Cmpe 451
Milestone 1

Group 10
Berkay Demirtaş
Ekin Güngör
Enes Aydoğduoğlu
Gülşah Çilingiroğlu
Muhammet Ekrem Gezgen
Onur Sefa Özçıbık
S. Gökberk Yılmaz
Safa Burak Altunel
Taha Kalkanlı

November 20, 2021

Contents

1. Executive Summary 4
2. List and Status of Deliverables/ Evaluation 4
3. Evaluation of Tools 8
4. Requirements Addressed 9
5. A Summary of Work Done by Each Team Member 10
6. Deliverables 12
6.1 Requirements 12
1. Functional Requirements 13
1.1. User Requirements 13
1.1.1. Sign Up 13
1.1.2. Sign In 13
1.1.3. Guest Users 13
1.1.4. Registered Users 13
1.2. System Requirements 15
1.2.1. Recommendation 15
1.2.2. Notification 15
1.2.3. Authorization 15
1.2.4. Community 15
1.2.5. Searching/Filtering 16
1.2.6. Home Page 16
1.2.7. Registered User 16
1.2.8. Tagging 16
2. Non-Functional Requirements 16
2.1. Security 16
2.2. Performance 16
2.3. Availability 16
2.4. Standarts 16
2.5. Privacy 17
6.2 Scenarios and Mockups 17
6.3.2 Scenario 2 24

6.3 Software Design documents in UML.....41
6.4.1 Use Case Diagram.....41
6.4.2 Class Diagram.....42
6.4.3 Sequence Diagrams.....43
6.4 Project plan & RAM.....51
6.4.1 Project Plan.....51

1. Executive Summary

We have done our first customer meeting and get feedbacks from our customers. We have implemented sign in, sign up and editing profile page functionalities until now. In addition to that, backend team started to implement post creation steps. We have deployed our project to the Amazon Web Services before customer meeting but we didn't create a continuous integration and continuous delivery(CI/CD) pipeline yet. Therefore, whenever we make some changes on our project, we need to deploy it again. This is not a practical way so we are planning to create CI/CD pipeline as soon as possible. Our backend and frontend are not connected because requests from frontend are not responded by the APIs in an expected way due to CORS error. We will fix this issue soon. We have decided to make some changes about implementation of Purbee. According to this decision, backend team will prepare a documentation about endpoints needed by frontend and android teams before implemention of those APIs. In this documentation, input-output format for each endpoint will be explained clearly. We also want to create a swagger page to clear up any kind of misunderstanding between backend team and other teams. As a next step of implementation of Purbee, we are planning to implement post creation steps.(creating post type and creating posts from post templates.) There are posts almost on every page. Therefore, implementation of post object will make implementation of the other pages easier. As second step, we will implement feed page of user. Feed page is the page displayed after sign in. Hence, implementation of feed page will make Purbee more presentable in next customer meeting.

2. List and Status of Deliverables/ Evaluation

We had a pilot version of our project for the first customer meeting. Though we lacked some functionalities we wanted to present, there was a solid output of what we've been trying to do since the beginning of semester.

Backend

Deployment	**Done**
Non-relational database	**Done**
Updated Project Details	**Done**

Login/Signup	**Done**
Updating Profile Page	**Done**
Implementation of Post	**Partially Done**

1- Deployment: We were asked to present a deployed version of our product in the customer meeting as it is more convenient to make things work on platforms other than local computer.
2- Non-Relational database: An application is nothing without a database, we had to have a database that is accessible everywhere.Therefore, we decided to have it on cloud. We decided a document oriented would be a better approach for our application. We have collections instead of tables
3- Updated Project Details: We prepared diagrams and requirements long before implementation. Therefore, we had to revise them thoroughly and make the necessary changes. Since we were about to start implementation, we had a more realistic approach and developed a bettter perspective of the project.
4- Login/Signup: It is the very first step of a potential user to use our application. Therefore, we decided to start implementing it at first. It was the page that made the most sense to start with.
5- Updating Profile Page: Right after registering or logging in, profile page welcomes the user. So, we continued with implementation of this page.
6- Implementation of Post: The most essential part of our project is the post since it's a social media platform. That is what we offer to the user, posts.

FrontEnd

Base App	**Done**
Page Navigation	**Done**
Deployment	**Done**
Login/Signup Page	**Done**
Updating Profile Page	**Done**
Connection to Backend	**In-progress**
Implementation of Post	**Postponed**
Implementation of Feed	**Postponed**
Updated Project Details	**Done**

1- Base App: Creating an app and start building of components in it was essential. We studied react, javascript, html, and css. This process took time. Then, we added some pages as components by time.
2- Deployment: We were asked to present a deployed version of our product in the customer meeting as it is more convenient to make things work on platforms other than local computer.
3- Login/Signup: It is the very first step of a potential user to use our application. Therefore, we decided to start implementing it at first. It was the page that made the most sense to start with. Data is not stored in database because we are yet to connect the backend.
4- Updating Profile Page: After logging in or registering, the first thing to navigate is profile page. They are not yet connected to backend, so we couldn't project checks done in backend like mail address control, avoiding duplicate user names etc. However once we connect to backend, these will be provided.
5- Connection to Backend: We encountered CORSE error while trying to reach api that our backend team provided. Gökberk has faced this issue beforehands and the way that solved the problem didn't work for our case, we suspect some change needs to be done in backend side. We are in progress to solve it, we will collaborate with the backend team.
6- Updating Project Details: We prepared diagrams and requirements long before implementation. Therefore, we had to revise them thoroughly and make the necessary changes. Since we were about to start implementation, we had a more realistic approach and developed a bettter perspective of the project.
7- Page Navigation: Navigation between pages is essential, however the ways we found online didn't work on our case. It turned out that the older method deprecated. A new routing version was added. There is a little work more to be done, but it was ready for the customer meeting.
8- Implementation of Post: One of the next steps we will take, it has a high priority beacuse that's what the user will interact with as the core part of our application.
9- Implementation of Feed: Once post is implemented feed will come next. To provide a more solid experience of user, this also needs to be done.

Android

Base App	**Done**
Deployment	**Done**
Login/Sign up	**Done**
User Authentication	**Done**
Backend Connection	**Done**
Implementation of Post	**Postponed**

1-Base App: Android is a different programming style that none of our development team is familiar with. Studying and creating an android app using Kotlin was the first step of development.

2- Deployment: We were asked to present a deployed version of our product in the customer meeting as it is more convenient to make things work on platforms other than local computer.

3- Log in/Sign up: It is the very first step of a potential user to use our application. Therefore, we decided to start implementing it at first. It was the page that made the most sense to start with.

4- User authentication: Users must go through an authentication process, it is a must to validate a user by checking credentials stored in the database.

5- Backend Connection: Our mobile app is fully connected to the backend, all the functionalities interact with the backend and make changes in database if necessary.

6- Implementation of Post: One of the next steps we will take, it has a high priority beacuse that's
what the user will interact with as the core part of our application.

3. Evaluation of Tools

Discord: We used Discord for main communication channel for meetings. We used screen sharing property on decisions, creation of diagrams. Also, camera option was useful when we needed more realistic and diagonal communication. Discord also has no time or user limitations on meetings. Because we arranged meetings often and these meetings were taking long periods, the time limit was important to us.

Slack: Main usage of slack was communicating with each other about subjects that have more importance than daily needs. Also, we were able to share GitHub links. So that, for being up to date on GitHub, it is crucial part of our communication.

Draw.io: Our team frequently needed diagrams. Draw.io is a platform where we can implement our ideas without any payment or any limitations about the used material number or size. Also, changing the format of to file to .png or .jpg file eased our tasks.

LucidChart: When team needed diagrams we used this application as another choice. It is easy to use and it has tools which implements our needs easily. The negative part of this application is it comes with some limitations in free version. Still, it was very useful in small diagrams.

Adobe XD: Our team used this program in creation of mockups. It offers some handful properties such as two users can implement same file at the same time. Also, it has designing tools. We could add predefined structures directly or via modifying them.

Online-Stopwatch.com (Duck Race): We sometimes used this tool for deciding which person should take which part of the main task. Also, we used it to decide the team members. Because of the funny way of getting randomness of it, we enjoyed in times which we distribute project roles.

GitHub: Our project is on GitHub. It uses git and updating a project is very easy. Also, it gives power to document our project clearly. We are able to share information about our meetings, weekly efforts, profiles etc. Issue property is also very useful. Issues are used to divide tasks to our group members and specifying the needs of the project in a proper manner.

Flask: Flask is a Python framework that enables programmers to create API functionalities with ease. Because our backend team has common knowledge about Python and Flask, we have used this tool. Also, their community is very big and helpful.

React: React is a JavaScript framework that serves easy ways to create user interfaces. It offers additional libraries to enrichen the contents. Also, with these libraries, programmers can be able to labor only with the states and the rendering processes of the application. Addition to having enormous community, they are free.

Kotlin: It runs over Java and JVM. This framework is used to create Android application. It is easy to pick up an application. Thus, we used it. Kotlin produced by Jetbrains. After 2021, it is open source and free.

MongoDB: It is NoSQL database and uses object documents to hold information. This property offers flexible dependencies. Also, it can be used by variety of coding languages or frameworks. They also offer a monitoring UI in their website to check the current state of the database.

AWS: They offer high variety of service with free option. We needed to create EC2 instance to run backend operations. Also, with their Gateway technology, it is easy to monitor and handle API calls. Also, they offer a system in which a programmer deploys their frontend react project within seconds. They have tons of documentations about all of their services.

4. Requirements Addressed

Sign up:

- 1.1.1.1. Users shall be able to start the sign up process by providing an user-name, e-mail and password in the sign up page.
- 1.1.1.2. The user shall provide unique user-name and e-mail in the sign up process.

Sign In:

- 1.1.2.1. Registered users shall be able to sign in via username/password or e-mail/password in the sign in page.
- 1.1.2.2. Registered users who signed in shall be able to sign out successfully when log out button clicked by registered user.

5. A Summary of Work Done by Each Team Member

Name	Summary of Contribution
Safa Burak Altunel	Implemented login and sign in feature in android application. İmplemented exception handling like no internet case. Created simple navigation drawer and a main page. Taken second meeting notes. Opened some issues on github. Added commits on requirements page.
Ekin Güngör	Revised and updated requirements and class diagram. Attended to meetings. Implemented sign_in, sign_up and profile_page endpoints. Dockerized the app. Deployed it to AWS. Partially implemented Post (in the sense of social media app) functionalities. Took care of GitHub branches.
Gülşah Çilingiroğlu	Actively participated revision of requirements and class diagram. Taken third meeting notes. I edited the wiki page to separate the grades of the 352 and 451 courses. Updated the sequence diagram 'create community'. Practiced kotlin during the sign in sign up page implementation process.
Taha Kalkanli	Attended all the meetings and participated. I reviewed and updated class diagrams, use case diagrams, project plan, and sequence diagrams with my friends. I also reviewed the updated sequence diagrams of everybody based on the updated class diagram on Wiki. I updated and uploaded the class diagram according to the changes made to on reviewed sequence diagrams. Made research about widely popular programming languages for frontend developing. Learned how to code React.js Refined the sign-in and sign-up page in a more stylish way. I implemented a web page where users can update their profile info after signing up. Deployed the frontend client to AWS. Created and edited the final Milestone 1 report.
Onur Sefa Ozcibik	Helped to implement API endpoints. Actively attended meetings. On requirements and class diagram update meetings, proposed lots of changes and improvements. Prepared AWS environment to deploy it.Updated wiki home page. Prepared documentation about endpoints. Prepared dockerfile. Revised a pull request.
Berkay Demirtas	I attended all the necessary meetings.I updated sending message sequence diagram.I updated requirements and class diagram.I was in the backend team. I attended discussion about which technology we will use as backend team. We decided to use Flask framework.I create a mongodb atlas database for our project and connected it to the our backend. There were a lot of bugs and a lot of missing points in first implementation of sign up and sign in endpoints. I fixed the bugs and implemented edge cases.I implemented fields.py and post_fields.py files. This files will be used in the implementation of post creation process. At this point they don’t have endpoint.I implemented edit profile page functionality. Name of this api is profile_page. This endpoint handles get and post requests. It updates the profile info of a user with post request

	and get the profile info of the user with an get request.We deployed our project to AWS by creating docker image as backend team.I made a presentation of Purbee for customer meeting and presented it during customer meeting.
Gokberk Yilmaz	Participating in almost all of the meetings. Working on updates and revision done on requirements and diagrams. Going over them thoroughly with the team. Taking notes of the 1st meeting. Studying react, javascript, html, and css. Implementing login and register pages without styling. Handling navigation of pages. Working on api call, trying to solve cors policy error.
Ekrem Gezgen	
Enes Aydogduoglu	Participating Zoom and Discord meetings. Actively participating revision and update of requirements, Class, Use-case and sequence diagrams. Opened a discussion for selecting the framework to be used in Android application. Studying Kotlin during the sign-in sign-up pages implementation process. Reviewing my teammates' works.

6. Deliverables

6.1 Requirements

Glossary

- **Admin:** The main rulers of the application. This position is only for application developer team.
- **Community:** A platform with some ideology around particular subject. It is the main communication channel to share created purposeful content.
- **Community Admin:** Superior manager of a community. This person manages all the data of the community and the community members.
- **Community Creation Page:**
- **Community Imperatives:** Compulsory fields that must take place in any of the community post types.
- **Community Post Types:** Varios post types in a community, either created and customized by community creator or given default(text and event).
- **Community Profile Page:** A dedicated page for every other community to show profile picture, manifest, list of subscribers, and timeline to users.
- **Content:** Anything published in a Community.
- **Default Post Types**: "Community Post Types" that exist without the effort of community creator, upon a creation of a community. Text and Event.
- **Engaging discussion:** Replying a comment.
- **Event:** A content which is an invitation and announcement for some activities.
- **Field:** A single piece of information that is part of a post or a template.
- **Follower:** A user who wants to see some other users content. To be a Follower, the user should be approved by the aimed user.
- **Guest User:** Person who is not registered to the application. This person still can use the application with some restrictions.
- **Interactive fields:** Poll and Participation fields.
- **Main Feed:** A Registered User Spesific page which shows posts of followed users and subscribed communities.
- **Manifest:** Introduction text of a user or a community to give brief explanation about themselves to other users.
- **Non-Follower:** A user who does not follow the specified user.
- **Notification Center:** A page which accumulates recieved notifications regarding to a registered user.
- **Post:** A content which has predefined structures to bring out some experience a user acquired or cumulate some knowledge.
- **Post Type Creation Page:** A page which is used to create new post types from given fields as a part of community profile page.
- **Privacy Level 1:** Community is public and user is not a member of community.
- **Privacy Level 2:** Community is public and user is a member of community.
- **Privacy Level 3:** Community is private and user is not a member of community.
- **Privacy Level 4:** Community is private and user is a member of community.
- **Profile Page:** A personal page for every other user to show personal data(profile picture,

biography, subscribed communities and timeline) to other users.

- **PSP (Purpose Spesific Post) Template:** Basic post template that could be configured to made a community default by community admin for members to use. Provided by the system.
- **Public Content:** Content which is visible to not only subscribers, but all users and guests.
- **Search Bar:** The bar that enables users to search for communities and registered users.
- **Subscription:** Action that a user can take. After the subscription, a user can reach all the content from subscribed community.
- **Timeline:** Posts that a user post to a public community.
- **Turning on notifications of a post:** The action of receiving notifications about any interactions related to the post.
- **User:** Person who registered to the application. This person can use all benefits of the application. For example he/she can create or join to a community while being able to keep contact with other users.

1. Functional Requirements

1.1. User Requirements

1.1.1. Sign Up

- 1.1.1.1. Users shall be able to start the sign up process by providing an user-name, e-mail and password in the sign up page.
- 1.1.1.2. The user shall provide unique user-name and e-mail in the sign up process. [NON FUNCTIONAL]
- 1.1.1.3. Users shall be able to sign up using their google and facebook accounts.

1.1.2. Sign In

- 1.1.2.1. Registered users shall be able to sign in via username/password or e-mail/password in the sign in page.
- 1.1.2.2. Registered users who signed in shall be able to sign out successfully when log out button clicked by registered user.
- 1.1.2.3. Registered sers shall be able to sign in using their google and facebook accounts.

1.1.3. Guest Users

- 1.1.3.1. Guest users shall be able to search for communities and profiles.
- 1.1.3.2. Guest users shall be able to view the public content of communities and profiles.

