

CMPE 451

Milestone 1 Report

Group 9

Mert Alkan

Ahmet Melih Aydoğdu

Zehranaz Canfes

Mustafa Emir Çolak

Ömer Barış Erkek

Melih Özcan

Ahmet İbrahim Şentürk

Niyazi Ülke

Table of Contents

Table of Contents

Executive Summary

- Introduction
- Project Status and Work Done So Far
- Moving Forward

List and Status of Deliverables

Evaluation of the status of deliverables and its impact on our project plan

- Home Page
 - Create Post Page
 - Edit Post Page
 - Deployment
- Satisfaction of Requirements**
- Summary
 - List of addressed requirements
 - Partially satisfied requirements
 - Explanations
 - Satisfied Requirements
 - List of unaddressed requirements
 - Unaddressed Functional Requirements
 - Unaddressed Non-Functional Requirements

Evaluation of Tools and Processes

- Evaluation of Tools
 - General
 - Discord
 - Whatsapp
 - Git
 - GitHub
 - Backend
 - AWS
 - Amazon S3
 - Django
 - Docker
 - MongoDB
 - Visual Studio Code
 - Postman
 - Google Maps
 - Android
 - IntelliJ IDEA
 - Gradle
 - Java and Native Android SDK
 - Frontend
 - Framework
 - IDE
 - Chrome DevTools
 - UI Framework

Evaluation of Processes

- Team Meetings
- LAB Meetings
- Issue Creation
- Code Reviews

Project Plan

Work Done by Each Team Member

Requirements

- Glossary

1. Functional Requirements
2. Non-Functional Requirements

Use Case Diagram

Class Diagram

Sequence Diagrams

1. View a User's Profile
2. View Posts on Homepage
3. Report a Post
4. Sign up
5. Sign in
6. Comment
7. Follow A User
8. Search a User
9. Search Posts Containing a Location
10. Create Post

Scenarios and Mockups

Creating Posts Scenario & Mockup

- Persona
- Preconditions
- Goals
- Acceptance Criteria
- Scenario and Mockups
- References

Viewing Posts Scenario & Mockup

- Persona
- Preconditions
- Goals
- Acceptance Criteria
- Scenario
- Mockups
- Homepage
- Post Details
- Reporting
- Homepage
- Post Details
- Reporting
- References

Searching and Filtering Scenario and Mockup

- Persona
- Preconditions
- Acceptance Criteria
- Goals
- Scenario
- Mockups

Project Plan

Communication Plan

Executive Summary

Introduction

As a group in CmpE 451 course, we aim to design a platform for people who want to share their stories based on locations they have been to and experienced. We are to implement a web and an Android application for our users to share stories with others and view them.

In this milestone, we implemented a web, backend and mobile for the start of our app. We were split into 3 teams, web and backend containing 3, mobile containing 2 people. After that, technologies to be used were decided by us. Then, we arranged meetings to be on the same page. We decided the main structure of our app such as design, data flow, connection of team's projects etc. And each team started to work on their project. We had a few flaws but with meetings and messaging each other, we overcome those conflicts. At the end we had a simple basis for our project. Then we deployed our app to Amazon EC2. Then a workflow was created to simplify the process. After these steps, we were to create a customer presentation. Again in a meeting, scenarios and flow of the presentation were decided.

Project Status and Work Done So Far

In the implementation of the project, the project plan was followed. The aim of the milestone 1 was establishing the connection between backend and frontend/android and also, creating the home page for our application. The posts could be created and showed on the home page and they can also be edited. We successfully fulfilled our goals for milestone 1. Each team worked collaboratively to achieve this goals. The work done until this point can be explained in each team's point of view.

The frontend and android teams created home pages and implemented create post and edit post pages. Throughout this process, they were communicating with backend team so that the data can be in the correct format. They were using the API functionalities that backend team created and every team should be on the same page to implement the structure of the application as intended. At the end, the pages were ready. The posts can be viewed, created and edited successfully.

The backend team implemented GET, POST, PUT and DELETE API functionalities and created workflows for . Backend team informed other teams about the format of the data that was required in API functionalities. Images were used in the posts and for this, a separate storage system used. This storage system was AWS S3 bucket. Backend team deployed the backend of the project to the AWS EC2 instance and also, helped frontend team to deploy.

The models that were used in the database were created by looking at the deliverables. Thus, every member of the group contributed to the model of the posts that was stored in the database.

Moving Forward

On the road to Milestone 1, once again we realized the importance of planning. We started with going over the requirements and our design. We changed and improved them and created a new project plan to help us follow the path towards the Milestone 1. We acted according to the project plan and this helped us to stay organized and thus be ready for every deadline. We will also continue in this manner and follow the project plan for our next milestone. Besides, for the

second milestone we will be doing more implementation and less planning, so it is crucial to follow the deadlines and the plan.

Moreover, each team were in contact throughout the first milestone. Integrating backend with Android and Frontend also requires a lot of communication between the teams. We saw that it helped us adapt to challenges and help each other.

After this milestone, we will enable signing up, log in with different user types, we will add liking, saving and commenting on a post features, as well as, profile pages, reporting posts, reseting passwords and following users. Therefore, we can say that the second milestone will contain a lot of coding and new features.

Lastly, it is a known fact that implementing a project and maintaining it is a challenge. We will face several bugs and problems throughout this semester and the most important part while facing the problems is to stay as a group and work together. As we did during CmpE 352, we will work hard and try to learn and improve as much as possible.

List and Status of Deliverables

This section lists all deliverables for practice-app; with their delivery status, due date, and delivery date.

Deliverable Name	Delivery Status	Due Date	Delivery Date
Deliverables Report	Delivered	20.11.2021	20.11.2021
Group Report(Milestone review)	Delivered	20.11.2021	20.11.2021
Individual Reports	Delivered (for every member)	20.11.2021	20.11.2021
Requirements	Delivered	20.11.2021	20.11.2021
UML Software Designs	Delivered	20.11.2021	20.11.2021
Scenarios and Mockups	Delivered	20.11.2021	20.11.2021
Project plan	Delivered	20.11.2021	20.11.2021
Communication Plan	Delivered	20.11.2021	20.11.2021
RAM	Delivered	20.11.2021	20.11.2021

Evaluation of the status of deliverables and its impact on our project plan

This section of the report includes our evaluation and comments about what we have done so far, compares our current status to the project plan at large, and briefly explains how we are going to improve the project.

Home Page

The home page (also the main page) is the page that welcomes users when they first open the application. Users see the posts (of the users that they follow) in a scrollable manner on this page. Users can also navigate to other pages from the main page.

All teams (i.e., frontend, backend, and android) have successfully implemented the baseline of the home page functionalities. First, frontend and android teams worked on the design of the home page, with the help of the last semester's mock-ups. After deciding on the data types (i.e., multipart form-data), the backend team managed to implement all the necessary endpoints (i.e., APIs). Finally, frontend and android teams integrated those APIs and have successfully connected backend endpoints.

Create Post Page

We created 'Create Post' page of Postory and its necessary endpoints in the backend. Creating posts is an important part of our application which lets users to pick time, locations, and tags of a post; as well as its title and body. We believe implementing this functionality in an early time was a really good decision. We are also planning to improve 'Create Post' page by implementing abilities to choose locations on the map, tag people, and choose a time span for a post.

Edit Post Page

We created 'Edit Post' page of Postory and its necessary endpoints in the backend. Being able to edit posts is a feature we decided to have in our application. It lets users to edit title, body, time, locations, and tags of a post. We think we did very well implementing it for Milestone 1. In its current stage, any person can edit any post; however we will let users edit their own posts only when we implement users in our application.

Deployment

Both frontend and backend are dockerized and deployed using AWS. On the backend, we used AWS S3 buckets to store images. They both function rather well without causing problems and we are able to reset the server as we want. CI / CD pipelines for both have been completed for both.

Satisfaction of Requirements

This part of the report aims to evaluate the satisfaction of project requirements in the first milestone. The satisfaction status of project requirements will be classified into three classes:

- 1- Satisfied
- 2- Partially satisfied
- 3- Unsatisfied

Summary

Until the first milestone, the team tried to implement the core functionalities of our application. We gladly note that every requirement we addressed is at least partially satisfied now. Both frontend and Android components communicate with the backend. The users can create, edit or view posts. However, the authentication system is not present now, so some functionalities are incomplete. For instance, a user should only be able to edit her posts, but since the authentication system is incomplete, it is as if everybody owns any post on the application. Another deficiency is that the user cannot adjust the precision of location and time of her post. Such requirements will be regarded as "partially satisfied"

List of addressed requirements

This part contains the project requirements that the team tried to address until Milestone 1. Explanations to partially satisfied requirements will be numbered in square brackets. At the end of partially satisfied requirements list, you can check the reasons why the requirements were not completely satisfied. Below are the lists of "partially satisfied" and "satisfied" requirements ordered in same order with the [requirements document](#) in the project repository:

Partially satisfied requirements

- **1.1.1.1.2** Registered Users shall be able to view posts.[1]
- **1.1.1.1.7** Registered Users shall be able to edit their own posts. [1]
- **1.1.2.4.1** Registered users shall be able to create posts by selecting a location or multiple locations.[1]
- **1.1.2.4.3** Registered users shall be able to add pictures to their posts.[1]
- **1.1.2.4.4** Registered users should add date and time to their posts at the desired specific level as they remember their memories.[1][2]
- **1.1.2.4.6** Registered users shall be able to tag their posts with words related to their posts.[1]
- **1.1.2.5.4** Registered users shall be able to view the posts from registered users they follow or posts that contain the places/tags they follow on the registered user's main page.[1][3]
- **1.1.2.13.1** Registered users shall be able to edit the main part of their posts.[1]
- **1.1.2.13.2** Registered users shall be able to edit the tags of their posts.[1]
- **1.1.2.13.3** Registered users shall be able to edit the locations on their posts.[1]
- **2.3.4** Registered users' posts shall be stored in database.[1]

Explanations

1. As authentication system is not complete, the registered user type does not exist.
2. There are time and location properties in posts, but the precision is not adjustable.
3. Following users, places or tags is not possible yet.

Satisfied Requirements

- **1.1.1.3.1** Guests shall be able to view the main page.
- **1.1.1.3.2** Guests shall be able to view public posts.
- **1.1.2.3.1** Guests shall be able to enter the web application without signing up.
- **2.4.1** There should be a web interface for the project.
- **2.4.1.1** Known web browsers(such as Chrome, Opera, Firefox, Microsoft Edge, Safari) should support the web site.
- **2.4.2** There should be an android-based mobile app for the project.
- **2.5.1.1** Response time shall be short enough not to harm interactivity.
- **2.5.1.2** Reducing the response time for users should be prioritized.

List of unaddressed requirements

This part includes requirements that were not addressed by the team. Clearly, all of these requirements are not satisfied for now. Below are the lists of unaddressed functional and non-functional requirements:

Unaddressed Functional Requirements

- **1.1.1.1.1** Registered Users shall be able to fill their profile page with their information. This information includes their names, surnames, profile photos.
- **1.1.1.1.3** Registered Users shall be able to sign in by using their username or email, and password.
- **1.1.1.1.4** Registered Users shall be able to sign out.
- **1.1.1.1.5** Registered Users shall be able to search posts.
- **1.1.1.1.6** Registered Users shall be able to filter posts.
- **1.1.1.1.8** Registered Users shall be able to save posts.
- **1.1.1.2.1** Admins shall be able to view posts.
- **1.1.1.2.2** Admins shall be able to revise reports of inappropriate post.
- **1.1.1.2.3** Admins shall be able to ban registered users.
- **1.1.1.2.4** Admins shall be able to remove a post.
- **1.1.1.2.5** Admins shall be able to remove a comment under a post.
- **1.1.1.2.6** Admins shall be able to send mail to registered users.
- **1.1.1.3.3** Guests shall be able to search posts.
- **1.1.1.3.4** Guests shall be able to filter posts.
- **1.1.1.3.5** Guests shall be asked to sign up for the rest of the registered user functionalities listed under 1.1.1.1.
- **1.1.1.3.6** Guests shall be able to sign up.
- **1.1.2.1.1** Guests shall be able to sign up via providing their name, surname and e-mail address, and choosing a username and password.
- **1.1.2.2.1** Registered users shall be able to sign in with their username/e-mail address and password.
- **1.1.2.4.2** Registered users shall be able to allow location services to find their current location and add a story to it.
- **1.1.2.4.5** Registered users shall be able to tag other people to their posts.
- **1.1.2.5.2** Registered users shall be able to view posts on the locations they choose from the map.
- **1.1.2.5.3** Registered users shall be able to view the number of views of a post.
- **1.1.2.6.1** Registered users shall be able to search for posts posted at a specific time and date.
- **1.1.2.6.1** Registered users shall be able to search for posts using the map.
- **1.1.2.7.1** Registered users shall be able to leave comments under the posts.

- **1.1.2.7.2** Registered users shall be able to like posts of other registered users that they follow or that have public profiles.
- **1.1.2.7.3** Registered users shall be able to like comments under a post.
- **1.1.2.8.1** Registered users shall be able to save the posts in their own profiles to access later.
- **1.1.2.9.1** Registered users shall be able to follow the registered users with public profiles without sending request.
- **1.1.2.9.2** Registered users shall be able to send follow requests to private profiles.
- **1.1.2.9.3** Registered users shall be able to follow locations.
- **1.1.2.9.4** Registered users will be able to follow a private profile only if the owner of the private profile approves the follow request.
- **1.1.2.10.1** Registered users shall be able to edit their bios, names, surnames.
- **1.1.2.10.2** Registered users shall have a feed to display their activities.
- **1.1.2.10.3** Registered users shall be able to choose whether the profile page will be accessible to guests or registered users.
- **1.1.2.11.1** Registered users shall be able to choose to receive notifications or not.
- **1.1.2.12.1** Registered users shall be able to report inappropriate posts.
- **1.2.1.1** System shall ask the user to verify their address.
- **1.2.2.1** System shall have a "Forgotten Password" option in case registered users forget their password. Registered users will enter their emails and a link where they can change their password will be sent by the system via their emails.
- **1.2.2.2** System shall provide the user a "Remember Me" option to save username and password in local.
- **1.2.3.1** System should request the guest users to sign up for them to use the Android application.
- **1.2.4.1** System shall notify admins in case of a report of an inappropriate post.
- **1.2.4.2** System shall notify registered users with private accounts when they receive a follow request.
- **1.2.4.3** System shall notify registered users when someone starts to follow them.
- **1.2.4.4** System shall notify registered users when someone likes their posts.
- **1.2.4.5** System shall notify registered users when someone comments on their posts.

Unaddressed Non-Functional Requirements

- **2.2.1.1** Registered user passwords should be at least 8 characters long and should contain at least 1 uppercase letter, 1 number, 1 symbol.
- **2.2.1.2** Registered user passwords should not contain the registered user's username.
- **2.2.2** A verification email should be sent to the registered user if password change is requested. Until the registered user verifies the change, the password should not be changed.
- **2.2.3** Admins should change their passwords every 180 days. This should be reminded to the admin via email.
- **2.2.4** After 4 unsuccessful password entries, the registered users should verify that it was them by clicking the verification link sent to them via email.
- **2.2.4.1** Until the verification is done, the registered user can't enter the app.
- **2.2.4.2** After the verification, the registered users should reset their passwords.
- **2.3.1** Registered users' sign up data shall be stored in database.
- **2.5.2.1** Measures shall be taken to shorten the downtime as much as possible.
- **2.6.1** W3C Activity Streams 2.0 Protocol will be followed throughout the project implementation.
- **2.6.2** W3C Standards will be followed.

Evaluation of Tools and Processes

In this section, the tools and processes used during the development of "practice-app" are briefly evaluated.

Evaluation of Tools

The team used several tools to develop the application. Each one of these tools serves different purposes. The evaluations include the opinions and experiences of team members.

General

Discord

We use discord as our main communication tool. Weekly meetings are held on discord. It has multiple voice and text channels so we can hold multiple meetings at the same time (was useful when we separated into multiple groups). It has a feature for pinning messages, we use that feature for saving important links. It allows multiple screen sharings at the same time which is useful at times.

Whatsapp

We used Whatsapp for short communications and finding some time for emergency meetings. It is useful for spontaneous communication but it has no way of providing structure. Because of that, we try to use discord for longer discussions.

Git

Git is a popular version control system that allows people to develop code in an organized manner. Nothing gets lost when one accurately use Git. In addition, everything becomes much more trackable.

GitHub

GitHub provides a beautiful user interface and environment that enhance the functionalities of Git. Firstly, GitHub Wiki allows us to make documentation for our project in the same environment as the code. Issues are beneficial for clarifying who is responsible for each task. They are also useful in referencing other wiki pages, issues, collaborators and pull requests.

Backend

AWS

AWS allowed us to deploy our application without much effort. There are plenty of tutorials that teach how one can deploy a containerized application. Also, the free services seem sufficient to create simple websites. We have not had huge problems regarding AWS.

Amazon S3

Amazon S3 is the tool we used to store the images. It is a cloud object storage service that allows upto 5 GB storage with AWS free tier. For Postory, we decided that 5 GB is enough. S3 buckets are easy to use with an AWS account and there are many tutorials on how to integrate S3 using Django. After the images are stored in buckets, a URL for each image will be created. Thus, storing the images on cloud makes it easier to receive images/files from Frontend and Android in the

form of Multipart/form data. Also, it eases the load on the database since we only keep the urls of the images in our Database.

Django

Django is a Web framework written for Python. Its Object-Relational Mapper(ORM), made it much easier to communicate with our MongoDB server. Its built-in admin panel also helped us for creating, updating and deleting DB objects.

As we were implementing our API functionalities, we used Django Rest Framework. Django Rest Framework provides a further abstraction for interacting with DB objects named serializers. Using its GenericAPIView it is easier to divide the endpoints to GET, POST, PUT and DELETE requests.

Docker

Docker is a virtualization tool commonly used in modern software development. Docker containers allowed us to deploy the application quickly. Once you create a container correctly, it is for sure that the application will run in the correct environment. Moreover, the deployment of Docker containers is facile. The only disadvantage is that Windows users need to install WSL(Windows Subsystem for Linux) to work with Docker properly.

MongoDB

MongoDB is a document-oriented NoSQL database. The data is stored as collections and documents and it can store high volumes of data. In our project, MongoDB is used with Django. To integrate the database with Django, Djongo engine is used. MongoDB docker image is used to deploy the database to AWS EC2 instance. Deploying and connecting the database to Django were quite simple. In our application, the models for collections created and Django created the collections automatically.

Visual Studio Code

Visual Studio Code(VSCode) is a code editor made by Microsoft. It supports a wide variety of plugins that simplify the development process. Integration with Git allows reverting local changes on specific lines. It also supports the use of the terminal without changing focus. We had decided to use VSCode because it supports connecting remotely with team members, but we did not try this functionality since we worked on different tasks. In summary, VSCode is a perfect tool for developers, and we have not noticed any problems.

Postman

Postman is a tool that is used for verifying the APIs you develop. We used it to test our API code while the frontend was not yet implemented. It allowed us to name our requests and save them for the purpose of using them again later instead of writing the same request each time.

Google Maps

Google maps is an online mapping platform. It is a wide service including satellite images, street maps, real-time traffic information, route planning etc. We are using their API to find the coordinates of the entered location for the post. We will use this more as we progress to our main goal of the project, since it will include choosing location from the map itself, selecting a route.

Android

IntelliJ IDEA

IntelliJ is a very suitable IDE for Android software development. It is %100 compatible with Android SDK, it provides tools for viewing XML layouts and it also comes with an emulator on which one can see how the app looks on a virtual Android device. It also has all the necessary tools for building and deploying applications.

Gradle

Gradle is a build automation tool, which makes it very easy to build, compile, package and deploy software. It is built-in to the IntelliJ IDE and when one starts an Android application it already comes with it. build.gradle file specifies the configuration of the build and one can add external libraries by adding the corresponding link to the build.gradle.

Java and Native Android SDK

The Android team used Java and Android SDK for the development. Java is not very concise, but it is regarded as faster than other options for Android development. It is easier to communicate with Android OS when Native SDK is used rather than a third-party framework such as React-Native.

The team was more experienced in Java compared to Kotlin. There is good documentation by Google on Android SDK for both Kotlin and Java. Java can be regarded as an old programming language, so it seems quite reliable. Although some people claim Kotlin is ousting Java, we observed a very active Java Android developer community.

There are common complaints of Android developers regarding unstable behaviour between different versions of IDE, Java and Gradle versions. Thankfully, we have not observed any of such problems but there is always such a risk.

For the formal documentation of Android SDK, [click here](#).

Frontend

Framework

As the frontend team, we did not have much experience with related frameworks. Hence, we have decided to use React JS framework, which is recommended by our TA.

The component-based structure of this framework allowed us to implement the program generically by with the help of re-usable code snippets (e.g., reusing the same renders in different components etc.).

We have used several tutorials/resources that helped us through the implementation, which are listed below:

- [Intro. to React](#)
- [List and Keys](#)
- [React State and Arrays](#)
- [Snackbar](#)

IDE

We mostly used Visual Studio Code for developing the frontend. It has useful features such as using a terminal in the app and it is well integrated with github which makes tracking our changes easier.

