MILESTONE REPORT 2

PROJECT DEVELOPMENT IN SOFTWARE ENGINEERING

Group Members

Berke Argın
Emrah Doğan
Muhammed Göktepe
Emir Hıfsı Öztoprak
Bilal Tekin
Elif Akalın
Yunus Emre Topal
Orkan Akısu

CMPE451
Fall 2021

1.Executive Summary

Project Description

Group Name: ProtoPost
Tag-id: v0.1
Group Logo:

We are designing a social media platform called Protopost that enables its users to create and share information in a community by creating posts in specific formats. Each user can create or join a community of their liking and create posts in some predetermined post format of their community. These predetermined post formats, which are called post templates, are created by community moderators and used in categorizing posts in that community. Each post template contains several compulsory or optional data fields to be filled, which can be of many types including location, image, video, text, date or a selection. Users can also filter posts based on their post templates and use queries on several data fields on that post template. Post template feature aims to provide an effective and accessible information sharing by providing a well-defined and characteristic outline for each piece of information in that community. Each post must belong to a community and put into that community's post feed. Users can also add other users as a friend after mutual consent and see their friends posts in their own user feed. Communities can be public or private, which in turn, will also define the privacy setting of the posts in that community. Currently, we aim to develop this platform both as a web and mobile application.

In summary, we aim to create a digital platform that enables its users to be a part of a purposeful community with each having their own specialized language for sharing information. In our platform, each community can define distinct and well-defined blocks of information which paves way for users to create specialized communities around a purpose.

Project Status

First half of our application is ready as it holds most of the core functionalities such as creating and joining communities, creating custom post templates and posts and presenting them in both user-related and community-related feeds. Second half of the application is on privacy settings, user profile and advanced filtering features that we aim to present in the final milestone.

As the backend team, we have implemented most of the core API functionalities such as creating communities, posts and post templates as well as some methods to retrieve data related to these objects. These methods are all tested and deployed in a docker container in an AWS instance. We also implemented a registration system that utilizes the built-in Django token authentication system. Since we had a lead on

implementation as the backend team, we spent some time researching and planning on implementing advanced filtering and privacy features for the final milestone. We decided not to start implementation right away as we experienced some setbacks due to improper planning and insufficient research in Milestone 1. We also sketched some initial plans on implementing remaining features such as advanced filtering and privacy settings. Also, taking some lessons from the mistakes we made in Milestone 1, we tried to automate deployment by using docker-compose files and bash scripts, and also tried to deploy regularly.

As the frontend team,

Milestone 1: We have implemented Register and Login pages, Create Community, Create Post, Create Post Template, My Post, My Communities, Communities and Feed Page. Due to some problems in backend connection Create Post, Create Post Template Page and Create Community pages do not respond.

Milestone 2: We have solved the backend connection problem stems from CSRF Token and authentication cookies. We have added a logo and application name to the website. Styling on postcards changed. We used Material UI card components to display posts. Post Template Creation page design is changed. We are adding fields dynamically and getting field names and types from them. We have successfully connected this page to the backend thus, we can create a post template on our application. Also, the create post page was successfully connected to the backend. Users can list post templates and based on their selections they can create posts in a community using this template. Currently we have added a search feature for communities. Users can search a community with text search. We have added alerts on the post creation and post template creation page to notify users about the status of the request. We fixed the subscribe button reaction time. We have tried to implement google maps for location fields but we encountered some issues on sending data to backend so we didn't deploy that version. It is in our local computer and currently works on the create post page.

As the mobile team,

So far we implemented following features for Android: registration, login, viewing user feed, viewing posts created by user, viewing and searching communities, viewing users subscribed communities, viewing a post in detail, viewing a community in detail, logging out, creating a new community, creating a post template for a community and subscribing to/leaving from a community. Unfortunately we couldn’t finish implementing post creation feature. But this is not a surprise considering the fact that we are missing one person from our team. You can see only two people contributed to the mobile branch. Because of that sometimes we can’t keep up with the requirements. During these implementations, we used axios service for making api calls and react-navigation for navigating between screens. Also, we made some visual updates that are not shown in the demo. You can check them in this report as well.

Moving Forward

As the Backend team,

Milestone 1:

Our primary objective is to fix integration problems between Backend API and Frontend services. After that we will be looking to implement features on privacies, commenting, interactions between users, and extensive filtering and query mechanisms.

We also deploy our API service as HTTPS, by getting a SSL certificate. We are also planning to write a CI/CD pipeline that includes automatic testing, dockerization, and AWS deployment.

Milestone 2:

Our goals after this stage in the project are to add new advanced end-points. For example, advanced post filtering, location system, searching for users, etc.

We can also add likes, comments, share posts, adding tags, privacy settings and post suggestions to the users. However, these features are not our first priority.

We are also planning to create a folder with unit tests and write unit test files classified according to their functionalities in this folder. We also have to write a separate SetUp function for a test function, which unnecessarily lengthens the code. We plan to solve this problem as well as we can.

As the mobile team,

Milestone 1: We have some features we couldn't manage to implement in time. They are post creating, post template creating and community creating. We will start with working on them first. After that, we will focus on some additional features that are missing on our screens such as: searching communities, searching posts, additional setting options. Also, we know that our applications user interface can use some polishing to look better. We will create better styles to make our application beautiful.

Milestone 2: As we mentioned in the project status, we have implemented required parts except post creation. Right now, the post creation feature is our top priority and we will work on that first. Also, we said that we will work on styling and make the interface more appealing for our users. We have some unfinished work on styling (you can see in this report) and we will finish it before the next customer demo. After these we will create a profile page for our users and we will implement required features for our users. In addition to that, we will try to add commenting feature as well.

As the frontend team,

Milestone 1: we implemented most of the pages but there are some problems in backend connections due to CSRF token and authentication cookies. We will try to solve those problems as soon as possible. Then we will start to implement new features such as search and filtering, profile page and settings. We will improve our styling to provide a better view for our users.

Milestone 2: We researched docker for Frontend application. With this we are dockerizing our application and we can run it everywhere. We are planning to combine docker with AWS EC2 instance by launch template to run frontend application with just one click. When we want to run our code, with the launch template we say that run these commands to run our frontend code. Also, we are searching to create an autoscale group for our application to create new instances in case of high traffic. Beside these, we are improving our frontend style, and backend integrations. We will create a profile page for all users and integrate with the backend. Also, users will be able to favorite posts, make some comments and visit other user profiles. Google map will be integrated to the Frontend part for posts. When users create posts, they can share a location.

2.List and Status of Deliverables

Deliverable	Status	Frequency	Description
Github Wiki Page	In Progress	Updated as needed	Documentation for the project and up-to-date information on team members, weekly meeting notes, weekly assessments
Github Issues	In Progress	Daily	Current and past issues about the project, with helpful and colorful labels.
Meeting Notes	In progress	Weekly	A summary of the discussions and decisions of the weekly scheduled meeting. A table of the division of weekly assignments can be seen at the end of every page.
Requirements	Complete	Completed	Requirements to be satisfied by the project. This page also includes a glossary and comparison of our project with other similar social networking websites.,
User Scenarios & Mockups	Complete	Completed	This page includes three scenarios that showcase the several uses of our project. A mock ui for both android and web, as well as detailed scenarios can be found here.
Design Diagrams	Complete	If Needed	These diagrams include a use case diagram, a class diagram and a sequence diagram.
Milestone Report	Complete	Completed	This document.
Backend API Django Project	Partially complete	Weekly	Django project that contains our code, configurations and logic behind our backend API.
Deployed Backend API Service	Partially complete	Weekly	Our backend API service which it's Swagger page can be reached on http://3.249.82.166:8000/api/schema/swagger-ui/
Mobile Application	Partially complete	Weekly	Mobile part of our application. You can find the latest version of it in our github pages mobile branch.
Web Application	Partially complete	Weekly	Web part of our application. You can find the latest version of it in our github pages Web_ReactJS branch. http://52.209.232.175/

3.Evaluation of the Status of Deliverables

- **Github Wiki Page:** The wiki page is continuously updated. A summary of our scheduled meetings are added every week. Pages such as “requirements”, “scenarios” and “design documents” are added according to the type of our weekly assignments. This pages provides an overall view of our current work and our progress in a compact manner.
- **Github Issues:** All open and closed issues regarding our project can be found here. The progression of our weekly assignments can be seen here. Custom labels such as “help needed” and “review needed” are used to increase productivity and alert other team members to a member’s progress in an assignment.
- **Meeting Notes:** Meeting notes are taken every week during scheduled meetings. They are then assembled by a team member as a summary in markdown language and uploaded on the wiki.
- **Requirements:** The requirements elicitation of the project is completed and can be easily accessed using the sidebar of the wiki.
- **User Scenarios and Mockups:** A mockup of the web and android UIs as well as example use scenarios can be found here. They can be accessed using the sidebar on the wiki.
- **Design Diagrams:** Use case, class and sequence diagrams have been created for the project. They are more “first draft” than a “final version”, and some changes might be required as the project proceeds.
- **Milestone Report:** This report.
- **Backend API Django Project:** We created and pushed the Django project that contains implementations for our backend API. It currently contains core functionalities of our application including community, post and post template creation, subscription, and post retrieval.
- **Deployed Backend API Service:** We deployed our backend service to the AWS instance. It’s swagger page can be reached here. Currently, it contains some token issues that create some complications during Frontend communication which will be fixed as soon as possible.
- **Mobile Application:** We connected our mobile application to our backend server. Users can Register and Login to our application. When logged in, they can view all communities, their subscribed communities, their feed and posts created by them. They can click on posts to see their details and they can click on communities to see community pages. They can view community posts, subscribe to or leave from a community. Also, they can logout from the application. But there are some issues on the backend related to registration and community subscription/unsubscription functionalities. Once they are fixed, they will work as intended.
- **Web Application:** We connected our web application to the backend server. Users can register and login to our application. When logged in they will be redirected to the feed page and will see the posts of the communities they have subscribed to. There is a sidebar at the left side of the screen. From there they can go to the Communities page where they can see all the communities that have been created. Also there is Create Post Page to create posts in a community. To create a post they need to create a post template first so there is a Post Template Create page to fulfil this requirement. Users can see his posts by clicking the My Posts label in the sidebar. Also users can create Communities by clicking Create Community label. Finally, users can log out from the app. Web application can be accessed from here.

4.Evaluation of Tools and Processes Used

- **Github:** Github is the main platform where our project will be developed. We have only used the wiki and issue management features of github so far. It will be incredibly useful once we start coding, since it will let us keep record of our progress, and give the option to branch and/or merge as we try and add more features to our code base.
- **Slack:** We use slack due to its tight integration with other platforms. Channels and integration are the main features that make slack invaluable for us.
- **Zoom:** We use zoom for our weekly meetings, since everyone has zoom installed and configured on their computers for the lectures.
- **Discord:** We use discord for asynchronous communication. In addition to the channels, voice chat and livestream capabilities, we also use discord to communicate with our TA.
- **Moqups:** This tool was used to create the mockup UI for the android version of our project. Moqups is a streamlined web app that helps create and collaborate in real-time on mockups and diagrams.
- **Lucidchart:** Lucidchart was used for the use case, class and sequence diagrams. It allows multiple team members to work on the same diagram simultaneously, and offers UML support.
- **Google Docs:** Google docs was used to create this document. Similar to lucidchart, it allows multiple team members to work on the same document.
- **ProjectLibre:** We used the project management software ProjectLibre to produce a Gantt chart of our team effort until now. ProjectLibre is an open-source project management software that supports many features like Gantt charts, network diagrams and PERT graphs.

Backend:

- **Django Framework:** Django Framework is one the most popular python web framework that is built on developing web applications with Model-View-Template software pattern. It is a very flexible tool with many documentations and a vast library. We decided to use this framework as we had some experience with it from last semester and were largely satisfied with its functionalities.

- **Django REST framework**: Django REST Framework is the most widely used library for building Web APIs in Django Framework. It provided us an additional layer of abstraction over Django framework such as serializer classes that enabled us to manipulate and retrieve data with ease. It also provides built-in request and response wrappers that reduce boilerplate code and simplify managing API entrypoints and endpoints, which was quite useful for our project.

- **Docker:** Docker is a container based virtualization tool that enables developers to develop and deploy their applications on clean, isolated environments called containers. We decided that proper dockerization of our application would be vital together with a CI/CD pipeline.

- **Swagger:** Swagger is an open source automatic documentation tool for RESTful APIs. This documentation is readable not only for humans but also for machines. It is easy to use and help developers to understand API endpoints fast by providing supported operations, parameters and outputs, authorization requirements, licenses needed.

- **MySQL:** MySQL is an open source relational database management system. It uses SQL for query language and is most commonly used in Web Servers.

- **Amazon AWS:** AWS is a cloud computing provider that provides a combination of many systems such as data storage, data processing, content delivery services to meet the service needs of businesses.

Web Frontend:

- **create-react-app:** This is a very important framework which is for creating single page react applications.

- **React-Pro-Sidebar:** This is an npm package which is created for React applications. This can be installed via npm tools.

- **React-Bootstrap:** This is a very popular style framework which is used for creating a very good looking web screens.

- **React Hooks:** React hook APIs provide an alternative to writing class-based components, and offer an alternative approach to state management and lifecycle methods.

- **React Axios:** Axios is a library that serves to create HTTP requests that are present externally.

- **React Redux:** React Redux is the official React binding for Redux. It allows React components to read data from a Redux Store, and dispatch Actions to the Store to update data. Redux helps apps to scale by providing a sensible way to manage state through a unidirectional data flow model.

- **React Navigation:** This library is used for redirecting users to any screen with parameters without refreshing the webpage.

- **Material UI:** MUI provides a robust, customizable, and accessible library of foundational and advanced components, enabling developers to build their own design system and develop React applications faster.

Android:

- **React Native:** React Native is a framework to develop mobile applications using React features as well as some other features developed for mobile. React Native is a powerful but hard tool to use. We struggled at the start but we learned how to use it effectively.
- **React Navigation:** React Navigation is a package developed for handling navigation inside the application. It has several different types of navigators such as Stack, Drawer, Tab. We used Stack Navigator for navigating between Welcome, Login, Registration Screens and we used Drawer to show 5 screens that a logged in user can access. It helped us a lot since navigation is an essential part for our application.
- **Expo Go:** Expo is a framework that lets us develop and deploy React applications much faster. Instead of running our application on a virtual device, we can just connect our phones to test our application and that accelerates the development time quite a lot. But while using that feature, we missed something. That is the difference between iOS and Android. Some stylings changed from iOS to Android on React and

we tested our application on our Iphones which resulted 2 white buttons on our demo.

- **React Axios:** Axios is a package to make API calls and fetch the result from an external website. Using it instead of the default fetch() function helps us to manage requests from a single page and with that, we don't have to worry about changing the URL of our server. Because instead of fixing the URL in every request code, we can just change it in one place.

5.Summary of Work Done

Member Name	Contributions
Elif Akalın	•
Berke Argın	• Attended to the lab sessions • Attended to group meetings • Fixing CORS and CSRF errors • Fixing the authorization problem of the API • Implementing Token-based Authentication Scheme • Configuring Docker files for container-based deployment • Writing some basic bash scripts for automated deployment • Reconfiguring GetCommunityData endpoint • Reconfiguring Community Serializer class • Doing preliminary research for advanced filtering implementation • Connecting Web-side to the Backend with Web Team • Doing preliminary research for privacy features • Researching APIs for geocoding • Implementing SearchPostsInCommunity method • Writing unit tests and testing code. • Reviewing pull requests of my teammate Emrah Doğan • Preparing the customer demo scenario • Creating issues for project management • Creating informative pull requests • Debugging and enhancing existing API endpoints • Refining swagger documentation
Orkan Akısu	
Emrah Doğan	• Attended to the lab sessions • Attended to group meetings • Fixing CORS and CSRF errors • Implementing GetHomeFeed method • Implementing SearchCommunities method • Doing preliminary research for advanced filtering implementation • Doing preliminary research for privacy features • Researching APIs for geocoding • Debugging existing methods from Milestone 1 • Enhancing endpoints based on frontend requests • Writing unit tests for functions • Adding some changes on existing unit test functions • Connecting Web-side to the Backend with Web Team • Debugging and enhancing API endpoints • Adding definitions to the Swagger documentation • Creating issues for Backend side • Reviewing pull requests of my teammate Berke Argın • Adding fields to Backend for Location and Date • Adding Json-LD headers to the Backend Endpoints

	(Not Deployed yet)(Backend-Emrah-JsonLD) ● Keeping Backend Notes for Meeting 4 ● Keeping Group Meeting at 02/12/2021 ● Keeped Group Meeting at 16/12/2021 (Not added to the wiki yet) (Coming Soon !!!)
Muhammed Göktepe	● Updating Wiki Page for CMPE451 ● Updating Sidebar for CMPE451 ● Connecting Create Post Template Page to backend ● Updating Create Post Template Page design ● Connecting Post Page to backend ● Tried to implement google maps API but encountered some issues. ● Added project logo and project name to the site. ● Fixed image display on posts using Carousel ● Presenting Web Frontend on Customer Demo 2 ● Merged Web branch to master branch for tagging. ● Wrote project status for frontend ● Updated project plan on Project Libre ● Updated RAM ● Updated tasks on tabular format ● Uploading this report and project plan to github repo ● Attending lab sessions ● Attending group meetings ● Reviewing pull requests made by Bilal and Emir ● Creating issues for Frontend development status ● Creating issue for project development ● Creating pull requests for Frontend
Emir Hıfsı Öztoprak	● Attending class and labs ● Attending weekly meetings with our group ● Learning more about React and React-redux ● Connecting Create Post page to backend ● Adding alert components to multiple parts ● Editing the design of the pages I previously implemented ● Implemented Search Community feature to All Communities Page ● Fixed the bug with the subscribe button in pages where the communities were listed ● Changed the design and the basic structure of the side card of the communities ● Creating 2 pull requests to Web_ReactJS branch ● Creating issues related to these mentioned tasks. ● Reviewing pull requests made by Muhammed and Bilal ● Research on different 3rd party APIs related to location feature.
Bilal Tekin	• Making some meetings with frontend team to evaluate the process we made and make some plans about the project. • Deploying frontend application to EC2 instance. • Reviewing frontend team code • Reviewing pull requests made by Muhammed and Emir • Creating issue for project development

	• Creating pull requests for Frontend • Filling necessary parts of the Group Milestone 2 • Attending group meetings. • Attending lab sessions • Attending group meetings • Trying to create posts and integrate with backend • Trying to create post template and integrate with backend • Searching docker to dockerize our applications • Searching AWS Launch template to run our code on EC2 servers with just one click. • Searching to integrate a domain to our frontend part. • Searching SSL to secure our frontend connection with backend connection • Updating weekly effort page in github
Yunus Emre Topal	• Presenting Customer Demo 2 • Attending lab sessions • Attending group meetings • Creating Community Creation screen for Android • Implementing Community Creation feature for Android • Creating Create Post Template screen for Android • Implementing Create Post Template feature for Android • Converting fetch api calls in Login and Register screen to axios api calls • Rearranging Post screen to show more detailed information about posts • Implementing Search Community feature for Android • Fixing Subscribe to a Community feature for Android • Reviewing 2 pull requests from my group member • Merging mobile branch with master branch for tagging • Filling Project Status part for mobile • Filling Moving Forward part for mobile • Filling Project Plan part for mobile • Filling tabular format of RoadMap part for mobile • Filling user interface part in Milestone 2 group report

6.Deliverables

a.Communication Plan

Participants	Place	Purpose	When
Team Members	Zoom	General Discussion	Every Thursday at 20.00
Team Members	Slack	Weekly Progress Review	When Needed
Team Members	Discord	General Discussion	When Needed
Team Members	WhatsApp	Urgent Communication	When Needed

b.Requirements

Green -> Completed
Yellow -> In Progress
Red -> Not started

1. Functional Requirements

1.1 User Requirements

- **1.1.1 Registration and Log in**
  - 1.1.1.1 A user shall be able to register by providing a valid e-mail address, a username, and a strong password.
  - 1.1.1.2 Users shall be able to register by providing their twitter or facebook social media accounts.
  - 1.1.1.3 Users shall be able to log in to their account with their email/username and password.
  - 1.1.1.4 If a user logged in to their account, then the next time they open the app they will be logged in automatically.
  - 1.1.1.5 Users shall be able to login by providing their social media account which they used during registration.
- **1.1.2 Home Page**
  - 1.1.2.1 Users should be able to see posts from their communities in their home page.

    - 1.1.2.2 Users should be able to see the communities that they're joined.
    - 1.1.2.3 Users should be able to name search and list other users provided that their accounts are public.
    - 1.1.2.4 Users should be able to name search and list communities.
    - 1.1.2.5 Users should be able to title search and list post in their home post feed.
- **1.1.3 Settings Page**
    - 1.1.3.1 Users should be able to change mode of the app (light and dark mode).
    - 1.1.3.2 Users should be able to disable notifications.
    - 1.1.3.3 Users should be able to log out from the app.
    - 1.1.3.4 Users should be able to add profile pictures from settings menu.
    - 1.1.3.5 Changing password/username
        - 1.1.3.5.1 Users shall be able to change their username if the username they wish to use is available.
        - 1.1.3.5.2 Users shall be able to change their password with e-mail confirmation.
    - 1.1.3.6 Account Privacy
        - 1.1.3.6.1 Users shall be able to make their profile public or private.
        - 1.1.3.6.2 Users shall be able to accept or send follow requests from/to other users.
        - 1.1.3.6.3 Users shall be able to block other users. Blocked users can't see the activities of that user and also can't message them.
- **1.1.4 Community Features**
    - 1.1.4.1 A user shall be able to create a community by specifying a unique name, description and privacy status (public or private). After a successful creation, user shall be a moderator of that community.
    - 1.1.4.2 A user shall be able to view a community page that will contain community post feed and information.
    - 1.1.4.3 Users shall be able to join to any public community.
    - 1.1.4.4 Users shall be able to leave a community that they're currently joined.
- **1.1.5 Moderating Features**
    - 1.1.5.1 Moderators can ban users if the users engage in an ill behaviour.
    - 1.1.5.2 Moderators should be able to assign other users as moderators.
    - 1.1.5.3 Moderators should be able to send users email invitations to the community.
    - 1.1.5.4 Moderators should be able to delete the community that they moderate.

        - 1.1.5.5 Moderators should be able to create community specific post tags that can be used in the posts.
    - **1.1.6 Post Templates**
        - 1.1.6.1 Community moderators should be able to create post templates for their communities.
        - 1.1.6.2 Community moderators should be able to select and add multiple data fields to the post templates.
        - 1.1.6.3 Users should be able to list and select post templates during post creation.
    - **1.1.7 Posts**
        - 1.1.7.1 Users should be able to see post feed of a community in its respective community page.
        - 1.1.7.2 Users should be able to create a post in a joined community by selecting a post template and filling out necessary fields.
        - 1.1.7.3 Users should be able to sort post by various criteria in post feeds.
        - 1.1.7.4 Users should be able to add existing tags to their posts.
        - 1.1.7.5 Users should be able to search for titles of posts with keywords using search bar in both home and community post feed.
        - 1.1.7.6 Users should be able to filter posts in terms of their communities, post templates, and the query methods defined for the data field types in the post template.
        - 1.1.7.7 Users should be able to like or comment to posts in both home and community post feed.

