COLUMBUS

CMPE 451 - Fall'21
FINAL MILESTONE REPORT
COLUMBUS
GROUP 7

Frontend
Abdulkadir Elmacı
Erencan Uysal
Merve Rabia Barın

Mobile
Onur Can Avcı
Ramazan Bulut

Backend
Hamza Akyıldız
Umut Kocasarı
Yağız Efe Şabanoğlu

Release Tag

https://github.com/bounswe/2021SpringGroup7/releases/tag/v0.9

12.01.2021

Table of Contents

1. Executive Summary **4**

Status of deliverables **4**

Status of all requirements **4**

Final release notes **7**

Information **7**

Overview **7**

Issue Summary **7**

Resolution **8**

Impacts **8**

Changes From Previous Milestones **8**

Final Project Evaluation: **8**

2. Summary of Individual Coding Work **8**

3. Core Functionality for the Scenario **17**

4. Standards **17**

Wikidata Integration **21**

5. API **21**

6. User Experience **49**

7. User Manual **70**

8. System Manual **70**

Minimum Requirements for Building and Running 70

Minimum Requirements for Frontend 70

Building the Images 70

Running the Images 71

Accessing the Running Images 71

Requirements 71

9. Project Plan **73**

10. Requirements **77**

1. Functional Requirements **79**

1.1 User Requirements 79

- 1.1.1 Profile 79
- 1.1.2 Posting 79
- 1.1.3 User to User Interaction 80

- 1.1.4 Searching 80
- 1.1.5 Recommendation 81

1.2 Guest Requirements 81
1.3 Admin Requirements 81
1.4 System Requirements 81

2. Nonfunctional Requirements 82

- 2.1 Portability 82
- 2.2 Accessibility 82
- 2.3 Performance 82
- 2.4 Reliability 82
- 2.5 Security 82
- 2.6 Legal and Ethical Issues 83
- 2.7 Implementation 83
- 2.8 Robustness 83

11. UML Software Designs 84

11.1. Use Case Diagrams 84

11.2. Class Diagrams 85

11.3. Sequence Diagrams 85

1. Executive Summary

1. Status of deliverables

The products that are delivered are:

- Frontend for the users and admins
- Backend for the users and admins
- A mobile app
- Manuals
  - System Manual
  - User Manual
- Project Plan
- Software Requirements Specification
- Software Design Documents
  - Use Case Diagrams
  - Class Diagram
  - Sequence Diagrams

The frontend is deployed here.
The backend is here and the swagger documentation can be found here.
For the mobile application APK you can find it here.

2. Status of all requirements

X : Completed
X : In progress
X : Not Started

1. Functional Requirements

1.1 User Requirements

1.1.1 Profile

1.1.1.1 Users shall be able to block other users.
1.1.1.2 Users shall be able to view any of their personal information (email, username and name, location, birthday if provided).
1.1.1.3 Users shall be able to edit any of their personal information (email, username and name, location, birthday if provided).
1.1.1.4 Users shall be able to see users that they follow and are followed by.
1.1.1.5 Users shall be able to delete their accounts.

1.1.2 Posting

1.1.2.1 Users shall be able to create posts with a story.
1.1.2.2 Users shall be able to add tags, photo, video and voice to their posts.
1.1.2.3 Users shall be able to add a location with its name or by specifying the location on the map interface.
1.1.2.4 Users shall be able to add multiple locations related to their story.

1.1.2.5 Users shall be able to specify time in their posts to declare exact time or time interval with a maximum of 10 years.

1.1.2.6 Users shall be able to edit their own posts.

1.1.2.7 Users shall be able to delete their own posts.

1.1.2.8 Users shall be able to like the post of their followings.

1.1.2.9 Users shall be able to comment on the post of their followings.

1.1.2.10 Users shall be able to view users who liked or commented to their posts.

1.1.2.11 Users shall be able to delete comments made to their own posts.

1.1.2.12 Users shall be able to reply to comments made by other users with comments.

1.1.2.13 Users should be able to save the posts of their followings.

1.1.2.14 Users should be able to pin comments made to their own posts to the top.

1.1.3 User to User Interaction

1.1.3.1 Users shall be able to view the posts of their following accounts.

1.1.3.2 Users shall be able to report other users posts, if the user has suspicion about its accuracy.

1.1.3.3 Users shall be able to follow other users.

1.1.3.4 Users shall be able to unfollow other users.

1.1.3.5 Users shall be able to view posts of other users that they follow or have public profile.

1.1.3.6 Users shall be able to view the followers and followings of other users that they follow or have public profile.

1.1.4 Searching

1.1.4.1 Users shall be able search other users by their username.

1.1.4.2 Users shall be able search locations by their name.

1.1.4.3 Users shall be able to specify the searched location on the map.

1.1.4.4 Users shall be able to do semantic search by using tags.

1.1.4.5 Users shall be able to filter the search results according to time.

1.1.5 Recommendation

1.1.5.1 Users shall be able to view recommended locations based on their system activity and current location.

1.2 Guest Requirements

1.2.1 Guests shall be able to create a new account by choosing a password (email confirmation) and providing email, username (name, location, birthday optionally).

1.2.2 Guests shall be able to login to the system by entering username and password if they already have an account.

1.2.3 Guests shall be able to only search and view locations.

1.3 Admin Requirements

1.3.1 Admins shall be able to login to the system by entering username and password.

1.3.2 Admins shall be able to review any content that is reported by users.

1.3.3 Admins shall be able to filter reported contents for the types of posts, comments,tags and accounts.

1.3.4 Admins shall be able to add users to Black List.

1.3.5 Admins shall be able to remove the users from Black List.

1.3.6 Admins shall be able to add the content of post, comment, and tags into the spam folder.

1.3.7 Admins shall be able to use any search functions that are used by users.

1.4 System Requirements

1.4.1 Sign Up

1.4.1.1 System shall require e-mail confirmation after sign-up.

1.4.1.2 System shall not allow multiple registration with the same e-mail.

1.4.2 Login

1.4.2.1 System shall require a password and a unique username to login.

1.4.2.2 The password shall be at least 8 characters.

1.4.2.3 The password shall consist of alphanumeric characters.

1.4.2.4 The username shall consist of alphanumeric characters.

1.4.3 Recommendation

1.4.3.1 The system should recommend posts and locations to users according to their activity on the system and their current locations.

1.4.4 Notifications

1.4.4.1 The system shall support a notification mechanism.

1.4.4.2 The system shall notify the user if the following events occur via notification box.

1.4.4.2.1 Any activity that happened on a user's post

1.4.4.2.1.1 Post is liked

1.4.4.2.1.2 Post is commented

1.4.4.2.1.2 A comment on the post is replied

1.4.4.2.2 Any activity that happened on a user profile

1.4.4.2.2.1 The user has a new follower

1.4.4.2.2.2 The private user gets following request

1.4.4.2.2.3 The user following request gets accepted

2. Nonfunctional Requirements

2.1 Portability

2.1.1 The system shall have an Android application and a web page.

2.2 Accessibility

2.2.1 Language of the Android application and we page shall be English.

2.3 Performance

2.3.1 The system response should be in 3 second.

2.3.2 The system should handle 500 requests per second.

2.3.3 The system's Android application size should be less than 200 MB.

2.4 Reliability

2.4.1 The system shall go down for maintenance once a month.

2.4.2 The maintenance shall not take more than 30 minutes.

2.4.3 Rest of time, downtime should be less than %1 for failures.

2.5 Security

2.5.1 The passwords shall be kept encrypted in database.

2.5.2 The web application shall only be accessed by HTTPS protocol.

2.6 Legal and Ethical Issues

2.6.1 Any data collected from users shall be used according to KVKK and GDPR.

2.6.2 New users shall accept "Terms of Use" and "Privacy Policy".

2.7 Implementation

2.7.1 The system shall use W3C standards and W3C Activity Streams Protocol for subscriptions and notifications.

2.7.2 The tagging system should be supported with Wikidata.org.

2.8 Robustness

2.8.1 System should restart the application in 45 minutes after failure.

2.8.2 Server admins should restart application manually in big failures.

2.8.3 Restart after big failures should take up to 4-5 hours.

2.8.4 Less than %2 of events/features should cause system failure.

3. Final release notes

You can find our final release tag in GitHub here. Our final release notes:

4. Information

Name: Columbus
Version: v0.9
Date of Release: 04 January 2022

5. Overview

This was the final release of our application. The application is now open for users to share their stories about their histories.

The new features added from the previous release:

- Users can now search with location name, tags, geolocation and date
- Users can now search users by their username
- User profile now has delete and report actions
- Users can make their profile private or public
- Users can now pin, delete and reply to the comments
- Users can now report and edit stories
- Notification system is added
- An admin page is supported

6. Issue Summary

You can find all of the closed issues for the final milestone here.
Brief summary for the addressed issues are:

- User could not search any post or user
- Search functionality is added for both posts and users
- User could not add any dates different than giving specifically day month or year
- Enhanced the dates by adding century, decade, year, year/month support for the posts
- User could not see the notifications
- Added notifications using the Activity Streams 2.0
- User profile was default public

- Added private optionality for the users
- Users could not report profiles or posts
- Reporting for the posts and the profile pages are added
- There was no existing admin page
- Added admin page for checking the reported contents
- Users could not delete, pin or reply to comments
- Users now can pin a comment on their posts
- Users can reply to any comments in the system
- Users can now delete the comments on their own posts

7. Resolution

- Profile page is refactored to add private/profile options
- With the date type changing posts are refactored to handle the new address type
- To support the notification system Activity Streams 2.0 is added to the system

8. Impacts

- The date type for the stories are updated and the all previous data is lost during the update
- Public/private operations on profile resulted in deleting the old users
- Additional Google Maps API key is added. To support it on the deployed website the website should be added to the CORS policy on Google Cloud

9. Changes From Previous Milestones

In the previous milestone, we tried to implement the fundamental features of our application. There was some feedback from customers. Date type was the most important feedback. We said that we will implement different types of dates but it was not ready in the previous milestone. There was also feedback for locations and searching. In the previous milestone, we have provided a location feature in profile. Understanding that this will cause misunderstandings, we removed this part from the profile. We hadn't implemented searching in the previous milestone. We tried to implement it by taking these feedback into consideration. We also listened to feedback on other groups. Especially the ones to the Postory. Our project subjects were the same, so we tried to follow the feedback on their project.

10. Final Project Evaluation:

In the final demo, we were able to deliver almost all the requirements. So,

overall feedback was good. There were points on the post component. We could implement it with more focus on the multimedia of the story. However we thought that story text is the fundamental of the post component, so we implemented images in this way.

We also realized that keeping in touch with the customer was a crucial point. The constant communication between the developers and the customer made us understand the requirements better. Thus enabled us to provide what the customer needs.
We realized that some of our requirements were not addressing our main requirement of the application, which was the stories with locations and dates. We stated lots of secondary requirements and they slowed down our development process.

2. Summary of Individual Coding Work

Member	Work
Abdulkadir Elmacı	- Added search page wıth fılters - Fixed multimedia upload error for create post - Wrote unit tests for the search page - Creating the Frontend Release - Fixing the date bug in the posts - Creating tag and release for the final milestone - Writing the system manual - Creating posts for the final database dump - Write executive summary for the Final Milestone Report with Erencan - Adding requirements to the final milestone - Adding design diagrams to the final milestone - Adding core functionality for the final milestone
Erencan Uysal	- Comment and like counts on post component - Reply comment feature - Date type change in the post creation screen - Date type change in the post component - Making date human readable in post component - Delete comment mechanism Pin comment mechanism - Edit post mechanism for web - Delete post mechanism for web - Report post mechanism for web - Api key added to previously created map component - Integration between backend

	and post component - Write user manual with my teammates - Write executive summary for final milestone report - Write my personal milestone report - Write unit tests for edit post - Write unit tests for comment component. - Write executive summary for the Final Milestone Report with Abdulkadir - Preparing user manual for web app with Merve - Creating stories for database dump

Hamza Akyıldız	- Pin comment api is refactored
	- Delete profile api is added
	- I added reply comment functionality to the create comment api
	- I refactored get comments api for getting replies
	- get likes of a specific story api is refactored
	- Get posts a specific user liked api is refactored
	- I implemented middleware for blocked users
	- Get Profile Information api is updated
	- Weekly Meeting about the low quality feature of the milestone 2 and how to improve them
	- privacy issues for blocking and private accounts are handled
	- explore page api
	- Report a tag API
	- Report a comment API
	- Admin Page Backend
	  - Authentication
	  - Login
	  - Get a Reported User
	  - Add a user to blacklist
	  - Remove a user from blacklist
	  - Get a Reported Story
	  - Add a story to spam folder
	  - Get a Reported Comment

	- Add a comment to spam folder - Get a Reported Tag - Add a tag to spam folder - Weekly Meeting about final milestone deliverables - Unit Tests - Authentication - Login - Get a Reported User - Add a user to blacklist - Remove a user from blacklist - Get a Reported Story - Add a story to spam folder - Get a Reported Comment - Add a comment to spam folder - Get a Reported Tag - Add a tag to spam folder - Report a comment API - Report a tag API - reply to a comment - Unit test refactorization - Pin comment - get comments - create comments - Get likes for a specific story - Get posts a specific user liked - Get Profile Information api
Merve Rabia Barın	- Converted birthday to human readable format - Added a menu to other profiles to report user and block user - Added report message dialog - Wrote unit test for convert birthday function - Wrote unit tests for Edit Profile Information - Created an admin page - Added view reported users, stories to Admin page - Added safe/unsafe actions for users, stories to Admin page - Created profile settings dialog for user's own profile

	- Added profile visibility options (private, public) - Made private accounts invisible to users other than their followers - Added delete profile option - Put notification box icon into Header - Created a notification box with two tabs: Activity, Follow Request - Made Activity tab to show user's activity stream - Displayed follow requests in Follow Request tab
Onur Can Avcı	- Add rerequest after close mobile app and open again to - Add new connectionsModal to component (to view followers and following) - Add other profile page - Add Unblock Icon to Other Profile - Add Follow Service and Integreted with UI - Remove location information from Profile page - Add progress bar modal to show image uploading process - Refactor S3Image Upload and add base64 arraybuffer library - Refactor Custom FormInput to manage width of the label and input - Refactor Custom Edit Profile Service Request and style - Add ProgressModal and Custom Modal to Edit Profile - Add Notification Page to home navigation - Add Notification Icon to Home Page - Add Notification Page and style - Add Notification endpoint to service - Add search user api to service - Add Default search page UI template - Add User Search component to

	Search and style - Add report and block user API to services - Add report and block user modal - Add report and block user integrated with service and other profile page - Add liked and shared section ui to profile page - Add user liked stories to services - Add refresh feature to profile page stories request - Add Follow request function according to public and private account - Add accept follow request endpoint to service - Add accept and reject request icon and integrate with api - Add Create Story Choose Date Modal - Refactor Create Story Page
Ramazan Bulut	- Attended weekly meetings and mobile meetings - Wrote User Manual for Mobile Application - Wrote System Manual for Mobile Application - Implement Map and Story Search Pages: - Implement Comment UI and Commenting functionality - Implement Comment Features: Pin, Reply, Delete, Edit, Report - Implement Story Card functionalities and UI : completely - Implement Home Page: completely - Implement Login Page: completely - Implementing Like functionality on Stories :completely - Write unit test for home page - Implement Map functionality in Create Story - Writing Personal efforts - Implementing Detailed Story Page :completely

	- Creating Stories for Mobile Application - Writing Individual MS report - Bug fixing in general
Umut Kocasarı	- Attended weekly meetings and backend meetings - Attended team meetings to combine the subparts of the system - Wrote API documentation of the report - Wrote status of requirements part of the report - Made research and discussions about activity stream and its implementation - Wrote tests for post delete functionality - Wrote tests for post edit functionality - Wrote tests for logout functionality - Wrote tests for home page functionality - Wrote tests for Django models - Migrated all Python unittests to Django unittests - Implemented date search functionality - Implemented geographical search functionality - Implemented title based search - Implemented story text based search - Implemented syntactic text based search, which is robust to typos - Added comment delete functionality - Added post delete functionality - Added post edit functionality - Added block unblock functionality - Added text based location search - Added user search using their usernames - Added pin comment functionality - Added necessary Django models for reports - Implemented report story functionality

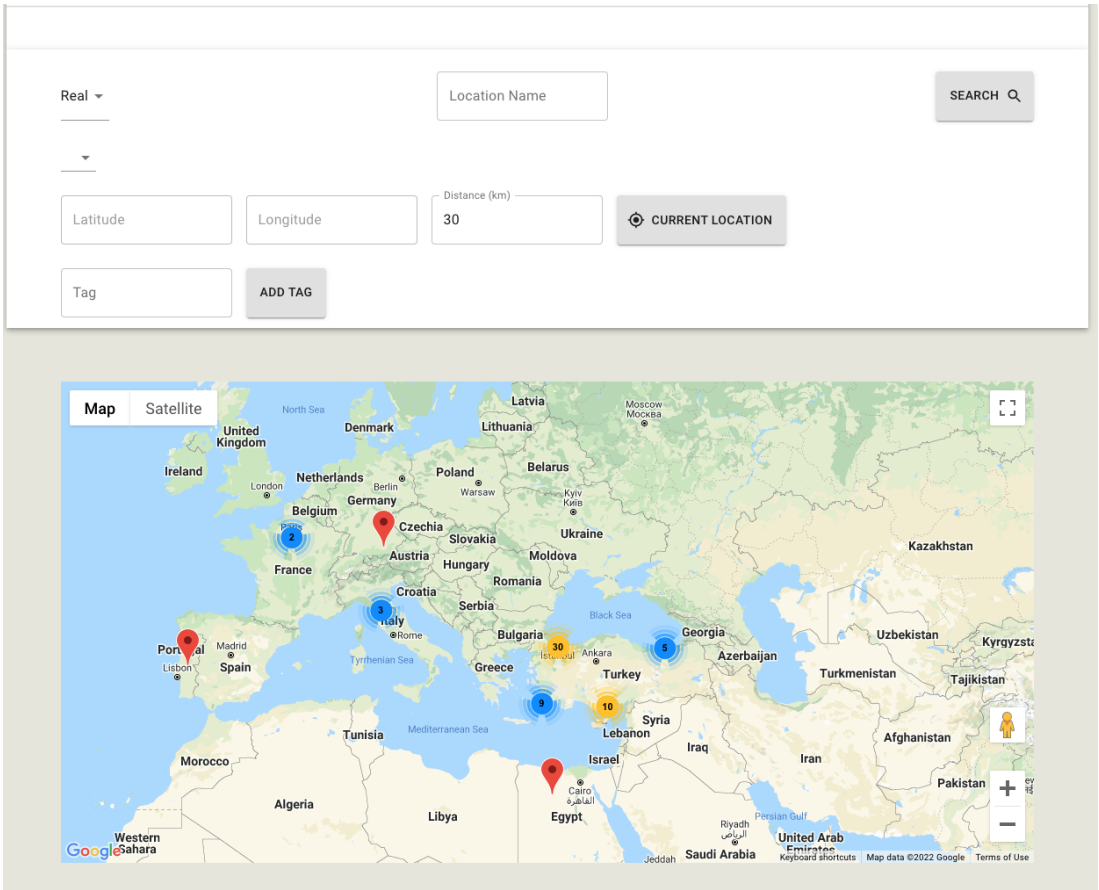
	- Implemented report user functionality - Combined different search methods to one, big, generalizable search method - Added location based searches to combined search - Added minimum maximum latitude longitude search to combined search - Thought about potential solutions and changed format of date and multimedia - Refactored post creation with the new format of date and multimedia - Resolved database issues occurred after the format change of date and multimedia - Refactored post edit functionality - Refactored all APIs which return stories as their outputs because of the change in the format of date and multimedia - Changed input format of block unblock API - Refactored the search date functionality according to new format - Refactored almost all the tests in the system according to new format - Wrote tests for the new format of date and multimedia - Wrote tests for report user functionality - Wrote tests for report story functionality - Wrote tests for story text search - Wrote tests for story location - Wrote tests for story location text - Wrote tests for minimum maximum latitude longitude search - Wrote tests for date search - Added urls to the output of user search - Made date search inclusive. It means century or decade based search also includes specific date based search

	- Resolved database migration errors caused by changes in models - Wrote totally 79 unit tests
Yağız Efe Şabanoğlu	- Attending backend meetings - Attending weekly team meetings - Wrote Standards Part of the Final Milestone Report - Review API Documentation of the report - Made research and discussions about activity stream and its implementation - Made research and discussions about Wikidata API and its implementation ● Made research and discussions about notifications in social media platforms and its implementation ● Writing tag filter for Search API ● Adding Follow/Unfollow Notifications feature ● Adding Like/Unlike Notifications feature ● Integrated Wikidata API to tag filter attribute ● Adding Follow Request Notifications feature ● Adding Comment Create Notifications feature ● Adding Comment Delete Notifications feature ● Adding Pin/Unpin story Notifications feature ● Implementing Get All of Notifications API ● Implementing Get Popular Tags of User API. ● Bug that corrupt Follow request Notifications is solved. ● Bug that makes total notifications number wrong is solved. ● Bug about wrong Response Type is solved. ● Wrote unit-tests for Activity Stream ● Wrote unit-tests for Follow/Unfollow Notifications ● Wrote unit-tests for Like/Unlike Notifications

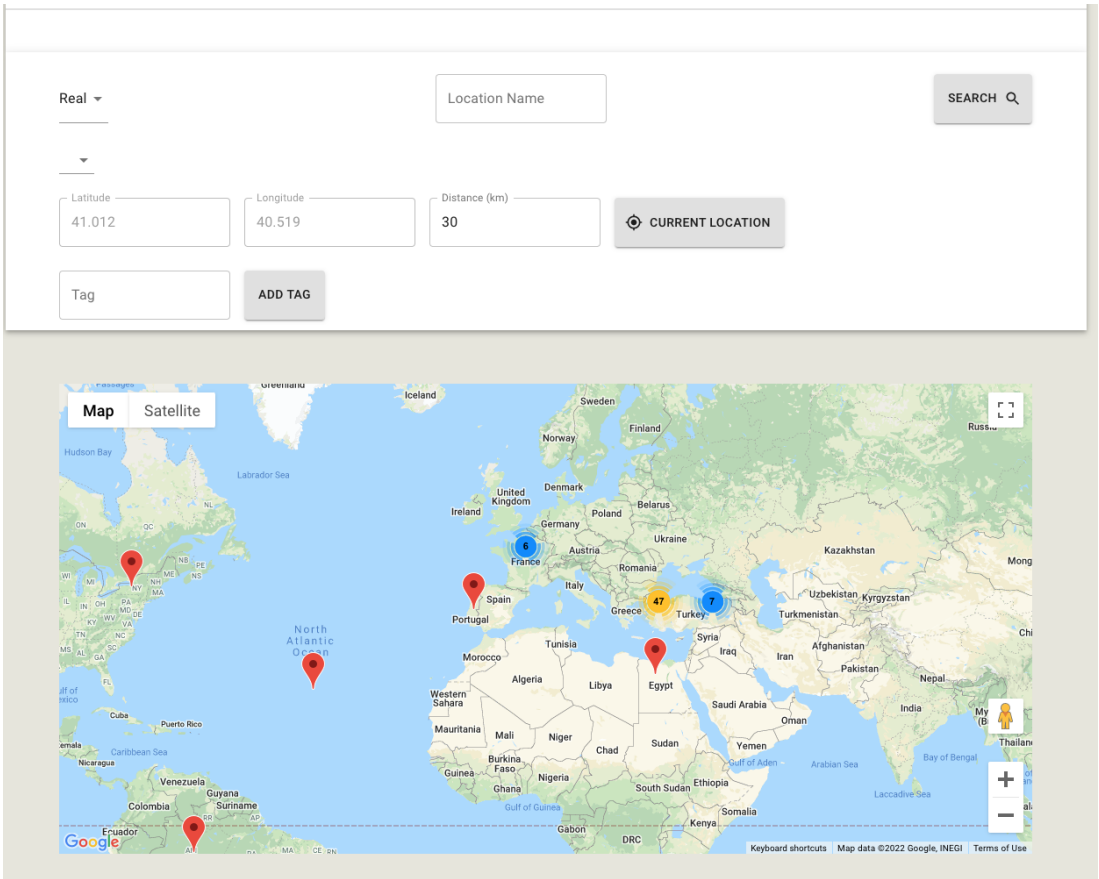
	• Wrote unit-tests for Follow Request Notifications • Wrote unit-tests for Comment Create Notifications • Wrote unit-tests for Comment Delete Notifications • Wrote unit-tests for Pin/Unpin Story Notifications • Wrote unit-tests for Get Popular Tags of User API

3. Core Functionality for the Scenario

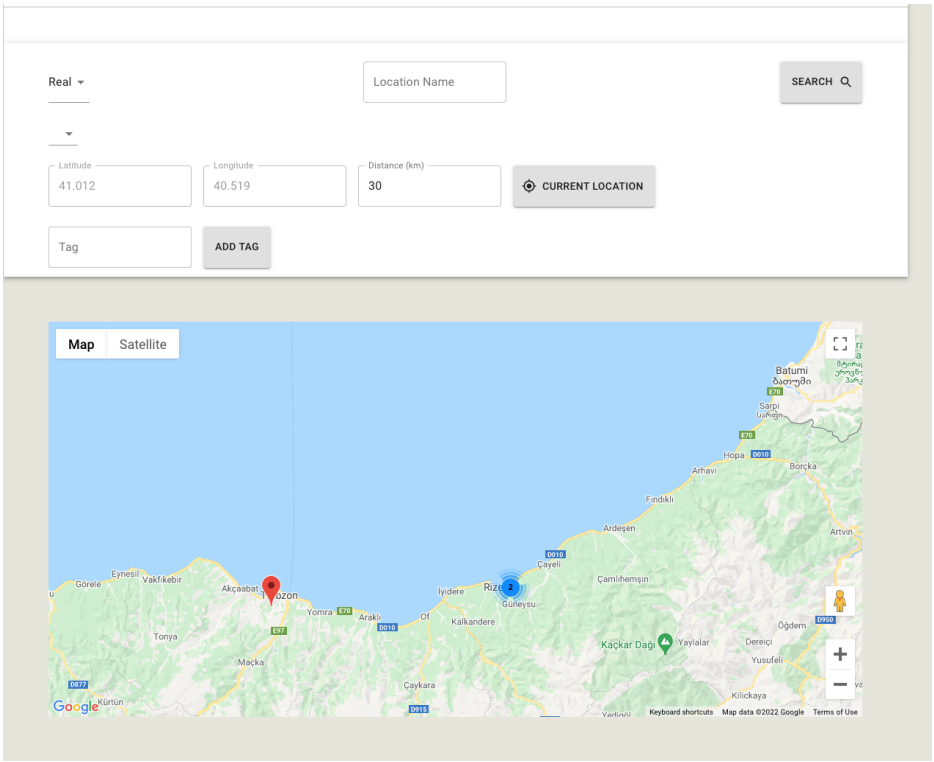
For the core functionality of the project, as described in the final milestone, Serhan wants to see the nearby posts by current locations around 30 km. So Serhan logs in to our app and goes to the search page. Selects the real locations. Serhan sees a page like this:



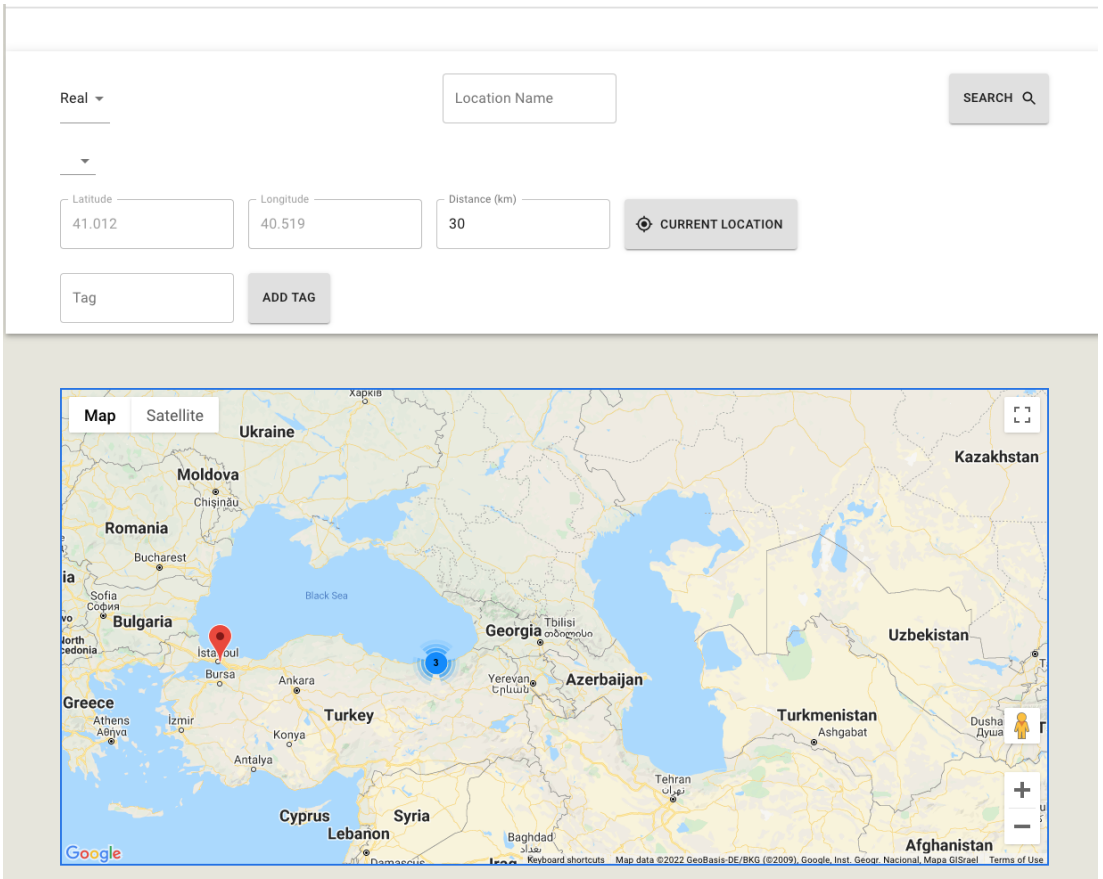
As Serhan does not want to see the posts from far away. Serhan clicks the current location(only works if you enable locations for the HTTP, more information here) or clicks on the map to select the location and sees that the latitude longitude is updated.



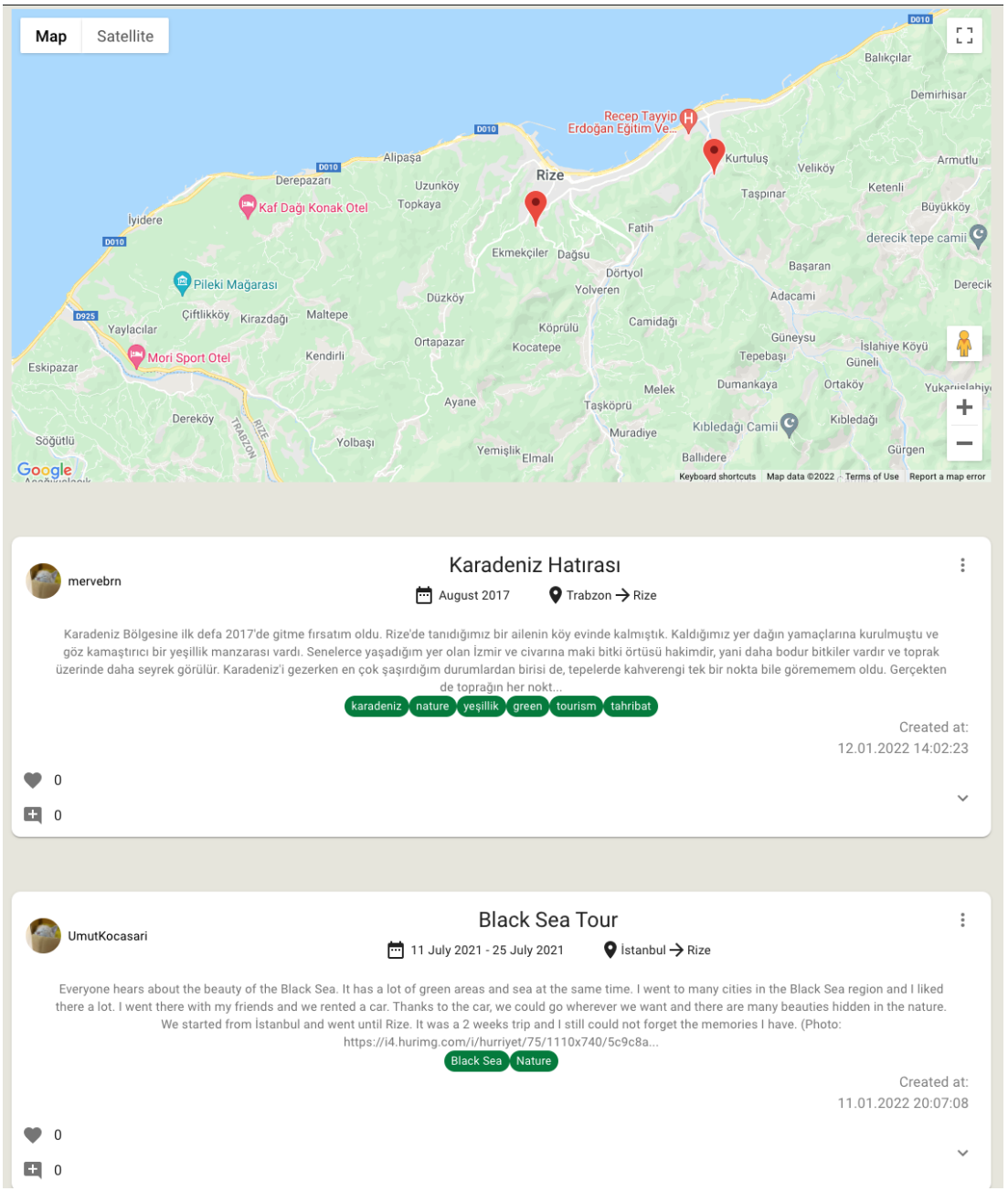
Then Serhan clicks the search button and sees the two posts on Rize.



Also Serhan sees a pinmark on Trabzon and Istanbul since the stories that are near Rize have multiple locations.



Then Serhan presses the blue circle to view the posts in Rize. He reads the posts and enjoys the stories.



4. Standards

Endpoint: /activitystream
Method: POST
Authorization: No Need.

Activity Stream keeps track of activities done on an app/site. Many contributions can be made to the user, customer and company by keeping/displaying or examining this record.

Currently, Activity Streams in Columbus contain user actions that cause **database changes** and **logout**.

Currently, the Activity Stream infrastructure is ready and operational. We entered **action-specific inputs to the Activity Stream Model in the database while this action was performed on the backend for every action we want to keep in the record.** The API takes the **offset** parameters that take the limit where you set how many activities you want to see and/or the **ID of the last ActivityStream** you want to see. In this way, you can see as many activities as you want and bring the activities up to the last activity you saw. It can expand the scope of the activity if desired. Finally, we added the dates when the activities took place to the model. In addition, we have reached a solution by doing user filtering on the activity stream to get notifications.

Typical Response:

```
{
            "@context": Context of the activity stream,
            "summary": General summary of the response object,
            "type": Type of the returned object based on scheme,
            "total_items": Total number of activities returned,
            "orderedItems":   List of activities :
                         [ {
                        "@context": Context of the activity stream,
                        "summary": General summary of the activity,
                        "id": Activity Stream Model Id,
                        "type": Type of the activity
                        "actor": People doing the activity,
                        "object": Object of the activity,
                        "target": Thing affected from activity
                        "tag" = Tags of story if type is about story
                        "date" = models.DateTimeField()
                        }]
}
```

Activities:

```
_setprofile: {activity.actor.username} set profile informations.

_updatepost: {activity.actor.username} updated {activity.story.id}

_createpost: {activity.actor.username} created post

_deletepost: {activity.actor.username} deleted {activity.story.id}

_logout: {activity.actor.username} logged out

_unlike: {activity.actor.username} uliked {activity.story.id}

_like: {activity.actor.username} liked {activity.story.id}
```

_unfollow: {activity.actor.username} unfollowed {activity.target.username}

_follow: {activity.actor.username} followed {activity.target.username}

_comment_update: {activity.actor.username} updated comment : {activity.comment.id}

_comment_create: {activity.actor.username} added comment to story : {activity.story.id}

_comment_delete: {activity.actor.username} deleted comment : {activity.story.id}

_followrequest : {activity.actor.username} has requested to follow {activity.target.username}

_block : {activity.actor.username} blocked {activity.target.username}

_unblock : {activity.actor.username} blocked {activity.target.username}

_pin : {activity.actor.username} pinned {activity.comment.id}

_unpin : {activity.actor.username} unpinned {activity.comment.id}

_reportstorycreate : {activity.actor.username} reported {activity.story.id}

_reportusercreate : {activity.actor.username} reported {activity.target.username}

_reportcommentcreate : {activity.actor.username} reported {activity.comment.id}

_reporttagtcreate :{activity.actor.username} reported {activity.tag.id}

Successful Response

```
{
  "@context": "https://www.w3.org/ns/activitystreams",
  "summary": "Activity stream",
  "type": "OrderedCollection",
  "total_items": 10,
  "orderedItems": [
    {
      "@context": "https://www.w3.org/ns/activitystreams",
      "summary": "string set profile informations.",
      "id": 13,
      "type": "SetProfile",
      "actor": {
        "type": "https://schema.org/Person",
        "@id": "string"
      }
    },
    {
      "@context": "https://www.w3.org/ns/activitystreams",+
      "summary": "string set profile informations.",
      "id": 12,
      "type": "SetProfile",
      "actor": {
        "type": "https://schema.org/Person",
        "@id": "string"
      }
    },
    {
      "@context": "https://www.w3.org/ns/activitystreams",
      "summary": "string updated 2 ",
      "id": 11,
      "type": "UpdatePost",
      "actor": {
        "type": "https://schema.org/Person",
        "@id": "string"
      },
      "object": {
        "type": "https://schema.org/ShortStory",
        "@id": 2
      }
    },
    {
      "@context": "https://www.w3.org/ns/activitystreams",
      "summary": "string created post",
      "id": 9,
      "type": "CreatePost",
      "actor": {
        "type": "https://schema.org/Person",
        "@id": "string"
      },
      "object": {
        "type": "https://schema.org/ShortStory",
        "@id": 5
      }
    },
    {
      "@context": "https://www.w3.org/ns/activitystreams",
      "summary": "string updated comment : 2  ",
      "id": 7,
      "type": "CommentUpdate",
      "actor": {
        "type": "https://schema.org/Person",
        "@id": "string"
      },
      "object": {
        "type": "https://schema.org/Comment",
        "@id": 2
      }
    },
    {
      "@context": "https://www.w3.org/ns/activitystreams",
      "summary": "string added comment to story : 1 ",
      "id": 6,
      "type": "CommentCreate",
      "actor": {
        "type": "https://schema.org/Person",
        "@id": "string"
```

```
      },
      "object": {
        "type": "https://schema.org/ShortStory",
        "@id": 1
      }
    },
    {
      "@context": "https://www.w3.org/ns/activitystreams",
      "summary": "string added comment to story : 1 ",
      "id": 4,
      "type": "CommentCreate",
      "actor": {
        "type": "https://schema.org/Person",
        "@id": "string"
      },
      "object": {
        "type": "https://schema.org/ShortStory",
        "@id": 1
      }
    },
    {
      "@context": "https://www.w3.org/ns/activitystreams",
      "summary": "string created post",
      "id": 3,
      "type": "CreatePost",
      "actor": {
        "type": "https://schema.org/Person",
        "@id": "string"
      },
      "object": {
        "type": "https://schema.org/ShortStory",
        "@id": 3
      }
    },
    {
      "@context": "https://www.w3.org/ns/activitystreams",
      "summary": "string created post",
      "id": 2,
      "type": "CreatePost",
      "actor": {
        "type": "https://schema.org/Person",
        "@id": "string"
      },
      "object": {
        "type": "https://schema.org/ShortStory",
        "@id": 2
      }
    },
    {
      "@context": "https://www.w3.org/ns/activitystreams",
      "summary": "string created post",
      "id": 1,
      "type": "CreatePost",
      "actor": {
        "type": "https://schema.org/Person",
        "@id": "string"
      },
      "object": {
        "type": "https://schema.org/ShortStory",
        "@id": 1
      }
    }
  ]
}
```


POST /user/activity_stream/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model / Example Value

```
{
  "limit": 0,
  "offset": 0
}
```

Parameter content type: application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

Wikidata Integration

```
class Tag(models.Model):
    story_id = models.ForeignKey(Story, on_delete=models.CASCADE)
    tag = models.CharField(max_length=100)
```

Columbus has a tag system for stories. In addition, there is a feature to filter the stories by tags while searching for stories in the search section. After reviewing the Wikidata API and speaking to our assistant, we thought that displaying posts with similar tags taken from Wikidata while filtering tags would be a very good example of using the Wikidata API.

You can see our tag models above. When the "X" tag filter is added in the Search API, **we get the story_ids of the tags with the tag string "X" in the tag table in the database and intersect it with the stories in the search response object at that moment**[1]. Then we get the tags related to the "X" object from the Wikidata API and write the **same process for these tags** as we did above.

For example:

Endpoint: /search/location_search/
Body : {
"search_text": "Anıtkabir",
"page_number": 1,
"page_size": 10,
"tags": ["Atatürk"]
}

This api call returns not only the stories with the text of Anıtkabir and having the tag Atatürk, but also the stories with the tags such as Mustafa Kemal Atatürk, the founding father of the Republic of Turkey, which is also associated with the text of Anıtkabir.