Chrome devtools

Chrome devtools made it easier to debug the frontend. It has a console in which it is possible to see relevant debug messages and try out changes for the code. Using the elements tab we can try changes to the HTML without recompiling the React code. Network tab is used for debugging the connection with the backend.

UI Framework

We decided to use React as our Frontend framework. It lets us create components that are dynamically updated and can be reused in different parts of the UI. However, we didn't choose any UI framework, we simply created all UI elements and their stylings by hand. In the future, we are planning to use a UI framework such as Material UI or Bootstrap to ease our work.

Evaluation of Processes

Working as a team is an intricate task. Good processes ensure that a team might collaborate well. The processes adopted for the development of "practice-app" are evaluated in this section.

Team Meetings

The team held meetings very frequently. At meetings, the members discussed the distribution of tasks. We also completed some tasks directly at the meetings. For instance, some merge requests required us to discuss the changes. In addition, we needed the same files or the same pieces of code for different features; so we wrote some of them during the meetings. Also, some team members had specific issues about Git, their codes or external APIs. We solved some of such problems at the meetings. Team members took meeting notes rotationally and posted them on GitHub Wiki for review.

LAB Meetings

We hold weekly meetings with our TA. The teaching assistant explains concepts about software engineering to us in detail. We also asked questions related to the content and scope of the assignment. In a nutshell, TA meetings were very beneficial for the development of the application.

Issue Creation

We continued the same approach that each person should create issues for herself. Our repository has a high number of GitHub issues, and it seems trackable. Different from before, we have also created GitHub issues related to bugs and pull requests while developing the application.

Code Reviews

The team decided to create branches for each new feature. After implementation is completed, each team member created a pull request for her code. Many people reviewed pull requests that include significant changes. Code reviews helped us to merge changes without damaging the whole system. Also, we learned from each other while examining the codes.

Project Plan

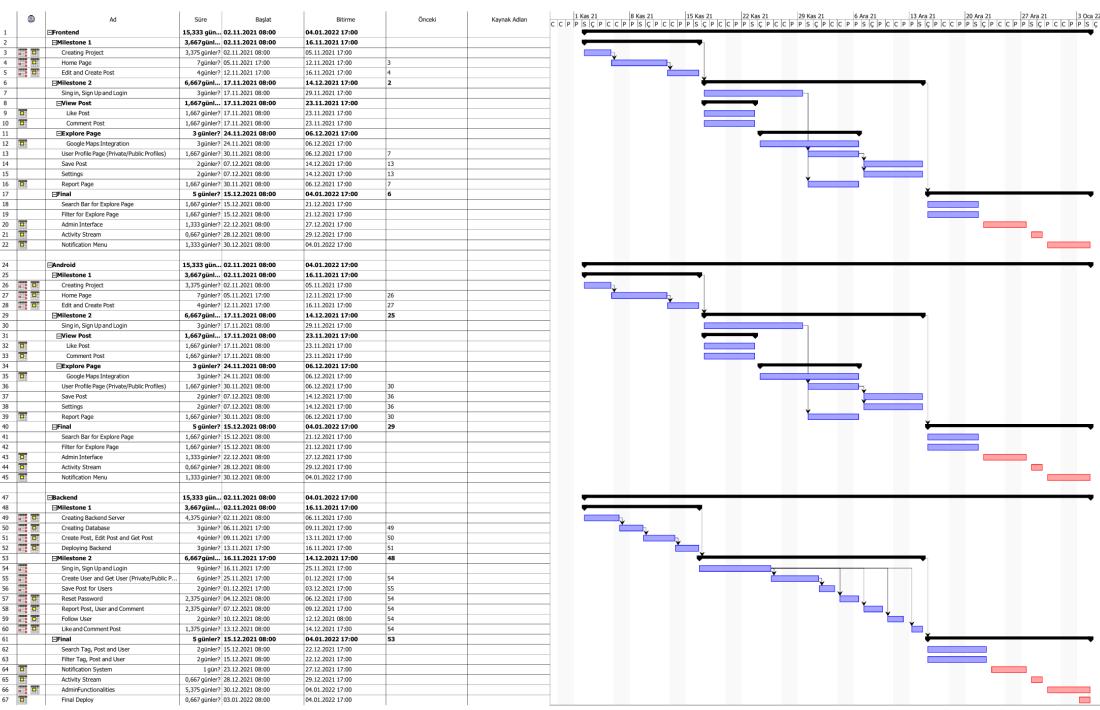


Figure 1: Project Plan for Postory

Work Done by Each Team Member

Name	Work Done
Mert Alkan	<ul style="list-style-type: none"> I corrected the ambiguity of our model name in the requirements. I created location and tag models and used them in the post model as lists. I created post creation API endpoint. Used Google Maps API to find the coordinates of locations Attended all group meetings and wrote notes for Lab 2 Meeting and Meeting 5 Wrote down introduction in executive summary for this Milestone Report Wrote down work done by myself in this milestone report Wrote individual milestone report
Ahmet Melih Aydoğdu	<ul style="list-style-type: none"> Reviewed the old deliverables and made changes Enhanced the glossary in requirements Finalized the project plan Updated follow a user sequence diagram Learned GitHub actions Learned how to integrate MongoDB with Django Researched and worked on Git to use it effectively Created AWS EC2 instance and set the configurations Created workflows for deployment Wrote docker commands for deployment Added corsheaders to backend Implemented GET, PUT and DELETE API functionalities Added image store mechanism to backend Helped frontend for deployment Worked on issues: #226, #239, #244, #262, #282, #283 Opened pull requests: #274, #275, #287, #288 Wrote reviews to pull requests: #259, #264, #277, #321, #322 Wrote the individual report for Milestone 1 Took meeting notes in Lab 3, Meeting 6 and Backend Meeting1

Name	Work Done
Zehranaz Canfes	<ul style="list-style-type: none"> • Change All Occurrences of 'user' to 'registered user' in the Repository - Issue #215 • Thought about new requirements to add - Issue #216 • Written questions to ask the customer - Issue #227 • Edited Requirement 1.1.1.1.2 - Issue #230 • Added the group name to ReadME and remove the members that left. • Changed the requirements by keeping actor/agent in mind. I separated the requirements that contained more than one functionality and added new terms to glossary. • Create Logo for Postory - Issue #344 • Read Django documentation to improve my knowledge and remembered the parts I forgot. Issue here • Learn MongoDB and its integration with Django - Issue #335. I made a research on Django and PyMongo. • Implemented an API for getting all posts from the DB - Issue #260 • I worked on bugs 1 and 2 • I made a research and found a way to store the images. I found Amazon S3 and shared the details with the backend team. Issue here • Written Two Posts for Postory App - Issue #329 • Learned GitHub Actions • Attended all meeting except the last one due to my exam. • Attended all labs and lectures • Wrote the Moving Forward part of the Executive summary of this report • Wrote individual report • Wrote weekly efforts • Opened issues: Milestone 1 Issues • Opened Pull Request: Get All Posts • Wrote reviews on: Backend's last Pull Request for Milestone, API for creating posts, Locations and Tags models are added and they are kept in lists, Workflows created, Get, Put, Delete Functionality • Kept the meeting notes of the backend meeting and meeting 2.

Name	Work Done
<i>Mustafa Emir Çolak</i>	<ul style="list-style-type: none"> • Implemented the general post component for the frontend • Implemented the general structure of the Create Post page for the frontend • Implemented the Edit Post page reusing the parts created in the Create Post page. • Fixed the sequence diagrams(7-8 of them) according to the feedback we had from the instructors • Did compile all the completed deliverables into a pdf. • Listed the status of deliverables in this report • Attended all the group meetings and written the meeting report for meeting 3. • Evaluated the IDE part and Chrome Devtools for evaluation of tools part
<i>Ömer Barış Erkek</i>	<ul style="list-style-type: none"> • Attended all weekly Discord meetings of our group • Attended all Lab meetings of our group • Attended all meetings of the Frontend team • Took meeting notes of Lab 4 meeting • Tracked my weekly efforts and putting them on our GitHub wiki • Created, documented, and closing GitHub issues for my works • Learned React framework • Implemented TopBar, SearchBar, TimeChooser, and TextChooser components for the frontend • Implemented my contributions in separate branches, pushed them to our GitHub repository, and created pull requests for them when my work is completed • Reviewed pull requests of my teammates • Added two posts to the Postory application • Presented some of the Frontend presentation of our Customer Demo • Written my individual report • Contributed to the group report
<i>Melih Özcan</i>	<ul style="list-style-type: none"> • Researched about Android in general and libraries • Implemented the show posts feature • Implemented edit posts feature • Implemented the create post feature • Added a refresh button • Took the meeting notes of Lab 5 • Attended Lab Meetings 1, 2, 3, 4, 5 and Group Meetings 1, 2, 3, 4, 5, 6 • Prepared the individual report • Contributed to the Group Report • Created the issues 245, 255, 307, 308, 313, 316, 318, 323 and 326 • Created the pull requests 284, 310, 315, 317, 320, 327 and 331

Name	Work Done
Ahmet <i>ibrahim</i> Şentürk	<ul style="list-style-type: none">• Implemented post buttons• Implemented add tag section• Implemented add people section• Implemented add location section• Implemented popup mechanism for unimplemented features• Deployed frontend server• Updated wiki home page• Updated wiki communication page• Finalized project plan• Updated report sequence diagram• Wrote a guideline for the milestone 1 presentation• Worked on issues: #223, #224, #225, #240, #241, #247, #256, #257, #258, #280, #303, #304• Opened PRs: #252, #268, #271, #272, #292, #321• Reviewed PRs: #250, #251, #267, #269, #270, #301, #309, #319, #322• Attended all of the meetings and lab sessions• Took meeting notes of Lab 1, Meeting 4 and Frontend Meeting. <p>2</p>

Name	Work Done
Niyazi Ülke	<ul style="list-style-type: none"> • Researched about Android development. • Created 3 issue templates for the repository: bug, documentation and programming. • Organized wiki homepage and sidebar and improved the explanations about the project. • Improved the explanation of time and location in our requirements. • Designed the logo of our project. • Upgraded meeting report template to include deadlines and better organization. • Initialized the Android project with Melih Özcan. • Implemented extend post functionality for Android application. • Wrote an UI test with Espresso for extend post functionality. • Wrote the base of create post activity for Android application. • Designed the toolbar of Android application and added the icons. • Wrote two posts for the Postory application before the presentation. • Presented Android application during the milestone presentation. • Wrote Satisfaction of Requirements part for this milestone report. • Wrote evaluation of Java and Android SDK for this milestone report. • Wrote individual milestone 1 report. • Took meeting notes at Meeting 1 and Android Meeting 1. • Worked on issues #217, #218, #231, #234, #237, #238, #246, #276, #291, #293, #300, #305, #324 • Opened pull requests #285, #294, #302, #306, #314, #322, #333 • Reviewed pull requests : #284, #310, #315, #317, #320, #327

Requirements

Glossary

- **Admin:** The administrator of the project who evaluates the reported posts and who is responsible of management.
- **All Users:** Guests, admins and registered users.
- **Bio:** About section of a user.
- **Ban:** Prohibiting a user from using the app.
- **Follow:** A way for users to see other users' posts or posts that have related tags in their main page.
- **Guest:** The people visiting the app but not yet signed up.
- **Inappropriate Content:** Text or images that disturb users.
- **Location:** A location might be a point or a region on the map.
- **Notification:** Giving information about the actions that is happening on the app.
- **Main Page:** The page that welcomes users when they first open the app. Users can navigate to other pages from main page.
- **Memory:** The reason a user posts. Users share their experiences and feelings by using posts.
- **Post:** A content which a user shared by using the app. This content can consists of place, time, pictures and tags.
- **Private Profile:** A user's page that can be viewed by other users if only the user give permission.
- **Profile Page:** A specific page for a registered user that includes her personal knowledge that she wants to share.
- **Public Profile:** A user's page that can be viewed by everyone.
- **Registered User:** The people who have signed up and verified their email addresses.
- **Report:** Informing against a post that may be inappropriate to the admins to be evaluated and deleted if necessary.
- **Reset Password:** After clicking the link sent by admins when registered users forget their passwords, the registered user selects a new password and signs in again.
- **Sign in:** Entering the application by providing the correct password and email.
- **Sign up:** Creating an account via providing email, username, password to use the application not as a guest.
- **Tag:** Identifying another user or a topic about the post.
- **Terms of Use:** An agreement that user accept while signing up.
- **View:** How many times a post is seen by other users.
- **View Post:** Display the post in a separate page.

1. Functional Requirements

► 1.1 User Requirements

- 1.1.1 User Types

- 1.1.1.1 Registered User

- 1.1.1.1 Registered Users shall be able to fill their profile page with their information. This information includes their names, surnames, profile photos.
 - 1.1.1.2 Registered Users shall be able to view posts.
 - 1.1.1.3 Registered Users shall be able to sign in by using their username or email, and password.

- **1.1.1.1.4** Registered Users shall be able to sign out.
- **1.1.1.1.5** Registered Users shall be able to search posts.
- **1.1.1.1.6** Registered Users shall be able to filter posts.
- **1.1.1.1.7** Registered Users shall be able to edit their own posts.
- **1.1.1.1.8** Registered Users shall be able to save posts.
- **1.1.1.2 Admin**
 - **1.1.1.2.1** Admins shall be able to view posts.
 - **1.1.1.2.2** Admins shall be able to revise reports of inappropriate post.
 - **1.1.1.2.3** Admins shall be able to ban registered users.
 - **1.1.1.2.4** Admins shall be able to remove a post.
 - **1.1.1.2.5** Admins shall be able to remove a comment under a post.
 - **1.1.1.2.6** Admins shall be able to send mail to registered users.
- **1.1.1.3 Guest**
 - **1.1.1.3.1** Guests shall be able to view the main page.
 - **1.1.1.3.2** Guests shall be able to view public posts.
 - **1.1.1.3.3** Guests shall be able to search posts.
 - **1.1.1.3.4** Guests shall be able to filter posts.
 - **1.1.1.3.5** Guests shall be asked to sign up for the rest of the registered user functionalities listed under 1.1.1.1.
 - **1.1.1.3.6** Guests shall be able to sign up.
- **1.1.2 User Interactions**
 - **1.1.2.1 Sign Up**
 - **1.1.2.1.1** Guests shall be able to sign up via providing their name, surname and e-mail address, and choosing a username and password.
 - **1.1.2.2 Sign In**
 - **1.1.2.2.1** Registered users shall be able to sign in with their username/e-mail address and password.
 - **1.1.2.3 Guest Entry**
 - **1.1.2.3.1** Guests shall be able to enter the web application without signing up.
 - **1.1.2.4 Creating Posts**
 - **1.1.2.4.1** Registered users shall be able to create posts by selecting a location or multiple locations.
 - **1.1.2.4.2** Registered users shall be able to allow location services to find their current location and add a story to it.
 - **1.1.2.4.3** Registered users shall be able to add pictures to their posts.
 - **1.1.2.4.4** Registered users should add date and time to their posts at the desired specific level as they remember their memories.
 - **1.1.2.4.5** Registered users shall be able to tag other people to their posts.
 - **1.1.2.4.6** Registered users shall be able to tag their posts with words related to their posts.
 - **1.1.2.5 Viewing Posts**
 - **1.1.2.5.1** Registered users shall be able to view the posts of the registered users, places and tags that they follow.
 - **1.1.2.5.2** Registered users shall be able to view posts on the locations they choose from the map.
 - **1.1.2.5.3** Registered users shall be able to view the number of views of a post.
 - **1.1.2.5.4** Registered users shall be able to view the posts from registered users they follow or posts that contain the places/tags they follow on the registered user's main page.

- **1.1.2.6 Searching and Filtering**
 - **1.1.2.6.1** Registered users shall be able to search for posts posted at a specific time and date.
 - **1.1.2.6.1** Registered users shall be able to search for posts using the map.
- **1.1.2.7 Commenting and Liking Posts**
 - **1.1.2.7.1** Registered users shall be able to leave comments under the posts.
 - **1.1.2.7.2** Registered users shall be able to like posts of other registered users that they follow or that have public profiles.
 - **1.1.2.7.3** Registered users shall be able to like comments under a post.
- **1.1.2.8 Saving Posts**
 - **1.1.2.8.1** Registered users shall be able to save the posts in their own profiles to access later.
- **1.1.2.9 Following**
 - **1.1.2.9.1** Registered users shall be able to follow the registered users with public profiles without sending request.
 - **1.1.2.9.2** Registered users shall be able to send follow requests to private profiles.
 - **1.1.2.9.3** Registered users shall be able to follow locations.
 - **1.1.2.9.4** Registered users will be able to follow a private profile only if the owner of the private profile approves the follow request.
- **1.1.2.10 Registered User Profile Page**
 - **1.1.2.10.1** Registered users shall be able to edit their bios, names, surnames.
 - **1.1.2.10.2** Registered users shall have a feed to display their activities.
 - **1.1.2.10.3** Registered users shall be able to choose whether the profile page will be accessible to guests or registered users.
- **1.1.2.11 Notification**
 - **1.1.2.11.1** Registered users shall be able to choose to receive notifications or not.
- **1.1.2.12 Reporting**
 - **1.1.2.12.1** Registered users shall be able to report inappropriate posts.
- **1.1.2.13 Editing Posts**
 - **1.1.2.13.1** Registered users shall be able to edit the main part of their posts.
 - **1.1.2.13.2** Registered users shall be able to edit the tags of their posts.
 - **1.1.2.13.3** Registered users shall be able to edit the locations on their posts.

► 1.2 System Requirements

- **1.2.1 Sign Up**
 - **1.2.1.1** System shall ask the user to verify their address.
- **1.2.2 Sign In**
 - **1.2.2.1** System shall have a "Forgotten Password" option in case registered users forget their password. Registered users will enter their emails and a link where they can change their password will be sent by the system via their emails.
 - **1.2.2.2** System shall provide the user a "Remember Me" option to save username and password in local.
- **1.2.3 Guest Entry**
 - **1.2.3.1** System should request the guest users to sign up for them to use the Android application.
- **1.2.4 Notification**
 - **1.2.4.1** System shall notify admins in case of a report of an inappropriate post.

- **1.2.4.2** System shall notify registered users with private accounts when they receive a follow request.
- **1.2.4.3** System shall notify registered users when someone starts to follow them.
- **1.2.4.4** System shall notify registered users when someone likes their posts.
- **1.2.4.5** System shall notify registered users when someone comments on their posts.

2. Non-Functional Requirements

► 2.1 Privacy

- **2.1.1** System should grant the registered users options to ensure their privacy.
 - **2.1.1.1** System should ask registered users to choose the permissions granted to the app such as tracing one's location and so on.
- **2.1.2** Registered users shall be informed about our privacy policy by the system.
 - **2.1.2.1** There will be a users' agreement (privacy policy) page related to our privacy policy.

► 2.2 Security

- **2.2.1** The registered user passwords should be encrypted in the database.
 - **2.2.1.1** Registered user passwords should be at least 8 characters long and should contain at least 1 uppercase letter, 1 number, 1 symbol.
 - **2.2.1.2** Registered user passwords should not contain the registered user's username.
- **2.2.2** A verification email should be sent to the registered user if password change is requested. Until the registered user verifies the change, the password should not be changed.
- **2.2.3** Admins should change their passwords every 180 days. This should be reminded to the admin via email.
- **2.2.4** After 4 unsuccessful password entries, the registered users should verify that it was them by clicking the verification link sent to them via email.
 - **2.2.4.1** Until the verification is done, the registered user can't enter the app.
 - **2.2.4.2** After the verification, the registered users should reset their passwords.

► 2.3 Database

- **2.3.1** Registered users' sign up data shall be stored in database.
- **2.3.2** Registered users' passwords shall be stored encrypted in database.
- **2.3.3** Registered users' profile activities shall be stored in database.
- **2.3.4** Registered users' posts shall be stored in database.

► 2.4 Availability

- **2.4.1** There should be a web interface for the project.
 - **2.4.1.1** Known web browsers(such as Chrome, Opera, Firefox, Microsoft Edge, Safari) should support the web site.
- **2.4.2** There should be an android-based mobile app for the project.