1.2 System Requirements

- **1.2.1 Notification**
    - 1.2.1.1 System notifies users when there is a new post in their communities.
    - 1.2.1.2 System notifies users when people like their posts.
    - 1.2.1.3 System notifies users when people comment their posts.
    - 1.2.1.4 System notifies users when people like comments.
    - 1.2.1.5 System notifies users when people comment comments.
    - 1.2.1.6 System gives an option to users disallow notifications that come from
        - 1.2.1.6.1 posts
        - 1.2.1.6.2 comments
        - 1.2.1.6.3 likes
- **1.2.2 Users and Communities**
    - 1.2.2.1 System must ensure uniqueness of the username and e-mail before registration.

    - 1.2.2.2 Users shall enter their password 2 times to prevent typos in registration.
    - 1.2.2.3 To register, users must confirm their e-mail adress using a link sent by the system.
    - 1.2.2.4 System must ensure uniqueness of the name of a community before community creation.
    - 1.2.2.5 System will not permit usage of priveledged moderator actions by regular users of the community.
    - 1.2.2.6 Users shall be redirected to their homepage after a successful login.
- **1.2.3 Post Templates and Data Fields**
    - 1.2.3.1 System must support post templates that contain multiple data field types including but not limited to text, image, location.
    - 1.2.3.2 System must be able to render posts that contain different data field types correctly in the UI.
    - 1.2.3.3 System must ensure the exclusivity of post templates to their respective communities (i.e. System should not allow post template usage outside its community.)
- **1.2.4 Recommendation**
    - 1.2.4.1 System must be able to recommend new communities to the users according to their current communities.
    - 1.2.4.2 System must be able to recommend the most followed communities to the users.
    - 1.2.4.3 System must be able to recommend the most liked posts in the users joined communities.
- **1.2.5 Search/Filter**
    - 1.2.5.1 System must support text search on names/ titles of users, communities and posts.
    - 1.2.5.2 System must support tag based filtering on posts.
    - 1.2.5.3 System must support queries on data fields based on their type. These queries will be used by users to filter specific posts.

2. Nonfunctional Requirements

2.1 Availability

- 2.1.1 Application shall be available on Android based mobile application stores.
- 2.1.2 System shall be accessible using modern web browsers.

2.2 Standards

- 2.2.1 The semantic taggings should be supported with Wikidata.org
- 2.2.2 The system must follow the rules of W3C Activity Streams Standards 2.0

2.3 Privacy

- 2.3.1 Users shall agree to the User Terms & Policy Agreement before registration.

2.4 Security

- 2.4.1 User's password will be encrypted with SHA-256 and only encrypted passwords will be contained in the database.
- 2.4.3 JWT or Authentication Tokens will be used to authenticate users.

c.Scenarios and Mockups

1. Scenario 1

Persona

- John Doe
- 23 Years Old
- Boğaziçi University Junior Student
- Lives in Hisarüstü
- Currently Pursuing Bachelor's Degree in Computer Engineering
- Admin of the Boğaziçi Housing Community

Story

- John Doe is an enthusiastic user of our app.
- He had to change 4 houses in 2 semesters, which he did not enjoy. He wants to make sure everyone can find a house in Hisarüstü at reasonable prices.
- He wants to make sure the Boğaziçi Housing community is a spam-free place.
- He checks for spam content daily and bans user accordingly.
- He just broke his old phone, and he is using a brand new phone.

Preconditions

- He is not a first-time user. He actually founded the Boğaziçi Housing community. However, he just broke his phone and he has to sign in for the first time on this phone.

Goals

- He wants to login with his new phone.
- He wants to check for spam content and ban users.

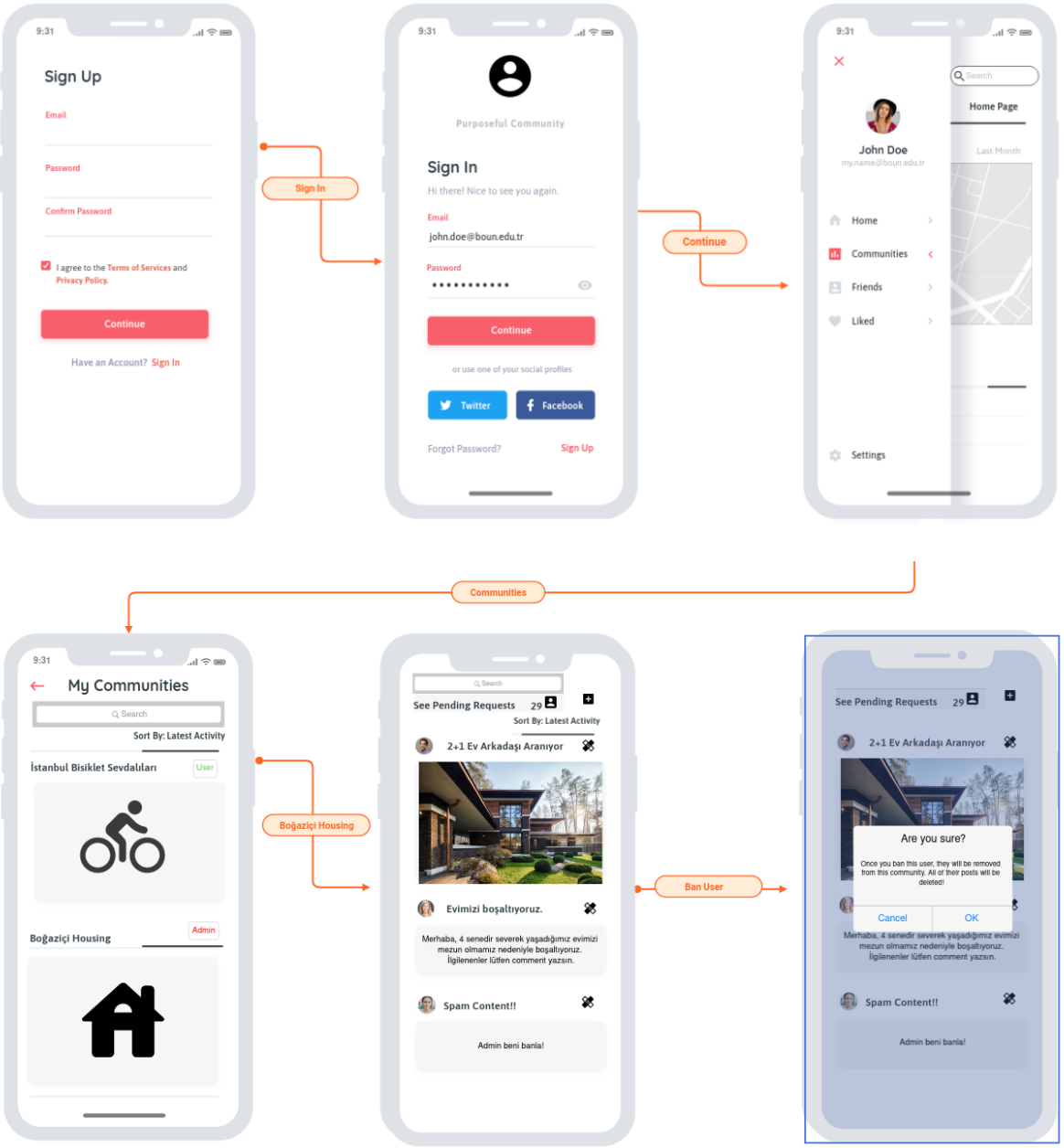
Acceptance Criteria

- (1.1.1.1) Unregistered users shall be able to register either using an username/password or their social media accounts.
- (1.1.1.2) Registered users shall be able to login either using their username/password or social media.
- (1.1.1.3.2) Users who forgot their password shall be able to reset their passwords by clicking "Forgot Password?".
- (1.1.1.2.4) Users shall be initially directed to their homepages.
- (1.2.5.3) Users shall be able to switch to their homepage, communities, friends and liked posts.
- (1.2.5.2 and 1.2.5.5) Users shall be able to search and filter posts according to several criteria.
- (1.2.8) Users shall be able to configure privacy options as well as app options from the settings menu.
- (1.2.6.2 and 1.2.6.4) Users shall be able to search for and sort communities by several criteria.
- (1.2.6.3) Users shall be able to see which communities they moderate.
- (1.2.7.5 and 1.2.7.6) Users shall be able to search for and sort posts by several criteria.

- (1.2.7.3) Users shall be able to report posts they deem unfit for the community.
- (1.2.7.4) Users shall be able to post on communities they belong to.
- (1.2.7.7) Community admins shall see an admin-only toolbar where they see pending requests, reported posts and so on.
- (1.2.7.8) Community admins shall be able to either delete posts or ban users.

Scenario with Mockups

- John Doe was an admin, but he broke his phone.
- He bought a new phone and installed the app.
- He opened the app for the first time on this phone. Since he never logged in on this phone before, he saw the sign-up screen.
- He tapped sign-in, since he was already an active user.
- He entered his username and password. He did not login using his social media, since he uses this app exclusively.
- He tapped continue.
- He was logged in and directed to his homepage. He used the sidebar to switch to communities view.
- He saw the 2 communities he was a part of: Istanbul bisiklet sevdalilari and Boğaziçi Housing. The communities were sorted by latest activity.
- He tapped Boğaziçi Housing, where he is the admin.
- He saw 29 pending requests, as well as a spam post.
- He decided to not only remove the post, but ban the user from the community.
- He saw a popup to confirm that he was sure.
- He banned the user from the community.


Sign Up
Email
Password
Confirm Password
I agree to the Terms of Services and Privacy Policy.
Continue
Have an Account? Sign In
Sign In
Purposeful Community
Sign In
Hi there! Nice to see you again.
Email
john.doe@boun.edu.tr
Password
Continue
or use one of your social profiles
Twitter
Facebook
Forgot Password?
Sign Up
Continue
Search
Home Page
John Doe
Last Month
Home
Communities
Friends
Liked
Settings
Communities
My Communities
Search
Sort By: Latest Activity
İstanbul Bisiklet Sevdalıları
User
Boğaziçi Housing
Admin
Boğaziçi Housing
See Pending Requests 29
Sort By: Latest Activity
2+1 Ev Arkadaşı Aranıyor
Evimizi boşaltıyoruz.
Merhaba, 4 senedir severek yaşadığımız evimizi mezun olmamız nedeniyle boşaltıyoruz. İlgilenenler lütfen comment yazsın.
Spam Content!!
Admin beni banla!
Ban User
Are you sure?
Once you ban this user, they will be removed from this community. All of their posts will be deleted!
Cancel
OK

2. Scenario 2

Sebastian Vettel

Persona

- Sebastian Vettel
- 33 years old
- He is from Heppenheim, West Germany
- He is married and has two children
- He is a Formula 1 driver for Aston Martin

Story

- Sebastian Vettel downloaded our app months ago and uses it to follow the news about F1 and F1 communities.
- Since he has been racing for 15 years, he has lots of friends who are Formula 1 drivers as well.
- Since F1 drivers travel around the world often, he can only keep in touch with his friends via our app.
- He wants to buy gifts for his friend Daniel Ricciardo for his birthday, but he forgot the exact date.
- He wants to make sure the date.
- He also wants to learn more about Daniel's hobbies so that he can buy a better gift for him.

Preconditions

- He is not a first-time user. He signs in with email and password.

Goals

- He wants to login with his phone.
- He wants to enter Daniel's profile.
- He wants to tap the "about" button to see Daniel's personal info.

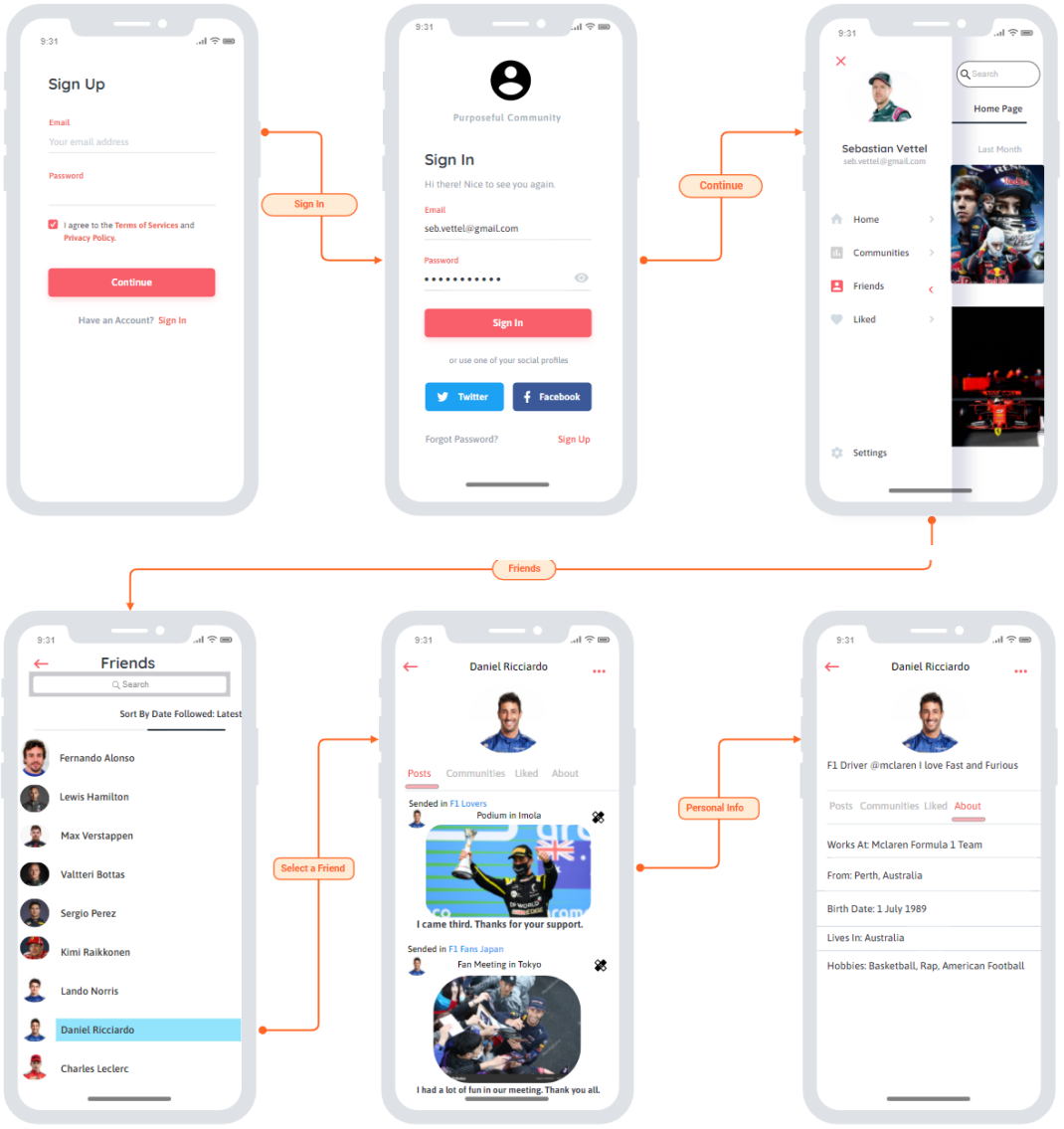
Acceptance Criteria

- (1.1.1.1.1 and 1.1.1.1.6) Unregistered users shall be able to register either using a username/password or their social media accounts.
- (1.1.1.2.1) Registered users shall be able to login either using their username/password or social media.
- (1.1.1.3.2) Users who forgot their password shall be able to reset their passwords by clicking "Forgot Password?".
- (1.1.1.2.4) Users shall be initially directed to their homepages.
- (1.2.5.3) Users shall be able to switch to their homepage, communities, friends and liked posts.
- (1.2.3 and 1.2.4) Users shall be able to search and filter posts according to several criteria.
- (1.1.1.4) Users shall be able to configure privacy options as well as app options from the settings menu.
- (1.2.6.4 and 1.2.7.5) Users shall be able to search for and sort friends by several criteria.
- (1.1.2.2) Users shall be able to post on communities they belong to.
- (1.1.1.4) Users shall be able to see information about their friends.
- (1.1.1.4.1) Users shall be able to see the public posts their friends shared.
- (1.1.1.4) Users shall be able to see the communities followed by their friends.
- (1.1.2.3) Users shall be able to see the posts that are liked by their friends.

Scenario with Mockups

- Sebastian Vettel was a user, and he has been using this app for four months.
- He opened the app on this phone and tapped sign-in, since he was already an active user.
- He entered his email and password.
- He tapped continue.
- He was logged in and directed to his homepage. He used the sidebar to switch to friends view.
- He saw his friends. They were sorted by date followed: latest.
- He tapped Daniel Ricciardo's page.
- He saw the posts Daniel sent to different public communities.
- He tapped the about button to see Daniel's personal information.

- He saw Daniel's bio and saw his birthday and hometown.
- He also saw the hobbies of Daniel and chose a gift based on these.



3. Scenario 3

Beth Harmon in 70's.

Persona

- Beth Harmon
- 73 years old
- Currently resides in Kentucky.
- Chess prodigy who currently holds a FIDE Elo 2673.

Story

- Beth Harmon is a professional chess player, mostly active in 60's and 70's.
- Even though her age, she likes to spend her time in a chess community to study chess and answer other peoples questions.
- She is currently retired, however she still follows the chess news and participates in tournaments.

Preconditions

- Beth already has an account and joined several communities.
- Beth is logged in to the website through a web browser.
- She is familiar with the community post filtering feature.

Goals

- Beth wants to browse the Chess Community page.
- She wishes to find a tournament which fits her schedule.

Acceptance Criteria

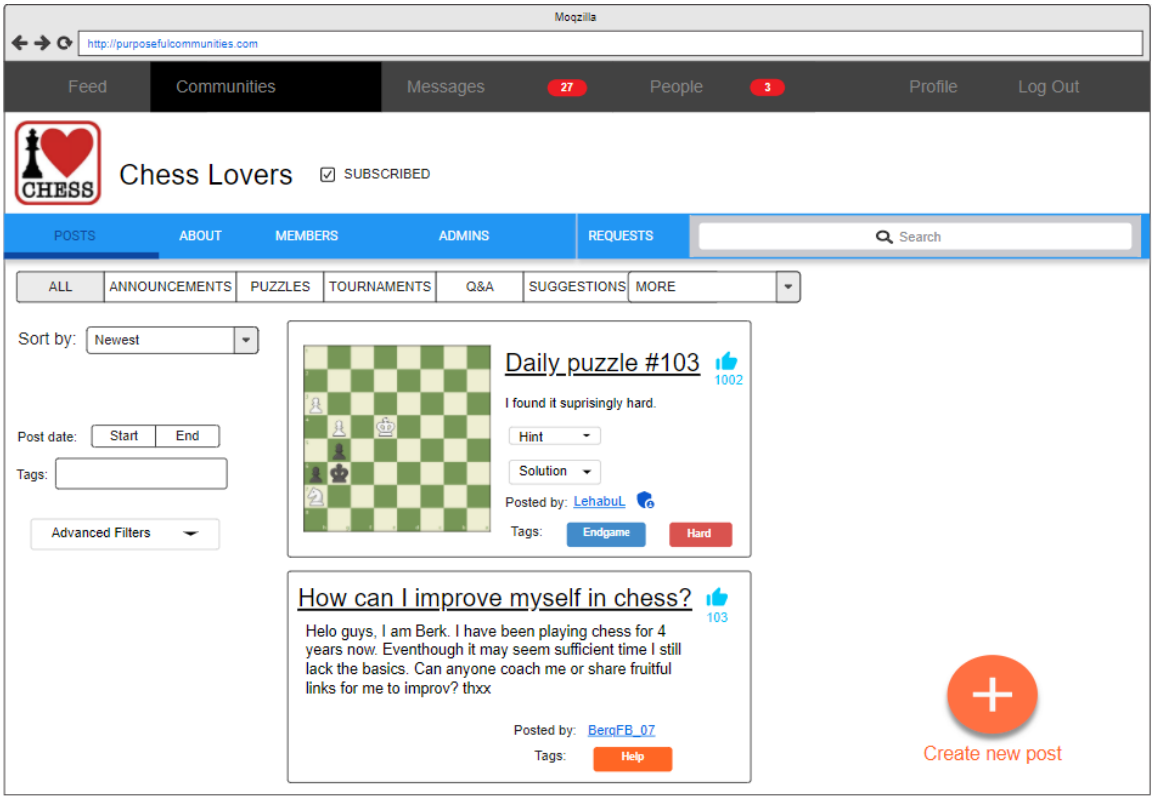
- (1.1.1.2.2) Users should be able to stay logged-in after closing the app or browser.
- (1.2.7.9) Posts must support multiple data field types including text, image, location.
- (1.2.4) For several types of data fields used in posts, there must be some predefined comparison -methods and queries to be used in filtering posts.
- (1.2.7.10) Users must use a predefined datatype template for posting.
- (1.2.4) Users should be able to add existing tags to their posts.
- (1.2.7.9) Post datatypes must support some text formatting including dropdown texts and hyperlinks.
- (1.2.3.3) Users should be able to browse posts in groups based on their datatypes.
- (1.2.7.11) Users should be search and filter specific types of posts with some predefined comparison methods and queries

Scenario with Mockups

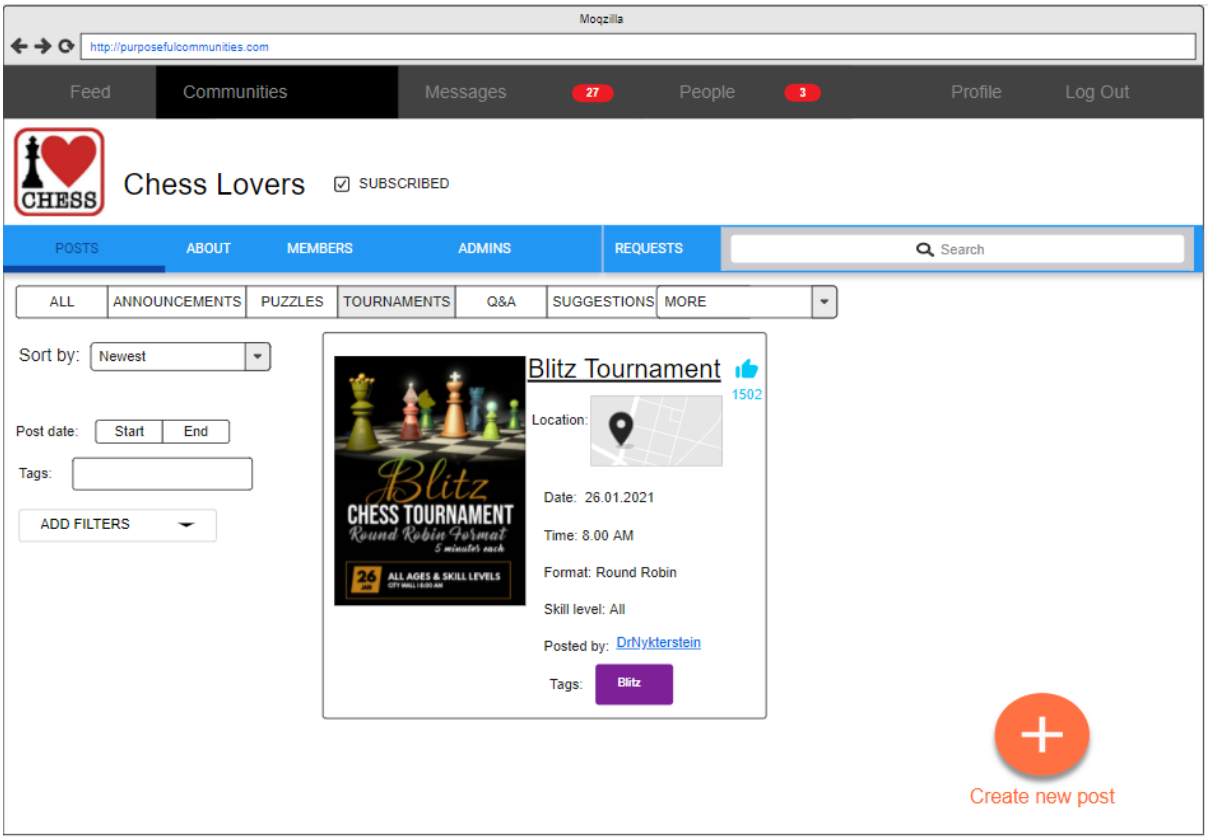
- Beth opens the "Chess Lovers" community page.
- She selects the "Tournaments" tab.
- She filters the tournaments posts by their location, date and time.
- She wishes to attend a tournament on that same day between 11.00 and 18:00.
- But there is no tournament fitting her preferences.
- So she decides to host her own tournament and let her fellow chess players know on the community.
- She presses the create new post button.
- She fills the predetermined fields of the tournament datatype and posts it.

