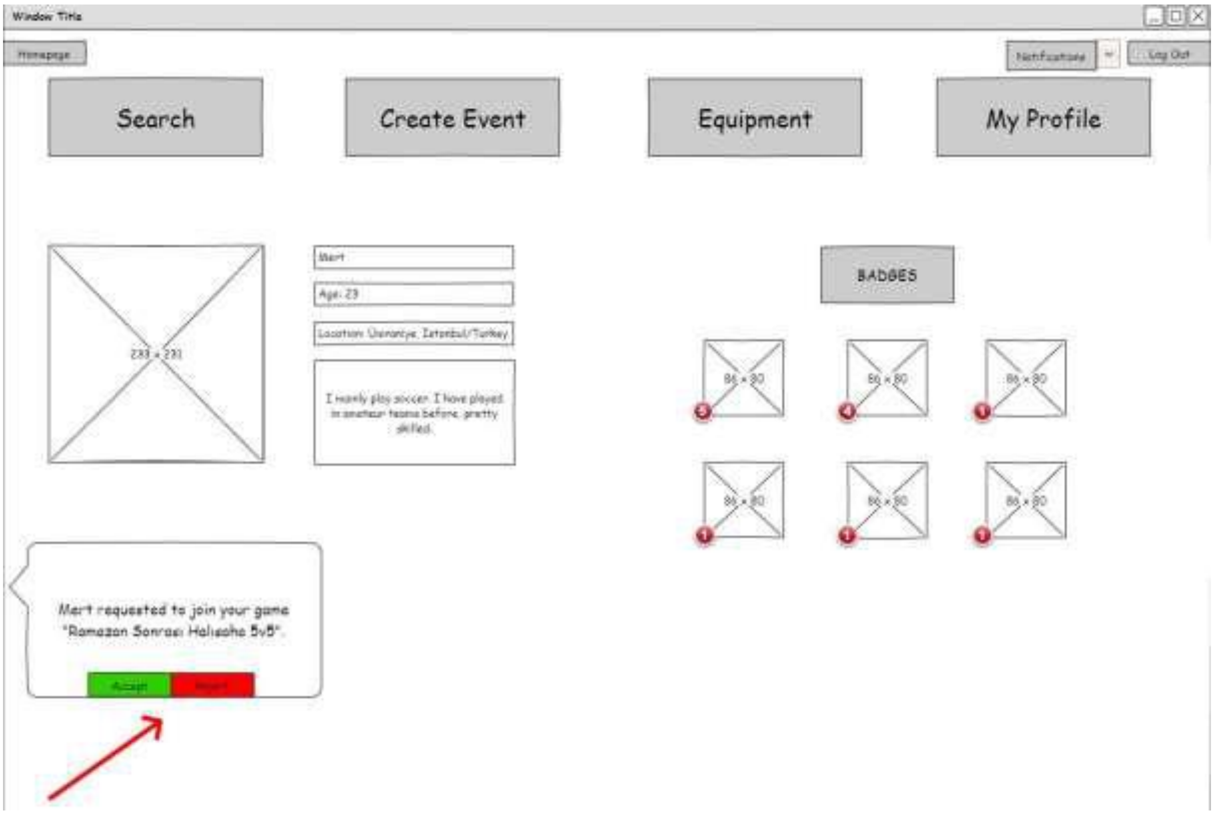
Successful!
Your event is now created and visible to other users. You will be notified when somebody requests to join your event, have fun!

7. He checks his notifications from time to time to see who have applied to join the game.





Mort wants to join your event!



Mert

Age: 23

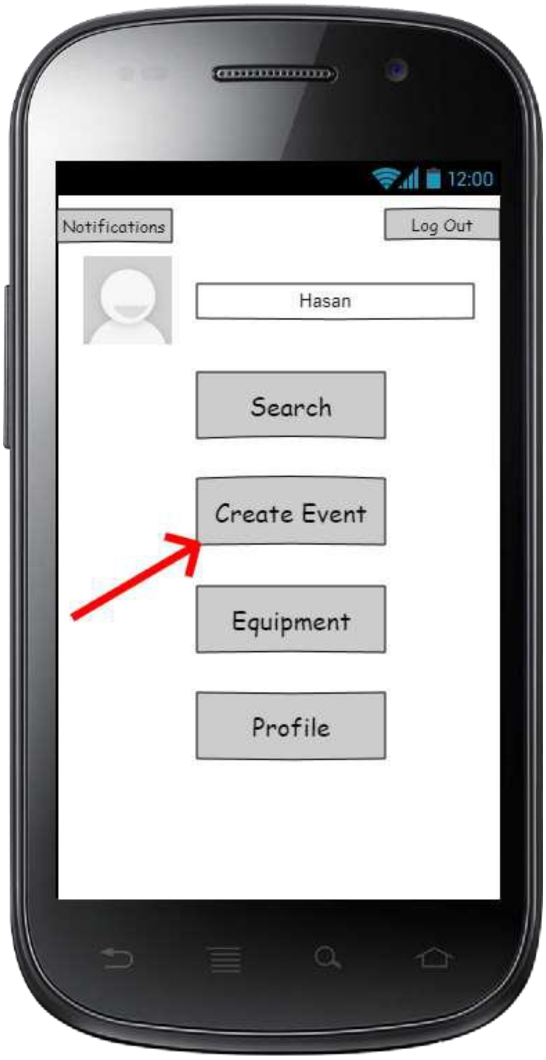
Location: Ümraniye, Istanbul/Turkey

I mainly play soccer. I have played in amateur teams before, pretty skilled.

Mert requested to join your game "Ramazan Sonrası Halısaha 5v5".

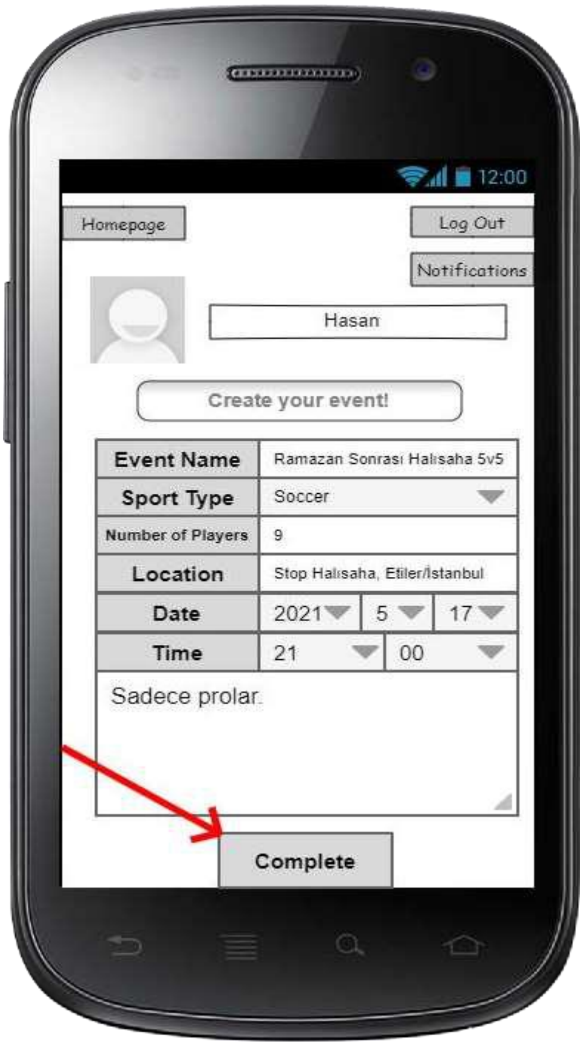
Android Mockup

1. Hasan is in the application homepage and clicks the button for event creation.



2. In the event creation page he types in the event name.
3. He selects the sport type as soccer from the default options provided.
4. He enters the number of players, location, date and time in the form displayed.