1.1.4. Registered Users

- 1.1.4.1. Common Features

- 1.1.4.1.1. Registered users shall inherit the Guest users' requirements specified in requirement 1.1.3.
- 1.1.4.1.2. Registered users' profile page shall have profile picture, biography, list of subscribed communities and timeline.
- 1.1.4.1.3. Registered users shall be able to change their profile picture from their profile page.
- 1.1.4.1.4. Registered users shall be able to subscribe to public communities from the community page.
- 1.1.4.1.5. Registered users shall be able to send request to the community admins for subscribing private communities from the community page.
- 1.1.4.1.6. Registered users shall be able to leave communities that they have subscribed before from the community page.
- 1.1.4.1.7. Registered users shall be able to report the content they find inappropriate or offensive from the post.
- 1.1.4.1.8. Registered users shall be able to directly follow other registered users with public profile from their profile page.
- 1.1.4.1.9. Registered users shall be able to send request for following registered users with private profile from their profile page.
- 1.1.4.1.10. Registered users shall be able to unfollow a registered user from their profile page.
- 1.1.4.1.11. Registered users shall be able to accept or reject following requests from the notification center.
- 1.1.4.1.12. Registered users should be able to contribute to content in a Community within the predetermined ranges that Community Admin has ruled. (You can check the privacy levels (1,2,3,4) from glossary part.)):
    - 1.1.4.1.12.1. Creating Posts from the community profile page via using a post type from community post types. (2,4)
    - 1.1.4.1.12.2. Interact with interactive fields of the post from post. (1,2,4)
    - 1.1.4.1.12.3. Commenting posts from post. (1,2,4)
    - 1.1.4.1.12.4. Hitting the like button whichin the post. (1,2,4)
    - 1.1.4.1.12.5. Replying to comments from the post. (1,2,4)
    - 1.1.4.1.12.6. View Community Posts from timelines and main feed. (1,2,4)
    - 1.1.4.1.12.7. View Community's other members from the community profile page. (1,2,4)
    - 1.1.4.1.12.8. Registered users shall be able to turn on notifications of posts published in the community from the post. (1,2,4)
- 1.1.4.1.13. Registered users shall be able to set privacy option of the Community they create in the community creation process from community creation page.
- 1.1.4.1.14. Registered users shall be able to set community profile photo in the community creation process from community creation page.
- 1.1.4.1.15. Registered users shall be able to set manifest in the community creation process from community creation page.
- 1.1.4.1.16. Registered users shall be able to search and filter posts according to the specific post type and its fields in the "main feed".

- 1.1.4.2. Community Admin

    - 1.1.4.2.1. Community admins shall inherit the Registered User's requirements specified in requirement part 1.1.4.1.

- 1.1.4.2.2. Community admins shall be able to set another subscriber of the community as a community admin from the community's subscriber list.
- 1.1.4.2.3. Community admins shall be able to warn Community Members because of hazardous posts from the post.
- 1.1.4.2.4. Community admins shall be able to remove Community Members from the Community from the community's subscriber list.
- 1.1.4.2.5. Community admins shall be able to edit the Community's community picture from the community profile page.
- 1.1.4.2.6. Community admins shall be able to edit the Community's community manifest from the community profile page.
- 1.1.4.2.7. Community admins shall be able to create post types using these fields from the post type creation page:
  - Plain text field.
  - Photo field.
  - Date-time field.
  - Location field.
  - Price field.
  - Participation field.
  - Document field.
  - Poll field.

1.2. System Requirements

1.2.1. Recommendation

- 1.2.1.1. System shall be able to recommend content and persons of interest.

1.2.2. Notification

- 1.2.2.1. Users who turned on their notifications to a post shall be notified with the developments on that post.
- 1.2.2.2. App should provide notifications to the notification center of the mobile device.

1.2.3. Authorization

- 1.2.3.1. The privacy levels of users who accept the invitations from community admins shall be set accordingly.
- 1.2.3.2. The privacy levels of users who removed from the community will be set accordingly.
- 1.2.3.3. The privacy levels of users will be set according to the privacy option of the community.

1.2.4. Community

- 1.2.4.1. Post histories of communities will be stored.
- 1.2.4.2. Community creator is the default Community Admin.

- 1.2.4.3. Communities' profile page shall have community picture, manifest, subscribed members list and timeline.
- 1.2.4.4. Communities have default post types named as default post types

1.2.5. Searching/Filtering

- 1.2.5.1. Upon searching related communities, people and posts shall be indexed under the search bar.

1.2.6. Home Page

- 1.2.6.1. System shall provide search bar, tabs and community posts.
- 1.2.6.2. System should provide community posts that users are registered and publicly posted contents of followed registered users.

1.2.7. Registered User

- 1.2.7.1. Registered users' profile page should have default profile picture and empty biography when a user have signed up.

1.2.8. Tagging

- 1.2.8.1 The semantics tagging shall be supported by wikidata.org

2. Non-Functional Requirements

2.1. Security

- 2.1.1 User passwords shall be encrypted or hashed
- 2.1.2 The system shall use https Protocol

2.2. Performance

- 2.2.1. 2000 users shall be supported
- 2.2.2. 50 requests shall be responded in a second
- 2.2.3. 4 seconds shall be the maximum response time

2.3. Availability

- 2.3.1. The product shall work as Web application in Google Chrome
- 2.3.2. The product shall work as Android application in Android version 10 and later versions.
- 2.3.3. The system should be available 95% of the time.
- 2.3.4. In the case of failure, the system should recover in at most 2 hours.

2.4. Standarts

- 2.4.1. The system shall use the W3C activity stream standard 2.0 to implement subscriptions and notifications

2.5. Privacy

- 2.5.1. User data shall be protected with compliance to KVKK
- 2.5.2. Users shall accept privacy policy and user terms before they start using the product

6.2 Scenarios and Mockups

6.3.1 Scenario 1

Persona

- Ahmet Yıldız
- 22 years old
- Student
- Currently pursuing bachelor's degree
- Part-time software developer
- Interested in artificial intelligence
- Crazy about hiking and camping
- He loves listening to music while coding

Story

- Ahmet is a junior CMPE student at Bogazici University
- He is a very hard-working and determined student.
- Ahmet loves the department that he is studying and does everything he can to be a good computer engineer.
- In addition to being a hard-working student, Ahmet is a very social person.
- Ahmet constantly follows the activities of school clubs and loves participating in camping activities.
- He uses an application that can follow college club's events.

Preconditions

- Ahmet has already signed up to the application.
- Ahmet previously had subscribed to some communities.

Acceptance Criteria

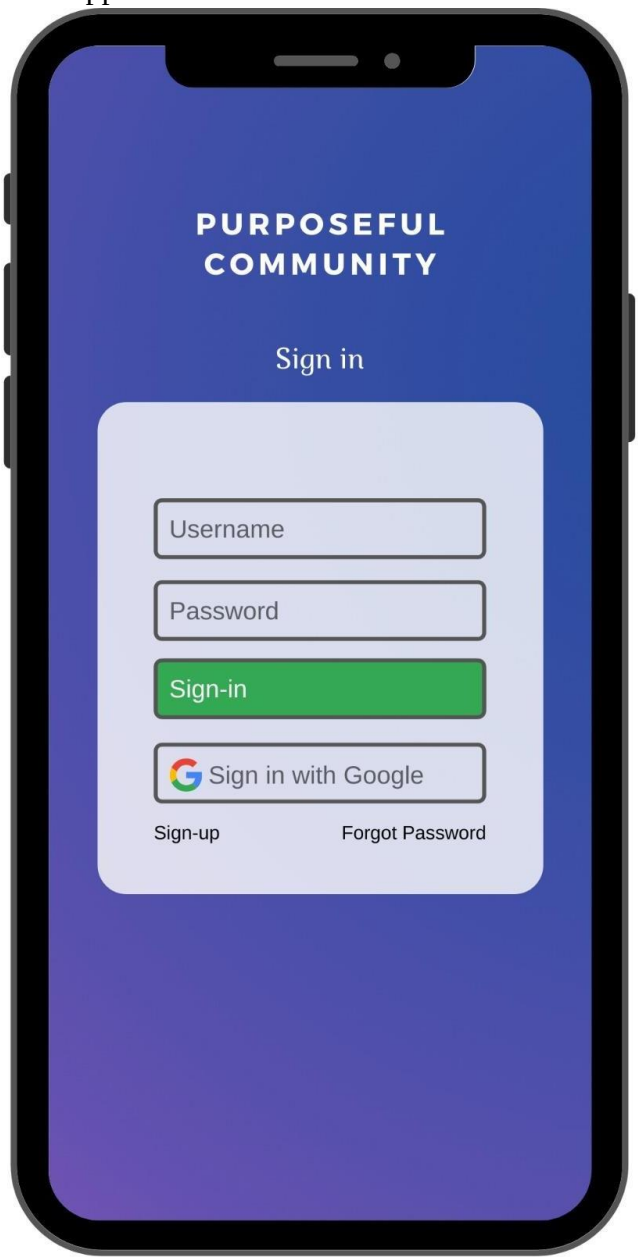
- 1.1.2.1. Registered users shall be able to sign in via username/password or e-mail/password.
- 1.1.4.2.1. Community members should be able to contribute to content in a Community within the predetermined ranges that Community Creator has ruled.
- 1.1.4.2.2. Community members shall be able to specify participation status to Events.
- 1.1.4.2.3. Community members shall be able to view Community Posts and Events.
- 1.1.4.2.4. Community members shall be able to view Community's other members.
- 1.1.4.2.5. Community members shall be able to turn on notifications of posts published in the community.
- 1.2.2.1. Users subscribe to the post shall be notified with the developments on that post.
- 1.2.4.1. Posts, shared by authorized members shall be displayed on the community timeline.
- 1.2.4.3. Posts shall have comment section.
- 1.2.6.1. System shall provide search bar, tabs and community posts.
- 1.2.6.2. System should provide community posts that users are registered.

Scenario with Mockups

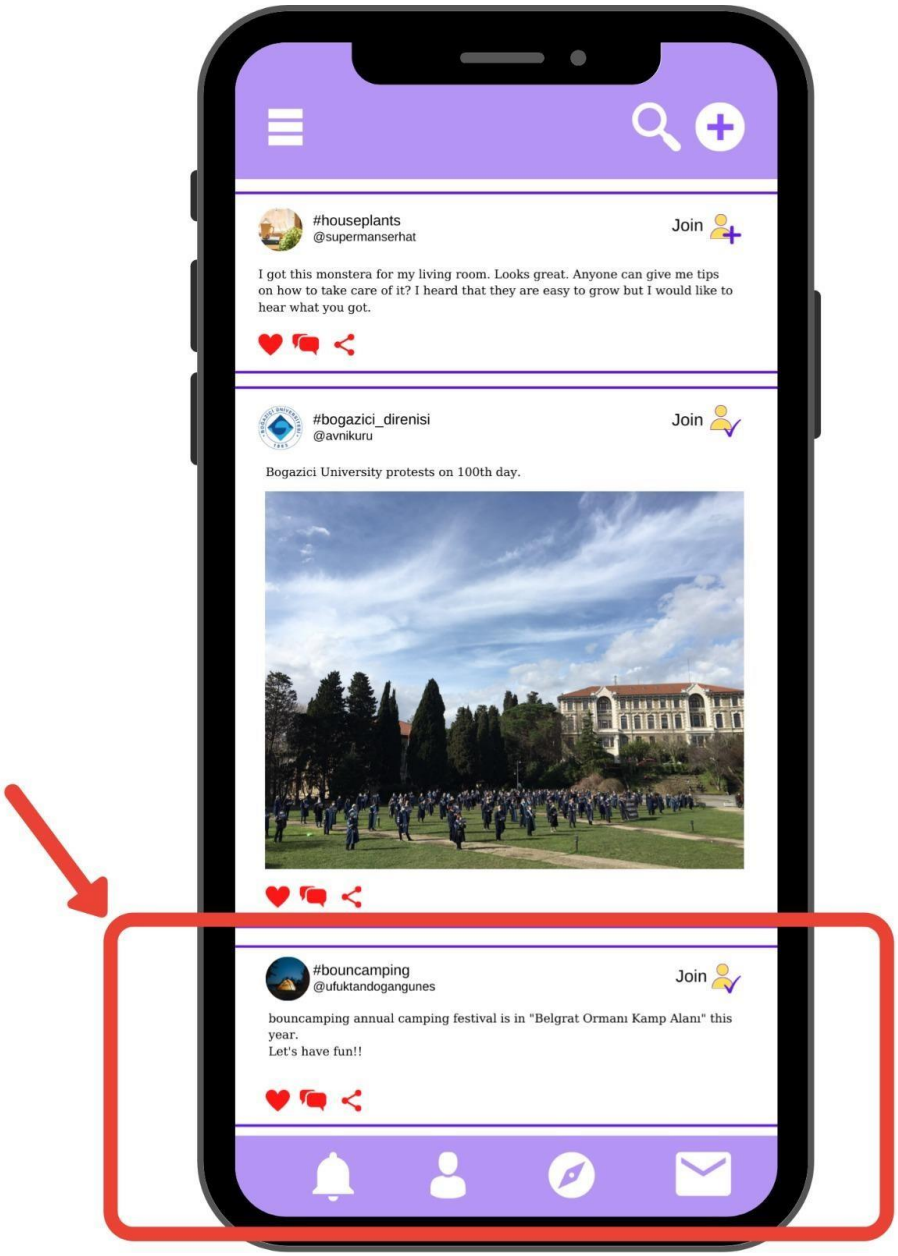
- Ahmet got tired while studying and took a break.
- He took his phone and opened the application.
- He previously had registered to the application so, he entered his e-mail address and password and he signed into the application.
- He saw #bouncamping's post on feed page. This post was about camping event and attracted his attention.
- This post, whose type is determined by the creator, contains the button "I will participate". Also this post has a counter showing the number of people who clicked the "I will participate button".
- Ahmet clicked the "I will participate button".
- "Can those who are considering participating in the event please click the "I will participate button"? Because I am worried about Covid-19, I do not think to come if the number of participants will be greater than 20." Ahmet left this comment under the post.
- He turned on notifications of the post because he wanted to be notified about the number of participants.
- When he re-entered the app after a while, he saw that there were some notifications.
- When he clicked on the notification bar, he saw that 30 people click the "I will participate button".

Because he hesitates over covid-19, he gave up on participating in camping event.

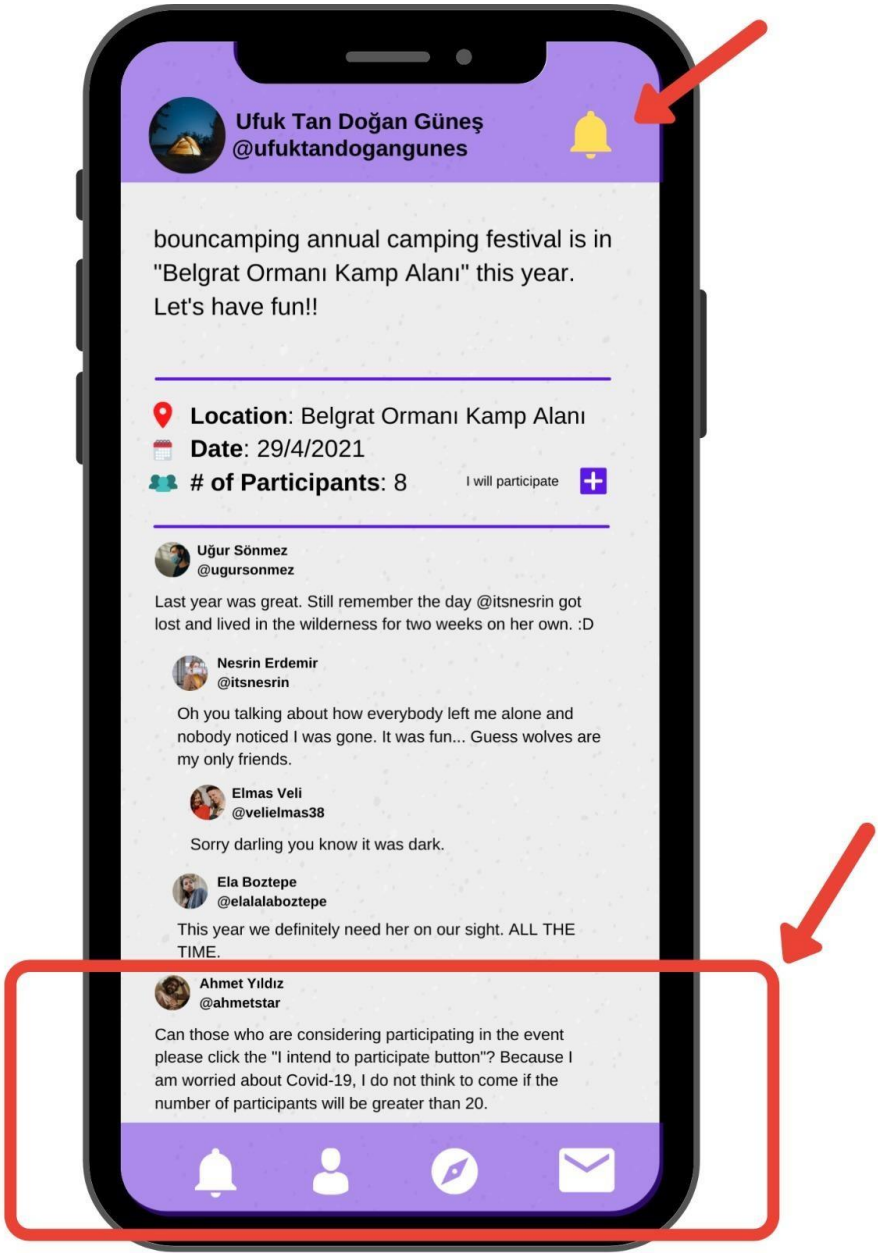
- 1- Ahmet signed into app.