5. API

API documentation could be found here:
https://github.com/bounswe/2021SpringGroup7/wiki/451-API-Documentation

[1] "same process for these tags as we did above"

API Documentation

docs

GET /docs/

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

guest

POST /guest/change_password/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "username": "string",
  "password": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /guest/home_page/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "page_number": 0,
  "page_size": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /guest/login/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "username": "string",
  "password": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /guest/register/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "username": "string",
  "email": "string",
  "first_name": "string",
  "last_name": "string",
  "password": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

search

POST /search/date_search/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "search_date": "string",
  "page_number": 0,
  "page_size": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /search/geographical_search/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "query_latitude": 0,
  "query_longitude": 0,
  "page_number": 0,
  "page_size": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /search/location_search/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "search_text": "string",
  "page_number": 0,
  "page_size": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /search/search/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

Parameter content type:
application/json

```
{
  "username": "string",
  "search_text": "string",
  "page_number": 0,
  "page_size": 0,
  "query_latitude": 0,
  "query_longitude": 0,
  "query_distance": 0,
  "location_text": "string",
  "max_latitude": 0,
  "max_longitude": 0,
  "min_latitude": 0,
  "min_longitude": 0,
  "search_date_type": "string",
  "search_date": 0,
  "search_year_start": 0,
  "search_month_start": 0,
  "search_day_start": 0,
  "search_hour_start": 0,
  "search_minute_start": 0,
  "search_year_end": 0,
  "search_month_end": 0,
  "search_day_end": 0,
  "search_hour_end": 0,
  "search_minute_end": 0
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /search/text_exact_search/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "search_text": "string",
  "page_number": 0,
  "page_size": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /search/title_exact_search/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "search_text": "string",
  "page_number": 0,
  "page_size": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /search/title_partial_search/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "search_text": "string",
  "page_number": 0,
  "page_size": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /search/user_search/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "search_text": "string",
  "page_number": 0,
  "page_size": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

test

GET /test/{username}

Parameters

Parameter	Value	Description	Parameter Type	Data Type
username	(required)	**Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

user

POST /user/accept_follow_request/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "request_id": 0,
  "accept": true
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/activity_stream/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "limit": 0,
  "offset": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/admin/action_reported_comments

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "login_hash": "string",
  "report_id": 0,
  "safe": true
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/admin/action_reported_stories

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "login_hash": "string",
  "report_id": 0,
  "safe": true
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/admin/action_reported_tags

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "login_hash": "string",
  "report_id": 0,
  "safe": true
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/admin/action_reported_users

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "login_hash": "string",
  "report_id": 0,
  "safe": true
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/admin/get_reported_comments

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "login_hash": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/admin/get_reported_stories

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "login_hash": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/admin/get_reported_tags

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "login_hash": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/admin/get_reported_users

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "login_hash": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/admin/login

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "admin_username": "string",
  "admin_password": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/admin/remove_users_from_black_list

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "login_hash": "string",
  "username": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/block/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "blocker": 0,
  "blocked": 0,
  "action_block": true
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/comment_create/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "username": "string",
  "story_id": 0,
  "text": "string",
  "parent_comment_id": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/comment_delete/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "comment_id": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/comment_update/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "comment_id": 0,
  "text": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/create_post/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

Parameter content type:
application/json

```
{
  "title": "string",
  "text": "string",
  "multimedias": [
    "string"
  ],
  "username": "string",
  "time_start": {},
  "time_end": {},
  "location": [
    "string"
  ],
  "tags": [
    "string"
  ]
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/delete_post/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "story_id": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/delete_profile/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "user_id": 0,
  "password": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/edit_post/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "story_id": 0,
  "title": "string",
  "text": "string",
  "multimedias": [
    "string"
  ],
  "time_start": {},
  "time_end": {},
  "location": [
    "string"
  ],
  "tags": [
    "string"
  ]
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/explore/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "username": "string",
  "page_number": 0,
  "page_size": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/follow/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "user_id": 0,
  "follow": 0,
  "action_follow": true
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/get_comments/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "story_id": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

GET /user/get_follow_request/

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

GET /user/get_likes/{story_id}/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
story_id	(required)		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

POST /user/get_notifications/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "user_name": "string",
  "limit": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

GET /user/get_popular_tags/{user_id}/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
user_id	(required)		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

GET /user/get_profile/{user_id}/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
user_id	(required)		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

POST /user/get_user_likes/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model Example Value

```
{
  "username": "string",
  "page_number": 0,
  "page_size": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/home_page/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model Example Value

```
{
  "username": "string",
  "page_number": 0,
  "page_size": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/like/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model Example Value

```
{
  "story_id": 0,
  "user_id": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/logout/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model Example Value

```
{
  "username": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/pin_comment/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "comment_id": 0,
  "story_id": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/profile_post/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "username": "string",
  "page_number": 0,
  "page_size": 0
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/report_comment/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "comment_id": 0,
  "reporter_id": 0,
  "report": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/report_story/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "username": "string",
  "story_id": 0,
  "text": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/report_tag/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "tag_id": 0,
  "reporter_id": 0,
  "report": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/report_user/

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

```
{
  "reported_username": "string",
  "reporter_username": "string",
  "report": "string"
}
```

Parameter content type:
application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

POST /user/set_profile/

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data			body	Model | Example Value

Parameter content type:
application/json


```
{
  "user_id": 0,
  "first_name": "string",
  "last_name": "string",
  "photo_url": "string",
  "birthday": "string",
  "biography": "string",
  "public": true
}
```


Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

[BASE URL:]

VALID {...}

Search Location with Name	
end point	http://ec2-35-158-103-6.eu-central-1.compute.amazonaws.com:8000/search/location_search/

body	{
	"search_text": "izmit",
	"page_number": 1,
	"page_size": 10
	}

curl command	curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-CSRFToken: EmaKKNsk6tH6akRKdB2SSxYCcizPOQmLGUB3PYaIvYBKlUxSbKfbclvvmlNfYh2b' --header 'Authorization: token 3a6ce576d1a6b4bc507185aa76c31d6985d2c344' -d '{"search_text": "izmit","page_number": 1,"page_size": 10}' 'http://ec2-35-158-103-6.eu-central-1.compute.amazonaws.com:8000/search/location_search/'

User Search	
end point	http://ec2-35-158-103-6.eu-central-1.compute.amazonaws.com:8000/search/user_search/

body	{
	"search_text": "mervebrn",
	"page_number": 1,
	"page_size": 10
	}

curl command	curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-CSRFToken: EmaKKNsk6tH6akRKdB2SSxYCcizPOQmLGUB3PYaIvYBKlUxSbKfbclvvmlNfYh2b' --header 'Authorization: token 3a6ce576d1a6b4bc507185aa76c31d6985d2c344' -d '{"search_text":

	"mervebrn","page_number": 1,"page_size": 10}' 'http://ec2-35-158-103-6.eu-central-1.compute.amazonaws.com:8000/search/user_search/'

Explore Page	
end point	http://ec2-35-158-103-6.eu-central-1.compute.amazonaws.com:8000/user/explore/
body	{ "username": "hamzaakyildiz", "page_number": 1, "page_size": 10 }
curl command	curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-CSRFToken: EmaKKNsk6tH6akRKdB2SSxYCcizPOQmLGUB3PYaIvYBKlUxSbKfbclvvmlNfYh2b' --header 'Authorization: token 3a6ce576d1a6b4bc507185aa76c31d6985d2c344' -d '{"username": "hamzaakyildiz","page_number": 1,"page_size": 10}' 'http://ec2-35-158-103-6.eu-central-1.compute.amazonaws.com:8000/user/explore/'

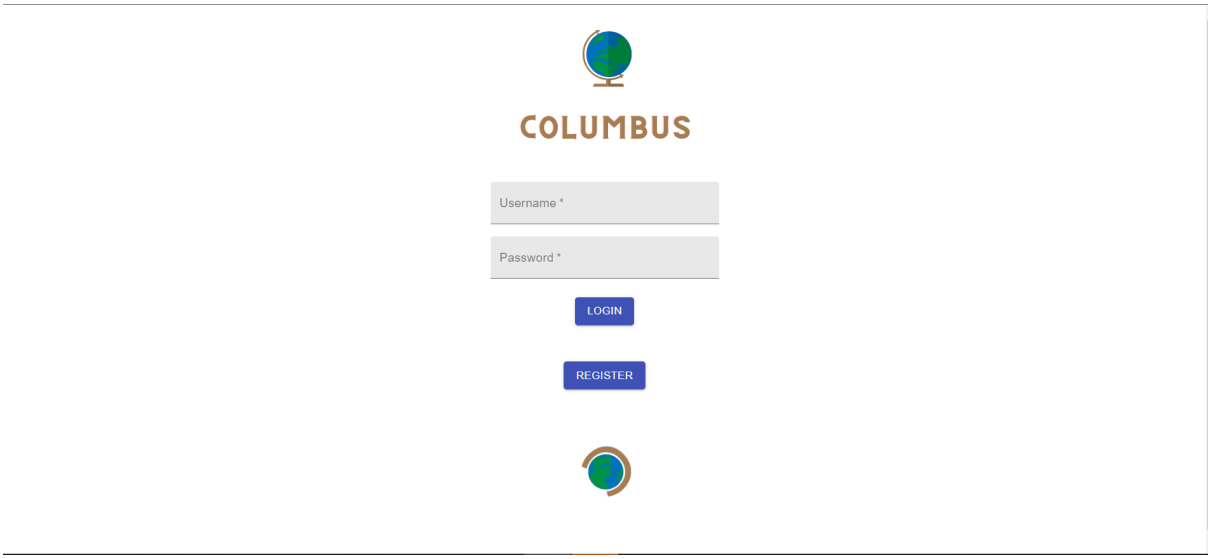
6. User Experience

WEB INTERFACES

Login Page

Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/pages/Login

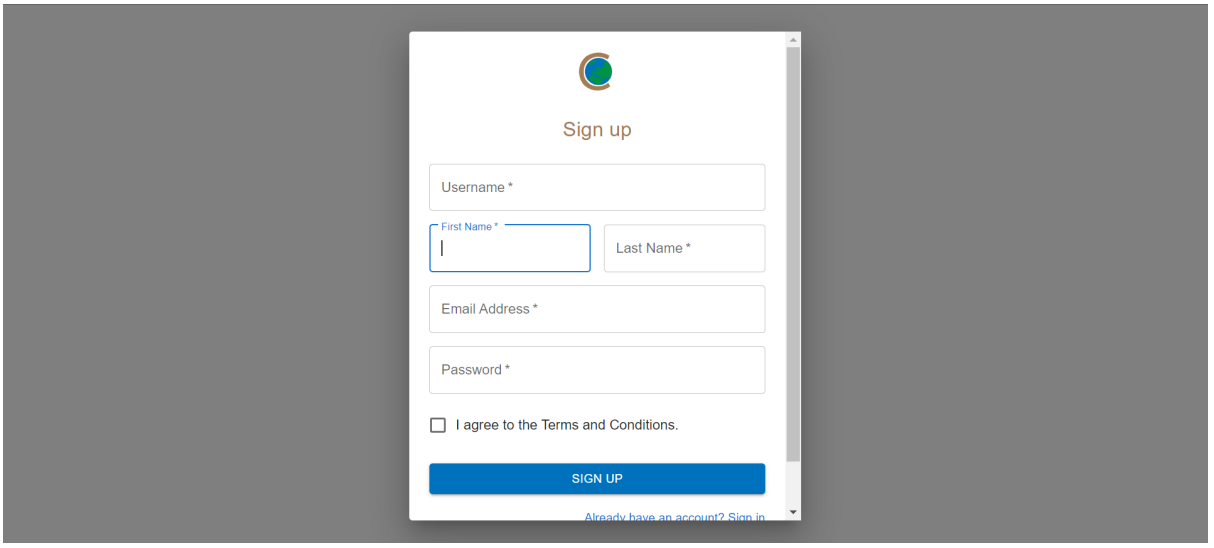
Login :



Register Dialog

Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/pages/Register

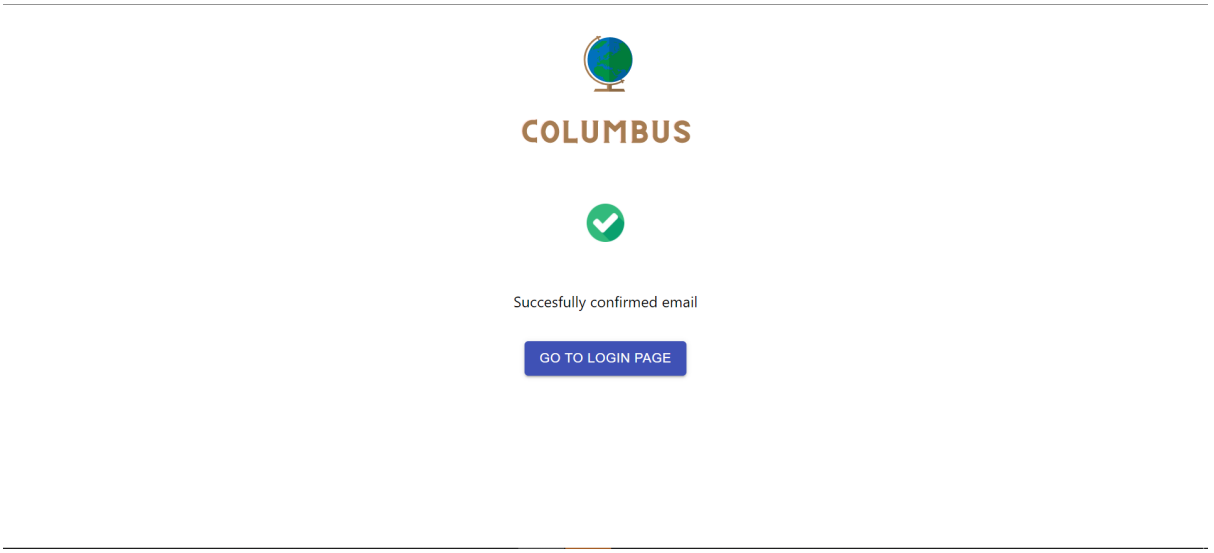
Register :



Email Confirmation Page

Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/pages/EmailConfirmationPage

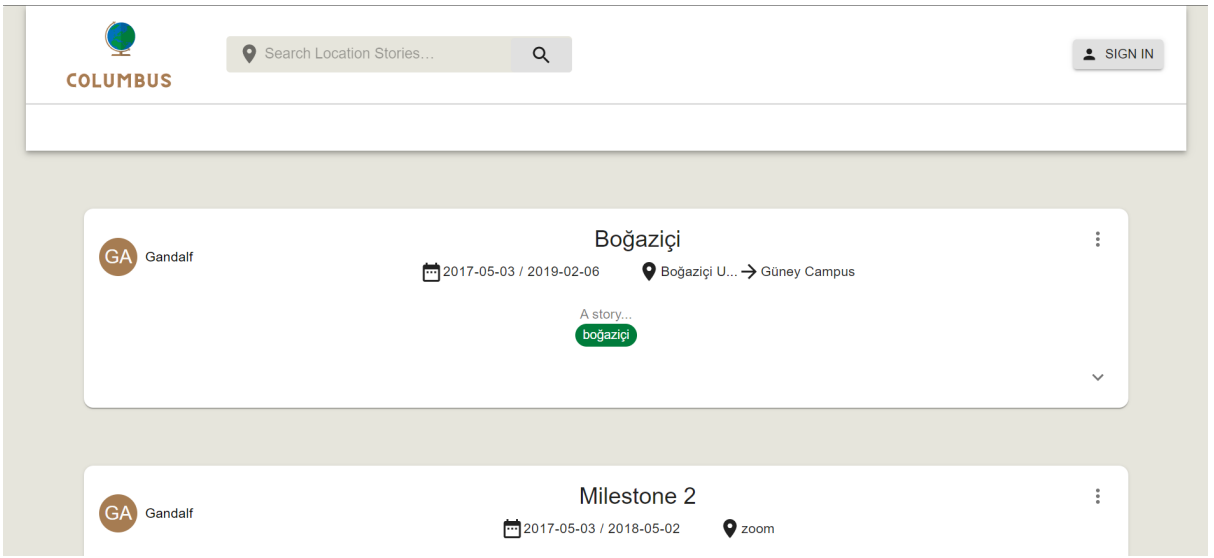
Email Confirmation Page :



Guest Home Page

Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/pages/Home

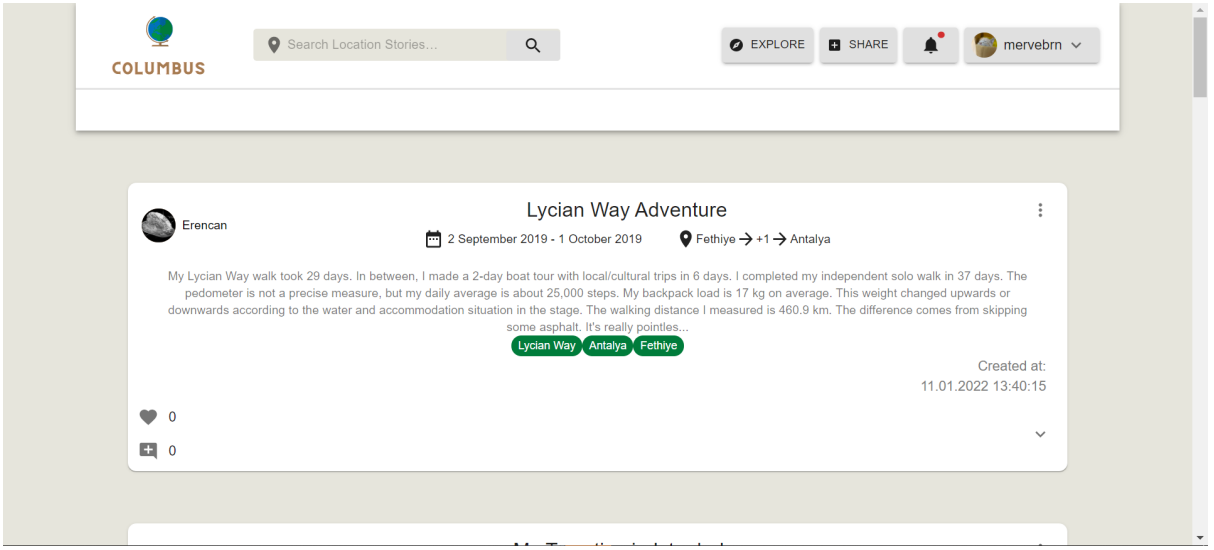
Guest Home Page :



User Home Page

Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/pages/Home

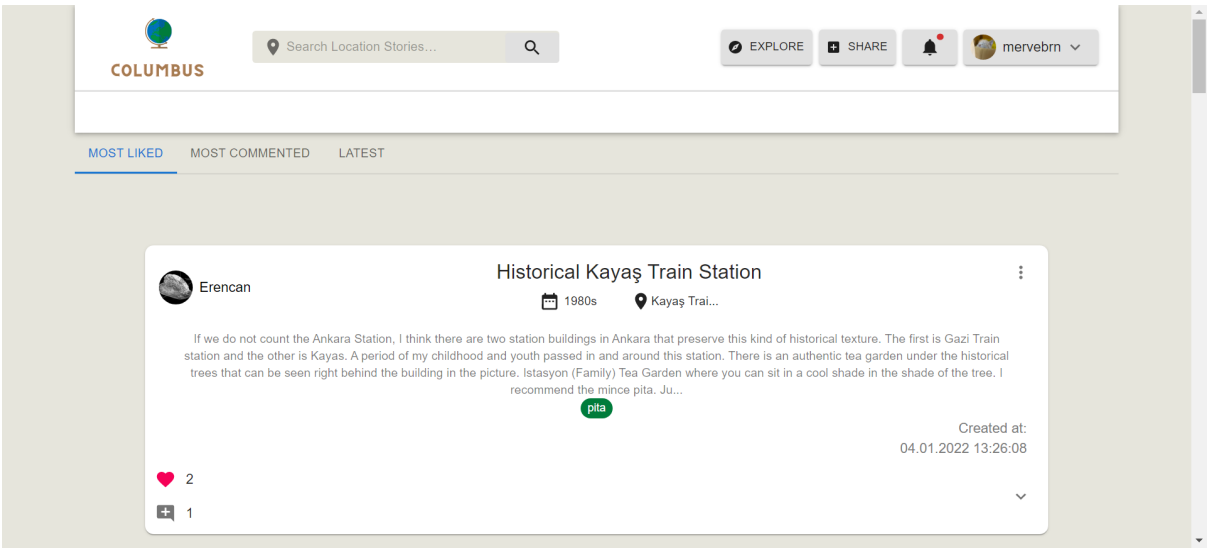
Home :



Explore Page

Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/pages/Explore

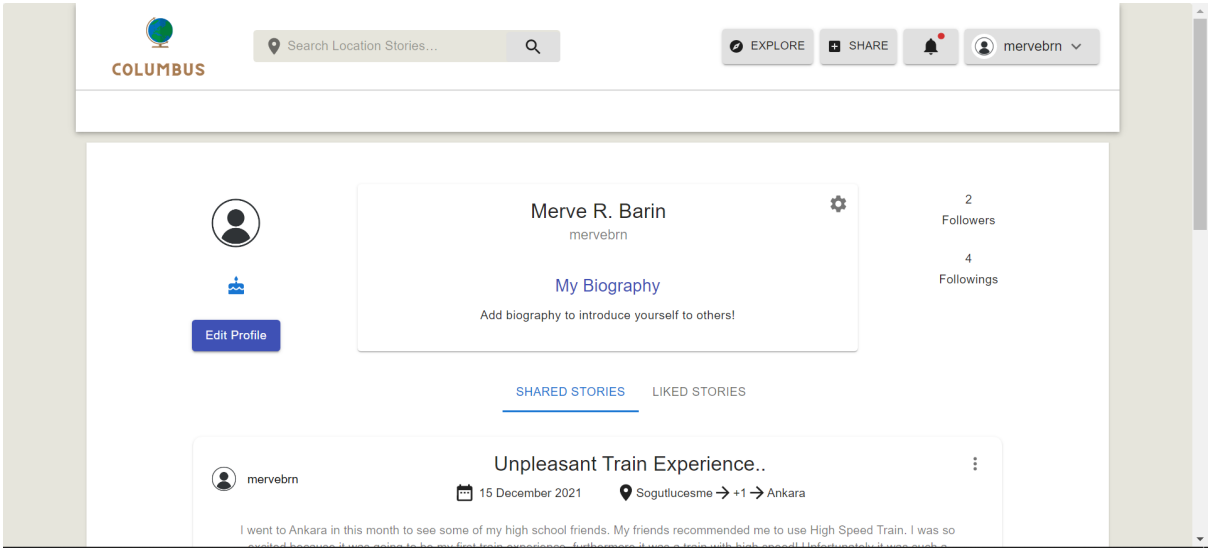
Explore:



User's own Profile Page

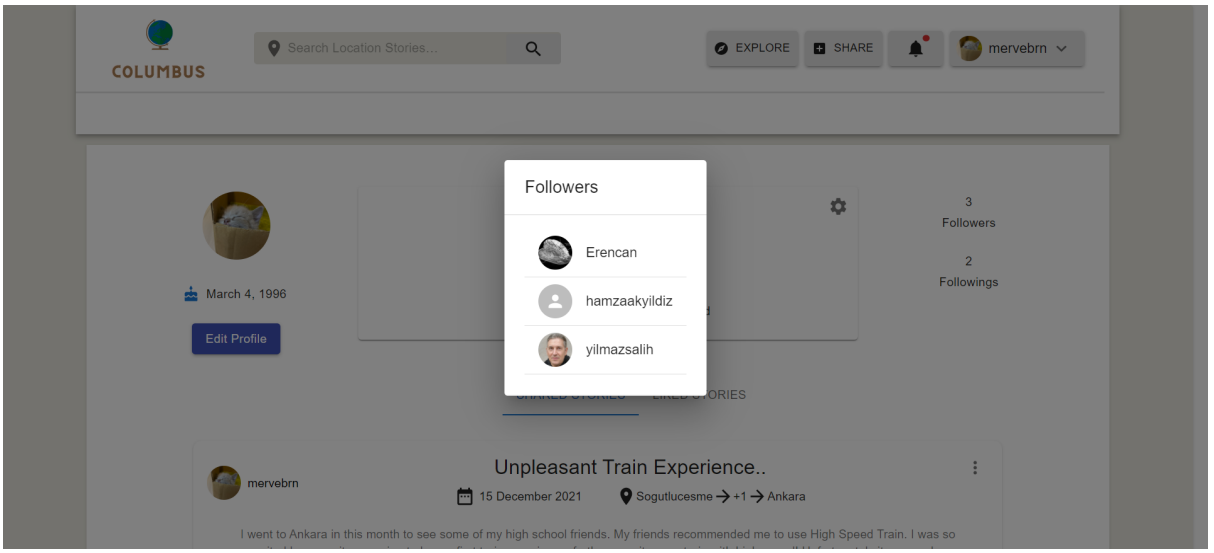
Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/pages/Profile

Viewing User's Own Profile :



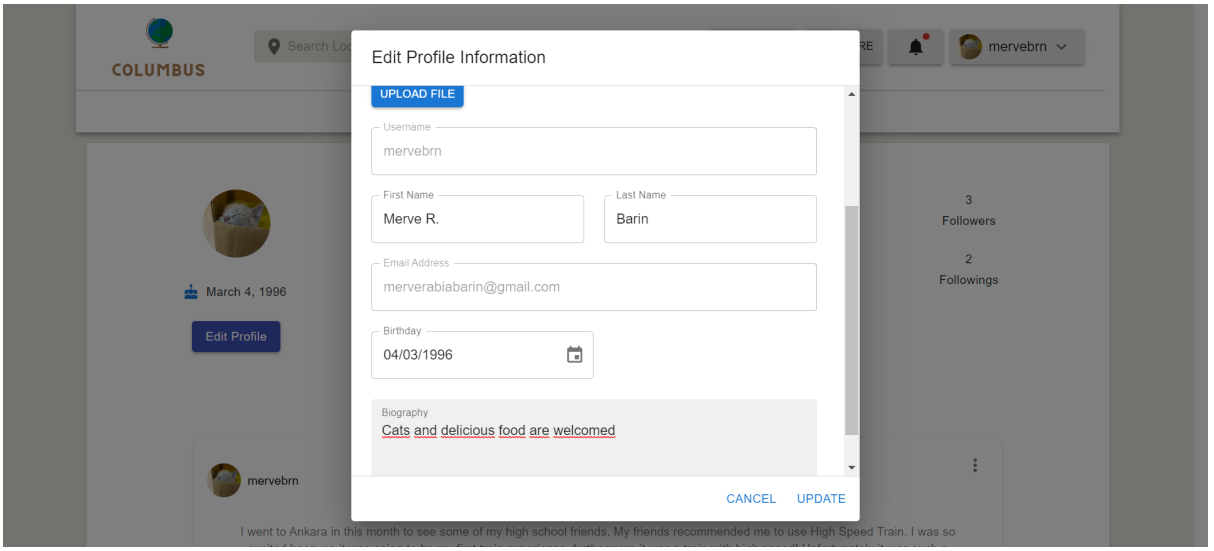
Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/components/Dialogs/FollowerDialog

Follower/Following Dialog Screenshot:



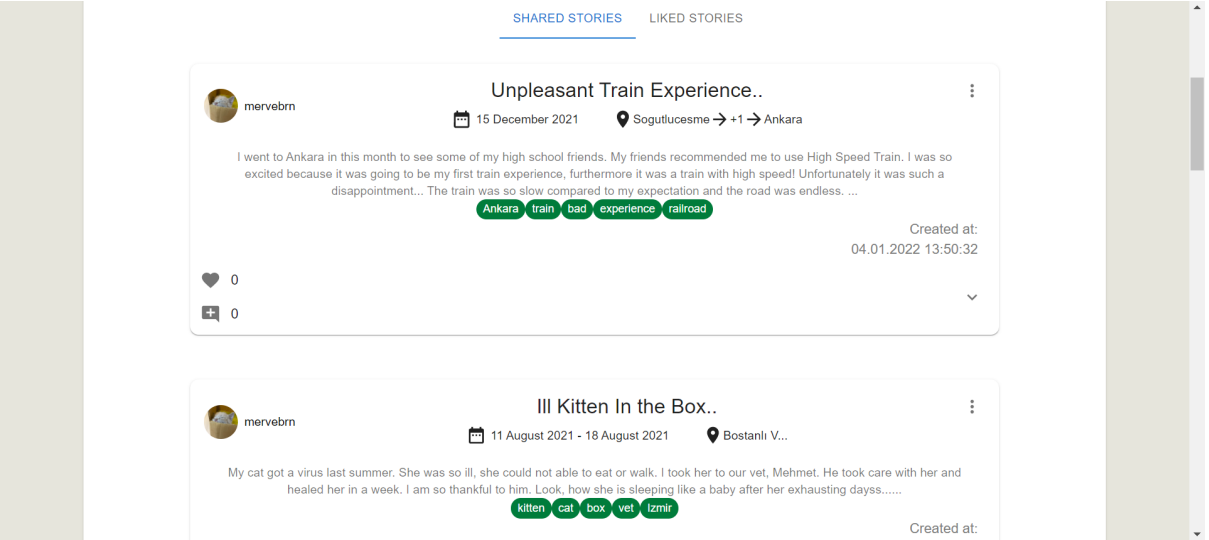
Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/components/Dialogs/EditProfileDialog

Edit Profile Dialog Screenshot:



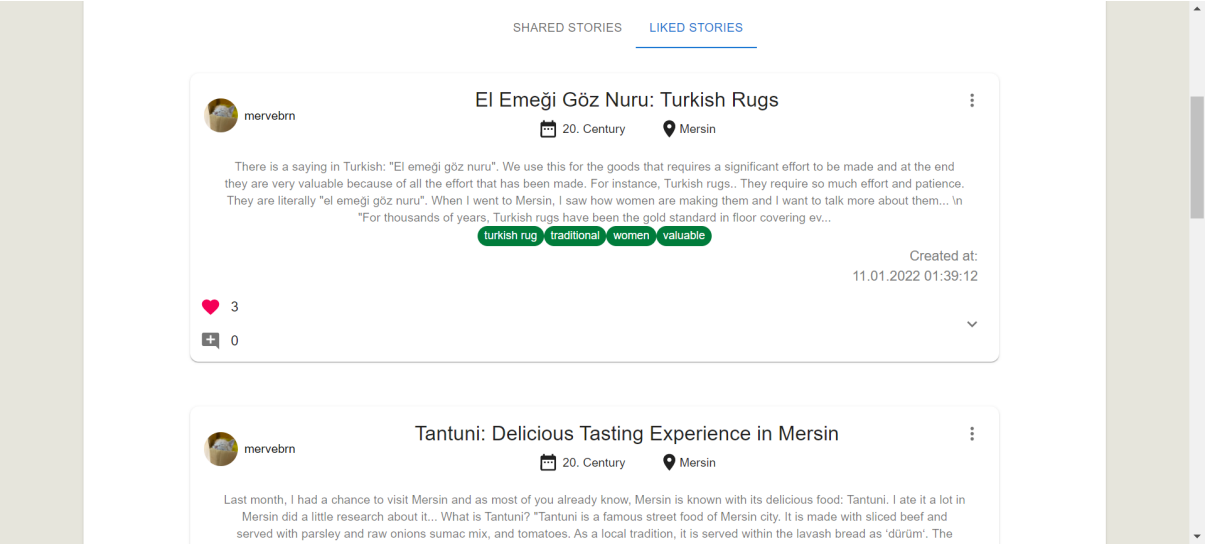
Link :
https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/frontend/src/components/PostScroll/ProfilePostScroll.js

Shared Stories:



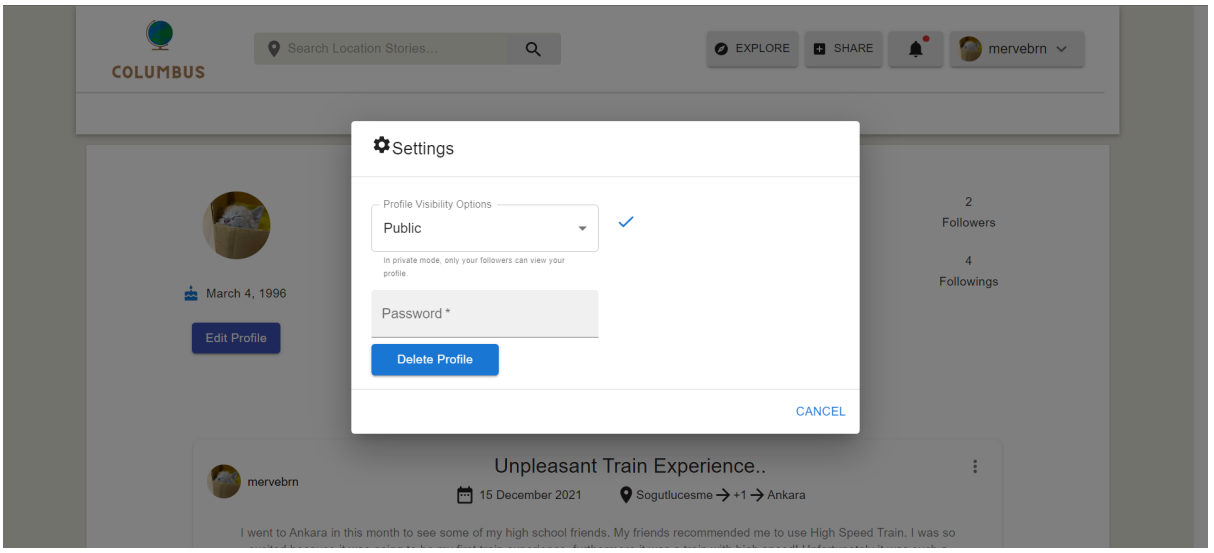
Link :
https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/frontend/src/components/PostScroll/LikedPostScroll.js

Liked Stories:



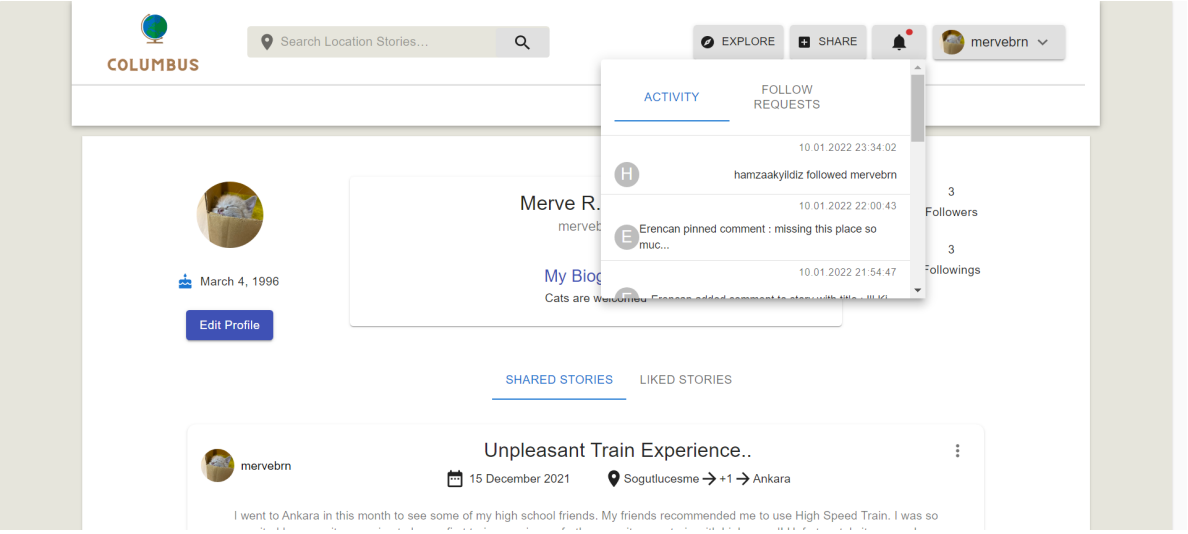
Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/components/Dialogs/SettingsDialog

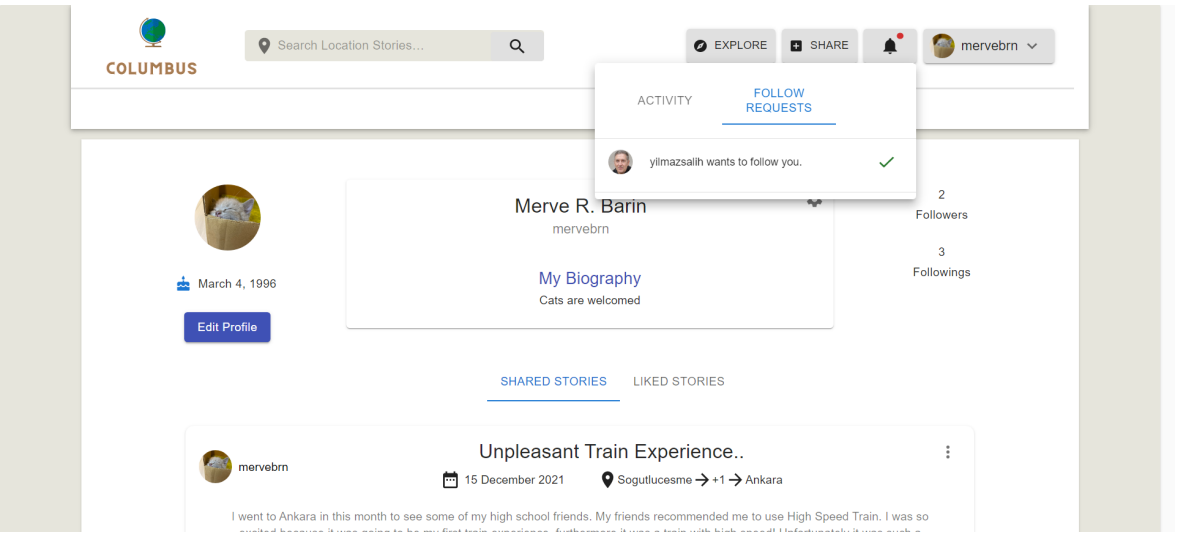
Settings:



Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/components/Notifications

Notification Box (Activities and Follow Requests):



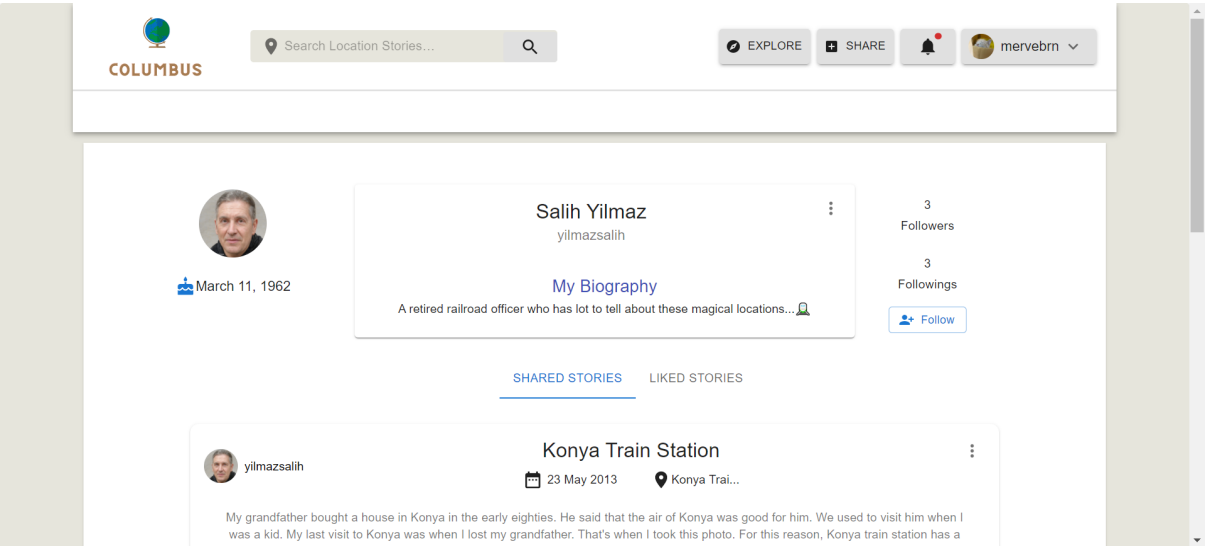


User Viewing Other Profiles

Link :

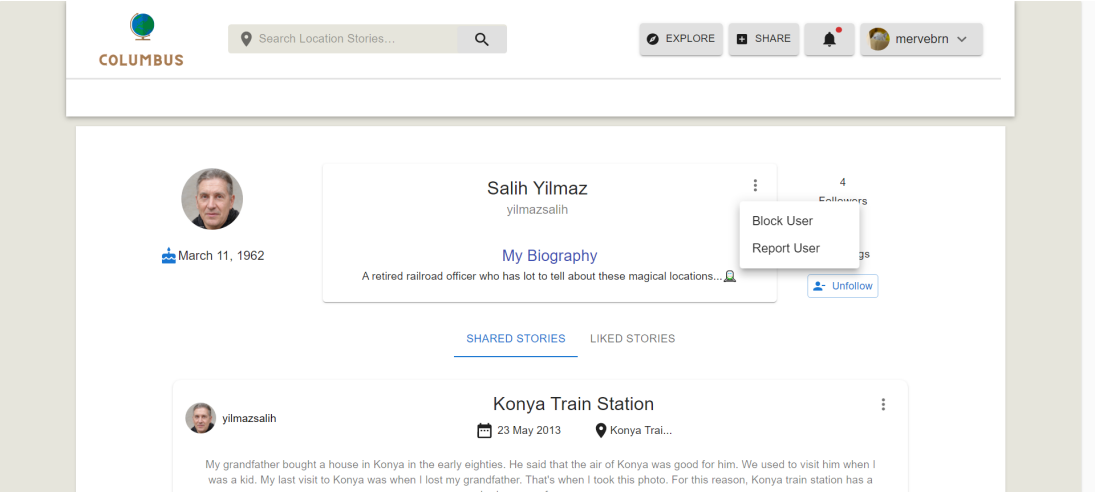
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/pages/Profile

Viewing Other Profile :



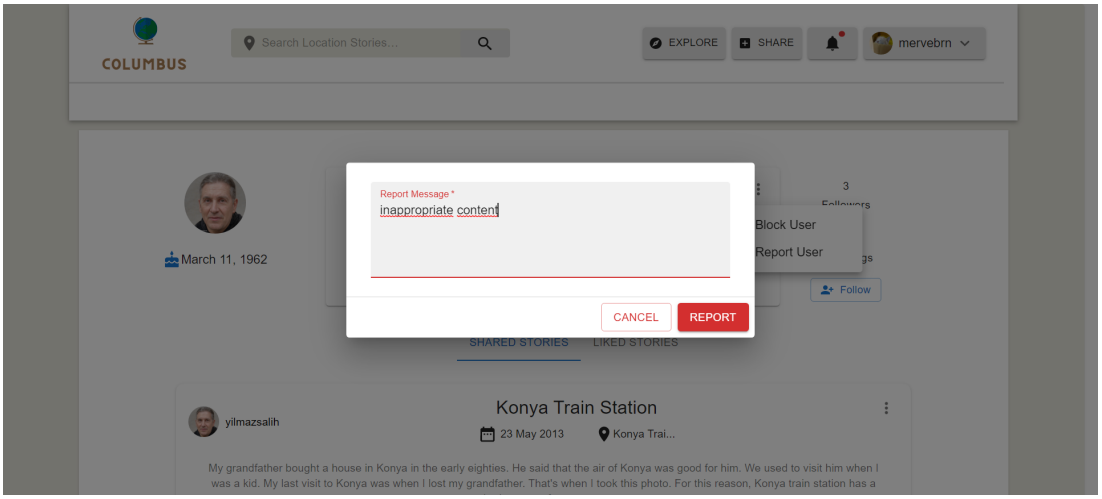
Link :
https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/frontend/src/pages/Profile/VerticalMenu.js

Report/Block User:



Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/components/Dialogs/ReportMessageDialog