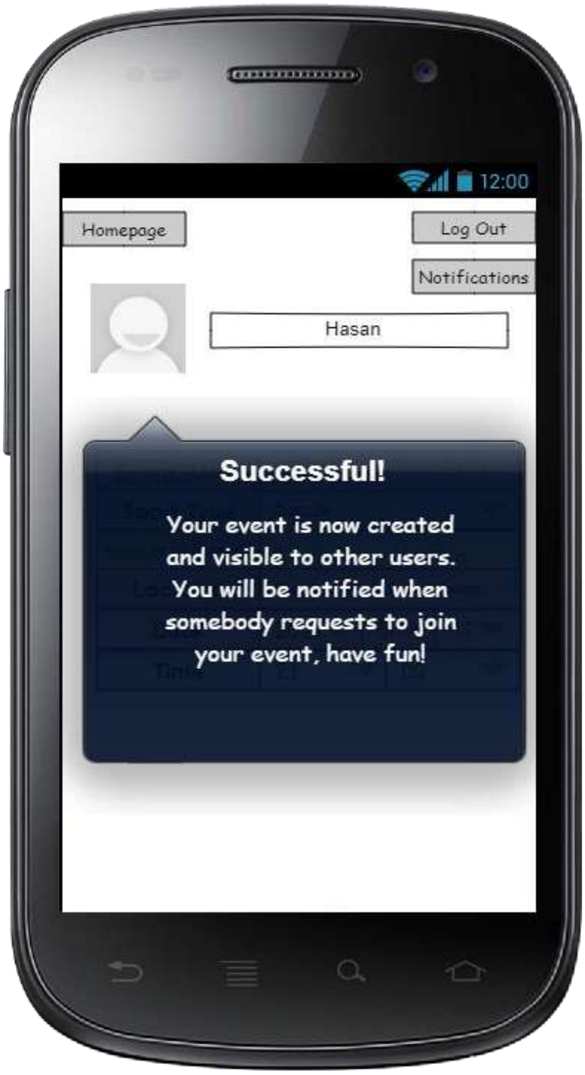
5. He clicks the "complete" button.



Event Name	Ramazan Sonrası Halısaha 5v5
Sport Type	Soccer
Number of Players	9
Location	Stop Halısaha, Etiler/İstanbul
Date	2021 5 17
Time	21 00

Sadece prolar.

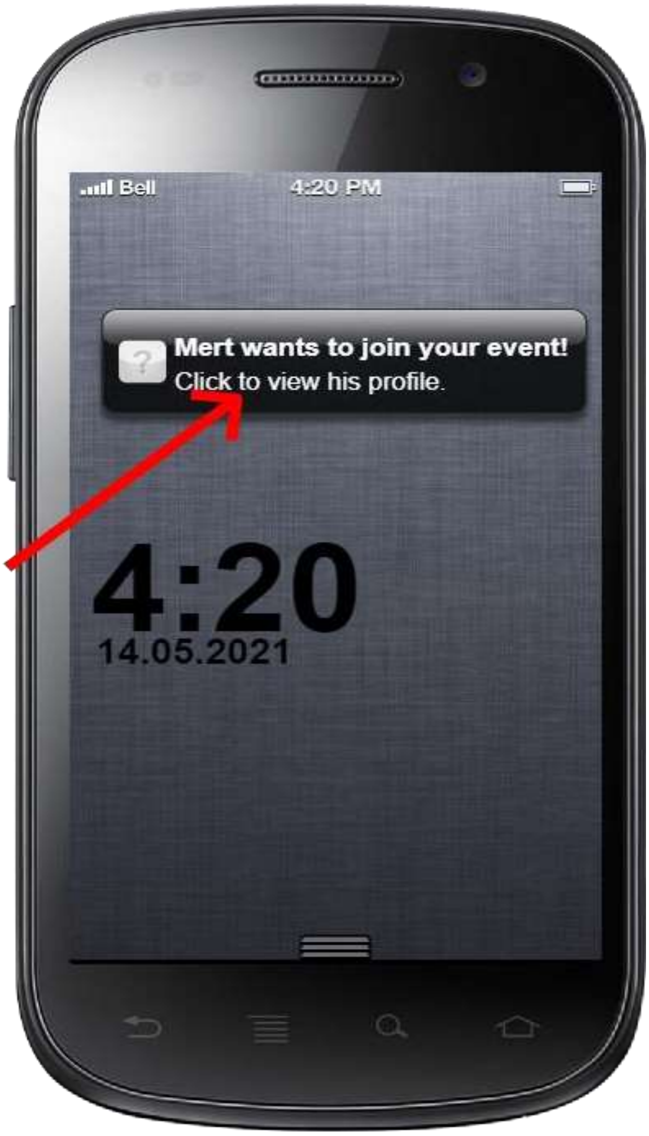
6. He receives a pop-up which states that event creation is successful, he leaves the website/application.



Successful!