► 2.5 Performance

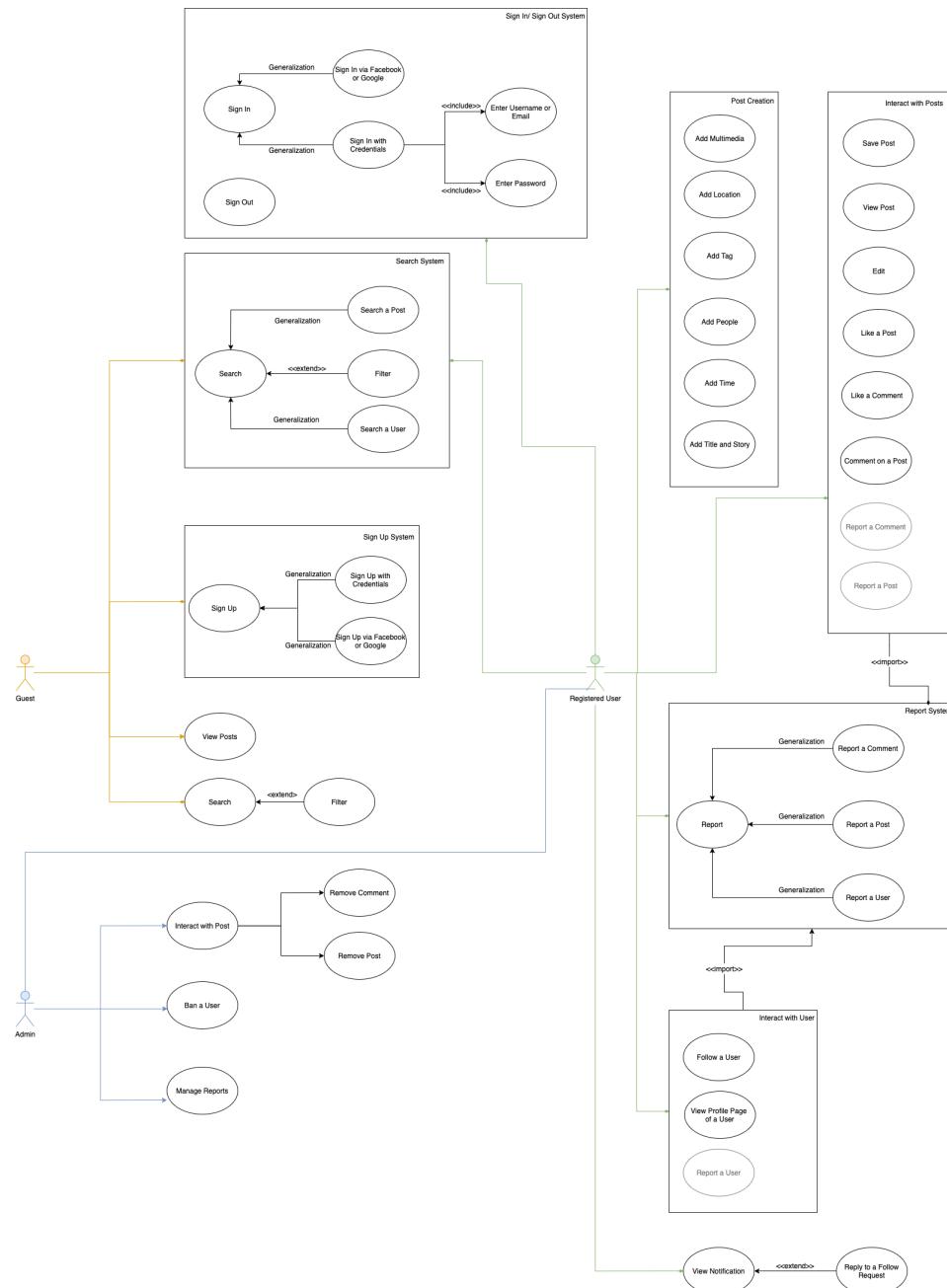
- **2.5.1 Response Time**
 - **2.5.1.1** Response time shall be short enough not to harm interactivity.
 - **2.5.1.2** Reducing the response time for users should be prioritized.
- **2.5.2 Maintenance**

- **2.5.2.1** Measures shall be taken to shorten the downtime as much as possible.

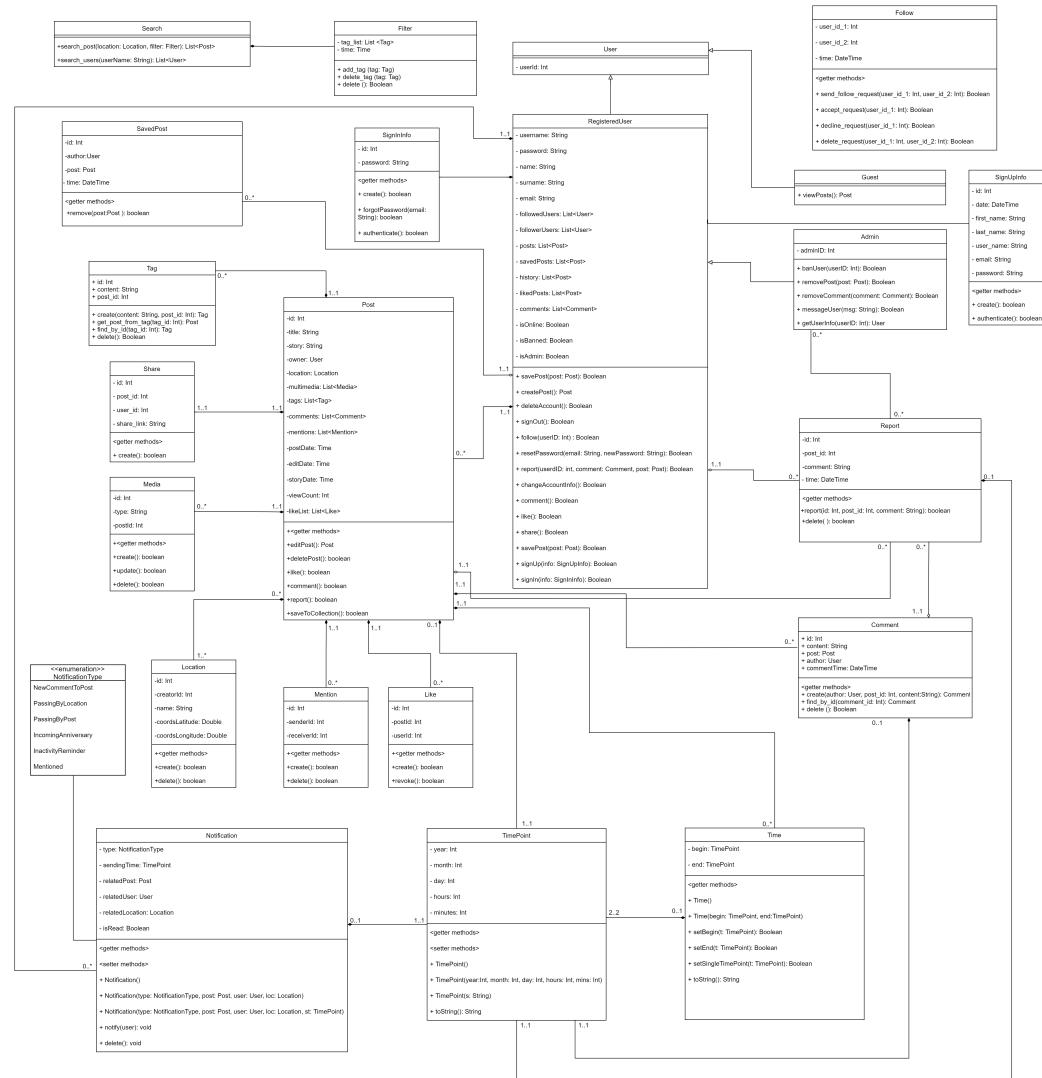
► **2.6 Protocols**

- **2.6.1** [W3C Activity Streams 2.0 Protocol](#) will be followed throughout the project implementation.
- **2.6.2** [W3C Standards](#) will be followed.

Use Case Diagram



Class Diagram



Sequence Diagrams

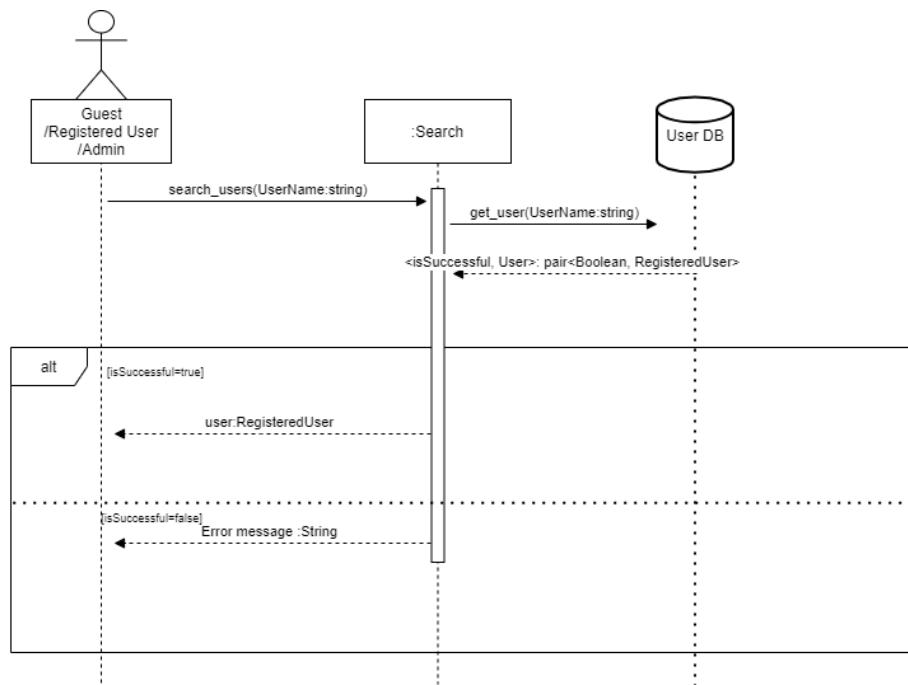
1. View a User's Profile

Actor: Guest, Registered User, or Admin

Preconditions: None

Actions:

- Clicks on a profile to see the user's profile page.



2. View Posts on Homepage

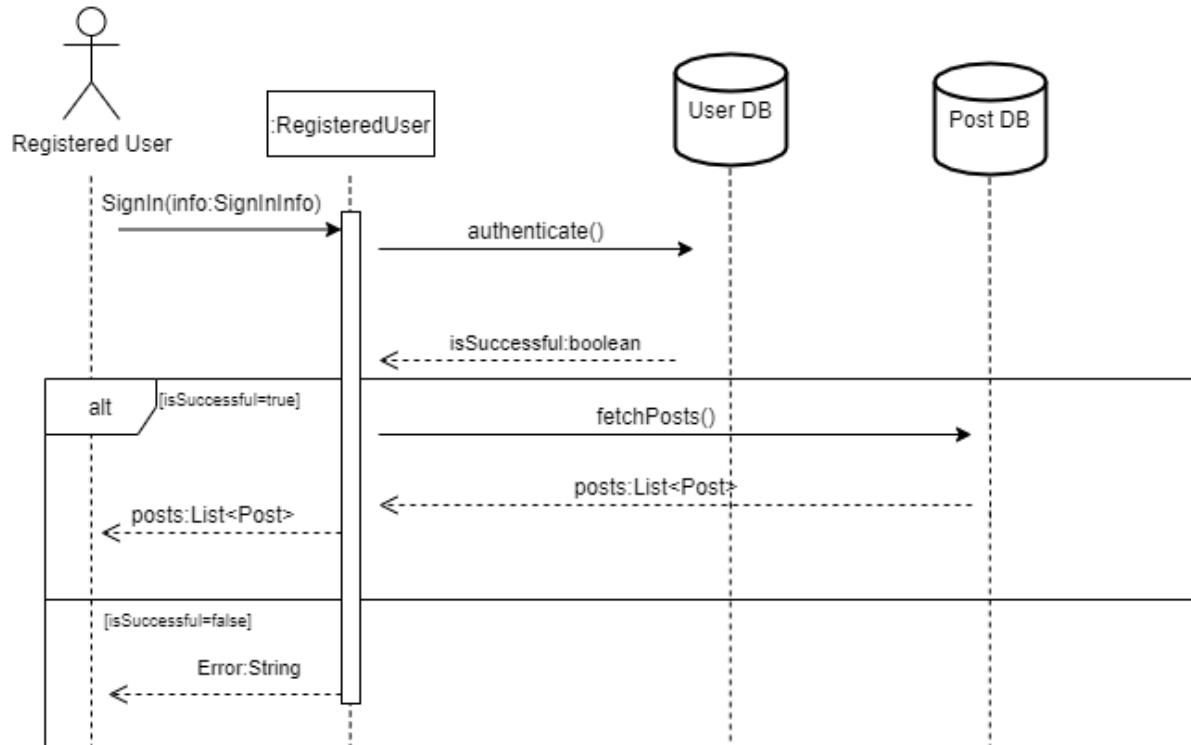
Actor: Registered User or Admin

Preconditions:

- User is registered to the application
- Their login credentials are saved.
- They follow some people, places, or tags

Actions:

- Opens app/website to see their homepage



3. Report a Post

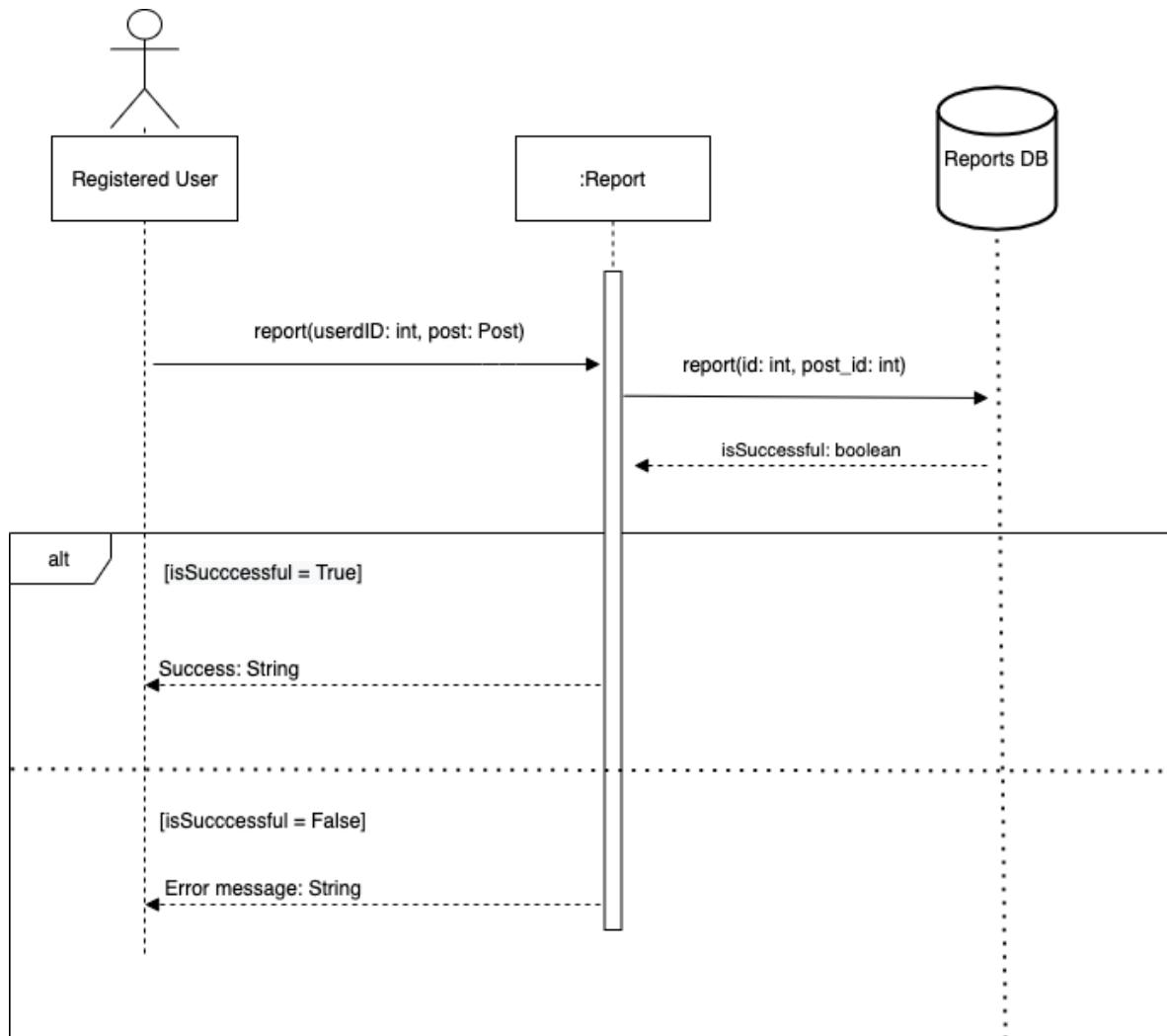
Actor: Registered User

Preconditions:

- User is registered to the application

Actions:

- Report a post that a registered user thinks is inappropriate



4. Sign up

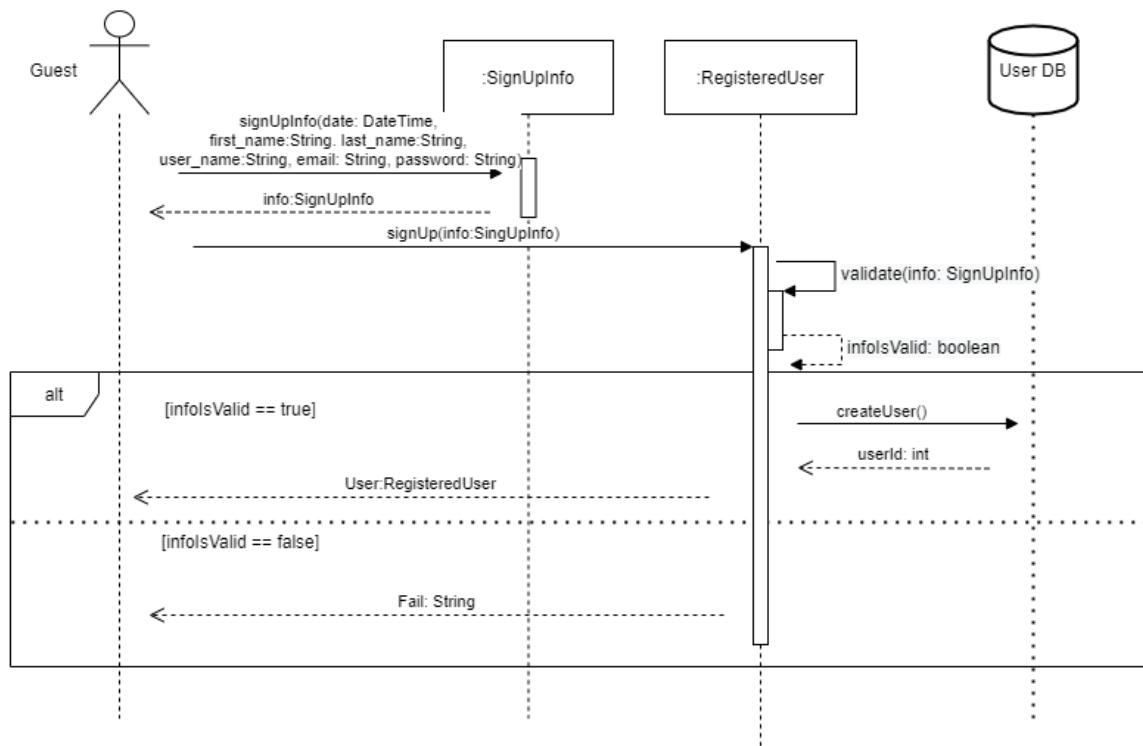
Actor: Guest

Preconditions:

- Since this is a user, he/she is not logged in.

Actions:

- Writes down a username, password, email, and his/her name surname.



5. Sign in

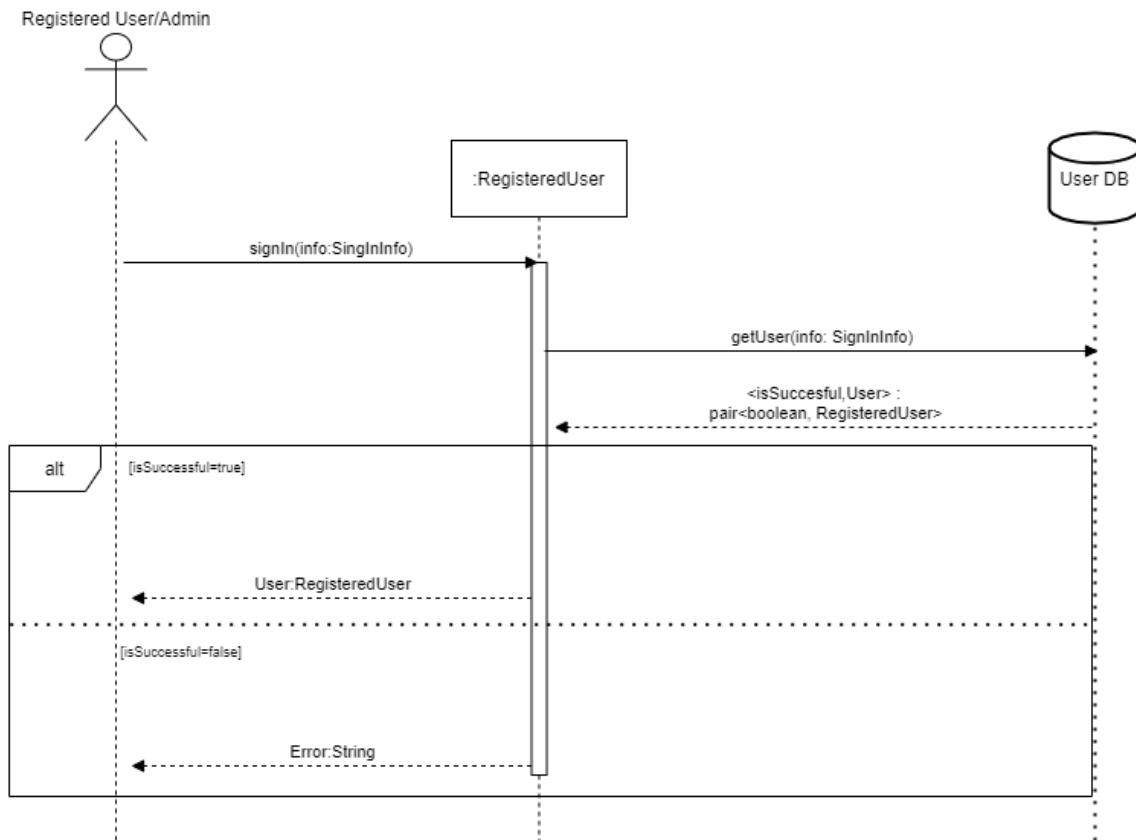
Actor: Registered User or Admin

Preconditions:

- Registered User/Admin is not logged in.