Mockup

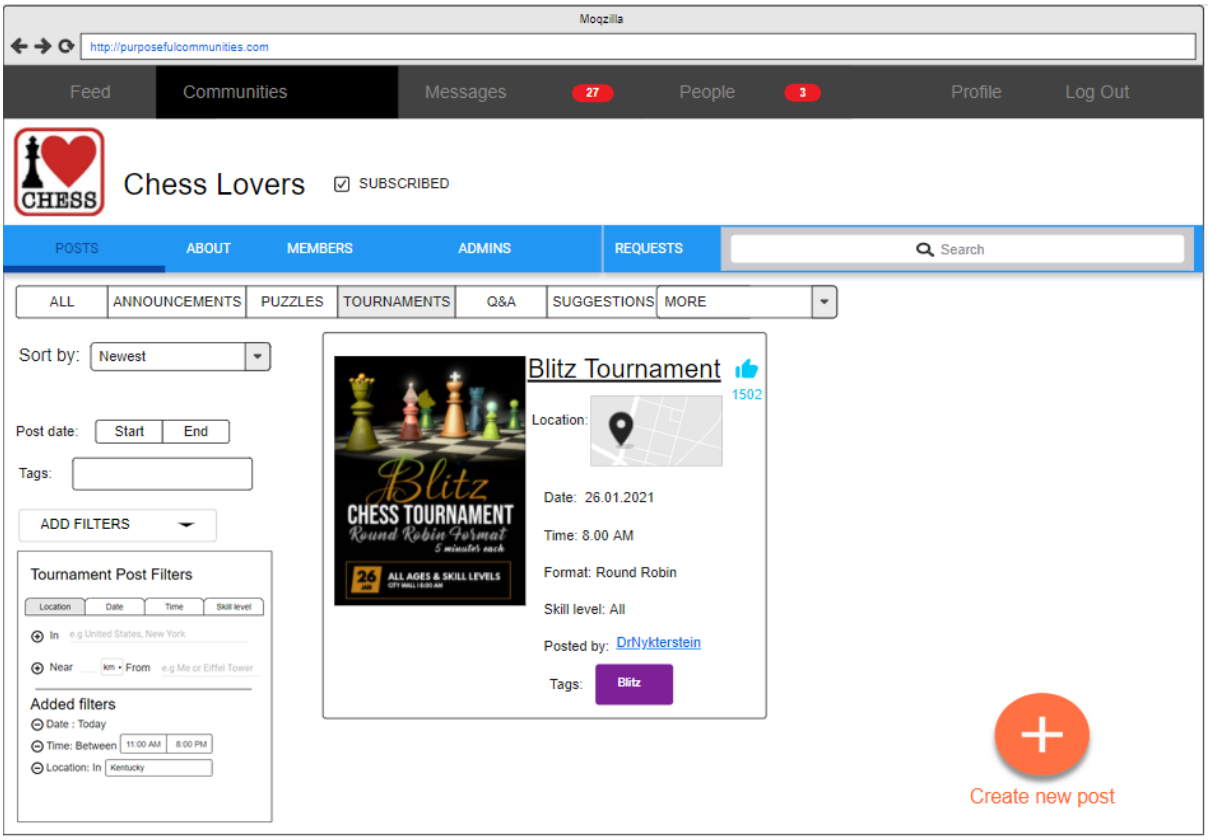
- Beth views the "Chess Lovers" page.



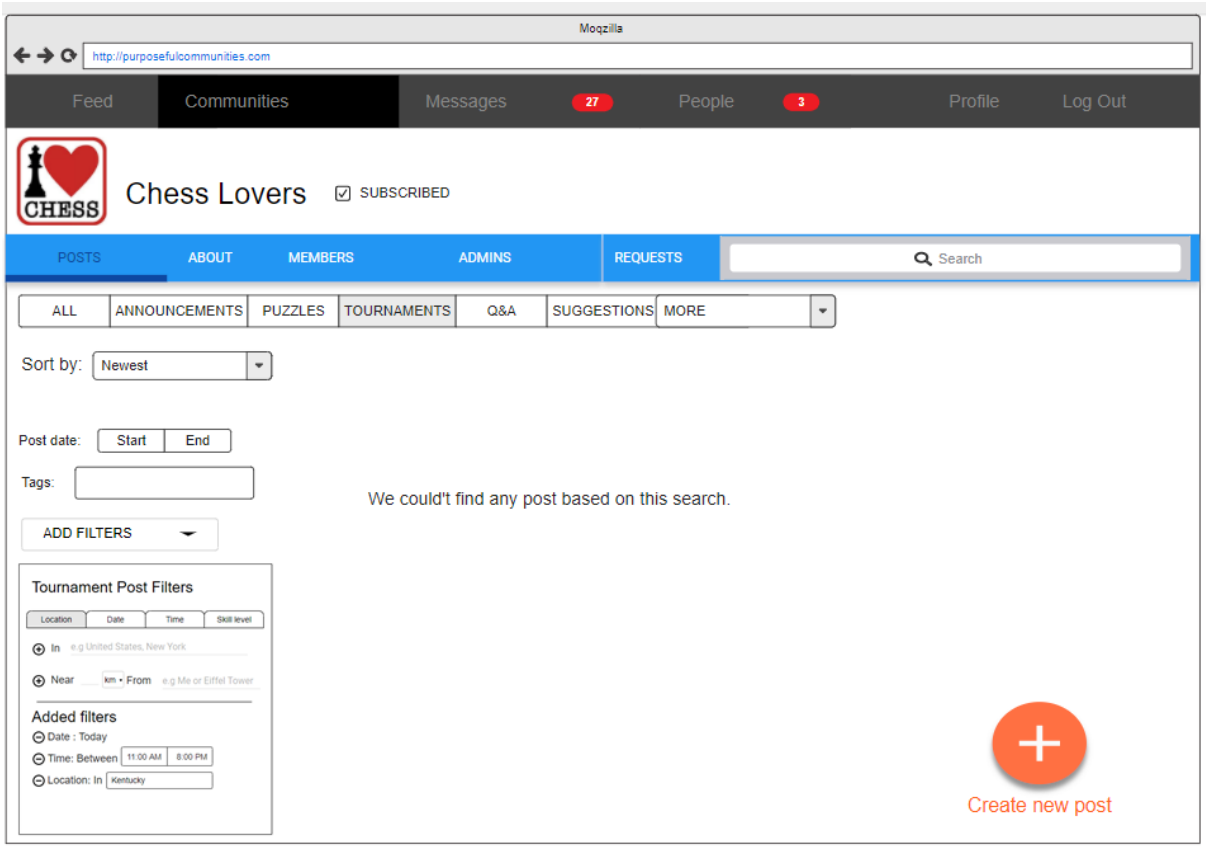
- She selects "Tournament" tab.



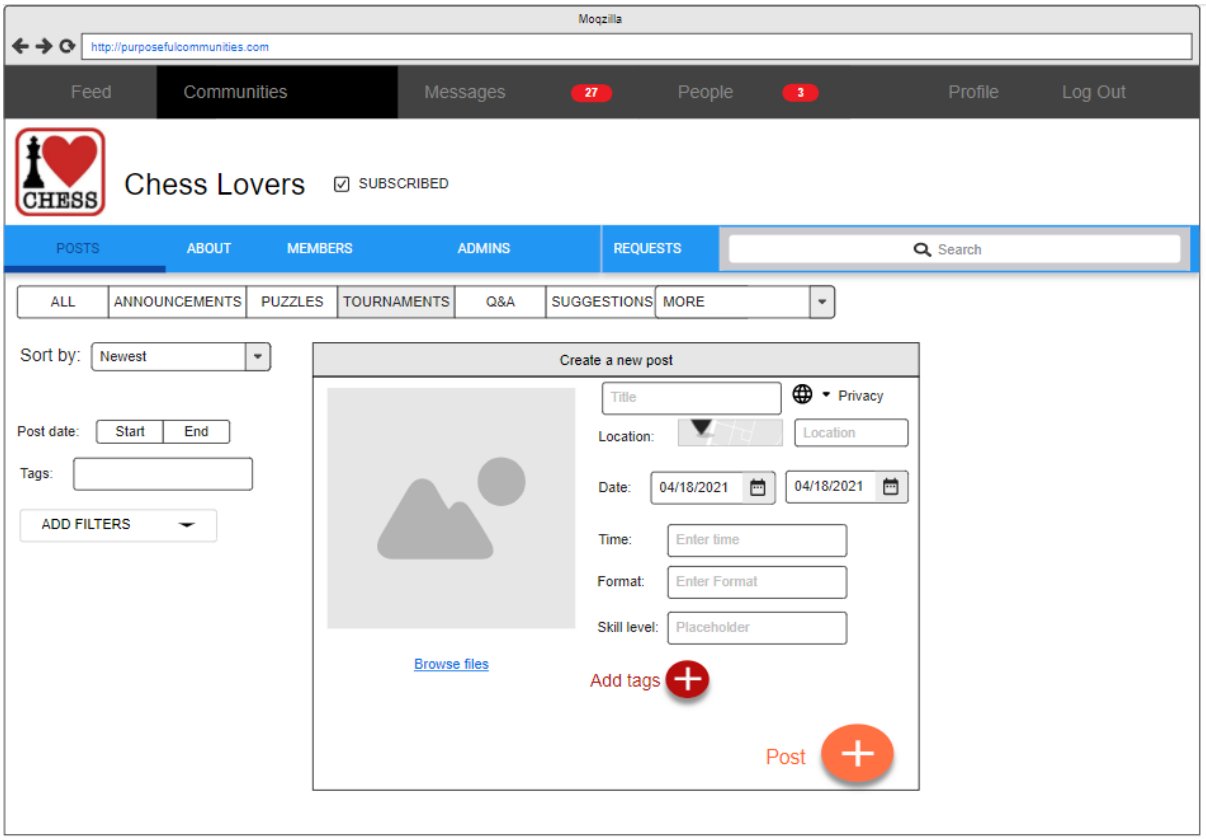
- She filters the posts on her preferences.



- There is no tournament fitting her.

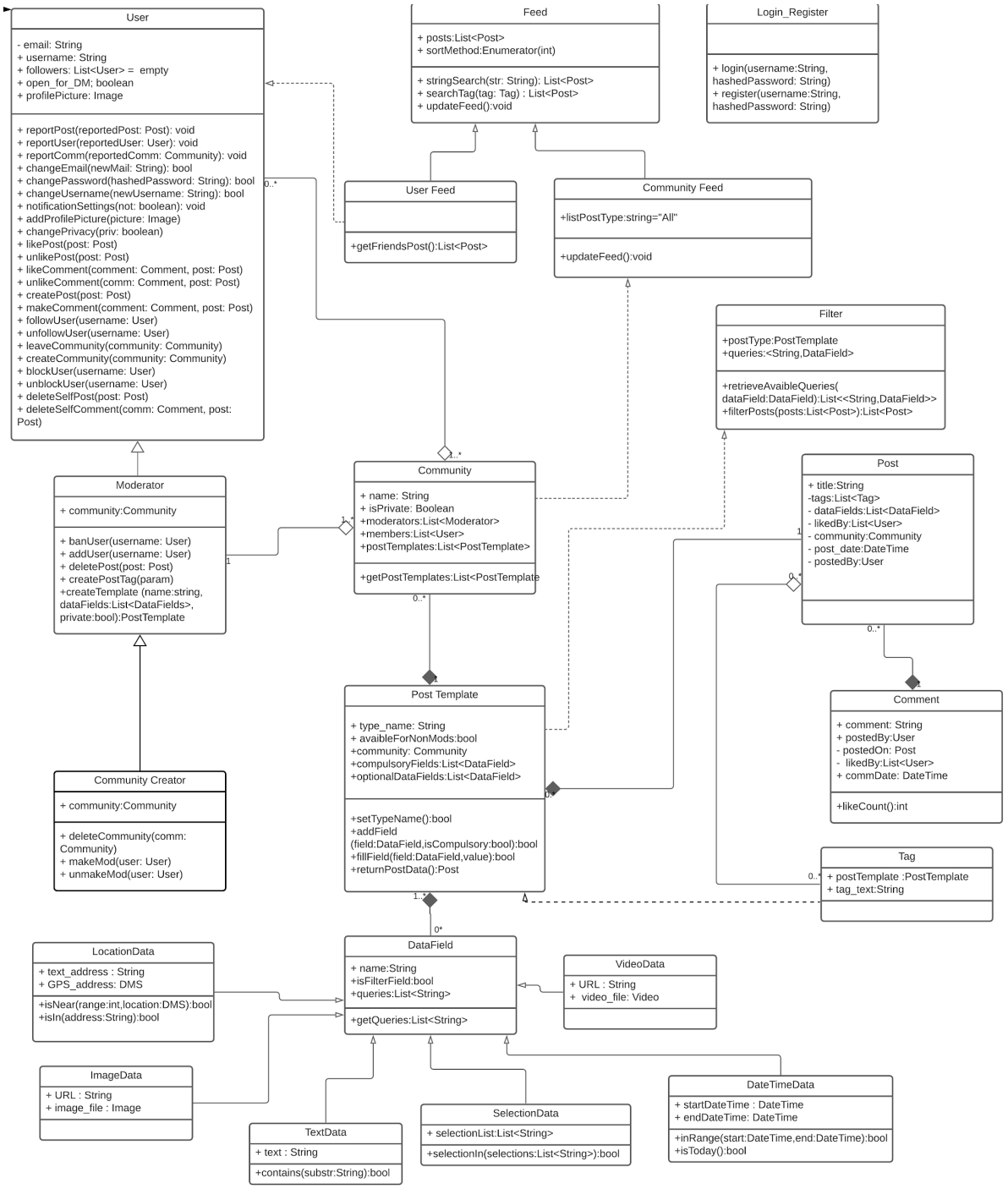


- She creates a new Tournament post.