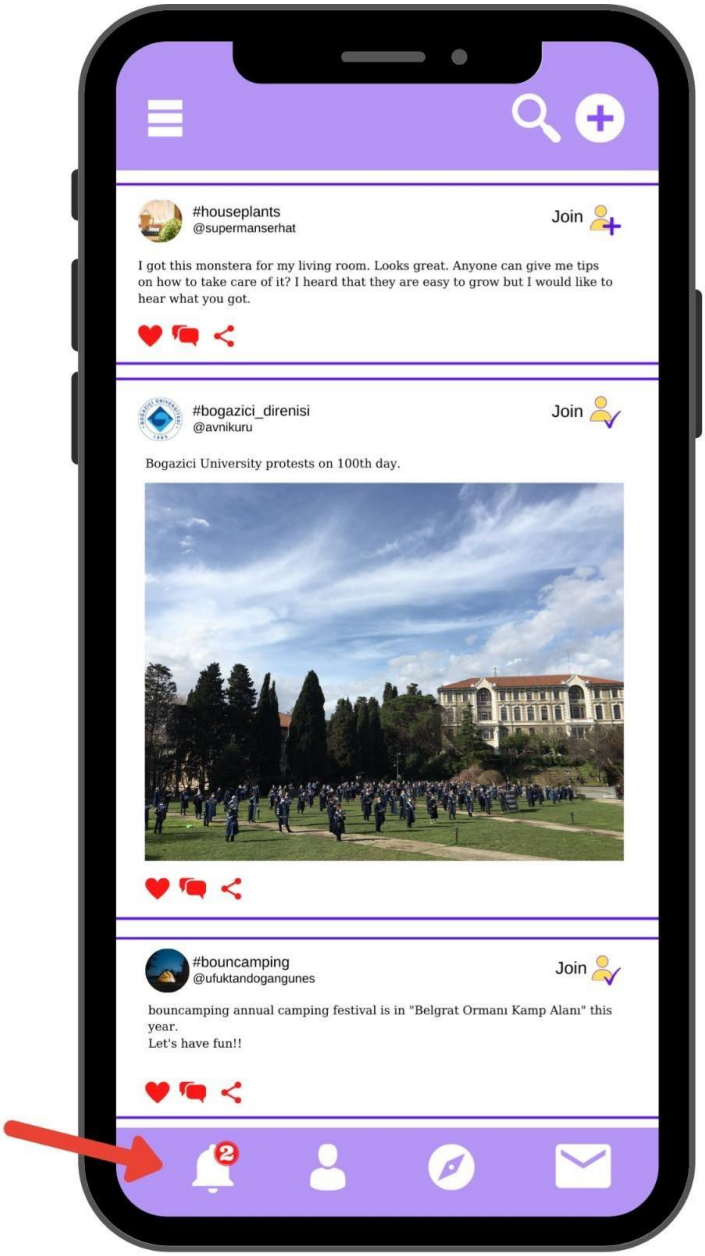
2- When signed into the application, the feed page appeared at first. #bouncamping's post took his attention.



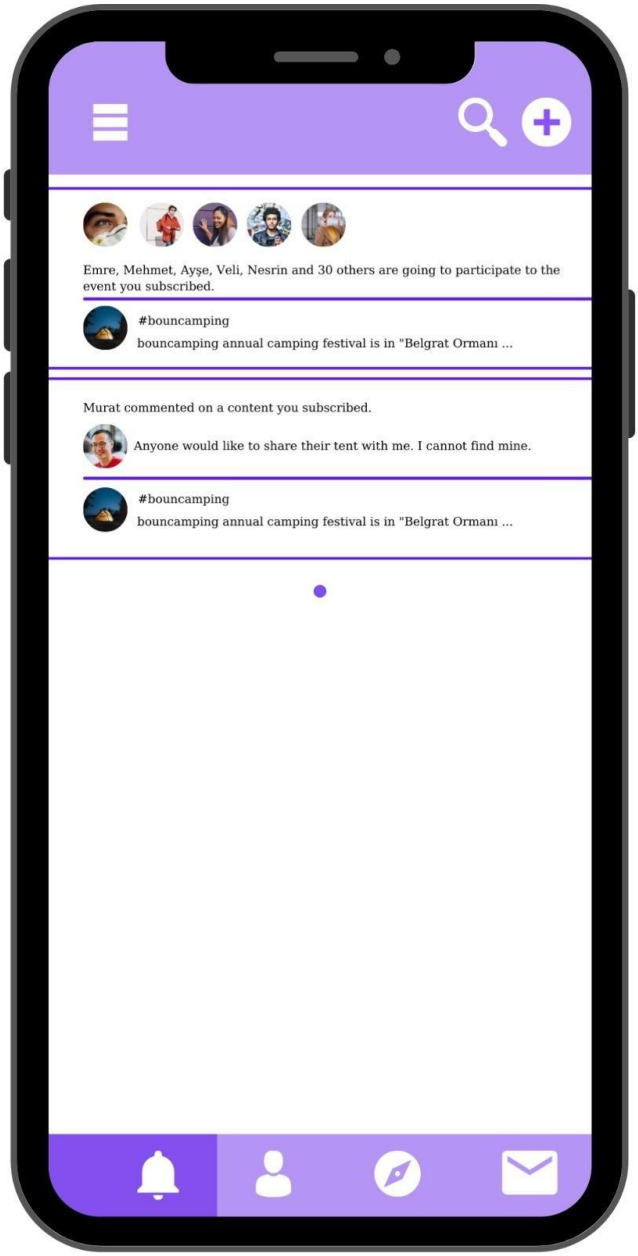
3- He clicked on the post to see the detailed content of the post. He clicked the participation button, left a comment under post and he subscribed to the post(clicked to the bell button).



4- When he re-entered the app after a while, he saw that there were some notifications (there was a mark on notification bar).



5- He clicked on the notification bar and saw app's notifications.


Emre, Mehmet, Ayşe, Veli, Nesrin and 30 others are going to participate to the event you subscribed.
#bouncamping
bouncamping annual camping festival is in "Belgrat Ormanı ...
Murat commented on a content you subscribed.
Anyone would like to share their tent with me. I cannot find mine.
#bouncamping
bouncamping annual camping festival is in "Belgrat Ormanı ...

6.3.2 Scenario 2

Persona

- Gökberk Yılmaz
- 21 years old
- Event Organizer
- Currently pursuing bachelor's degree
- Leader of Compec in Bogazici university

Story

Gokberk Yılmaz is a computer engineering student in Bogazici University. Because he is interested in IT, he want to be a part of Compec. After 2 years of membership, he was elected as leader of the club. However many people are unaware of the club's meetings and events. Therefore he wants to solve this problem.

Goals

- He wants to promote his club.
- Therefore he wants to create a public community for his club.

Preconditions

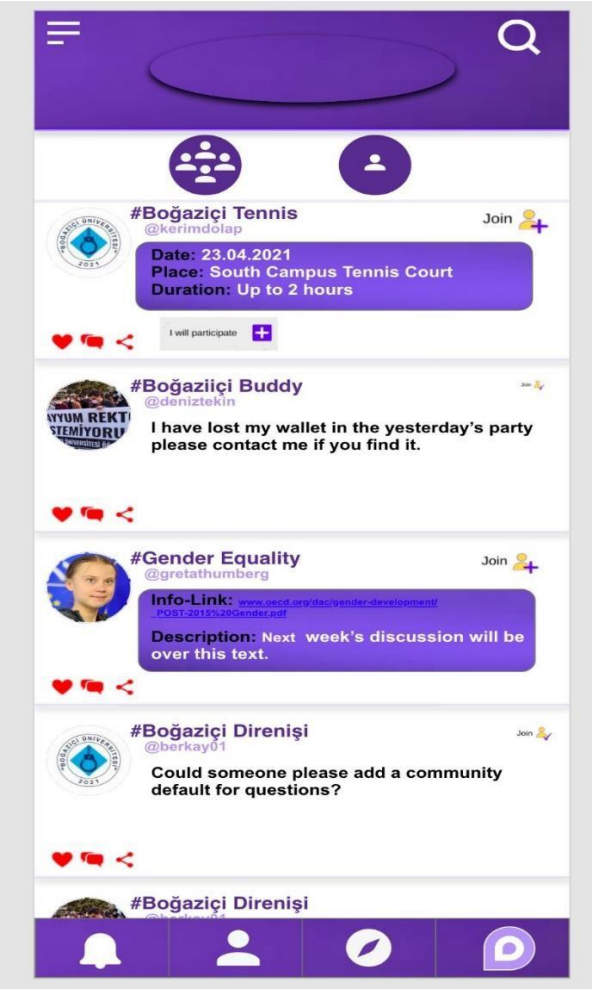
- Gökberk has already signed up to the application.
- Gökberk has already signed in to the application.
- Gökberk previously had subscribed to some communities.

Acceptance Criteria

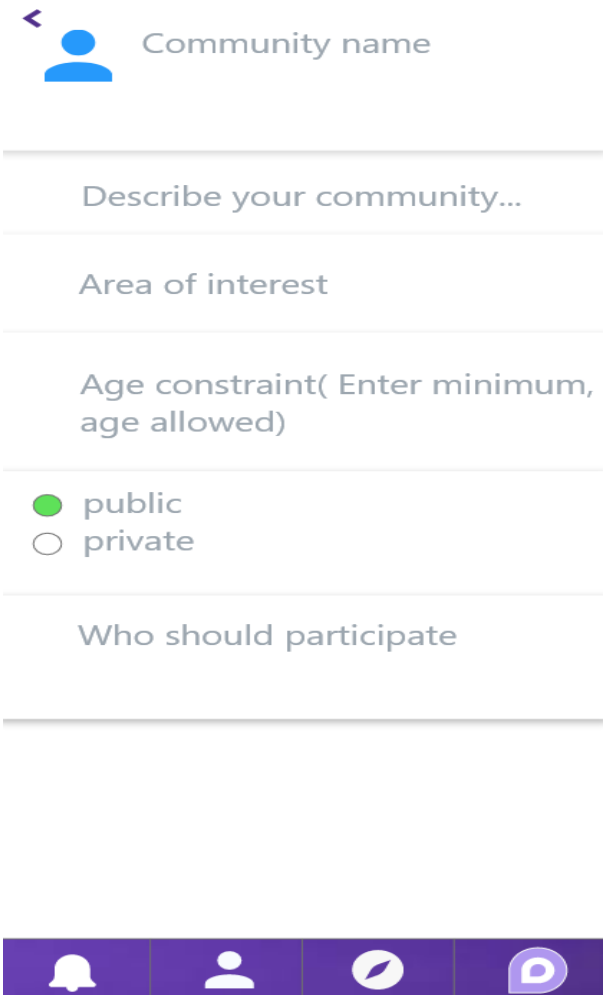
- 1.1.2.1. Registered users shall be able to sign in via username/password or e-mail/password.
- 1.1.4.3.4. Community admins shall be able to set privacy options of the Community.
- 1.1.4.3.3. Community admins shall be able to give authorization to the Community members.
- 1.1.4.3.2. Community Creator is the default Community Admin.
- 1.1.4.3.8. Community admins shall be able to publish/edit the Community Manifest
- 1.2.3.1. Users who accept the invitations shall re-authorized.
- 1.2.4.2. Users shall be authorized according to the community creators decisions.

Scenario with Mockups

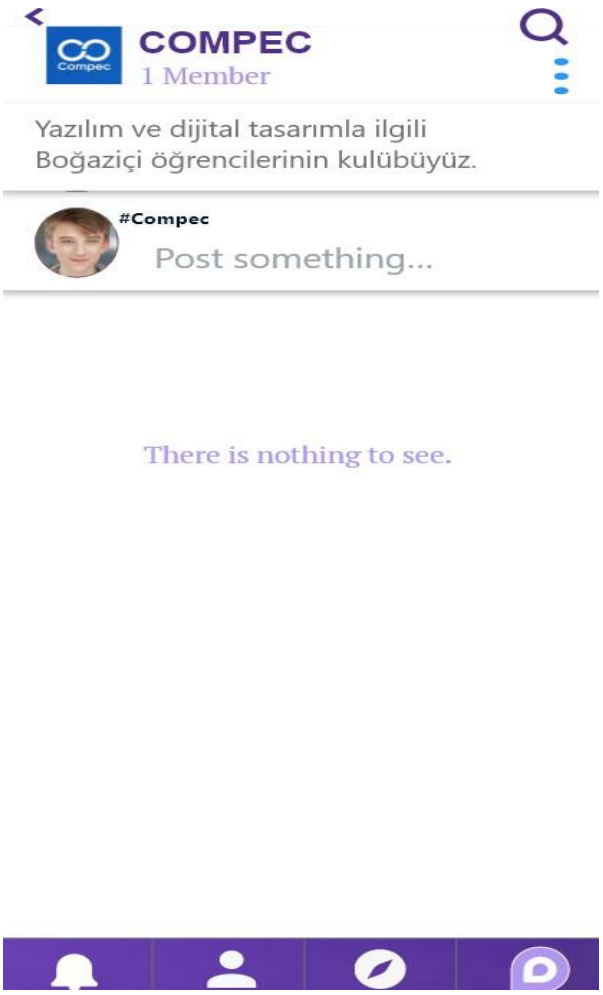
- He logs into his account and main page is opened



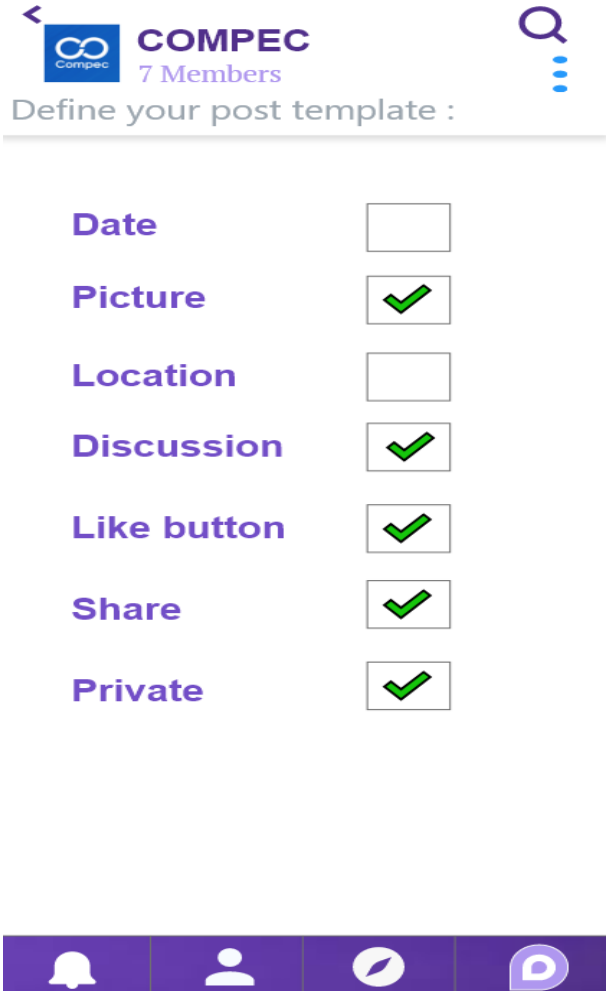
- He clicks create a new community button and writes some necessary informations about community



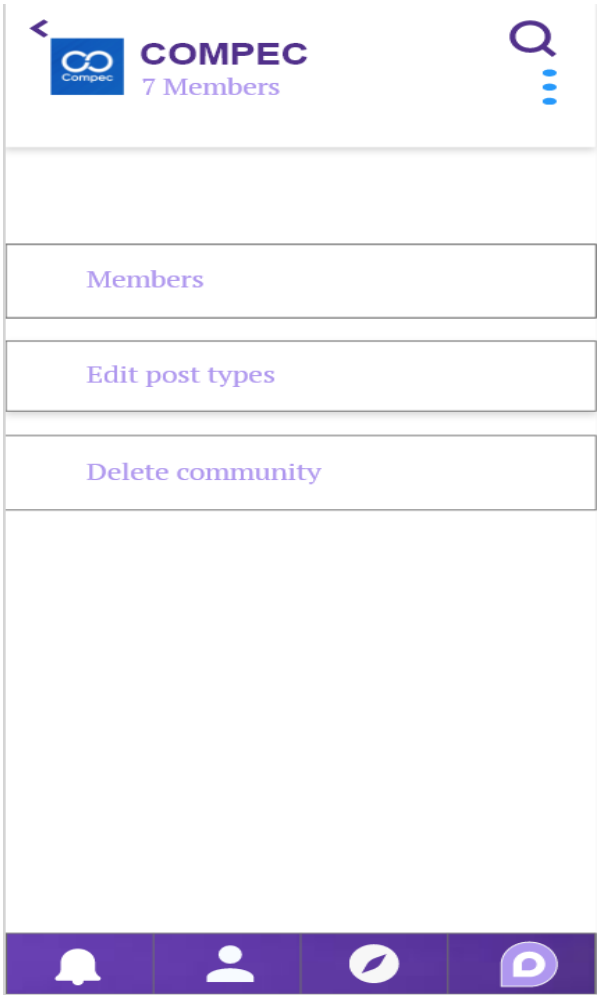
- App navigates him to community page



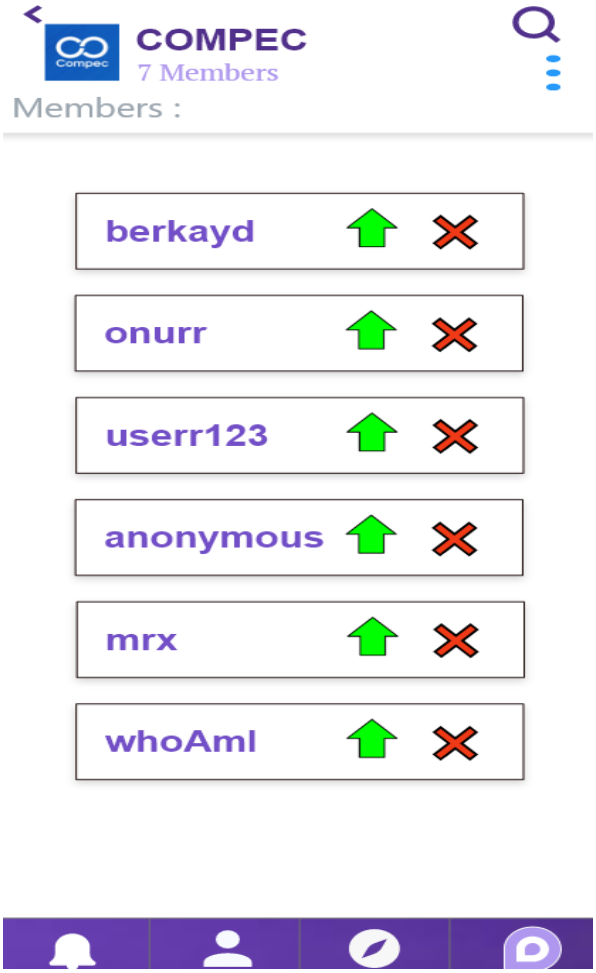
- He decides that some features and information fields of the posts should always be filled. Therefore he opens "edit post type" section and marks the necessary fields.