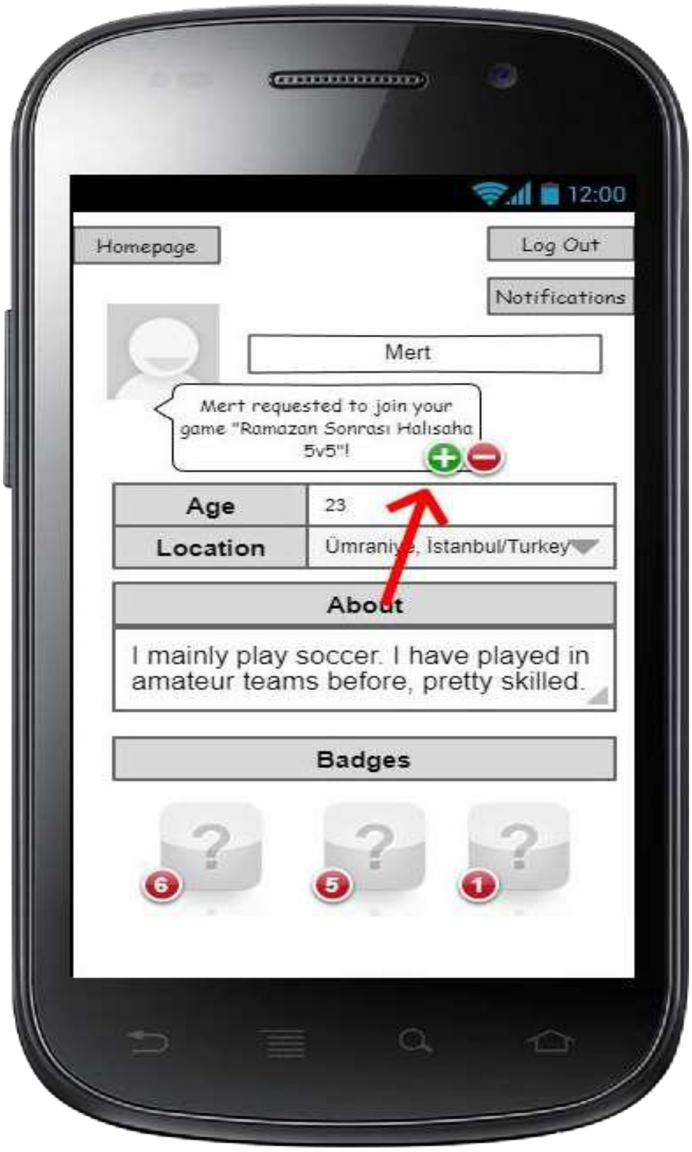
Your event is now created and visible to other users. You will be notified when somebody requests to join your event, have fun!

7. He checks his notifications from time to time to see who have applied to join the game.



Mert wants to join your event!
Click to view his profile.

8. He views the profile of the user and either accepts or rejects the user to his game by clicking on the corresponding buttons.



Mert

Mert requested to join your game "Ramazan Sonrası Halısaha 5v5"!

Age	23
Location	Ümraniye, İstanbul/Turkey

About

I mainly play soccer. I have played in amateur teams before, pretty skilled.

Scenario 3 & Mockup

User and Their Persona

- Ferit Şahin
- 22 years old
- Lives in Levent, Istanbul
- Currently a senior computer engineering student at Boğaziçi University
- Likes to ski and participate in community events

Story

Ferit enjoys skiing and he joined the university's ski club in the first year of college. He goes to ski resorts with his friends on holidays. He wants to ski in Uludağ with his friends in the next semester. But he needs to buy a new snowboard because his old one is broken. While wandering around campus, he saw our ads on the wall and it seems interesting to him. Then he visited our site by reading the QR code in the ad and registered our website.

Precondition

- Ferit is a registered user of our website and logged in.
- He has already given his location information.

Goals

- Ferit wants to go to the ski event he signed up to on our website.
- He wants to buy a snowboard for this event.

Actions

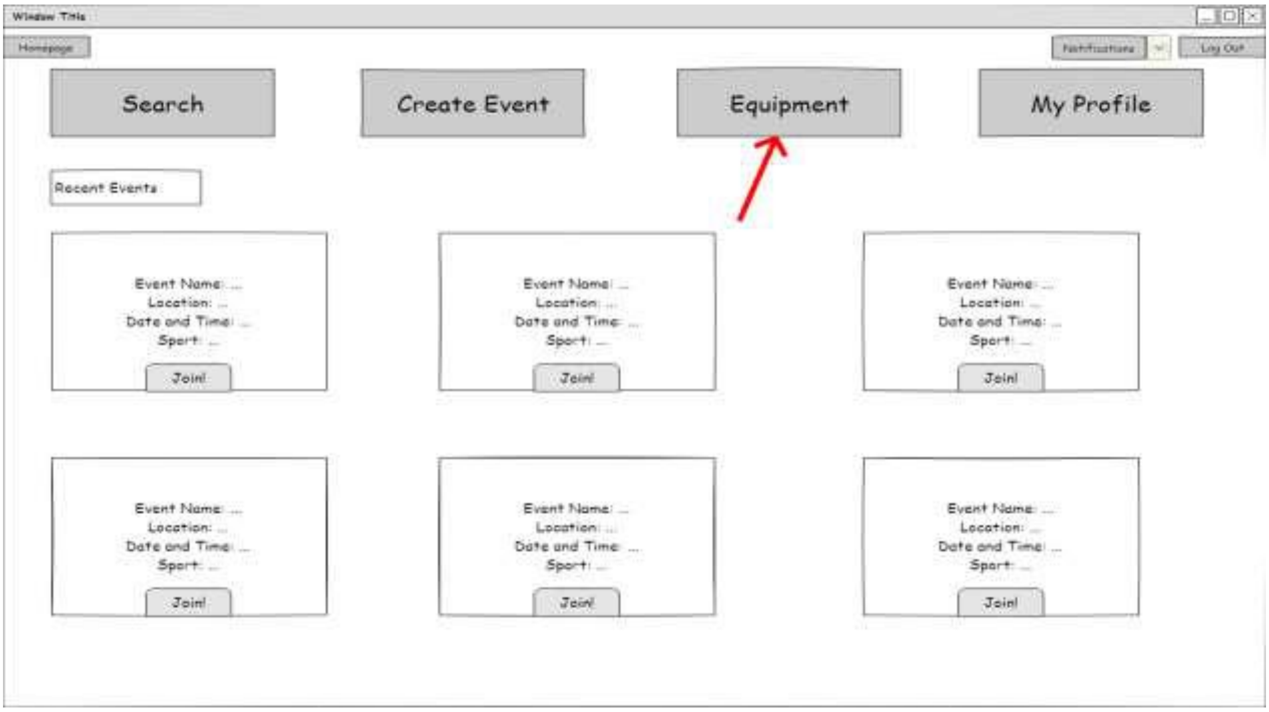
1. Ferit visits our application homepage 'bounswe_2021springgroup4.com'.
2. He clicks on 'Equipment' on the home page.
3. He clicks on 'Find Equipment' on 'Equipment' page.
4. He clicks on 'Skiing' on 'Sport Category' page
5. He sees several types of skiing equipment like snowboards, helmets, gloves, etc.
6. He clicks on 'Snowboard' from the types.
7. He sees various Snowboard equipment post pages (their photo, title, etc.) on this page. Then he chooses 'Google Map' option to filter the results.
8. Snowboard equipment posts that only selected area on the map are visible on this page. He chooses one of them and clicks on link.
9. He now sees the equipment page. In this page, he could see equipment photos, equipment description, comment and give a badge to the seller.