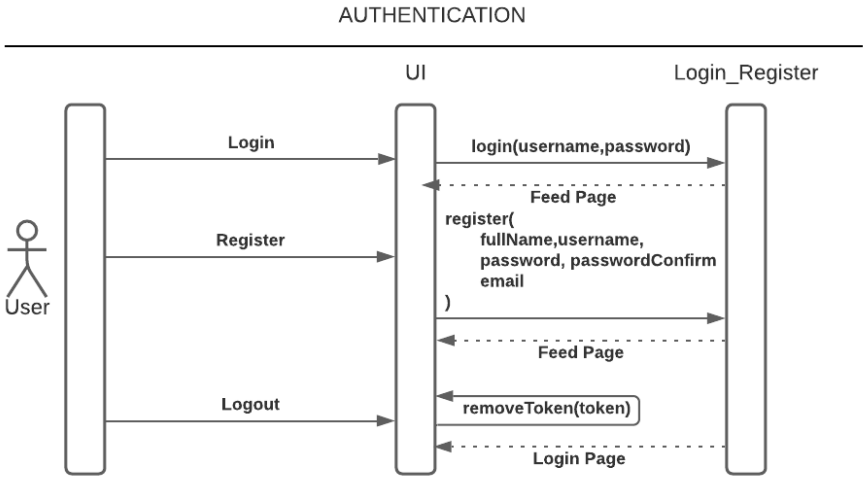


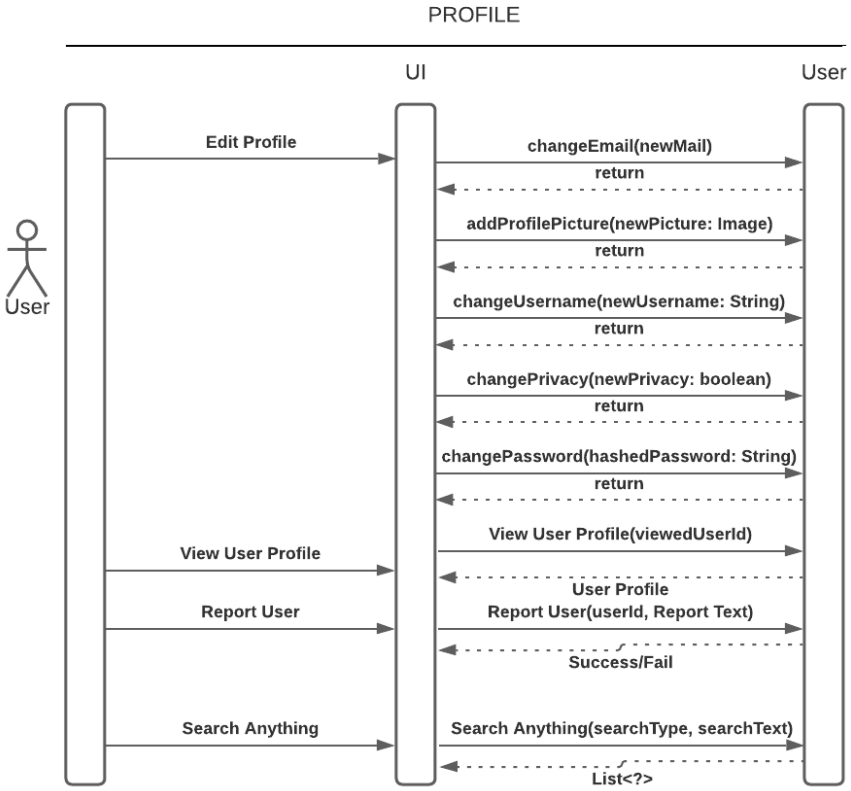
d.Software design documents in UML

1. Class Diagram

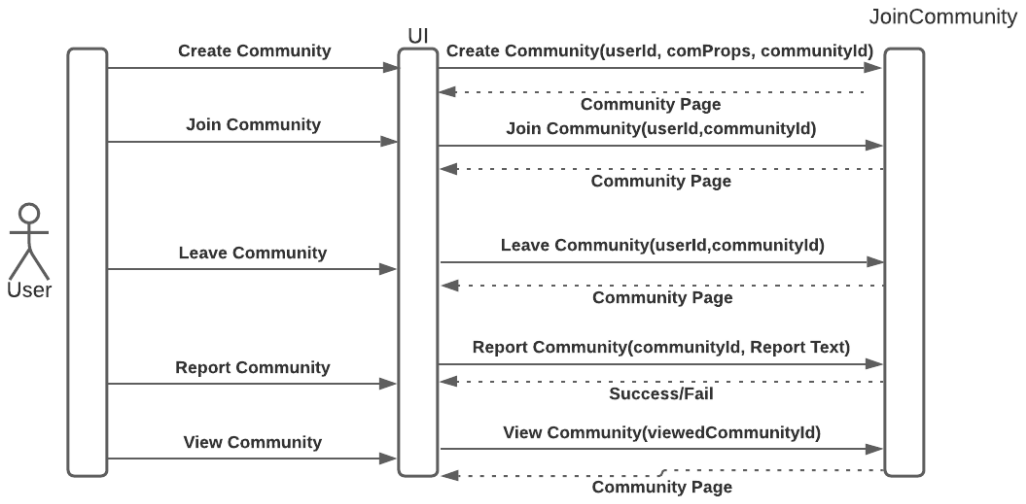


2. Sequence Diagram

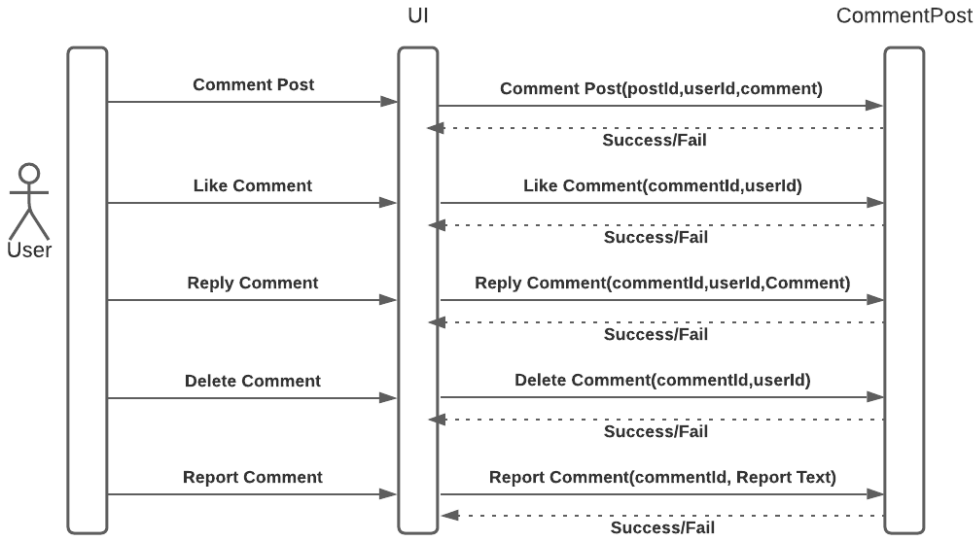




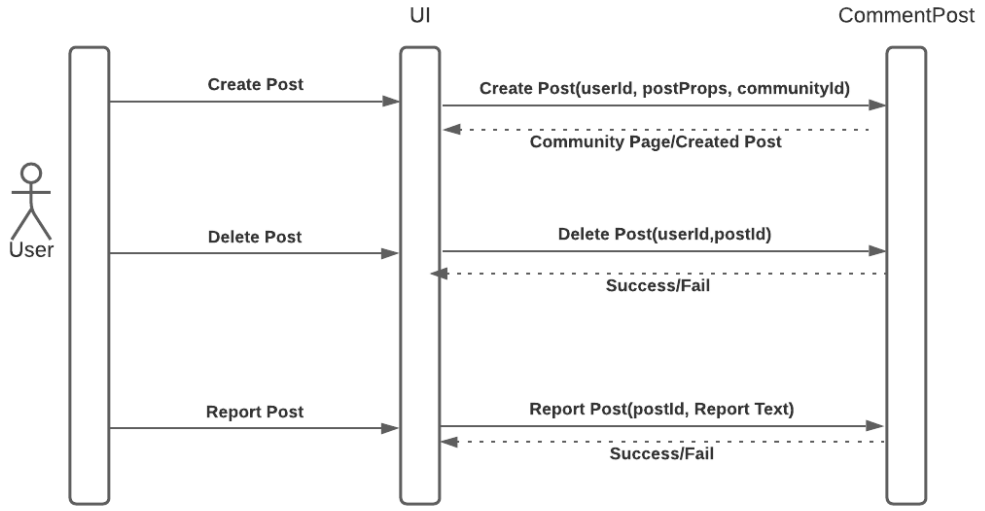
Community



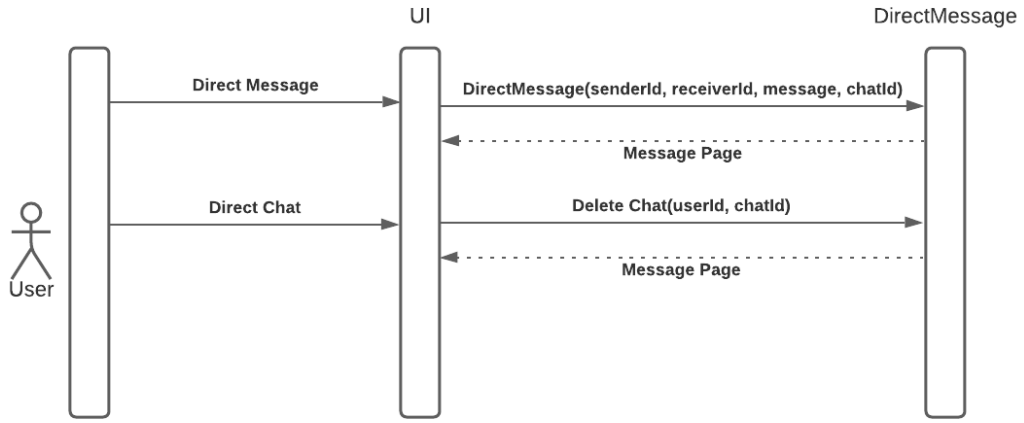
COMMENT



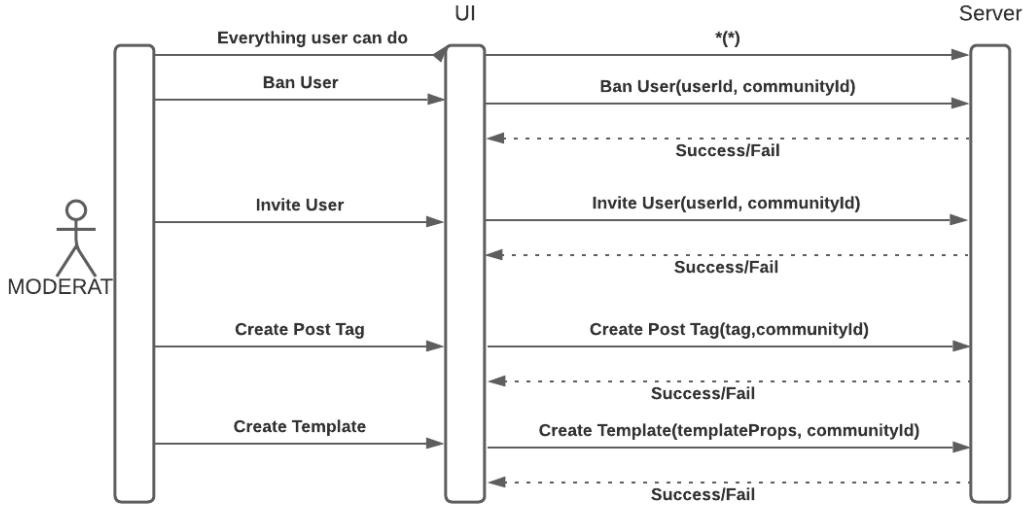
POST

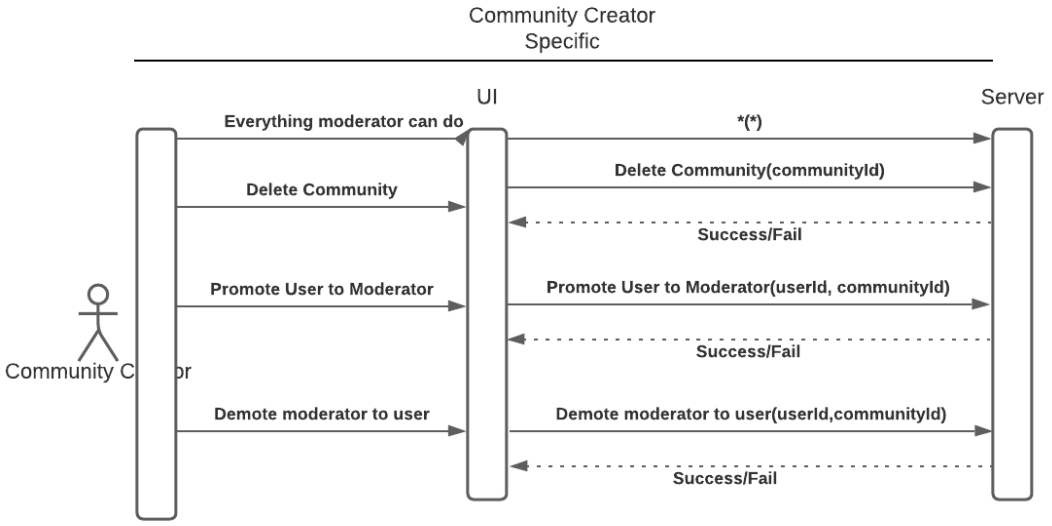


CHAT / MESSAGE



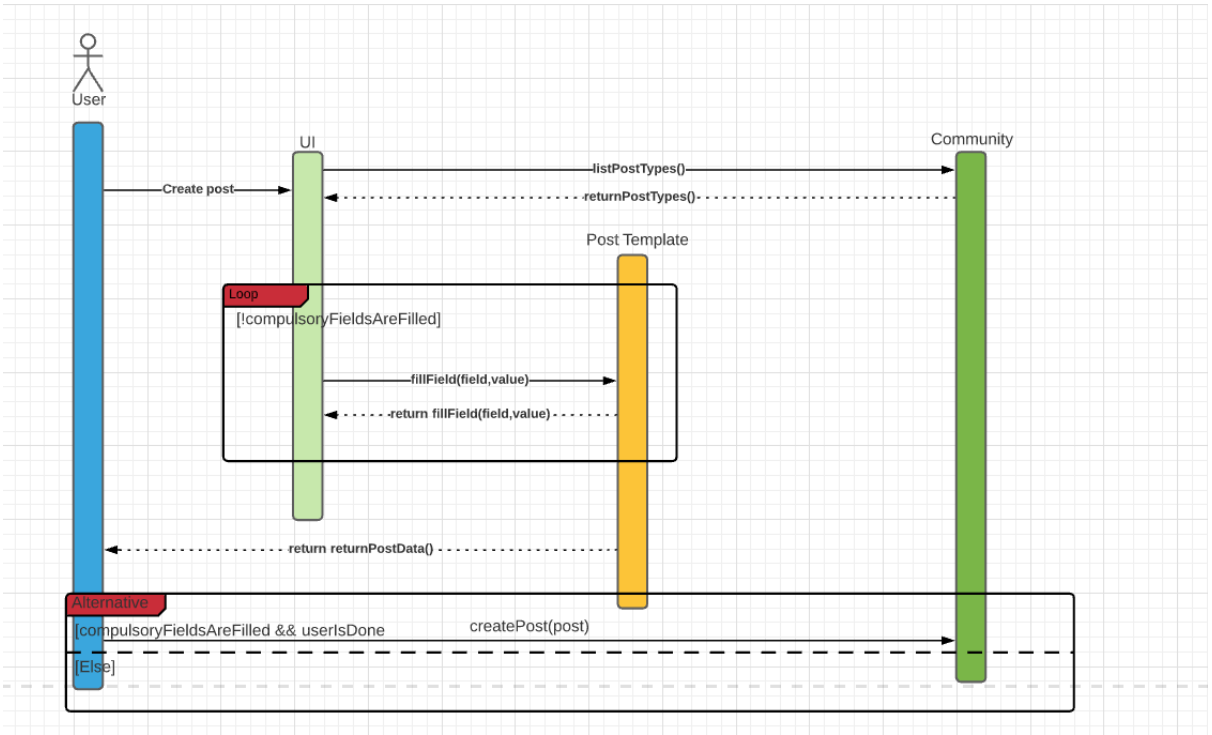
MODERATOR SPECIFIC





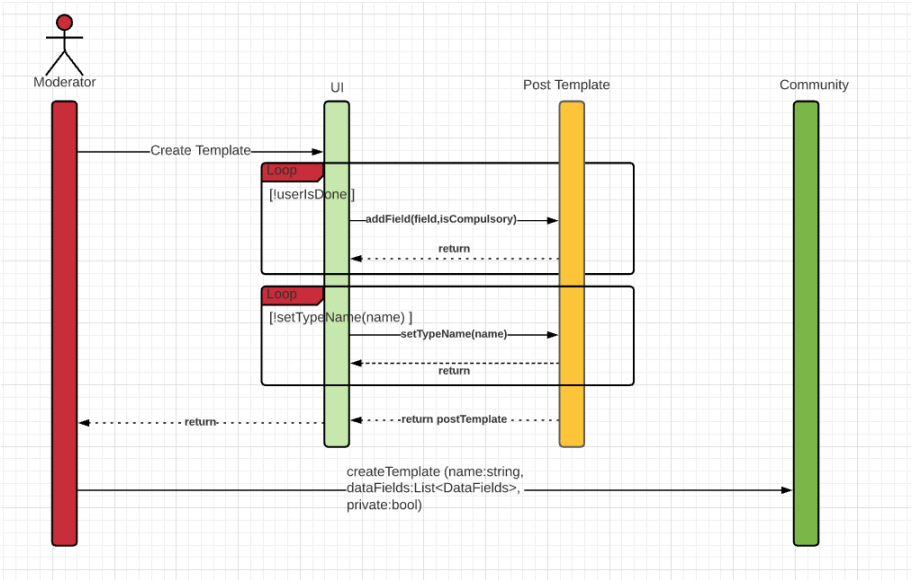
1. Creating Post in a Community

- User can create a post using existing post templates in community.



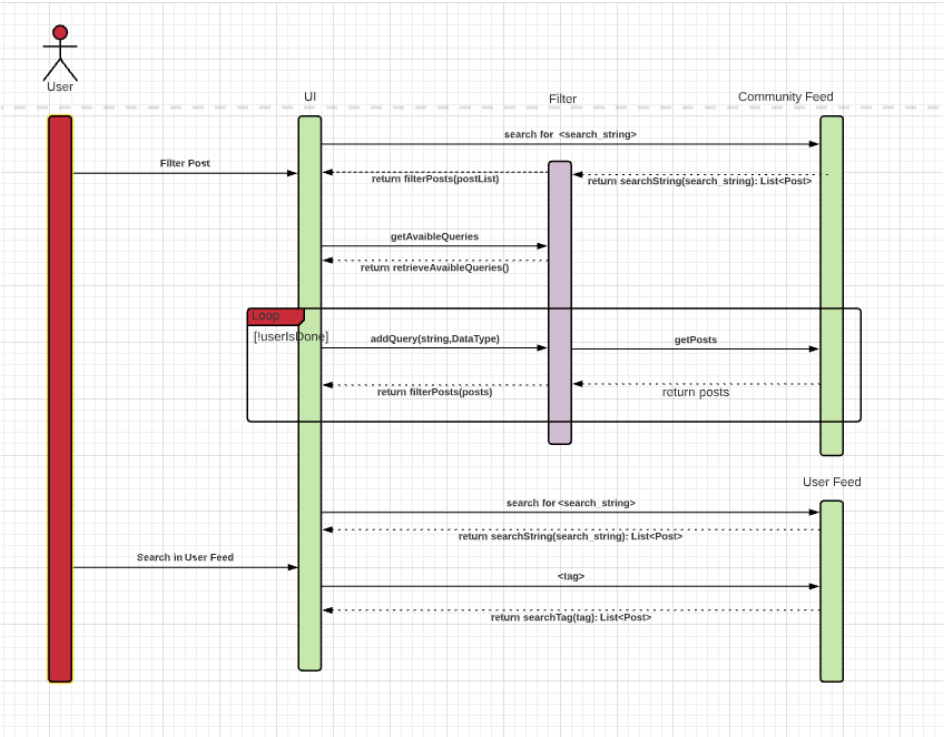
2. Creating Post Templates in a Community

- Moderators can create post templates in community.
- Moderators can add many fields to templates.

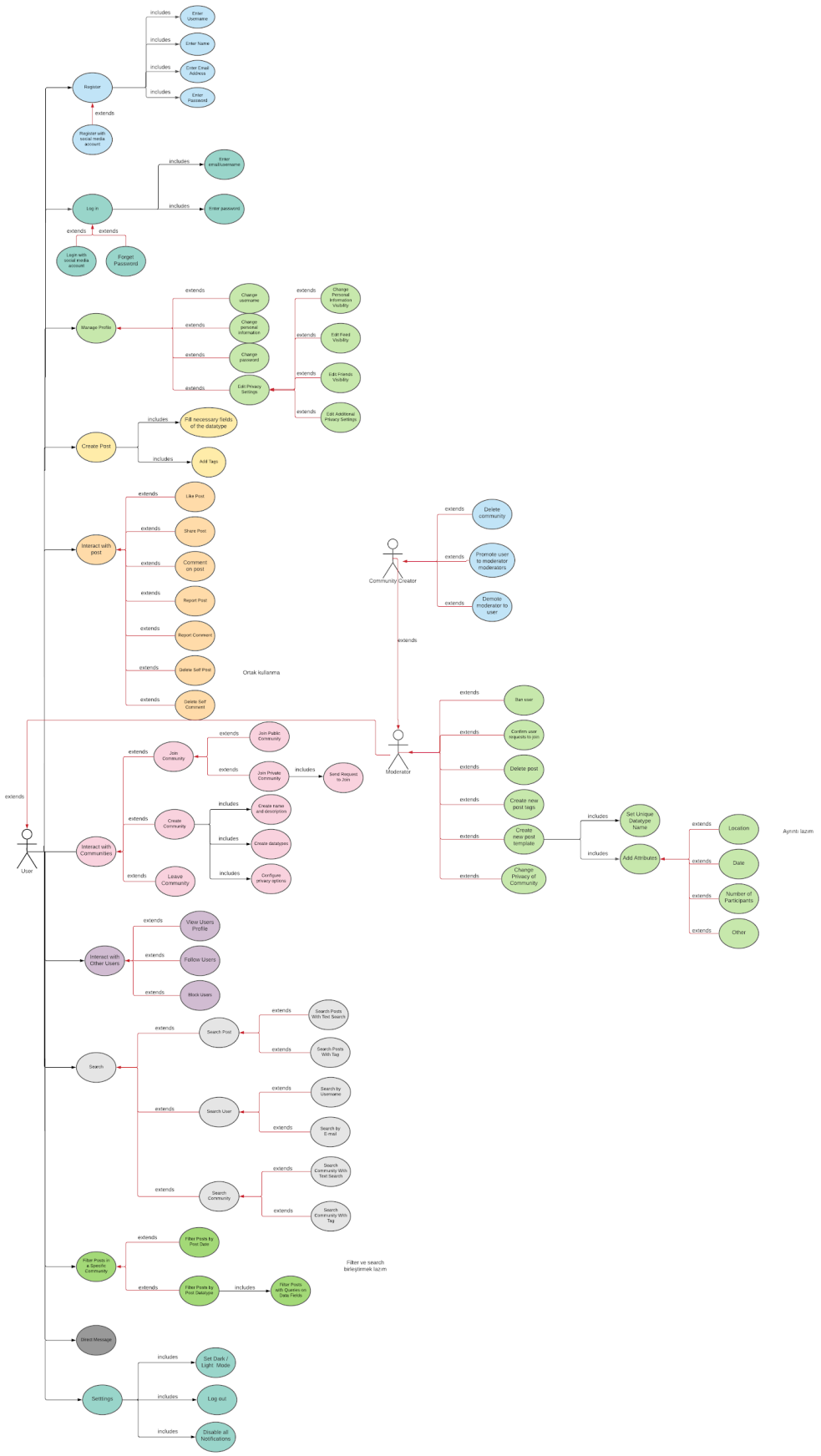


3. Filtering and Searching in Feeds

- Users can filter posts in communities.
- Users can search posts in their feed.



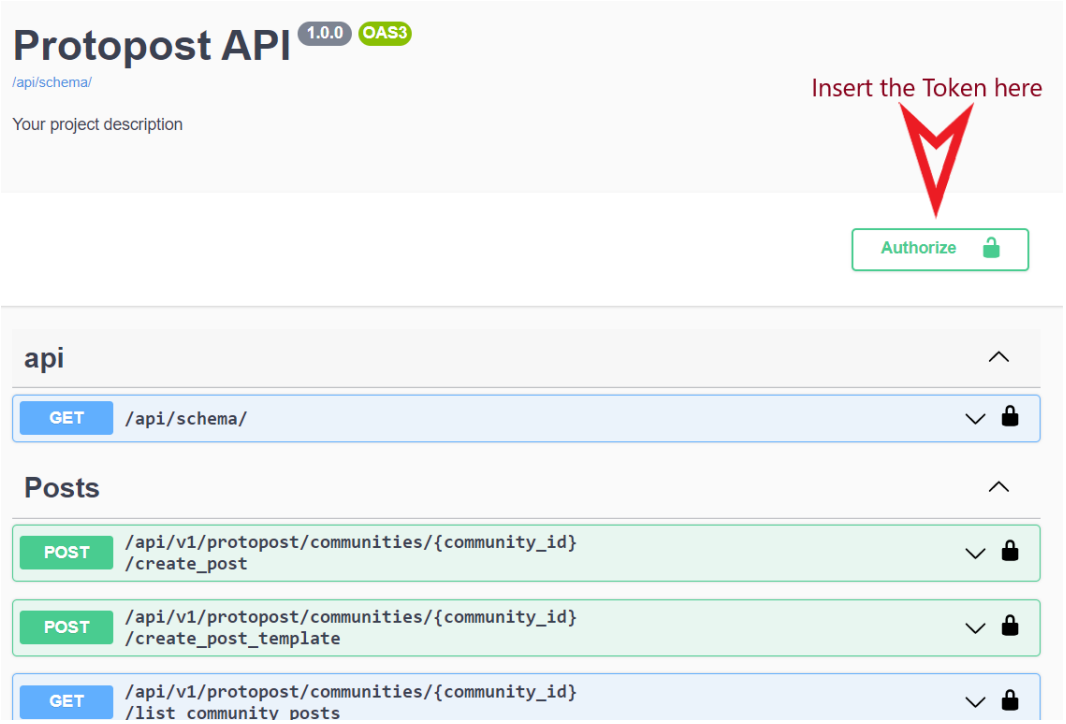
3. Use Case Diagram



API Endpoints

Our Backend API and its documentation can be accessed here. We are using a dynamic API documentation page so we included that page as the link.

You can use this page to experiment with our API. For authentication, you can add "Token c6ed16d517824d612985c8432d1d5a1d9a8d3603" to the authentication field in the swagger page.



If token does not work, you can use login function with JSON body:

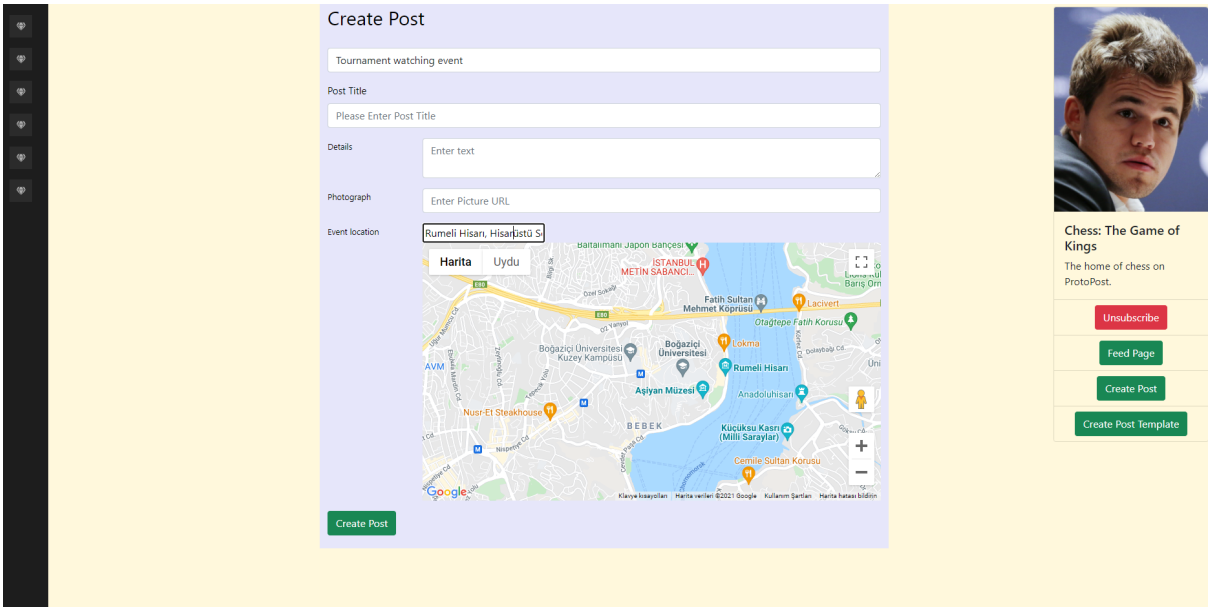
{"username": "tester","password": "tester123"}

After the execution, retrieve the token from the response and insert it into the authentication field as "Token <token-key>".

We also provide a Postman collection as a guide that includes CreateCommunity, CreatePostTemplate and CreatePost API calls which are the core functionalities of our API.