- After a few days, some new people join the community and Gökberk decides to promote " onurr " as moderator of the community. For that, he clicks " three dot ".



- He opens " members " section and clicks the green arrow of " onurr ".


COMPEC
7 Members
Members :
berkayd
onurr
userr123
anonymous
mrx
whoAml

6.3.3 Scenario 3

Persona

- Mabel Brooks
- 21 years old
- Currently pursuing bachelor's degree in Birmingham
- She is going to be in Bogazici University next semester as an Erasmus Student

Story

- Mabel Brooks is a Computer Science student in University of Birmingham.
- She is very excited about her next semester in Bogazici.
- In her last days in England, she wants to learn more about Bogazici, its courses and atmosphere.
- She even became a member of Bogazici Community in the app in order to meet some people and follow some student clubs before coming here.

Goals

- She wants to get to know Bogazici and discover new student clubs.
- She wants to find her department's community so that she can find answers to any question she may have about the department and courses.

Preconditions

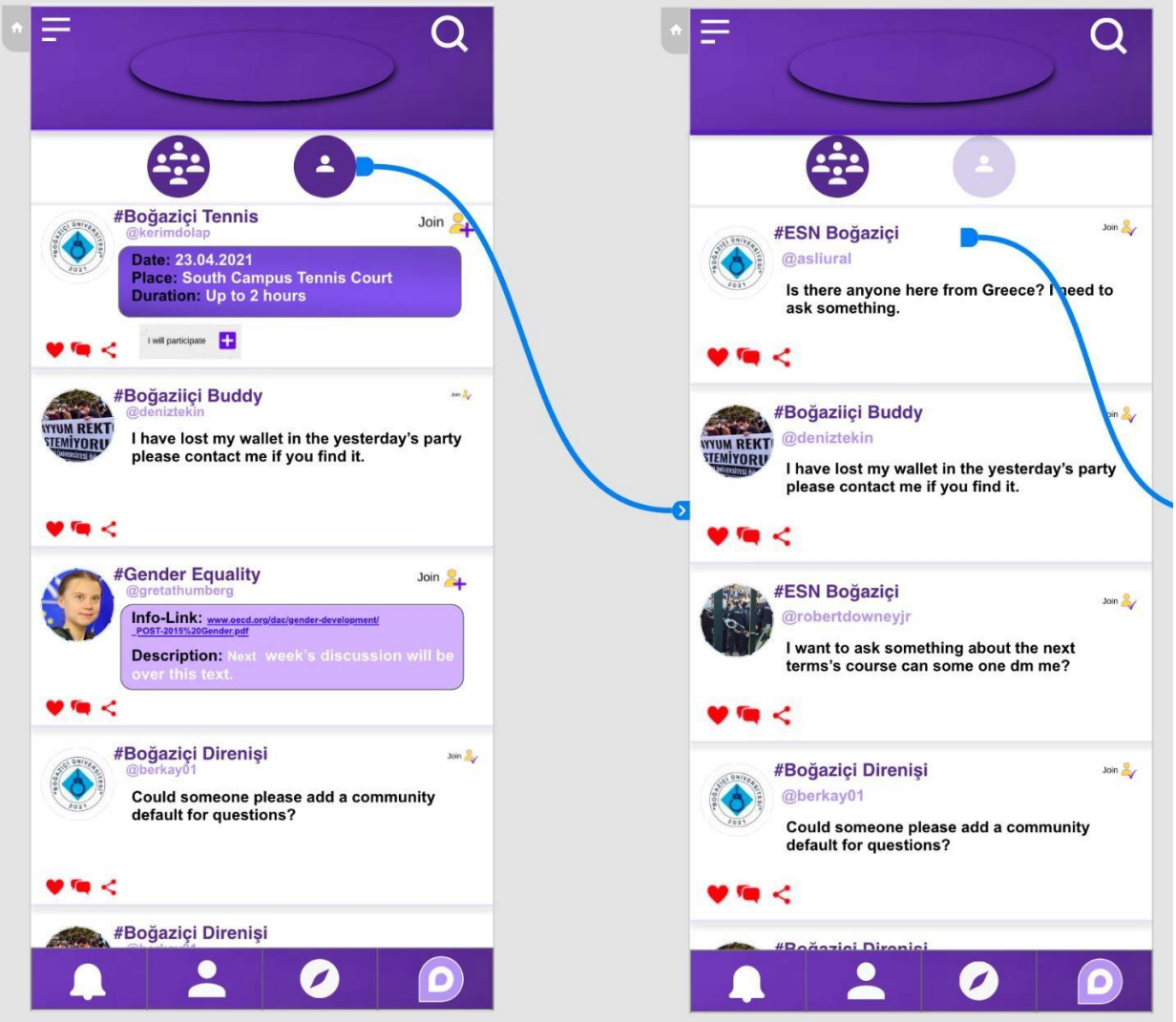
- Mabel has already signed up to the application.
- Mabel has already logged in to the application.
- Mabel previously had subscribed to Bogazici ESN, Boğaziçi Buddy and Boğaziçi Direnişi Communities.

Acceptance Criteria

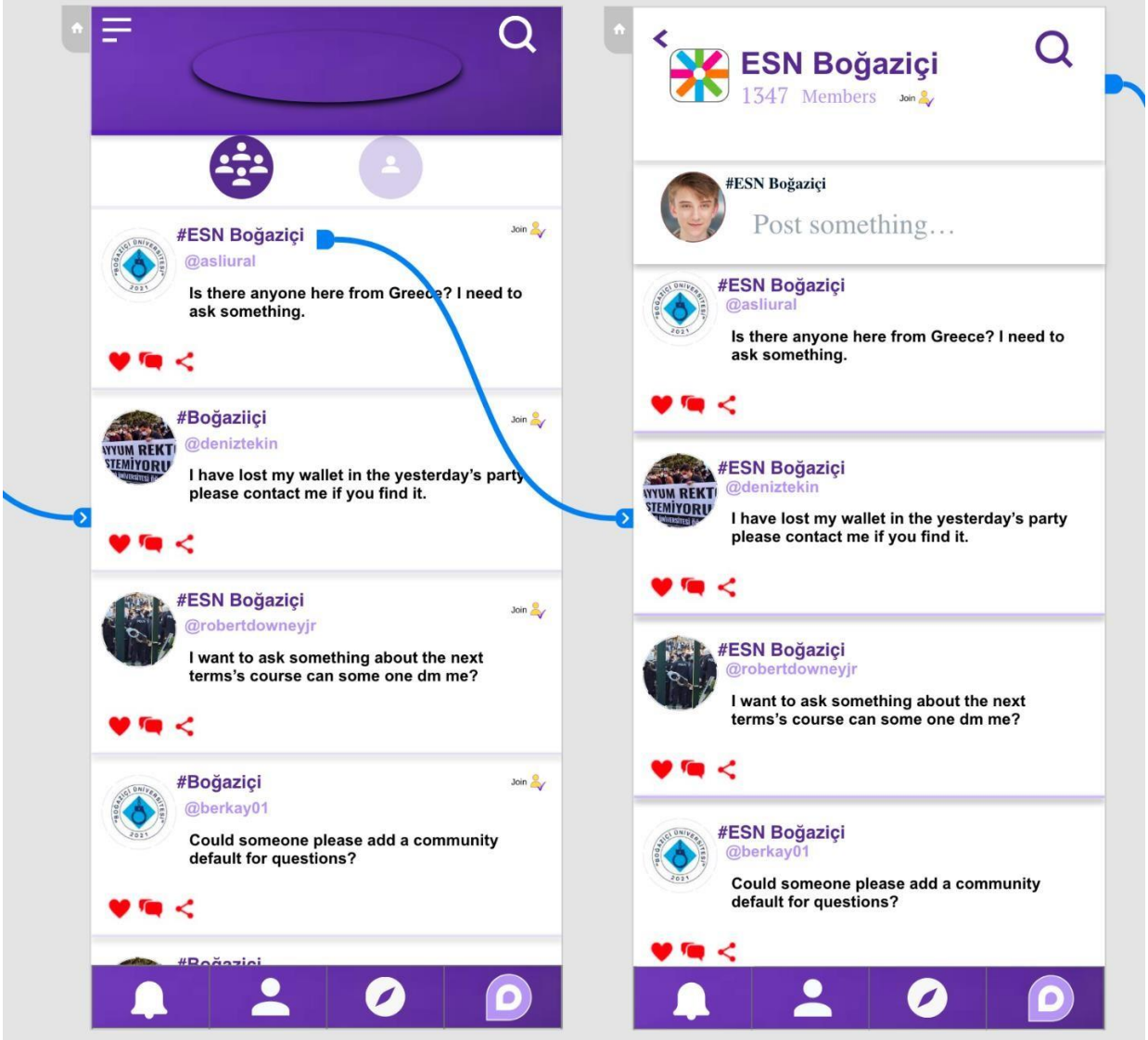
- 1.1.4.2.1. Community members should be able to contribute to content in a Community within the predetermined ranges that Community Creator has ruled.
- 1.1.4.2.3. Community members shall be able to view Community Posts and Events.
- 1.2.1.1. System shall be able to recommend content and persons of interest.
- 1.2.3.3. According to the community privacy options users having the right to see the posts shall be authorized accordingly.
- 1.2.4.1. Posts, shared by authorized members shall be displayed on the community timeline.
- 1.2.4.3. Posts shall have comment section.
- 1.2.5.1. Upon searching related people and posts shall be indexed.
- 1.2.5.2. Upon searching, search history will be displayed.
- 1.2.6.1. System shall provide search bar, tabs and community posts.
- 1.2.6.2. System should provide community posts that users are registered.

Scenario with Mockups

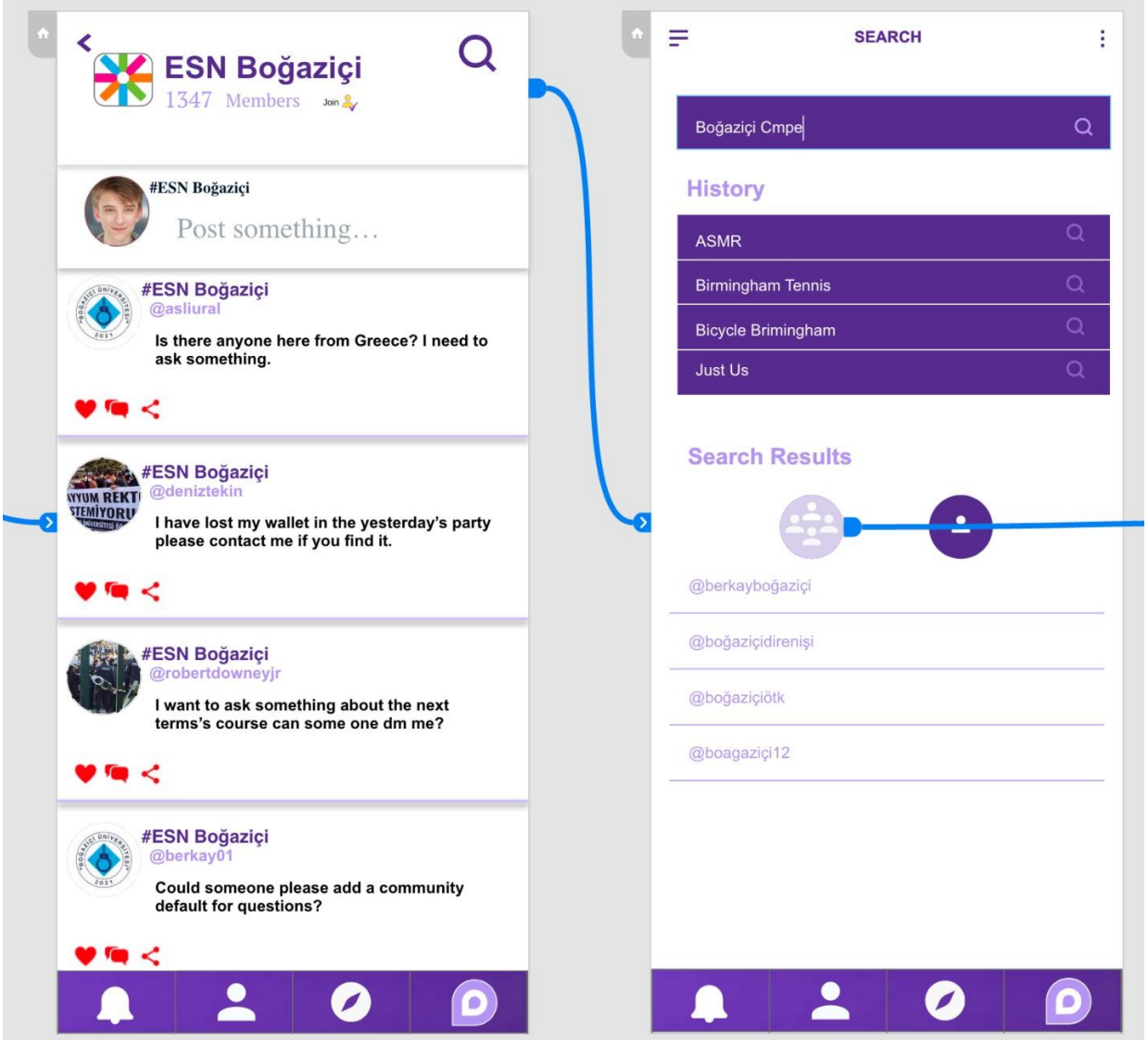
- She opened the app and scrolled down on the feed. She only wanted to see community posts, so she chose community filtering option.