Actions:

- Enter username or email, and password. Then, sign in.



6. Comment

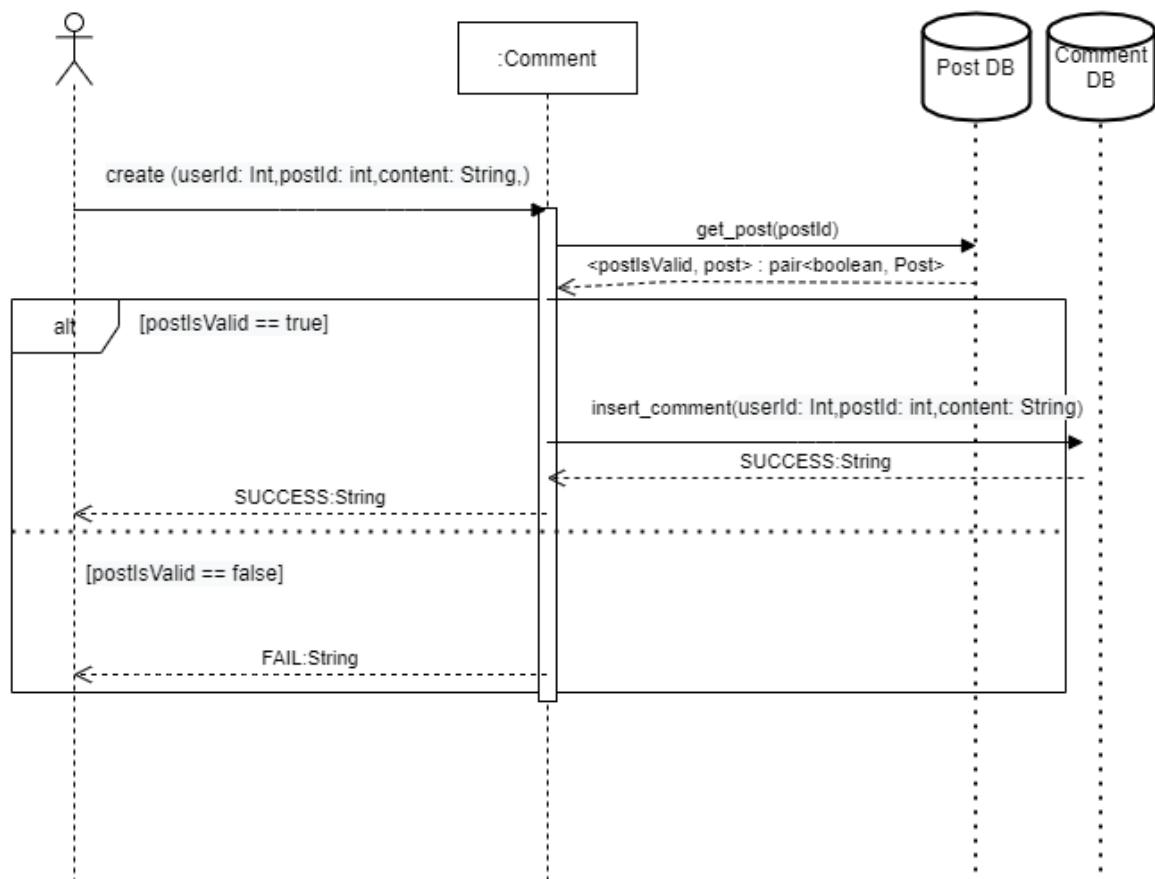
Actor: Registered User or Admin

Preconditions:

- User is registered to the application.
- Their login credentials are saved.
- They are on the page of a post.

Actions:

- User comments on a post.



7. Follow A User

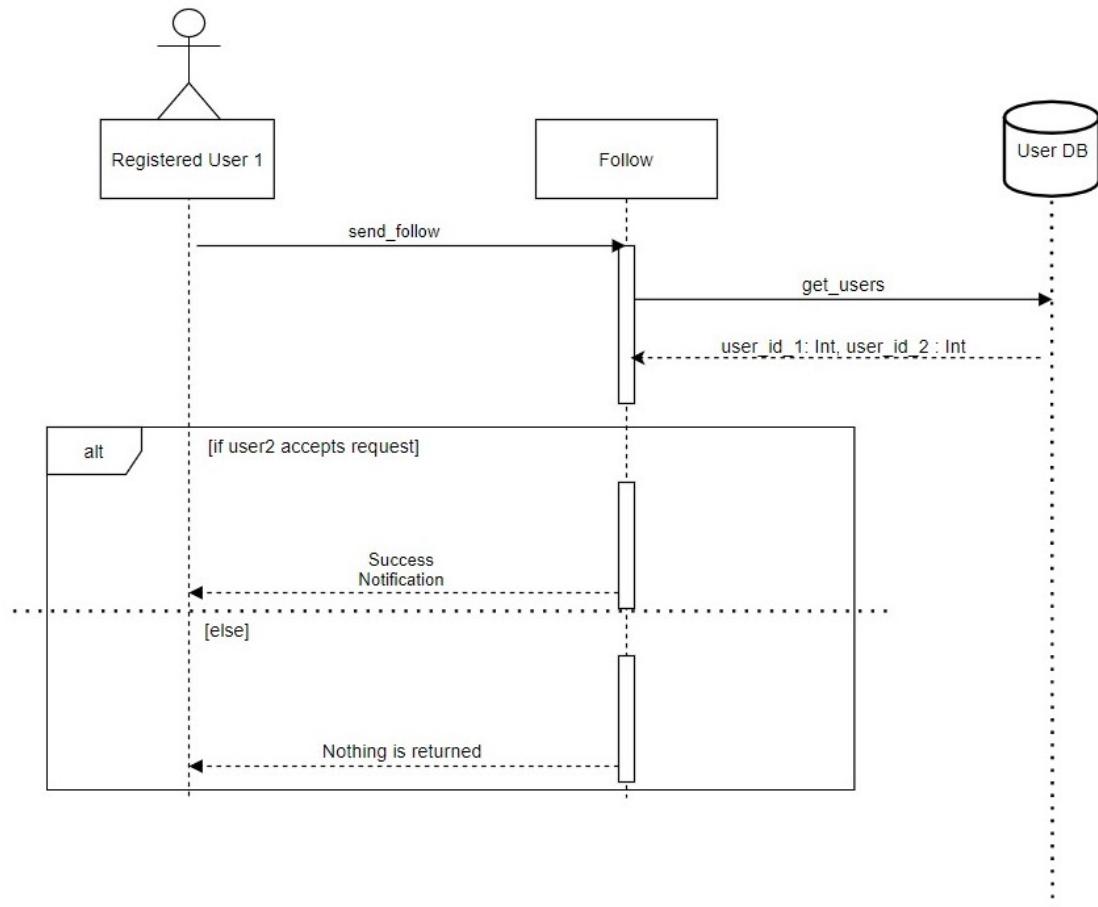
Actor: Registered User

Preconditions:

- The user is signed in and is checking the profile of another user.

Actions:

- Clicks the follow button.



8. Search a User

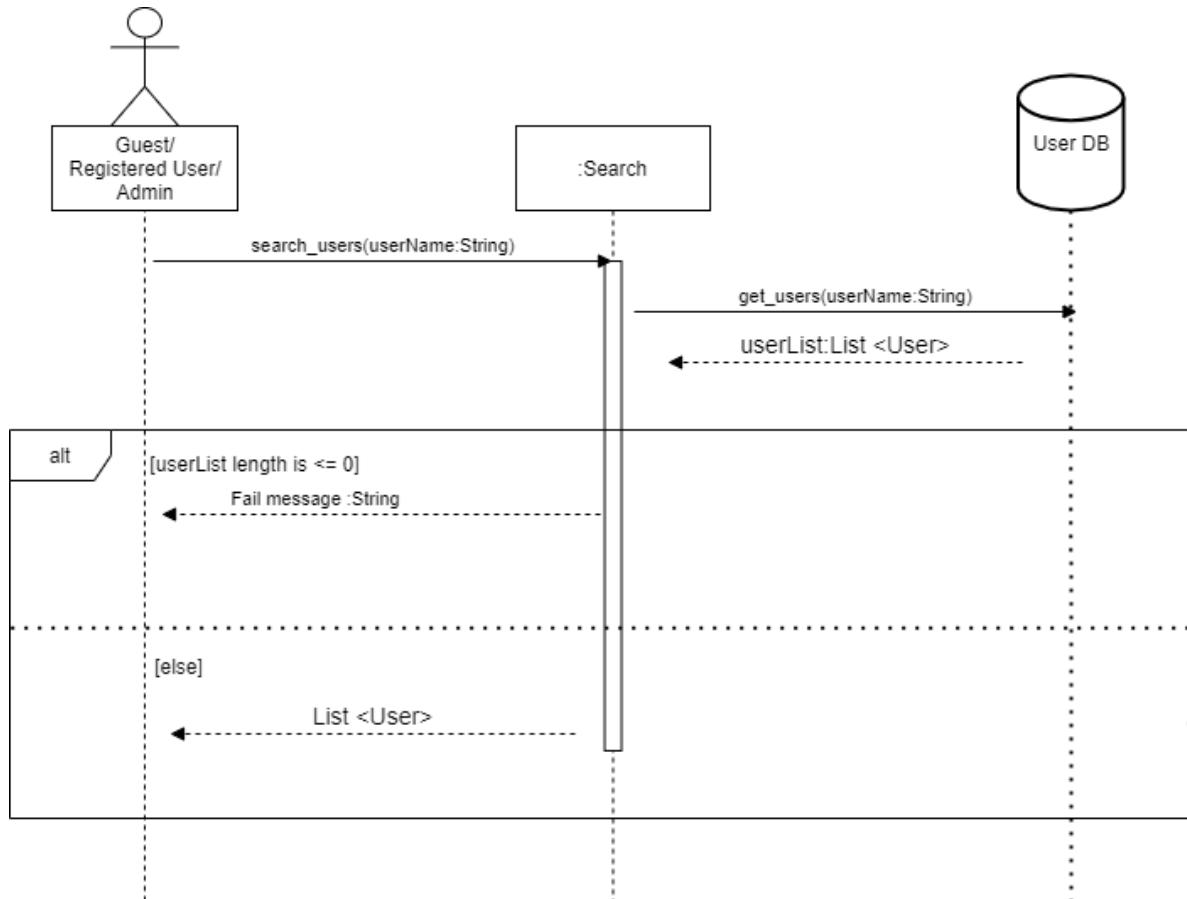
Actor: Guest, Registered User, or Admin

Preconditions:

- The user is on the search page for users.

Actions:

- Writes a user name and search for users with that name.



9. Search Posts Containing a Location

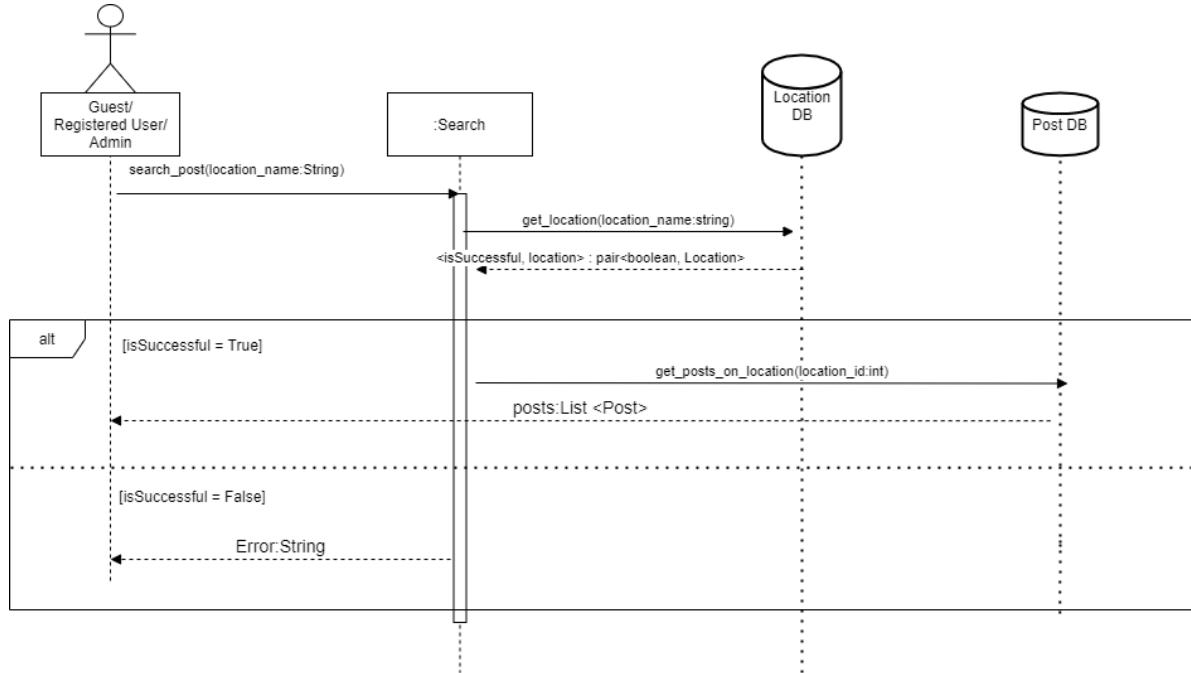
Actor: Guest, Registered User, or Admin

Preconditions:

- The user is on explore page for locations.

Actions:

- Writes a location name and search for posts with that location.



10. Create Post

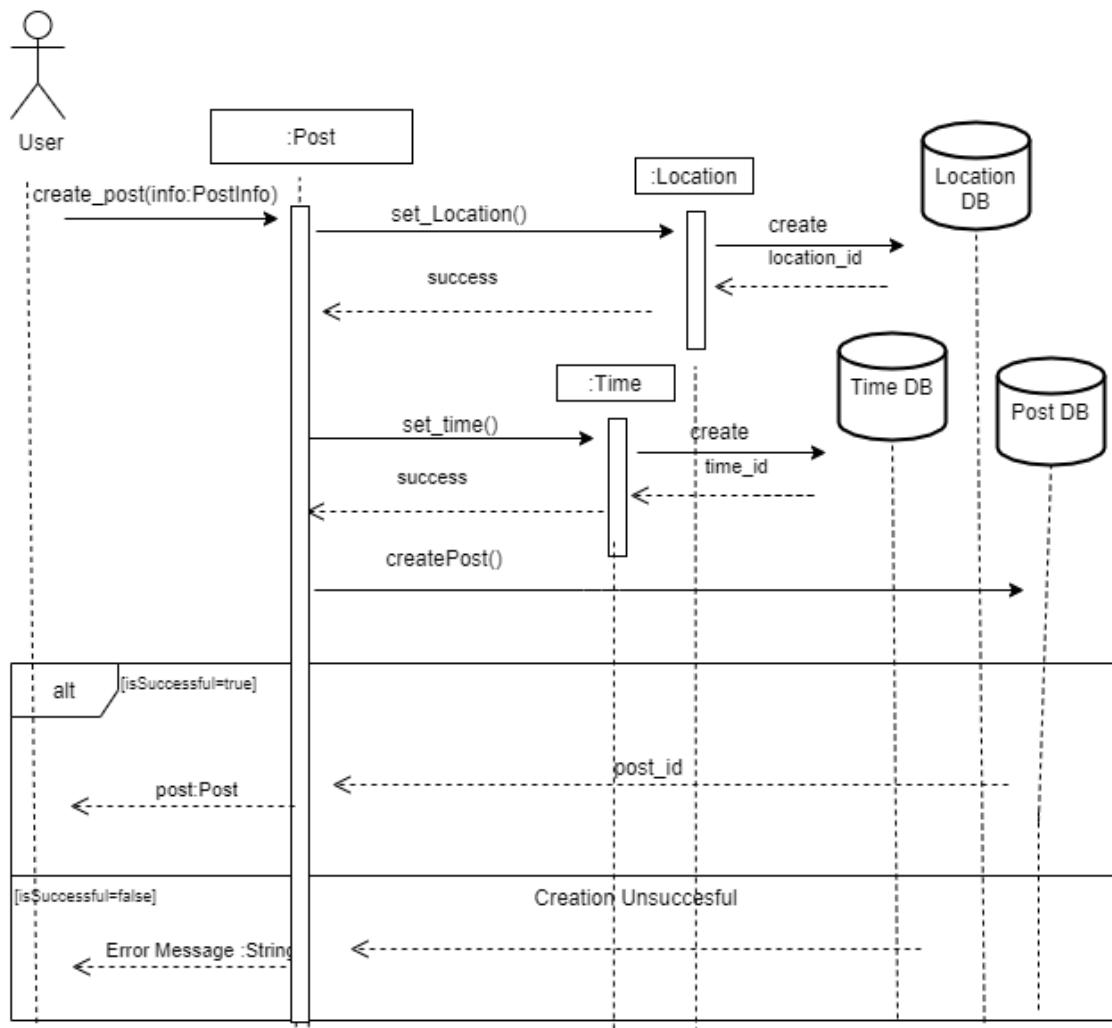
Actor: Registered User or Admin

Preconditions:

- User is registered to the application.

Actions:

- Clicks to create post button



Scenarios and Mockups

Creating Posts Scenario & Mockup

Persona



Nuri Bilge Ceylan is 62 years old. He is graduated from Boğaziçi University, Department of Electrical and Electronics Engineering.

However, he is not an engineer, he is a film director, screenwriter and photographer.

He was not a bright student but he was passionate about art.

He improved himself in cinema, art and philosophy thanks to cinema club and university's library.

He wants to inspire future generations by sharing his memories and experiences.

Preconditions

- User is signed up and logged in.
- Registered user is on the main page.

Goals

- Nuri Bilge Ceylan wants to post a story.
- He wants to create a post and add multimedia to it.

Acceptance Criteria

- **1.1.2.4.1:** Registered users shall be able to create posts by selecting a location.
- **1.1.2.4.2:** Registered users shall allow location services to find their current location and add a story to it.
- **1.1.2.4.3:** Registered users shall be able to add pictures to their stories.

- **1.1.2.4.4:** Registered users shall be able to choose to share their posts via other social platforms after they create them.
- **1.1.2.4.5:** Registered users should add date and time to their posts as they remember their memories. This date and time can be in the form as the registered user remembers it. They can be entered at the desired specific level.
- **1.1.2.4.6:** Registered users shall be able to tag other people to their stories.
- **1.1.2.4.7:** Registered users shall be able to tag their stories with specific topics.
- **1.1.2.4.8:** There shall be a create post button in both android and web at the end of the main page.

Scenario and Mockups

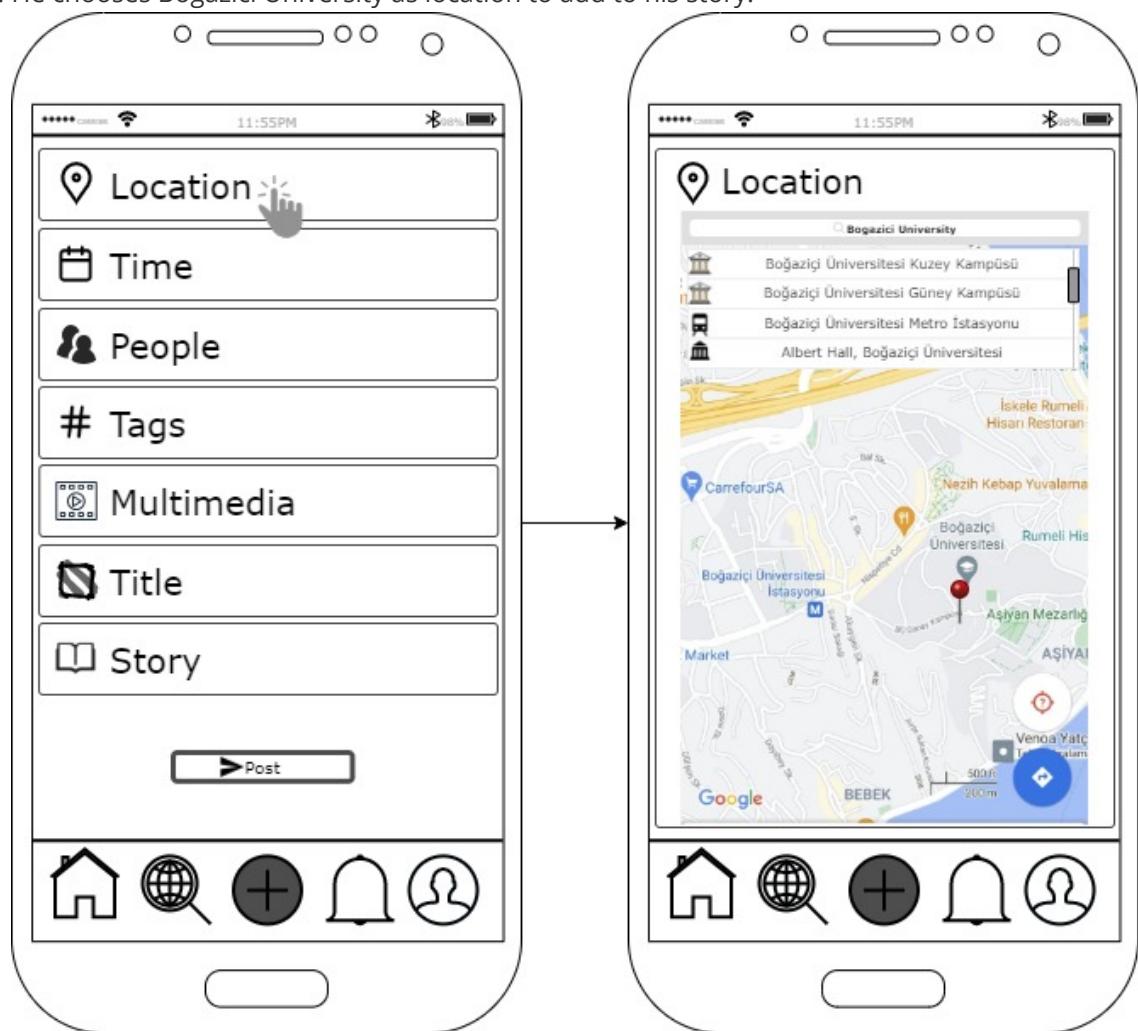
Nuri Bilge Ceylan returns home from a alumni meeting.
He lays down to get some rest, then a moment from university comes up to his head which reminds him cinema club of the university.
He starts to think about it, he thinks about his good memories and how he decided to be film director.