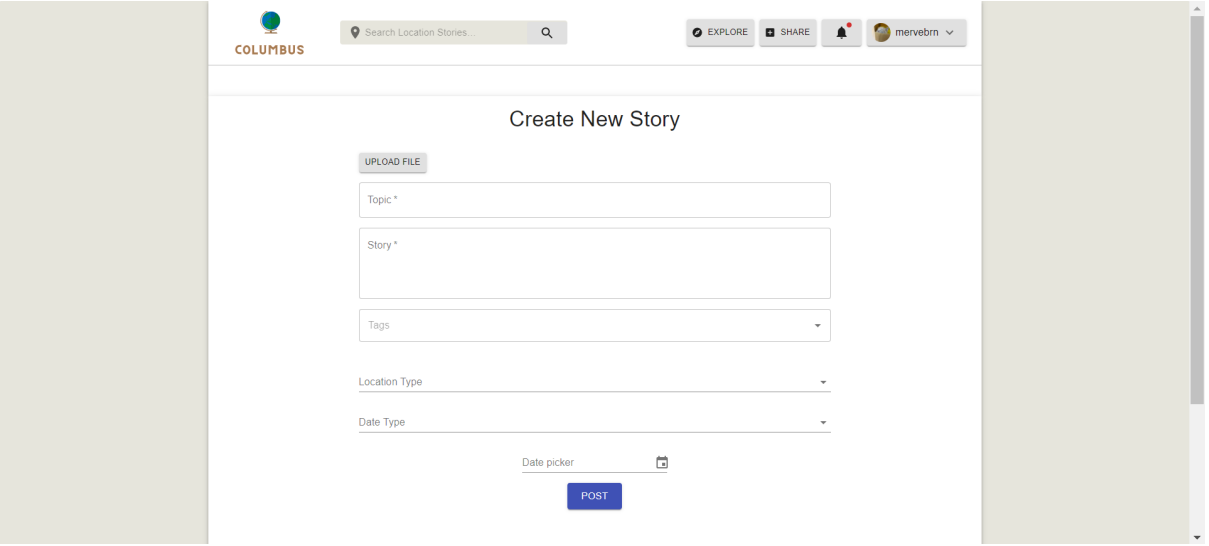
Report Message Dialog:



Create Story Page

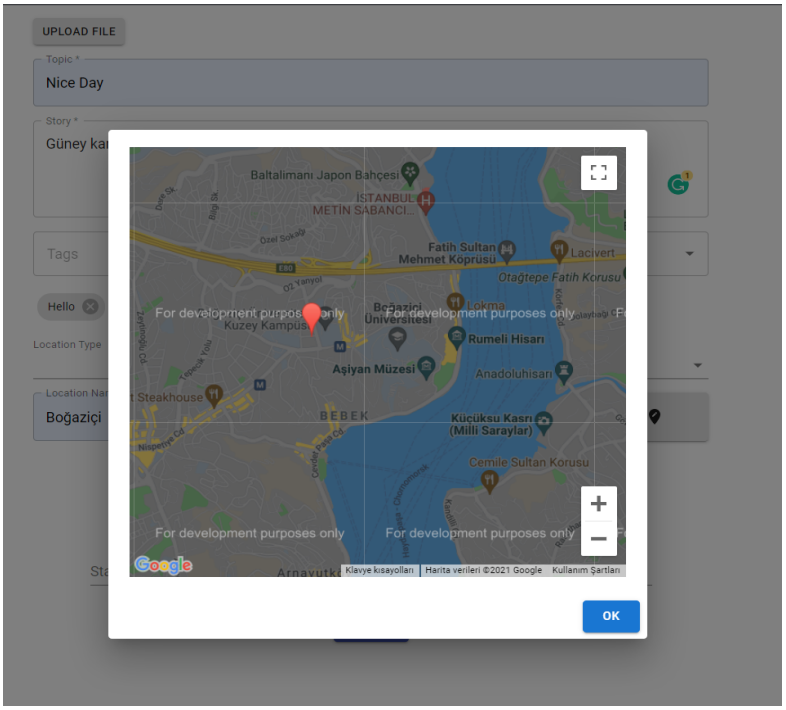
Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/pages/CreatePostPage

Screenshot :



Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/components/GoogleMaps
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/components/Dialogs/MapsDialog

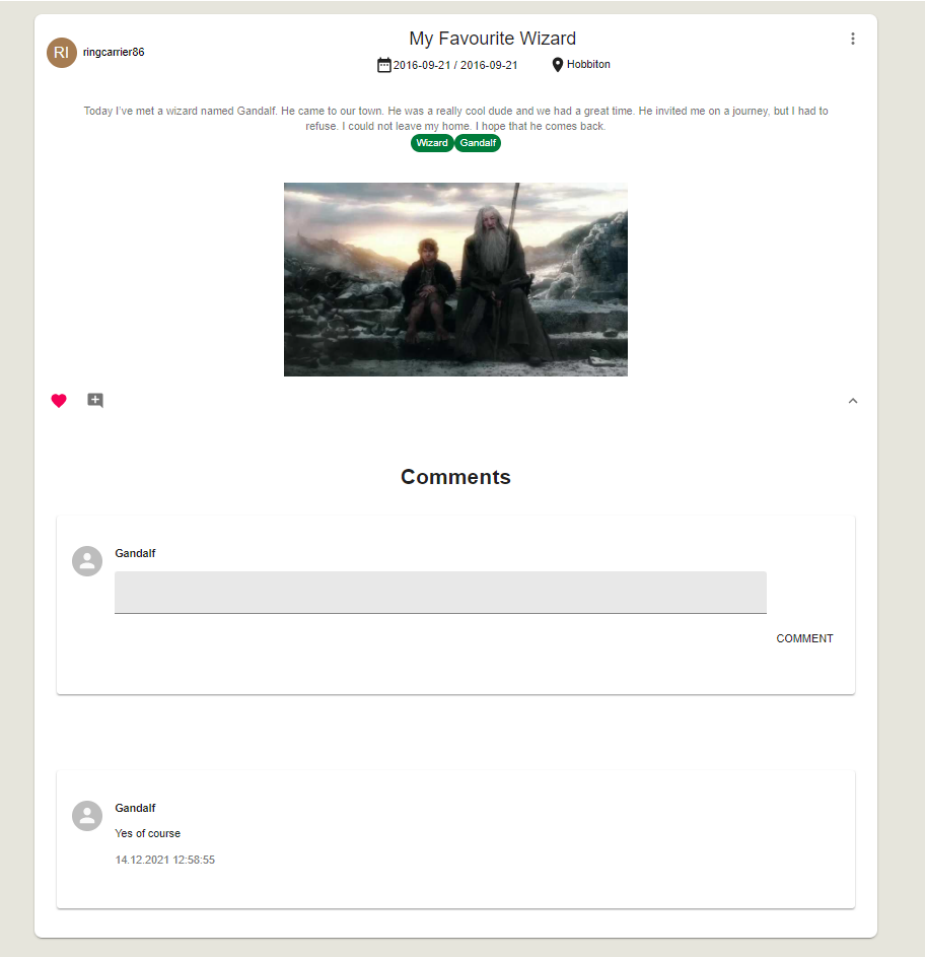
Map Dialog in Create Story:



Post component with comment and like feature

Link :
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/components/Post

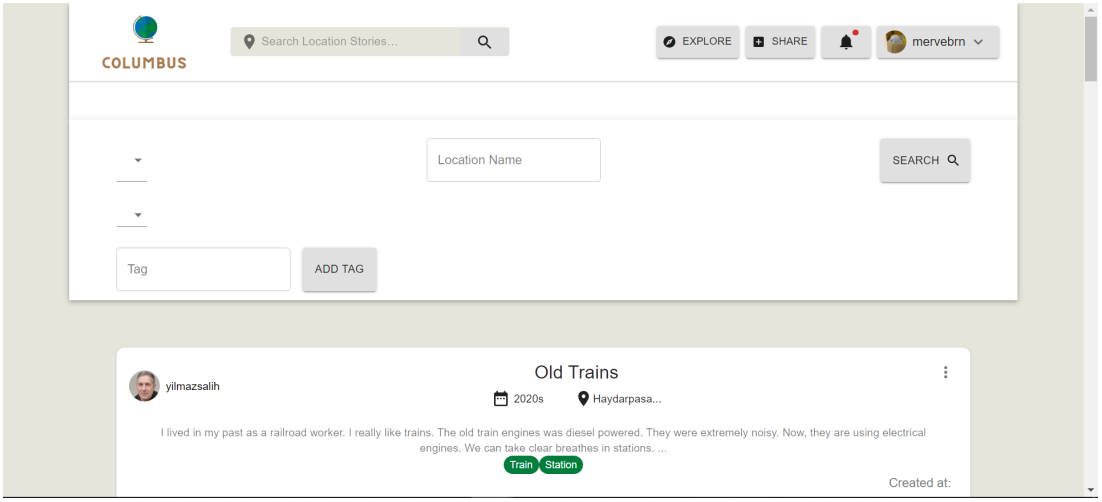
Screenshot :



Search Page

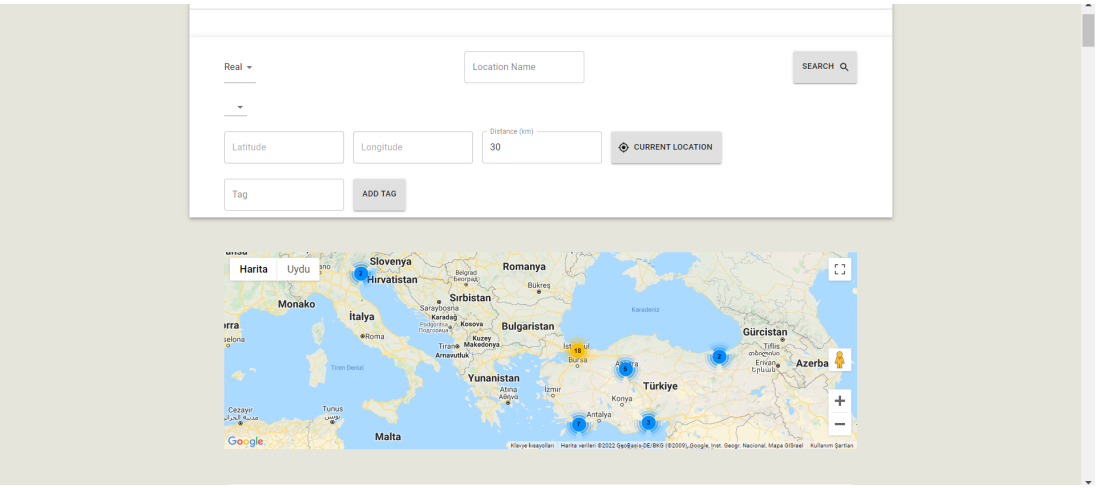
Link:
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/pages/Search

Search Page:



Link:
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/components/GoogleMaps

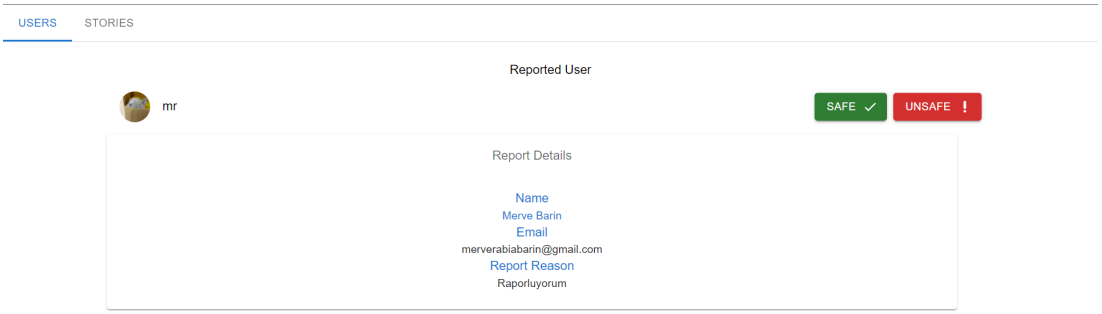
Maps:



Admin Page

Link:
https://github.com/bounswe/2021SpringGroup7/tree/master/Columbus/frontend/src/pages/Admin

Admin Page:

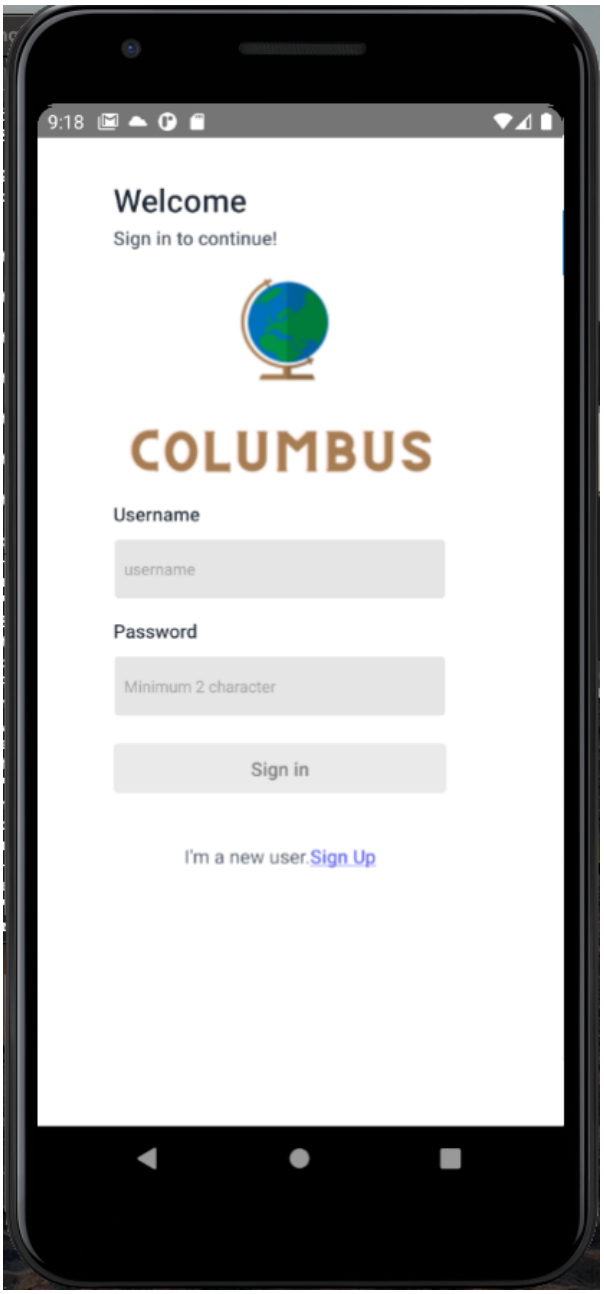


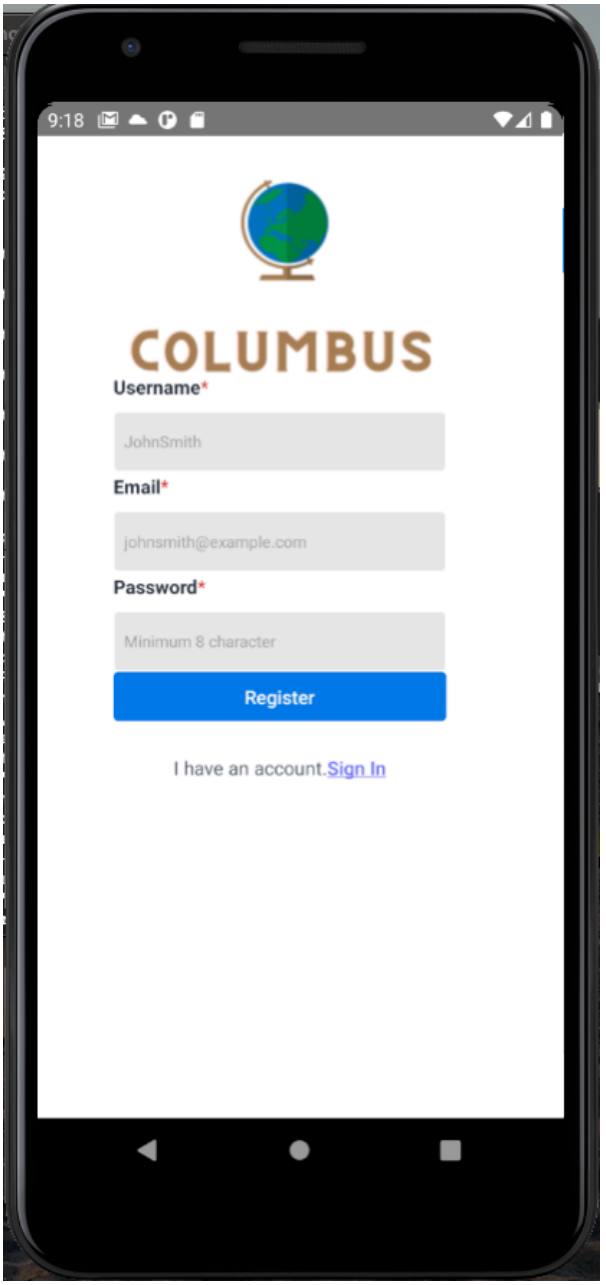
MOBILE INTERFACES

Login Page and Register Page:

Login:https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/Login/Login.js

Register:https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/Register/Register.js

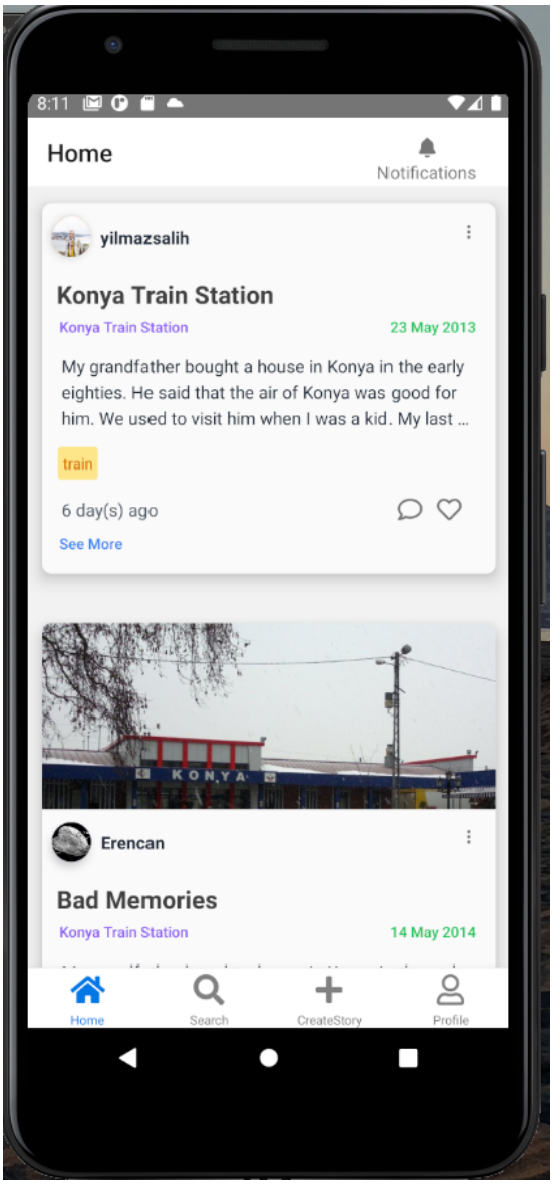


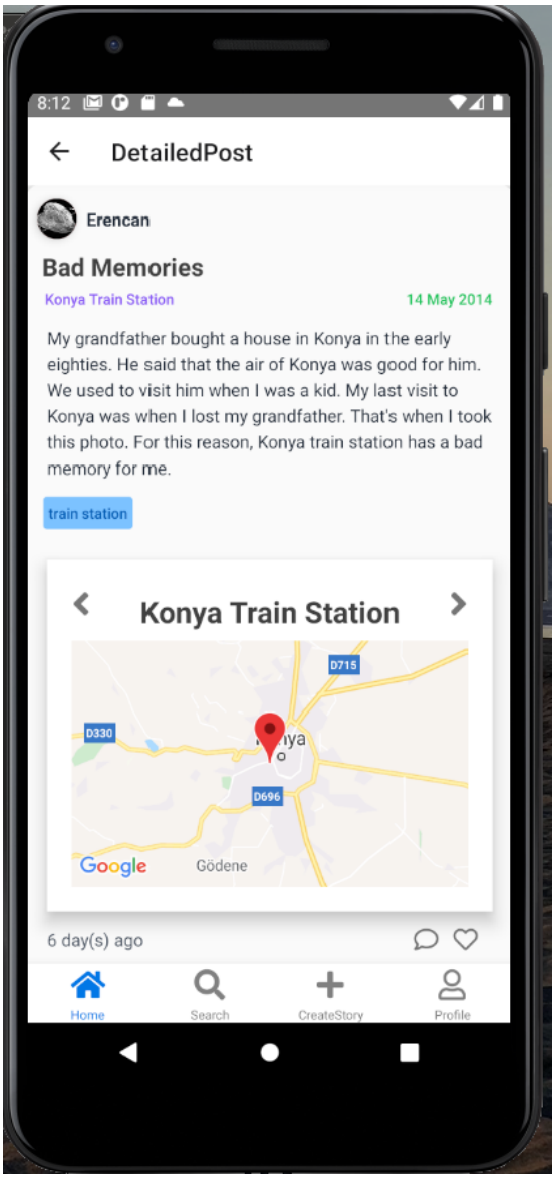


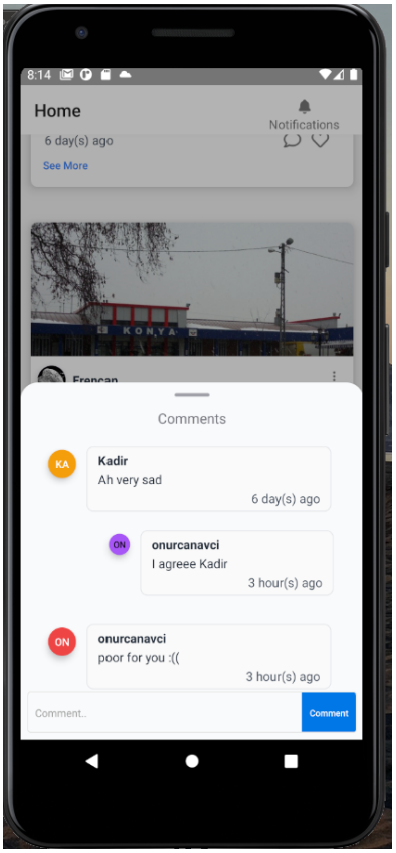
Home Page and Post Detail Page:

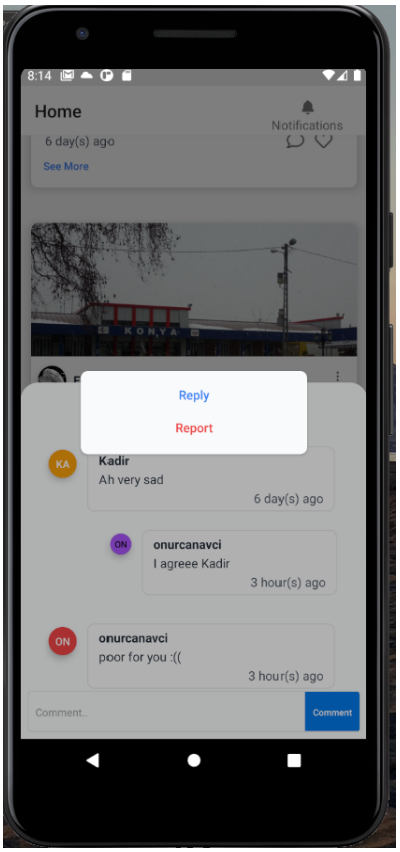
Home:https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/Home/Home.js

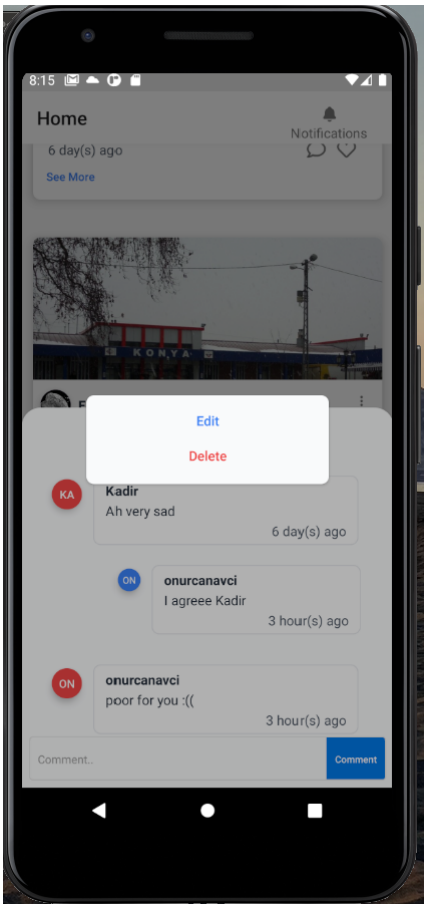
Post-Detail:https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/DetailedPost/DetailedPost.js

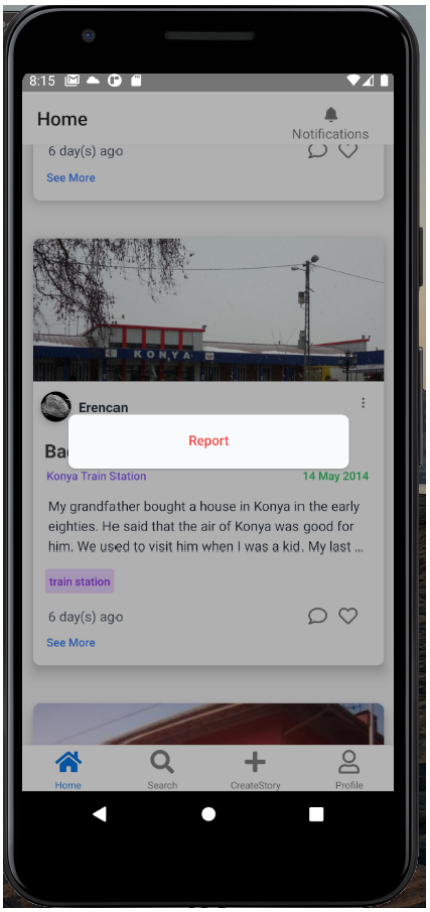





8:14
Home
Notifications
6 day(s) ago
See More
Comments
KA
Kadir
Ah very sad
6 day(s) ago
ON
onurcanavci
I agreee Kadir
3 hour(s) ago
ON
onurcanavci
poor for you :((
3 hour(s) ago
Comment..
Comment

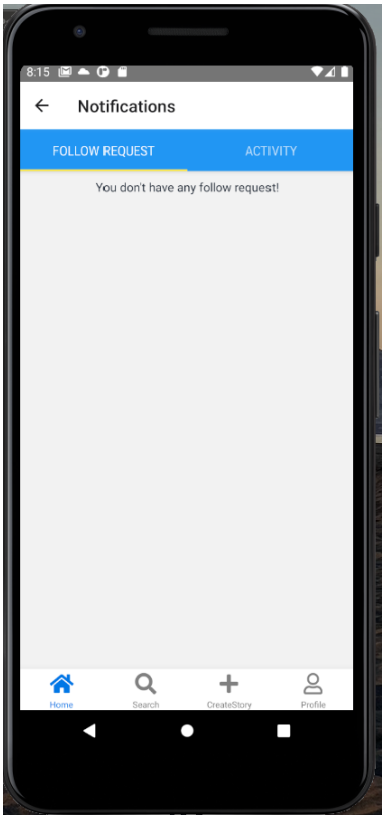

8:14
Home
Notifications
6 day(s) ago
See More
Reply
Report
KA
Kadir
Ah very sad
6 day(s) ago
ON
onurcanavci
I agreee Kadir
3 hour(s) ago
ON
onurcanavci
poor for you :((
3 hour(s) ago
Comment..
Comment

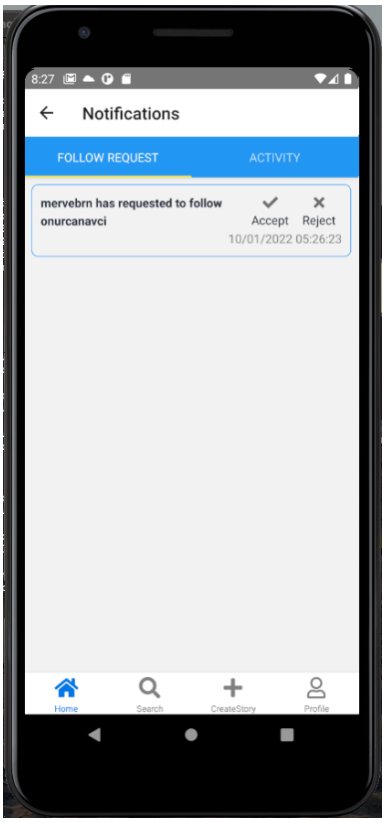

8:15
Home
Notifications
6 day(s) ago
See More
Edit
Delete
KA
Kadir
Ah very sad
6 day(s) ago
ON
onurcanavci
I agreee Kadir
3 hour(s) ago
ON
onurcanavci
poor for you :((
3 hour(s) ago
Comment..
Comment

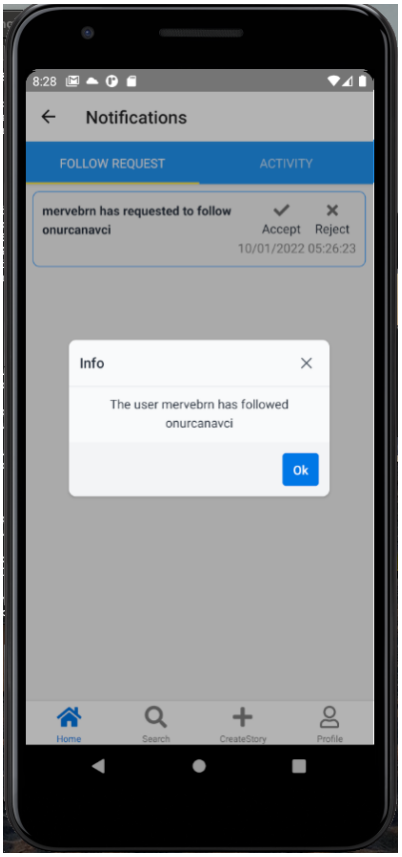

8:15
Home
Notifications
6 day(s) ago
See More
Erencan
Report
Konya Train Station
14 May 2014
My grandfather bought a house in Konya in the early eighties. He said that the air of Konya was good for him. We used to visit him when I was a kid. My last ...
train station
6 day(s) ago
See More
Home
Search
CreateStory
Profile

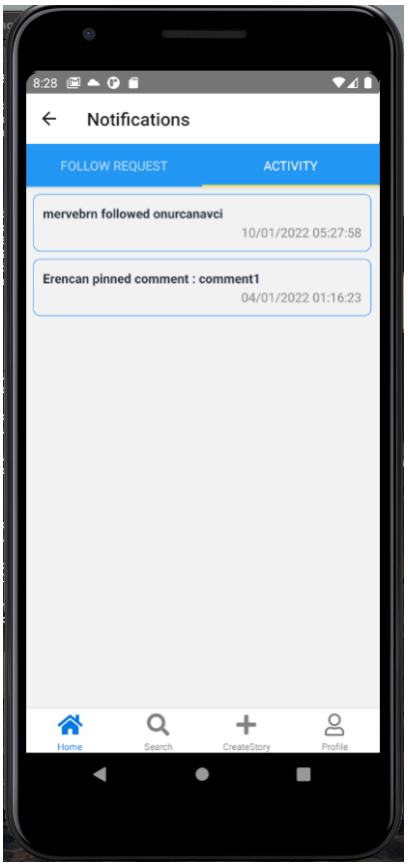
Notification Page:

Notification:https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/Home/components/Notification/Notification.js



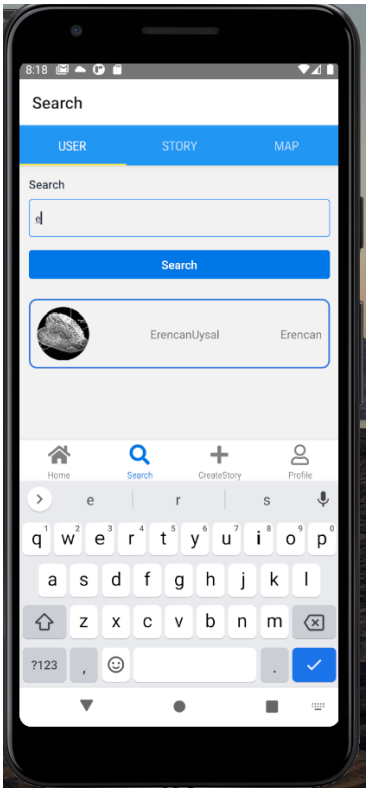


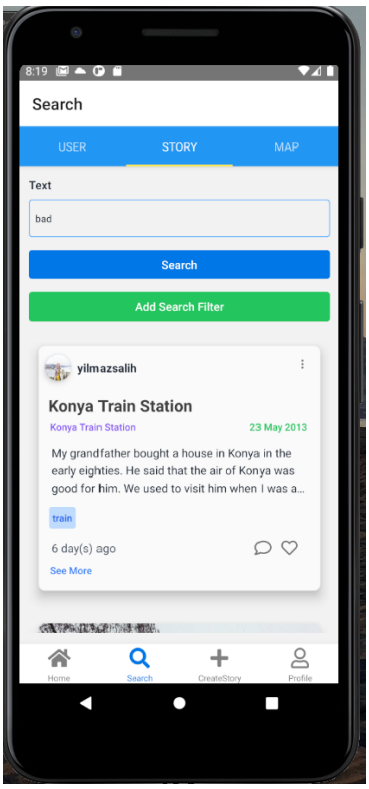


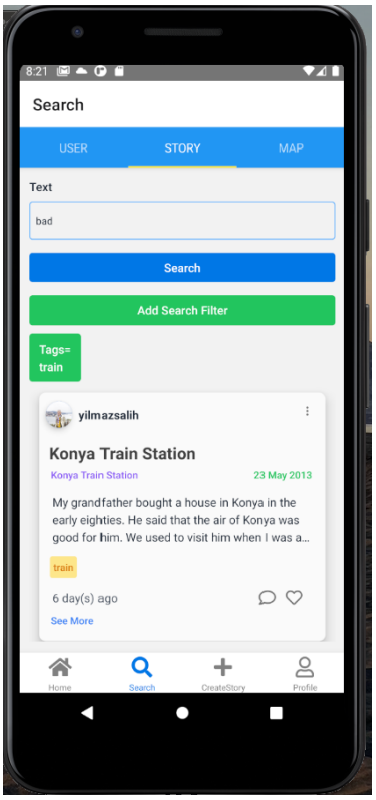


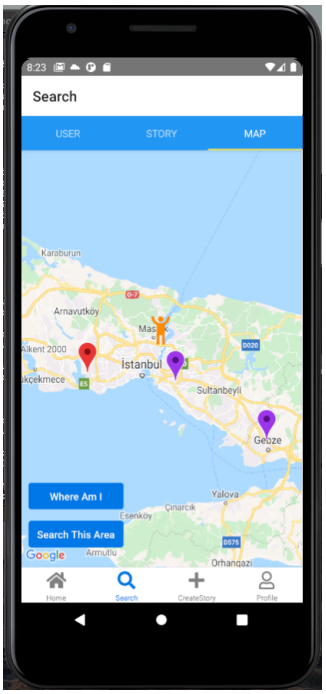
Search Page:

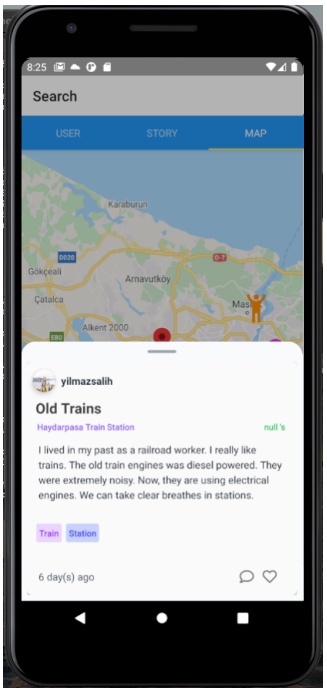
Search:https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/Search/Search.js
UserSearch(Component):https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/Search/components/LocationSearch/LocationSearch.js
LocationSeach(Component):https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/Search/components/LocationSearch/LocationSearch.js
StorySearch(Component):https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/Search/components/StorySearch/StorySearch.js





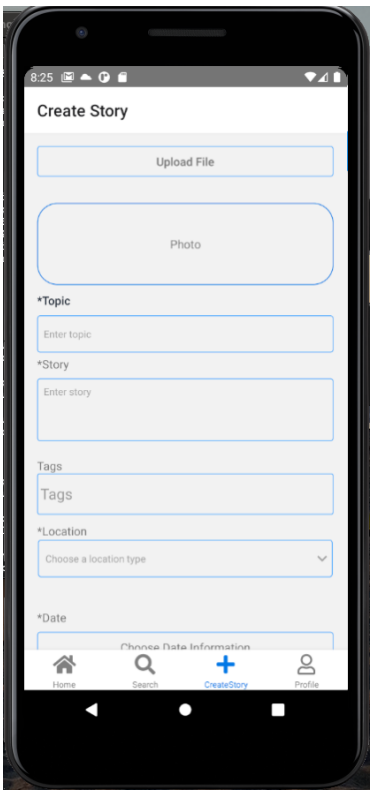


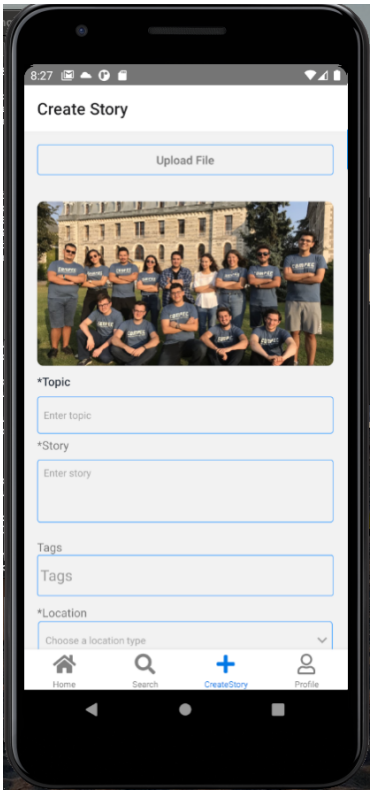


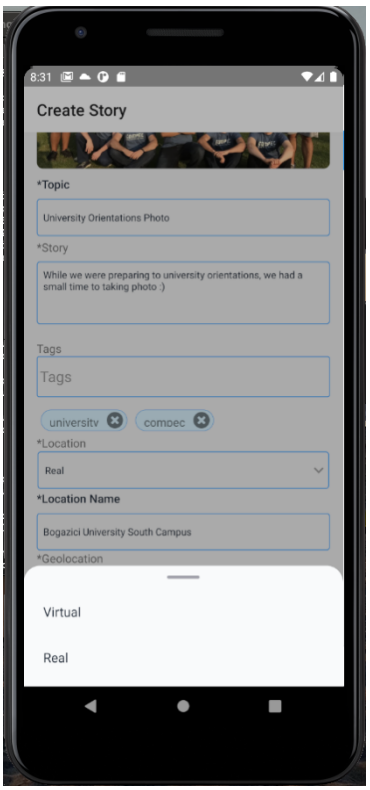


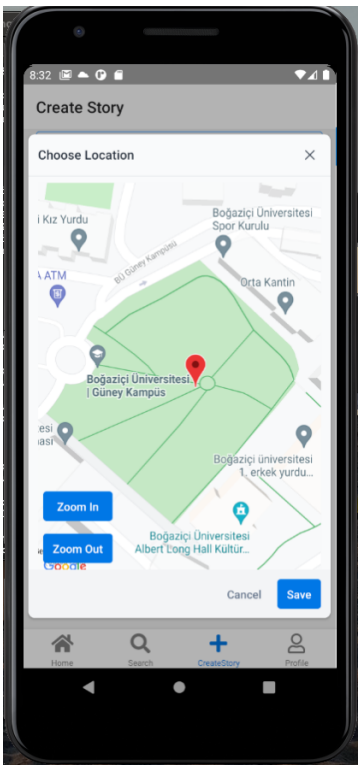
CreateStory Page:

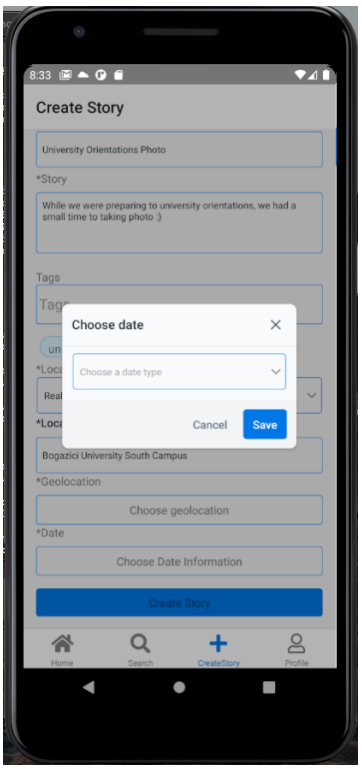
CreateStory:https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/CreateStory/CreateStory.js