Acceptance Criteria

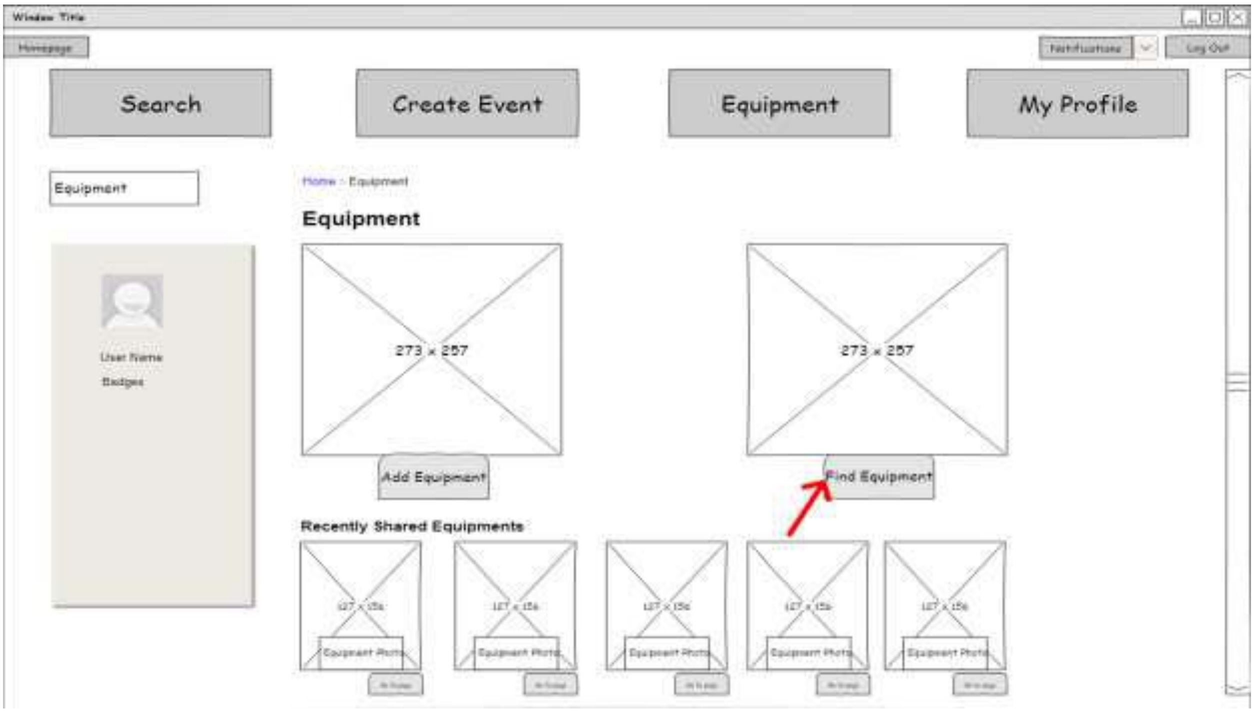
- 3.1.1.6.1 User shall be able to search for equipment sales by sport and location.
- 3.1.1.6.2 User shall be able to check information of equipment sellers and read descriptions written by them.
- 3.1.2.4.3 The system shall recommend sport-related content (gadgets, clothing, tools, merchandise) to users for the sports they are interested in.

Web Mockup

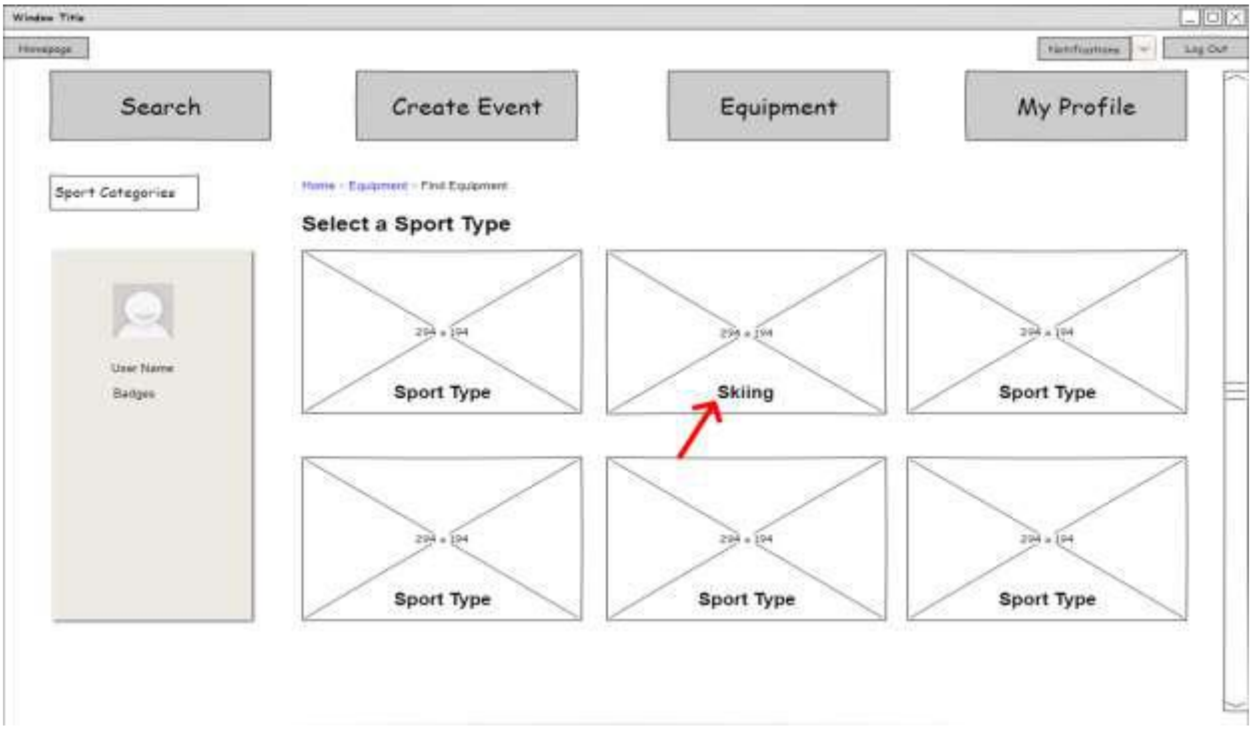
1. Ferit visits our application homepage 'bounswe_2021springgroup4.com'.
2. He clicks on 'Equipment' on the homepage.