► Android

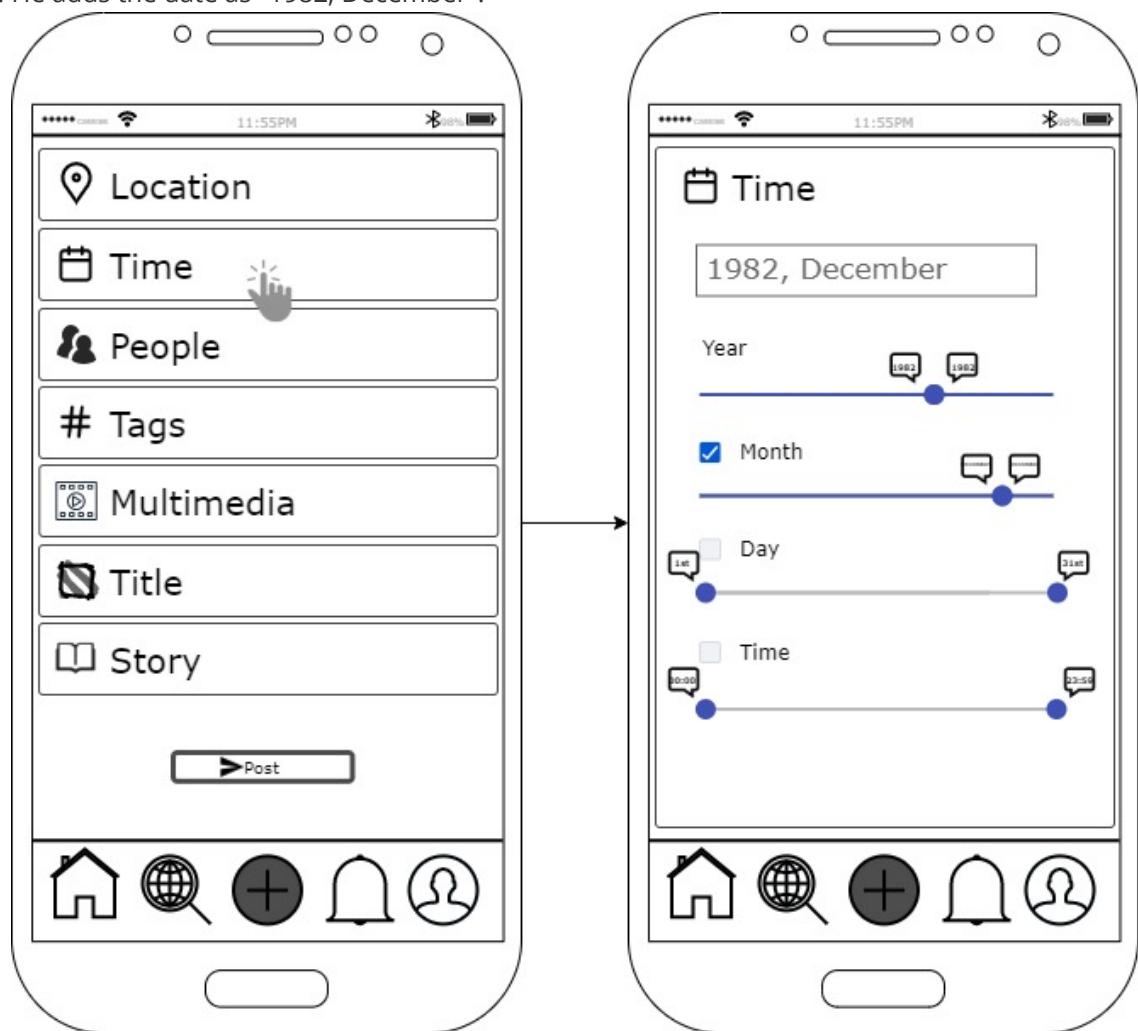
1. He is on the main page and clicks on the create post button.



2. He chooses Bogazici University as location to add to his story.



3. He adds the date as "1982, December".



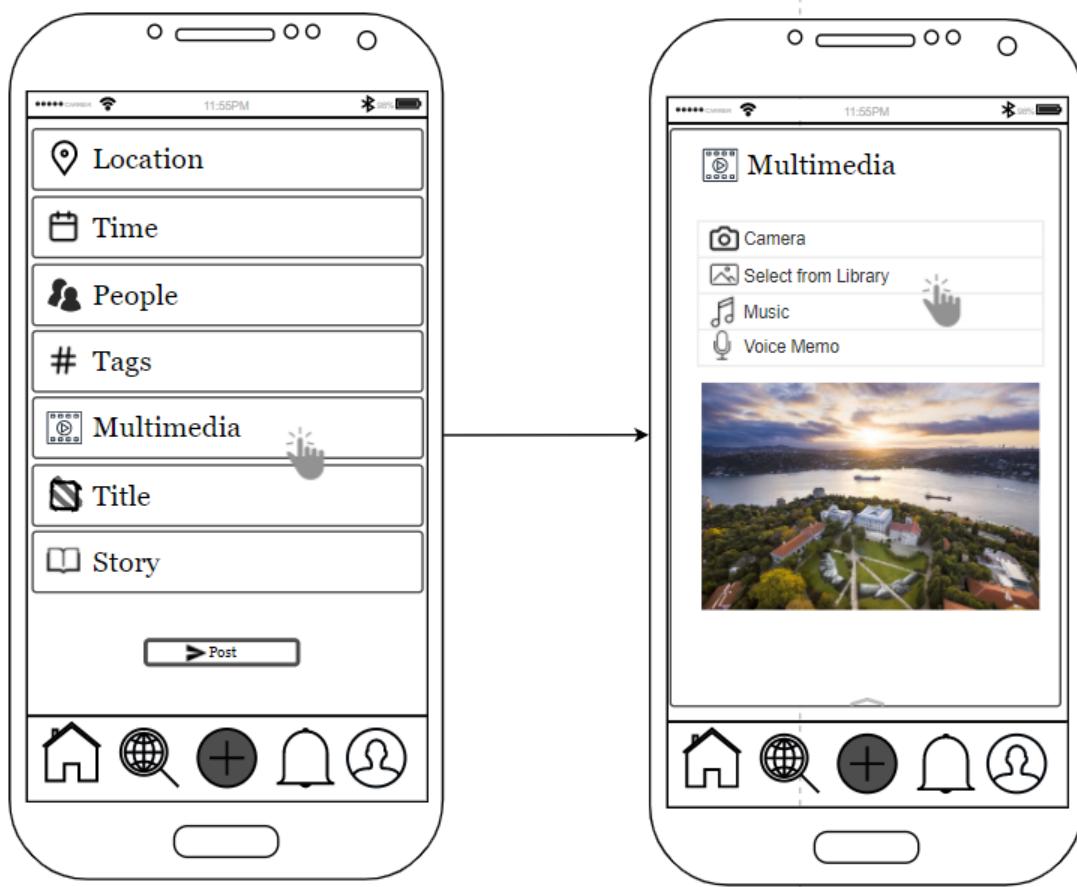
4. He then tags his friends to the story, so that they remember that day together.



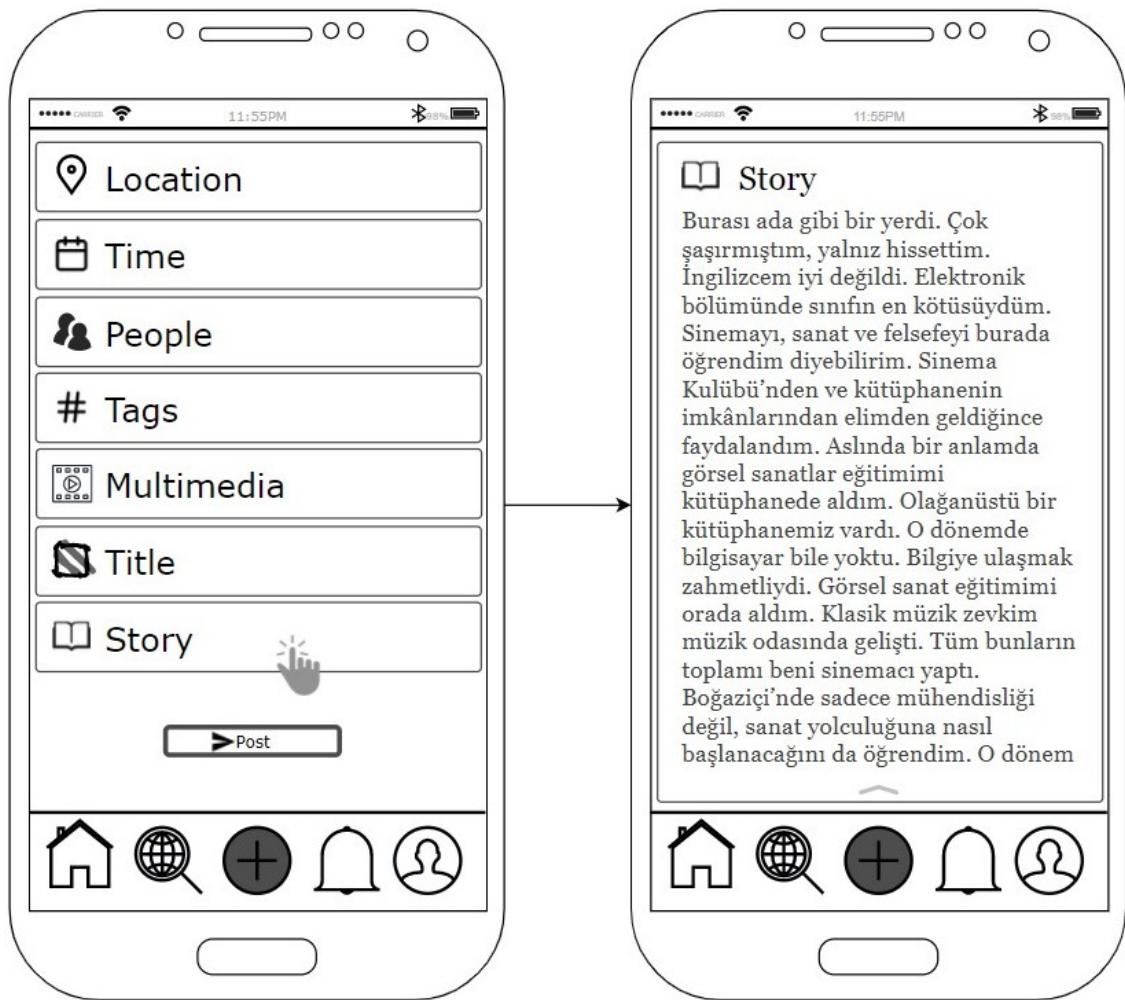
5. He tags specific topics to the story.



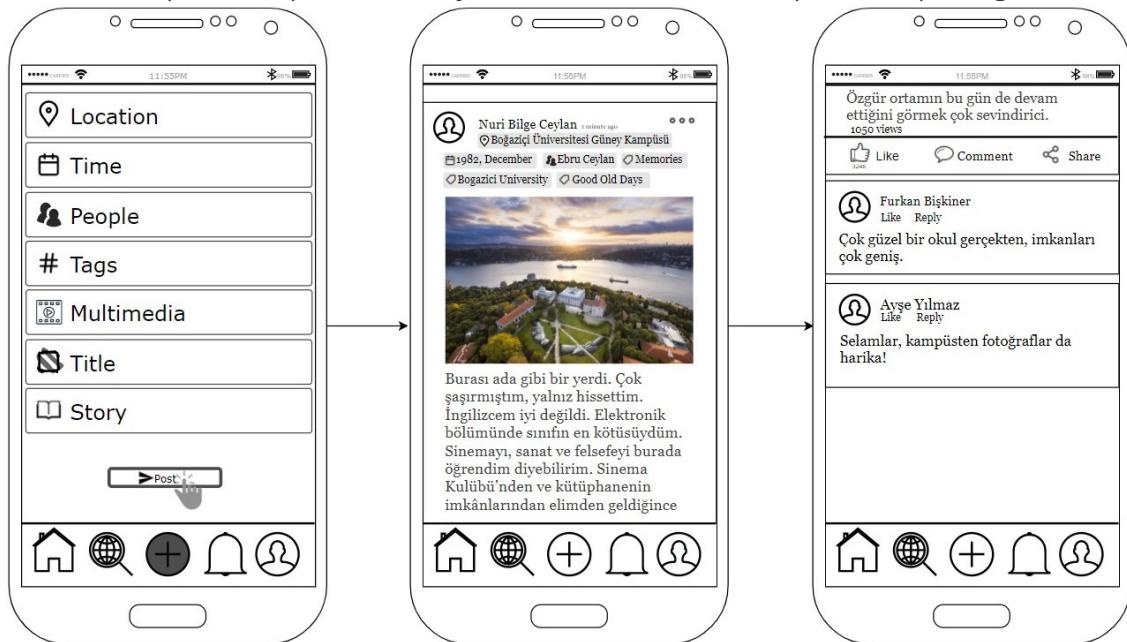
6. He clicks on "Multimedia" and chooses a picture of university.



7. He writes down his thoughts and memories about the university and adds a title.



8. He clicks on "post" and posts the story. Then, he reviews his own post after posting.



► Web

- He is on the main page and clicks on + button to post a story.

Group 9

Mehmet Beyoğlu 3 hours ago

İstanbul, Beyoğlu 1960-1970 Tag1 Tag2 Tag3 Turkish

I remember Beyoğlu as once it was. I had a powerful connection with there. Easy to say, I spent half my life over there. You know they say: "Eski Beyoğlu'na takım elbiseleri girilmemiş", there is a reason why that is said. Now I don't remember the exact year but... [Continue reading](#)

3246 17

Suma Rexx 3 hours ago

Roxxi Cinema (Roxxi) 2012-2017 Lisa Kadıköy Turkish Emre Kadıköy

Hic unutmuyorum, lise yıllarında okuldan çıkmış, Şişhane'den Kadıköy'e geyerdim. Önce vapurda atlığım o temiz hava ve arkadaşlarla güldüğümüz, eğlendığımız anılar gelir akıma. Vapurdan sonra da sonradık hep birbirinize, sizin yoksa hemen Roxxi'ye giderik. Cumas akşamı bir film izlemeden evlerimize doğruluyoruz. Ne yazık ki kapasiteli, amları kaldı sadice. Sadice benim değil, tüm eski İstanbul'un aması...

3246 17

You are signed in as:
Nun Bilge Ceylan

+

2. He is on the creation page. He specifies the information of the story. (Specifying the information steps are the same ones for mobile app)

The screenshot shows a web-based application for creating stories. At the top, there's a header with 'Group 9', a search bar, and navigation icons. The main content area has a title 'Boğaziçi' and a text input box containing a story about a trip to Boğaziçi University. To the right, there are five input fields with icons: 'Location' (location pin), 'Time' (calendar), 'People' (two people icon), 'Tags' (hash tag), and 'Multimedia' (camera icon). Below these is a sign-in message: 'You are signed in and posting as: Nuri Bilge Ceylan'. On the far right is a large circular 'Post' button with a right-pointing arrow.

Group 9

www.group9.com

Search Group 9

Boğaziçi

Burası ada gibi bir yerdı. Çok şaşırmışım, yalnız hissettim. İngilizcem iyi değildi. Elektronik bölümünde sınıfın en kötüsüydüm. Sinemayı, sanat ve felsefeyi burada öğrendim diyebilirim. Sinema Kulübü'nden ve kütüphanenin imkânlarından elimden geldiğince faydaladım. Aslında bir anlamda görsel sanatlar eğitimimi kütüphanede aldım. Olağanüstü bir kütüphanemiz vardı. O dönemde bilgisayar bile yoktu. Bilgiye ulaşmak zahmetliydi. Görsel sanat eğitimi orada aldim. Klasik müzik zevkim müzik odasında gelişti. Tüm bunların toplamı beni sinemacı yaptı. Boğaziçi'nde sadece mühendisliği değil, sanat yolculuğuna nasıl başlanacağını da öğrendim. O dönemde de Boğaziçi'nde yasak yoktu, her şey serbestti. Özgür ortamın bu gün de devam ettiğini görmek çok sevindirici.

Location

Time

People

Tags

Multimedia

You are signed in and posting as:
Nuri Bilge Ceylan

Post

3. He clicks the post button and now is on the page of the published story.

The screenshot shows a social media post on a platform like Facebook or LinkedIn. The post is titled "Group 9" and includes a profile picture of Nuri Bilge Ceylan. The post text is a memoir about her time at Boğaziçi University, mentioning various campus buildings and activities. Two photos of the university's campus are included: one aerial view of the Bosphorus and a building, and another of a large, ornate building. Below the post are two comments from other users: "Furkan Bışkınler" and "Ayşe Yılmaz". A comment input field at the bottom right says "Type your comment here!"

References

- <https://www.hurriyet.com.tr/gundem/vapur-sefer-saatlerinde-degisiklik-40839878>
- <http://evvel.org/rexx-reks-sineması-kapandi>
- <http://www.leblebitozu.com/eski-beyoglunu-anlatan-mutlaka-okumaniz-gereken-11-oyku/>
- https://haberler.boun.edu.tr/tr/haber/dunya-kentlerini-gezen-duvarlarin-otesinde-projesi-bo_gazicinde
- <https://tanitim.boun.edu.tr/nuri-bilge-ceylan>
- https://tr.wikipedia.org/wiki/Nuri_Bilge_Ceylan
- https://universityinnovation.org/wiki/School:Boğaziçi_University
- https://en.wikipedia.org/wiki/Boğaziçi_University

Viewing Posts Scenario & Mockup

Persona

Alex is a 28 years old software developer working at Google. He is a hardworking person but he often gets distracted.

Although Alex loves his job, he can easily get bored at work. Moreover, he sometimes gets tired of the heavy workload.

He is a long time registered user of our app, so when he gets bored or tired, he opens the app and reads stories of his friends and the other people that he follows.

Preconditions

- Alex is a registered user of our app.
- Alex has logged in to the app and his login credentials are saved.
- Alex follows some people, places and tags.

Goals

- To end his boredom, Alex wants to view location stories of people/places/tags that he follows, he also wants to see details and comments of the posts that he is interested in.
- He wants to interact with the posts by commenting on them and liking them.
- He also wants to help the community by reporting inappropriate content and marketing.

Acceptance Criteria

- **1.1.2.5.2** Registered users shall be able to view posts of the registered users that they follow.
- **1.1.2.9.4** Posts from followed registered users or posts that contain the followed places or tags will show up on the registered user's main page.
- **1.1.2.7.1** Registered users shall be able to leave comments on the posts.
- **1.1.2.7.3** Registered users shall be able to like posts.
- **1.1.2.12.1** Registered users shall be able to report registered users, posts, or comments that they seem inappropriate.