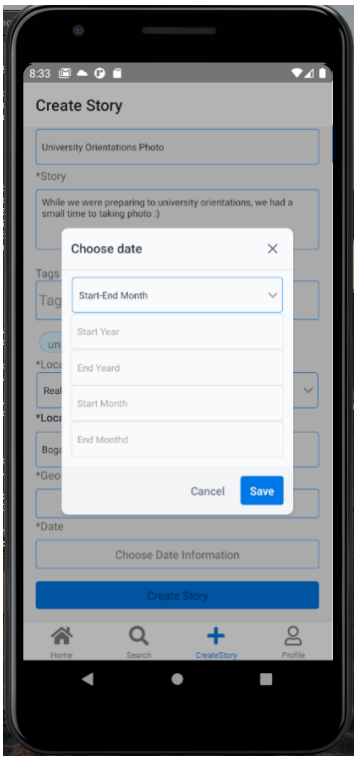


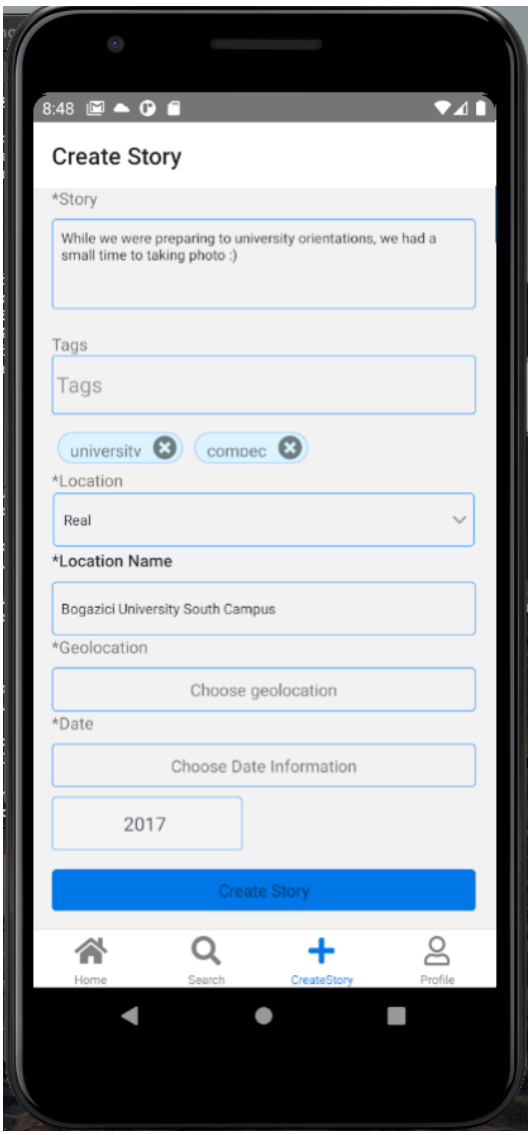


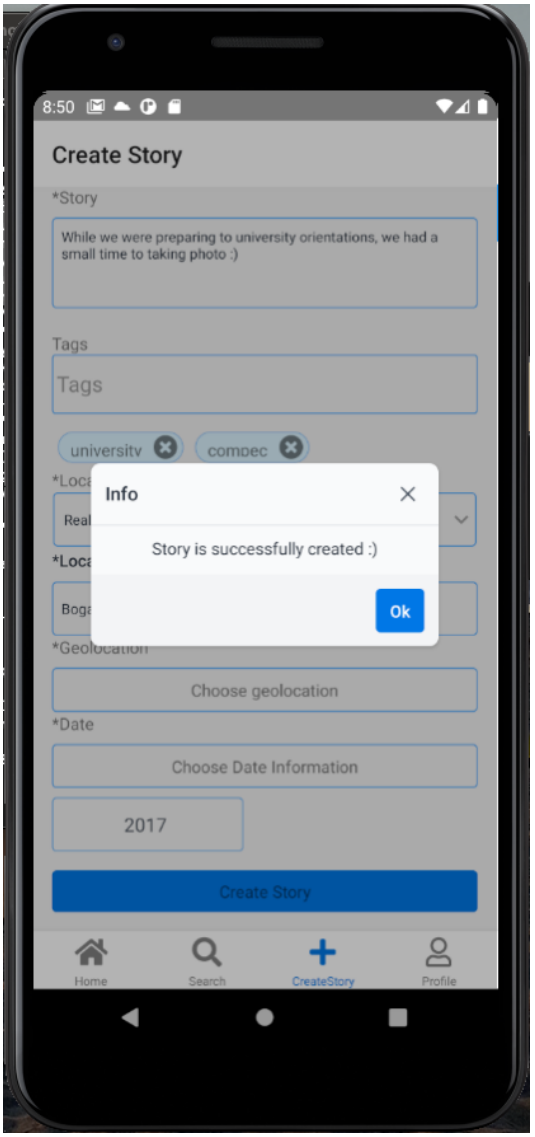








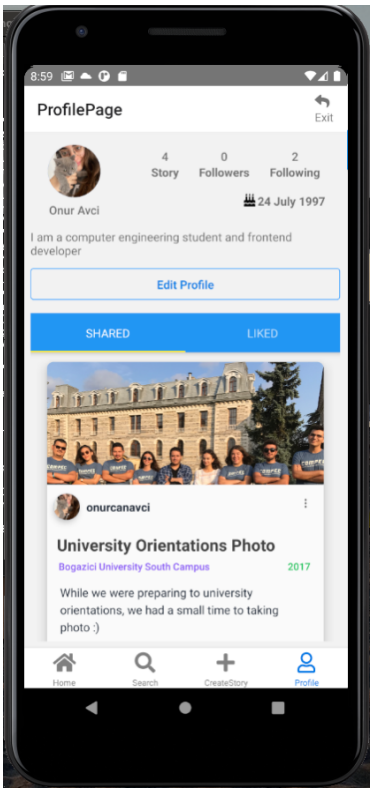

8:48
Create Story
*Story
While we were preparing to university orientations, we had a small time to taking photo :)
Tags
Tags
university
compec
*Location
Real
*Location Name
Bogazici University South Campus
*Geolocation
Choose geolocation
*Date
Choose Date Information
2017
Create Story
Home
Search
CreateStory
Profile

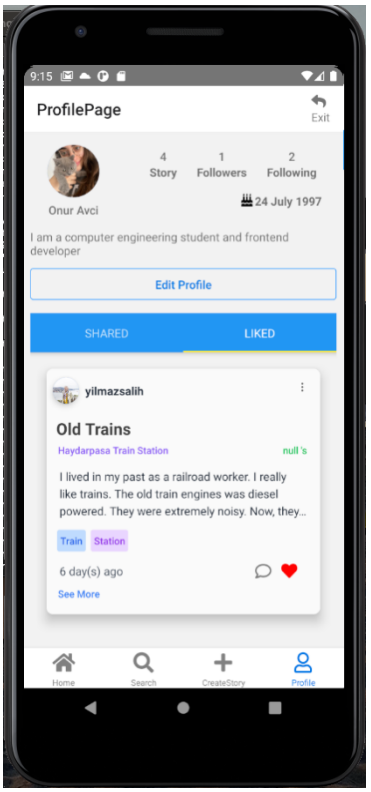

8:50
Create Story
*Story
While we were preparing to university orientations, we had a small time to taking photo :)
Tags
Tags
university
compec
Info
Story is successfully created :)
Ok
Choose geolocation
*Date
Choose Date Information
2017
Create Story
Home
Search
CreateStory
Profile

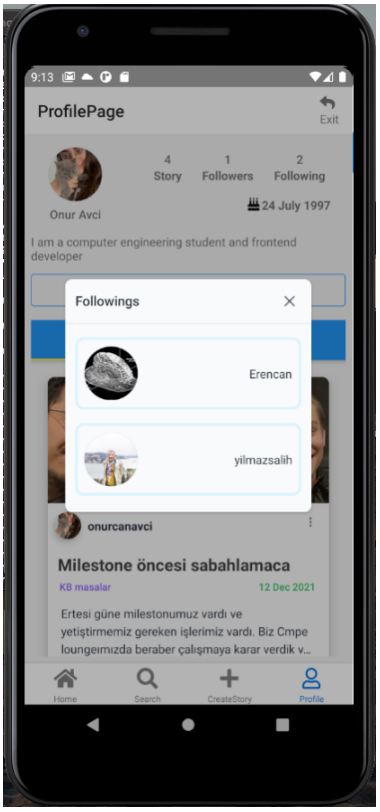
Profile Page:

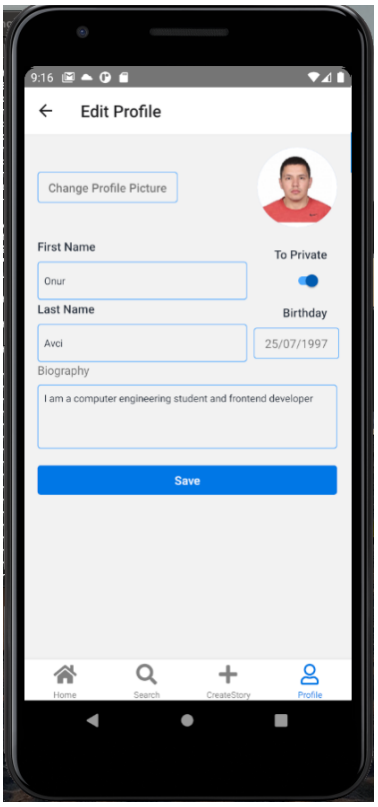
Profile:https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/Profile/Profile.js

EditProfile:https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/Profile/views/EditProfile/EditProfile.js



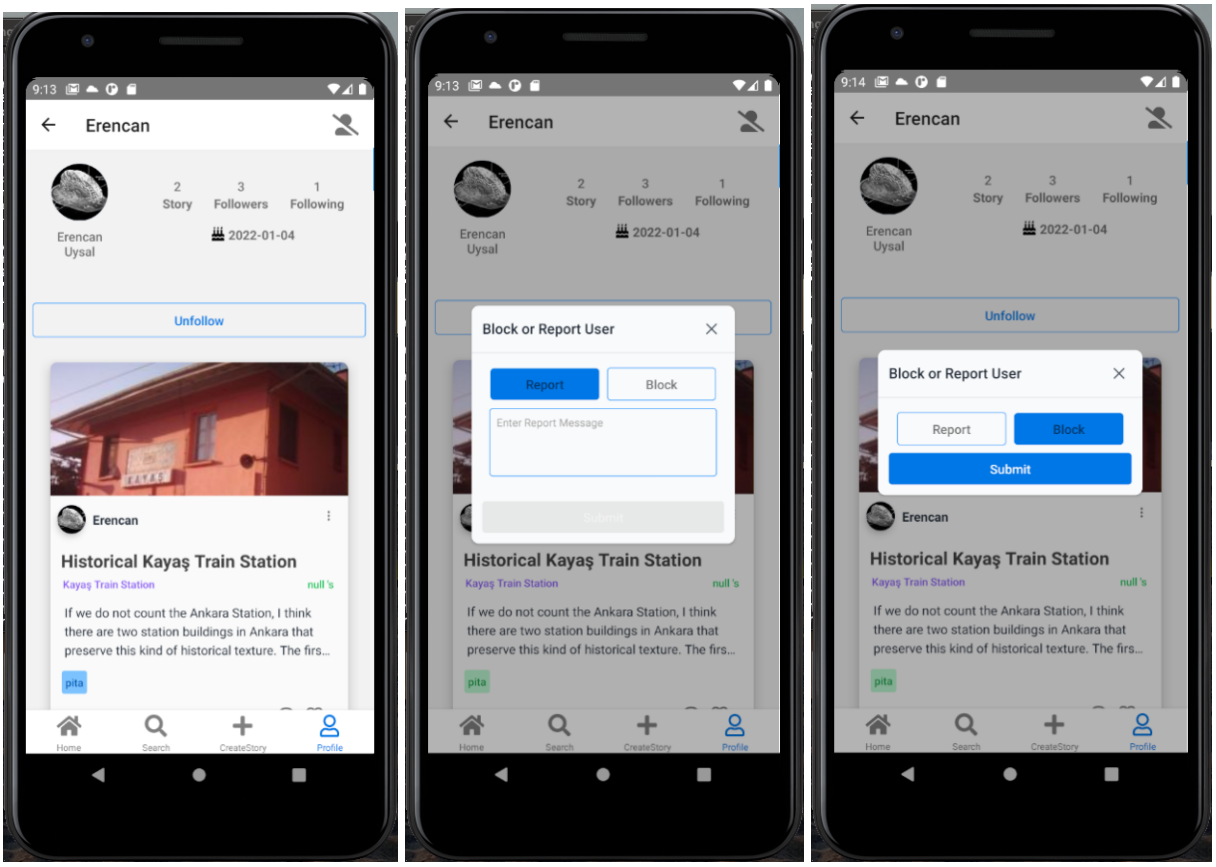






Other Profiles Page:

OtherProfile:https://github.com/bounswe/2021SpringGroup7/blob/master/Columbus/mobile/Columbus/src/views/OtherProfiles/OtherProfiles.js

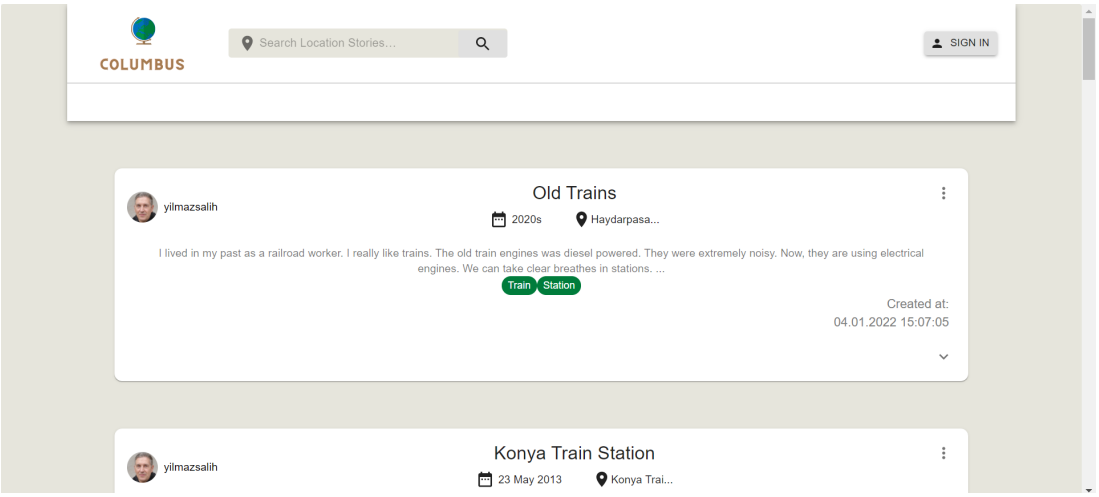


7. User Manual

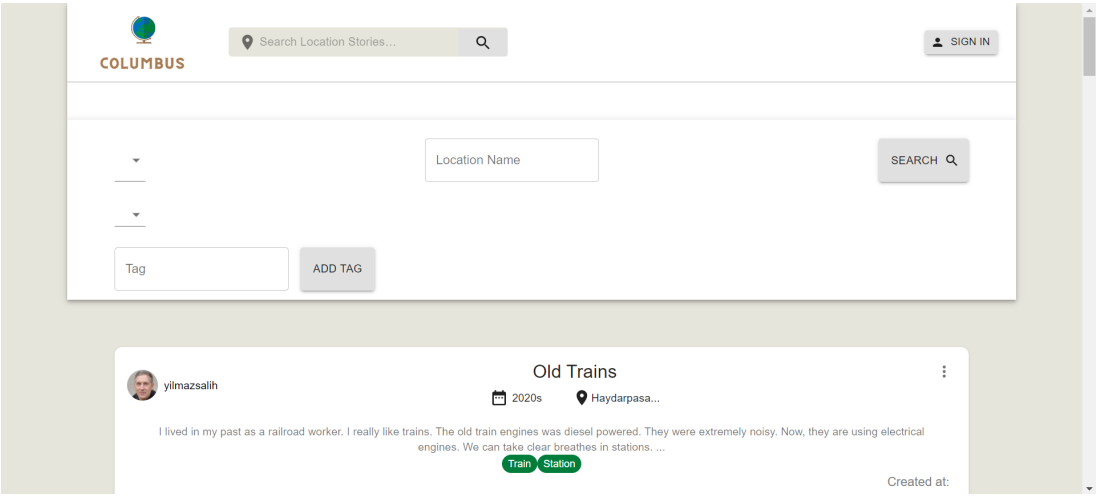
WEB

Guest

Before you register to Columbus, you are considered as '*Guest*' and able to perform a limited set of actions on the application.
You can view the explore page and the stories in it, but you are not able to like these stories or make comments on them. Also you cannot view user profiles.
Guest Page:

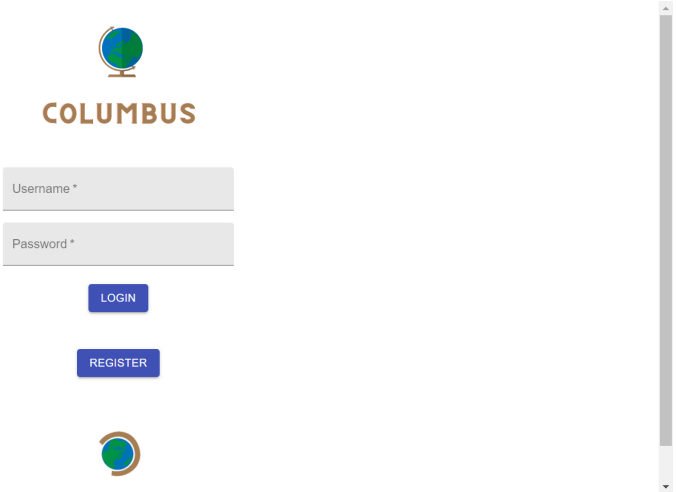


When you are a *guest,* you can still perform search functionality as users by clicking the search icon in the search bar.
Search Page:



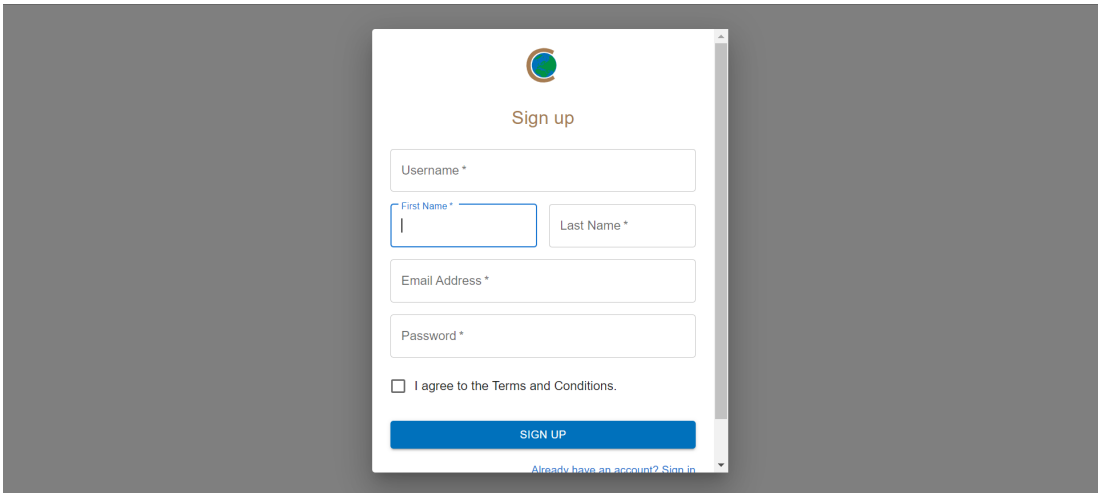
REGISTER

If you want to create a profile for yourself, you can click the Sign In button where it redirects you to the Login/Register page.
Login Page:

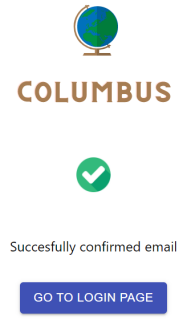


Click the Register button to open the Register dialog. You can register if you provide your username, first name, last name, email address and a password. Also, you must accept the Terms and Conditions to register.

Register Dialog:



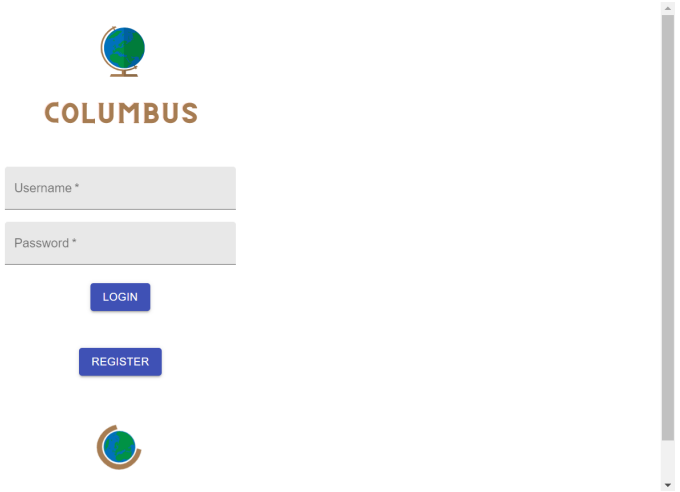
After clicking the Sign Up button, a verification code will be sent to your email. When you click this link, you will be able to login to the Columbus application with your credentials.



User

LOGIN

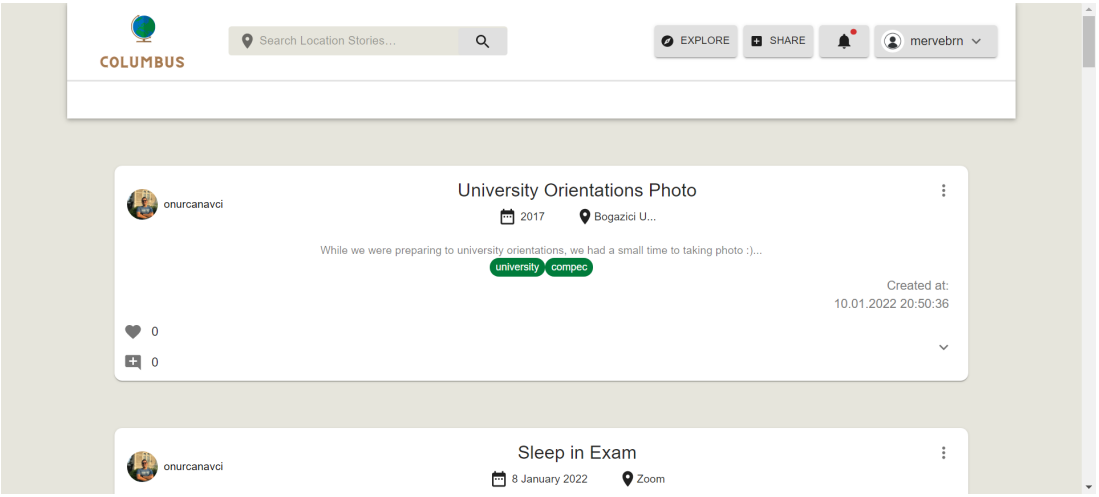
If you have already created an account for Columbus, you are considered as a *User*. You should login by entering your username and password.



HOME

When you logged in to the application, the home page will welcome you. This page contains posts from your followings. You can access all functionalities of posts from here.

Home Page:

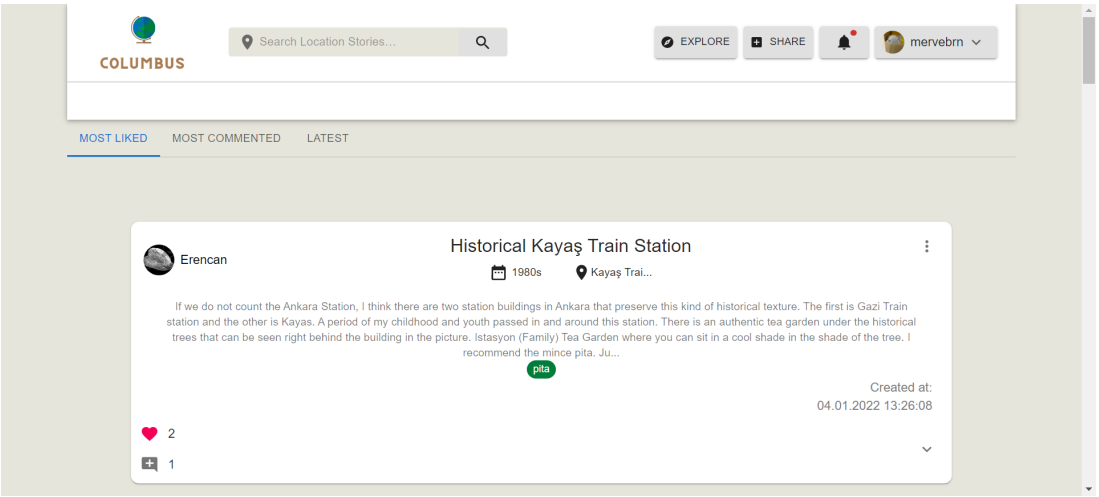


You can also reach that page by clicking the Columbus logo on the left side of the header.

EXPLORE

To explore new stories, you can click the Explore button in the header. There are three tabs displaying other user stories: Most Liked, Most Commented, Latest.

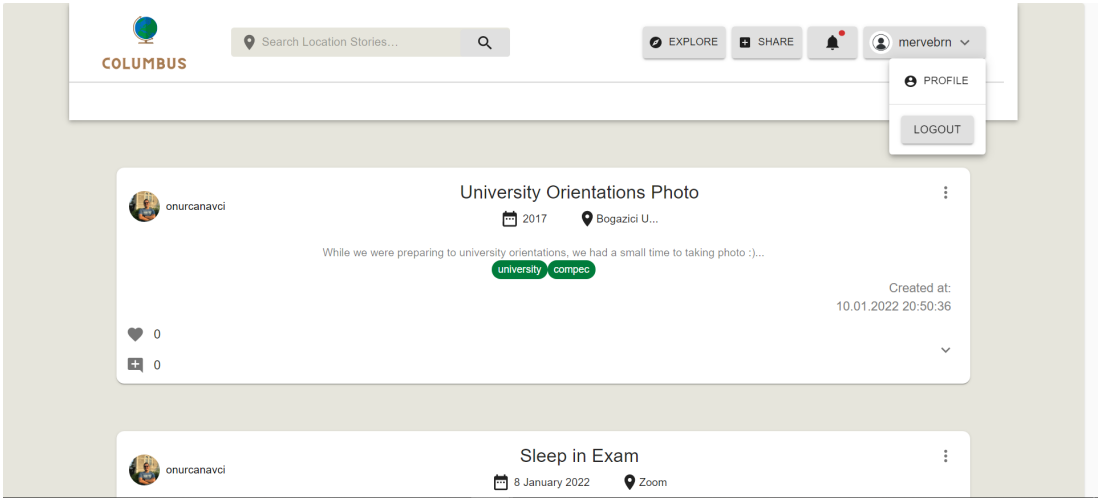
Explore Page:



USER'S OWN PROFILE

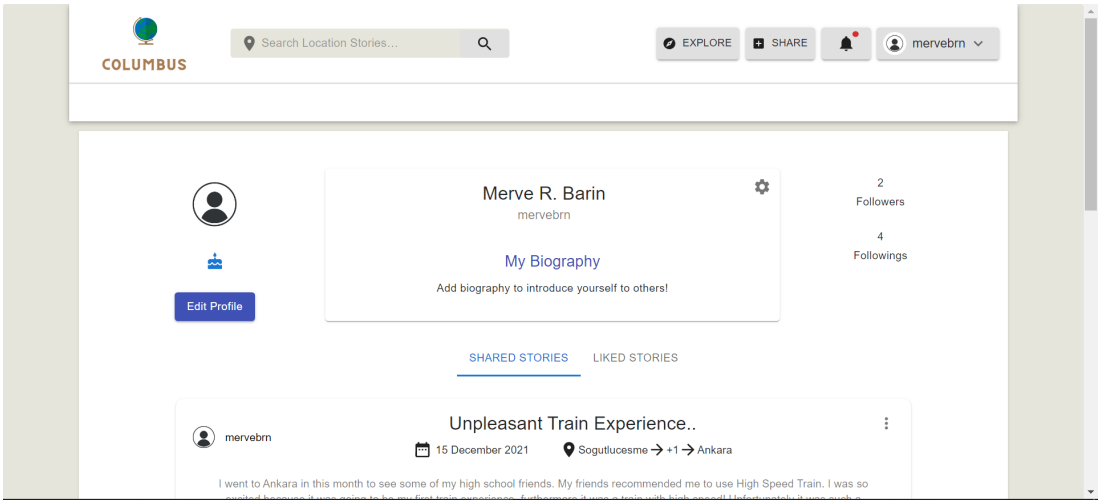
To access your profile, you can click the Profile button from the menu in the header.

Profile Navigation Button:



In your profile page, you can view/edit your profile information, view posts (stories) that you have shared and you have liked, view followings and followers,

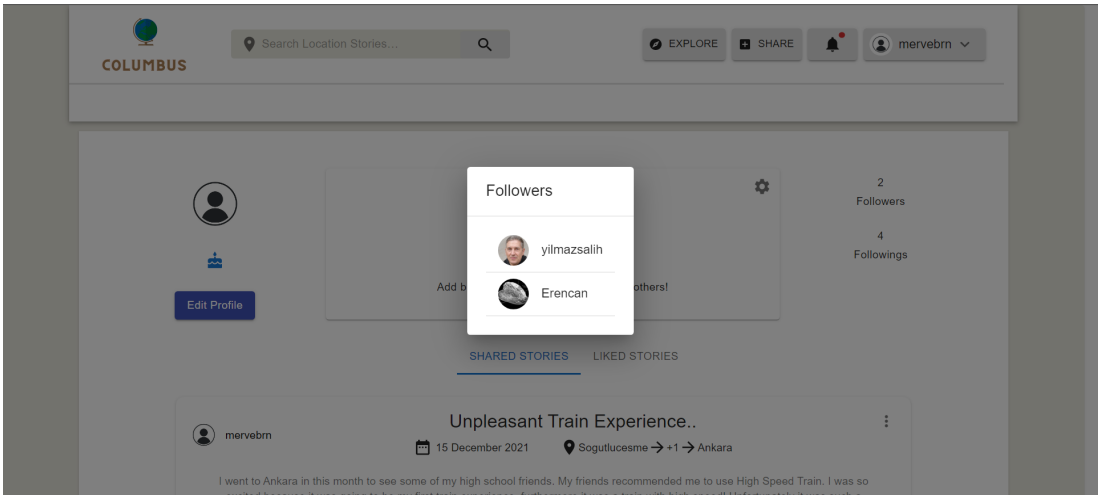
Profile Page:



Followers/Followings

You can see your followers or followings by clicking the relevant button. The dialog you will see when you click the *Followers* button is as below. You can also navigate to another user's profile by clicking their username.

View Followers:

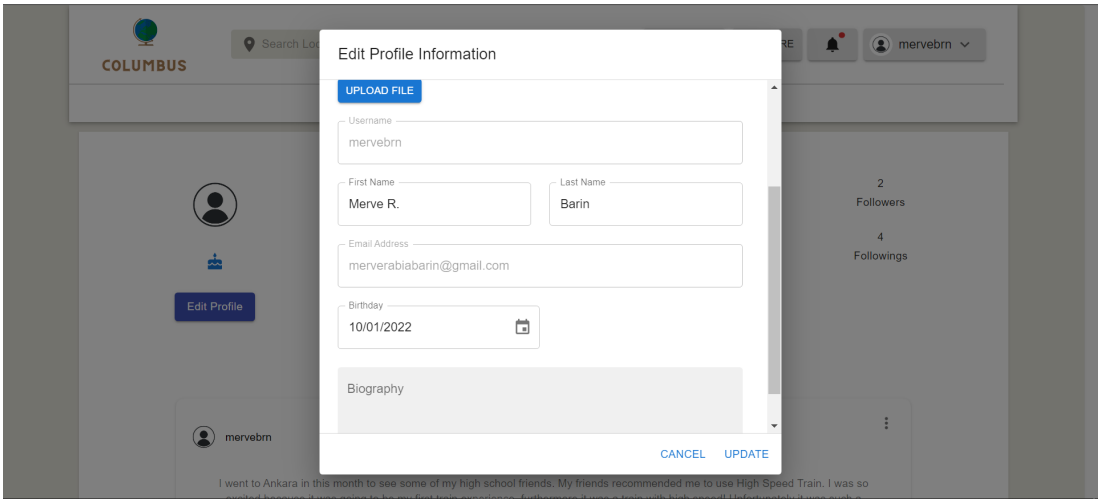


Edit Profile Information

If you want to change or add something new to your profile information, you may click the Edit *Profile* button.

You can add/change profile photo, first name, last name, birthday, biography. *You cannot change your username, email or password.*

Edit Profile Information:



Profile Visibility

There are two profile visibility options for a user's profile: ***Public*** and ***Private***.

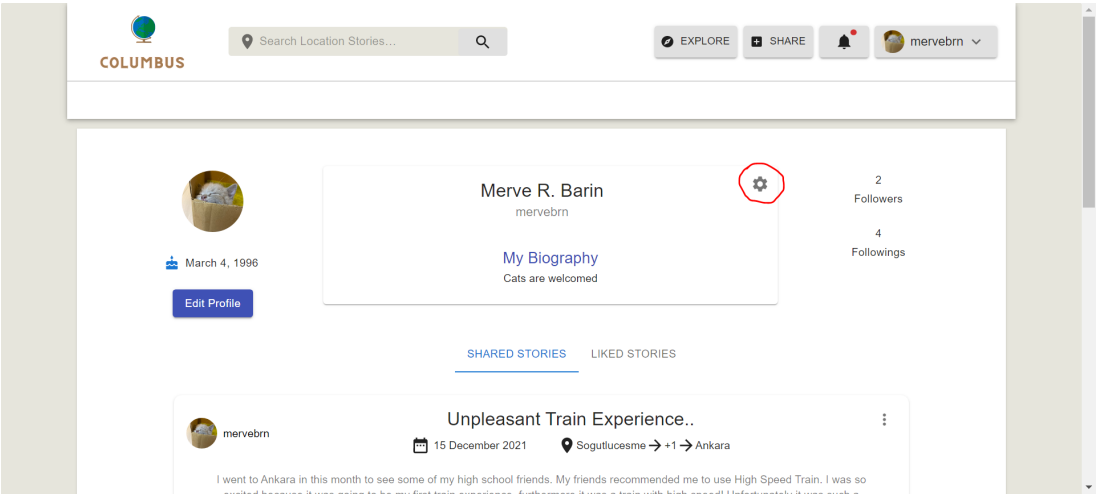
Public Mode

All users can see profile information, the posts and followers/followings of public users.

Private Mode

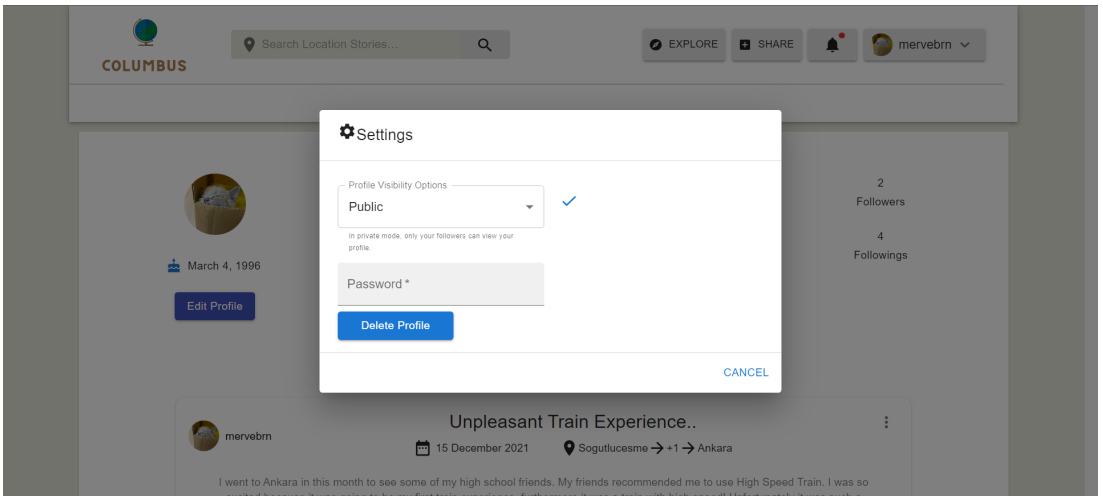
In private mode, only your followers can view your posts and profile information.

When you register for the Columbus app, your profile is *public in default*. You can change it from Settings. Click settings icon to open Settings Dialog.

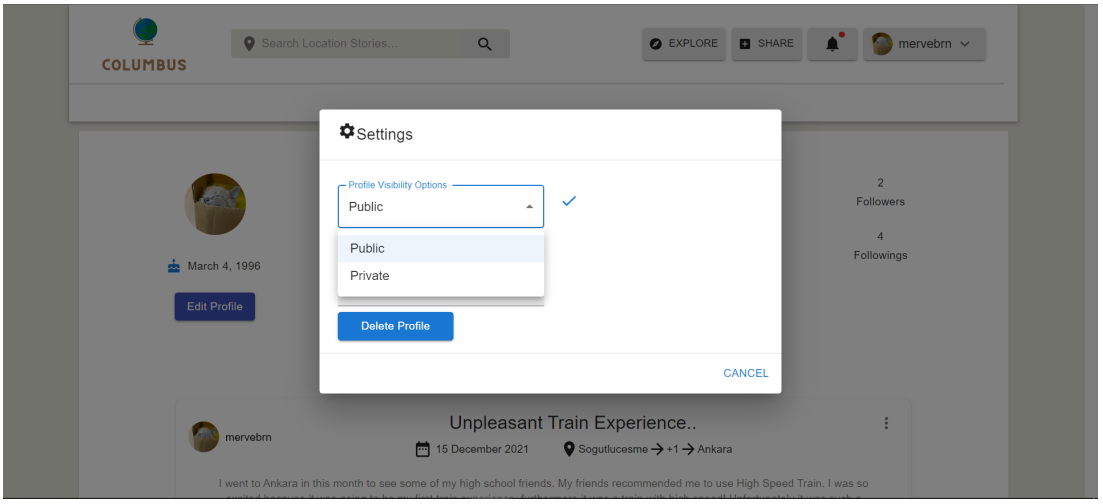


You can also delete your profile in Settings by entering your password.

Settings:



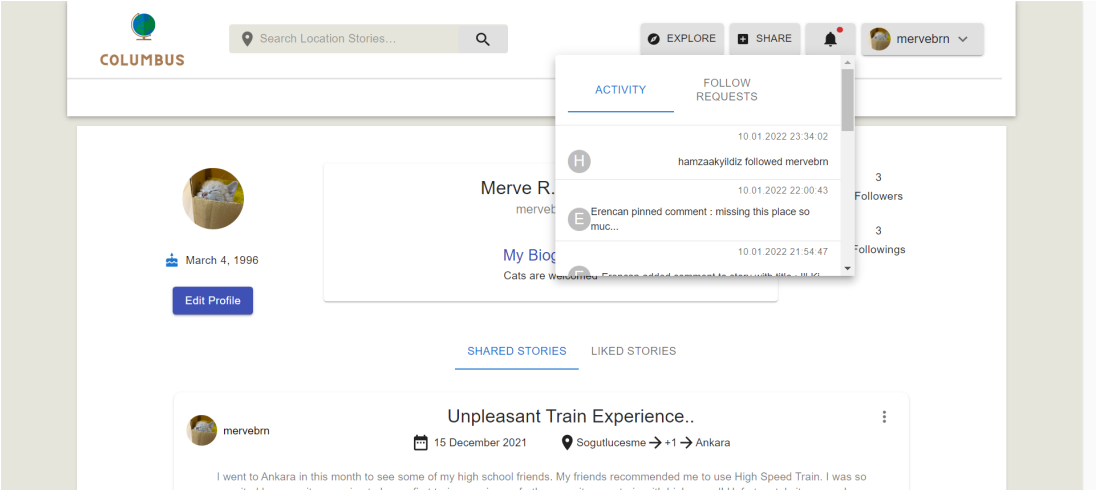
Profile Visibility Modes in Settings:



Notification Box

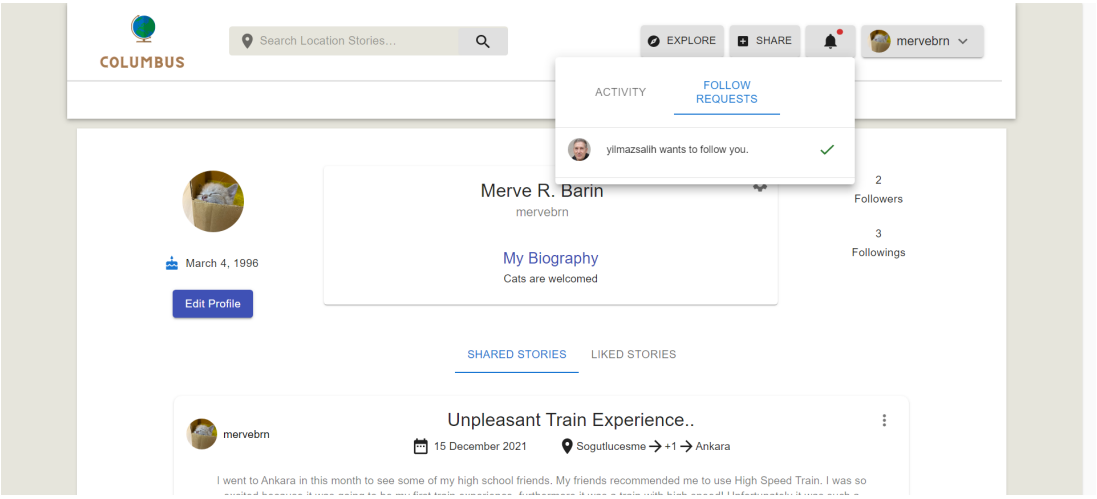
Activities: A user can view the system activities of its followings and followers in Activities tab. You can access Activities by clicking the Notification button in the header.

Viewing Activities:



Follow Requests: If you have a private profile, other users should send a request to follow you. You can see these users in Follow Requests tab.

Viewing Follow Requests:



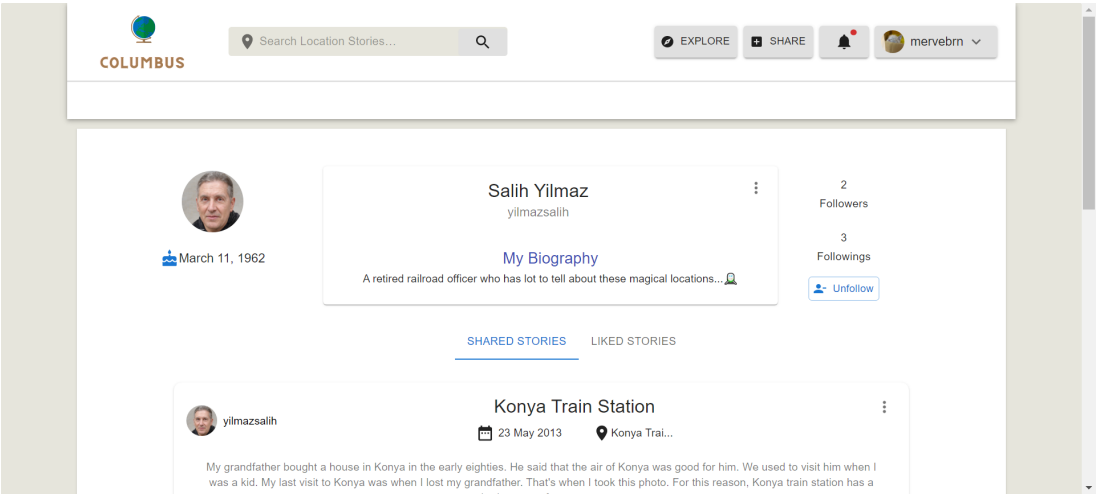
OTHER USER'S PROFILE

Users of the Columbus application are able to view the profiles of other users.

Public Profiles

If a user has a public profile, other users can view its profile information (*username, first name, last name, biography, birthday, profile photo*), posts that the user has shared and liked, user's followers and followings.

Viewing a Public Profile:

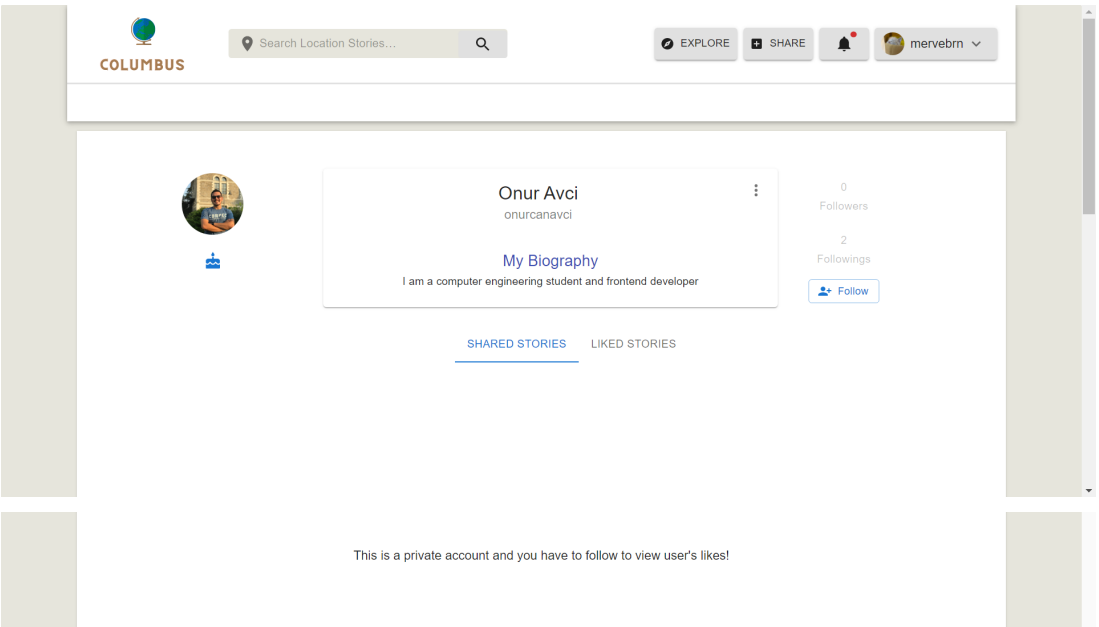


Private Profiles

If a user has a private profile, other users that do not follow the user can only view its *username, first name, last name, biography, profile photo.* They cannot view the posts that the user has shared and liked, the user's followers and followings.