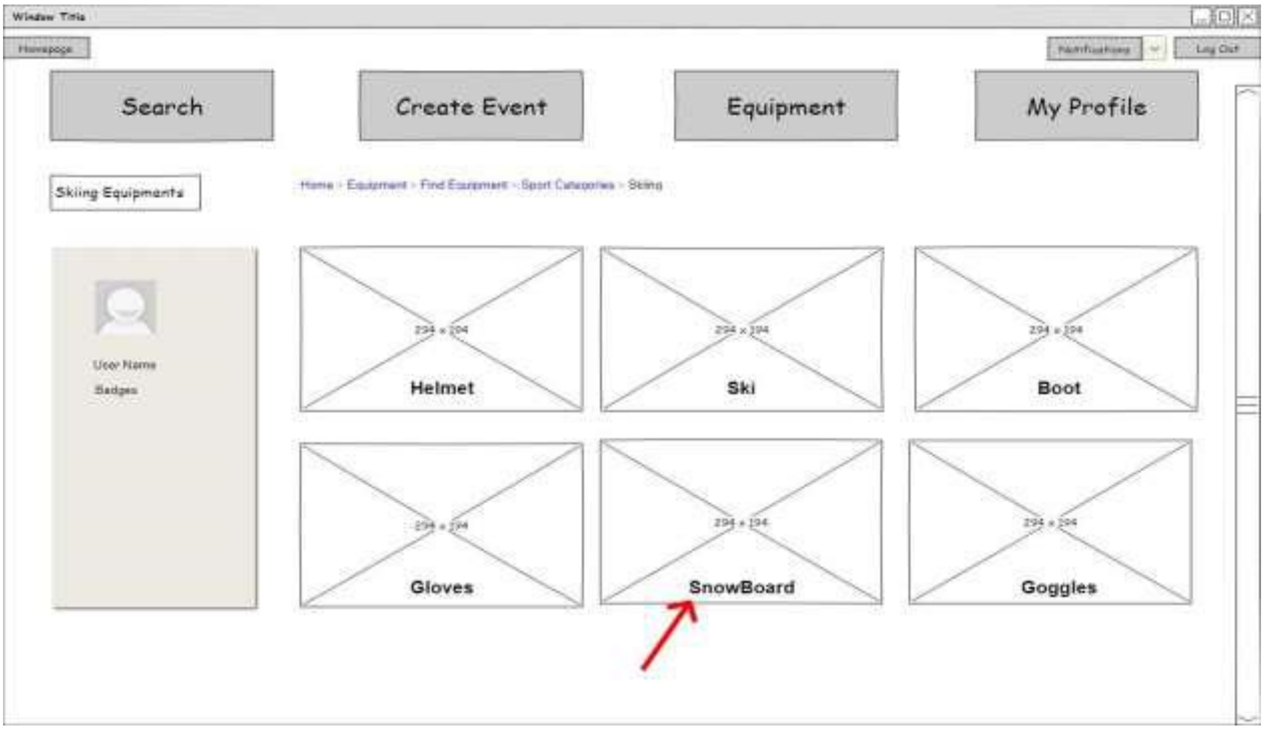
3. He clicks on 'Find Equipment' on the 'Equipment' page.



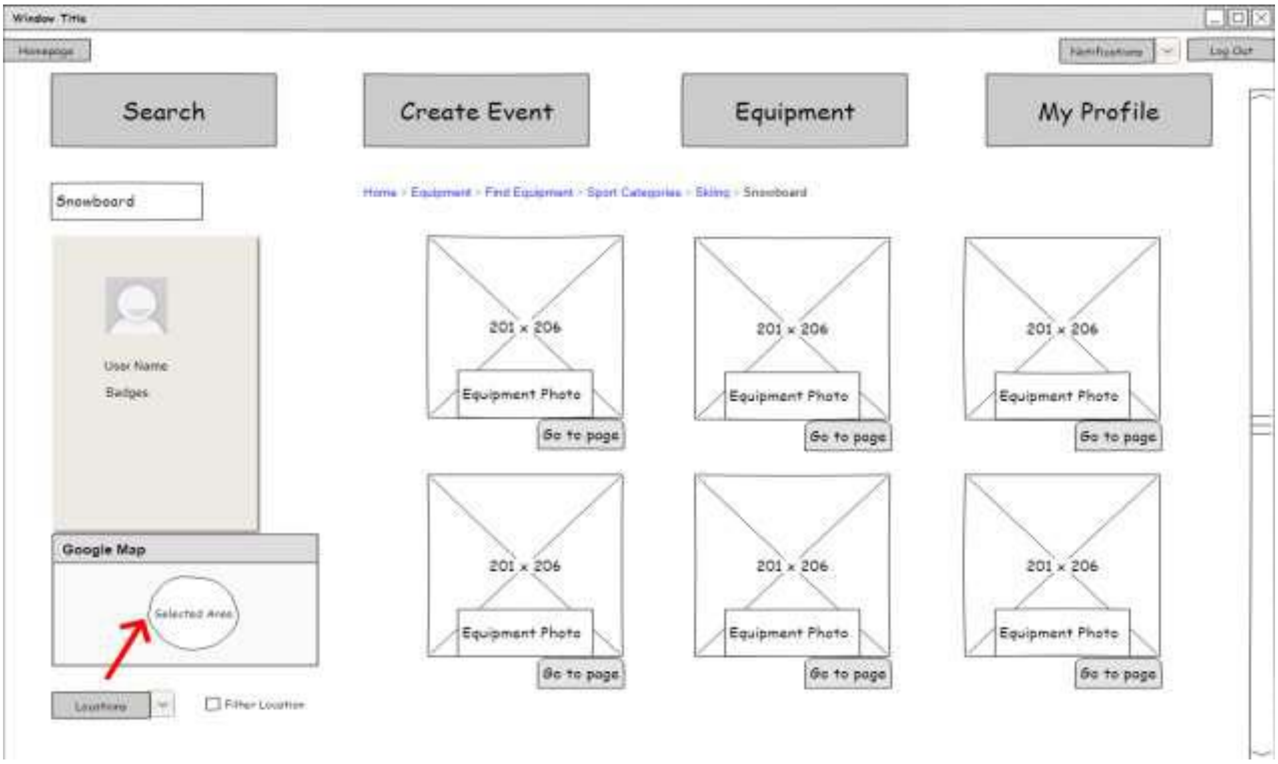
4. He clicks on 'Skiing' on the 'Sport Categories' page.