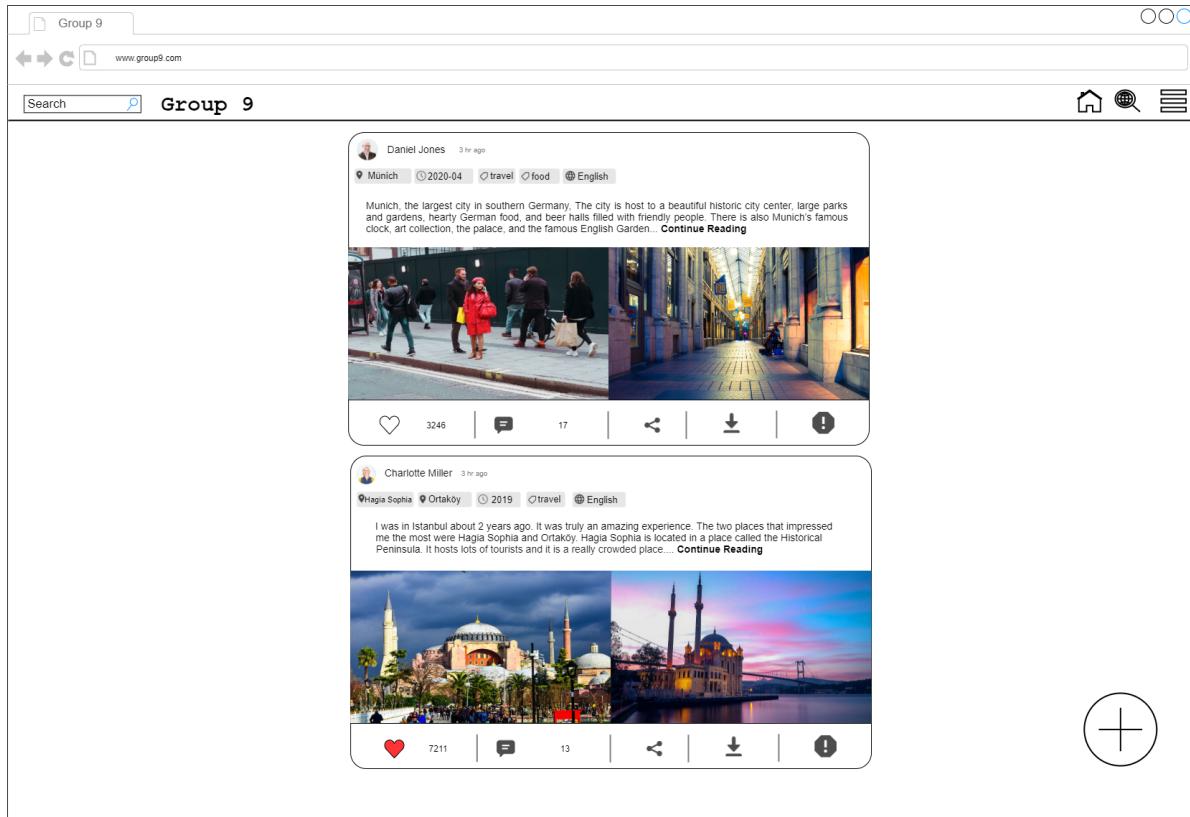
Scenario

1. After he's confident his manager is not around, Alex opens the app on his phone or work computer.
2. As he has already logged in, he is redirected to the main page where he can see the posts of the registered users, places, and tags that he follows.
3. He skims through the posts, likes the posts of his friends because he is a good friend.
4. He sees a highly rated story from his friend in which she talks about her trip to Istanbul two years ago. Only two photos and the first few lines of the post are shown on the main page.
5. He clicks on the post to see its full text, images, and comments.
6. After reading the post, looking at the images, and reading the comments; he comments about the Hagia Sophia photo that she took.
7. He also likes the story, because he enjoyed reading her experiences and also he liked the photos that she took .
8. He returns to the main page. While scrolling, he sees a post that has an advertisement.
9. He reports it to the admins.

Mockups

► Web

Homepage



Post Details

Group 9

www.group9.com/post/123456

Search Group 9

Charlotte Miller 3 hr ago

Hagia Sophia Ortaköy 2019 Travel English

I was in Istanbul about 2 years ago. It was truly an amazing experience. The two places that impressed me the most were Hagia Sophia and Ortaköy. Hagia Sophia is located in a place called the Historical Peninsula. It hosts lots of tourists and it is a really crowded place. The mosaics inside the Hagia Sophia drew my attention. A place worth seeing. There are many cafes and restaurants in Ortaköy. I sat in a very cute cafe with sea view in Ortaköy. I ate baked potato, which the Turks call Kumpir, in Ortaköy. Ortaköy Mosque view was incredible.

Lily Roberts 3 hr ago

I visited Istanbul about 10 years ago, it was a really exciting trip.

George Taylor 3 hr ago

You take nice photos.

Emily Evans 3 hr ago

I also sat in a very nice cafe in Ortaköy.

Sarah Williams 3 hr ago

Very nice photos. I felt like I was in Istanbul

Hagia Sophia looks imposing. I am planning to visit one day too.

7211

Share

Download

Report

A large image of the Hagia Sophia dome and surrounding minarets under a cloudy sky. To its right is a smaller image of the Ortaköy Mosque at dusk or night, illuminated with lights reflecting on the water.

▶

Reporting

Group 9

www.group9.com

Search Group 9

Jacob Davis 3 hr ago
New York 2021 English

I want to share with you an incident that happened to me at a McDonald's branch in New York last year. McDonald's is more than a restaurant. It's a symbol for everything burger-related, even though no one I know truly loves their burgers. Actually, its reputation for nearly everything... [Continue Reading](#)



Reporting Reason:
Marketing
Inappropriate Content

Thomas Wilson 3 hr ago
Canada 2018 Travel English

There are high mountain ranges, sparse plateaus, desert-like areas, forested valleys, stunning lakes, seemingly endless prairies, modern cities, as well as a tremendous assortment of flora and fauna. Not to mention, there's a Tim Hortons in almost every city and town across the country... [Continue Reading](#)



766 8

+

Group 9

www.group9.com

Search Group 9

Jacob Davis 3 hr ago
New York 2021 English

I want to share with you an incident that happened to me at a McDonald's branch in New York last year. McDonald's is more than a restaurant. It's a symbol for everything burger-related, even though no one I know truly loves their burgers. Actually, its reputation for nearly everything... [Continue Reading](#)

This post is actually an advertisement disguised as a location story. It is promoting the brand McDonald's.

Thomas Wilson Canada

There are high mountains, endless prairies, moose, Tim Hortons in almost

Report

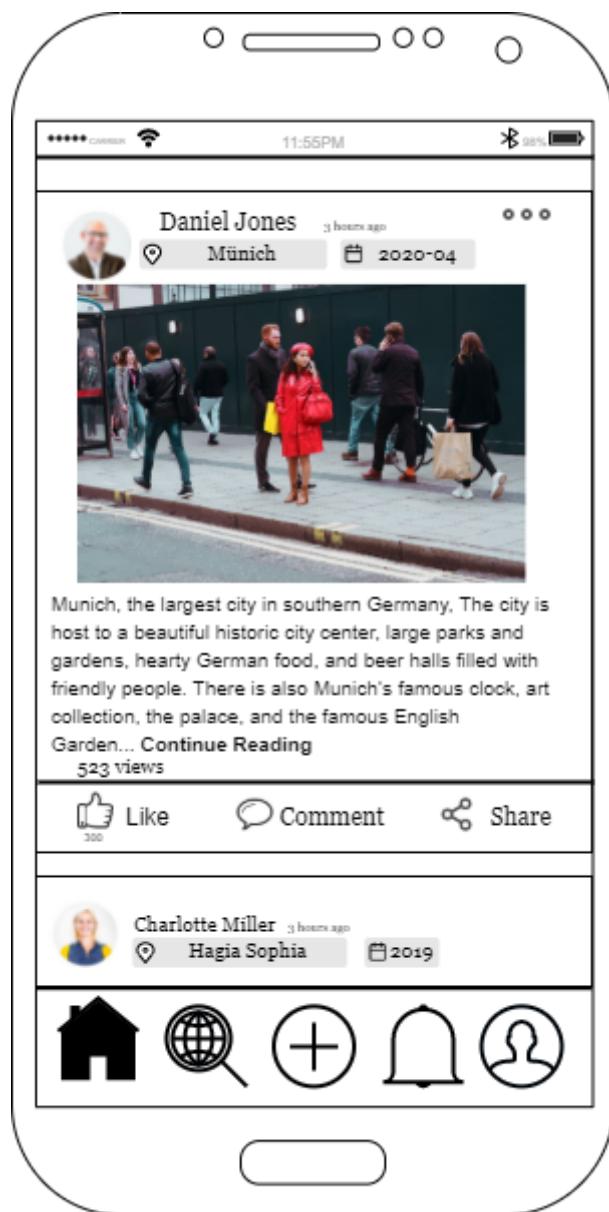
Reporting Reason:
Marketing
Inappropriate Content

+

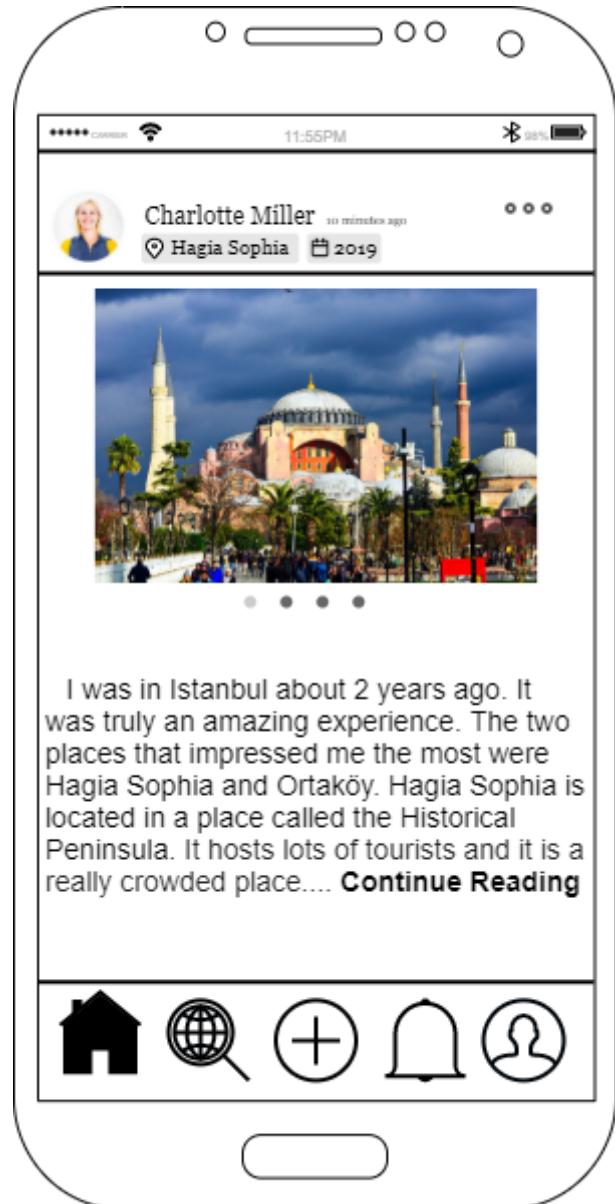
The reporting overlay contains the following text:
This post is actually an advertisement disguised as a location story. It is promoting the brand McDonald's.
Report
Reporting Reason:
Marketing
Inappropriate Content

► Android

Homepage

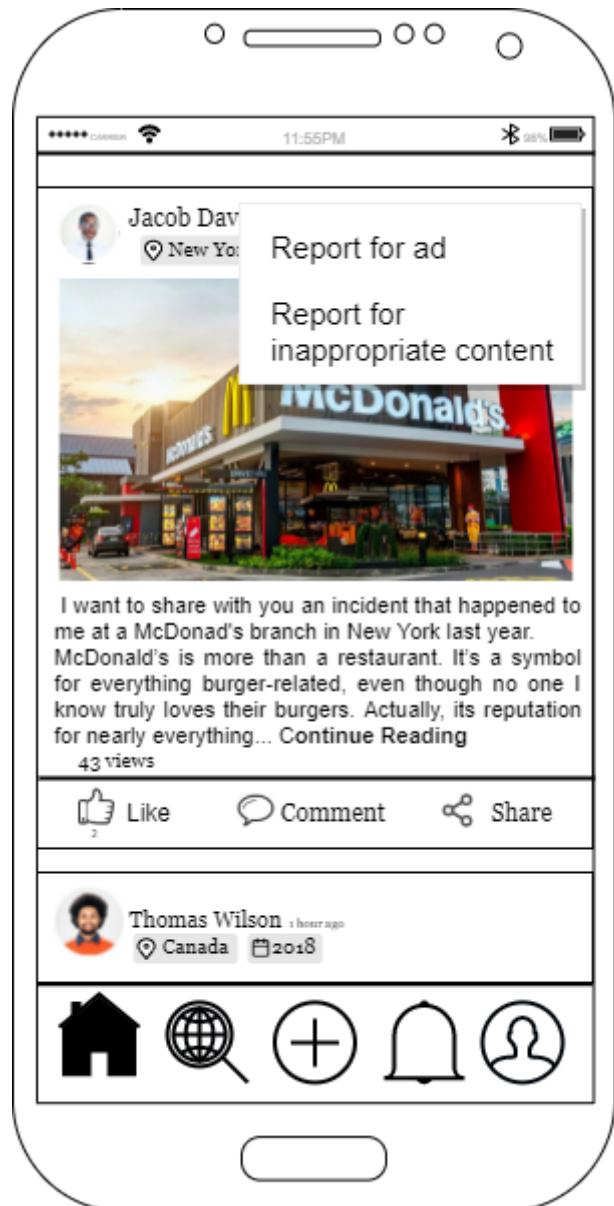


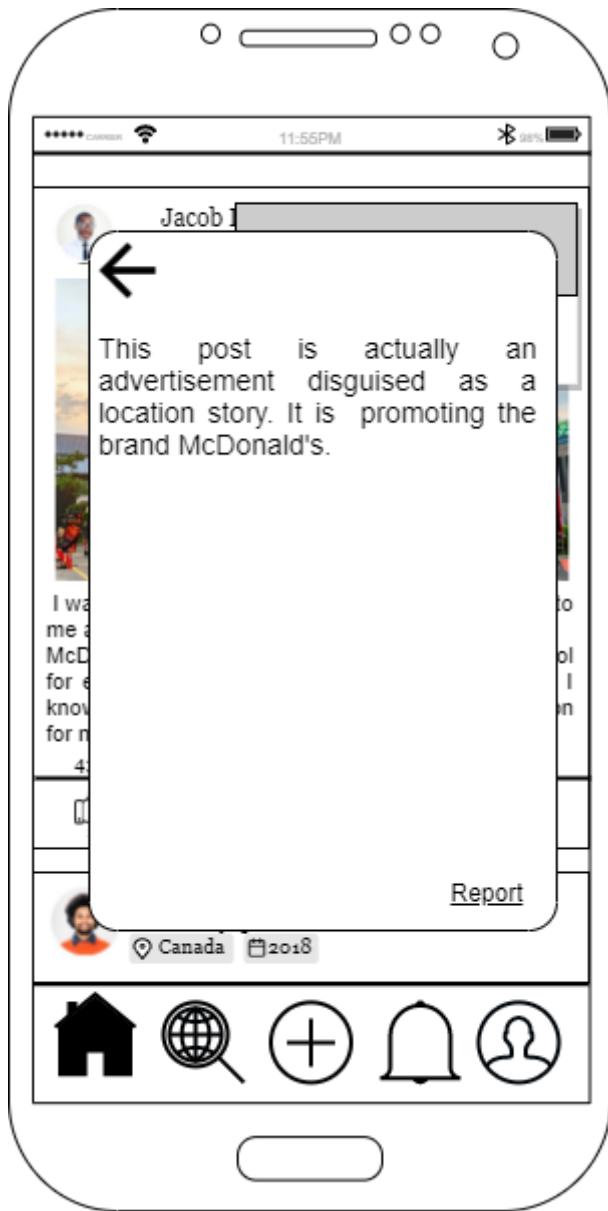
Post Details





Reporting





References

- Image References

[Profile photos](#)

[Istanbul picture](#)

[Hagia Sophia picture](#)

[First ad](#)

[Second ad](#)

[First picture of first Homepage post](#)

[Second picture of first Homepage post](#)

[First picture of second Reporting post](#)

[Second picture of second Reporting post](#)

[Münich](#)

[McDonald's](#)

[Canada](#)

Searching and Filtering Scenario and Mockup

Persona

Mehmet Kaya is 22 years old. He studies Political Science at Boğaziçi University.

Mehmet loves basketball, literature and history. He is very curious and intelligent.

He likes to learn about other people's life experiences.

For some time, he has been wondering how student life at Boğaziçi University has changed in years.

As he is interested in sports, he needs to answer some questions like:

1. What type of sports events were held?
2. How students participated in these events?
3. How did sports grounds at the university change?

Preconditions

- Not signed up

Acceptance Criteria

- 1.1.1.3.* Guest Requirements
- 1.1.2.4.5 Registered users shall be able to add date and time to their posts as they remember their memories. This date and time may be in the form as the registered user remembers it.
- 1.1.2.5.1 Users shall be able to view public posts on a location selected by them
- 1.1.2.6.2 Users shall be able to search for posts posted at a specific time and date using search bar.
- 1.1.2.6.3 Users shall be able to filter posts based on location, date and specific labels.
- 1.2.1.2 System shall request name, surname, e-mail address, username and password from guests who sign up.
- 1.2.1.3 Registered users shall verify their email addresses after signing up.
- 1.2.6.4 A time slider shall be available in order to see posts in different dates and times.

Goals

- Wants to learn about past sports events at Boğaziçi University by reading posts.
- Wants to comment on a post.

Scenario

Mehmet was born a researcher, so he visits the library hoping to find some information on the topic. Unfortunately, he only finds boring documents.

He hears about the app from a friend, but wants to check it before signing up.

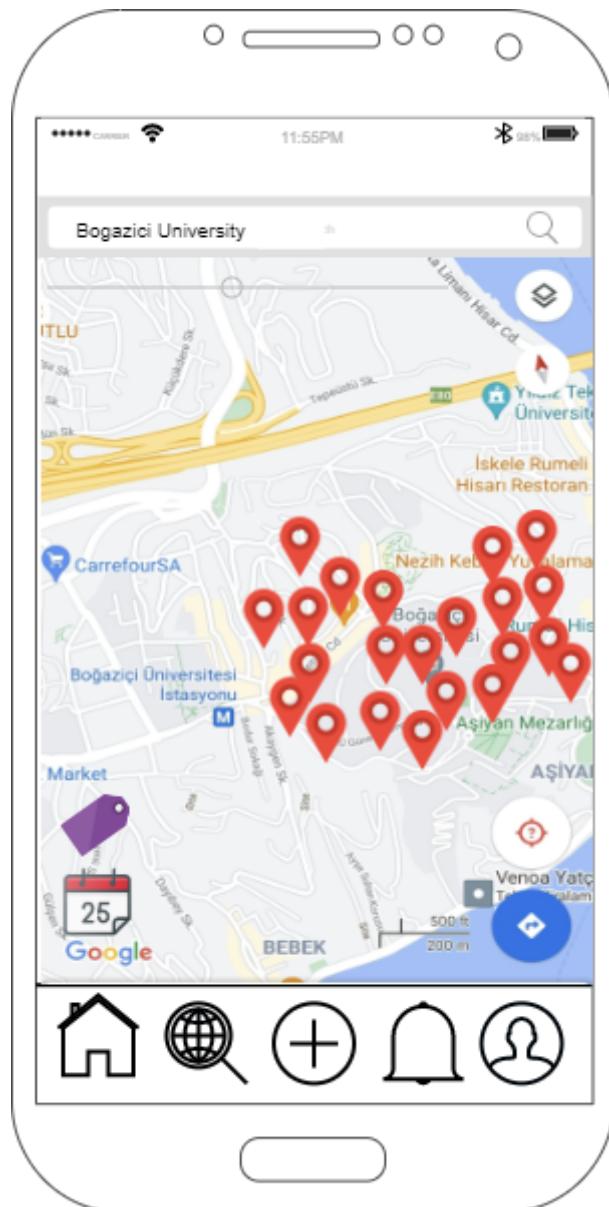
1. Mehmet wants to see the posts at Bogazici university, so from explore page, he zooms in to see Bogazici University closer.
2. He realizes that there are too many stories posted from Bogazici University, so, he decides to use filters to see the posts that interest him more.
3. Mehmet uses the sports tag since he is interested in basketball first.
4. But again, Mehmet is not happy with the results, there are too many of them.
5. Mehmet then uses the slider to filter the dates of the stories, he wants to see the stories from 2019 when there were heaps of friendly football games between universities.
6. He then clicks to see the details of a story from the results and starts reading it.

7. He gets goosebumps while reading the story of the Sultans game wishing to be a member of that team, and wants to like the post.
8. However because he is a guest user system does not allows him to like a post and navigates him to the sign-up page.
9. He then enters his name, surname, username, e-mail and password.
10. After that, the e-mail verification page is opened.
11. Mehmet opens his mailbox, then sees the e-mail coming from the site.
12. He then clicks the link for verification and the site navigates him to the story page, he can like it now!