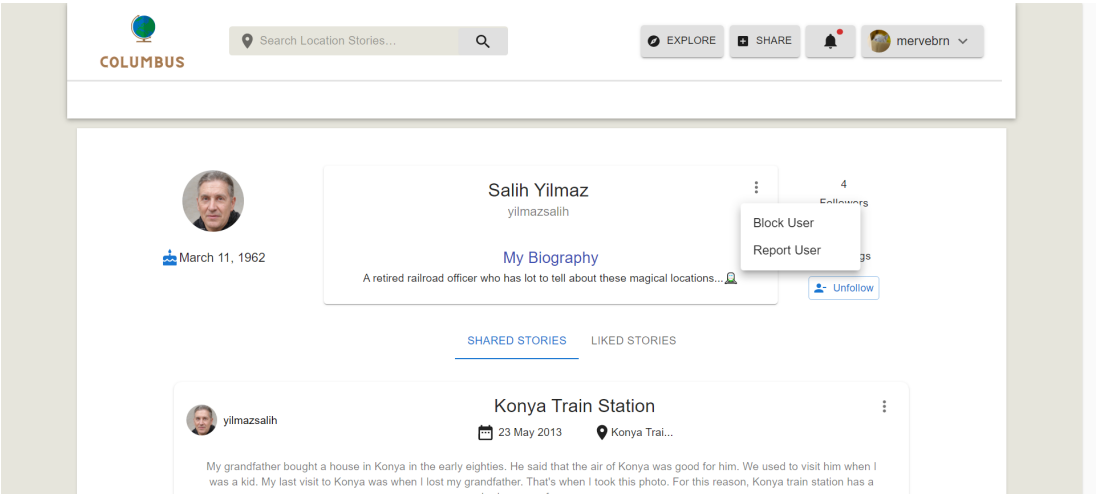
You can *send a follow request* by *clicking the Follow button* and wait for the user to accept your follow request.

Viewing a Private Profile:



Block Other User

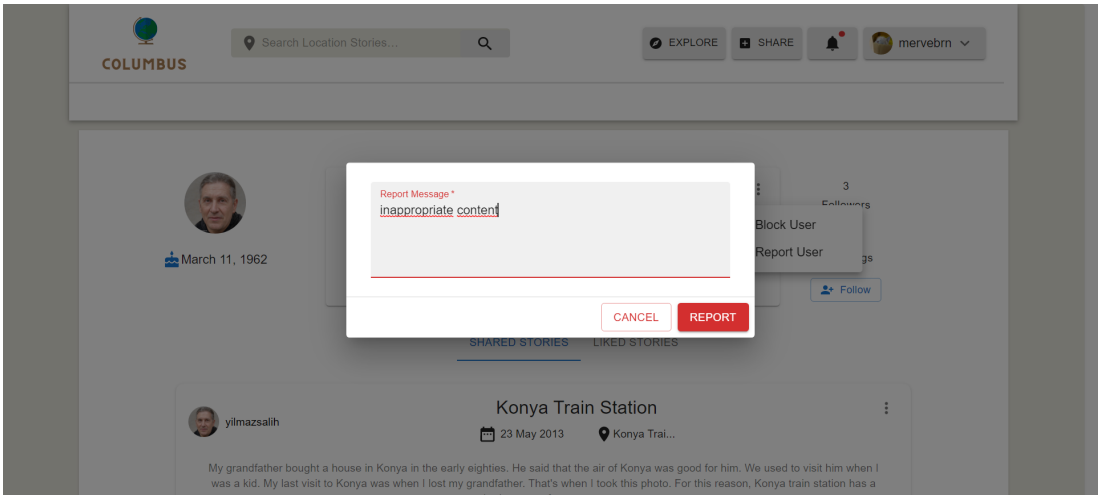
You can block a user by clicking the Block User button in the three dot icon.



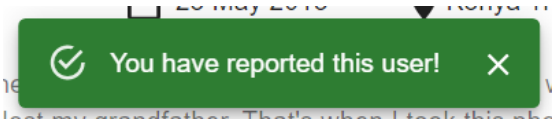
Report Other User

You can report a user by clicking the Report User button in the three dot icon. You may enter the reason for the report in Report Message field. This message will help admins to consider the report.

Report Message:



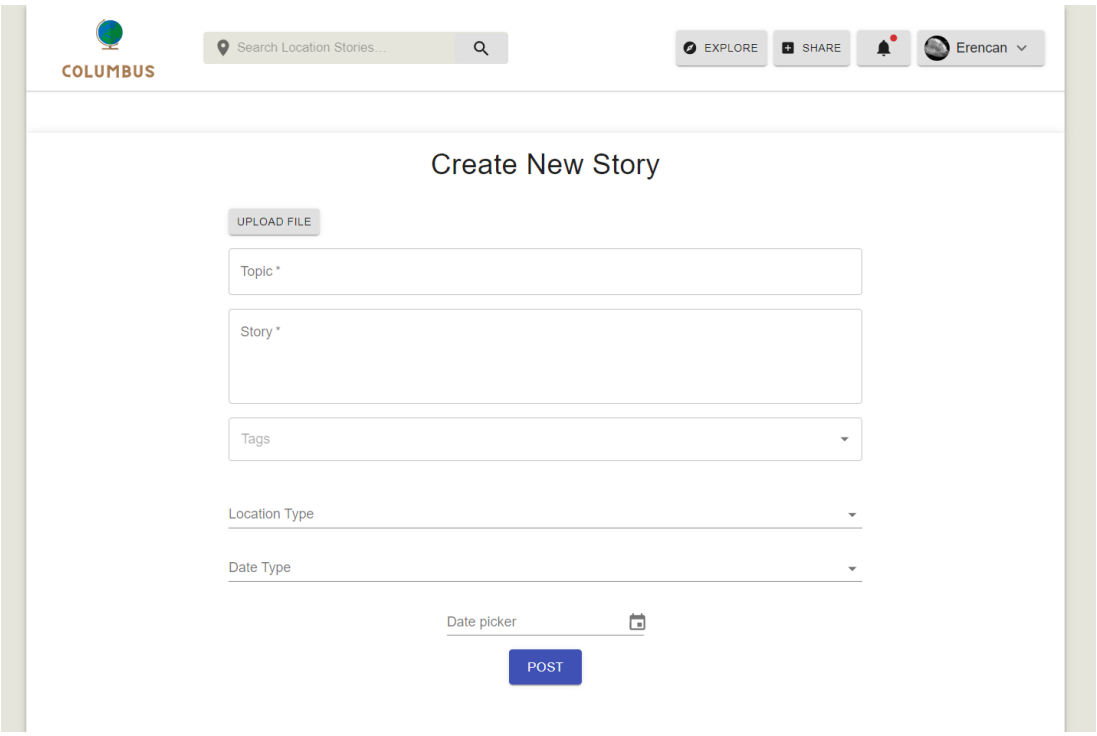
Successful Report Message:



SHARE STORY

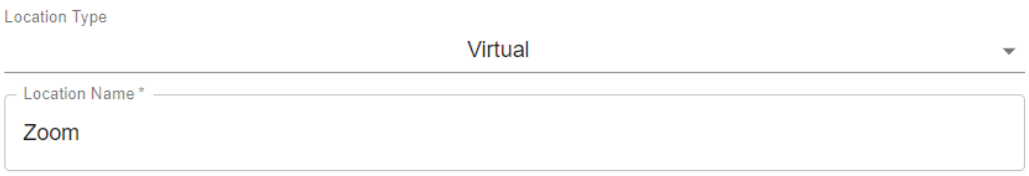
If you want to share a story, you should click the share button on the header. When you click that button, story creation page will open.

Share Page:



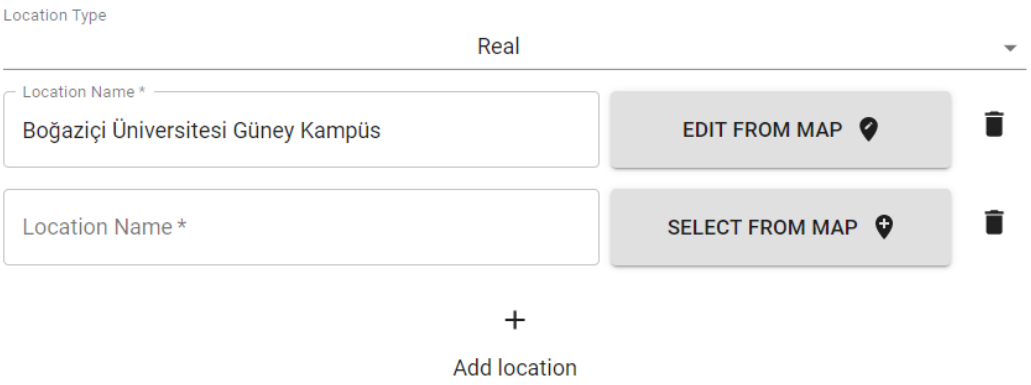
In our application, a user can create stories with selecting wide range of types of locations and dates. You can upload a photo with using “upload file” button. Topic, story, location and date fields are required. You can also add tags to your story but it is optional. You can choose two types of location: real and virtual. We added virtual locations considering that people have started to use social media and virtual meeting applications much more frequently lately. You can add “zoom” or “instagram” to your stories. You can add at most one virtual location to your stories.

Virtual location:



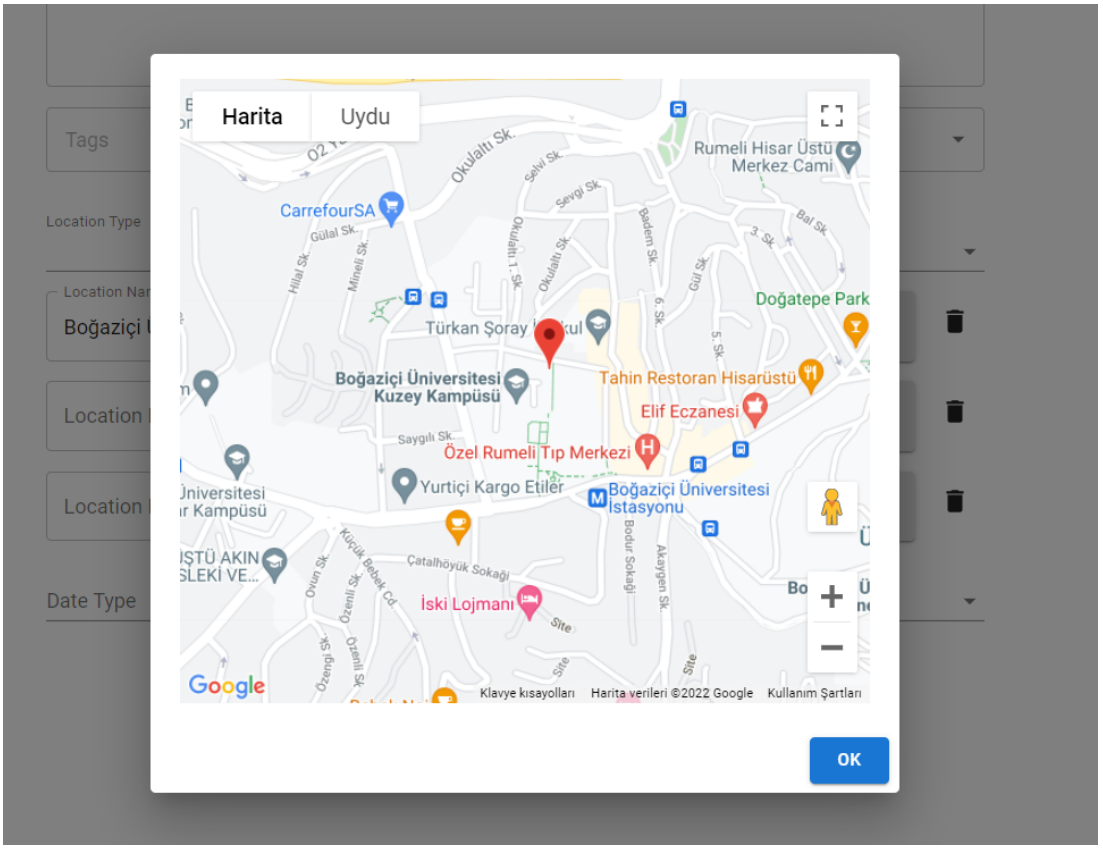
You can add at most three real locations to your stories. You should also pick a location from map to your real locations.

Real location:



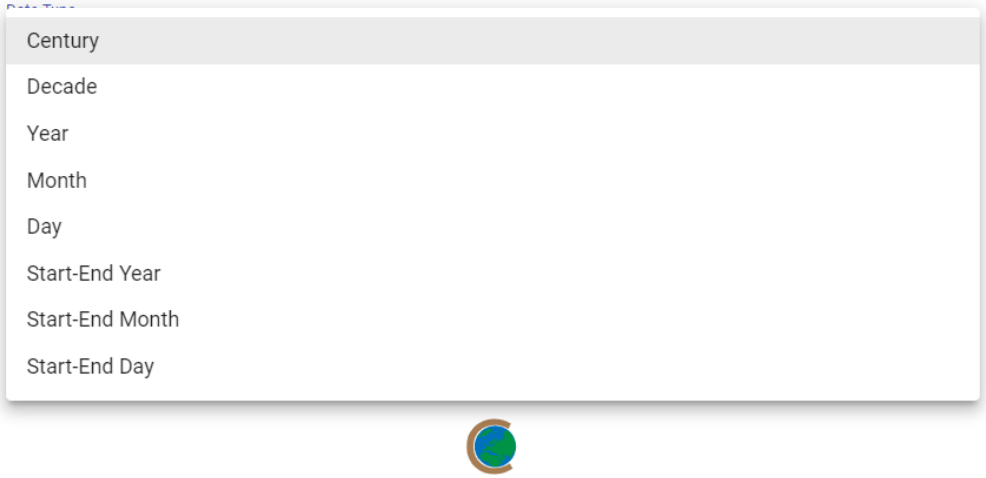
If you click the “Select from map” button, you can select a location from the map. You can also edit this location after adding it. If you click the garbage icon, the relevant location will be deleted from your story.

Location map:

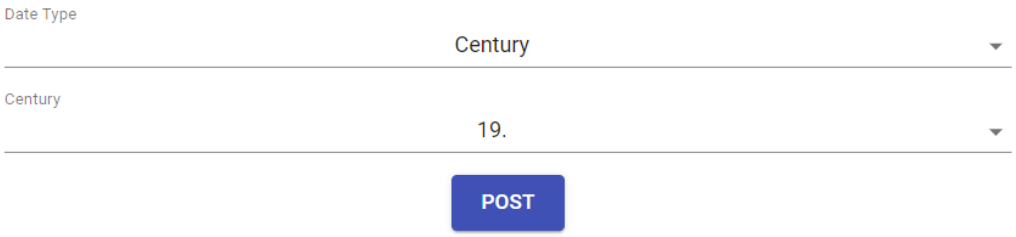


You can select one type of date from different kinds of dates. We offer century, decade, year specific, month specific and day specific dates. You can also select two dates from year specific, month specific and day specific dates.

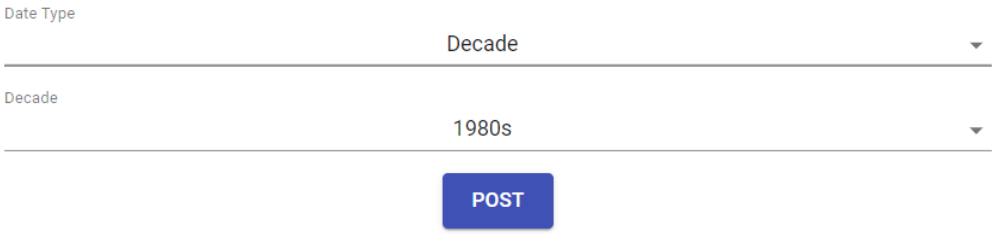
Date type selection:



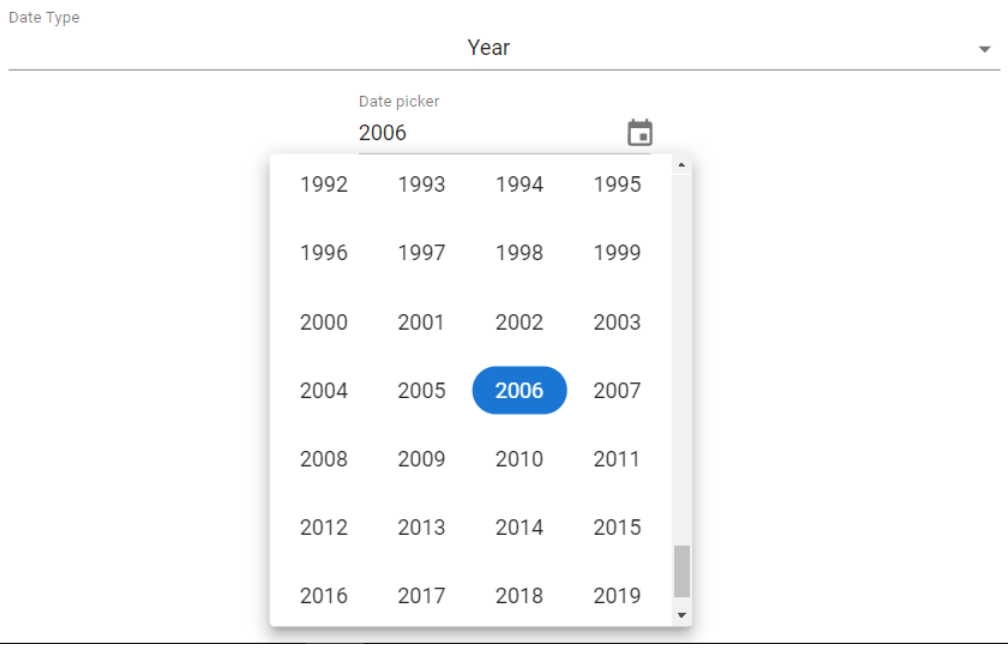
Century selection:



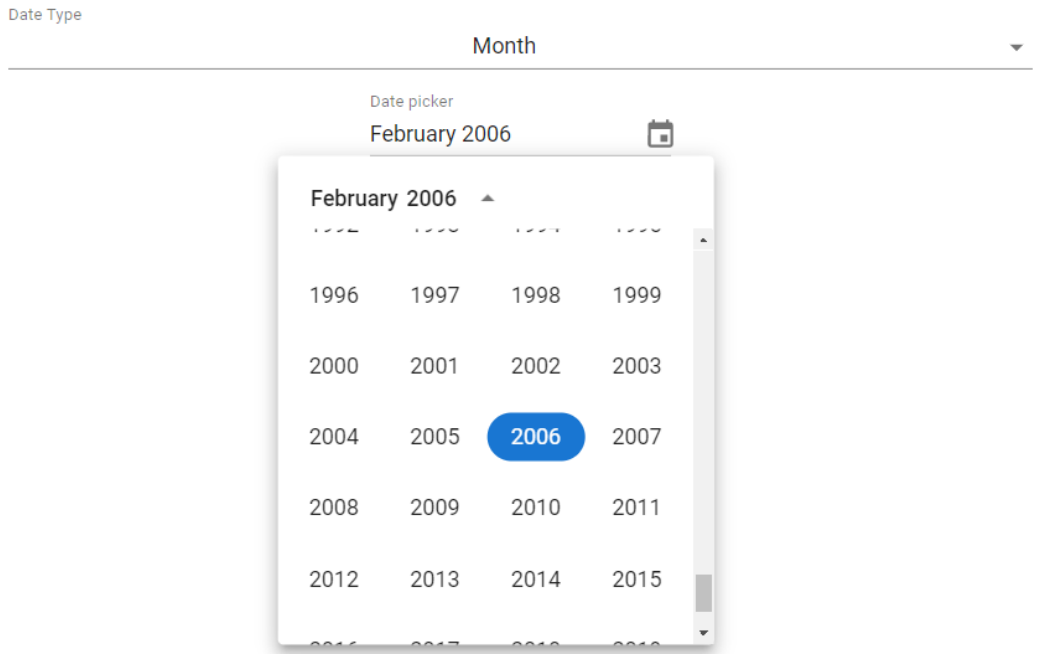
Decade selection:



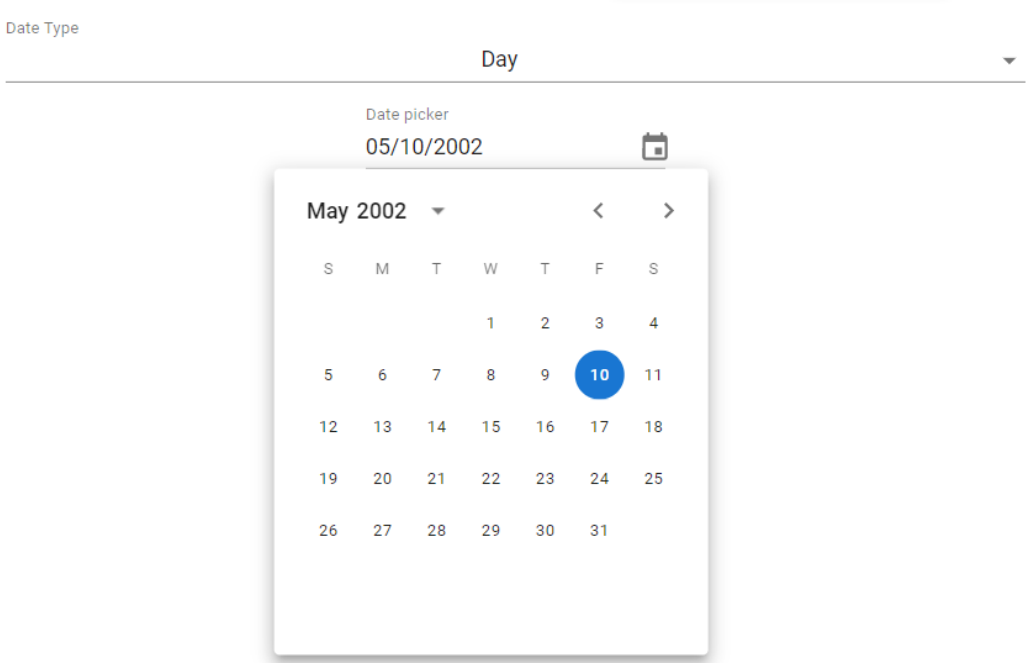
Year specific:



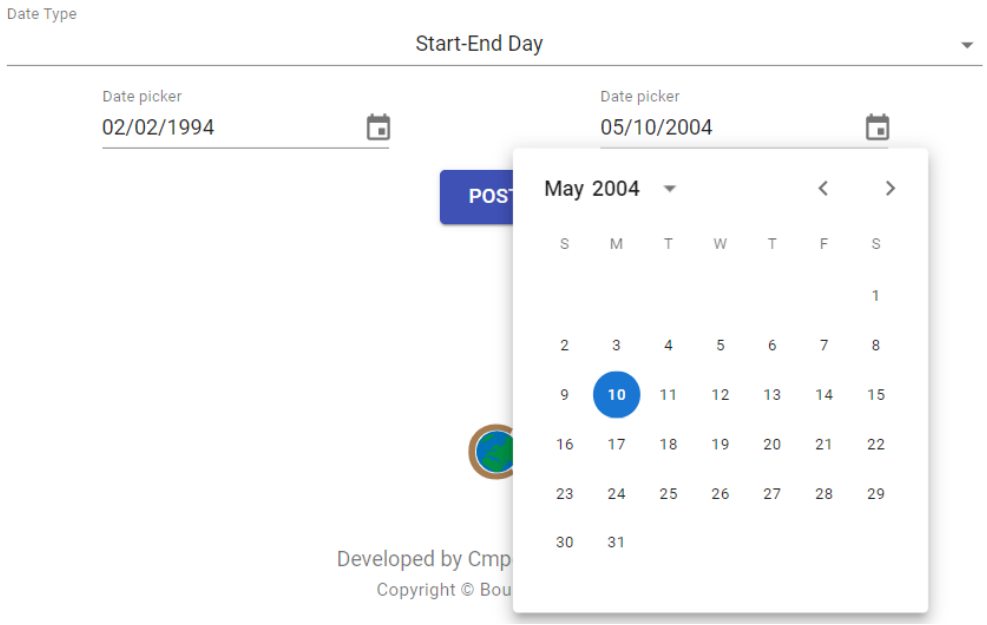
Month specific:



Day specific:



Start-end dates:



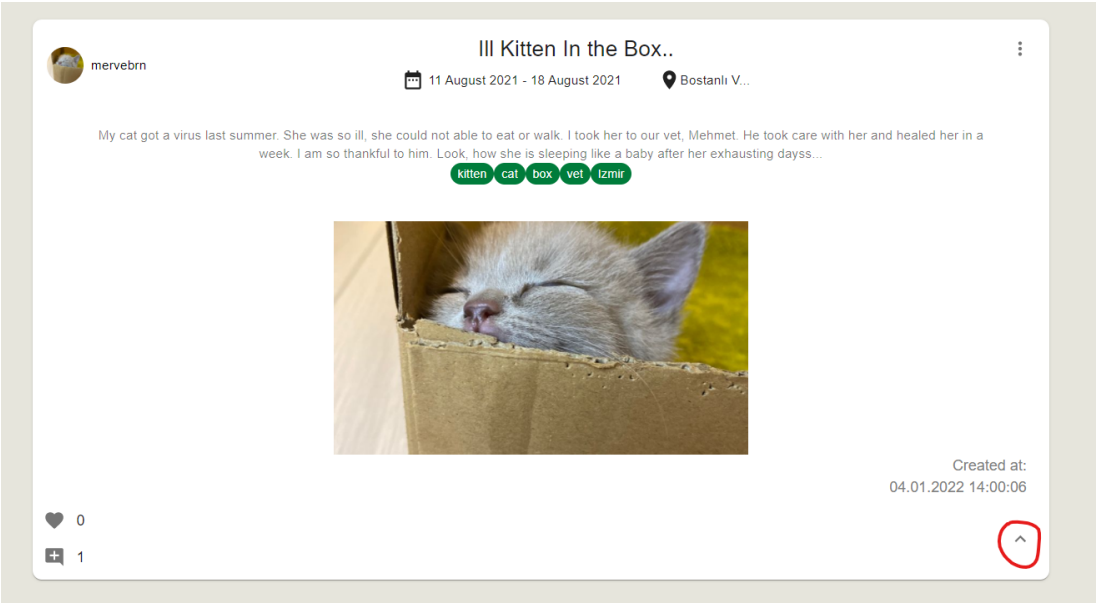
After selecting the date and location, you can share your story by clicking the "Post" button.

POST and COMMENTS

While you are navigating through pages, you will frequently encounter posts. Post is the main component of our application. You can see posts on your home page,

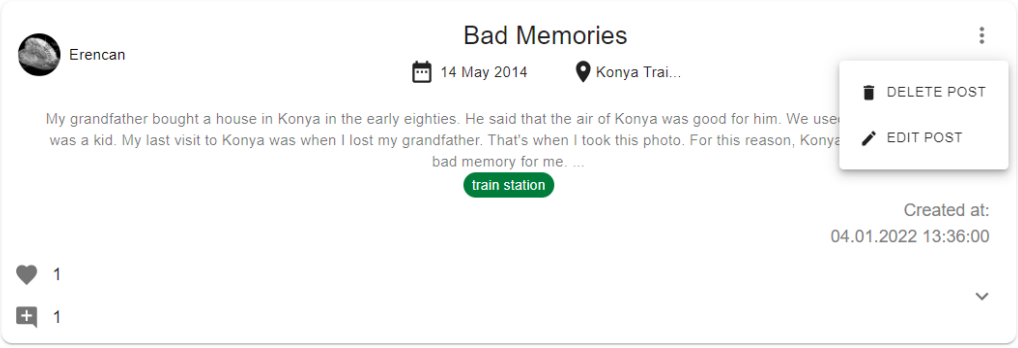
explore page, search page and others profiles. If you click the arrow button on the lower-right of the post component, you can expand the story. You can view images and you can see the rest of the story, if any.

Expand story:



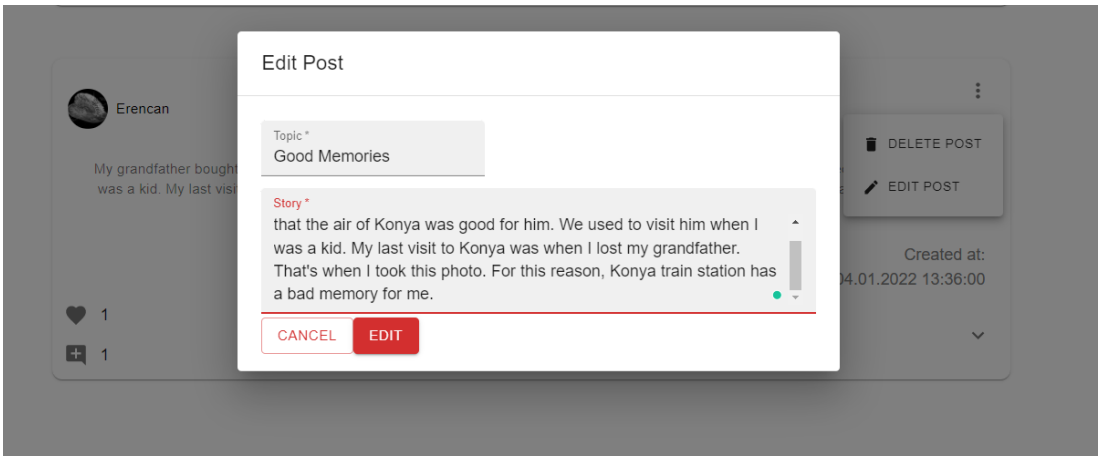
If a post belongs to you, you can edit or delete this post by using the three dots icon on the upper-right side of the post.

Settings of users own post:



If you want to edit your post, you can click the edit post button. For now, we only offer editing for text and title parts. Date and location should be carefully selected in the creation step.

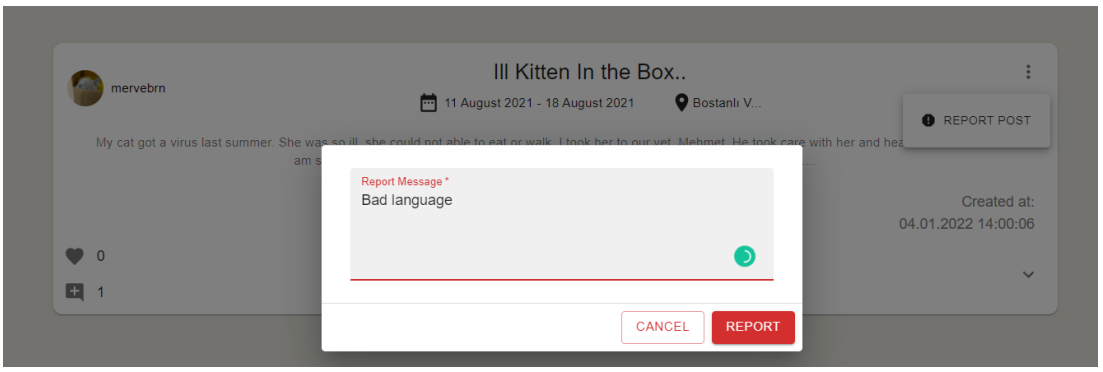
Edit post:



If you want to delete a post, you can click the delete post button. If this deletion is successful, you will see a message in the lower-left of your screen. Attention! your post will be immediately deleted.

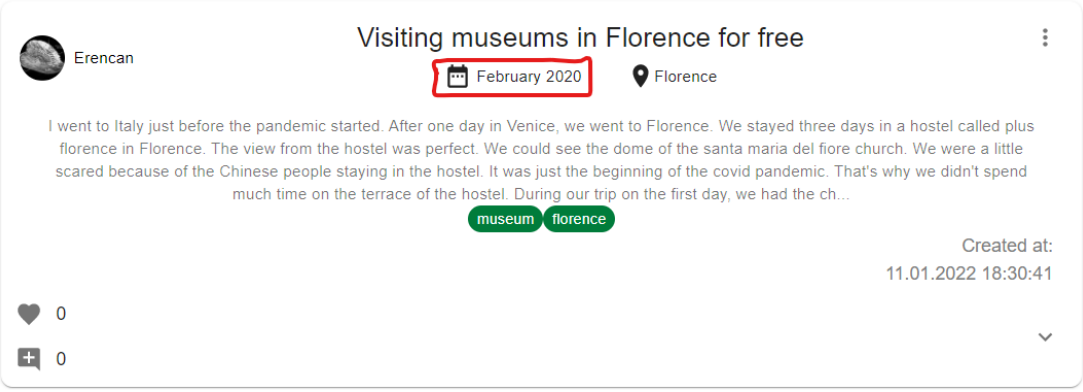
If you are viewing someone else's posts, press the three dots icon and the report post option will appear. You can report this post by writing your reason for reporting.

Post report:



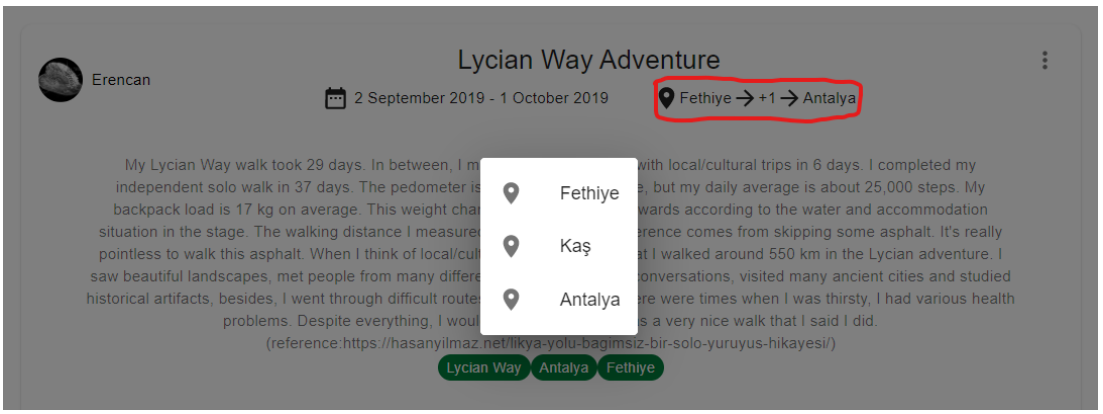
You can go to the date search page with clicking date part on the post component. This page will show you all the stories filtered with that story's dates.

Date of the post:



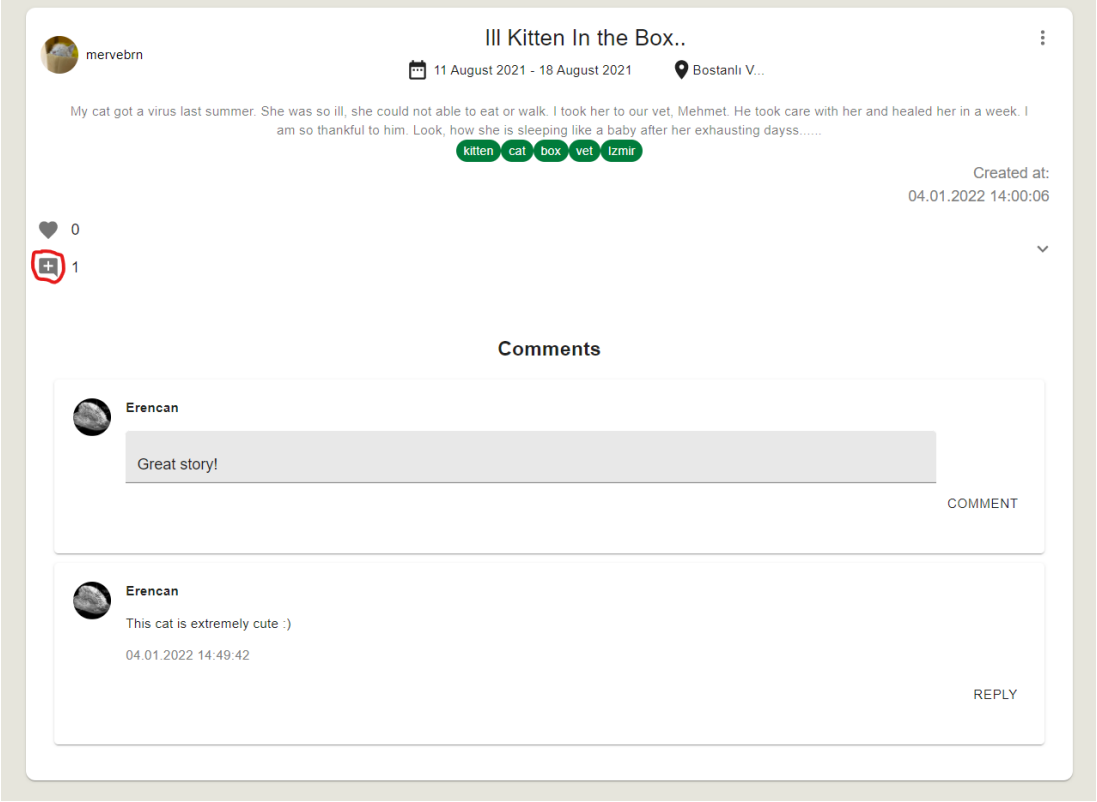
There can be multiple locations on a post. You can see all this locations with clicking the location icon under the title of a story. When you click a location, it will navigate you to the search page with location filter. Thus you can see stories on nearby locations.

Location of the post:



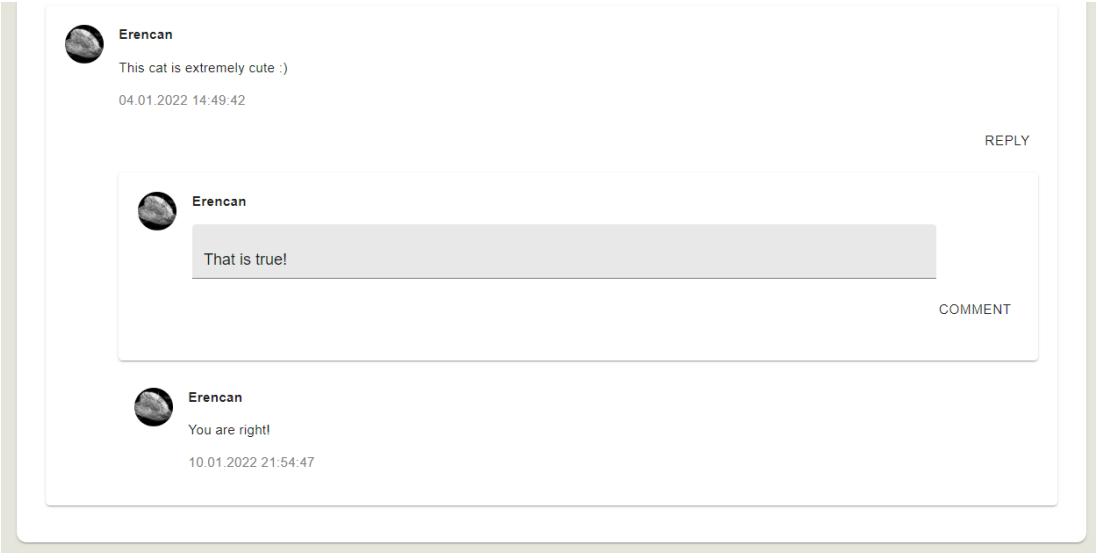
You can make comments on posts if you are a logged in user. You can basically click the plus button on the lower-left of the post component to view other comments. When you click this button, you can also see the comment field where you can write your comment. If you click the “comment” button, your comment will be added to that story.

Commenting:



We have a comment reply feature in our app. You can reply to a comment with a comment. You should click the “reply” button on a comment, then you can click the “comment” button to add your reply.

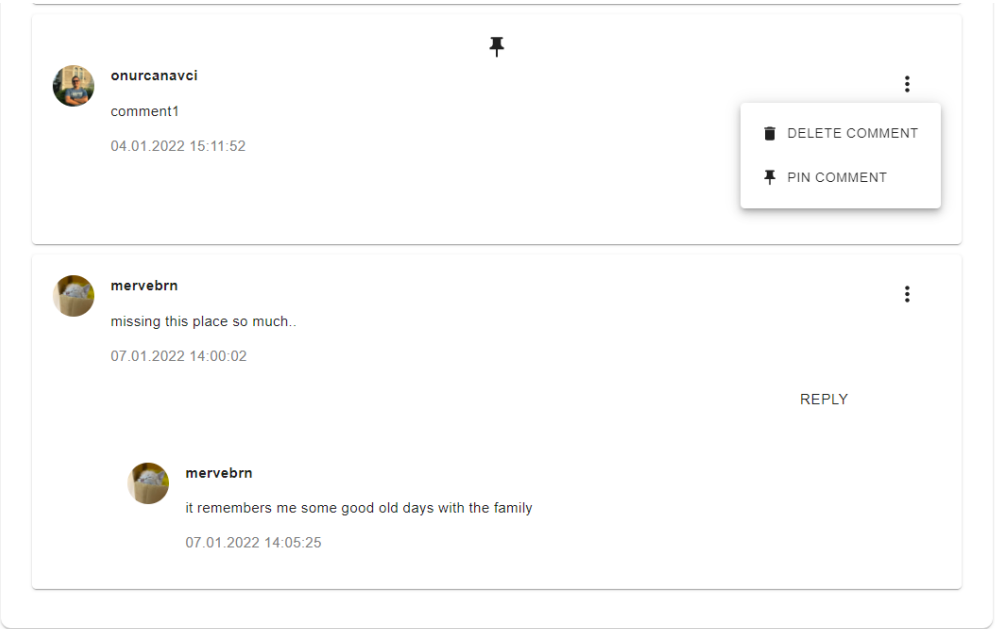
Comment reply:

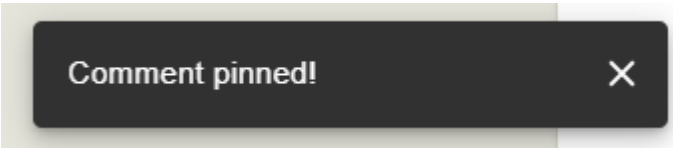


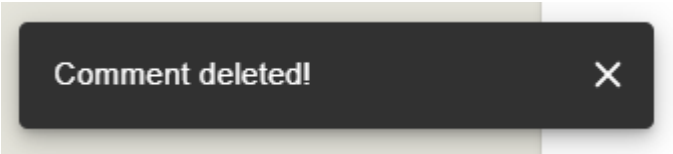
While viewing your own posts, you may notice that there is a three dots icon above the comments. You can reach delete and pin-unpin comment features from this icon. You can pin and unpin comments with using “pin comment” button. Pinned comments will appear in the first places on the comments section with a pin icon. If

you don't like a comment, you can delete it with using "delete comment" button. Attention! comment will be immediately deleted. When you delete or pin a comment, if there are not any problem, you can see a message on lower-left part of the screen. If there is a problem, this message box will state that.