Implementation of Core Functionalities

We tried to implement a location feature using Google Maps API. As you can see in the picture we successfully added the map to the create post page if the field type is location. There is a search bar to find a place and at the bottom there is a map. When you search a place on the search bar map finds the most relevant place and focuses the map to that place's coordinates. But we encountered some problems while sending data to the backend so we didn't include this feature on Customer Demo 2. It will be added on Customer Demo 3. We will add this map to the post card component. We will add markers on maps to select a place from the map.



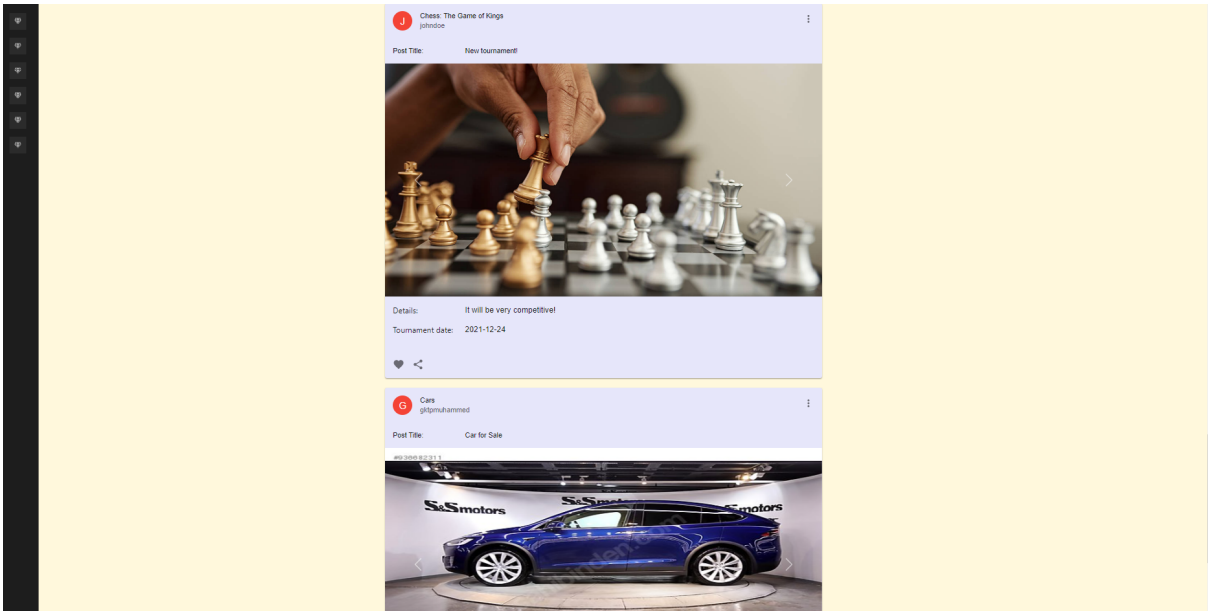
As the Backend team, we had implemented the DataField object to hold JSON as content. Therefore, retrieving and returning location data was a trivial task in Backend. However, since we will be providing advanced post filtering features in the final milestone, we are planning to write some backend processing features (e.g. distance calculation) in the future.

User Interface/ User Experience

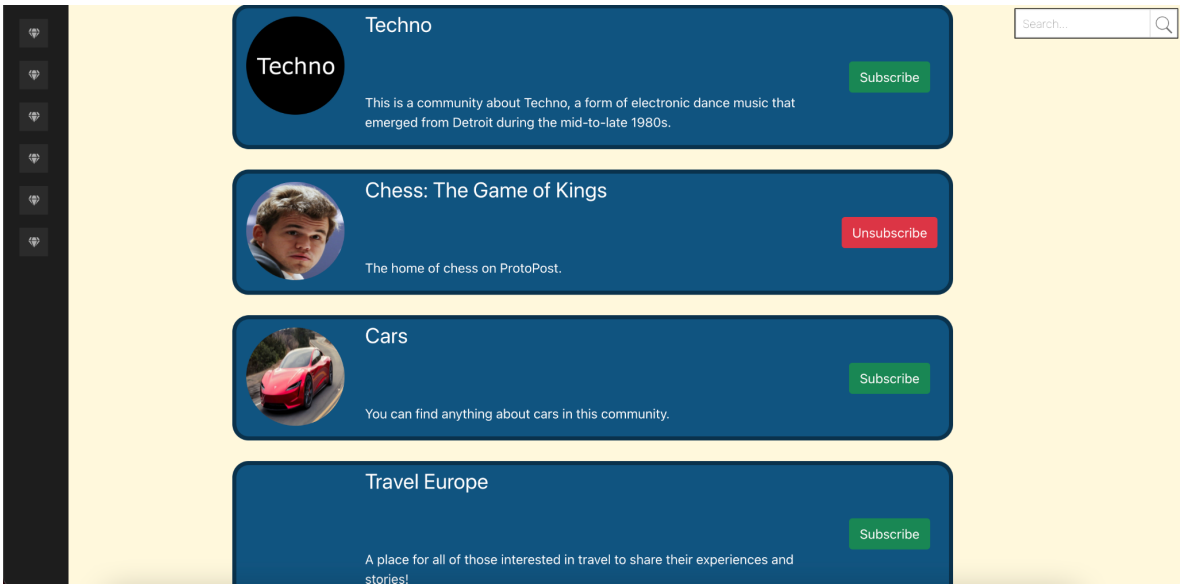
Frontend:

Link to project repository in github:
https://github.com/bounswe/2021SpringGroup1/tree/master/Web

Feed Page:

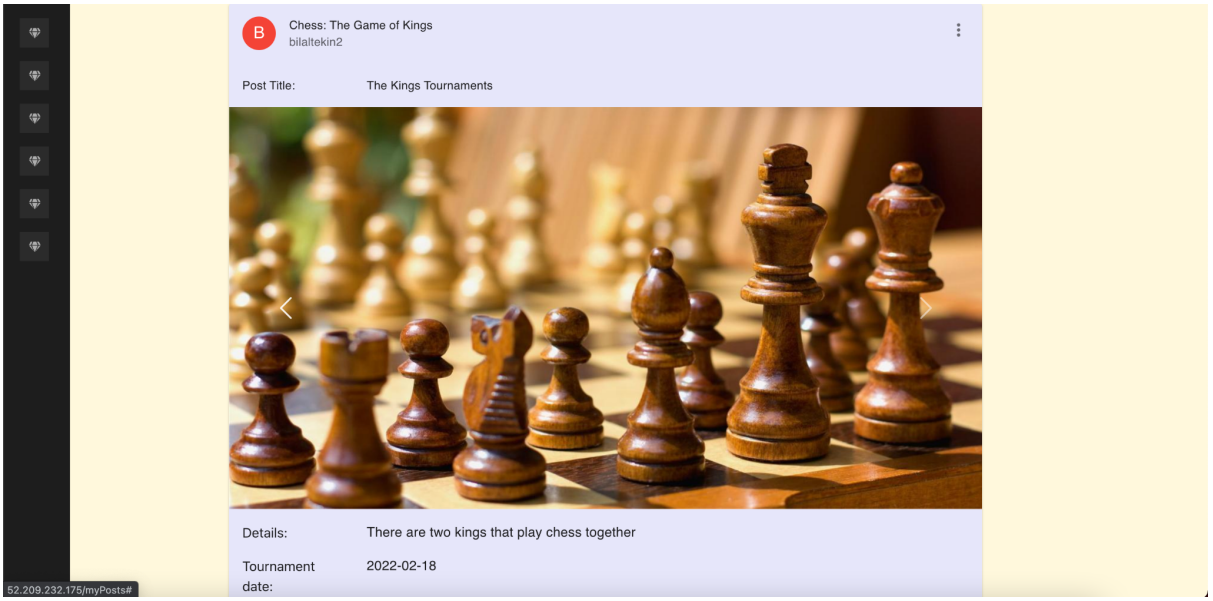


Communities Screen:

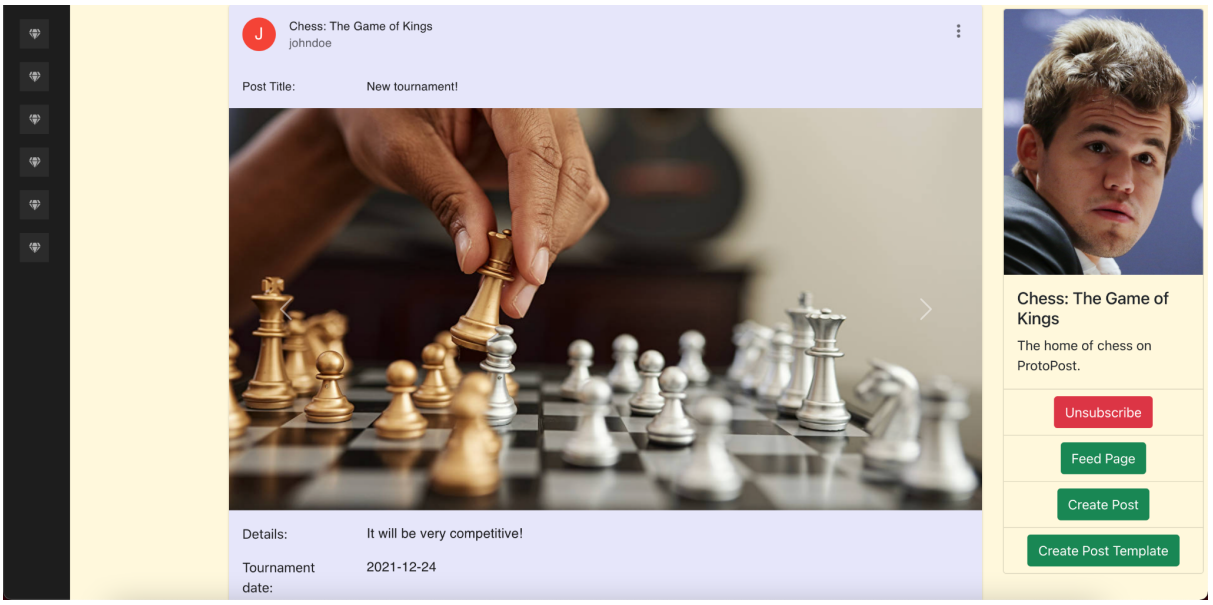


My Communities Screen:

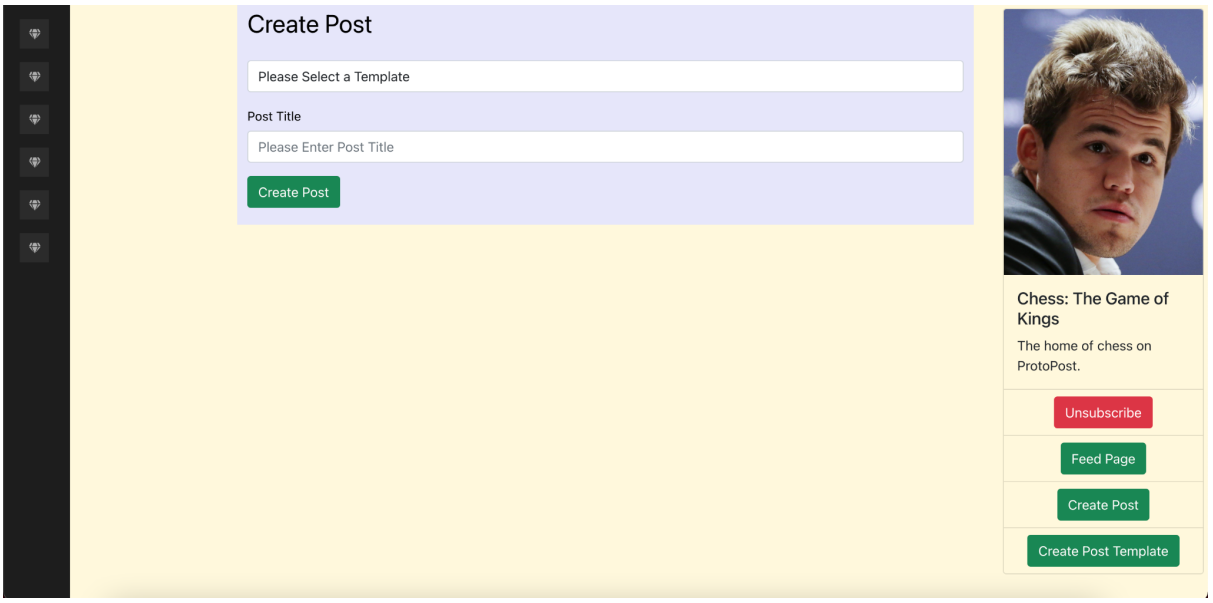
My Posts Screen:



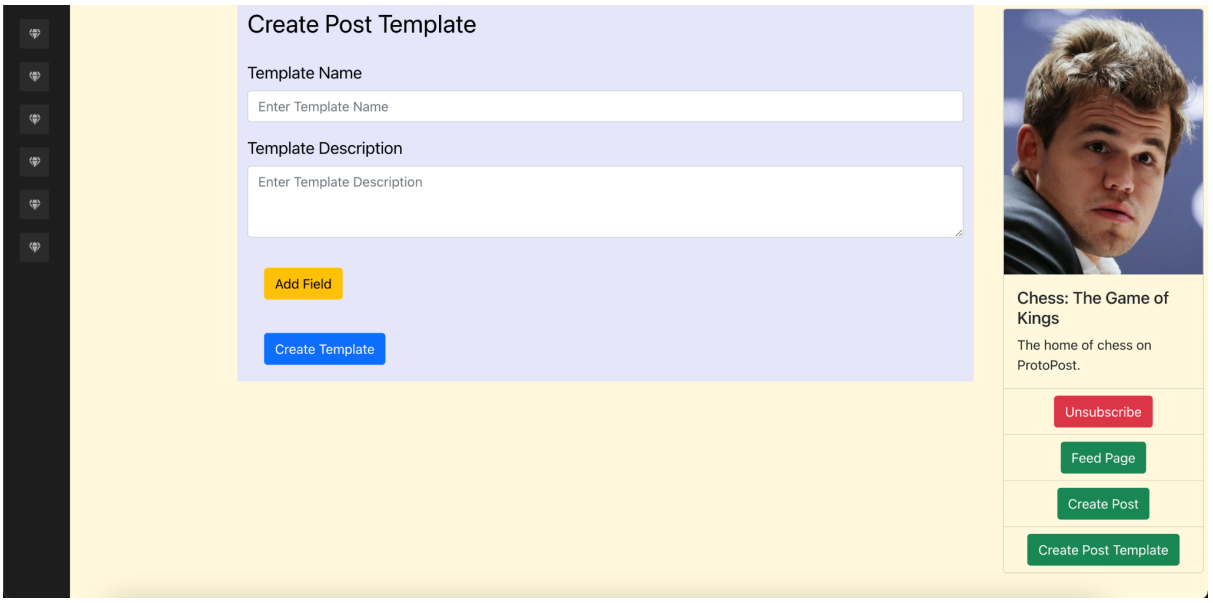
Community Screen:



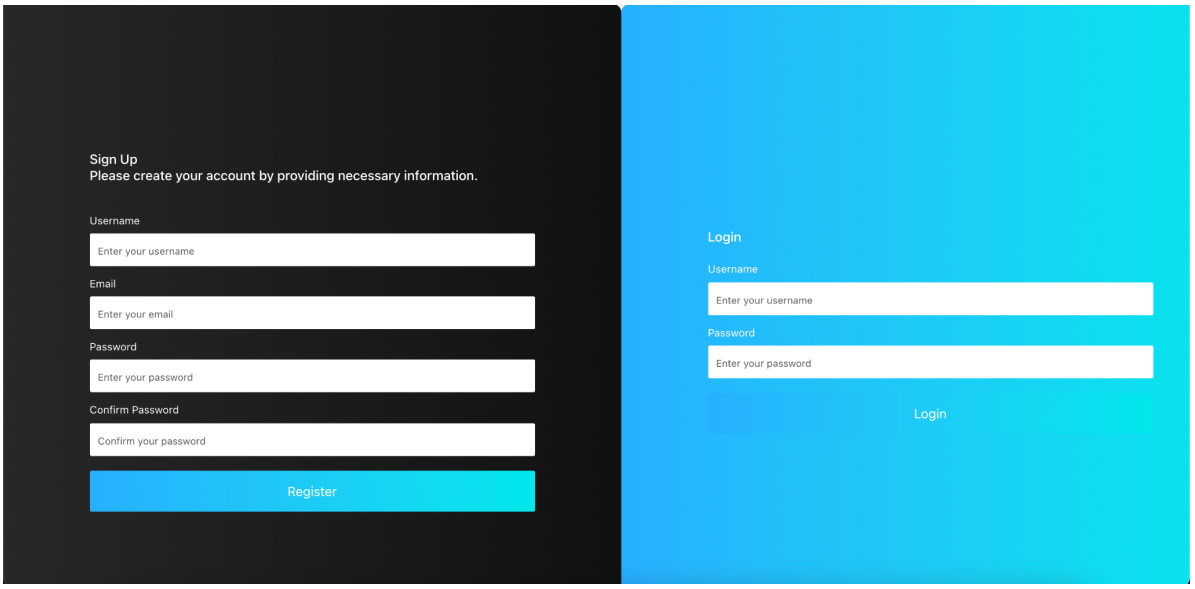
Create Post Screen:



Create Post Template Screen:



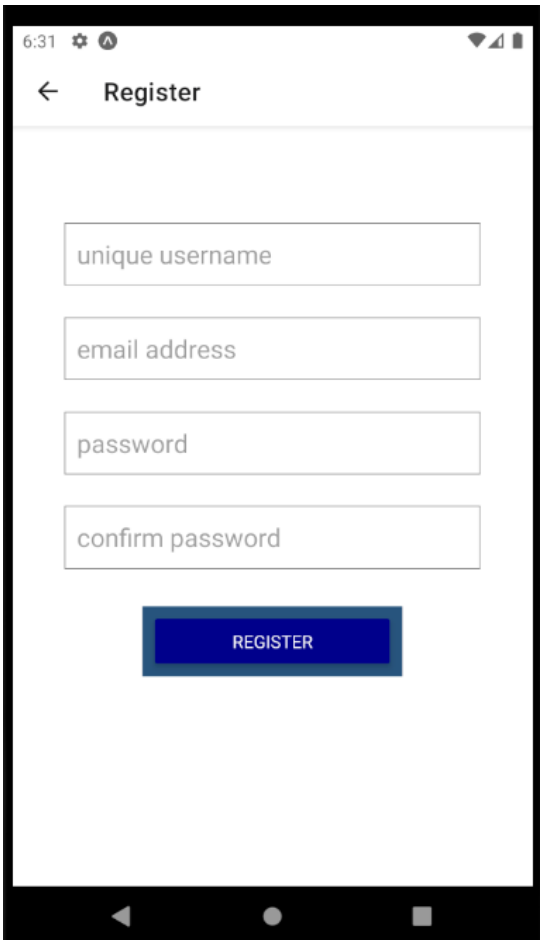
Login/Registration Screen:

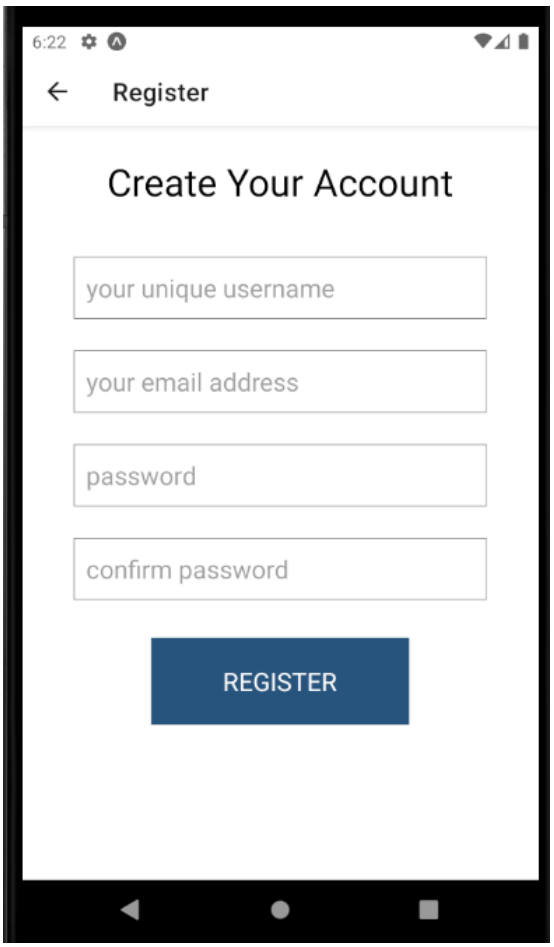


ANDROID: We were working on style changes but couldn't finish them before the customer demo. So I will put both versions here (lefft old, right new).