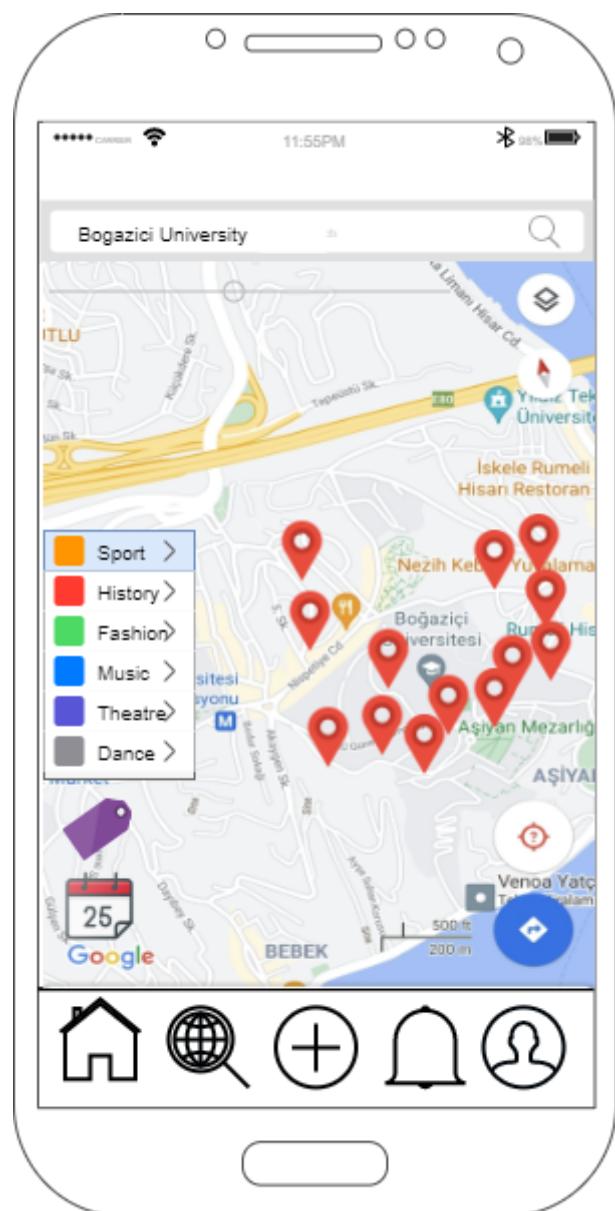
Mockups

- Android Mockup

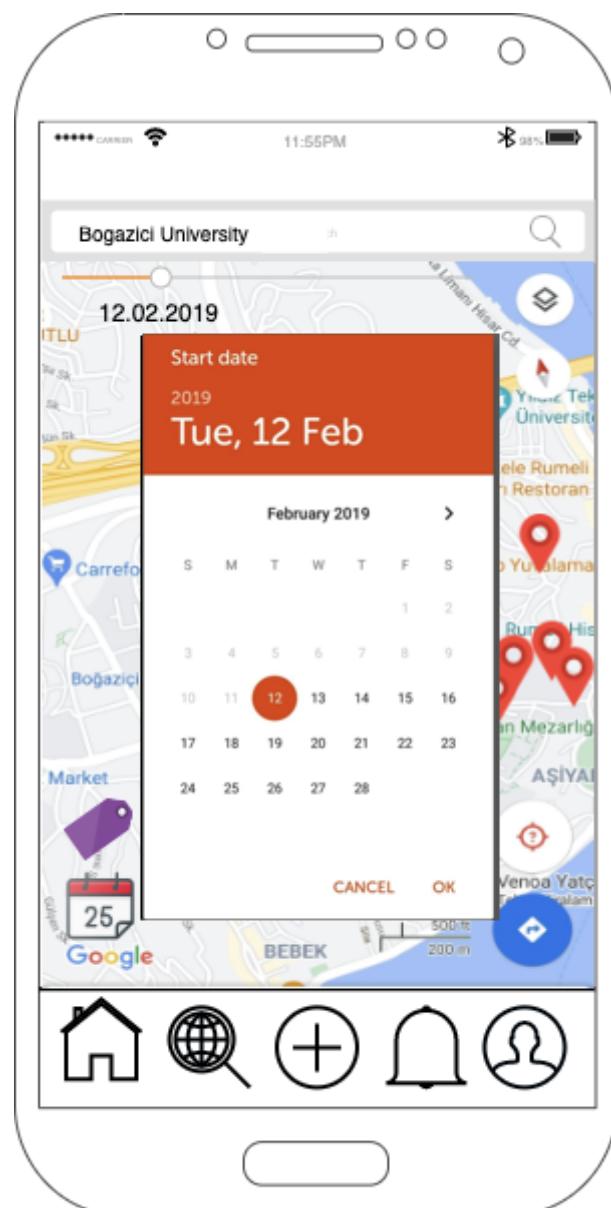
Mehmet starts exploring posts.



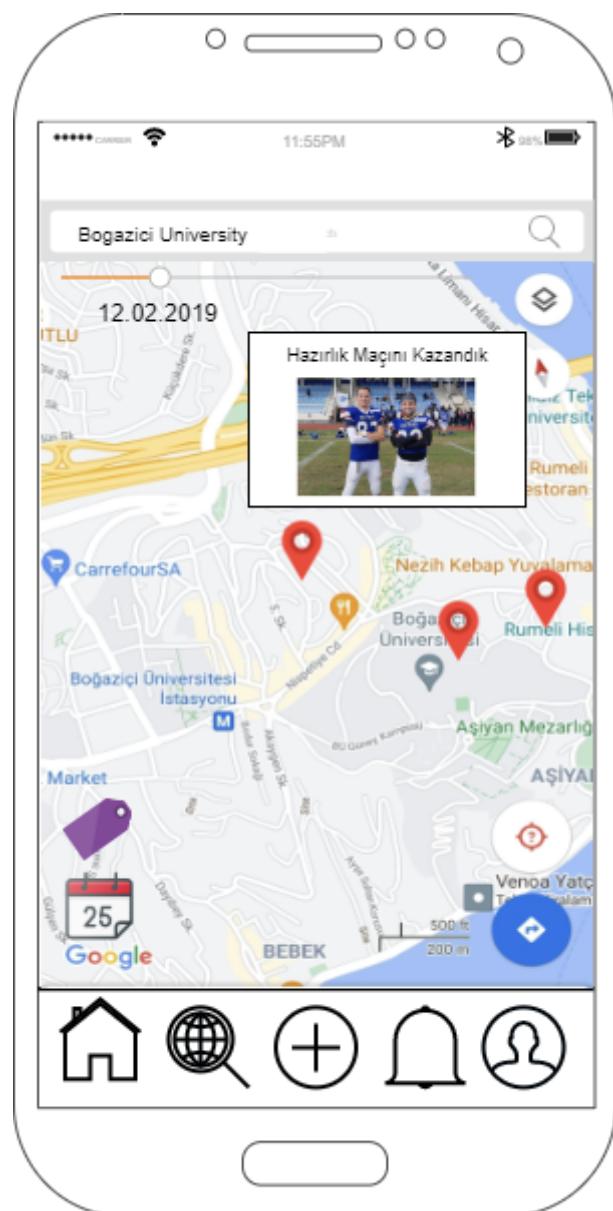
Mehmet filters sports stories.



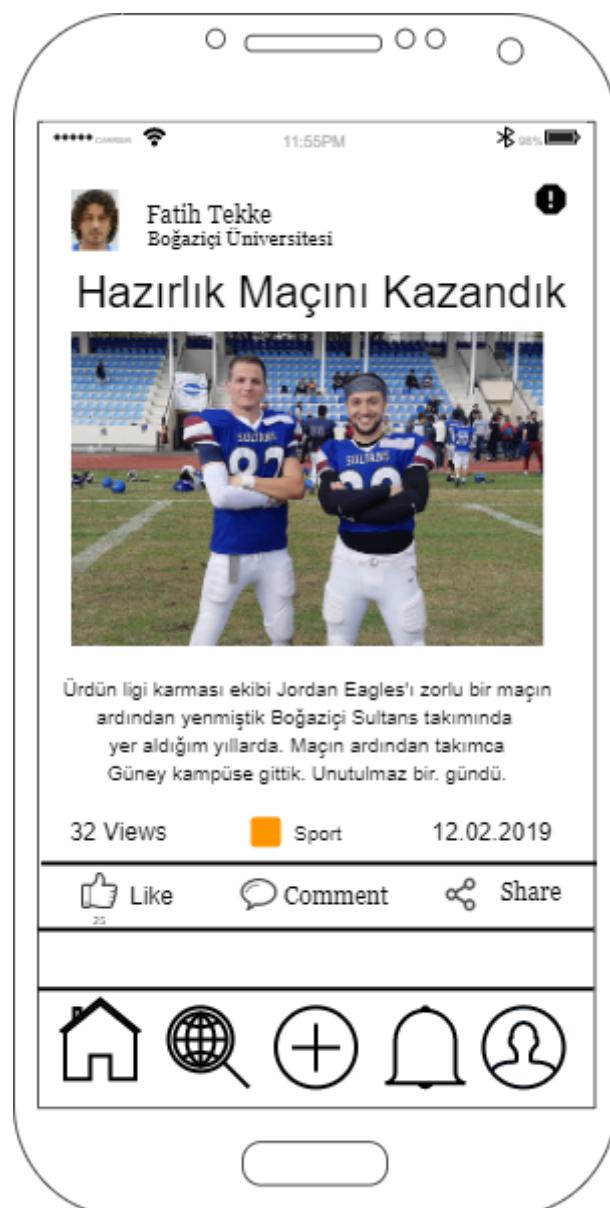
Mehmet uses calendar and slider to filter by date.



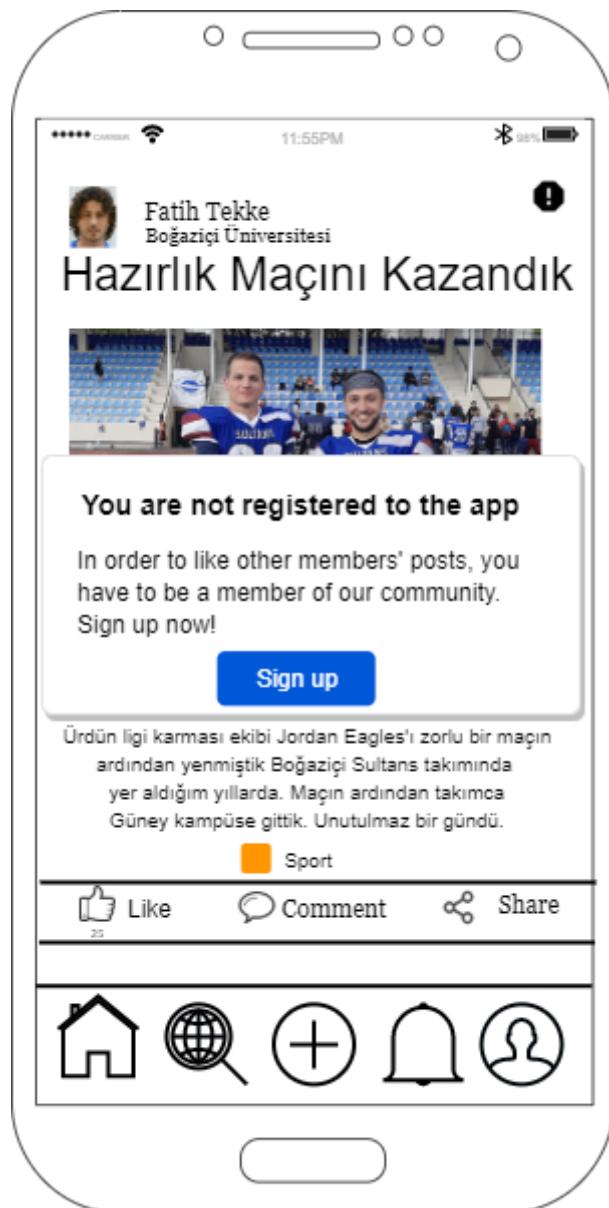
Now, there are 3 relevant posts left. Mehmet clicks on one of them to see the preview.



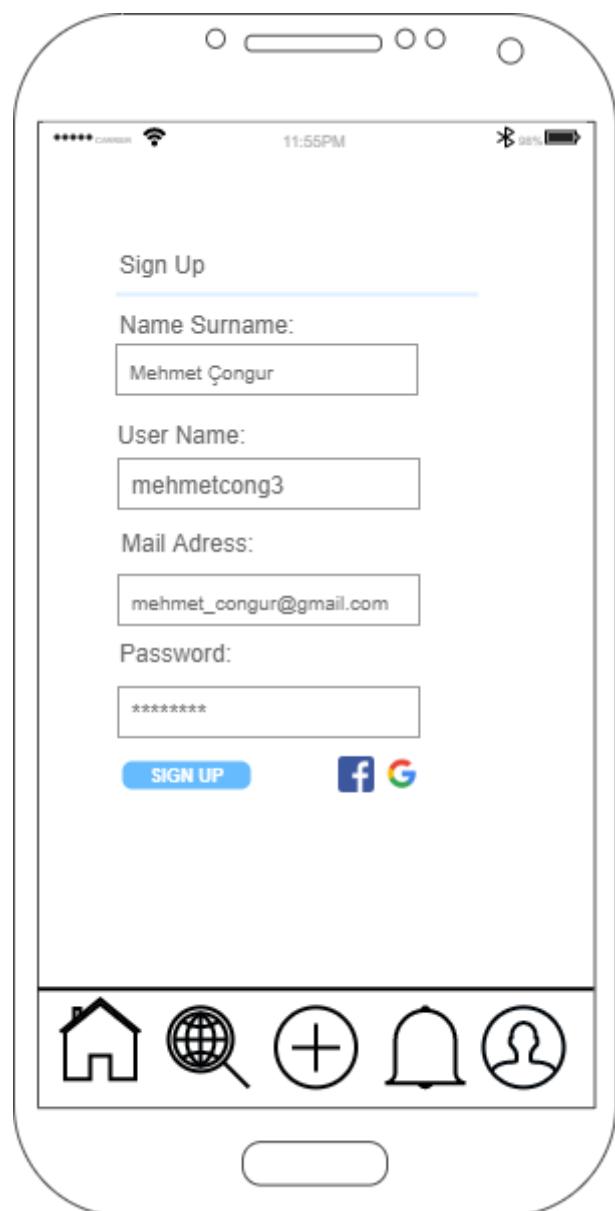
Mehmet clicks on the post again to see it.



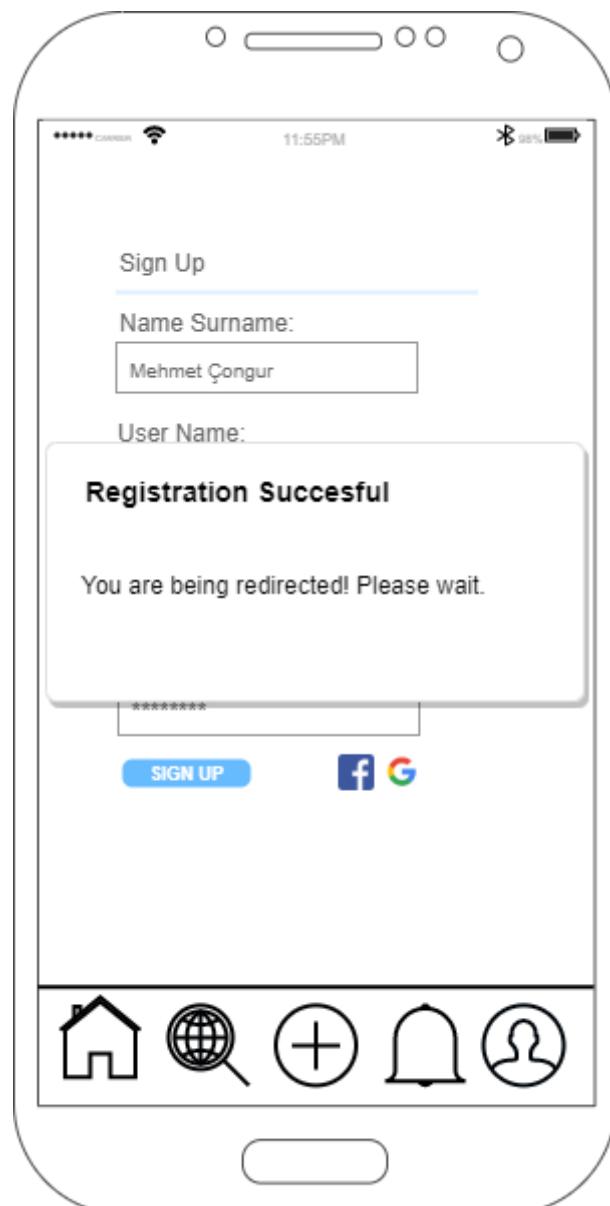
He wants to make a comment and clicks the comment button. However, he's a guest. He has to sign up first.



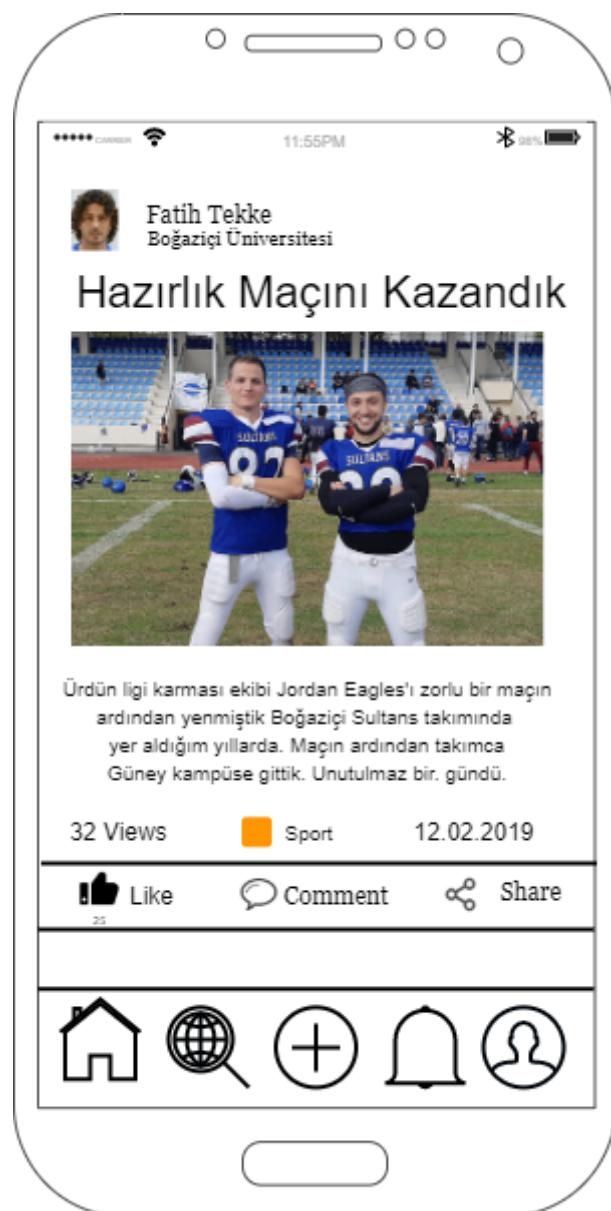
He fills in the fields to sign up.



After signing up, he is redirected to the post.

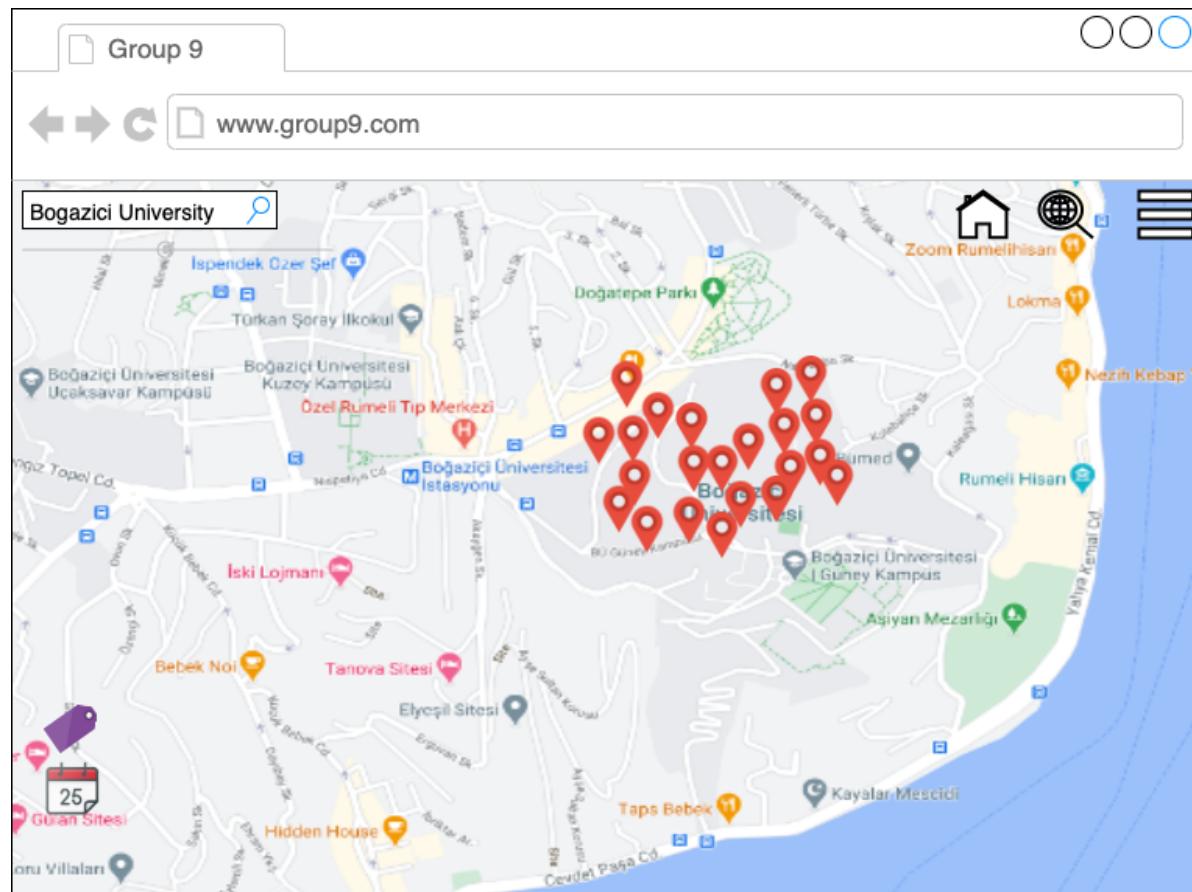


He is now able to like the post.