Comments on users own posts:



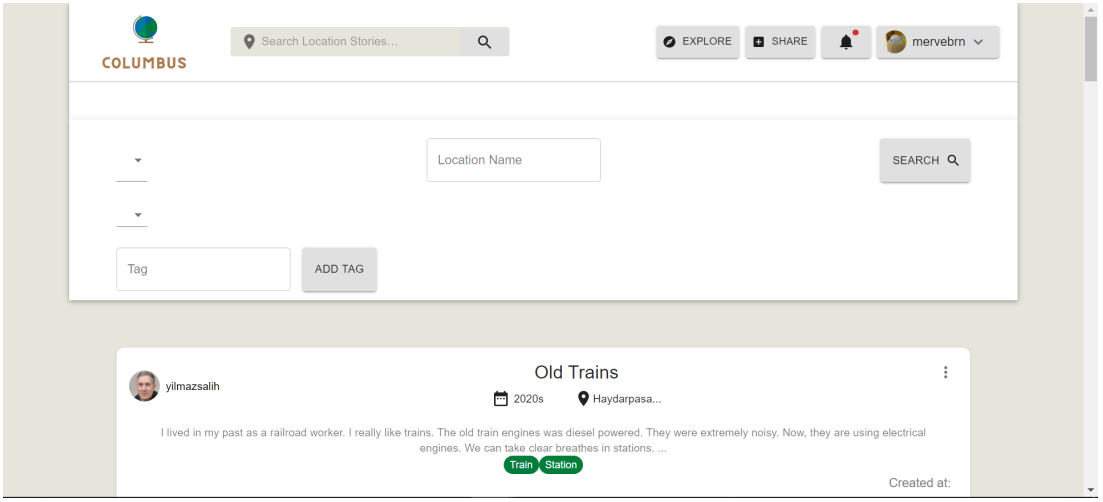




SEARCH and FILTER

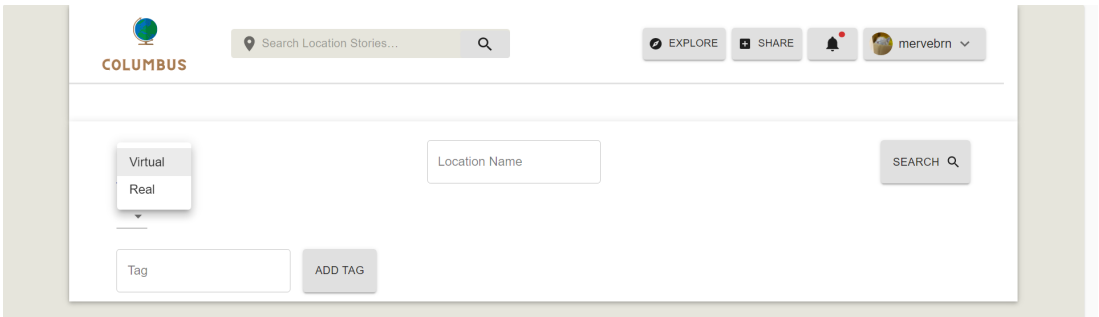
You can access Search page by clicking the Search icon in the header.

Search Page:



Location Search

There are two types of location: Real and Virtual.



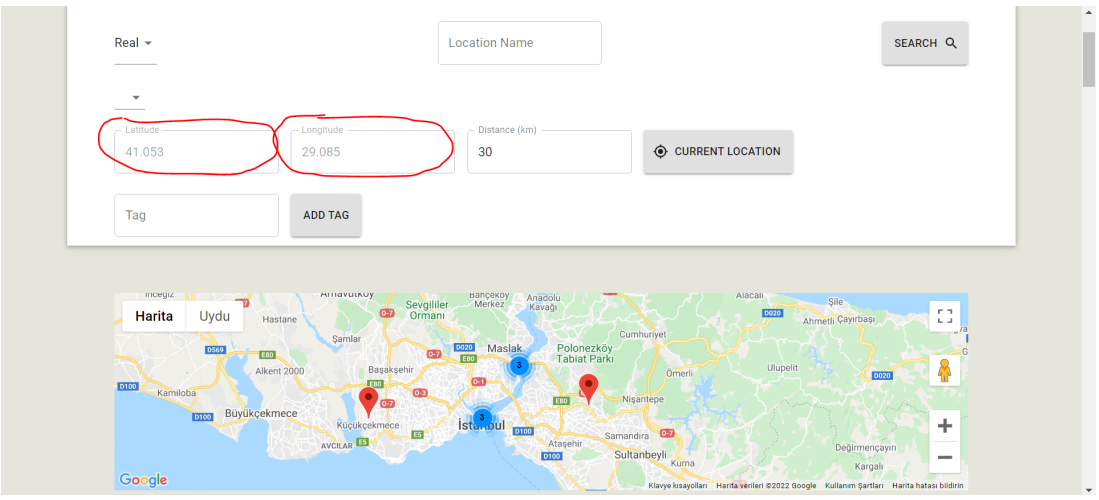
***Real Location Search*:** When you select the Real Location option, you can view the map.

- Search based on pinned location
- Search based on current location
- Search based on clusters

Search based on pinned location

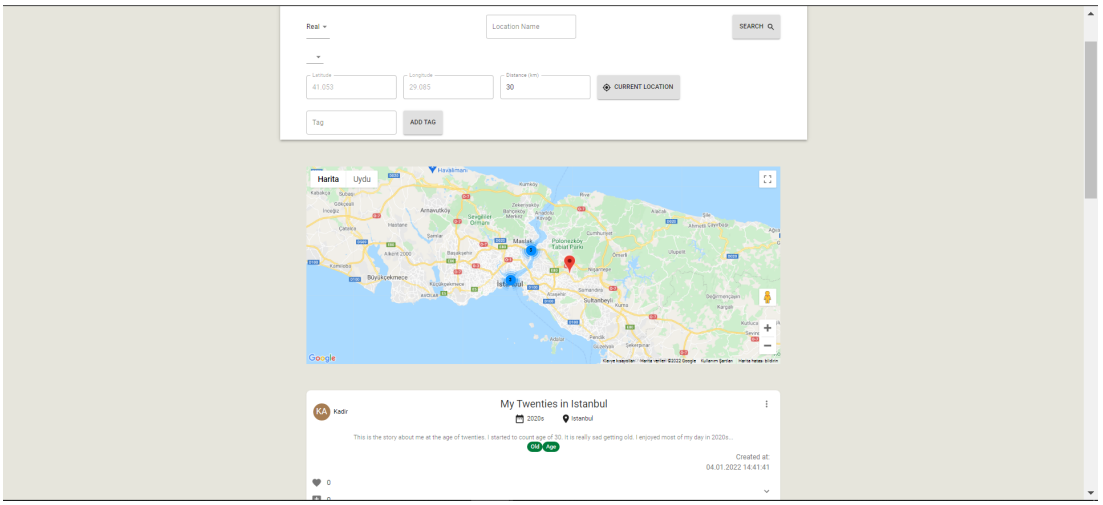
When you click somewhere in the map, its latitude and longitude are displayed in Latitude and Longitude fields.

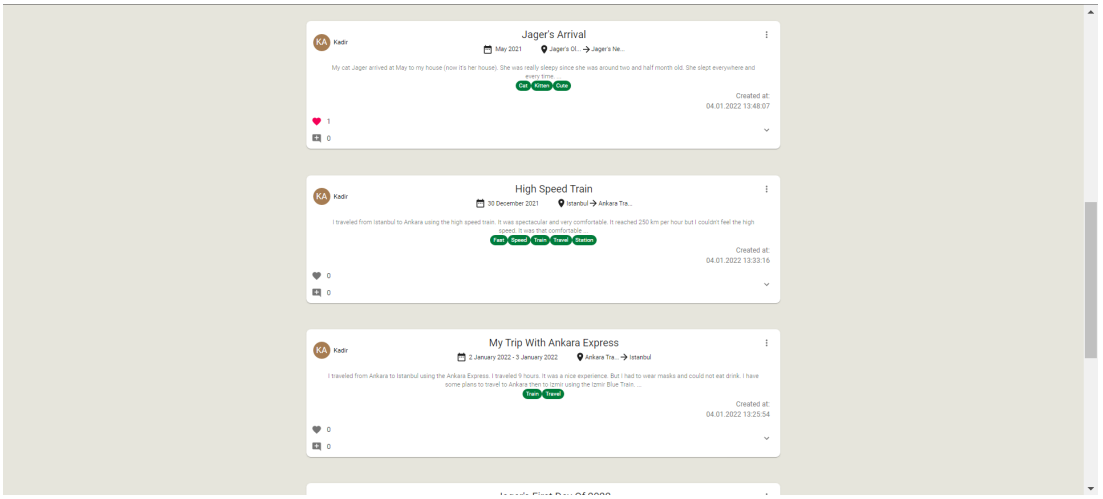
Pin a Location on Map:



After selecting coordinates, when you click the Search button, you can see stories shared in the selected area of given distance (30 km by default). You can change the distance in Distance field next to Longitude.

View Stories Around the Selected Area:





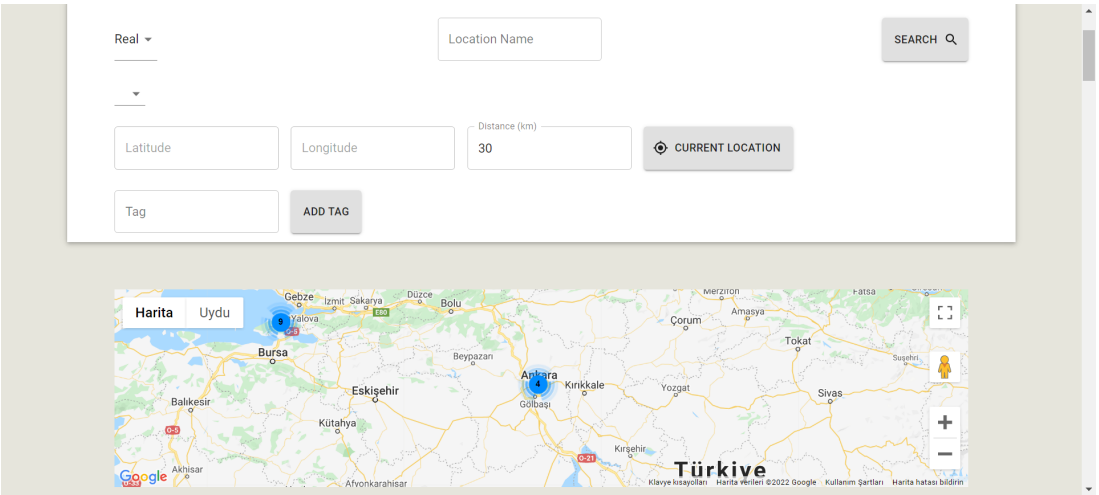
Search based on current location

You may want to search location stories around you without selecting your location on the map. You may use **Current Location** button to get the latitude and longitude of your location, then view stories around you by clicking the Search button.

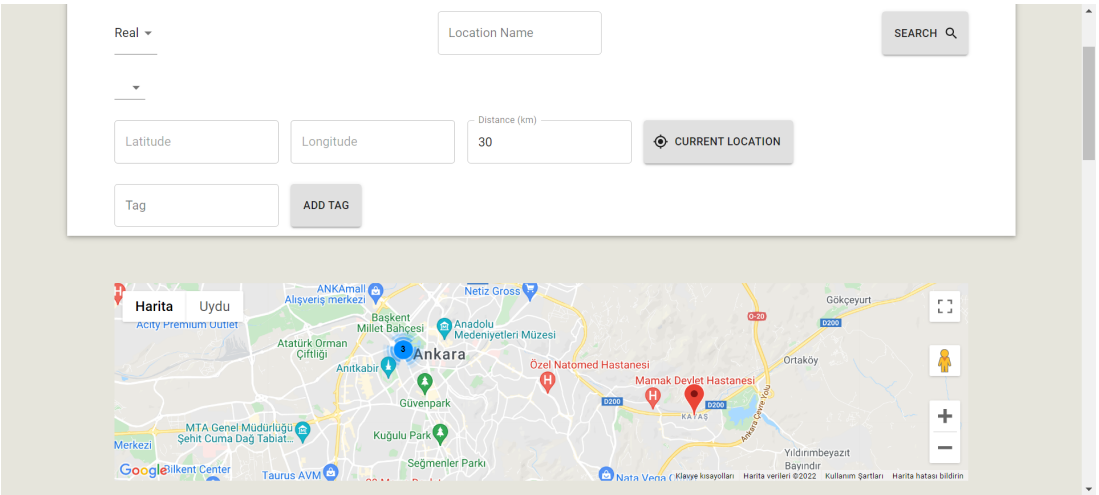
Search based on clusters

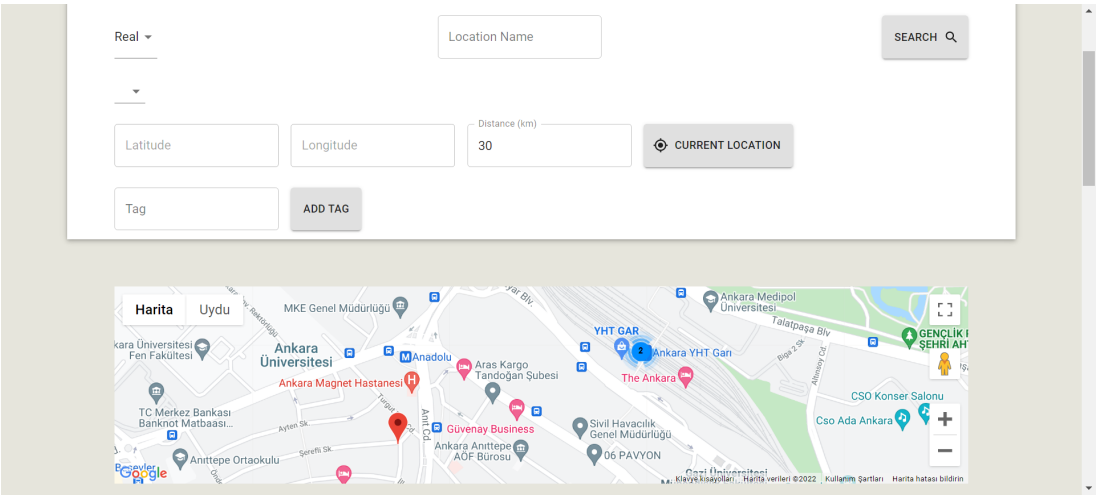
The stories shared around the same area are displayed as clusters in the map. You can see the number of stories shared around a location. When you click the number, you can see the locations of the posts more specifically.

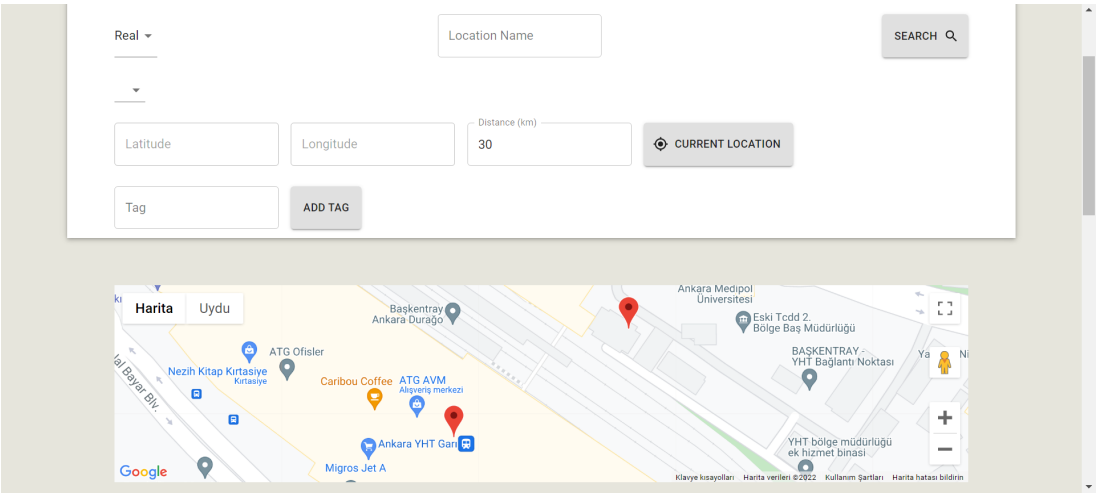
View Clustered Stories:
(There are 4 stories around Ankara)



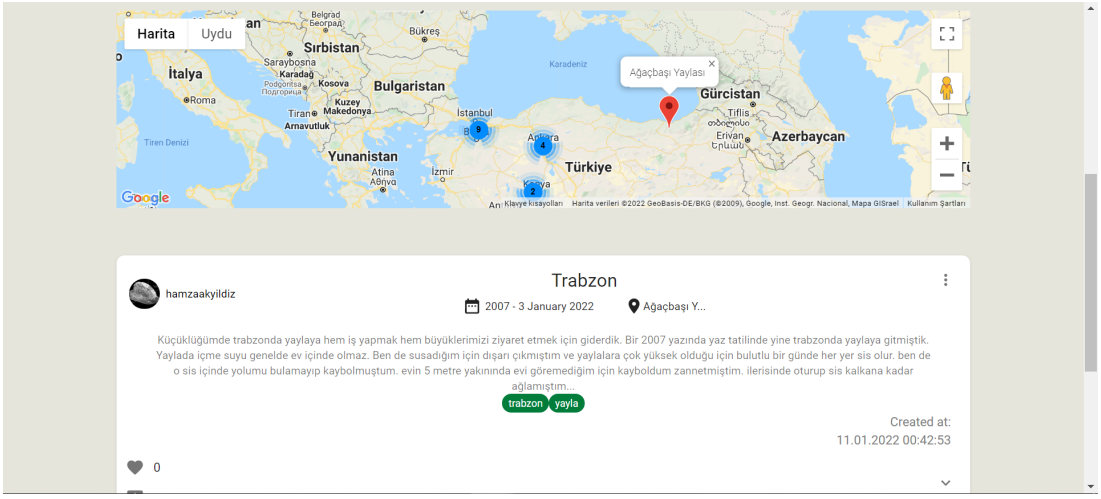
View Clusters in Detail:







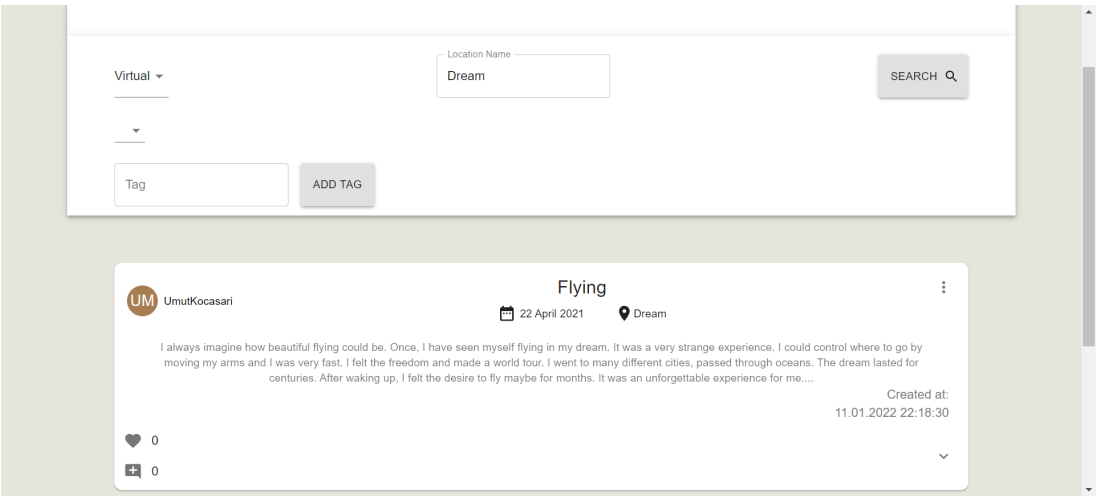
You can view the specific story when you click on the pin.



Virtual Location Search: You can select the Virtual option to search for locations that were not pinned on the map.

If you want to search for a specific keyword, you can enter it in the *Location Name* field then click Search button.

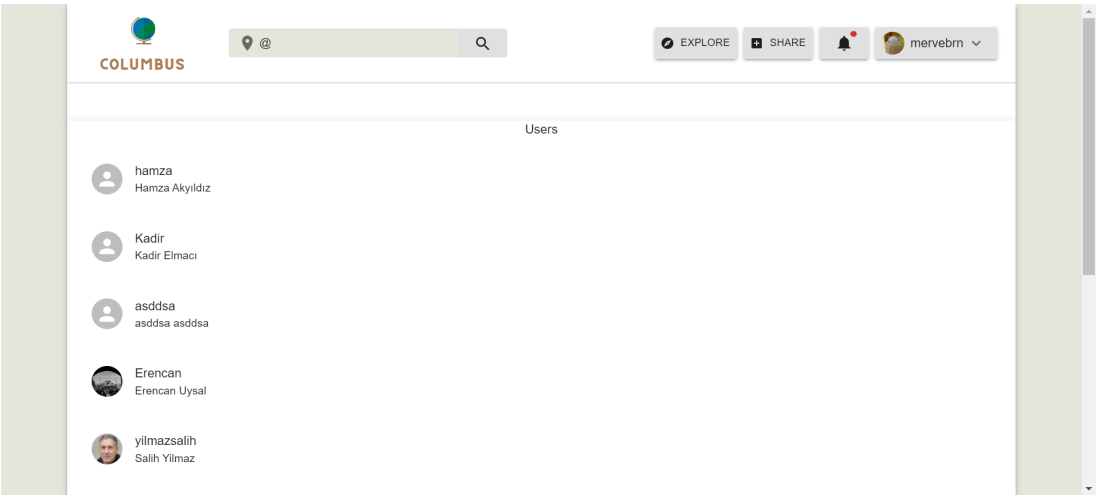
Virtual Location Search:



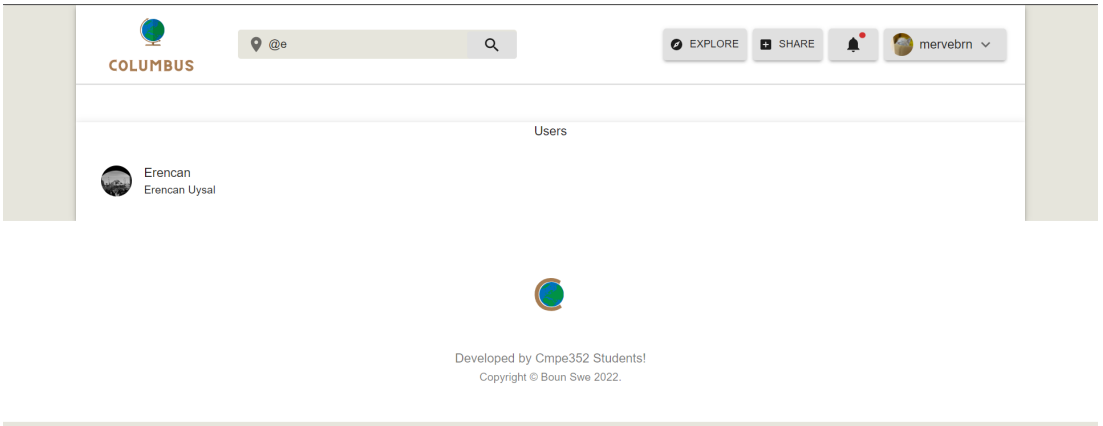
User Search

If you want to search for a specific username, you should start your search with the '@' character.

Example User Search 1:



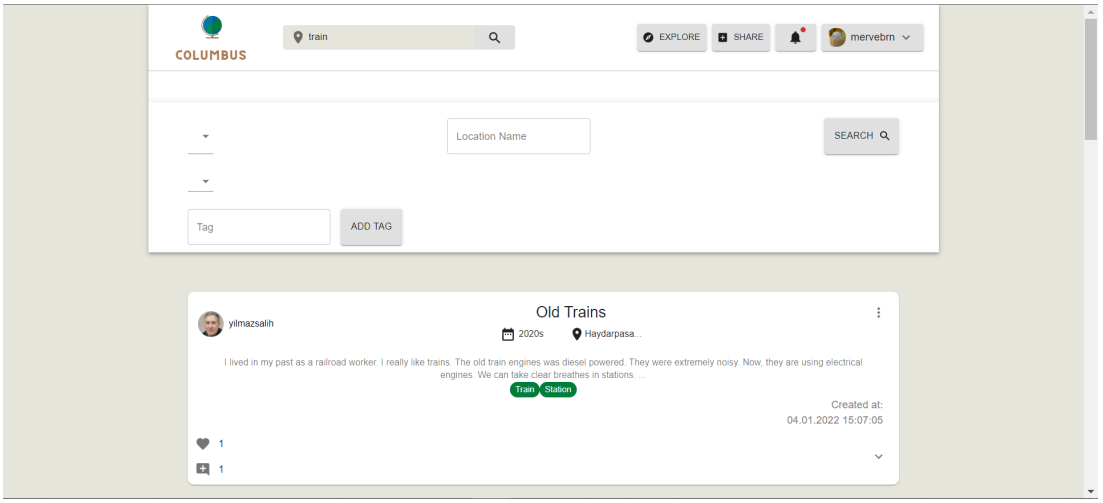
Example User Search 2:

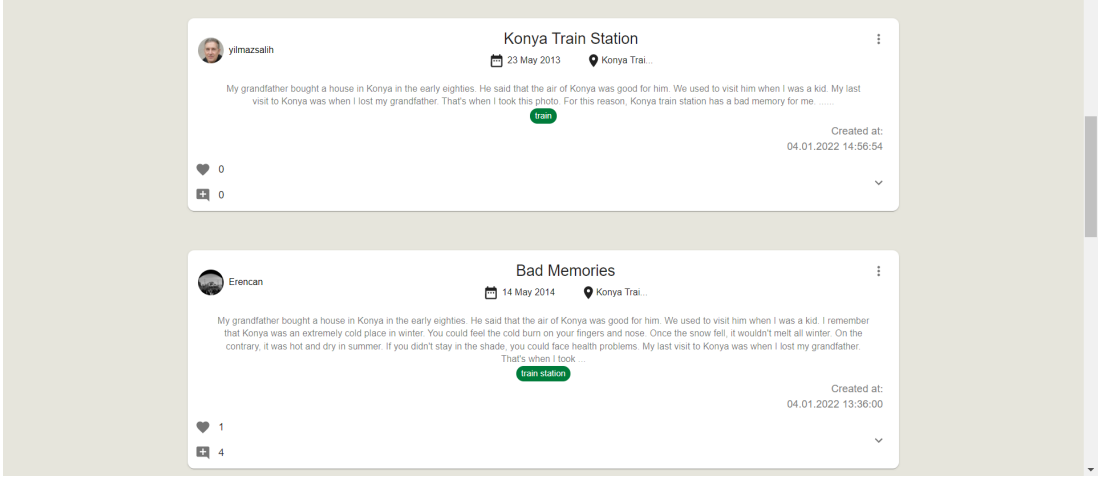


Keyword Search

If you want to search for posts that contain a specific keyword, you should enter the keyword into the Search bar, then click the Search icon.

Example Keyword Search:



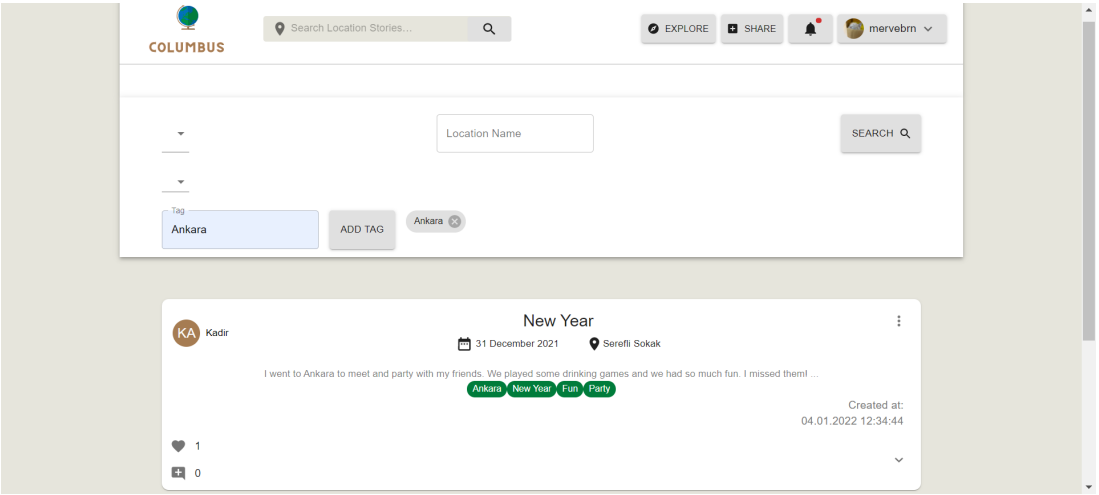


Filtering

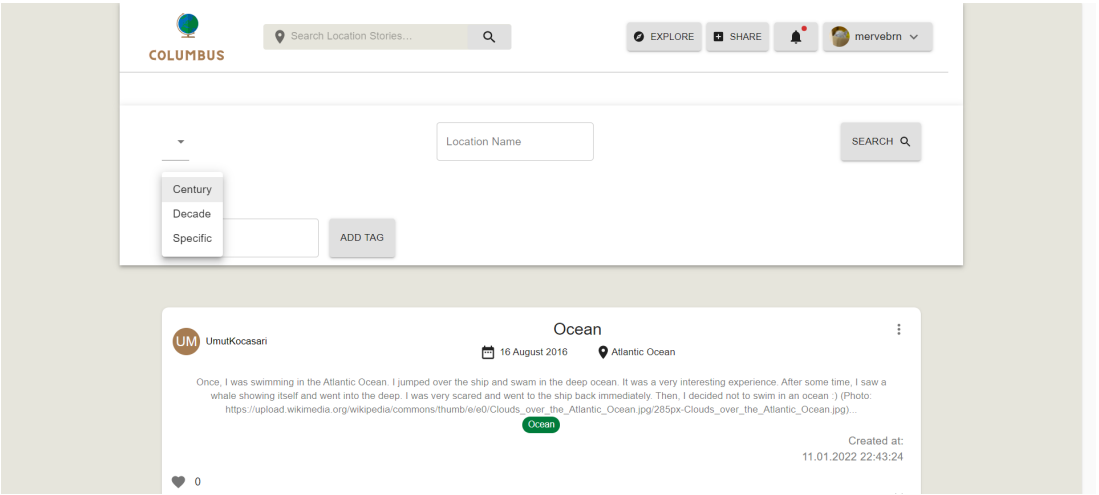
You can filter search results. There are several filtering options.

- *Filter by Tag*
- *Filter by Time*

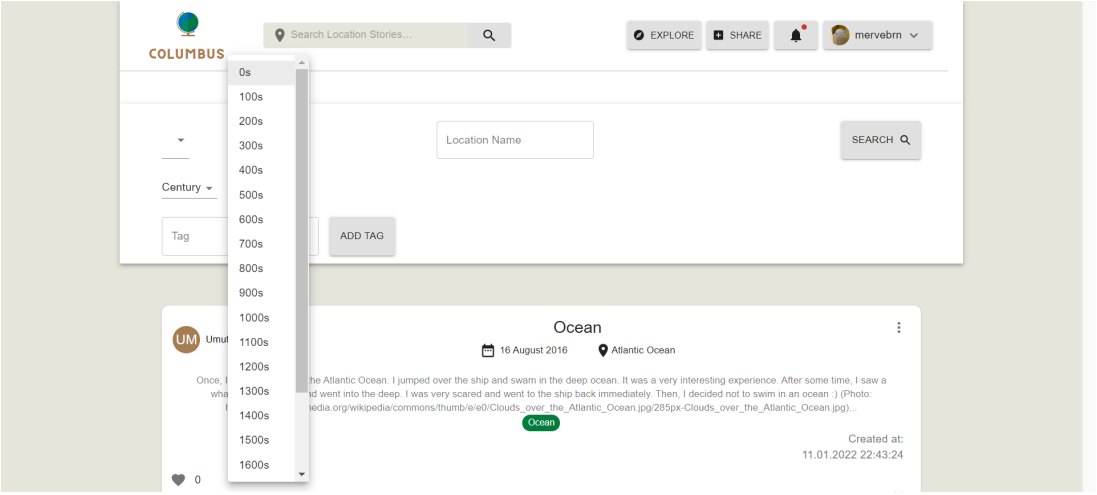
Filter by Tag: If you want to retrieve only the posts that have a specific tag, you can enter the tag into the Tag field and click ADD TAG button. If you press the Search button, only the posts with the entered tag will be displayed on the page.
Example Tag Filtering:



Filter by Time: If you want to retrieve only the posts with a specific time type (Century, Decade, Specific), you can select the type you want.
Time Type Filtering Options:

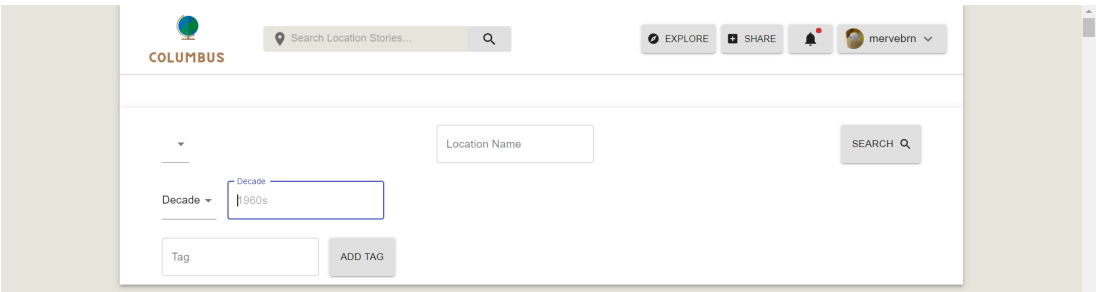


Century Filtering:



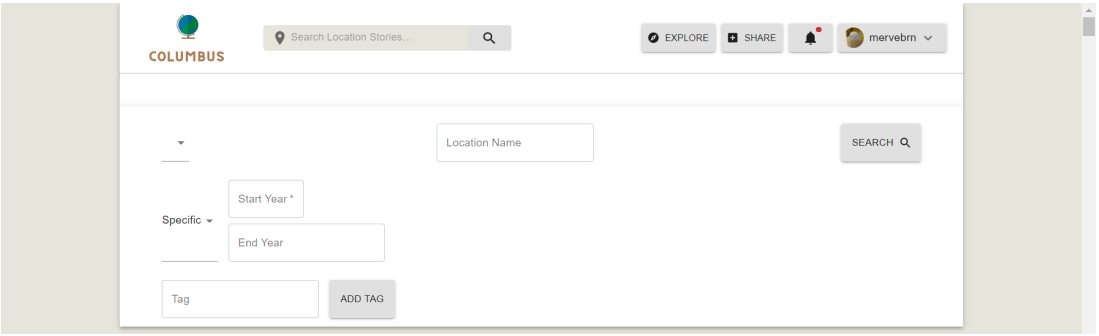
In decade filtering, you should add ‘s’ after decade.

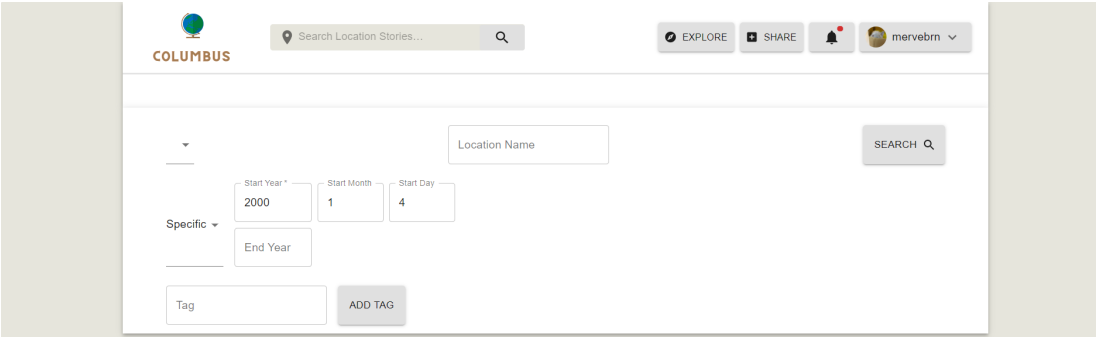
Decade Filtering:



In Specific Time Filtering, you must enter the start date, and the end date is optional. When you select year, month and day fields also will be displayed. They are optional.

Specific Time Filtering:





MOBILE

Guest

Before you register to Columbus, you are considered as '*Guest*' and able to perform a limited set of actions on the Web Site. However, you can not commit an action in Mobile Application as a guest. Hence, you should register and login by using credentials to use the mobile app. You can follow instructions in the next section (Login and Register) to register and login via mobile application.

Register:

If you do not have an account for Columbus or you want to make a fresh start to Columbus, you can register by using "Register Page". To reach this page, you can click highlighted text that says "Sign Up" on the landing page. First of all, you should fill in the required fields: username, email and password in such a way as to meet the specified conditions:

- Username must be unique.
- Passwords must be at least 8 characters.
- Email must be valid

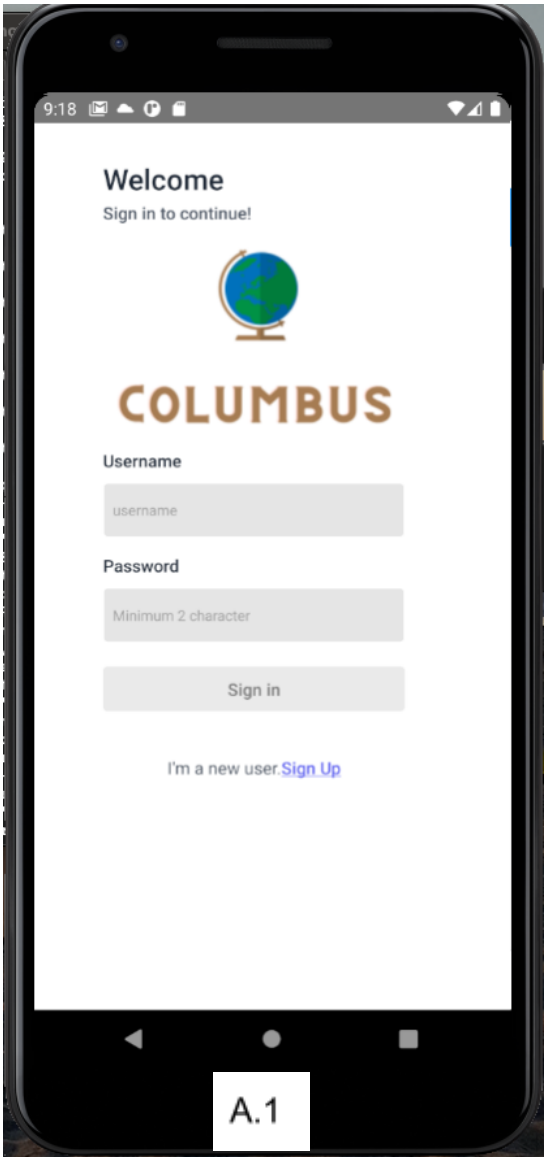
If these restrictions are not met, the register button will not be active, or you will be notified with a pop up. Then, you can try with other combinations of username and email.
In case of success, you need to activate your account. We will send you an email for activation. You can complete this process by clicking on the link in the email.

Login:

If you already have an account for Columbus, you can directly login with your username and password. To perform login, at first, you should type your username and password in corresponding fields correctly. After that, clicking the "Sign In" button will redirect you to "Home Page" (to be mentioned in next pages).

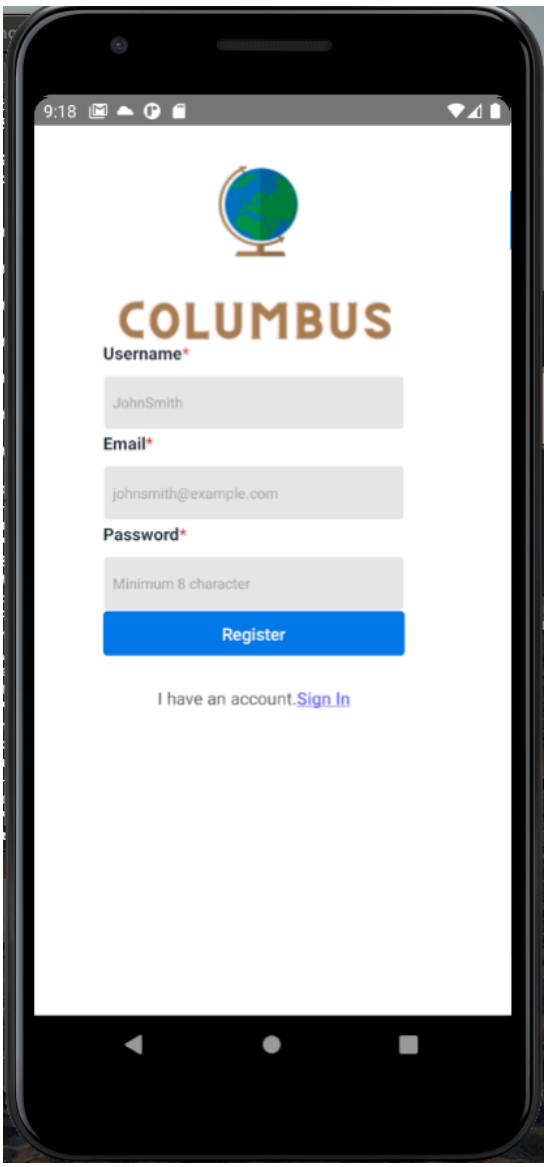
After this process, you will not need to enter this information every time you want to access the application. We will keep your credentials in a secure way.