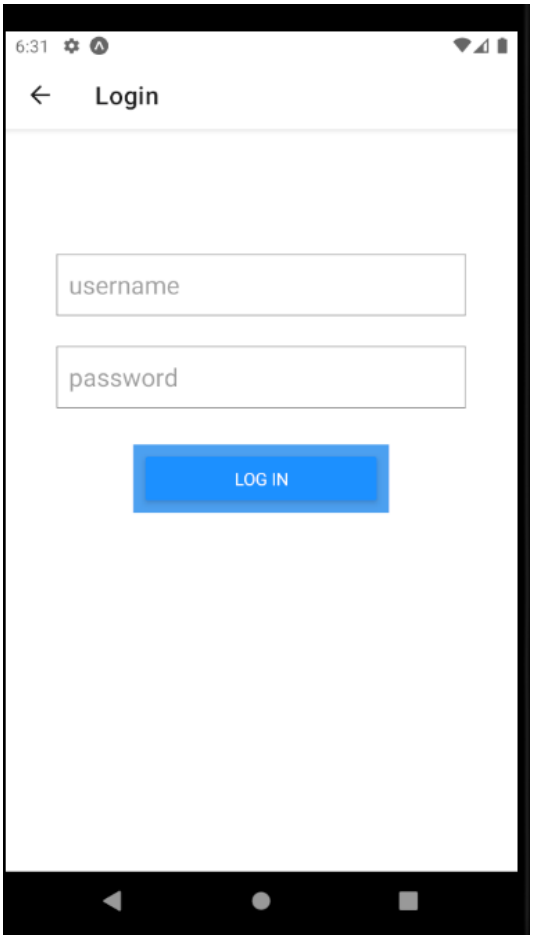
Welcome Screen:

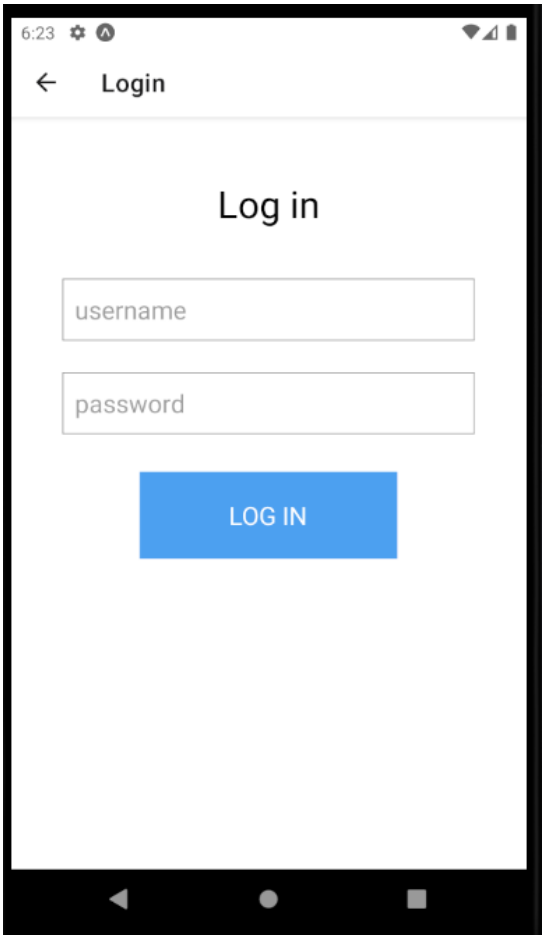
Register Screen:



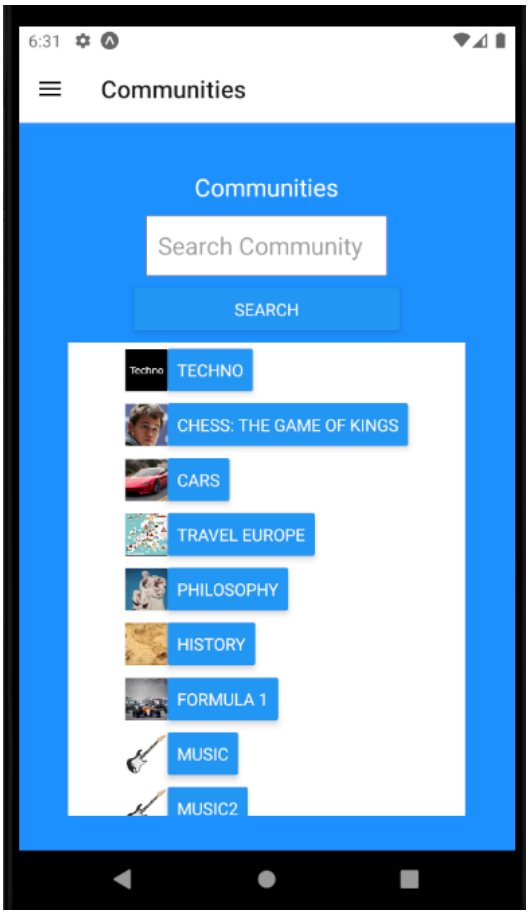


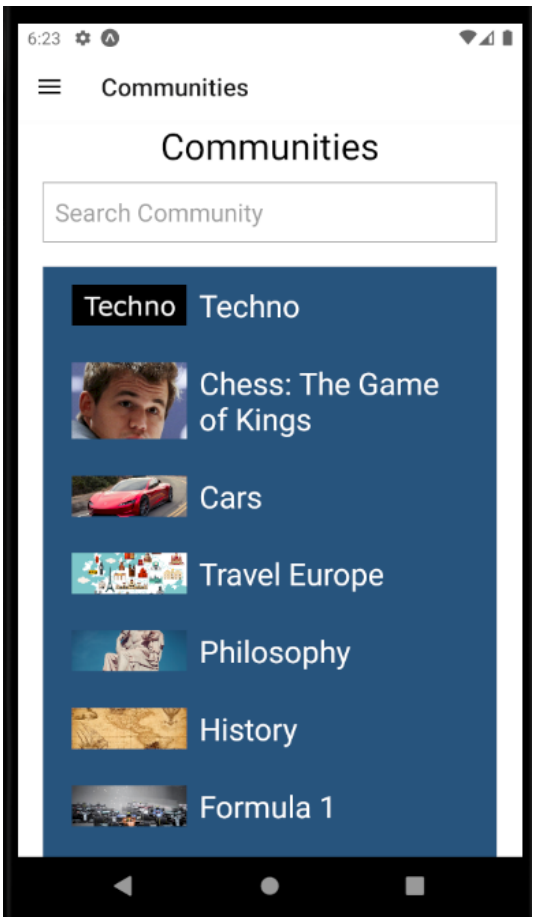
Login Screen:



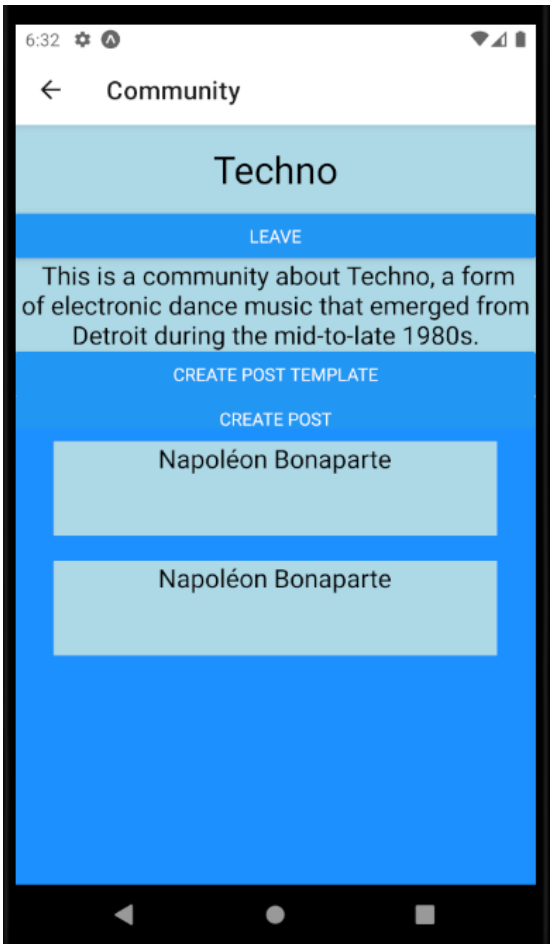


Communities Screen:

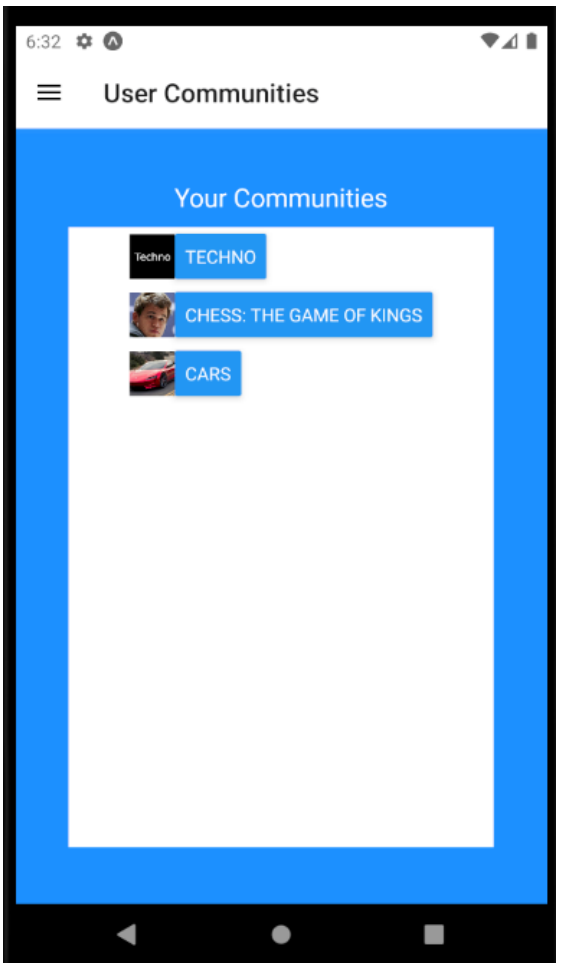


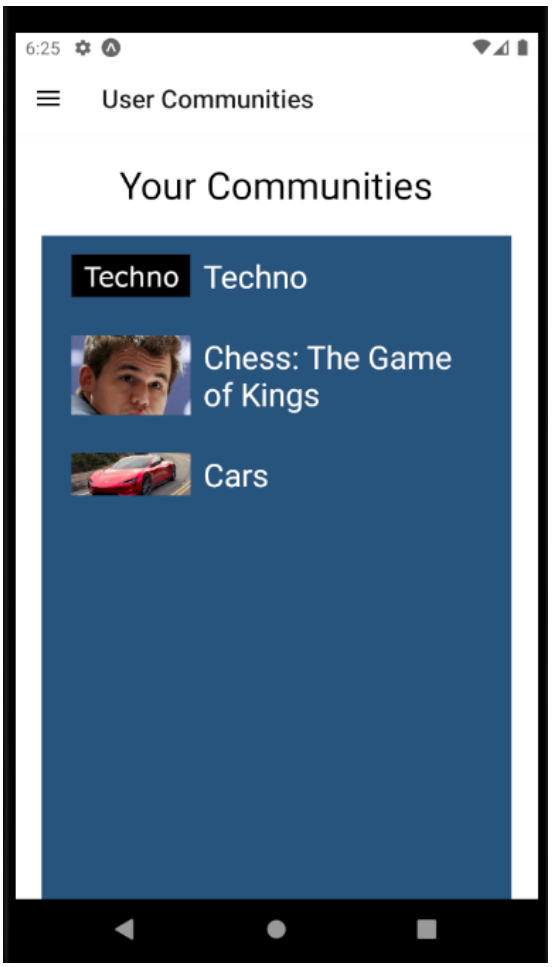


Community Screen:

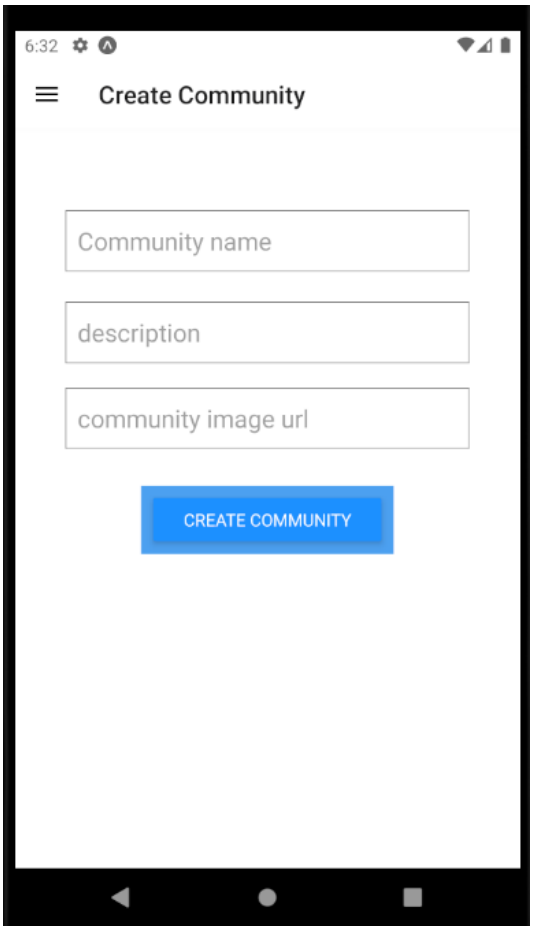


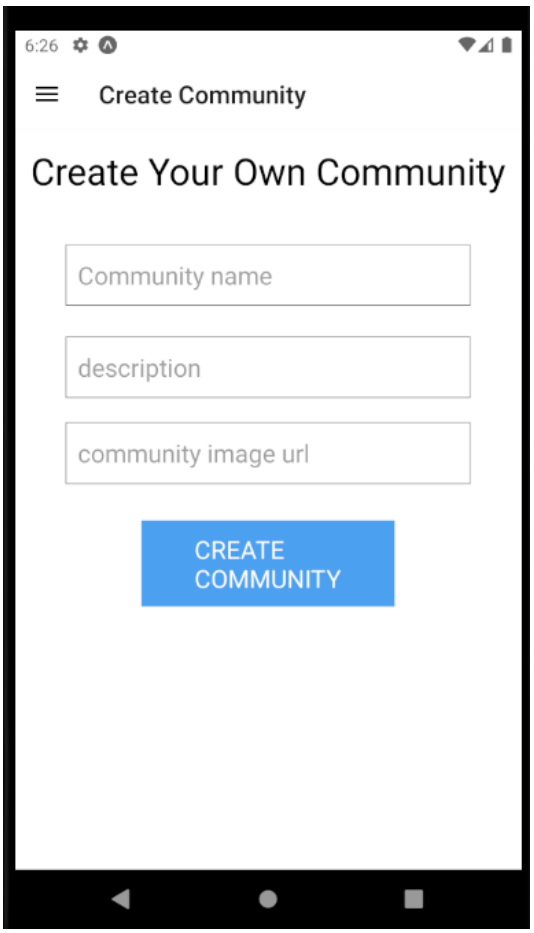
Users Communities Screen:





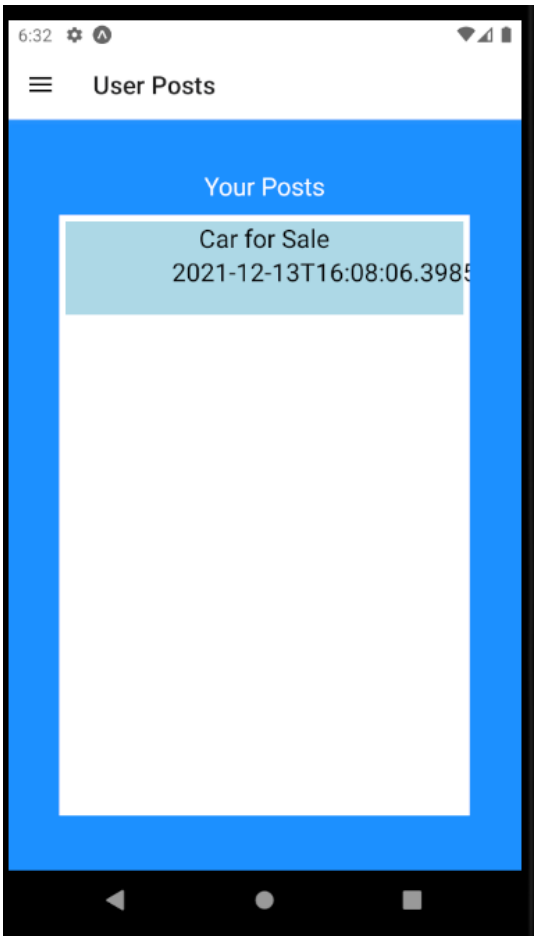
Create Community Screen:

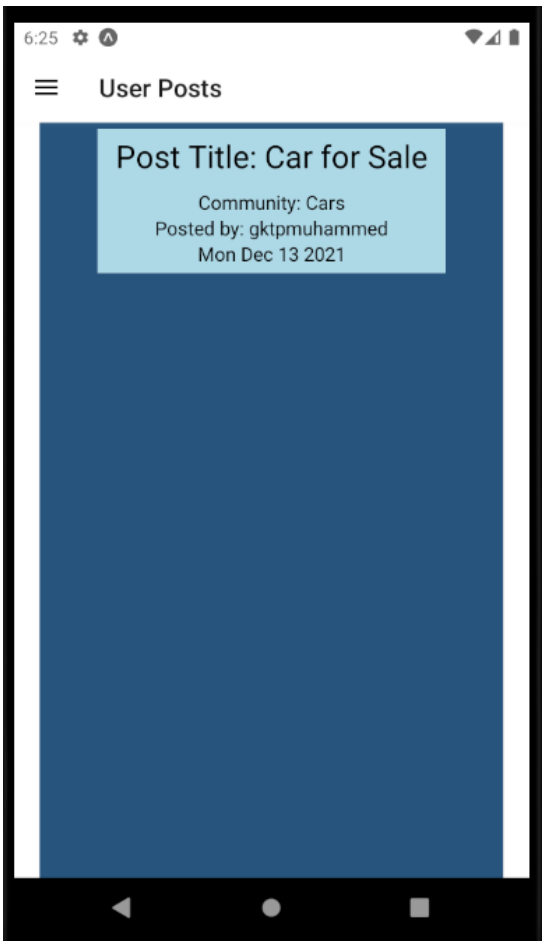

6:32
Create Community
Community name
description
community image url
CREATE COMMUNITY


6:26
Create Community
Create Your Own Community
Community name
description
community image url
CREATE COMMUNITY

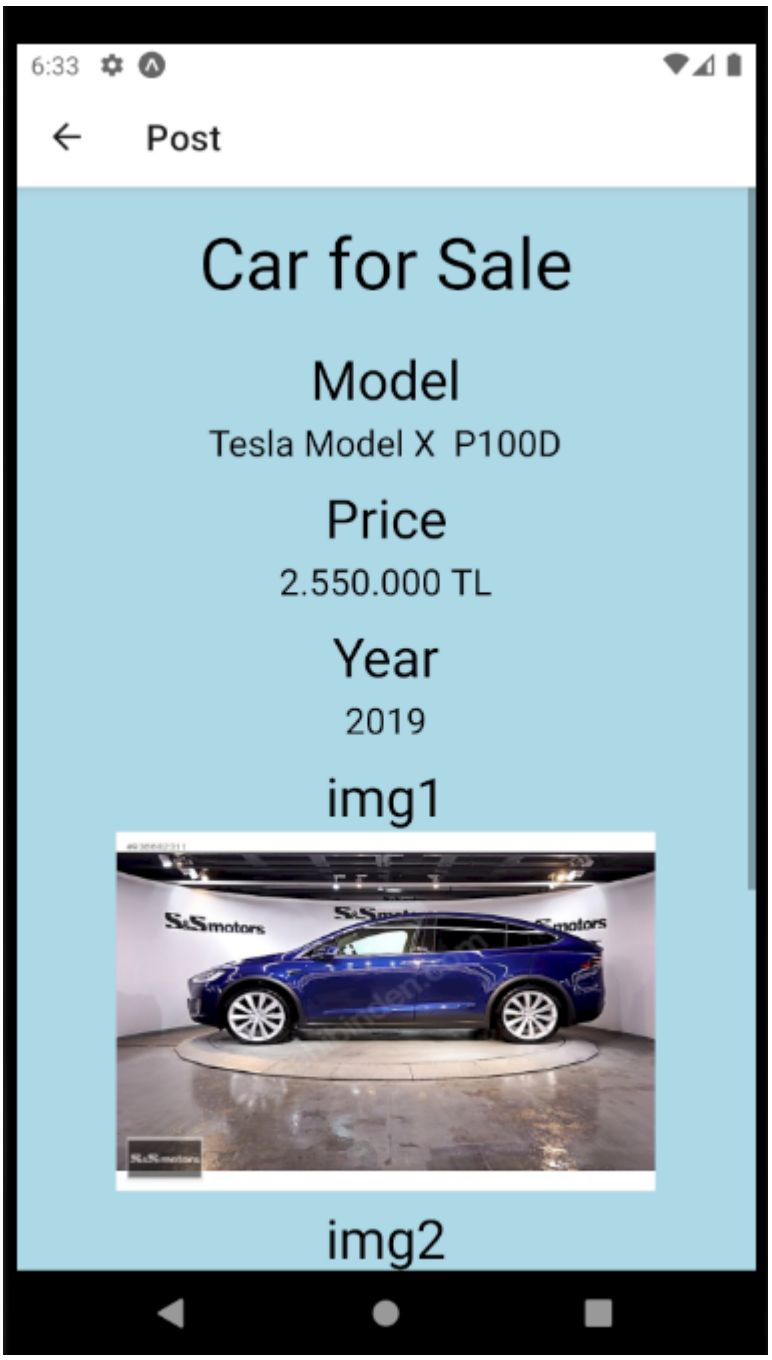
Feed Screen:

Users Posts Screen:

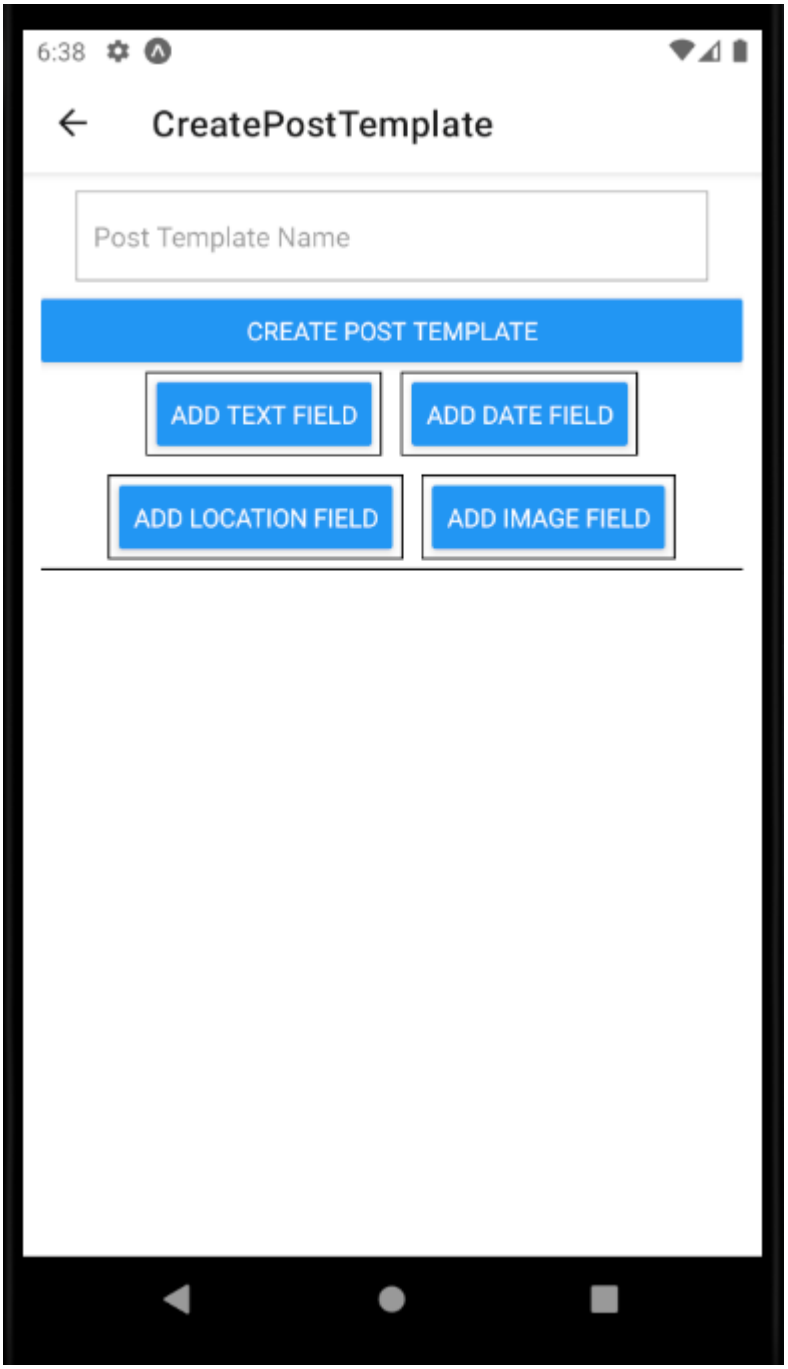




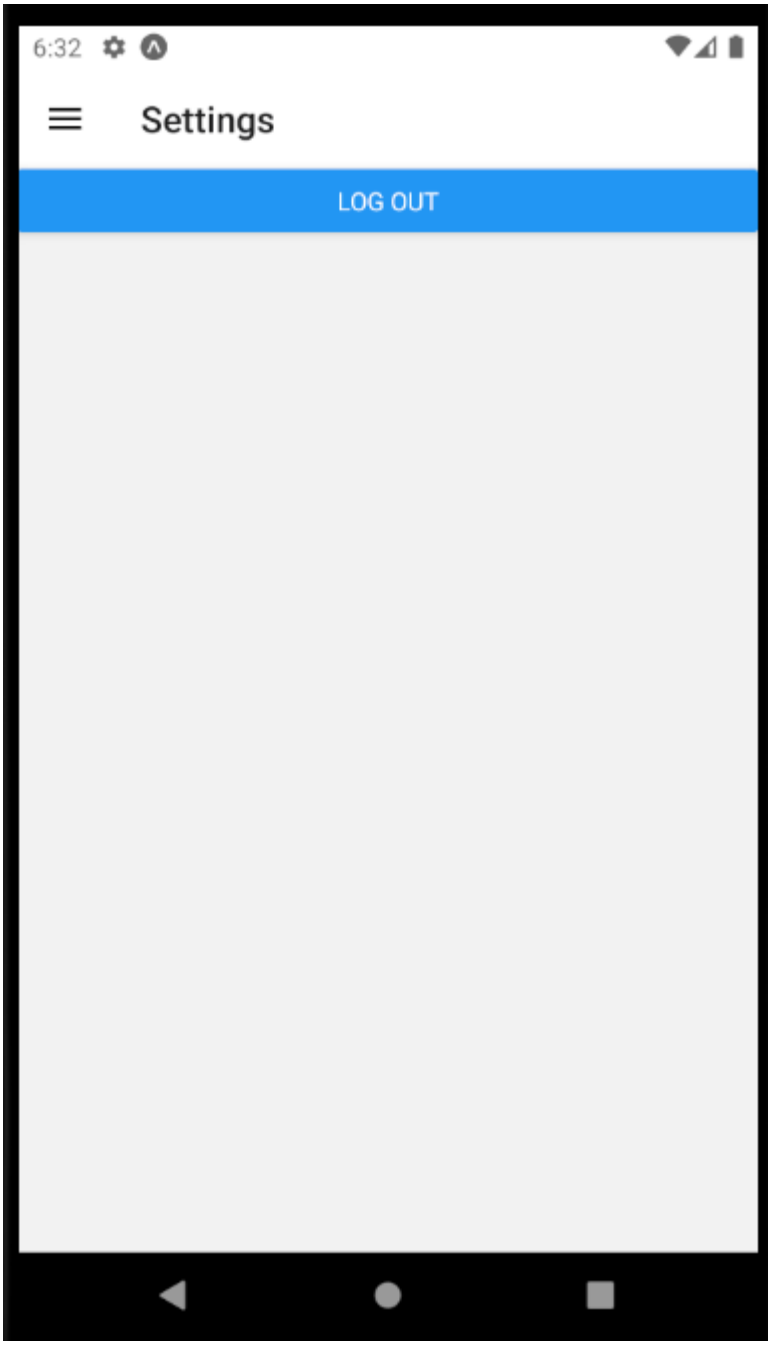
Post Screen: This page is same in both versions.



Create Post Template Screen: This page is same in both versions.



Setting Screen: This page is same in both versions.



Standards

We made the necessary additions to the functions for the JsonLD format in the Backend-Emrah-JsonLD branch, but we did not add a merge request to the Backend branch in case of problems before the presentation.

The only thing we need is creating a file that describes aliases.

e.Project Plan and RAM

Green -> Lead
Blue -> Secondary
Yellow -> Contributor
Magenta -> Approval
Orange -> Review
Red -> No contribution

	Orkan Akisu	Elif AKALIN	Emrah DOĞAN	Berke ARGIN	Muhammed GÖKTEPE	Emir Hifsı ÖZTOPRAK	Bilal TEKİN	Yunus Emre TOPAL
Initialization								
Reviewing the group repository								
Refreshing knowledge on Github platform								
Updating wiki page for the current semester								
Requirement & Design Documents								
Reviewing UML diagrams								
Reviewing requirements document from last semester								
Reviewing mockups								
Updating requirements document								
Updating Sequence diagrams								
Updating Customer Questions								
Backend								
Studying Django Framework to catch up								
Participating in Backend Meeting 1								

Learning Activity Streams								
Learning JSON-LD								
Sketching the initial outline for Backend API								
Initializing the Django project for the backend app								
Migrating the model from the practice app to the new Django project.								
Implementing initial registration functions								
Implementing initial post template functions.								
Implementing initial posting functions								
Implementing initial community view functions								
Implementing initial post feed functions.								
Writing unit tests for the Backend API								
Learning Django Rest framework								
Participating in Backend Meeting 2								
Writing serializer classes for the REST API framework								
Adapting and reimplementing existing functions with respect to Django REST API framework.								
Participating in Backend Meeting 3								
Adding Swagger auto-documentation functionality to the code.								
Configuring endpoints for JSON-LD standards								
Learning Amazon AWS								
Making database migration from SQLite to MySQL								
Deploying backend services to AWS instance								
Working on fixing CSRF Token errors								
Working on fixing CORS errors								
Dockerizing the backend application								
Configuring secrets on backend codebase								
Milestone 2								
Fixing the CORS and CSRF error								

Fixing the authorization problem								
Implementing token-based authorization scheme								
Configuring dockerfiles for container based deployment								
Implementing GetHomeFeed method								
Implementing SearchCommunities method								
Implementing SearchPostInCommunity method								
Reconfiguring GetCommunityData endpoint								
Reconfiguring Community serializer class								
Doing preliminary research for advanced filtering implementation								
Doing preliminary research for privacy features								
Researching APIs for geocoding								
Writing a simple bash script for a deployment automation								
Debugging existing methods from Milestone 1								
Enhancing endpoints based on frontend requests								
Writing unit tests for functions								
Refining Swagger documentation								
Web Frontend								
Studying ReactJS-Bootstrap								
Participating in Frontend Meeting 1								
Learning React Hooks								
Learning React Axios								
Learning React Navigation								
Learning React Redux-thunk								
Participating in Frontend Meeting 2								
Creating React App								
Creating Register Page								
Creating Login Page								

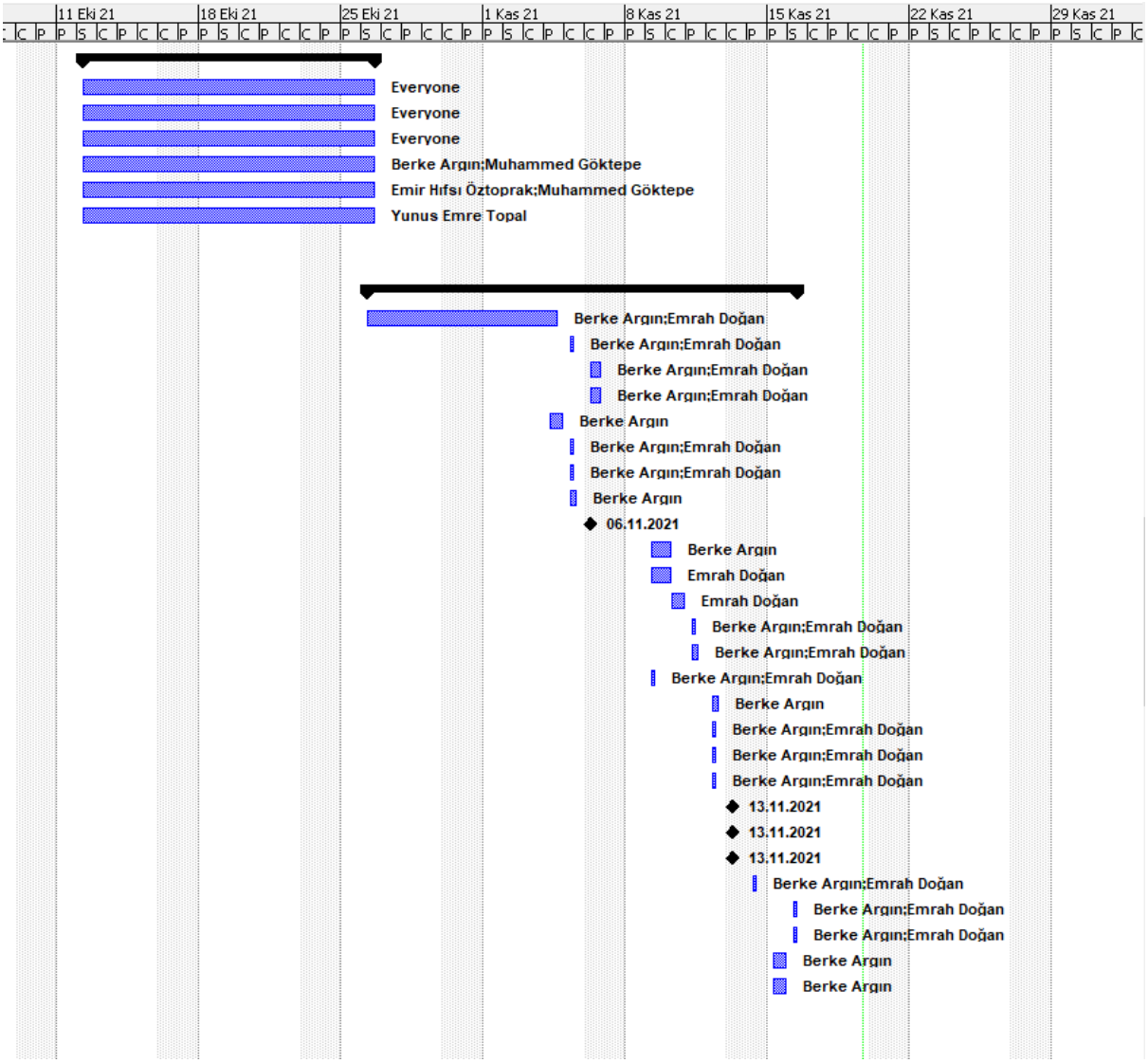
Creating Create Post Template Page								
Creating Create Post Page								
Creating My Posts Page								
Creating Communities Page								
Creating Feed Page								
Creating Create Community Page								
Creating My Communities Page								
Creating Sidebar								
Backend Integration								
Login Backend Integration								
Register Backend Integration								
Communities Backend Integration								
Deployment of Frontend to AWS								
Milestone 2								
Connecting Create Post Template Page to backend								
Updating Create Post Template Page design								
Connecting Create Post Page to backend								
Tried to implement google maps API but encountered some issues.								
Added project logo and project name to the site.								
Fixed image display on posts using Carousel								
Presenting Web Frontend on Customer Demo 2								
Merged Web branch to master branch for tagging.								
Updating Post component with Material UI								
Searching Docker to dockerize our applications								
Searching Launch template to automate deployment								
Searching AutoScale to increase EC2 machines in case of high traffic								
Deploying of Frontend Application								
Fixing the bug related to the Subscribe button								

Implementing Search Community Feature								
Adding alert component to Create Community Page								
Changing the structure of Community Side Card buttons								
Adding alert component to Create Post Page								
Adding alert component to Create Post Template Page								
Mobile Frontend								
Learning React Native								
Creating Initial App								
Creating Welcome Screen								
Creating Login Screen								
Implementing API Call for Login Feature								
Creating Registration Screen								
Implementing API call for Registration feature								
Creating Feed Screen								
Implementing API call for getting Feed data								
Creating Communities Screen								
Implementing API call for getting Communities								
Creating User Posts Screen								
Implementing API call for getting User Posts								
Creating User Communities Screen								
Implementing API call for getting User Communities								
Creating Settings Screen								
Creating Post Screen								

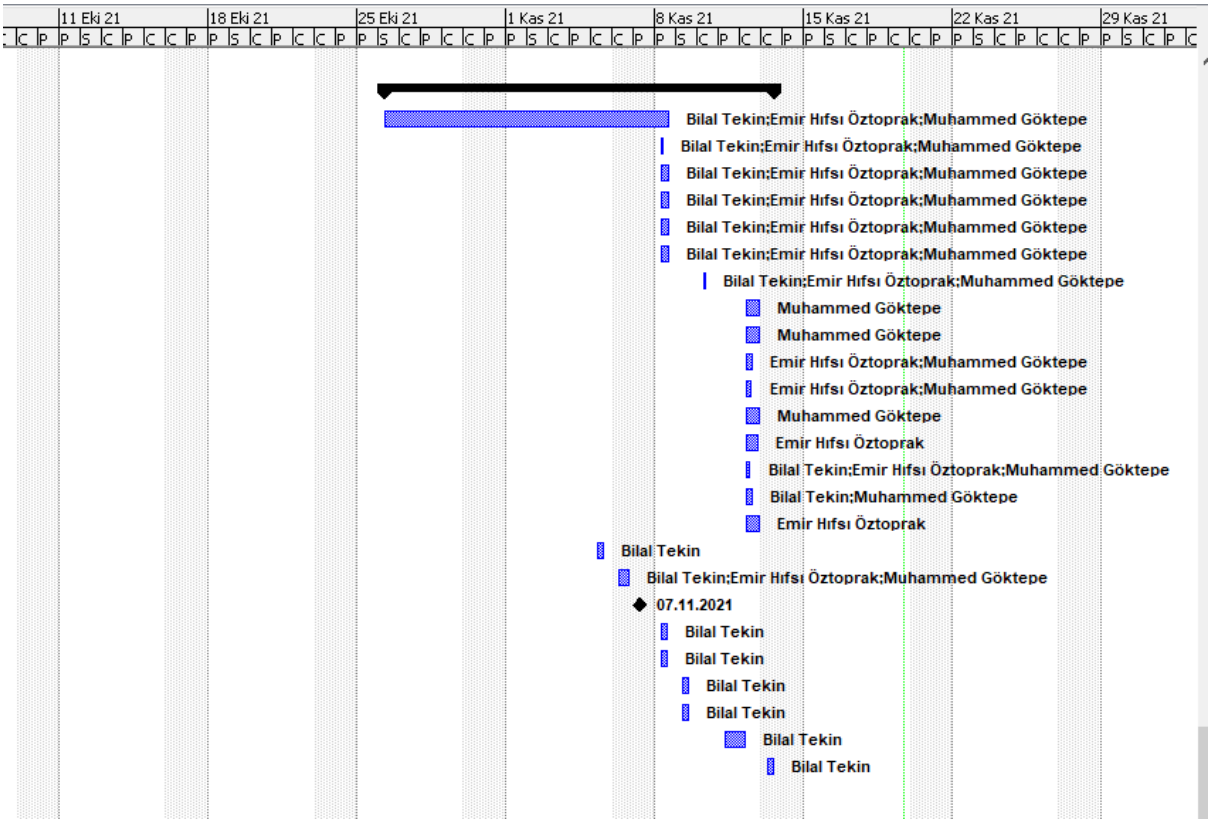
Creating Community Screen								
Creating Drawer Navigation								
Creating Axios Service								
Creating Community Creation Screen								
Milestone 2								
Implementing Community Creation Feature								
Creating Post Template Creation Screen								
Implementing Post Template Creation Feature								
Converting fetch api calls to axios api calls								
Implementing Search Community Feature								
Displaying text and image fields in Post Screen								
Fixing Subscribe to a Community Feature								
Tokenizing axios api calls								

7.RoadMap

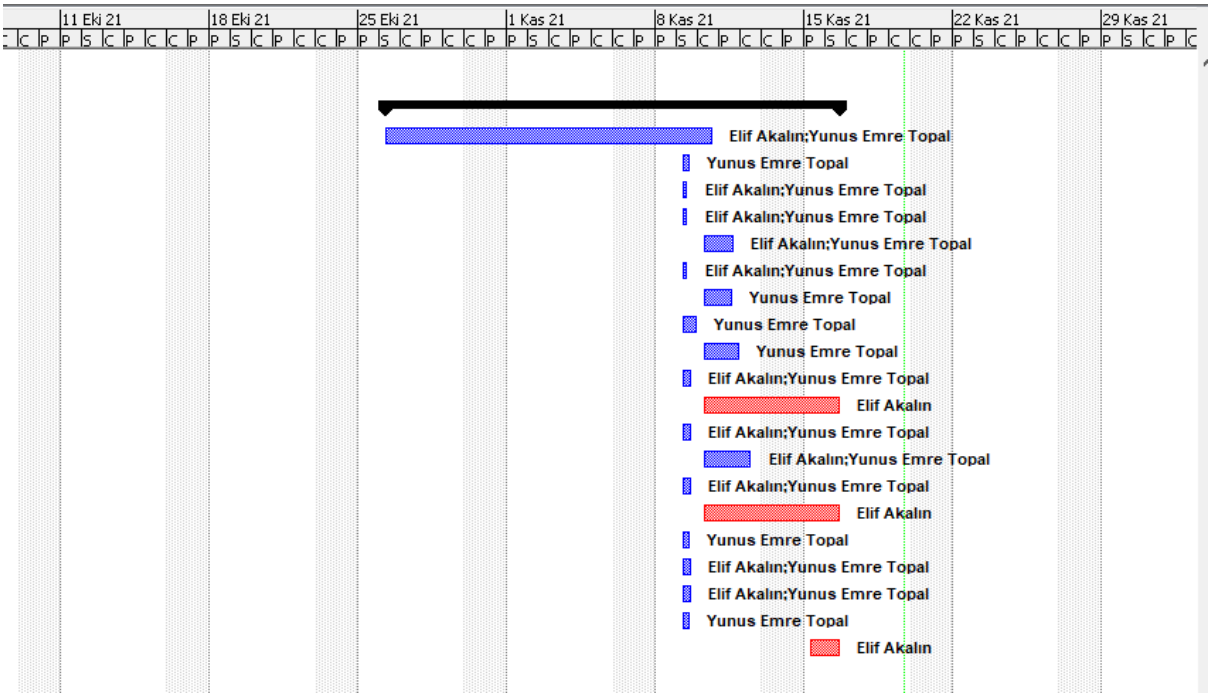
		Ad	Süre	Başlat	Bitirme	Kaynak Adları
92		**Review Project**	3,667 günl...	**12.10.2021 08:00**	**26.10.2021 17:00**	
93		Review Requirements	14,375 günler?	12.10.2021 08:00	26.10.2021 17:00	Everyone
94		Review Diagrams	14,375 günler?	12.10.2021 08:00	26.10.2021 17:00	Everyone
95		Review Mockups	14,375 günler?	12.10.2021 08:00	26.10.2021 17:00	Everyone
96		Update Requiremets	3,667 günler?	12.10.2021 08:00	26.10.2021 17:00	Berke Argın;Muhammed Göktepe
97		Update Sequence Diagrams	3,667 günler?	12.10.2021 08:00	26.10.2021 17:00	Emir Hıfsı Öztoprak;Muhammed Göktepe
98		Update Questions	14,375 günler?	12.10.2021 08:00	26.10.2021 17:00	Yunus Emre Topal
99		**Backend**	5,167 günl...	**26.10.2021 08:00**	**16.11.2021 12:00**	
100		Studying Django Framework to catch up	2,667 günler?	26.10.2021 08:00	04.11.2021 17:00	Berke Argın;Emrah Doğan
101		Participating in Backend Meeting 1	0,167 günler?	05.11.2021 08:00	05.11.2021 12:00	Berke Argın;Emrah Doğan
102		Learning Activity Streams	0 günler?	06.11.2021 08:00	06.11.2021 20:00	Berke Argın;Emrah Doğan
103		Learning JsonLD	0 günler?	06.11.2021 08:00	06.11.2021 20:00	Berke Argın;Emrah Doğan
104		Sketching the initial outline for Backend API	0,667 günler?	04.11.2021 08:00	05.11.2021 00:00	Berke Argın
105		Initializing the Django project for the backen	0,167 günler?	05.11.2021 08:00	05.11.2021 12:00	Berke Argın;Emrah Doğan
106		Migrating the model from the practice app to	0,167 günler?	05.11.2021 08:00	05.11.2021 12:00	Berke Argın;Emrah Doğan
107		Implementing initial registration functions	0,333 günler?	05.11.2021 08:00	05.11.2021 16:00	Berke Argın
108		Implementing initial post template functions.	0 günler?	06.11.2021 08:00	06.11.2021 08:00	Berke Argın
109		Implementing initial posting functions	1 gün?	09.11.2021 08:00	10.11.2021 08:00	Berke Argın
110		Implementing initial community view function	1 gün?	09.11.2021 08:00	10.11.2021 08:00	Emrah Doğan
111		Implementing initial post feed functions.	0,667 günler?	10.11.2021 08:00	11.11.2021 00:00	Emrah Doğan
112		Writing unit tests for the Backend API	0,167 günler?	11.11.2021 08:00	11.11.2021 12:00	Berke Argın;Emrah Doğan
113		Learning Django Rest framework	0,292 günler?	11.11.2021 08:00	11.11.2021 16:00	Berke Argın;Emrah Doğan
114		Participating in Backend Meeting 2	0,167 günler?	09.11.2021 08:00	09.11.2021 12:00	Berke Argın;Emrah Doğan
115		Writing serializer classes for the REST API fra	0,333 günler?	12.11.2021 08:00	12.11.2021 16:00	Berke Argın
116		Adapting existing functions to Django REST A	0,167 günler?	12.11.2021 08:00	12.11.2021 12:00	Berke Argın;Emrah Doğan
117		Participating in Backend Meeting 3	0,167 günler?	12.11.2021 08:00	12.11.2021 12:00	Berke Argın;Emrah Doğan
118		Adding Swagger auto-documentation functio	0,167 günler?	12.11.2021 08:00	12.11.2021 12:00	Berke Argın;Emrah Doğan
119		Configuring endpoints for JSON-LD standard	0 günler?	13.11.2021 08:00	13.11.2021 08:00	Emrah Doğan
120		Learning Amazon AWS	0 günler?	13.11.2021 08:00	13.11.2021 08:00	Berke Argın;Emrah Doğan
121		Making database migration from SQLite to My	0 günler?	13.11.2021 08:00	13.11.2021 08:00	Berke Argın;Emrah Doğan
122		Deploying backend services to AWS instance	0 günler?	14.11.2021 08:00	14.11.2021 12:00	Berke Argın;Emrah Doğan
123		Working on fixing CSRF Token errors	0,167 günler?	16.11.2021 08:00	16.11.2021 12:00	Berke Argın;Emrah Doğan
124		Working on fixing CORS errors	0,167 günler?	16.11.2021 08:00	16.11.2021 12:00	Berke Argın;Emrah Doğan
125		Dockerizing the backend application	0,667 günler?	15.11.2021 08:00	16.11.2021 00:00	Berke Argın
126		Adding secrets to the backend codebase	0,667 günler?	15.11.2021 08:00	16.11.2021 00:00	Berke Argın