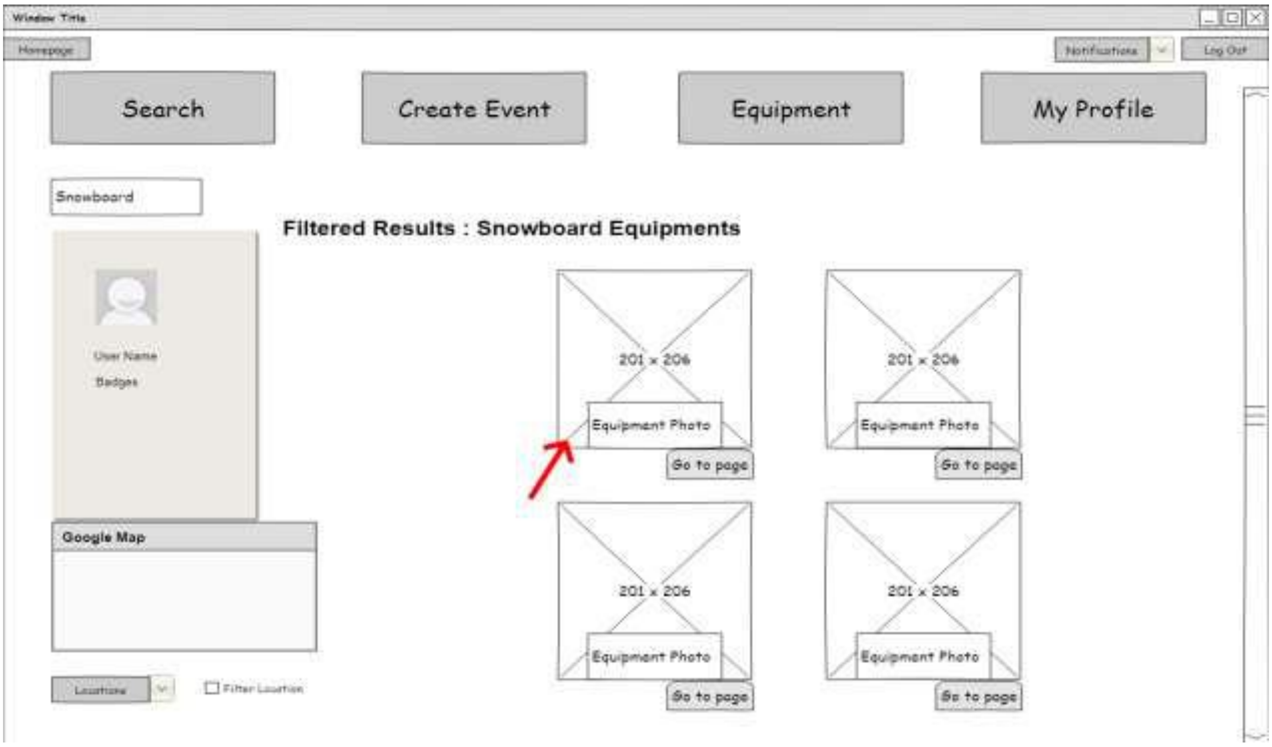
5. He sees several types of skiing equipment like snowboards, helmets, gloves, etc.
6. He clicks on 'Snowboard' from the types.



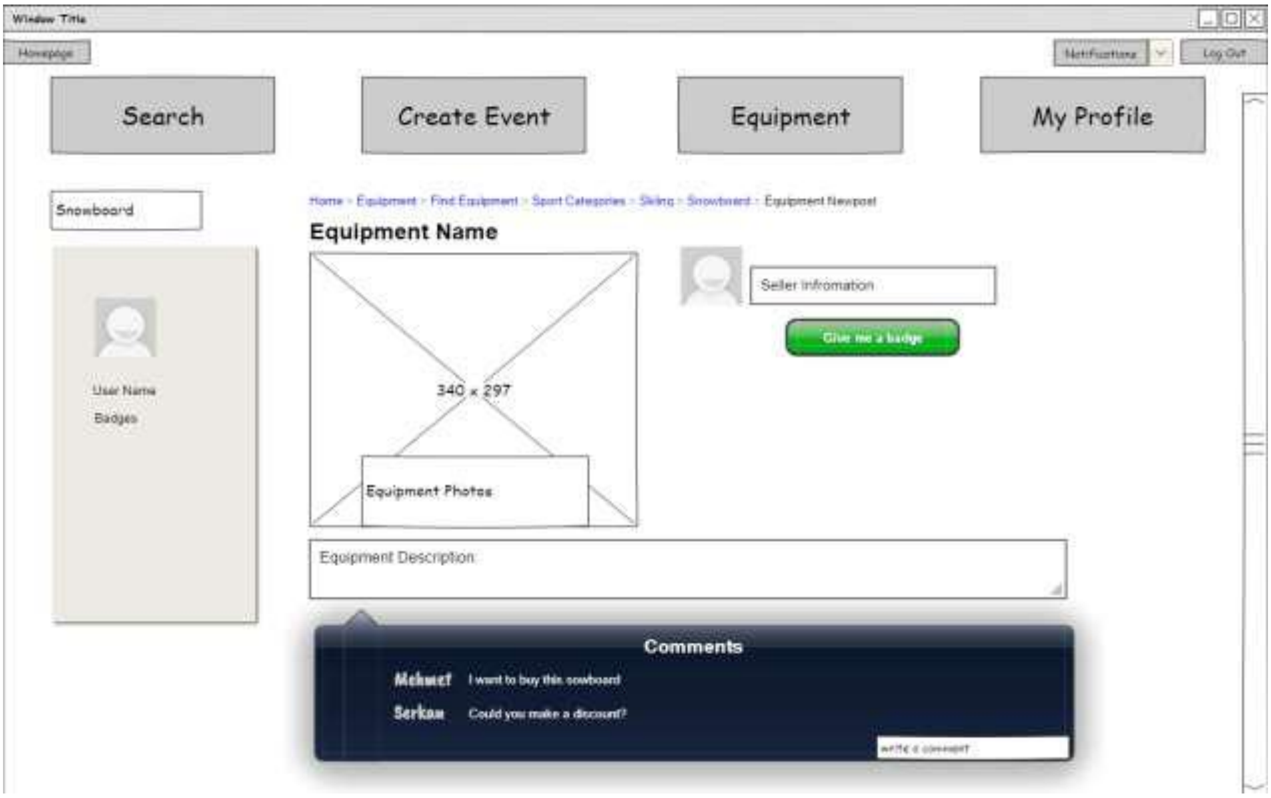
7. He sees various snowboard equipment posts (their photo, title, etc.) on this page. Then he clicks the 'Select Location' option to filter the results.



8. Snowboard equipment posts from the selected area on the map are visible on this page. He chooses one of them and clicks on their link.