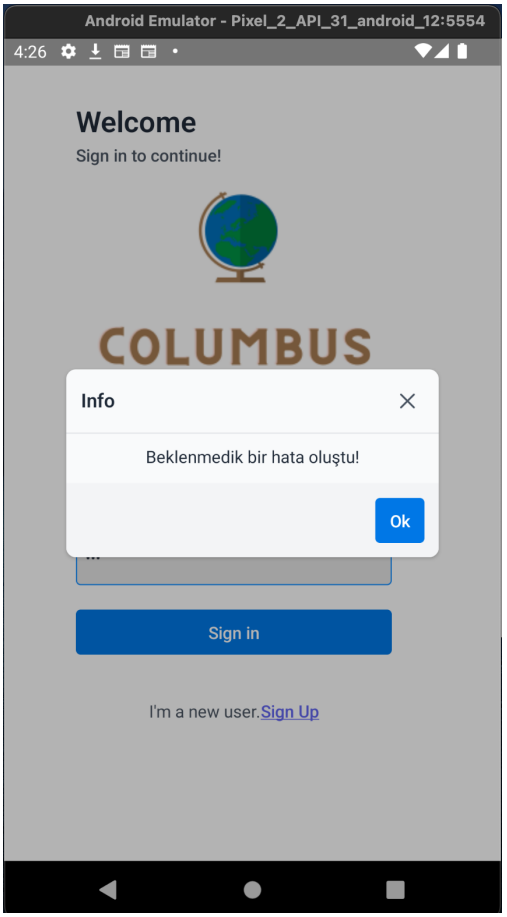
If you have security and privacy concerns about the application, you can check this website also: https://support.google.com/android/?hl=en#topic=7313011


9:18
Welcome
Sign in to continue!
COLUMBUS
Username
username
Password
Minimum 2 character
Sign in
I'm a new user.Sign Up

A.1

A.1.1


9:18
COLUMBUS
Username*
JohnSmith
Email*
johnsmith@example.com
Password*
Minimum 8 character
Register
I have an account.Sign In


Android Emulator - Pixel_2_API_31_android_12:5554
4:26
Welcome
Sign in to continue!
COLUMBUS
Info
Beklenmedik bir hata oluştu!
Ok
Sign in
I'm a new user.Sign Up

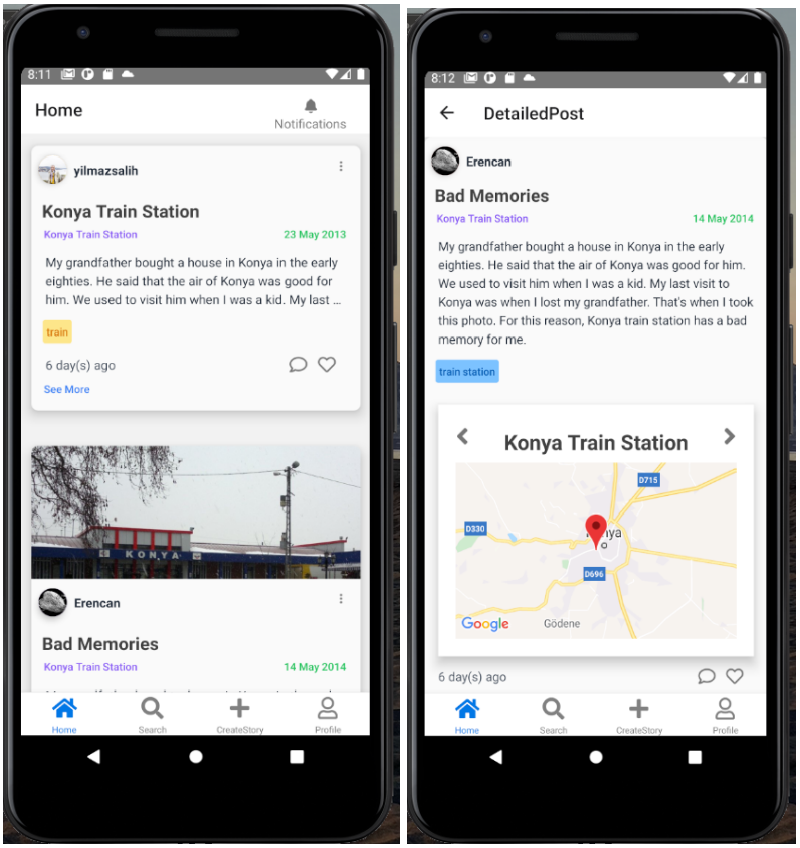
Home Page and Post Details:

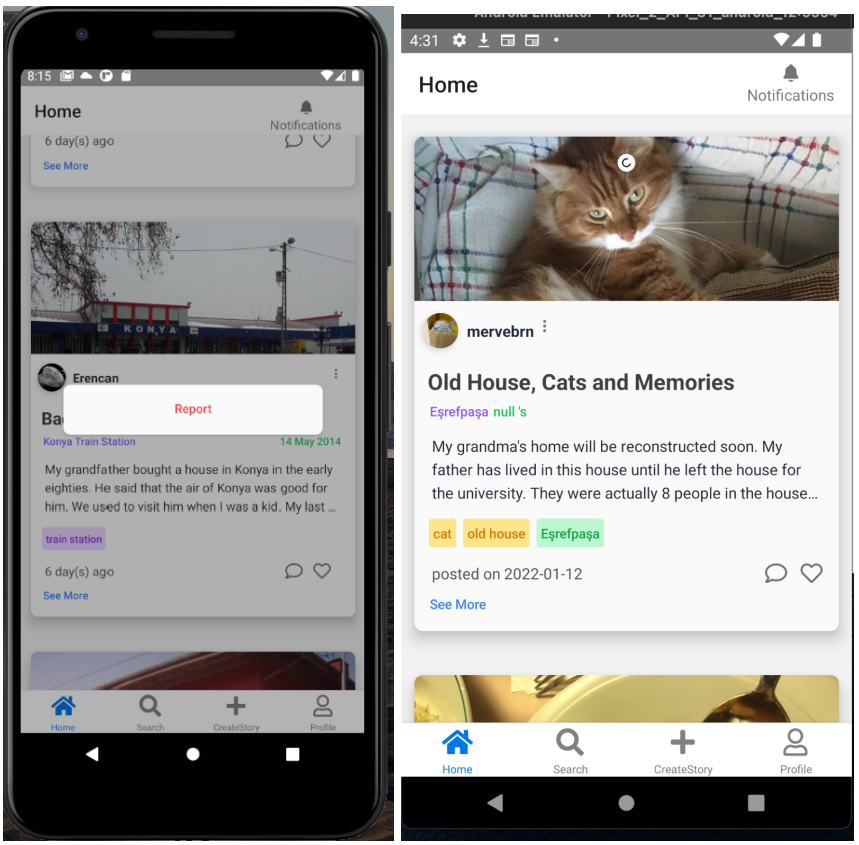
When you login the application, then the home page will be open you can view the stories of users you follow. In each card, you can see details of each story: Where this story takes place, username and profile picture of who shared this story, first three lines of the story and when the story was shared (relative to you).

By clicking highlighted text saying “See More”, you can see the features of the stories in detail on the page that opens. In addition to information on a card, you can see the whole story and location or locations for this story on the map below.

By sliding the screen down, you can refresh your feed.

For a story that you think is harmful for the community, you can share your problem with us by clicking the three dots icon in the upper right corner of the card.





You can interact with people by commenting and liking.

When you click on the bubble icon at the bottom right of the card, you can view the comments, make comments for the story, reply to the comments as comments, and interact with the comments by pressing and holding them for a while for the story you clicked.

To make a comment for a story, you should write down your comment in the text section at the bottom of the screen. After that, clicking the “Comment” button with blue color on the right will send your comment.

During the interaction phase:

- You can edit or delete your own comments
- You can report others’ comments
- You can pin comments if the story holding the comments belongs to you

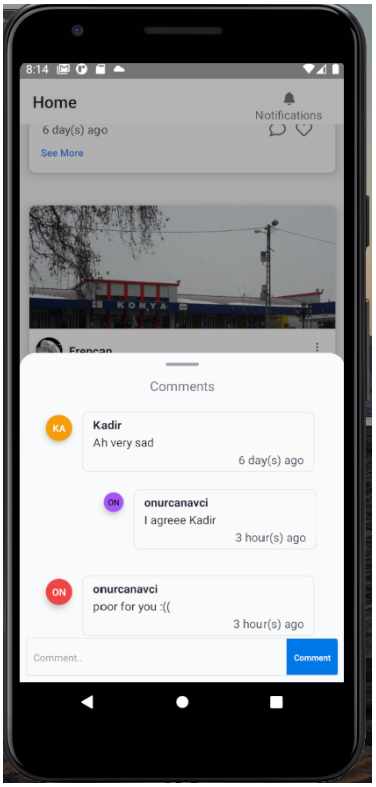
To edit the comment, you can press the edit button in the popup that opens, make changes in the text section that opens, and press the confirmation button.

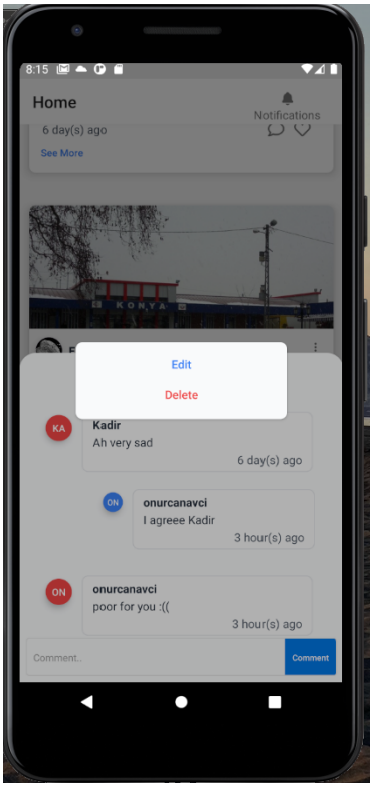
To delete the comment, you can confirm the statement in the warning section that will open by pressing the delete button in the popup.

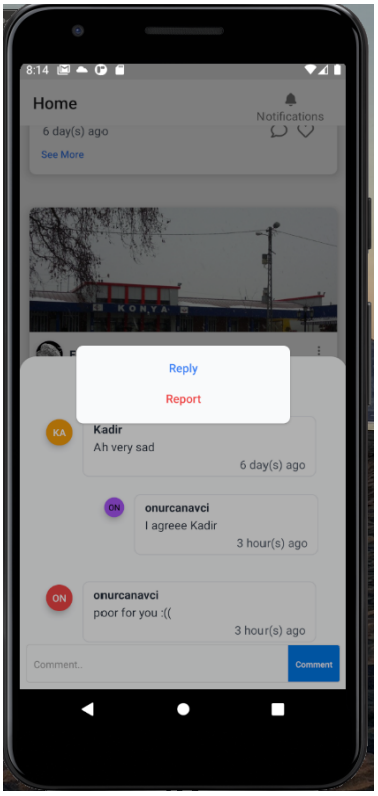
To pin or unpin the comment, just press the pin or unpin button, relatively, in the popup that opens.

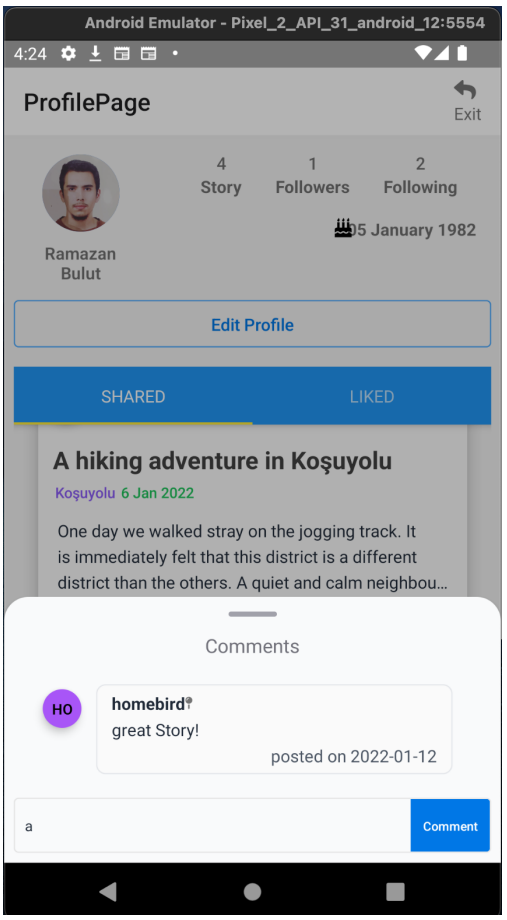
To report the comment, you can press the report button in the popup that opens, write down your problem with the comment in the text section that opens, and press the confirmation button.

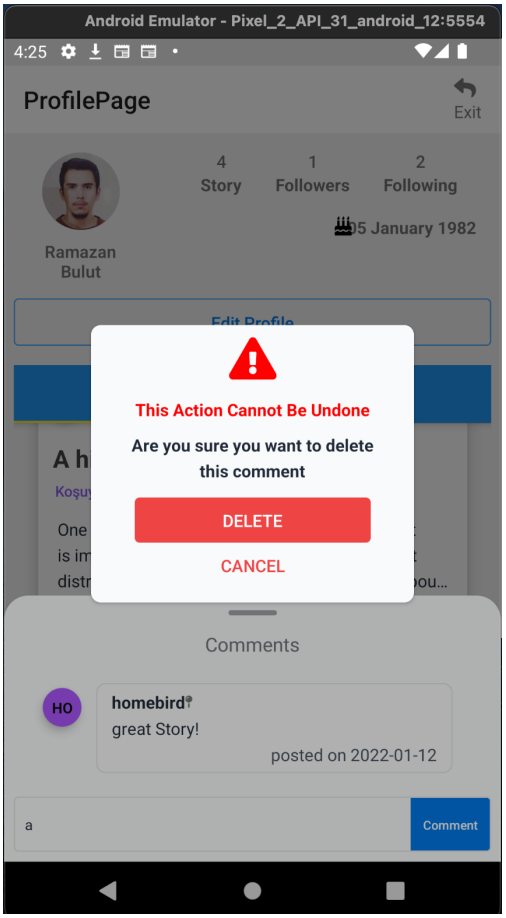
To reply to the comment, just press the reply button in the popup that opens, then follow instructions for commenting. An informative text will appear on the left side of the text section so you can distinguish who you are replying to. You can cancel your replying action just by clicking this text.

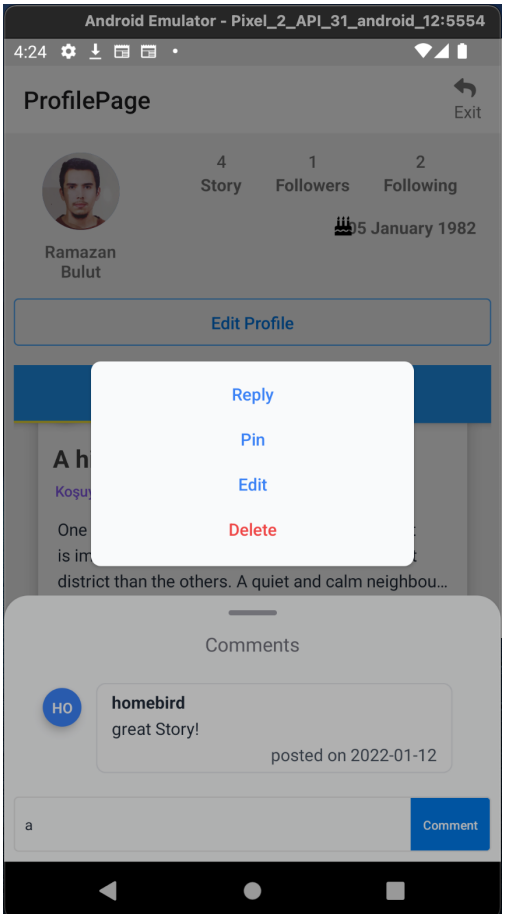


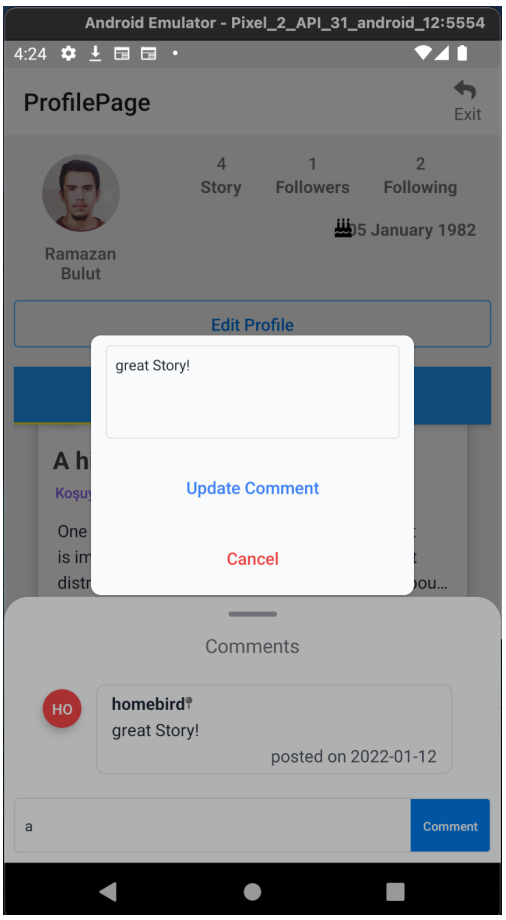


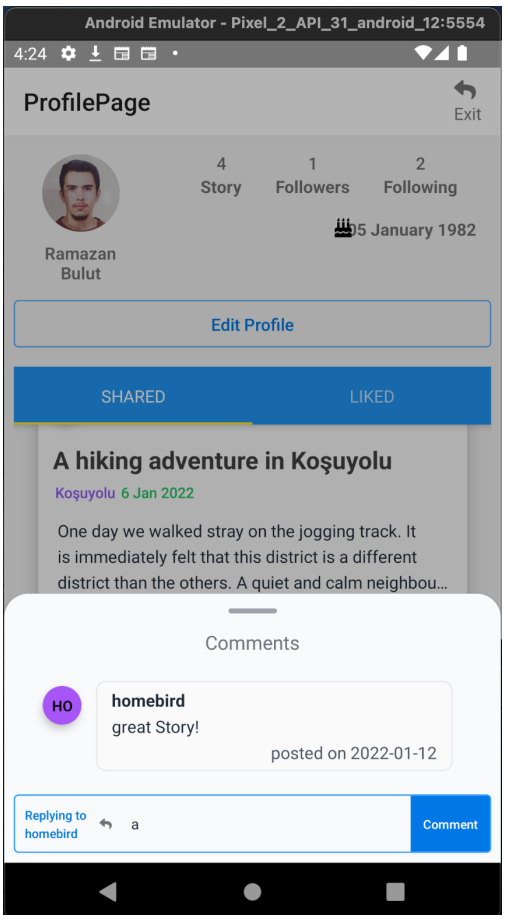

8:14
Home
Notifications
6 day(s) ago
See More
Reply
Report
Kadir
Ah very sad
6 day(s) ago
onurcanavci
I agreee Kadir
3 hour(s) ago
onurcanavci
poor for you :(
3 hour(s) ago
Comment..
Comment


Android Emulator - Pixel_2_API_31_android_12:5554
4:24
ProfilePage
Exit
4
Story
1
Followers
2
Following
Ramazan
Bulut
5 January 1982
Edit Profile
SHARED
LIKED
A hiking adventure in Koşuyolu
Koşuyolu 6 Jan 2022
One day we walked stray on the jogging track. It is immediately felt that this district is a different district than the others. A quiet and calm neighbou...
Comments
HO
homebird
great Story!
posted on 2022-01-12
a
Comment


Android Emulator - Pixel_2_API_31_android_12:5554
4:25
ProfilePage
Exit
4
Story
1
Followers
2
Following
Ramazan
Bulut
5 January 1982
This Action Cannot Be Undone
Are you sure you want to delete this comment
DELETE
CANCEL
Comments
HO
homebird
great Story!
posted on 2022-01-12
a
Comment


Android Emulator - Pixel_2_API_31_android_12:5554
4:24
ProfilePage
Exit
4
Story
1
Followers
2
Following
Ramazan
Bulut
5 January 1982
Edit Profile
Reply
Pin
Edit
Delete
district than the others. A quiet and calm neighbou...
Comments
HO
homebird
great Story!
posted on 2022-01-12
a
Comment


Android Emulator - Pixel_2_API_31_android_12:5554
4:24
ProfilePage
Exit
4
Story
1
Followers
2
Following
5 January 1982
Ramazan
Bulut
Edit Profile
great Story!
Update Comment
Cancel
Comments
HO
homebird
great Story!
posted on 2022-01-12
a
Comment

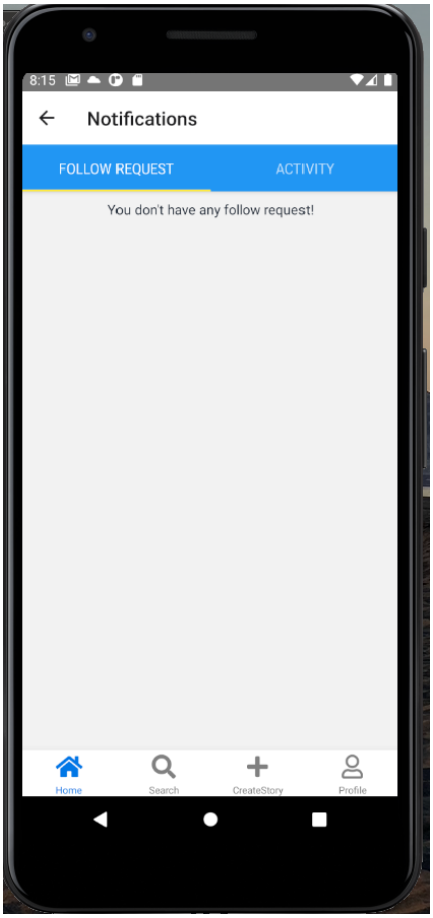

Android Emulator - Pixel_2_API_31_android_12:5554
4:24
ProfilePage
Exit
4
Story
1
Followers
2
Following
5 January 1982
Ramazan
Bulut
Edit Profile
SHARED
LIKED
A hiking adventure in Koşuyolu
Koşuyolu 6 Jan 2022
One day we walked stray on the jogging track. It is immediately felt that this district is a different district than the others. A quiet and calm neighbou...
Comments
HO
homebird
great Story!
posted on 2022-01-12
Replying to homebird
a
Comment

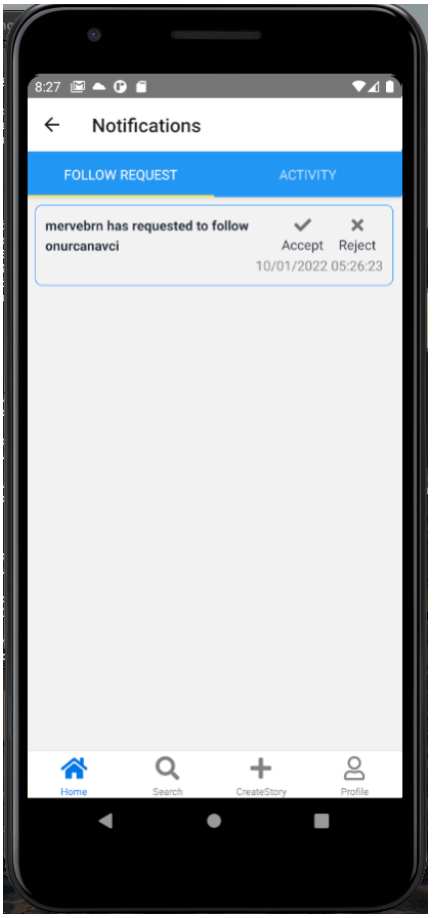
Notification Page:

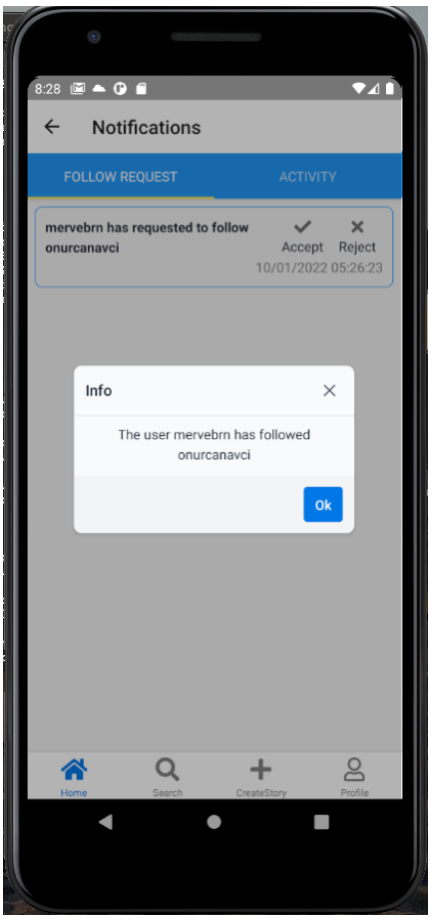
To reach this page, you can click on the bell icon in the upper right corner of the homepage. This page consists of two separate sections where you can view your follow requests and your or others' activities.

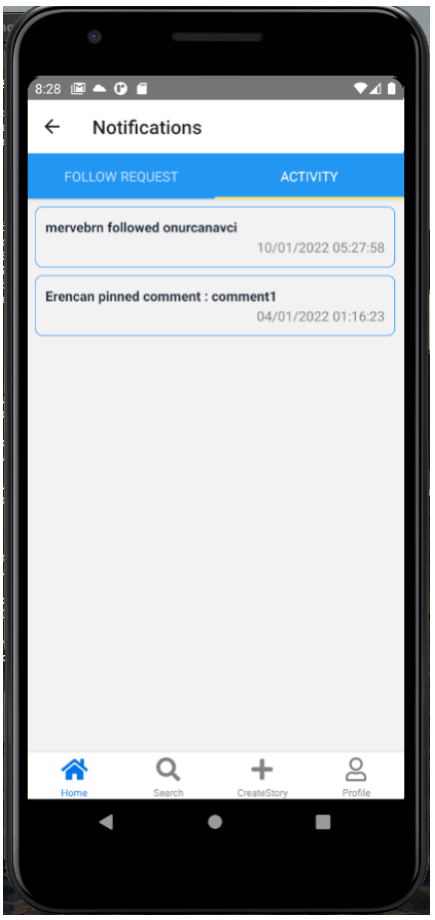
You can accept or reject the requests of users who want to follow you in the follow requests section by pressing the corresponding buttons. As you can see below, you will be notified with a popup according to your reactions.

In the activity section, you can see your and other users' activities and when these acts occur such as pinning, liking, commenting, following and sharing.






8:28
Notifications
FOLLOW REQUEST
ACTIVITY
mervebrn has requested to follow onurcanavci
Accept
Reject
10/01/2022 05:26:23
Info
The user mervebrn has followed onurcanavci
Ok
Home
Search
CreateStory
Profile


8:28
Notifications
FOLLOW REQUEST
ACTIVITY
mervebrn followed onurcanavci
10/01/2022 05:27:58
Erencan pinned comment : comment1
04/01/2022 01:16:23
Home
Search
CreateStory
Profile

Search Page:

You can jump to the search page with the navigation bar always at the bottom of the screen. There are 3 sections on this page where you can use different search methods.

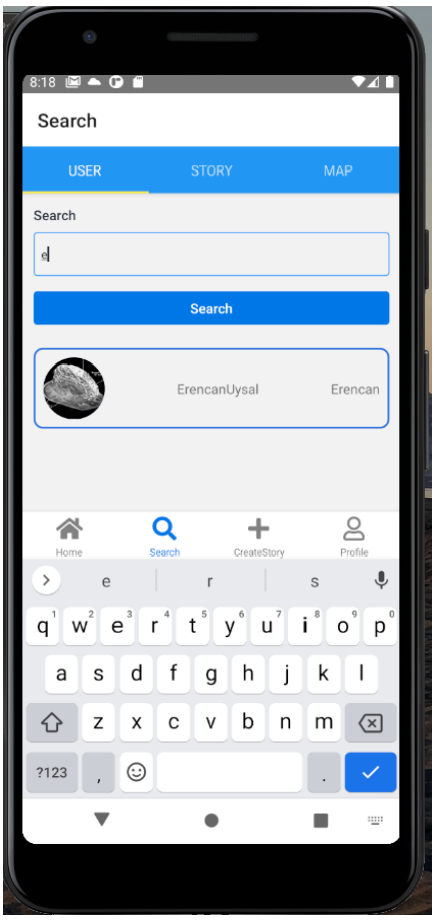
In the user section, you can find users by entering the word or words you want to search for in the text section of the user section and then pressing the search button. Also, by pressing each result, you can go to users' profile pages.

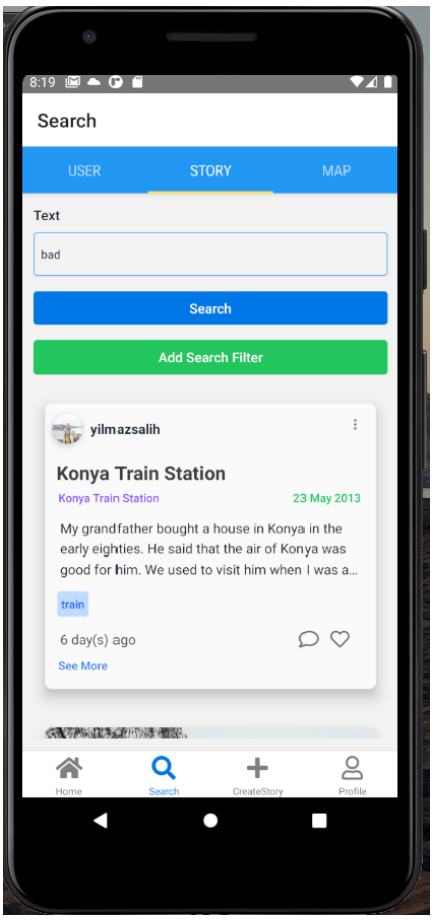
The story section is designed for detailed search of the stories shared in the application. In this section, you can search by filtering the stories as content, time and tag.

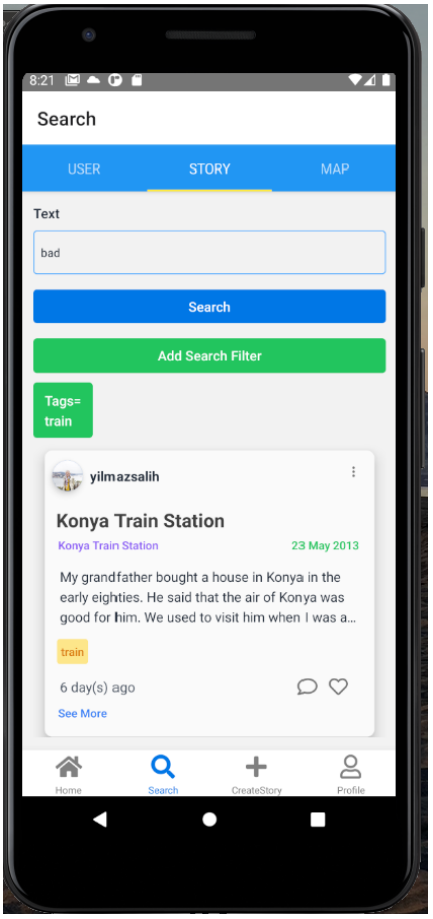
If you want to do your search only as the content of a story, just enter the word or words in the text area in the upper part you want to search and press the “Search” button. If results can be found, you will be able to see these results as a list from the bottom section and you will be able to interact with the results like on the home page. If there is no result, you will see a text that says “No Result Found”.
If you want to do a search by filtering the tags, you need to press the tags in the view that appears after pressing the “Add Search Filter” button. Next, you will see a text field where you can enter the tags you want to search. After entering each tag in this field, you can proceed by confirming it. After completing this part, you can apply tag filters by pressing the “Apply” button.

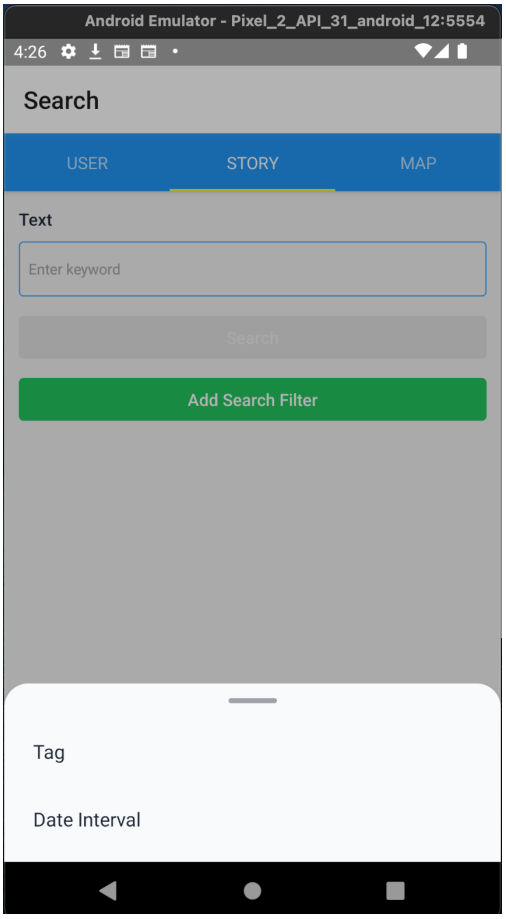
You can see it by looking at the boxes under the “Add Search Filter” button where the filters are applied. If you click on these boxes, the filter of the box you clicked will be canceled.

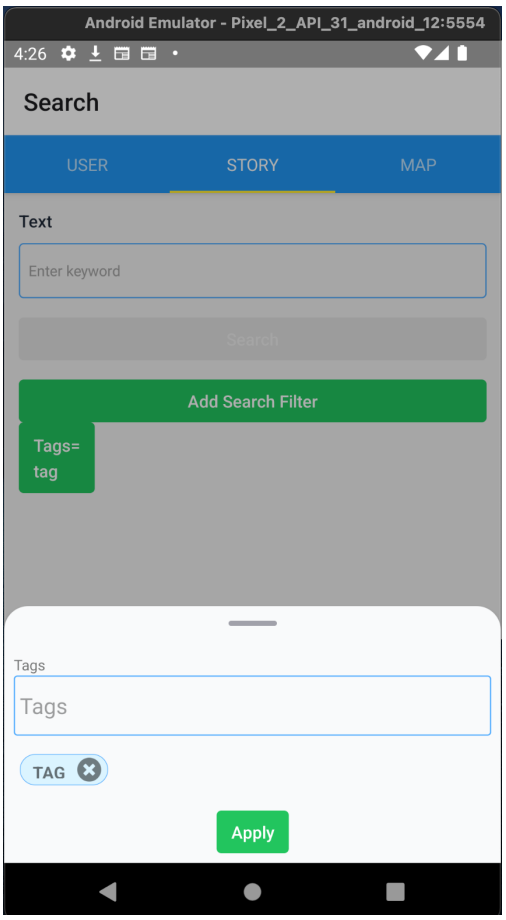
If you want to search for the date a story happens, when you click on the add search filter, you need to proceed by selecting the date part on the screen that appears. You can filter the stories in this range by selecting the start or end time or both on the screen that will appear. Boxes will appear when the filter is applied.

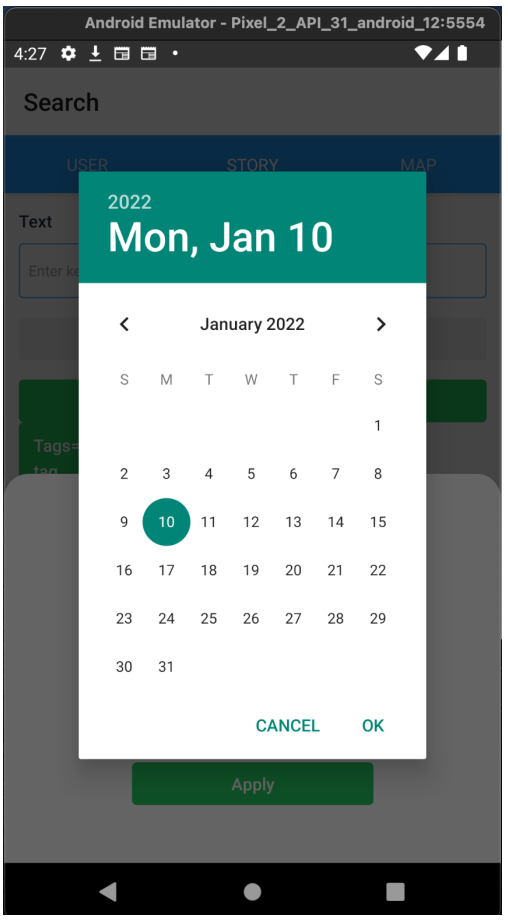

8:18
Search
USER
STORY
MAP
Search
e
Search
ErencanUysal
Erencan
Home
Search
CreateStory
Profile

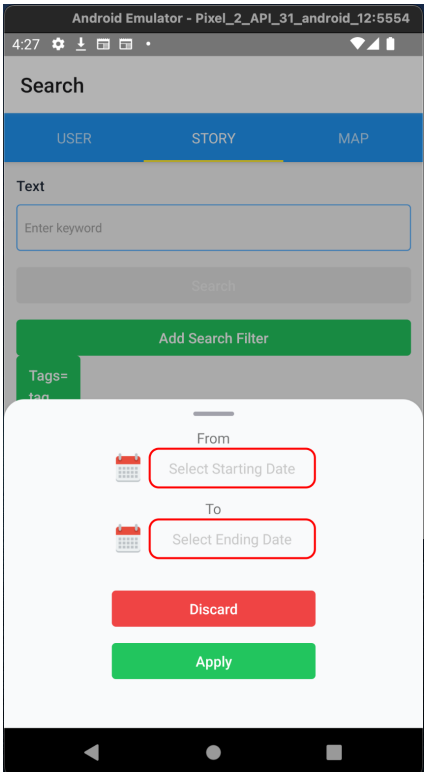

8:19
Search
USER
STORY
MAP
Text
bad
Search
Add Search Filter
yilmazsalih
Konya Train Station
Konya Train Station
23 May 2013
My grandfather bought a house in Konya in the early eighties. He said that the air of Konya was good for him. We used to visit him when I was a...
train
6 day(s) ago
See More
Home
Search
CreateStory
Profile


8:21
Search
USER
STORY
MAP
Text
bad
Search
Add Search Filter
Tags=
train
yilmazsalih
Konya Train Station
Konya Train Station
23 May 2013
My grandfather bought a house in Konya in the early eighties. He said that the air of Konya was good for him. We used to visit him when I was a...
train
6 day(s) ago
See More
Home
Search
CreateStory
Profile


Android Emulator - Pixel_2_API_31_android_12:5554
4:26
Search
USER
STORY
MAP
Text
Enter keyword
Search
Add Search Filter
Tag
Date Interval


Android Emulator - Pixel_2_API_31_android_12:5554
4:26
Search
USER
STORY
MAP
Text
Enter keyword
Search
Add Search Filter
Tags=
tag
Tags
Tags
TAG
Apply

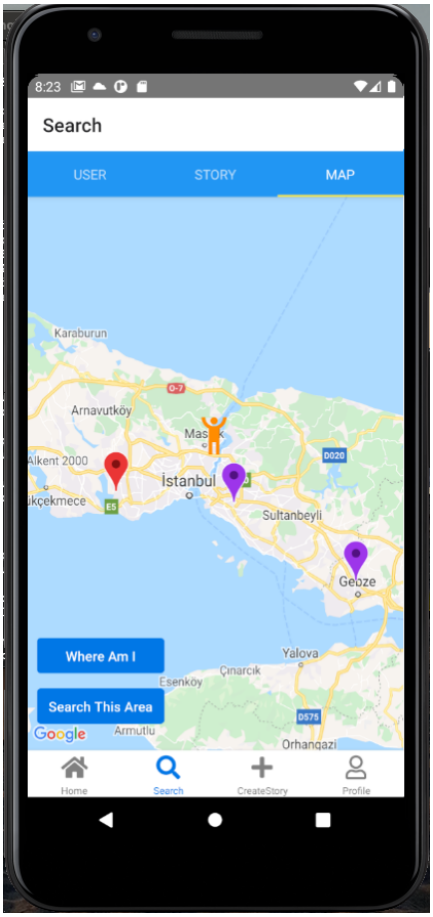

Android Emulator - Pixel_2_API_31_android_12:5554
4:27
Search
2022
Mon, Jan 10
January 2022
S M T W T F S
1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
CANCEL
OK
Apply

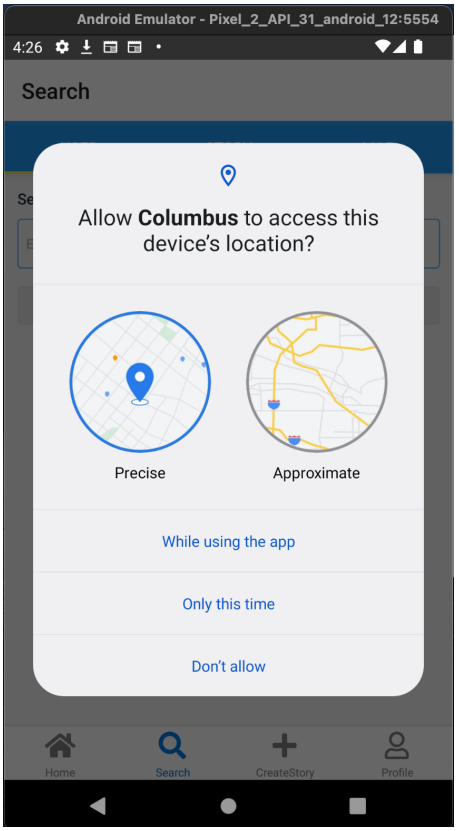

Android Emulator - Pixel_2_API_31_android_12:5554
4:27
Search
USER
STORY
MAP
Text
Enter keyword
Search
Add Search Filter
Tags=
From
Select Starting Date
To
Select Ending Date
Discard
Apply

You can adjust the area by zooming in or out, then you can reach the stories that took place in the area shown on the map by pressing the search this area button. Also, when you get lost on the map or want to go to your current location, you can press the Where am i button. This action will take you to your current location.

Markers on the map represent a shared story at that location. When you click on the markers, you can view the details of the story on the screen below and interact with the story.

.





4:46
Search
USER
STORY
MAP
Otomobil Müzesi
Acıbadem
aslak Hastanesi
Beykoz
İstinye Park
UmutKocasari
Old Days of Bogazici
Bogazici University
It is hard to believe that we study at 150 years old buildings. When one hears about that or sees the buildings, it does not seem very different however when we look at the old photos, we see how different their surroundings are. I felt very strange when I saw the pictures of my school after seeing the old buildings a lot of times. The pictures told us the surrounding of our campus were much more natural and authentic. (Photo: https://i.pinimg.com/originals/e9/96/8a/e9968a3eee35e301e73faf8700a394aa.jpg)
Old days
Bogazici University
posted on 2022-01-11

Create Story Page:

This screen is designed for you to share your stories. You need to create the story you want to share according to the following rules:

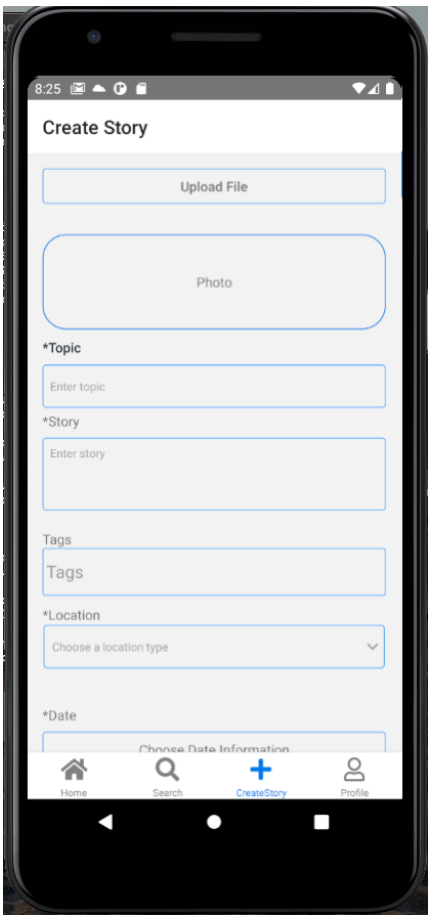
- You should enter a topic to related text area
- You should enter your story to related text area
- You should enter a location. It could be a virtual or real location.
- You should enter a date. It could be interval, decade, century or single date.
- You may add tags to your content
- You may add a photo to your content.