		Ad	Süre	Başlat	Bitirme	Kaynak Adları
127		**Frontend**	4,667 günl...	**26.10.2021 08:00**	**13.11.2021 16:00**	
128		Studying ReactJS-Bootstrap	3,333 günler?	26.10.2021 08:00	08.11.2021 17:00	Bilal Tekin;Emir Hıfsı Öztoprak;Muhammed Göktepe
129		Participating in Frontend Meeting 1	0,111 günler?	08.11.2021 08:00	08.11.2021 10:40	Bilal Tekin;Emir Hıfsı Öztoprak;Muhammed Göktepe
130		Learning React Hooks	0,333 günler?	08.11.2021 08:00	08.11.2021 17:00	Bilal Tekin;Emir Hıfsı Öztoprak;Muhammed Göktepe
131		Learning React Axios	0,333 günler?	08.11.2021 08:00	08.11.2021 17:00	Bilal Tekin;Emir Hıfsı Öztoprak;Muhammed Göktepe
132		Learning React Navigation	0,333 günler?	08.11.2021 08:00	08.11.2021 17:00	Bilal Tekin;Emir Hıfsı Öztoprak;Muhammed Göktepe
133		Learning React Redux-thunk	0,333 günler?	08.11.2021 08:00	08.11.2021 17:00	Bilal Tekin;Emir Hıfsı Öztoprak;Muhammed Göktepe
134		Participating in Frontend Meeting 2	0,111 günler?	10.11.2021 08:00	10.11.2021 10:40	Bilal Tekin;Emir Hıfsı Öztoprak;Muhammed Göktepe
135		Creating Register Page	0,667 günler?	12.11.2021 08:00	13.11.2021 00:00	Muhammed Göktepe
136		Creating Login Page	0,667 günler?	12.11.2021 08:00	13.11.2021 00:00	Muhammed Göktepe
137		Creating Create Post Template Page	0,283 günler?	12.11.2021 08:00	12.11.2021 15:47	Emir Hıfsı Öztoprak;Muhammed Göktepe
138		Creating Create Post Page	0,255 günler?	12.11.2021 08:00	12.11.2021 15:06	Emir Hıfsı Öztoprak;Muhammed Göktepe
139		Creating My Posts Page	0,667 günler?	12.11.2021 08:00	13.11.2021 00:00	Muhammed Göktepe
140		Creating Communities Page	0,593 günler?	12.11.2021 08:00	12.11.2021 22:13	Emir Hıfsı Öztoprak
141		Creating Feed Page	0,19 günler?	12.11.2021 08:00	12.11.2021 13:33	Bilal Tekin;Emir Hıfsı Öztoprak;Muhammed Göktepe
142		Creating Create Community Page	0,283 günler?	12.11.2021 08:00	12.11.2021 15:47	Bilal Tekin;Muhammed Göktepe
143		Creating My Communities Page	0,667 günler?	12.11.2021 08:00	13.11.2021 00:00	Emir Hıfsı Öztoprak
144		Initialize React App	0,333 günler?	05.11.2021 08:00	05.11.2021 16:00	Bilal Tekin
145		Creating example screens and components.	0 günler?	06.11.2021 08:00	06.11.2021 20:00	Bilal Tekin;Emir Hıfsı Öztoprak;Muhammed Göktepe
146		Creating Sidebar	0 günler?	07.11.2021 08:00	07.11.2021 08:00	Bilal Tekin
147		integration of Axios	0,333 günler?	08.11.2021 08:00	08.11.2021 16:00	Bilal Tekin
148		integration and creating examples of Redux	0,333 günler?	08.11.2021 08:00	08.11.2021 16:00	Bilal Tekin
149		Integration of React Navigation	0,333 günler?	09.11.2021 08:00	09.11.2021 16:00	Bilal Tekin
150		Implementation of React-bootstrap	0,333 günler?	09.11.2021 08:00	09.11.2021 16:00	Bilal Tekin
151		Backend integration	1 gün?	11.11.2021 08:00	12.11.2021 08:00	Bilal Tekin
152		AWS server creation and deployment	0,333 günler?	13.11.2021 08:00	13.11.2021 16:00	Bilal Tekin

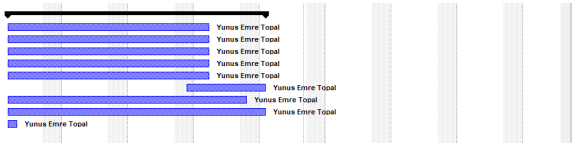


		Ad	Süre	Başlat	Bitirme	Kaynak Adları
153		**Mobile**	5,333 günl...	**26.10.2021 08:00**	**16.11.2021 17:00**	
154		Learning React Native	4 günler?	26.10.2021 08:00	10.11.2021 17:00	Elif Akalın;Yunus Emre Topal
155		Creating Initial App	0,333 günler?	09.11.2021 08:00	09.11.2021 16:00	Yunus Emre Topal
156		Creating Welcome Screen	0,167 günler?	09.11.2021 08:00	09.11.2021 13:00	Elif Akalın;Yunus Emre Topal
157		Creating Login Screen	0,167 günler?	09.11.2021 08:00	09.11.2021 13:00	Elif Akalın;Yunus Emre Topal
158		Implementing API call for Login feature	0,667 günler?	10.11.2021 08:00	11.11.2021 17:00	Elif Akalın;Yunus Emre Topal
159		Creating Registration Screen	0,167 günler?	09.11.2021 08:00	09.11.2021 13:00	Elif Akalın;Yunus Emre Topal
160		Implementing API call for Registration feature	1,333 günler?	10.11.2021 08:00	11.11.2021 16:00	Yunus Emre Topal
161		Creating Feed Screen	0,667 günler?	09.11.2021 08:00	10.11.2021 00:00	Yunus Emre Topal
162		Implementing API call for getting Feed data	1,667 günler?	10.11.2021 08:00	12.11.2021 00:00	Yunus Emre Topal
163		Creating Communities Screen	0,333 günler?	09.11.2021 08:00	09.11.2021 17:00	Elif Akalın;Yunus Emre Topal
164		Implementing API call for getting Communities	1,667 günler?	10.11.2021 08:00	16.11.2021 17:00	Elif Akalın
165		Creating User Posts Screen	0,333 günler?	09.11.2021 08:00	09.11.2021 17:00	Elif Akalın;Yunus Emre Topal
166		Implementing API call for getting User Posts	0,833 günler?	10.11.2021 08:00	12.11.2021 13:00	Elif Akalın;Yunus Emre Topal
167		Creating User Communities Screen	0,333 günler?	09.11.2021 08:00	09.11.2021 17:00	Elif Akalın;Yunus Emre Topal
168		Implementing API call for getting User Commur	1,667 günler?	10.11.2021 08:00	16.11.2021 17:00	Elif Akalın
169		Creating Settings Screen	0,333 günler?	09.11.2021 08:00	09.11.2021 16:00	Yunus Emre Topal
170		Creating Post Screen	0,333 günler?	09.11.2021 08:00	09.11.2021 17:00	Elif Akalın;Yunus Emre Topal
171		Creating Community Screen	0,333 günler?	09.11.2021 08:00	09.11.2021 17:00	Elif Akalın;Yunus Emre Topal
172		Creating Drawer Navigation	0,333 günler?	09.11.2021 08:00	09.11.2021 16:00	Yunus Emre Topal
173		Creating Axios Service	0,667 günler?	15.11.2021 08:00	16.11.2021 17:00	Elif Akalın



		Ad	Süre	Başlat	Bitirme	Kaynak Adları
174		**Backend**	6 günler?	**19.11.2021 08:00**	**15.12.2021 00:00**	
175		Fixing the CORS and CSRF error	0,667 günler?	25.11.2021 08:00	27.11.2021 17:00	Berke Argın;Emrah Doğan
176		Fixing the authorization problem	2,375 günler?	25.11.2021 08:00	27.11.2021 17:00	Berke Argın
177		Implementing token-based authorization schem	2,375 günler?	25.11.2021 08:00	27.11.2021 17:00	Berke Argın
178		Configuring dockerfiles for container based de	0,667 günler?	24.11.2021 08:00	25.11.2021 00:00	Berke Argın
179		Inplementing GetHomeFeed method	0,667 günler?	08.12.2021 08:00	09.12.2021 00:00	Emrah Doğan
180		Implementing SearchCommunities method	0,667 günler?	08.12.2021 08:00	09.12.2021 00:00	Emrah Doğan
181		Implementing SearchPostInCommunity method	0,667 günler?	14.12.2021 08:00	15.12.2021 00:00	Berke Argın
182		Reconfiguring GetCommunityData endpoint	0,667 günler?	09.12.2021 08:00	10.12.2021 00:00	Berke Argın
183		Reconfiguring Community serializer class	0,667 günler?	09.12.2021 08:00	10.12.2021 00:00	Berke Argın
184		Doing preliminary research for advanced filteri	2,667 günler?	03.12.2021 08:00	14.12.2021 17:00	Berke Argın;Emrah Doğan
185		Doing preliminary research for privacy feature	2,667 günler?	03.12.2021 08:00	14.12.2021 17:00	Berke Argın;Emrah Doğan
186		Researching APIs for geocoding	1 gün?	02.12.2021 08:00	06.12.2021 17:00	Berke Argın;Emrah Doğan
187		Writing a simple bash script for a deployment a	0,333 günler?	01.12.2021 08:00	01.12.2021 16:00	Berke Argın
188		Debugging existing methods from Milestone 1	5,667 günler?	19.11.2021 08:00	13.12.2021 17:00	Berke Argın;Emrah Doğan
189		Enhancing endpoints based on frontend reque	4 günler?	26.11.2021 08:00	13.12.2021 17:00	Berke Argın;Emrah Doğan
190		Writing unit tests for functions	3,333 günler?	26.11.2021 08:00	09.12.2021 17:00	Berke Argın;Emrah Doğan
191		**Frontend**	3,333 günl...	**01.12.2021 08:00**	**14.12.2021 17:00**	
192		Connecting Create Post Template Page to back	4,375 günler?	09.12.2021 08:00	13.12.2021 17:00	Muhammed Göktepe
193		Updating Create Post Template Page design	4,375 günler?	09.12.2021 08:00	13.12.2021 17:00	Muhammed Göktepe
194		Connecting Create Post Page to backend	4,375 günler?	09.12.2021 08:00	13.12.2021 17:00	Emir Hifsı Öztoprak
195		Tried to implement google maps API but encou	5,375 günler?	09.12.2021 08:00	14.12.2021 17:00	Muhammed Göktepe
196		Added project logo and project name to the sit	0,333 günler?	13.12.2021 08:00	13.12.2021 16:00	Muhammed Göktepe
197		Fixed image display on posts using Carousel	3,375 günler?	10.12.2021 08:00	13.12.2021 17:00	Muhammed Göktepe
198		Presenting Web Frontend on Customer Demo	0,333 günler?	14.12.2021 08:00	14.12.2021 16:00	Muhammed Göktepe
199		Merged Web branch to master branch for tagg	0,333 günler?	14.12.2021 08:00	14.12.2021 16:00	Muhammed Göktepe
200		Updating Post component with Material UI	4,375 günler?	09.12.2021 08:00	13.12.2021 17:00	Muhammed Göktepe
201		Searching Docker to dockerize our applications	2,375 günler?	01.12.2021 08:00	03.12.2021 17:00	Bilal Tekin
202		Deploying Frontend part to EC2	0,333 günler?	14.12.2021 08:00	14.12.2021 16:00	Bilal Tekin
203		Searching Launch template on AWS	3,375 günler?	04.12.2021 08:00	07.12.2021 17:00	Bilal Tekin
204		Searching AutoScale on AWS	3,375 günler?	07.12.2021 08:00	10.12.2021 17:00	Bilal Tekin
205		Trying to integrate create post with backend	1,375 günler?	10.12.2021 08:00	11.12.2021 17:00	Bilal Tekin
206		Trying to integrate create post template with b	1,375 günler?	11.12.2021 08:00	12.12.2021 17:00	Bilal Tekin
207		Integrate listing page with backend	2,375 günler?	12.12.2021 08:00	14.12.2021 17:00	Bilal Tekin
208		Fixing the bug related to the Subscribe button	5,375 günler?	08.12.2021 08:00	13.12.2021 17:00	Emir Hifsı Öztoprak
209		Implementing Search Community Feature	4,375 günler?	09.12.2021 08:00	13.12.2021 17:00	Emir Hifsı Öztoprak
210		Adding alert component to Create Community	4,375 günler?	09.12.2021 08:00	13.12.2021 17:00	Emir Hifsı Öztoprak
211		Changing the structure of Community Side Car	2,375 günler?	11.12.2021 08:00	13.12.2021 17:00	Emir Hifsı Öztoprak
212		Adding alert component to Create Post Page	3,375 günler?	10.12.2021 08:00	13.12.2021 17:00	Emir Hifsı Öztoprak
213		Adding alert component to Create Post Templ	3,375 günler?	10.12.2021 08:00	13.12.2021 17:00	Emir Hifsı Öztoprak

214		**Mobile**	6,667 günl...	**16.11.2021 08:00**	**13.12.2021 17:00**	
215		Creating Community Creation Screen	21,375 günler?	16.11.2021 08:00	07.12.2021 17:00	Yunus Emre Topal
216		Implementing Community Creation Feature	21,375 günler?	16.11.2021 08:00	07.12.2021 17:00	Yunus Emre Topal
217		Creating Post Template Creation Screen	21,375 günler?	16.11.2021 08:00	07.12.2021 17:00	Yunus Emre Topal
218		Implementing Post Template Creation Feature	21,375 günler?	16.11.2021 08:00	07.12.2021 17:00	Yunus Emre Topal
219		Converting fetch api calls to axios api calls	21,375 günler?	16.11.2021 08:00	07.12.2021 17:00	Yunus Emre Topal
220		Implementing Search Community Feature	8,375 günler?	05.12.2021 08:00	13.12.2021 17:00	Yunus Emre Topal
221		Displaying text and image fields in Post Screen	25,375 günler?	16.11.2021 08:00	11.12.2021 17:00	Yunus Emre Topal
222		Fixing Subscribe to a Community Feature	27,375 günler?	16.11.2021 08:00	13.12.2021 17:00	Yunus Emre Topal
223		Tokenizing axios api call	1 gün?	16.11.2021 08:00	17.11.2021 08:00	Yunus Emre Topal



Tabular Format

Backend

Task	Start Date	End Date	Assignees
Studying Django Framework to catch up	26.10	04.11	Berke, Emrah
Participating in Backend Meeting 1	05.11	05.11	Berke, Emrah
Learning Activity Streams	06.11	10.11	Berke, Emrah
Learning JsonLD	06.11	10.11	Berke, Emrah
Sketching the initial outline for Backend API	04.11	05.11	Berke
Initializing the Django project for the backend app	05.11	05.11	Berke, Emrah
Migrating the model from the practice app to the new Django project.	05.11	06.11	Berke, Emrah
Implementing initial registration functions	06.11	07.11	Berke
Implementing initial post template functions.	09.11	11.11	Berke
Implementing initial posting functions	09.11	11.11	Berke
Implementing initial community view functions	10.11	11.11	Emrah
Implementing initial post feed functions.	11.11	11.11	Emrah
Writing unit tests for the Backend API	09.11	12.11	Berke, Emrah
Learning Django Rest framework	11.11	12.11	Berke, Emrah
Participating in Backend Meeting 2	09.11	09.11	Berke, Emrah

Writing serializer classes for the REST API framework	12.11	12.11	Berke
Adapting existing functions to Django REST API framework.	12.11	13.11	Berke, Emrah
Participating in Backend Meeting 3	12.11	12.11	Berke, Emrah
Adding Swagger auto-documentation functionality to the code.	13.11	13.11	Berke, Emrah
Configuring endpoints for JSON-LD standards	13.11	13.11	Emrah
Learning Amazon AWS	13.11	14.11	Berke, Emrah
Making database migration from SQLite to MySQL	13.11	13.11	Berke, Emrah
Deploying backend services to AWS instance	14.11	15.11	Berke, Emrah
Working on fixing CSRF Token errors	16.11	16.11	Berke, Emrah
Working on fixing CORS errors	16.11	16.11	Berke, Emrah
Dockerizing the backend application	15.11	16.11	Berke
Adding secrets to the backend codebase	15.11	16.11	Berke
Milestone 2			
Fixing the CORS and CSRF error	25.11	27.11	Berke, Emrah
Fixing the authorization problem	25.11	27.11	Berke
Implementing token-based authorization scheme	25.11	27.11	Berke

Configuring dockerfiles for container based deployment	24.11	25.11	Berke
Implementing GetHomeFeed method	08.12	09.12	Emrah
Implementing SearchCommunities method	08.12	09.12	Emrah
Implementing SearchPostInCommunity method	14.12	15.12	Berke
Reconfiguring GetCommunityData endpoint	09.12	10.12	Berke
Reconfiguring Community serializer class	09.12	10.12	Berke
Doing preliminary research for advanced filtering implementation	03.12	14.12	Berke, Emrah
Doing preliminary research for privacy features	03.12	14.12	Berke, Emrah
Researching APIs for geocoding	02.12	06.12	Berke, Emrah
Writing a simple bash script for a deployment automation	01.12	01.12	Berke
Debugging existing methods from Milestone 1	19.11	13.12	Berke, Emrah
Enhancing endpoints based on frontend requests	26.11	13.12	Berke, Emrah
Writing unit tests for functions	26.11	09.12	Berke, Emrah
Adding definitions to Endpoints on Swagger	18.12	18.12	Berke, Emrah

Frontend - Web

Studying ReactJS-Bootstrap	26.10	08.11	Bilal, Muhammed, Emir
Participating in Frontend Meeting 1	08.11	08.11	Bilal, Muhammed, Emir
Learning React Hooks	08.11	10.11	Bilal, Muhammed, Emir
Learning React Axios	08.11	10.11	Bilal, Muhammed, Emir
Learning React Navigation	08.11	10.11	Bilal, Muhammed, Emir
Learning React Redux-thunk	08.11	10.11	Bilal, Muhammed, Emir
Participating in Frontend Meeting 2	10.11	10.11	Bilal, Muhammed, Emir
Creating Register Page	12.11	15.11	Muhammed
Creating Login Page	12.11	15.11	Muhammed
Creating Create Post Template Page	12.11	15.11	Muhammed, Emir
Creating Create Post Page	12.11	15.11	Muhammed, Emir
Creating My Posts Page	12.11	15.11	Muhammed
Creating Communities Page	12.11	15.11	Emir
Creating Feed Page	12.11	15.11	Muhammed, Emir,Bilal
Creating Create Community Page	12.11	15.11	Muhammed, Bilal
Creating My Communities Page	12.11	15.11	Emir
Initialize React App	05.11	05.11	Bilal
Creating example screens and components.	06.11	15.11	Muhammed, Emir, Bilal
Creating Sidebar	07.11	07.11	Bilal

integration of Axios	08.11	08.11	Bilal
integration and creating examples of Redux	08.11	08.11	Bilal
Integration of React Navigation	09.11	09.11	Bilal
Implementation of React-bootstrap	09.11	09.11	Bilal
Backend integration	11.11	15.11	Bilal
AWS server creation and deployment	13.11	15.11	Bilal
Milestone 2			
Connecting Create Post Template Page to backend	09.12	13.12	Muhammed
Updating Create Post Template Page design	09.12	13.12	Muhammed
Connecting Create Post Page to backend	09.12	13.12	Emir
Tried to implement google maps API but encountered some issues.	09.12	14.12	Muhammed
Added project logo and project name to the site.	13.12	13.12	Muhammed
Fixed image display on posts using Carousel	10.12	13.12	Muhammed
Presenting Web Frontend on Customer Demo 2	14.12	14.12	Muhammed
Merged Web branch to master branch for tagging.	14.12	14.12	Muhammed
Updating Post component with Material UI	09.12	13.12	Muhammed

Searching Docker to dockerize our applications	01.12	03.12	Bilal
Deploying Frontend part to EC2	14.12	14.12	Bilal
Searching Launch template on AWS	04.12	07.12	Bilal
Searching AutoScale on AWS	07.12	010.12	Bilal
Trying to integrate create post with backend	10.12	11.12	Bilal
Trying to integrate create post template with backend	11.12	12.12	Bilal
İntegrate listing page with backend	12.12	14.12	Bilal
Fixing the bug related to the Subscribe button	08.12	13.12	Emir
Implementing Search Community Feature	09.12	13.12	Emir
Adding alert component to Create Community Page	09.12	13.12	Emir
Changing the structure of Community Side Card buttons	11.12	13.12	Emir
Adding alert component to Create Post Page	10.12	13.12	Emir
Adding alert component to Create Post Template Page	10.12	13.12	Emir

Frontend - Mobile

Milestone 1			
Learning React Native	26.10	10.11	Yunus Emre, Elif
Creating Initial App	09.11	09.11	Yunus Emre,
Creating Welcome Screen	09.11	09.11	Yunus Emre, Elif
Creating Login Screen	09.11	09.11	Yunus Emre, Elif
Implementing API call for Login feature	10.11	15.11	Yunus Emre, Elif
Creating Registration Screen	09.11	09.11	Yunus Emre, Elif
Implementing API call for Registration feature	10.11	15.11	Yunus Emre,
Creating Feed Screen	09.11	10.11	Yunus Emre,
Implementing API call for getting Feed data	10.11	16.11	Yunus Emre,
Creating Communities Screen	09.11	10.11	Yunus Emre, Elif
Implementing API call for getting Communities	10.11	16.11	Elif
Creating User Posts Screen	09.11	10.11	Yunus Emre, Elif
Implementing API call for getting User Posts	10.11	16.11	Yunus Emre, Elif
Creating User Communities Screen	09.11	10.11	Yunus Emre, Elif
Implementing API call for getting User Communities	10.11	16.11	Elif
Creating Settings Screen	09.11	09.11	Yunus Emre,
Creating Post Screen	09.11	10.11	Yunus Emre, Elif
Creating Community Screen	09.11	10.11	Yunus Emre, Elif
Creating Drawer Navigation	09.11	09.11	Yunus Emre,

Creating Axios Service	15.11	16.11	Elif

Milestone 2			
Creating Community Creation Screen	16.11	7.12	Yunus Emre,
Implementing Community Creation Feature	16.11	7.11	Yunus Emre,
Creating Post Template Creation Screen	16.11	7.11	Yunus Emre,
Implementing Post Template Creation Feature	16.11	7.11	Yunus Emre,
Converting fetch api calls to axios api calls	16.11	7.11	Yunus Emre,
Implementing Search Community Feature	5.12	13.12	Yunus Emre,
Displaying text and image fields in Post Screen	16.11	11.12	Yunus Emre,
Fixing Subscribe to a Community Feature	16.11	13.12	Yunus Emre,
Tokenizing axios api call			