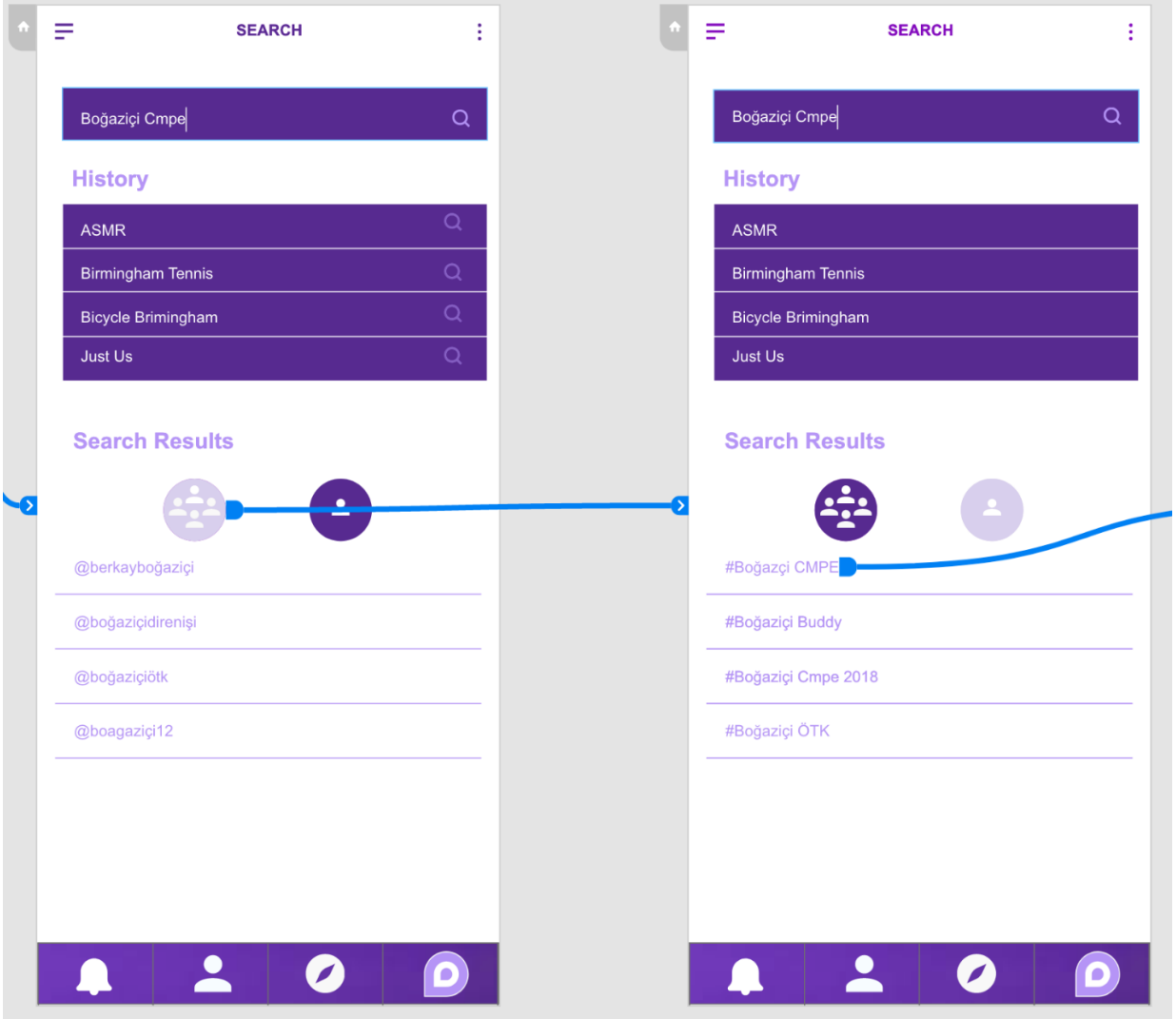
- She saw #ESN Bogazici post on the feed and went to community's page by clicking on the community name above the post.



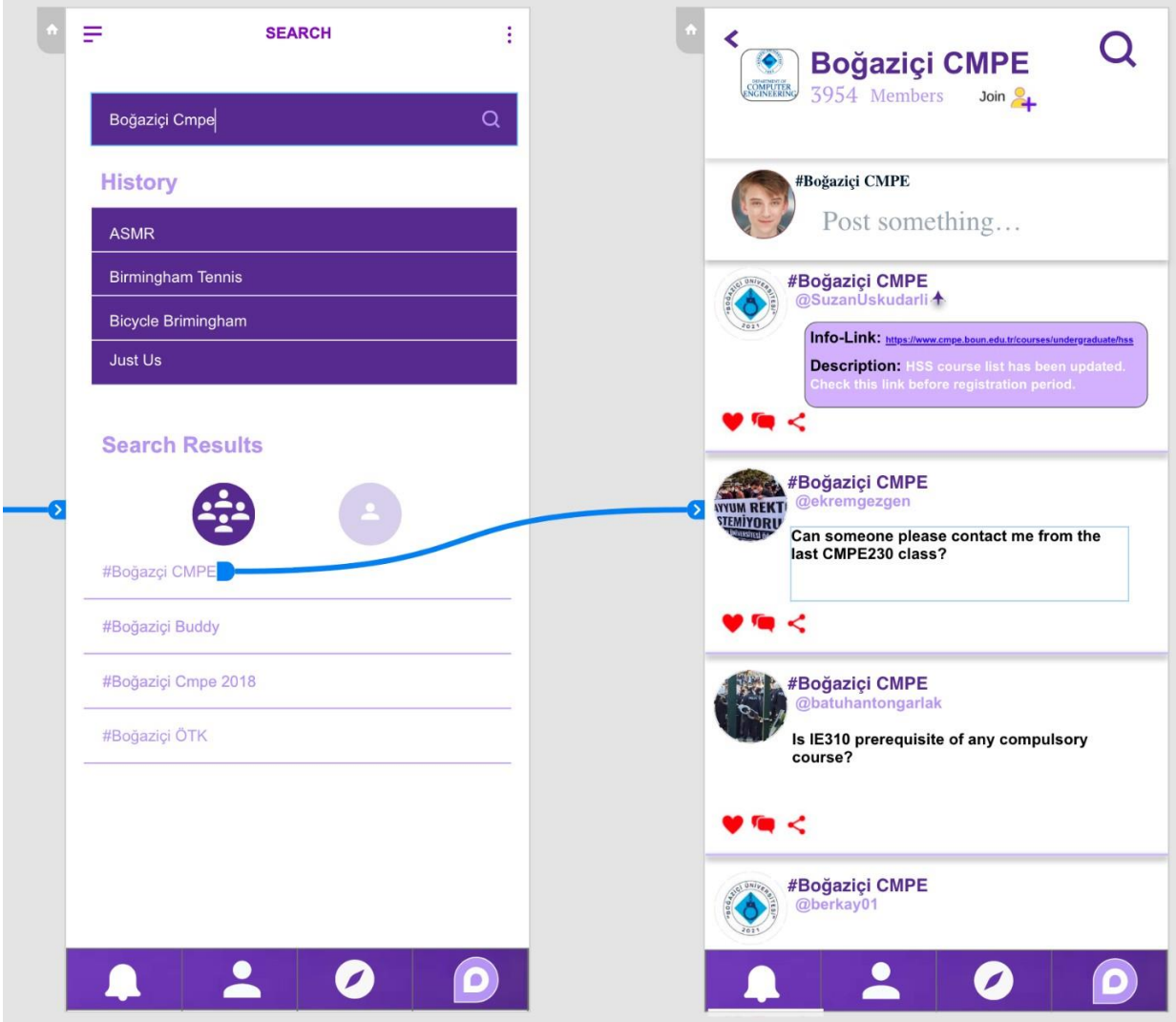
- She wanted to find someone in her department but there was not any post related to Cmpe or posted by someone in the department in ESN Community. In order to find the department's community, she clicked on the search button and typed 'Bogazici Cmpe' in the search box.



- At first, user profiles appeared in the search page as default. Below the search history, she saw the filtering option just as on the feed for community and user profiles.



- She chose community filter and found Bogazici Cmpe Community.


SEARCH
Boğaziçi Cmpe
History
ASMR
Birmingham Tennis
Bicycle Brimingham
Just Us
Search Results
#Boğazçi CMPE
#Boğaziçi Buddy
#Boğaziçi Cmpe 2018
#Boğaziçi ÖTK
Boğaziçi CMPE
3954 Members
Join
#Boğaziçi CMPE
Post something…
#Boğaziçi CMPE
@SuzanUskudarli
Info-Link: https://www.cmpe.boun.edu.tr/courses/undergraduate/hss
Description: HSS course list has been updated. Check this link before registration period.
#Boğaziçi CMPE
@ekremgezgen
Can someone please contact me from the last CMPE230 class?
#Boğaziçi CMPE
@batuhantongarlak
Is IE310 prerequisite of any compulsory course?
#Boğaziçi CMPE
@berkay01

6.3.4 Scenario 4

Persona

- Maria Sharapova
- She is 34 years old.
- She is a professional tennis player.

Story

- She came to Antalya for a tournament.
- After the tournament, she wants to stay there more and organize tennis events with talented young people who are interested in tennis.
- Since she dreams of settling in this city and become a tennis coach after her retirement, she wants to meet new people through these events.

Goals

- She wants to find new people to play tennis with by creating new community post type, tennis event.

Preconditions

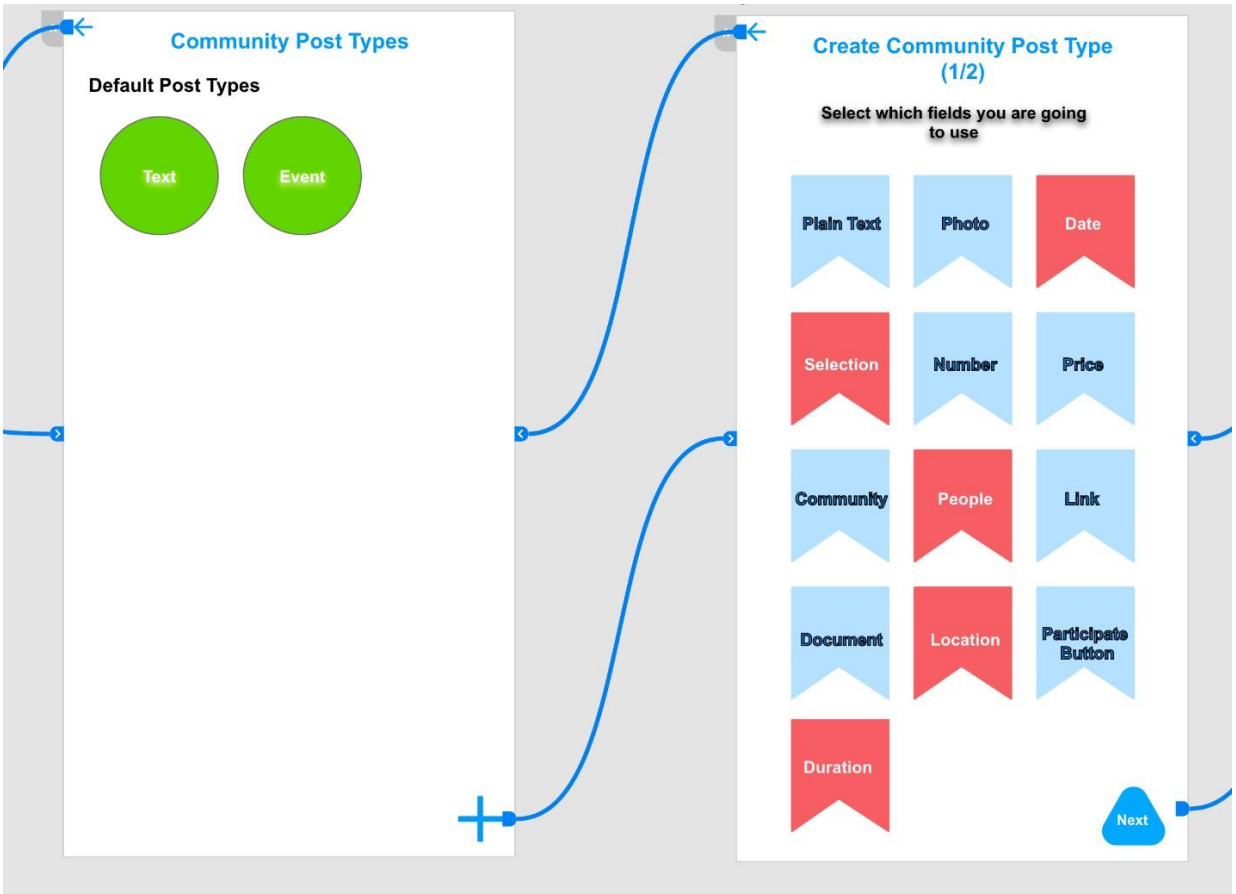
- She has already signed up to the application.
- She has already logged in to the application.
- She is the admin of Antalya Tennis Community.
- She has clicked settings of post types of Antalya Tennis Community.

Acceptance Criteria

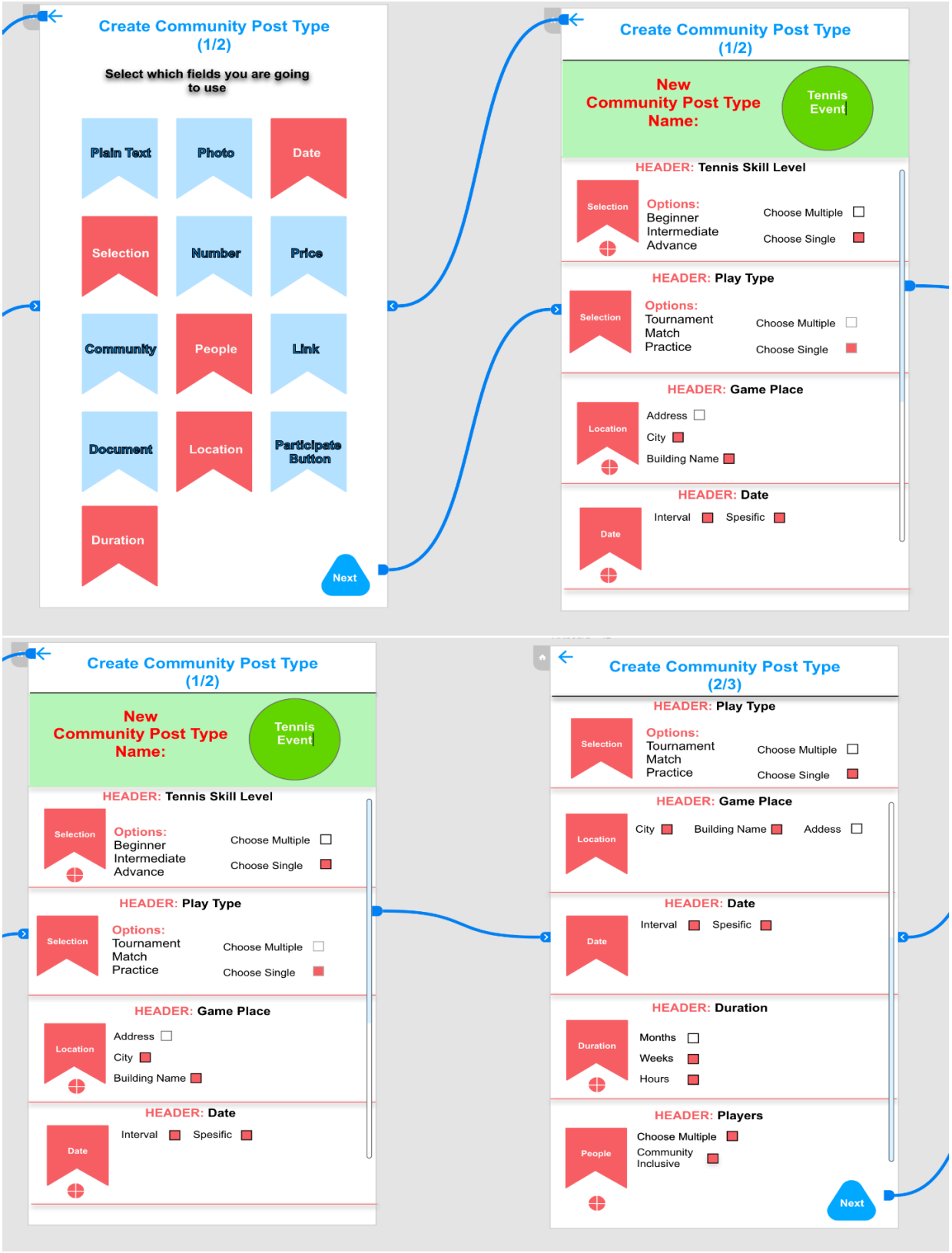
- 1.1.2.1. Registered users shall be able to sign in via username/password or e-mail/password.
- 1.1.4.3.10. Community admins shall be able to create Community Post Types for the purpose of the Community.