9. He now sees the post. In this page, he can see photos and a description, comment and give a badge to the seller.

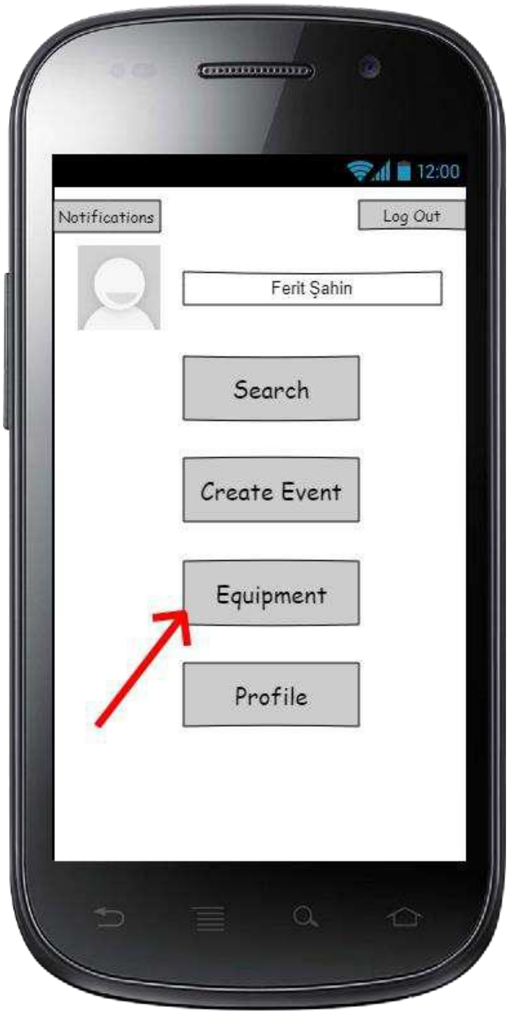


Mehmet I want to buy this sowboard

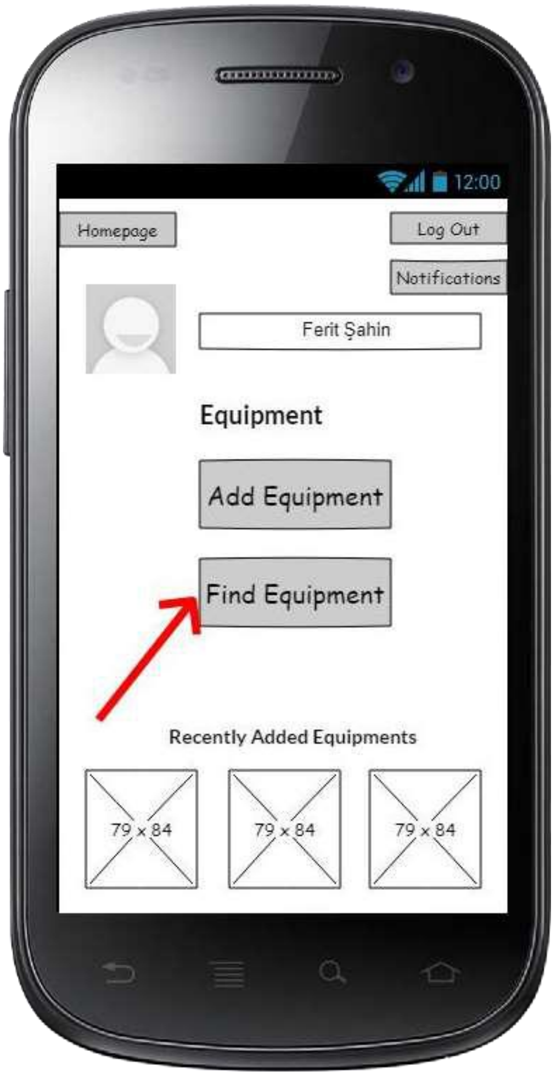
Serkan Could you make a discount?

Android Mockup

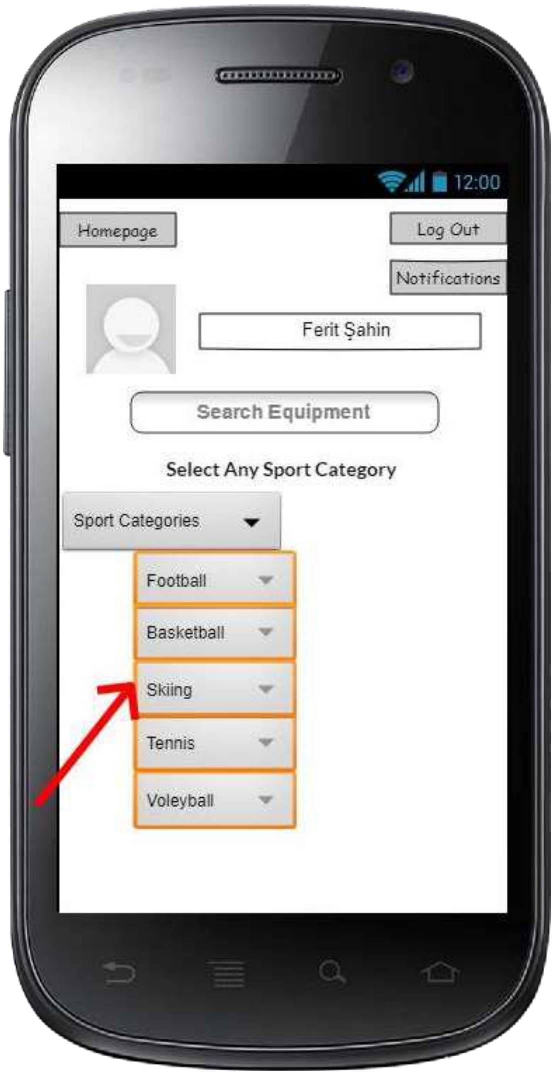
1. Ferit downloaded our application, and is a registered user of our app. He opens the application from his phone. He sees the homepage.
2. He clicks on 'Equipment' on the homepage.