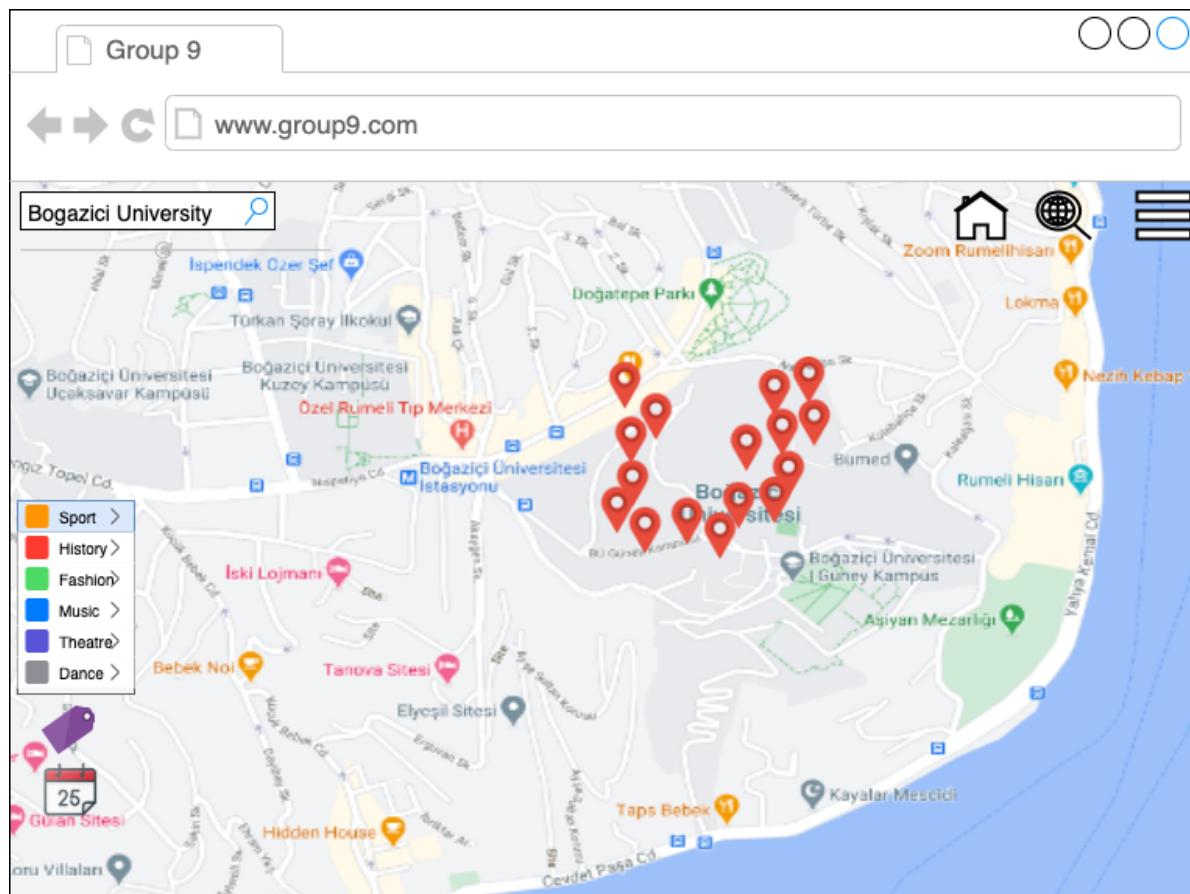


► Web Mockup

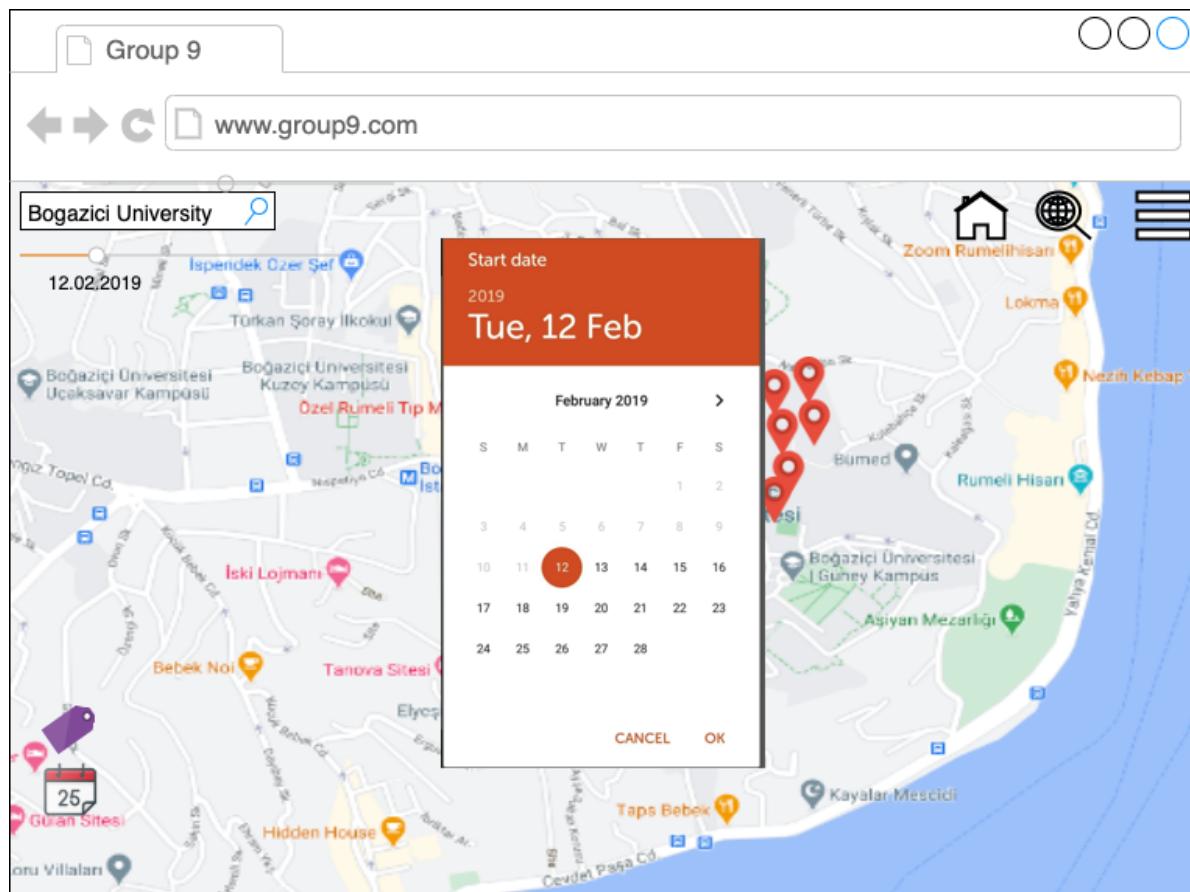
Mehmet starts exploring posts.



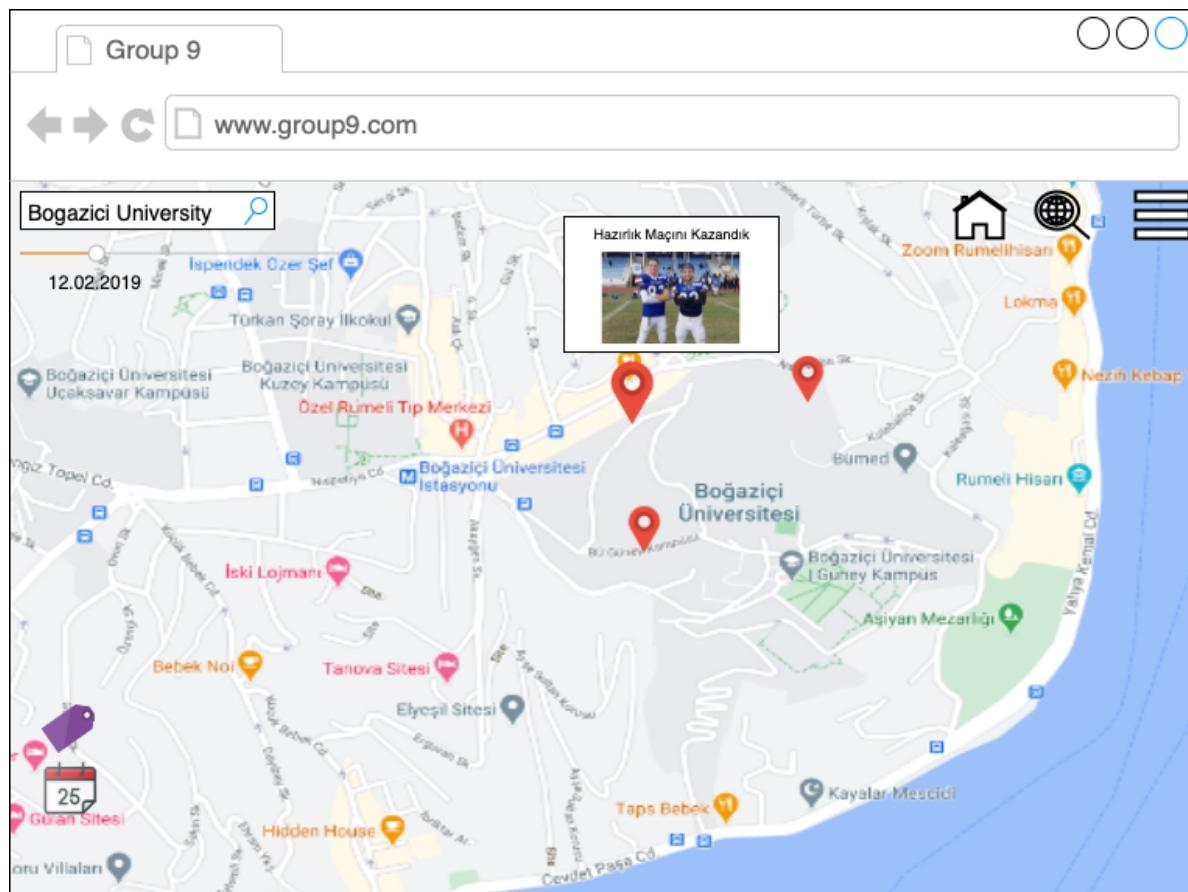
Mehmet filters sports stories.



Mehmet uses calendar and slider to filter by date.



Now, there are 3 relevant posts left. Mehmet clicks on one of them to see the preview.



Mehmet clicks on the post again to see it.

Group 9

www.group9.com

Bogazici University

Hazırlık Maçını Kazandık



Fatih Tekke
Boğaziçi Üniversitesi

Sport

Ürdün ligi karması ekibi Jordan Eagles'i zorlu bir maçın ardından yenmişik Boğaziçi Sultans takımında yer aldığım yıllarda.
Maçın ardından takımca Güney kampüse gittik.
Unutulmaz bir gündü.

Like Comment Share

32 Views 12.02.2019

He wants to make a comment and clicks the comment button. However, he's a guest. He has to sign up first.

The screenshot shows a web browser interface with the following details:

- Top Bar:** Shows "Group 9" and three blue circular icons.
- Address Bar:** Shows "www.group9.com".
- Page Header:** Shows "Bogazici University" and a blue globe icon.
- Page Title:** "Hazırlık Maçını Kazandık"
- Image:** A photo of two men in football uniforms (numbered 07 and 22) standing on a field.
- Post Content:** "In order to like other members' posts, you have to be a member of our community. Sign up now!"
- Call-to-Action:** A blue "Sign up" button.
- Right Side:** Includes icons for home, search, and menu, along with some text in Turkish: "du bir maçın nda ittik."
- Bottom Left:** "32 Views" and "12.02.2019".

He fills in the fields to sign up.

Group 9

www.group9.com

Sign Up

Name Surname:
Mehmet Congur

User Name:
mehmetcong3

Mail Adress:
mehmet_congur@gmail.com

Password:

SIGN UP

After signing up, he is redirected to the post.

The screenshot shows a web browser window with the URL www.group9.com in the address bar. The browser interface includes a back/forward button, a refresh button, and three circular icons in the top right corner. The main content area displays a sign-up form with the following fields:

- Name Surname:
- User Name:
- Mail Adress:
- Password:

Below the form is a blue "SIGN UP" button. To the right of the form, there are social media sharing icons for Facebook and Google.

A modal dialog box is overlaid on the page, containing the following text:

Registration Successful
You are being redirected! Please wait.

He is now able to like the post.

Group 9

www.group9.com

Bogazici University

Hazırlık Maçını Kazandık



Fatih Tekke
Boğaziçi Üniversitesi

Sport

Ürdün ligi karması ekibi Jordan Eagles'i zorlu bir maçın ardından yenmişik Boğaziçi Sultans takımında yer aldığım yıllarda.
Maçın ardından takımca Güney kampüse gittik.
Unutulmaz bir gündü.

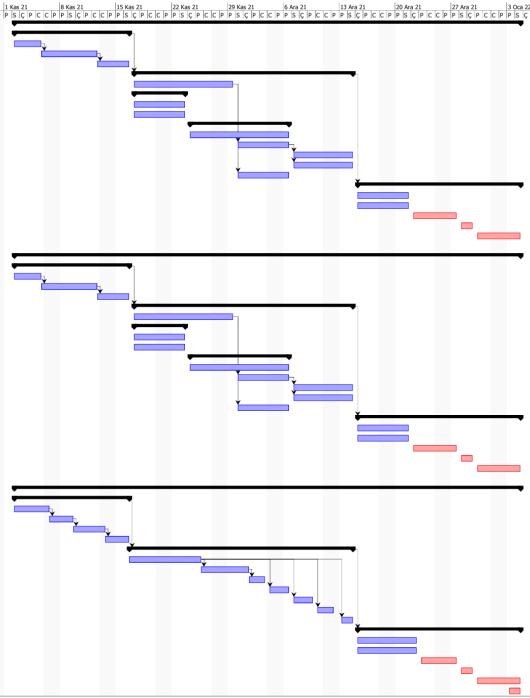
Like Comment Share

32 Views 12.02.2019

[Source of photo](#)

Project Plan

	Ad	Sıre	Başlıt	Bitime	Onisko	Kaynak Adın
1	Frontend	15,333	gönl... 01.11.2021 08:00	04.01.2022 17:00		
2	[Milestone 1]	3,647gönl... 01.11.2021 08:00		16.11.2021 17:00		
3	Creating Project	3,393gönl... 01.11.2021 08:00		16.11.2021 17:00		
4	Home Page	7gönl... 01.11.2021 17:00		12.11.2021 17:00	3	
5	Edit and Create Post	4gönl... 12.11.2021 17:00		16.11.2021 17:00	4	
6	[Milestone 2]	6,679gönl... 17.11.2021 08:00		14.12.2021 17:00	2	
7	Sign In, Sign Up, LogIn	2,049gönl... 17.11.2021 08:00		13.11.2021 17:00		
8	New Post	1,627gönl... 17.11.2021 08:00		23.11.2021 17:00		
9	Like Post	1,602gönl... 17.11.2021 08:00		13.11.2021 17:00		
10	Comment Post	1,601gönl... 17.11.2021 08:00		13.11.2021 17:00		
11	[Explore Page]	1,627gönl... 24.11.2021 08:00		16.11.2021 17:00		
12	Google Maps Integration	3gönl... 24.11.2021 08:00		16.11.2021 17:00		
13	User Profile Page (Private/Public/Profile)	1,601gönl... 30.11.2021 08:00		16.12.2021 17:00	7	
14	Save Post	2gönl... 01.12.2021 08:00		14.12.2021 17:00	13	
15	Edit Post	2gönl... 07.12.2021 08:00		14.12.2021 17:00	13	
16	Report Page	1,602gönl... 30.11.2021 08:00		16.12.2021 17:00	7	
17	[Post]	1,627gönl... 15.12.2021 08:00		04.01.2022 17:00	6	
18	Search Bar for Explore Page	1,601gönl... 15.12.2021 08:00		13.12.2021 17:00		
19	Filter for Explore Page	1,602gönl... 15.12.2021 08:00		13.12.2021 17:00		
20	Admin Interface	1,313gönl... 22.12.2021 08:00		27.12.2021 17:00		
21	Activity Stream	0,681gönl... 28.12.2021 08:00		12.1.2022 17:00		
22	Notification Menu	1,139gönl... 28.12.2021 08:00		04.01.2022 17:00		
23	[Backend]	15,333	gönl... 01.11.2021 08:00	04.01.2022 17:00		
24	[Milestone 1]	3,647gönl... 01.11.2021 08:00		16.11.2021 17:00		
25	Creating Project	3,375gönl... 01.11.2021 08:00		05.11.2021 17:00		
26	Home Page	7gönl... 05.11.2021 17:00		12.11.2021 17:00	26	
27	Edit and Create Post	4gönl... 12.11.2021 17:00		16.11.2021 17:00	27	
28	Deploy Backend	6,679gönl... 17.11.2021 08:00		14.12.2021 17:00	48	
29	Sign In, Sign Up, LogIn	2,049gönl... 17.11.2021 08:00		29.11.2021 17:00		
30	New Post	1,627gönl... 17.11.2021 08:00		13.12.2021 17:00		
31	Comment Post	1,602gönl... 17.11.2021 08:00		13.12.2021 17:00		
32	[Explore Page]	1,627gönl... 24.11.2021 08:00		16.11.2021 17:00		
33	Google Maps Integration	3gönl... 24.11.2021 08:00		16.11.2021 17:00		
34	User Profile Page (Private/Public/Profile)	1,601gönl... 30.11.2021 08:00		16.12.2021 17:00	30	
35	Save Post	2gönl... 01.12.2021 08:00		14.12.2021 17:00	36	
36	Settings	2gönl... 07.12.2021 08:00		14.12.2021 17:00	36	
37	Report Page	1,602gönl... 30.11.2021 08:00		16.12.2021 17:00	30	
38	[Post]	1,627gönl... 15.12.2021 08:00		04.01.2022 17:00	29	
39	Search Bar for Explore Page	1,602gönl... 15.12.2021 08:00		13.12.2021 17:00		
40	Filter for Explore Page	1,602gönl... 15.12.2021 08:00		13.12.2021 17:00		
41	Admin Interface	1,313gönl... 22.12.2021 08:00		29.12.2021 17:00		
42	Activity Stream	0,682gönl... 28.12.2021 08:00		04.01.2022 17:00		
43	Notification Menu	1,139gönl... 28.12.2021 08:00		04.01.2022 17:00		
44	[Backend]	15,333	gönl... 01.11.2021 08:00	04.01.2022 17:00		
45	[Milestone 1]	3,647gönl... 01.11.2021 08:00		16.11.2021 17:00		
46	Creating Backend Server	4,737gönl... 01.11.2021 08:00		16.11.2021 17:00		
47	Create User and Edit Post and Get Post	3gönl... 06.11.2021 17:00		19.11.2021 17:00	49	
48	Deploying Backend	4gönl... 13.11.2021 17:00		13.11.2021 17:00	50	
49	[Backend]	6,679gönl... 16.11.2021 17:00		14.12.2021 17:00	48	
50	Create User and Edit Post and Get Post	4,696gönl... 01.12.2021 08:00		19.12.2021 17:00		
51	Delete User	2gönl... 25.11.2021 17:00		19.12.2021 17:00	54	
52	Save Post for Users	2gönl... 01.12.2021 17:00		03.12.2021 17:00	55	
53	Reset Password	2,775gönl... 04.12.2021 08:00		16.12.2021 17:00	54	
54	Delete Comment and Comment	2,735gönl... 04.12.2021 08:00		12.12.2021 08:00	54	
55	Follow User	2gönl... 10.12.2021 08:00		12.12.2021 08:00	54	
56	Like and Comment Post	1,737gönl... 13.12.2021 08:00		14.12.2021 17:00	54	
57	[Search]	1,627gönl... 15.12.2021 08:00		04.01.2022 17:00	53	
58	Search Post, Post and User	2gönl... 15.12.2021 08:00		22.12.2021 17:00		
59	Filter Tag, Post and User	2gönl... 15.12.2021 08:00		22.12.2021 17:00		
60	Notification System	1gönl... 23.12.2021 08:00		27.12.2021 17:00		
61	Create Admin	0,681gönl... 03.01.2022 08:00		29.12.2021 17:00		
62	Delete Admin	5,379gönl... 30.12.2021 08:00		04.01.2022 17:00		
63	Final Deploy	0,682gönl... 03.01.2022 08:00		04.01.2022 17:00		



ProjectPlan451- Sayfa

Communication Plan

Attendees	Medium	Time	Purpose
All team members	Zoom	Every Tuesday @ 15:00 - 16:50	LAB Session
All team members	Discord, Trello	Every Tuesday @ 21:00	General Meeting
All team members	Whatsapp, Discord	Anytime	Urgent Communication
All team members	Discord, Trello	Thursday @ 21:00	Alternative General Meeting