Scenario with Mockups

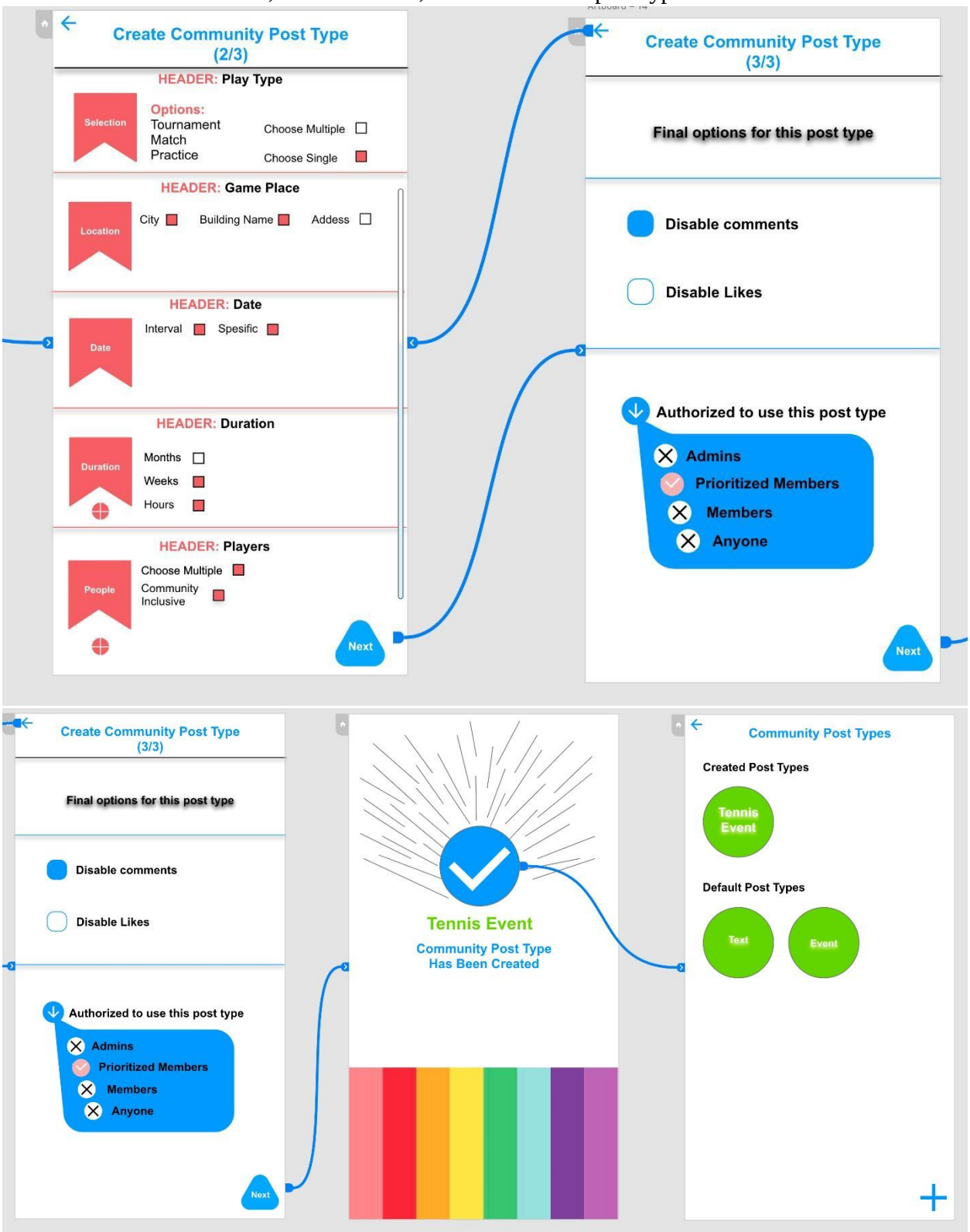
- She is on the community post types page in the Antalya Tennis Community. She clicks "+" button on the bottom right and the button navigates her to create post type page (1/2). On this page, there are default options to select. She selects the features that she wants it to be in the post type she will create.
- She is on the community post types page in the Antalya Tennis Community. She clicks "+" button on the bottom right and the button navigates her to create post type page (1/2). On this page, there are default options to select. She selects the features that she wants it to be in the post type she will create.



- On the page (2/2), she customizes the features she has selected.

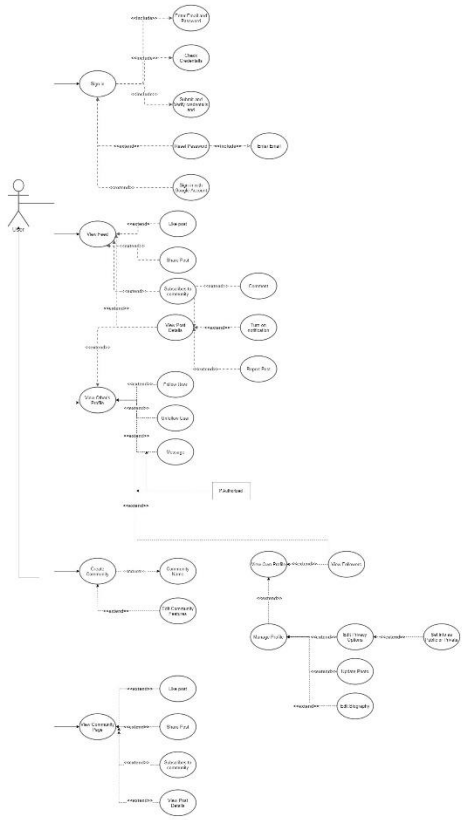

Create Community Post Type
(1/2)
Select which fields you are going to use
Plain Text
Photo
Date
Selection
Number
Price
Community
People
Link
Document
Location
Participate Button
Duration
Next
Create Community Post Type
(1/2)
New Community Post Type Name:
Tennis Event
HEADER: Tennis Skill Level
Selection
Options:
Beginner
Intermediate
Advance
Choose Multiple
Choose Single
HEADER: Play Type
Selection
Options:
Tournament
Match
Practice
Choose Multiple
Choose Single
HEADER: Game Place
Location
Address
City
Building Name
HEADER: Date
Date
Interval
Spesific
Create Community Post Type
(1/2)
New Community Post Type Name:
Tennis Event
HEADER: Tennis Skill Level
Selection
Options:
Beginner
Intermediate
Advance
Choose Multiple
Choose Single
HEADER: Play Type
Selection
Options:
Tournament
Match
Practice
Choose Multiple
Choose Single
HEADER: Game Place
Location
Address
City
Building Name
HEADER: Date
Date
Interval
Spesific
Create Community Post Type
(2/3)
HEADER: Play Type
Selection
Options:
Tournament
Match
Practice
Choose Multiple
Choose Single
HEADER: Game Place
Location
City
Building Name
Addess
HEADER: Date
Date
Interval
Spesific
HEADER: Duration
Duration
Months
Weeks
Hours
HEADER: Players
People
Choose Multiple
Community Inclusive
Next

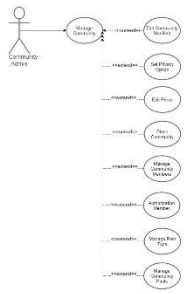
- She clicks "OK" button on the bottom right and the button navigates her to community post types page. There are 3 different post types, 2 are under default post types, and the one she has created, "tennis event", is under created post types.