3. He clicks on 'Find Equipment' on the 'Equipment' page.

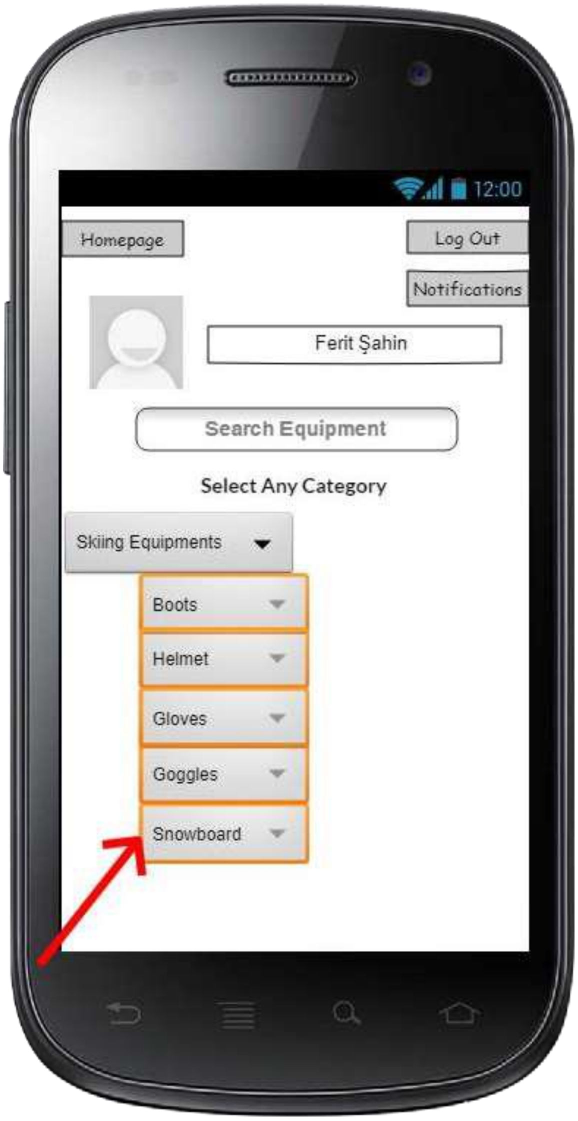


4. He clicks on 'Skiing' on the 'Sport Categories' page.

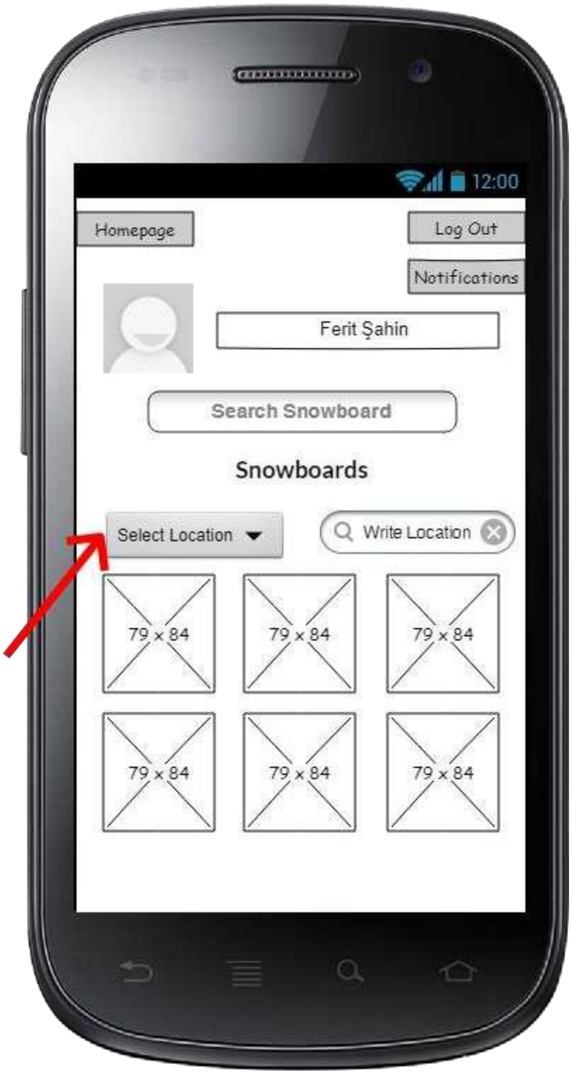


5. He sees several types of skiing equipment like snowboards, helmets, gloves, etc.

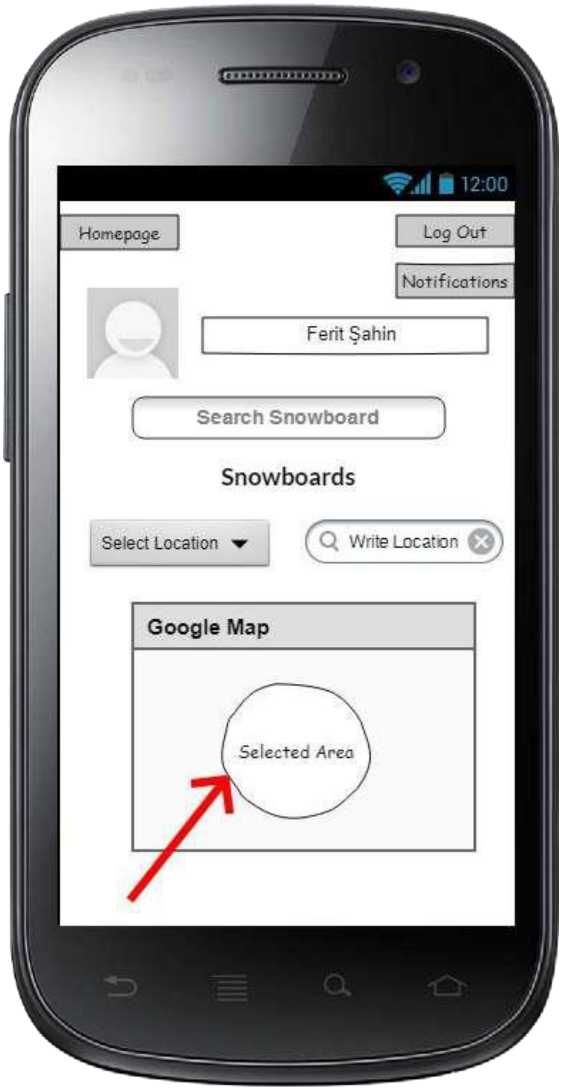
6. He clicks on 'Snowboard' from the category.



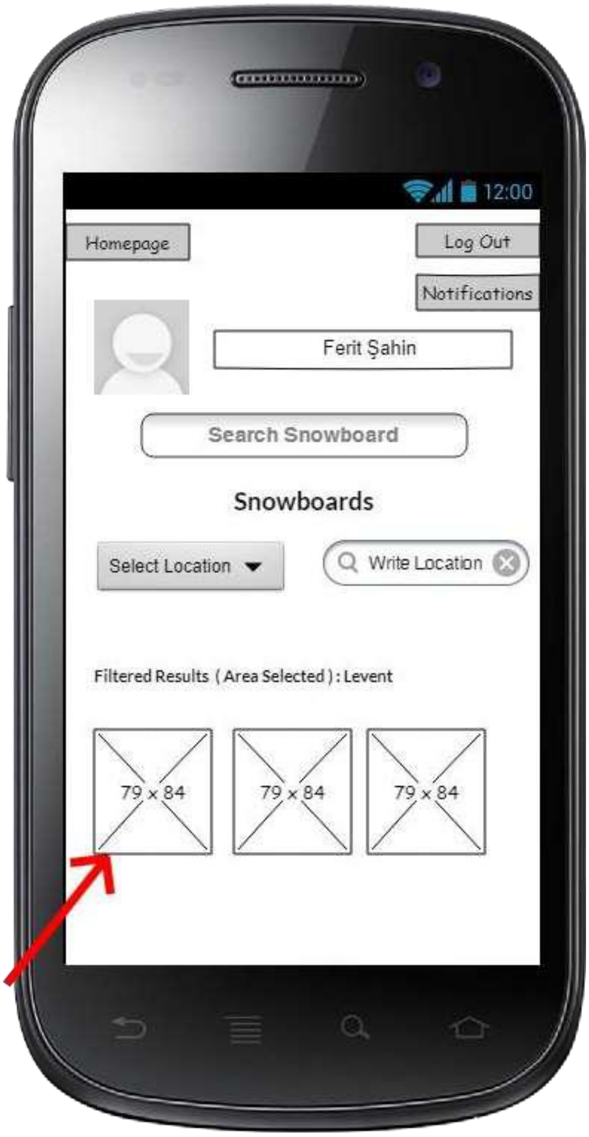
7. He sees various snowboard equipment posts (their photo, title, etc.) on this page. Then he clicks the 'Select Location' option to filter the results.



7.a. He selects an area on the map.



8. Snowboard equipment posts from the selected area on the map are visible on this page. He chooses one of them and clicks on their link.



9. He now sees the post. In this page, he can see photos and a description, comment and give a badge to the sell