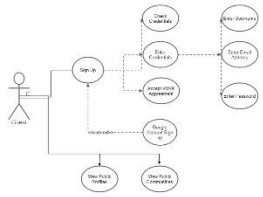


6.3 Software Design documents in UML

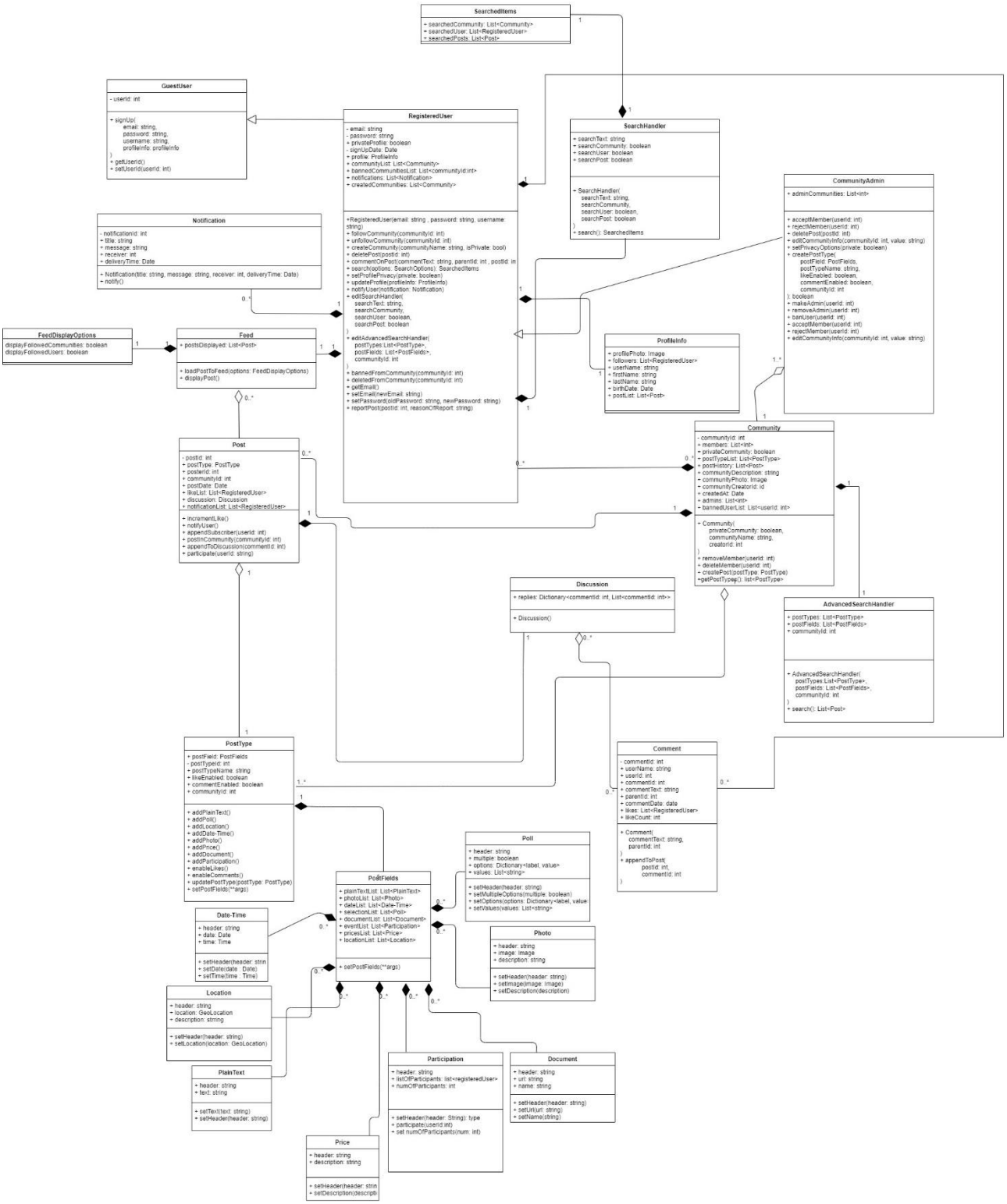
6.4.1 Use Case Diagram





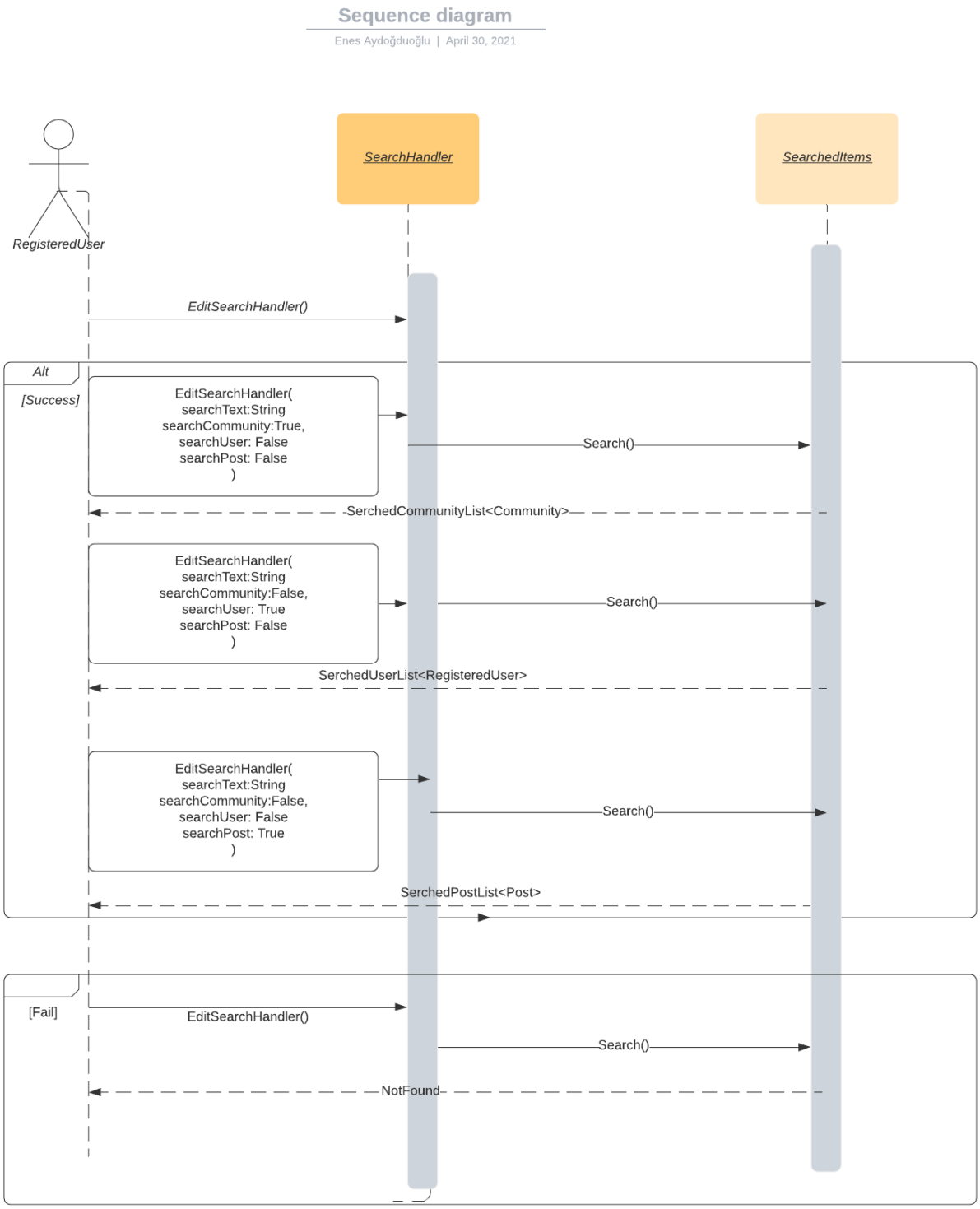


6.4.2 Class Diagram

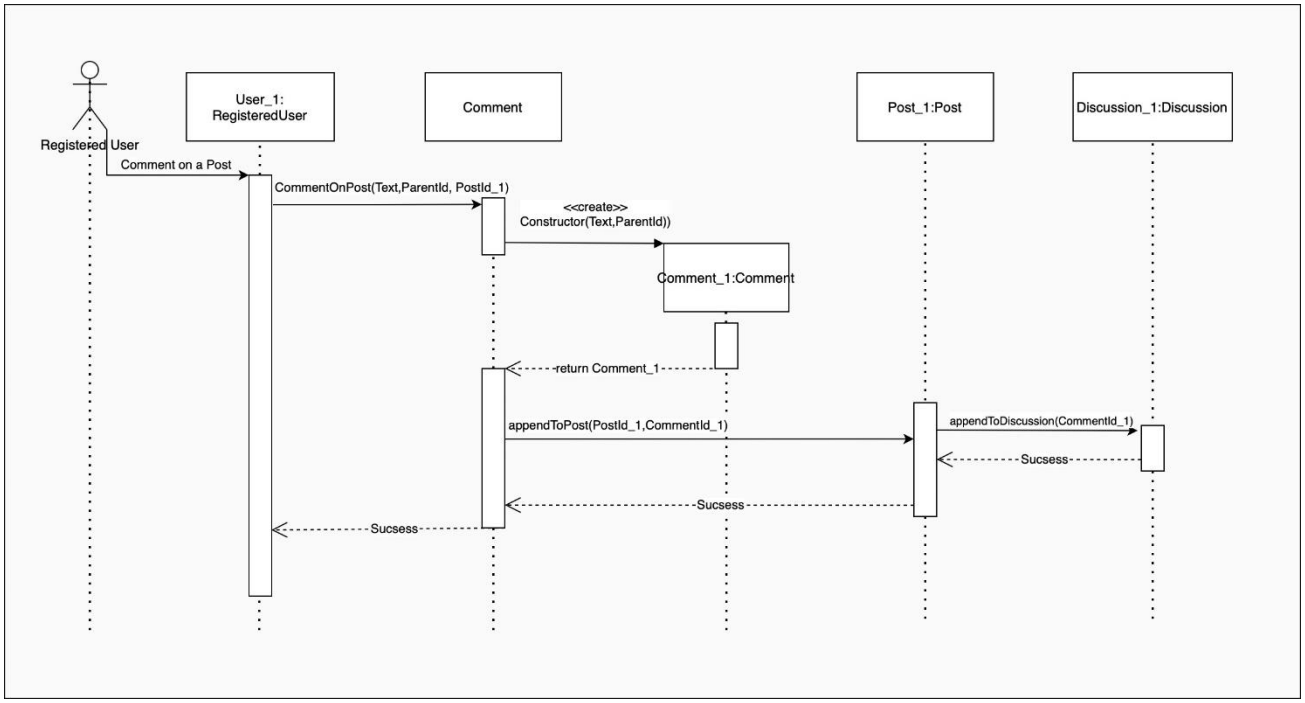


6.4.3 Sequence Diagrams

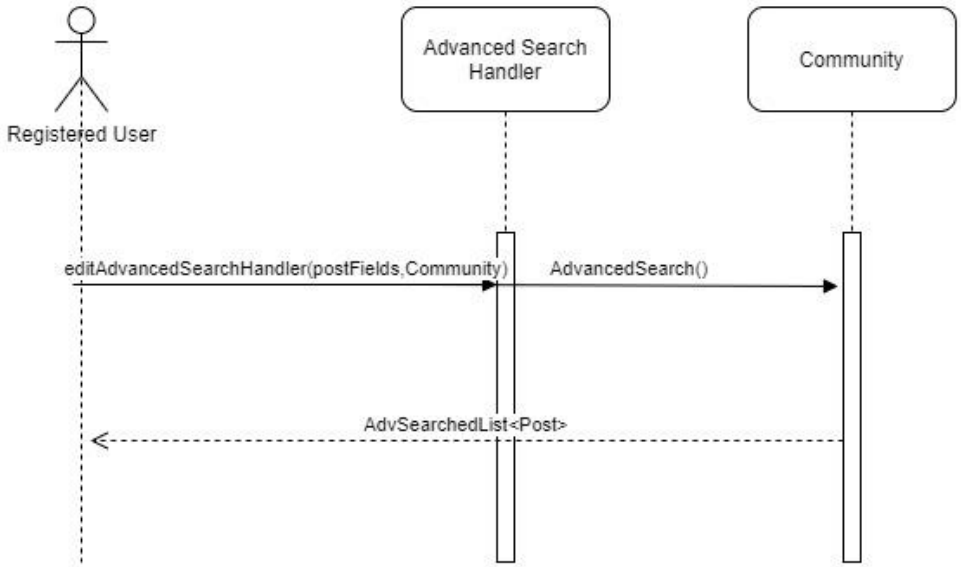
6.4.3.1 Search



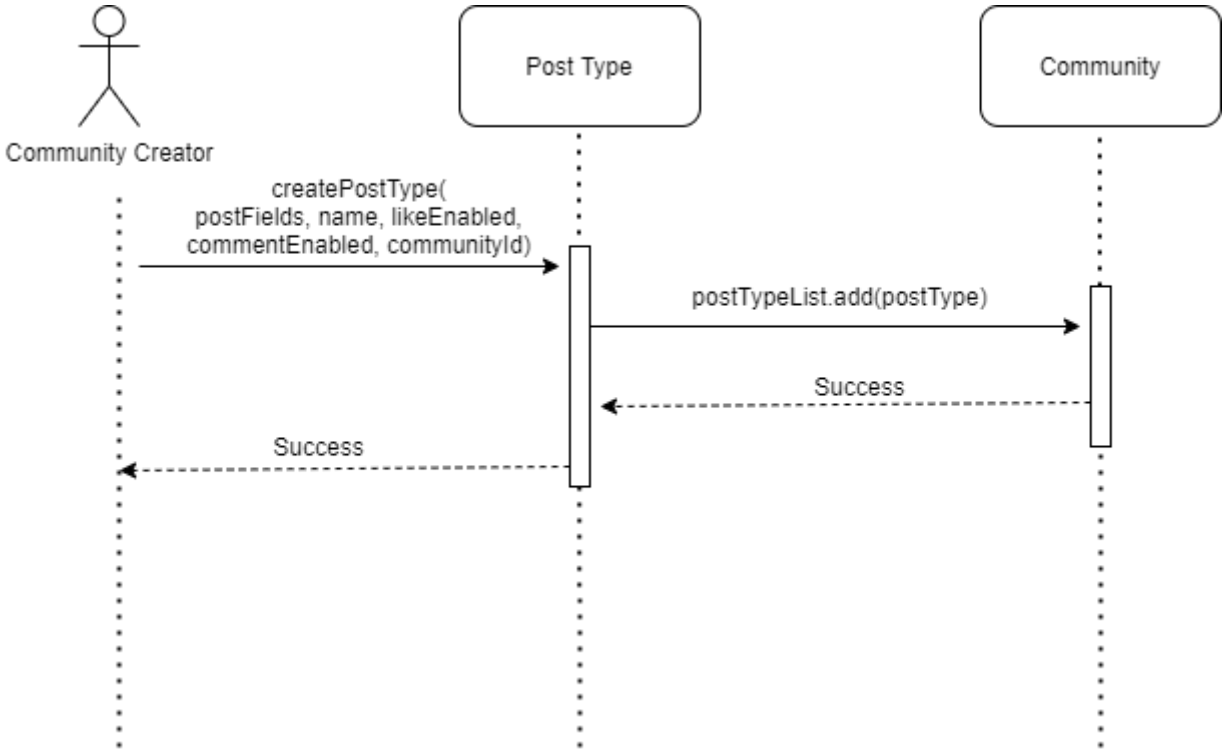
6.4.3.2 Comment On A Post



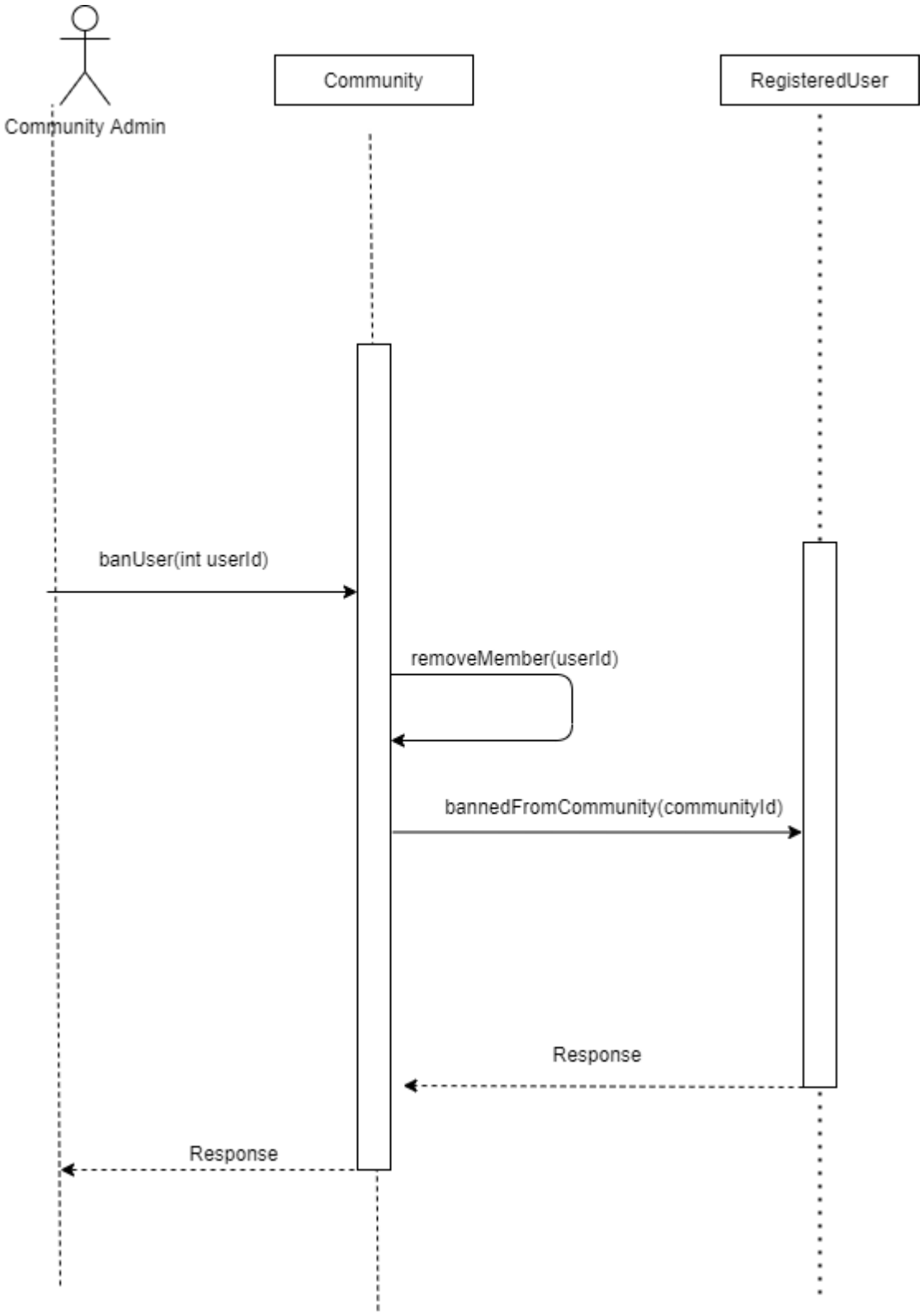
6.4.3.3 Advanced Search



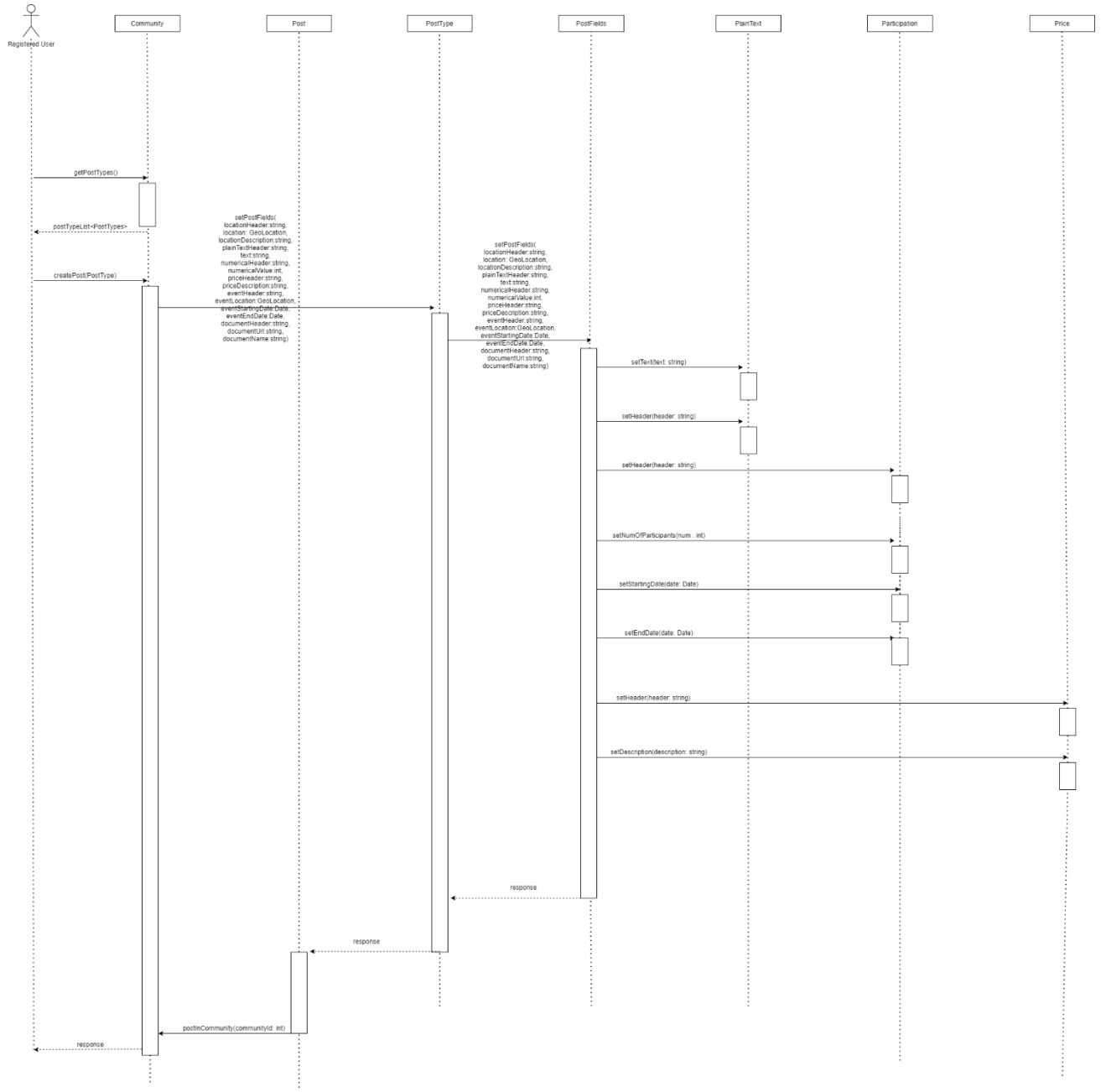
6.4.3.4 Create Post Type



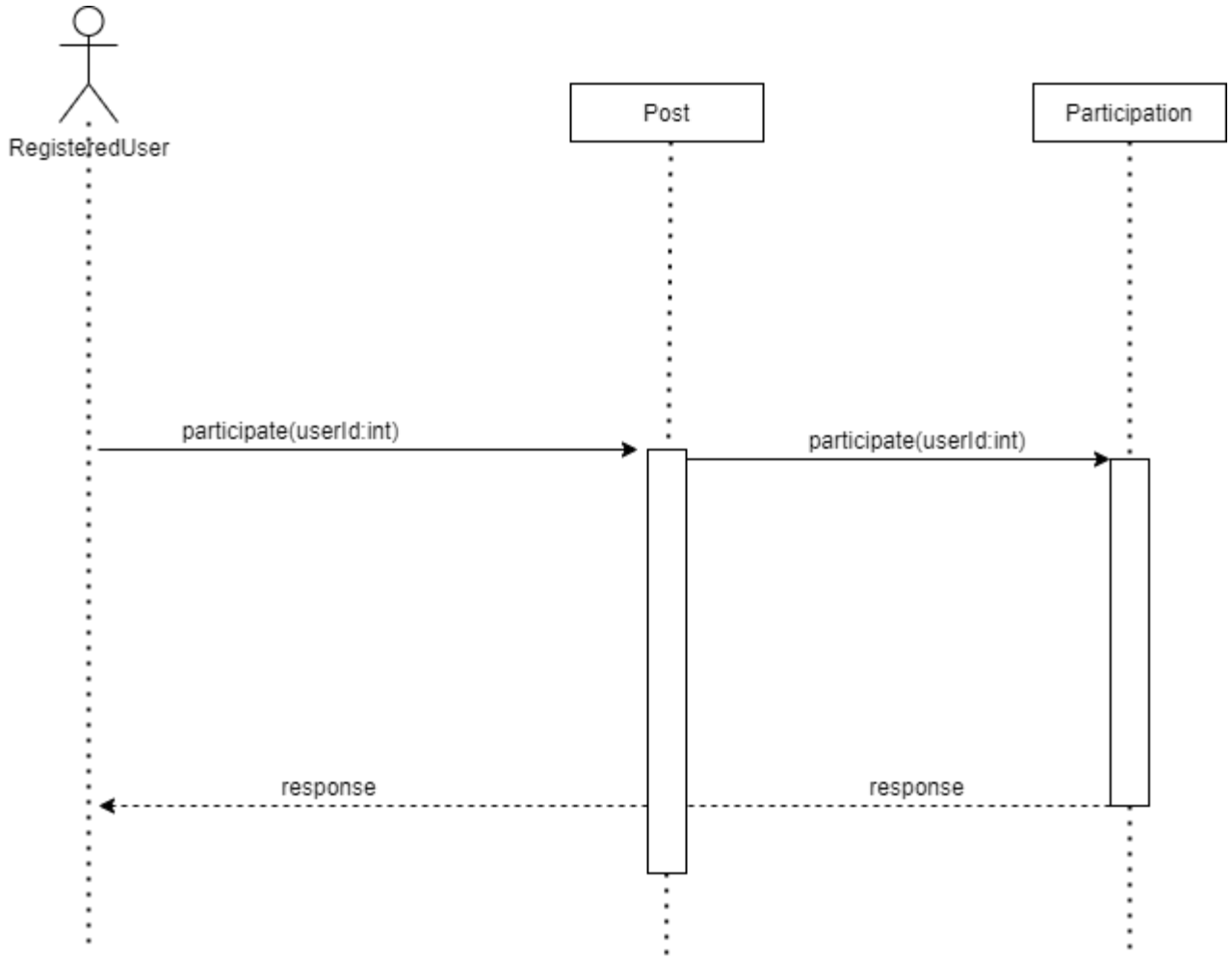
6.4.3.5 Block A Member From Community



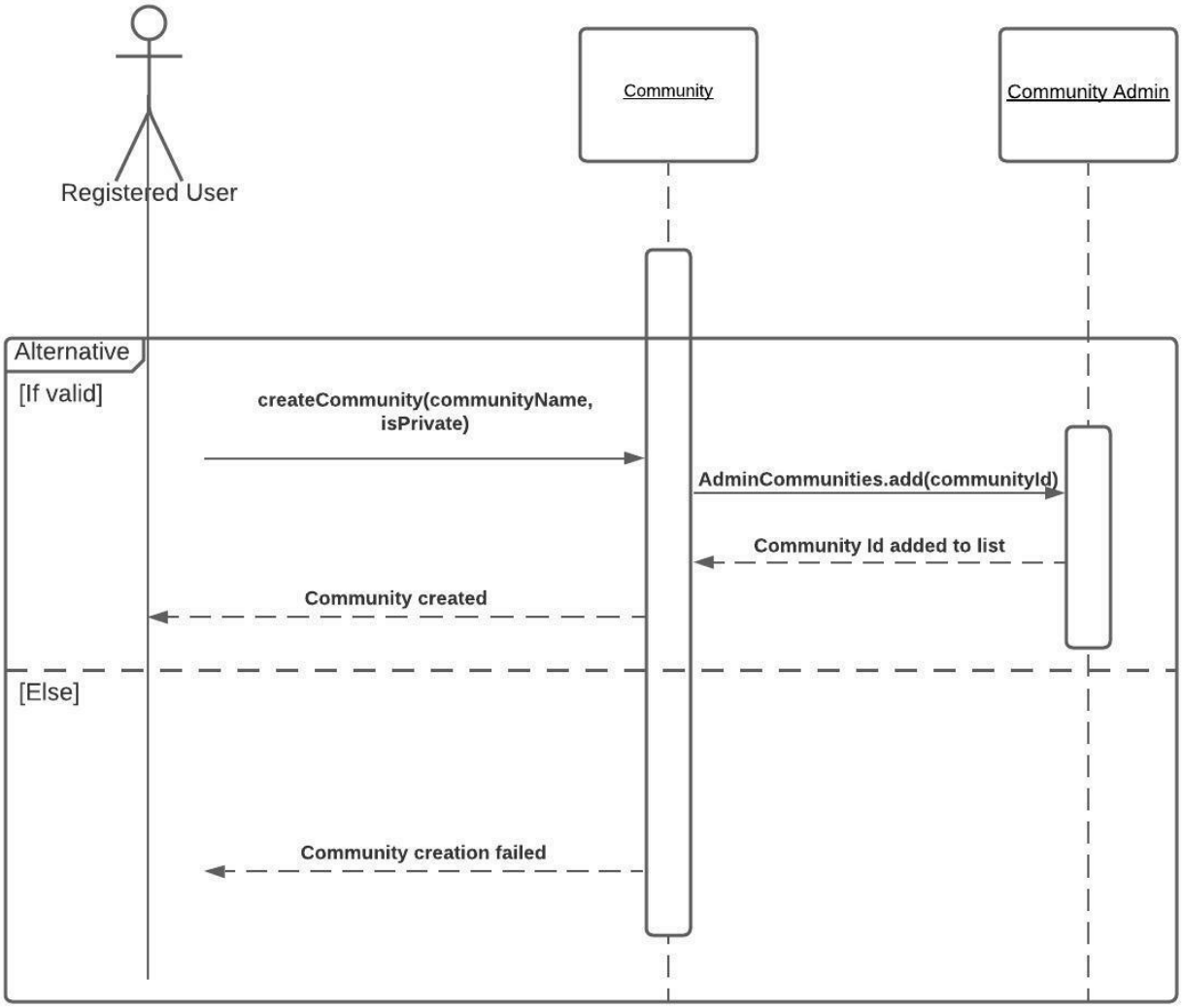
6.4.3.6 Create A Post According To A Post Type



6.3.4.8 Interact With A Post



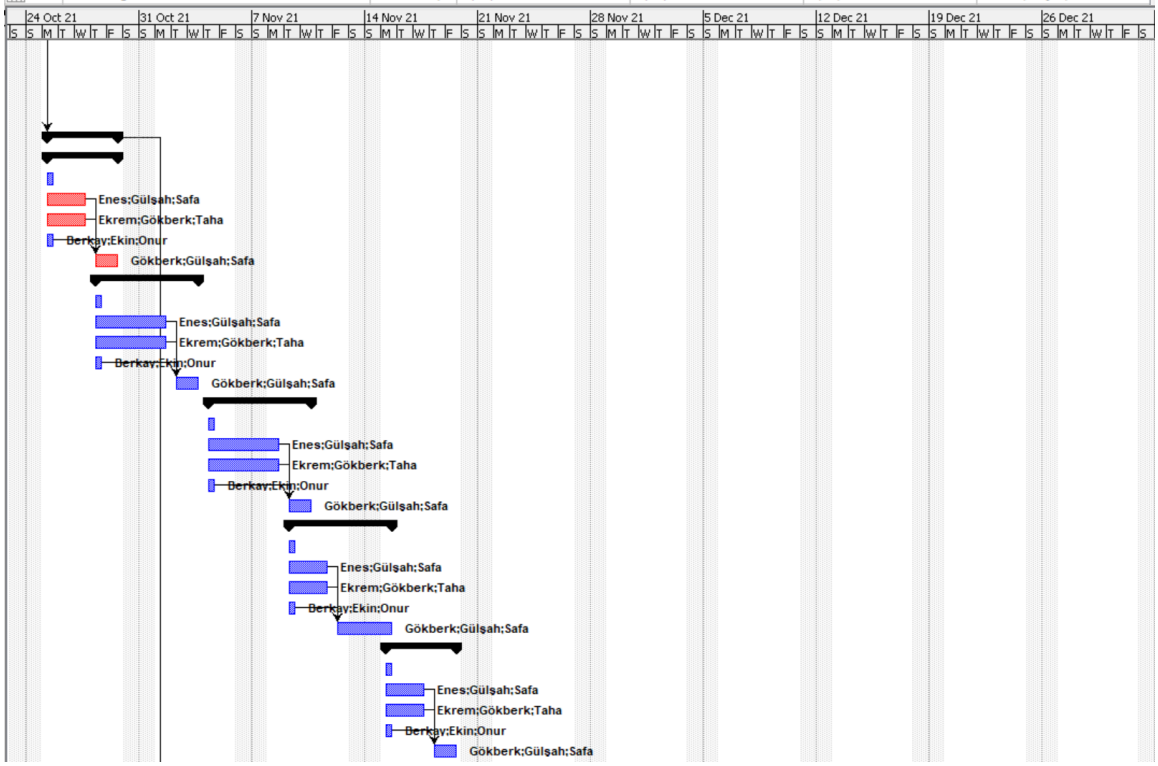
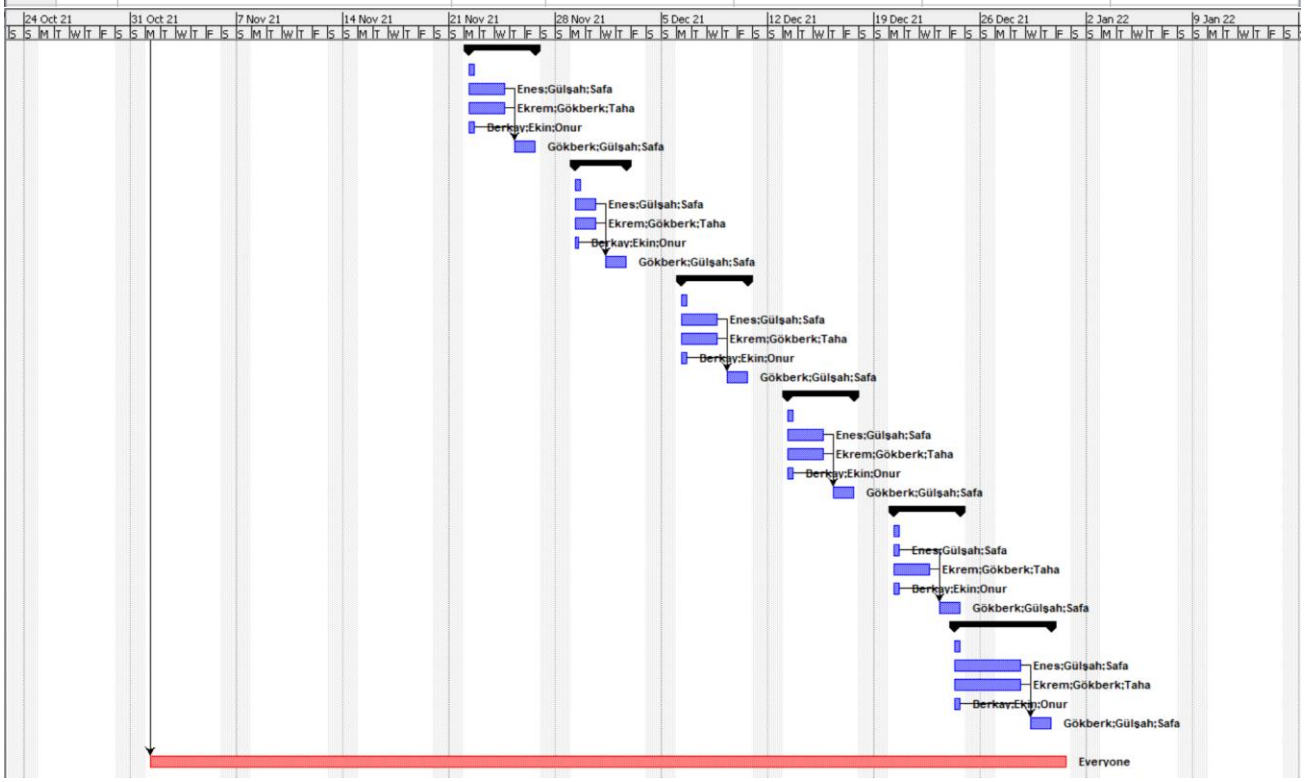
6.3.4.9 Create Community



6.4 Project plan & RAM

6.4.1 Project Plan

Preparing Deliverables for Final Milestone	5 days?	**10/25/21 8:00 AM**	**10/29/21 5:00 PM**	**1**	
Implementation of Login Page	5 days?	**10/25/21 8:00 AM**	**10/29/21 5:00 PM**		
Deciding Login Page Features	1 day?	10/25/21 8:00 AM	10/25/21 5:00 PM		
Android Implementation	3 days?	10/25/21 8:00 AM	10/27/21 5:00 PM		Enes;Gülşah;Safa
Front-End Implementation	3 days?	10/25/21 8:00 AM	10/27/21 5:00 PM		Ekrem;Gökberk;Taha
Back-End Implementation	1 day?	10/25/21 8:00 AM	10/25/21 5:00 PM		Berkay;Ekin;Onur
Testing	2 days?	10/28/21 8:00 AM	10/29/21 5:00 PM	73;74;75	Gökberk;Gülşah;Safa
Implementation of Feed	5 days?	**10/28/21 8:00 AM**	**11/3/21 5:00 PM**		
Deciding Feed Features	1 day?	10/28/21 8:00 AM	10/28/21 5:00 PM		
Android Implementation	3 days?	10/28/21 8:00 AM	11/1/21 5:00 PM		Enes;Gülşah;Safa
Front-End Implementation	3 days?	10/28/21 8:00 AM	11/1/21 5:00 PM		Ekrem;Gökberk;Taha
Back-End Implementation	1 day?	10/28/21 8:00 AM	10/28/21 5:00 PM		Berkay;Ekin;Onur
Testing	2 days?	11/2/21 8:00 AM	11/3/21 5:00 PM	80;79;81	Gökberk;Gülşah;Safa
Implementation of Community Page	5 days?	**11/4/21 8:00 AM**	**11/10/21 5:00 PM**		
Deciding Community Page Features	1 day?	11/4/21 8:00 AM	11/4/21 5:00 PM		
Android Implementation	3 days?	11/4/21 8:00 AM	11/8/21 5:00 PM		Enes;Gülşah;Safa
Front-End Implementation	3 days?	11/4/21 8:00 AM	11/8/21 5:00 PM		Ekrem;Gökberk;Taha
Back-End Implementation	1 day?	11/4/21 8:00 AM	11/4/21 5:00 PM		Berkay;Ekin;Onur
Testing	2 days?	11/9/21 8:00 AM	11/10/21 5:00 PM	86;87;85	Gökberk;Gülşah;Safa
Implementation of User Profile Page	5 days?	**11/9/21 8:00 AM**	**11/15/21 5:00 PM**		
Deciding User Profile Page Features	1 day?	11/9/21 8:00 AM	11/9/21 5:00 PM		
Android Implementation	3 days?	11/9/21 8:00 AM	11/11/21 5:00 PM		Enes;Gülşah;Safa
Front-End Implementation	3 days?	11/9/21 8:00 AM	11/11/21 5:00 PM		Ekrem;Gökberk;Taha
Back-End Implementation	1 day?	11/9/21 8:00 AM	11/9/21 5:00 PM		Berkay;Ekin;Onur
Testing	2 days?	11/12/21 8:00 AM	11/15/21 5:00 PM	93;91;92	Gökberk;Gülşah;Safa
Implementation of Community Creation Page	5 days?	**11/15/21 8:00 AM**	**11/19/21 5:00 PM**		
Deciding Community Creation Page Features	1 day?	11/15/21 8:00 AM	11/15/21 5:00 PM		
Android Implementation	3 days?	11/15/21 8:00 AM	11/17/21 5:00 PM		Enes;Gülşah;Safa
Front-End Implementation	3 days?	11/15/21 8:00 AM	11/17/21 5:00 PM		Ekrem;Gökberk;Taha
Back-End Implementation	1 day?	11/15/21 8:00 AM	11/15/21 5:00 PM		Berkay;Ekin;Onur
Testing	2 days?	11/18/21 8:00 AM	11/19/21 5:00 PM	99;97;98	Gökberk;Gülşah;Safa



101	**Implementation of Post Type Creation Page**	5 days?	**11/22/21 8:00 AM**	**11/26/21 5:00 PM**		
102	Implementation of Post Type Creation Page	1 day?	11/22/21 8:00 AM	11/22/21 5:00 PM		
103	Android Implementation	3 days?	11/22/21 8:00 AM	11/24/21 5:00 PM		Enes;Gülşah;Safa
104	Front-End Implementation	3 days?	11/22/21 8:00 AM	11/24/21 5:00 PM		Ekrem;Gökberk;Taha
105	Back-End Implementation	1 day?	11/22/21 8:00 AM	11/22/21 5:00 PM		Berkay;Ekin;Onur
106	Testing	2 days?	11/25/21 8:00 AM	11/26/21 5:00 PM	105;103;104	Gökberk;Gülşah;Safa
107	**Implementation of Search Page**	4 days?	**11/29/21 8:00 AM**	**12/2/21 5:00 PM**		
108	Deciding Search Page Features	1 day?	11/29/21 8:00 AM	11/29/21 5:00 PM		
109	Android Implementation	2 days?	11/29/21 8:00 AM	11/30/21 5:00 PM		Enes;Gülşah;Safa
110	Front-End Implementation	2 days?	11/29/21 8:00 AM	11/30/21 5:00 PM		Ekrem;Gökberk;Taha
111	Back-End Implementation	0.667 days?	11/29/21 8:00 AM	11/29/21 2:20 PM		Berkay;Ekin;Onur
112	Testing	2 days?	12/1/21 8:00 AM	12/2/21 5:00 PM	111;110;109	Gökberk;Gülşah;Safa
113	**Implementation of Message Page**	5 days?	**12/6/21 8:00 AM**	**12/10/21 5:00 PM**		
114	Deciding Message Page Features	1 day?	12/6/21 8:00 AM	12/6/21 5:00 PM		
115	Android Implementation	3 days?	12/6/21 8:00 AM	12/8/21 5:00 PM		Enes;Gülşah;Safa
116	Front-End Implementation	3 days?	12/6/21 8:00 AM	12/8/21 5:00 PM		Ekrem;Gökberk;Taha
117	Back-End Implementation	1 day?	12/6/21 8:00 AM	12/6/21 5:00 PM		Berkay;Ekin;Onur
118	Testing	2 days?	12/9/21 8:00 AM	12/10/21 5:00 PM	115;116;117	Gökberk;Gülşah;Safa
119	**Implementation of Post Details**	5 days?	**12/13/21 8:00 AM**	**12/17/21 5:00 PM**		
120	Deciding Post DetailsFeatures	1 day?	12/13/21 8:00 AM	12/13/21 5:00 PM		
121	Android Implementation	3 days?	12/13/21 8:00 AM	12/15/21 5:00 PM		Enes;Gülşah;Safa
122	Front-End Implementation	3 days?	12/13/21 8:00 AM	12/15/21 5:00 PM		Ekrem;Gökberk;Taha
123	Back-End Implementation	1 day?	12/13/21 8:00 AM	12/13/21 5:00 PM		Berkay;Ekin;Onur
124	Testing	2 days?	12/16/21 8:00 AM	12/17/21 5:00 PM	122;123;121	Gökberk;Gülşah;Safa
125	**Implementation of Admin Panel**	5 days?	**12/20/21 8:00 AM**	**12/24/21 5:00 PM**		
126	Deciding Admin Panel Features	1 day?	12/20/21 8:00 AM	12/20/21 5:00 PM		
127	Android Implementation	1 day?	12/20/21 8:00 AM	12/20/21 5:00 PM		Enes;Gülşah;Safa
128	Front-End Implementation	3 days?	12/20/21 8:00 AM	12/22/21 5:00 PM		Ekrem;Gökberk;Taha
129	Back-End Implementation	1 day?	12/20/21 8:00 AM	12/20/21 5:00 PM		Berkay;Ekin;Onur
130	Testing	2 days?	12/23/21 8:00 AM	12/24/21 5:00 PM	129;127;128	Gökberk;Gülşah;Safa
131	**Implementation of Notification Page**	5 days?	**12/24/21 8:00 AM**	**12/30/21 5:00 PM**		
132	Deciding Notification Page Features	1 day?	12/24/21 8:00 AM	12/24/21 5:00 PM		
133	Android Implementation	3 days?	12/24/21 8:00 AM	12/28/21 5:00 PM		Enes;Gülşah;Safa
134	Front-End Implementation	3 days?	12/24/21 8:00 AM	12/28/21 5:00 PM		Ekrem;Gökberk;Taha
135	Back-End Implementation	1 day?	12/24/21 8:00 AM	12/24/21 5:00 PM		Berkay;Ekin;Onur
136	Testing	2 days?	12/29/21 8:00 AM	12/30/21 5:00 PM	135;134;133	Gökberk;Gülşah;Safa
137	End-to-end Testing	5 days?	7/22/21 8:00 AM	7/28/21 5:00 PM		
138	Preparing Deliverables for Milestone 3	45 days?	11/1/21 8:00 AM	12/31/21 5:00 PM	70	Everyone