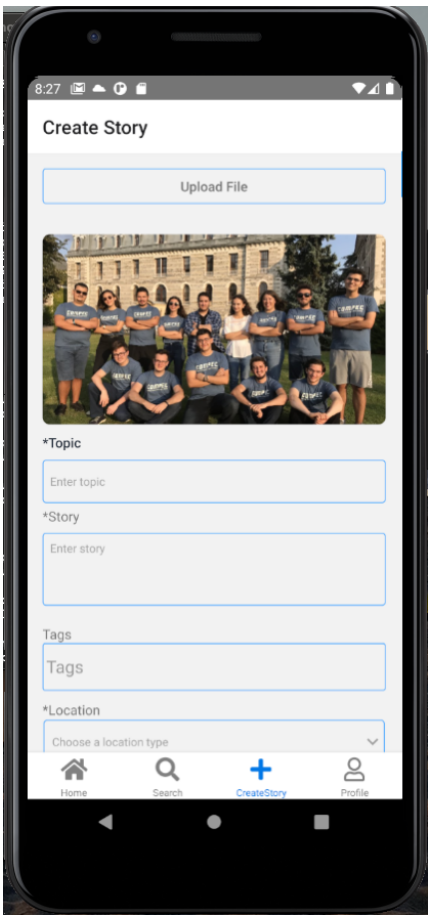
To add a photo to your story, you need to click on the upload File button at the top. Next, you need to select the photo you want to upload from your phone's photo library. At this stage, your operating system may request access to your photos for the first time. After this step, if the upload process is successful, the photo you upload will be displayed in the area under the upload file button.

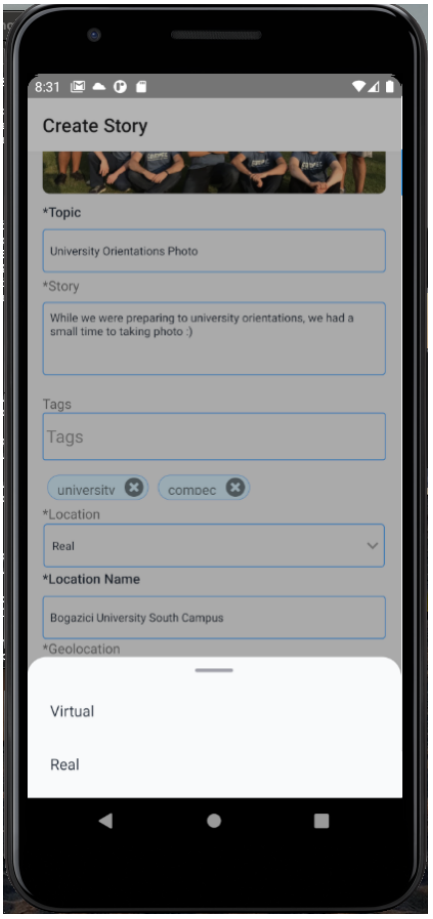
To specify a location in your story, you first need to choose the type of location as “virtual” or “real”. If you have chosen a virtual location, it is sufficient to specify the name of the location in the text field that appears. If you want to specify a real location, you need to select a location by clicking on the map that appears in addition to this. You can use the buttons to zoom in and out on the map. After selecting a location, you can change the location by moving the marker, or you can finish adding a location by clicking the “Save” button.

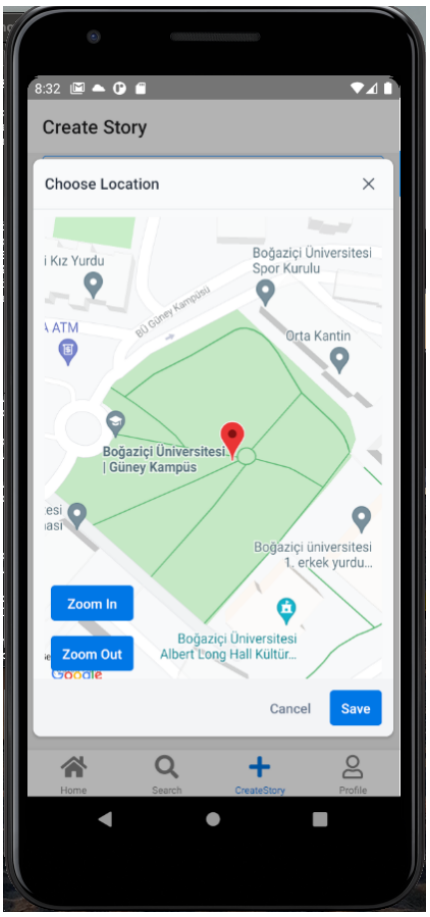
Here's what you need to do to put the date in your story:

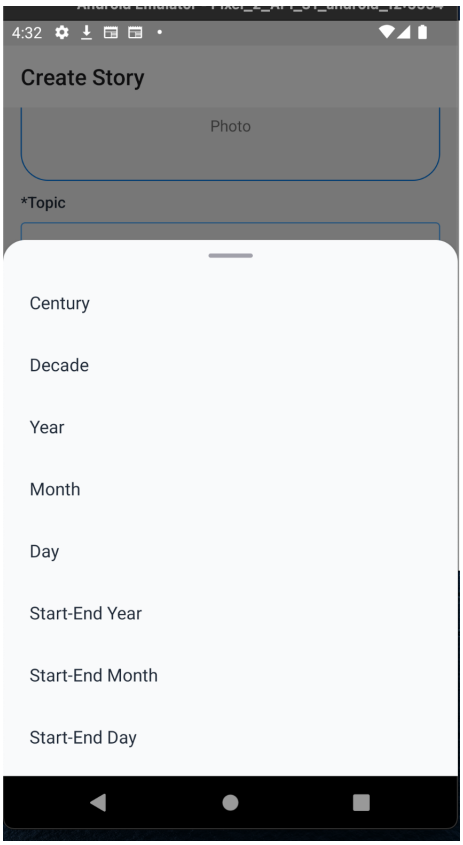
- Firstly select date type: century, decade, interval, or specific
- Then, type or select dates according to your date type selection.

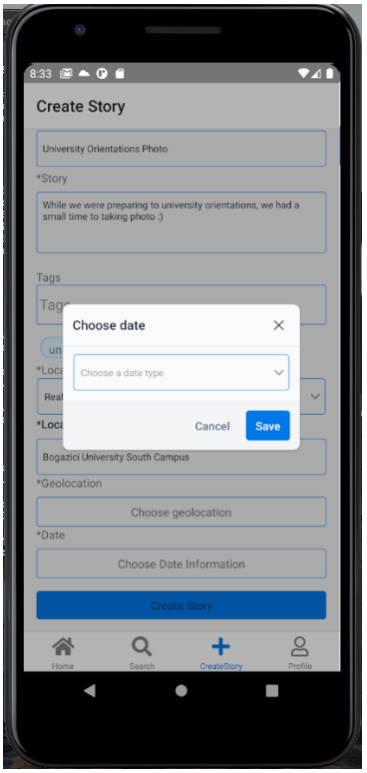

8:25
Create Story
Upload File
Photo
*Topic
Enter topic
*Story
Enter story
Tags
Tags
*Location
Choose a location type
*Date
Choose Date Information
Home
Search
CreateStory
Profile

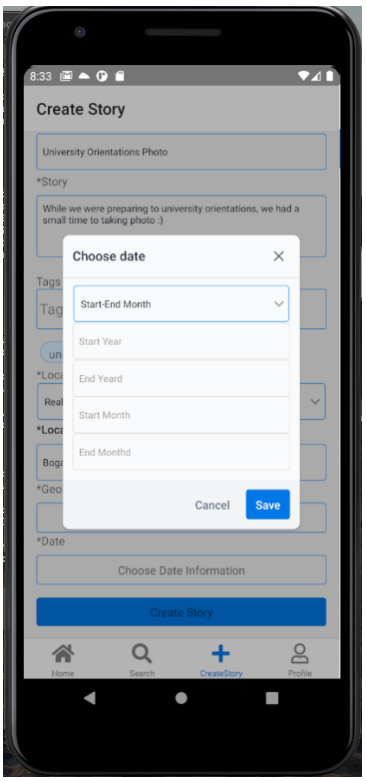

8:27
Create Story
Upload File
*Topic
Enter topic
*Story
Enter story
Tags
Tags
*Location
Choose a location type
Home
Search
CreateStory
Profile

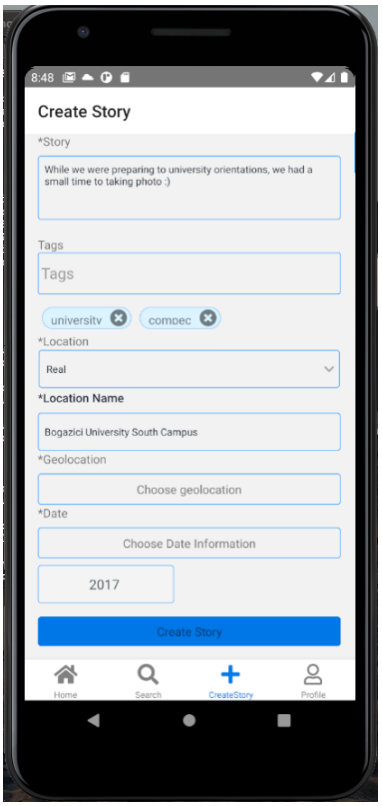

8:31
Create Story
*Topic
University Orientations Photo
*Story
While we were preparing to university orientations, we had a small time to taking photo :)
Tags
Tags
universitv
compec
*Location
Real
*Location Name
Bogazici University South Campus
*Geolocation
Virtual
Real

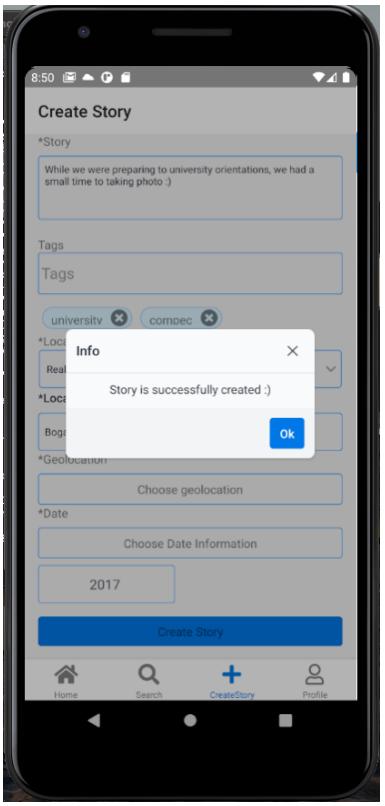

8:32
Create Story
Choose Location
Boğaziçi Üniversitesi Spor Kurulu
Orta Kantin
Boğaziçi Üniversitesi Güney Kampüs
Boğaziçi üniversitesi erkek yurdu...
Boğaziçi Üniversitesi Albert Long Hall Kültür...
Zoom In
Zoom Out
Cancel
Save
Home
Search
CreateStory
Profile


4:32
Create Story
Photo
*Topic
Century
Decade
Year
Month
Day
Start-End Year
Start-End Month
Start-End Day


8:33
Create Story
University Orientations Photo
*Story
While we were preparing to university orientations, we had a small time to taking photo :)
Tags
Choose date
Choose a date type
Cancel
Save
Bogazici University South Campus
*Geolocation
Choose geolocation
*Date
Choose Date Information
Create Story
Home
Search
CreateStory
Profile





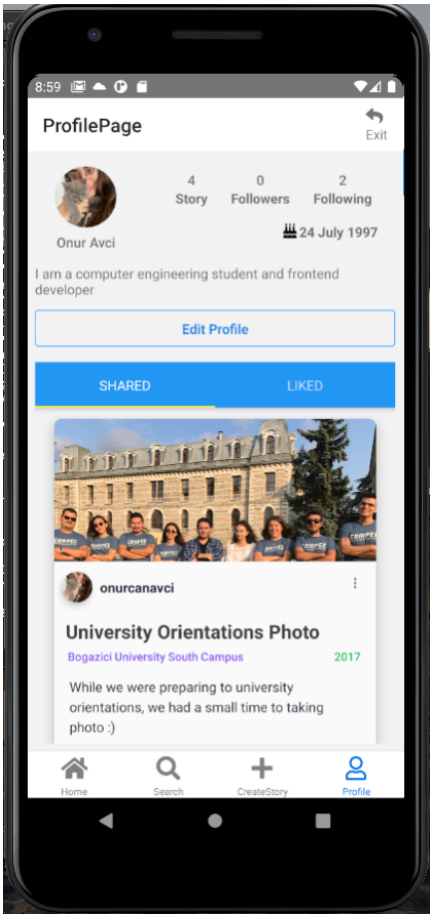


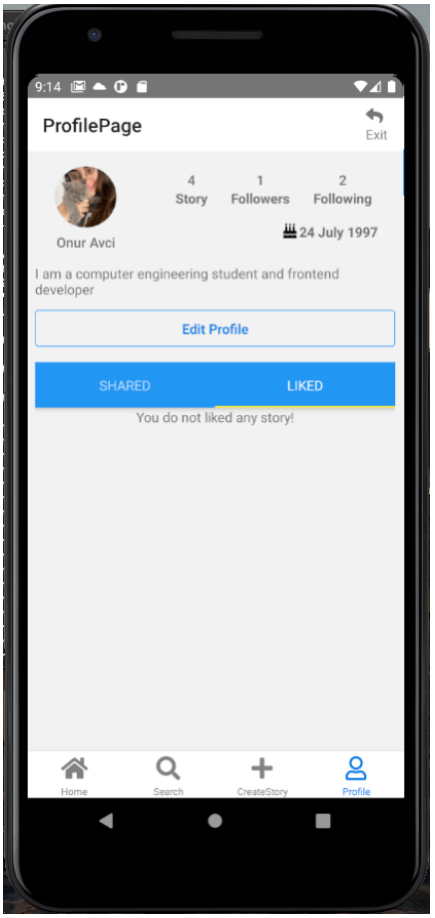
Profile Page:

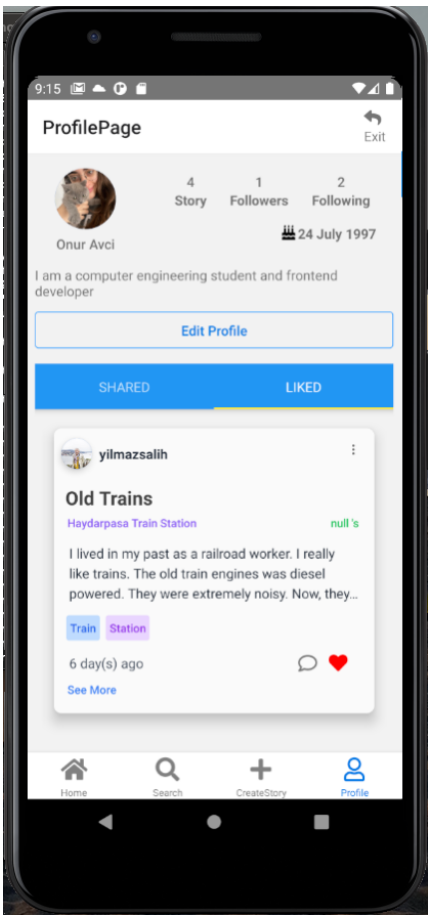
You can reach your own profile page by clicking the icon on the far right from the navigation bar below. On this page, you can view their followers and following by clicking on the part where the numbers are shown. You can go to the profiles of the users by clicking the profile cards in this section.

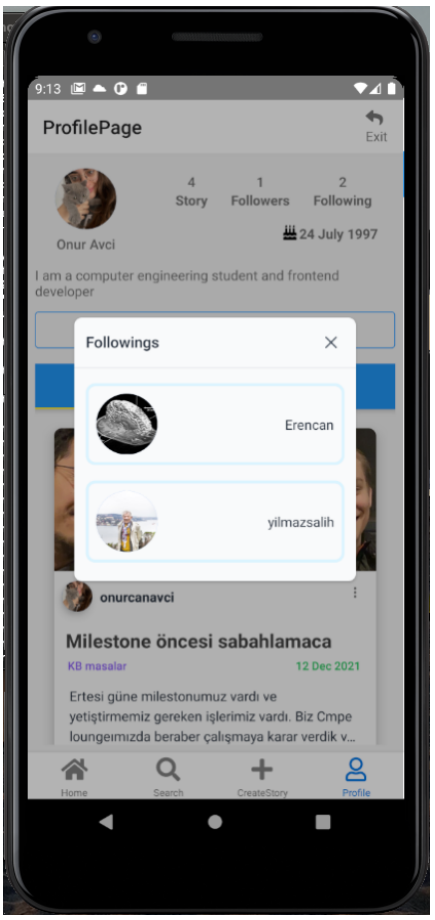
You can view the stories you have shared and liked separately.

When you press the “Edit Profile” button, it will take you to another screen where you can edit your profile and information.





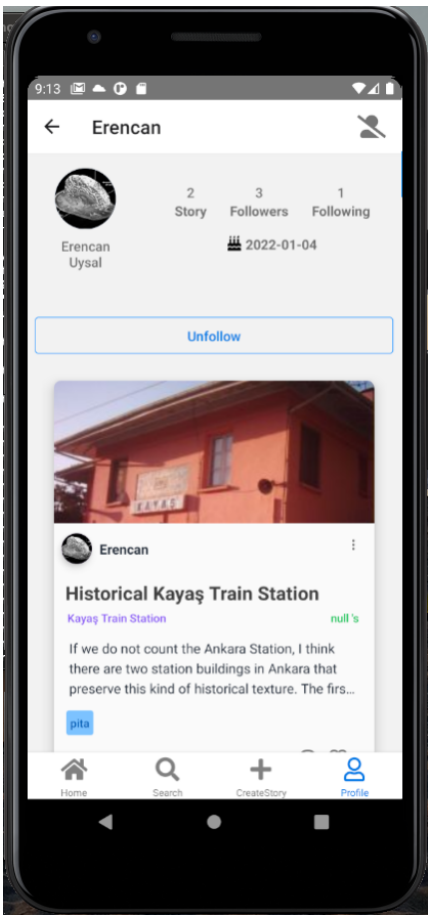


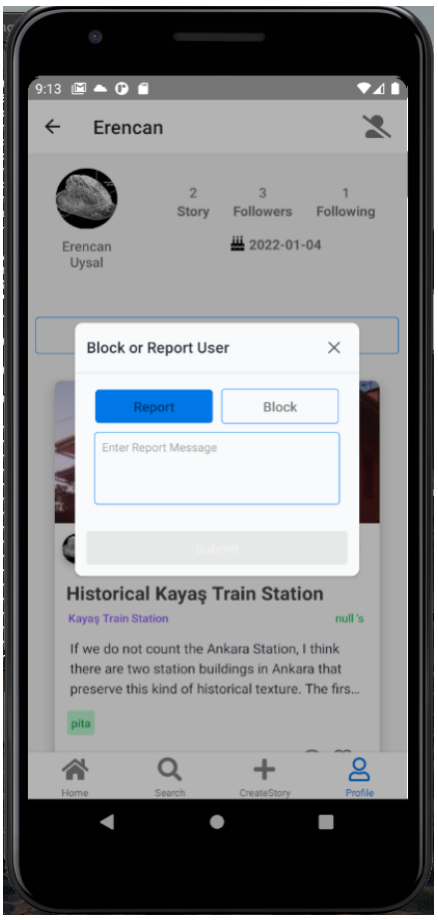


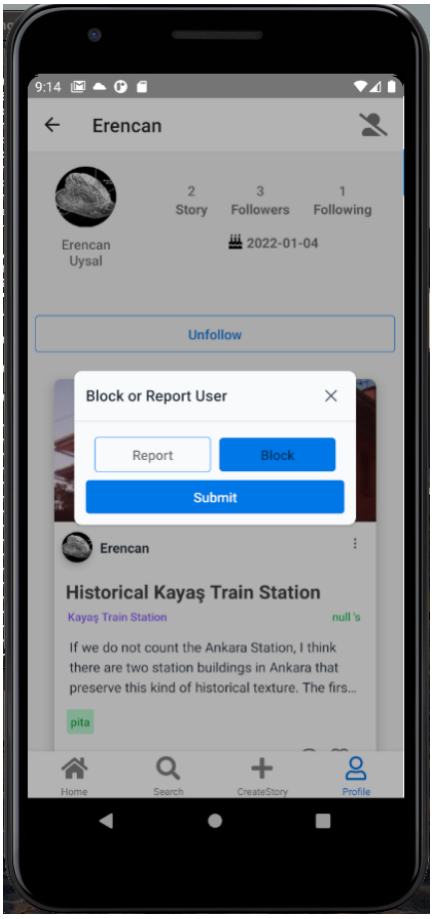
To reach the profiles of others, you can go to their profiles by clicking on the usernames in the story cards, the user search section, and the followers and followers section.

To reach the profiles of others, you can go to their profiles by clicking on the usernames in the story cards, the user search section, and the following and following sections. On these pages, you can still see the posts shared and liked by the user, unless the user's profile is private.

Here, you can set whether you follow the user or not. In addition, you can block or report the user from the menu that appears when you click the button on the top right.


9:13
Erencan
2
Story
3
Followers
1
Following
Erencan
Uysal
2022-01-04
Unfollow
Erencan
Historical Kayaş Train Station
Kayaş Train Station
null 's
If we do not count the Ankara Station, I think there are two station buildings in Ankara that preserve this kind of historical texture. The firs...
pita
Home
Search
CreateStory
Profile

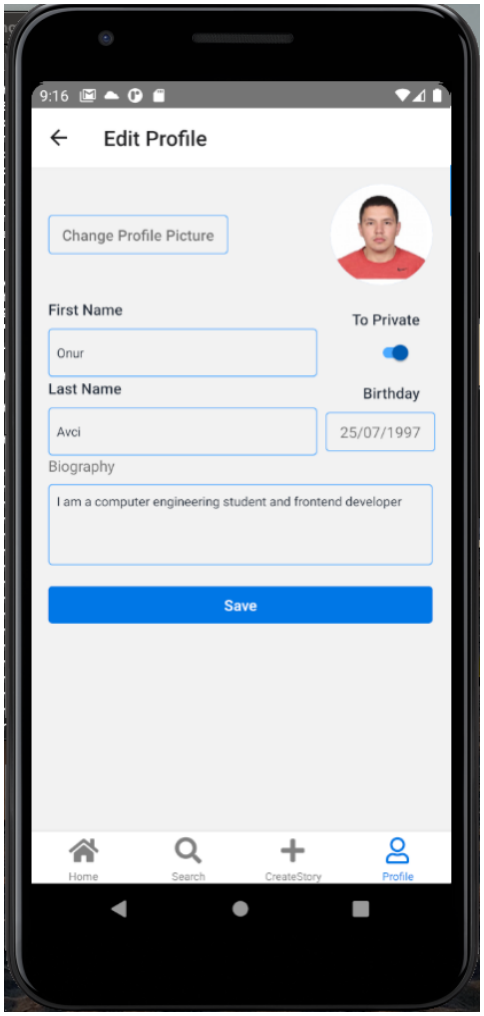

9:13
Erencan
2
Story
3
Followers
1
Following
Erencan
Uysal
2022-01-04
Block or Report User
Report
Block
Enter Report Message
Submit
Historical Kayaş Train Station
Kayaş Train Station
null 's
If we do not count the Ankara Station, I think there are two station buildings in Ankara that preserve this kind of historical texture. The firs...
pita
Home
Search
CreateStory
Profile


9:14
Erencan
2
Story
3
Followers
1
Following
Erencan
Uysal
2022-01-04
Unfollow
Block or Report User
Report
Block
Submit
Erencan
Historical Kayaş Train Station
Kayaş Train Station
null 's
If we do not count the Ankara Station, I think there are two station buildings in Ankara that preserve this kind of historical texture. The firs...
pita
Home
Search
CreateStory
Profile

Edit Profile Page:

This page is designed for you to edit your profile. You can make the following edits to this page:

- You can change your profile picture by clicking “Change Profile Picture” button
- You can change your first and last name by changing related text areas
- You can make your private by using switch function
- You can change your birthday
- You can change your bio.



8. System Manual

Minimum Requirements for Building and Running

To be able to build the frontend and backend files Docker* support is necessary with the following minimum requirements:

- Docker version >= 20.10.6
- Docker compose version = 3.8
- Min. of 4GB memory to build the docker images
- Min. of 1GB memory to run the docker images

Minimum Requirements for Frontend

The frontend supports these browsers:

- The version usage is above 0.2% versions of browsers are supported
- The opera mini and dead versions of browsers are not supported.
- More information about the browser usages can be found here

Building the Images

To build the application, please follow the below steps:

1. Make sure that Docker is installed in the computer. If not please follow the instructions on their website.
2. Go to the directory **Columbus** in the source code of the repository
3. Go to the **Frontend** directory
4. Change the **REACT_APP_API_URL** in the **.env** file to where the backend will be deployed.
5. For linux, run the command in the **Columbus** directory *echo "MAIL=Columbus_451_columbus\nDEBUG=True\n" > backend/src/.env* to set the run environment for the backend.
6. Run the following command to build the images for backend and frontend. If you have not enabled the using docker without sudo add the sudo prefix for each command:
    a. For Linux: docker-compose build
    b. For other OSs: docker compose build
    c. If you want to build the specific image specify the service name in the docker-compose file:
        i. For backend: docker-compose build backend
        ii. For frontend: docker-compose build frontend
7. The backend and frontend images should start the build

Running the Images

1. Run the built images by running the following commands, you can also use -d command to start the containers at detach state:
    a. For linux: docker-compose up
    b. For otherOSs: docker compose up

c. For only one of the services use the following commands:
   i. For backend: docker-compose up backend
   ii. For frontend: docker-compose up frontend

Accessing the Running Images

1. You can access the frontend with HTTP at the deployed URL and the backend at 8000 port at the deployed URL

MOBILE APPLICATION (ANDROID)

Requirements

Environment

- Android Studio
  - You can download Android Studio via https://developer.android.com/studio
- Android SDK
  - minSdkVersion = 21
  - compileSdkVersion = 30
  - targetSdkVersion = 30

  You can use Android Studio to install SDK by selecting checkboxes while installing it. After that you should set environment variables:

  1. Get Android SDK directory on SDK Manager of Android Studio
     a. "Preferences" dialog, under Appearance & Behavior → System Settings → Android SDK.
     b. Configure the **ANDROID_HOME** environment variable as directory of SDK or **c.**
     c. You can also create a "local.properties" file in the directory: "Columbus/mobile/Columbus/android/". Then place code below:

```
sdk.dir=export ANDROID_HOME= path to
ANDROID_SDK_HOME
```

     d. If you want to use emulator you should add another environment variable:

```
%LOCALAPPDATA%\Android\Sdk\platform-tools
```

- Node
  - On Windows:

You can install it via Chocolatey (package manager system) by running following command:
`choco install nodejs` as Administrator Command Prompt.

- On Mac OS:

  `brew install node`
  `brew install watchman` via Homebrew package manager system

- Java Development Kit
  - 1.8
- npm (Node package manager) **or** Yarn
  - 8.1.0 (npm)
  - 1.22.17 (yarn)

- react-native-cli
  - yarn add react-native-cli **or**
  - npm i react-native-cli

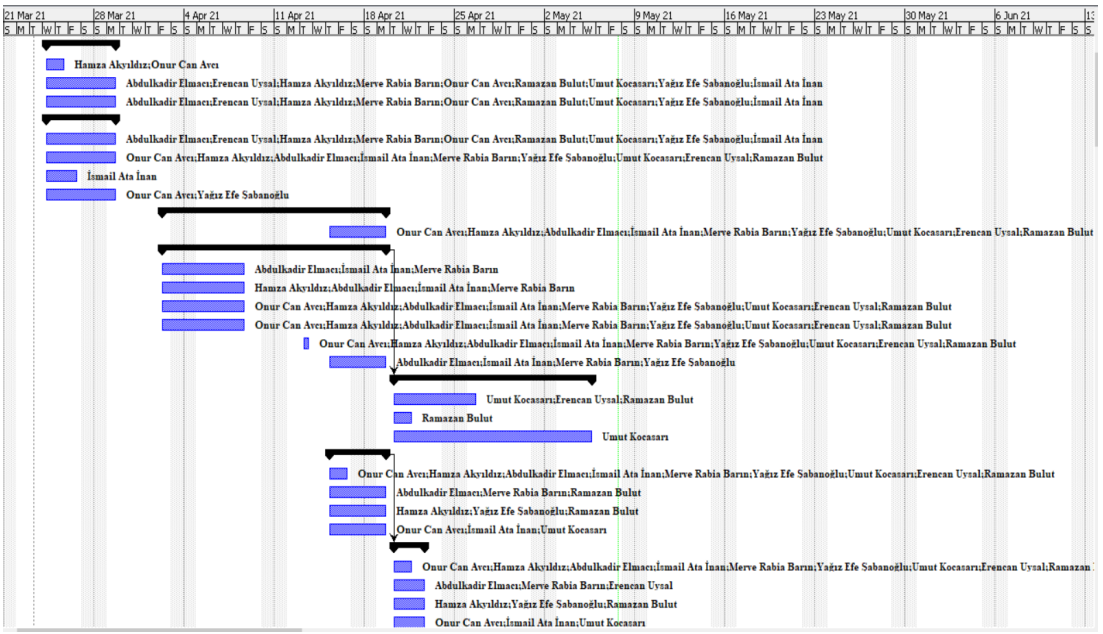
You can look at these web page for any trouble:
- https://reactnative.dev/docs/environment-setup
- https://reactnative.dev/docs/running-on-device

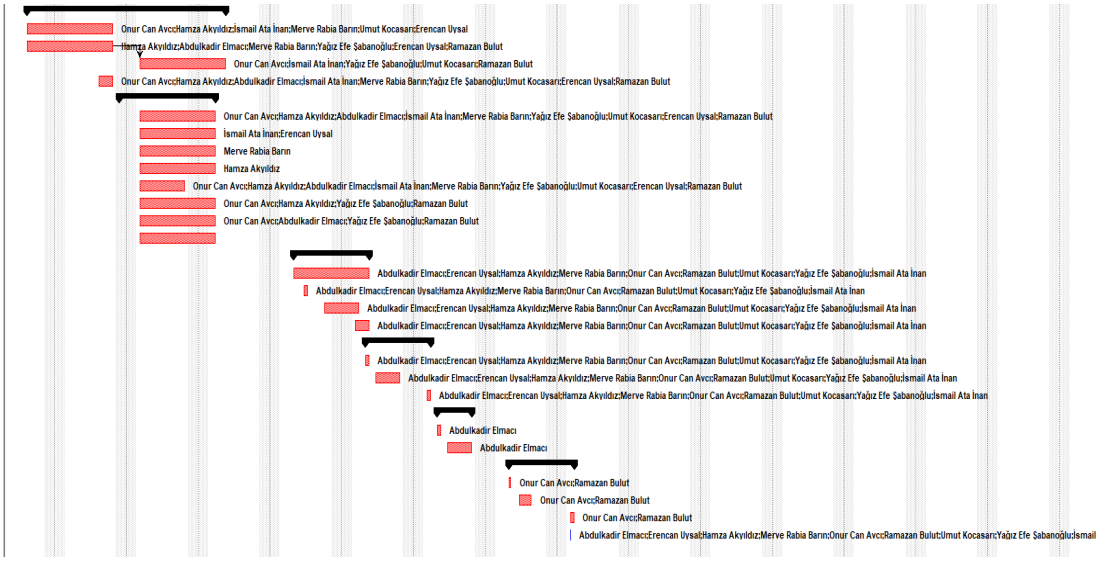
To RUN:
1. yarn **or** npm install in directory: **Columbus/mobile/Columbus**
2. react-native run-android **or** yarn android

Also You can find APK here:
https://github.com/bounswe/2021SpringGroup7/blob/CM-37/Columbus/mobile/app-release.apk
You can directly install it to your Mobile Phone or Emulator.

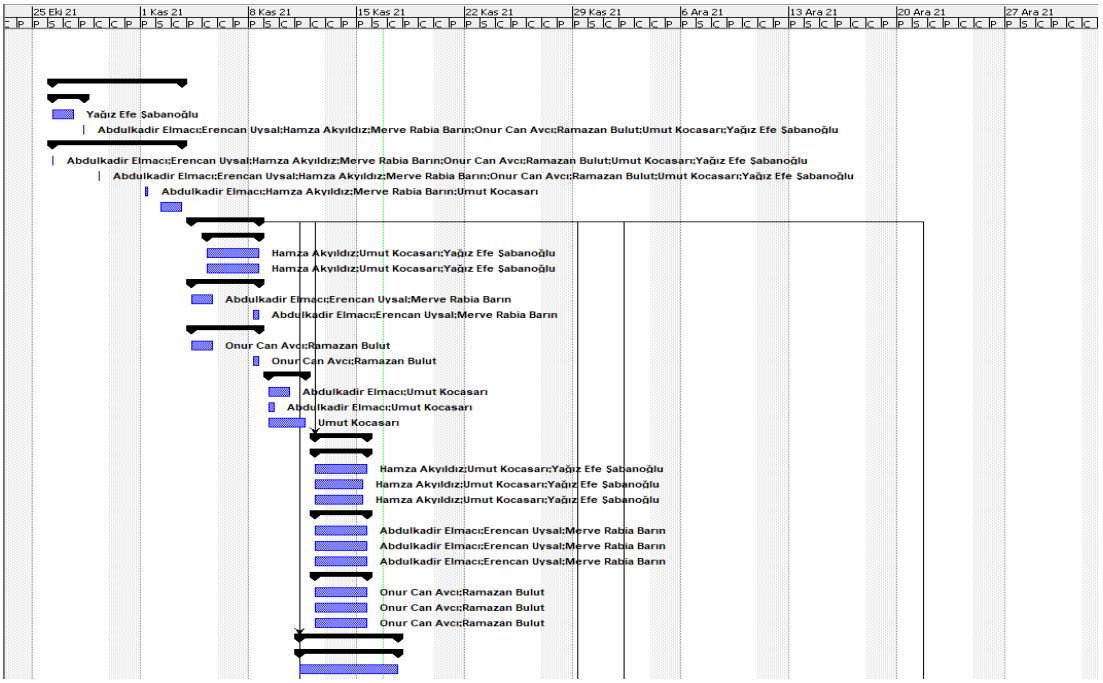
9. Project Plan

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		**ORIENTATION**	4 days?	**3/24/21 8:00 AM**	**3/29/21 5:00 PM**		
2		Creating Slack for Team Communication	2 days?	3/24/21 8:00 AM	3/25/21 5:00 PM		Hamza Akyıldız;Onur Can Avcı
3		Researching about Version Control System	4 days?	3/24/21 8:00 AM	3/29/21 5:00 PM		Abdulkadir Elmacı;Erencan U...
4		Researching About Favourite GitHub Repositories	4 days?	3/24/21 8:00 AM	3/29/21 5:00 PM		Abdulkadir Elmacı;Erencan U...
5		**DOCUMENTATION**	4 days?	**3/24/21 8:00 AM**	**3/29/21 5:00 PM**		
6		Creating Communication Plan	4 days?	3/24/21 8:00 AM	3/29/21 5:00 PM		Abdulkadir Elmacı;Erencan U...
7		Creating Personal Wikipages	4 days?	3/24/21 8:00 AM	3/29/21 5:00 PM		Onur Can Avcı;Hamza Akyıl...
8		Creating Wiki Home Page	3 days?	3/24/21 8:00 AM	3/26/21 5:00 PM		İsmail Ata İnan
9		Researching Platforms Similar to Our Project	4 days?	3/24/21 8:00 AM	3/29/21 5:00 PM		Onur Can Avcı;Yağız Efe Şa...
10		**NAME**	12 days?	**4/2/21 8:00 AM**	**4/19/21 5:00 PM**		
11		Recommending an App Name	3 days?	4/15/21 8:00 AM	4/19/21 5:00 PM		Onur Can Avcı;Hamza Akyıl...
12		**REQUIREMENTS**	12 days?	**4/2/21 8:00 AM**	**4/19/21 5:00 PM**		
13		Creating Glossary	5 days?	4/2/21 8:00 AM	4/8/21 5:00 PM		Abdulkadir Elmacı;İsmail Ata...
14		Determining the requirements	5 days?	4/2/21 8:00 AM	4/8/21 5:00 PM		Hamza Akyıldız;Abdulkadir El...
15		Adding Functional Requirements	5 days?	4/2/21 8:00 AM	4/8/21 5:00 PM		Onur Can Avcı;Hamza Akyıl...
16		Adding Non-Functional Requirements	5 days?	4/2/21 8:00 AM	4/8/21 5:00 PM		Onur Can Avcı;Hamza Akyıl...
17		Customer Meeting with TA	1 day?	4/13/21 8:00 AM	4/13/21 5:00 PM		Onur Can Avcı;Hamza Akyıl...
18		Updating the requirements after customer meeti	3 days?	4/15/21 8:00 AM	4/19/21 5:00 PM		Abdulkadir Elmacı;İsmail Ata...
19		**QUESTIONS**	12 days?	**4/20/21 8:00 AM**	**5/5/21 5:00 PM**	**12**	
20		Determining Questions	5 days?	4/20/21 8:00 AM	4/26/21 5:00 PM		Umut Kocasarı;Erencan Uys...
21		Classifying Questions	2 days?	4/20/21 8:00 AM	4/21/21 5:00 PM		Ramazan Bulut
22		Arranging answers of the questions	12 days?	4/20/21 8:00 AM	5/5/21 5:00 PM		Umut Kocasarı
23		**SCENARIOS**	3 days?	**4/15/21 8:00 AM**	**4/19/21 5:00 PM**		
24		Research on Scenarios	2 days?	4/15/21 8:00 AM	4/16/21 5:00 PM		Onur Can Avcı;Hamza Akyıl...
25		Creating User Scenarios	3 days?	4/15/21 8:00 AM	4/19/21 5:00 PM		Abdulkadir Elmacı;Merve Ra...
26		Creating Admin Scenarios	3 days?	4/15/21 8:00 AM	4/19/21 5:00 PM		Hamza Akyıldız;Yağız Efe Şa...
27		Creating Guest Scenario	3 days?	4/15/21 8:00 AM	4/19/21 5:00 PM		Onur Can Avcı;İsmail Ata İn...
28		**MOCKUPS**	3 days?	**4/20/21 8:00 AM**	**4/22/21 5:00 PM**	**23**	
29		Research on Mockups	2 days?	4/20/21 8:00 AM	4/21/21 5:00 PM		Onur Can Avcı;Hamza Akyıl...
30		Creating User Mockups	3 days?	4/20/21 8:00 AM	4/22/21 5:00 PM		Abdulkadir Elmacı;Merve Ra...
31		Creating Admin Mockups	3 days?	4/20/21 8:00 AM	4/22/21 5:00 PM		Hamza Akyıldız;Yağız Efe Şa...
32		Creating Guest Mockups	3 days?	4/20/21 8:00 AM	4/22/21 5:00 PM		Onur Can Avcı;İsmail Ata İn...

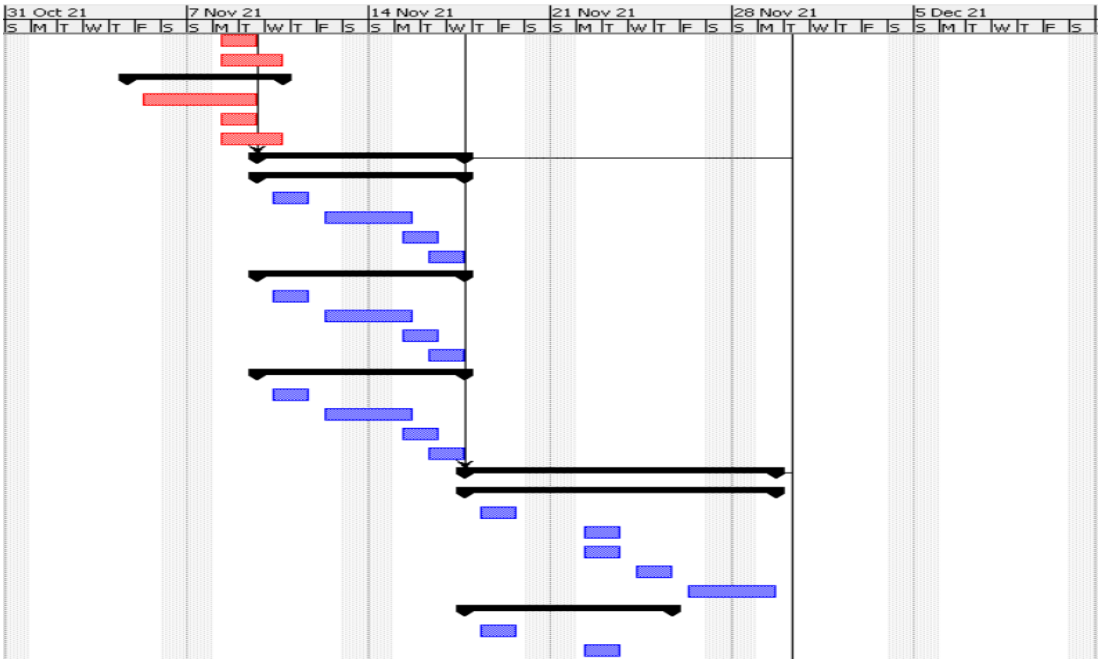



Onur Can Avcı;Hamza Akyıldız;İsmail Ata İnan;Merve Rabia Barın;Umut Kocasarı;Erencan Uysal
Hamza Akyıldız;Abdulkadir Elmacı;Merve Rabia Barın;Yağız Efe Şabanoğlu;Erencan Uysal;Ramazan Bulut
Onur Can Avcı;İsmail Ata İnan;Yağız Efe Şabanoğlu;Umut Kocasarı;Ramazan Bulut
Onur Can Avcı;Hamza Akyıldız;Abdulkadir Elmacı;İsmail Ata İnan;Merve Rabia Barın;Yağız Efe Şabanoğlu;Umut Kocasarı;Erencan Uysal;Ramazan Bulut
Onur Can Avcı;Hamza Akyıldız;Abdulkadir Elmacı;İsmail Ata İnan;Merve Rabia Barın;Yağız Efe Şabanoğlu;Umut Kocasarı;Erencan Uysal;Ramazan Bulut
İsmail Ata İnan;Erencan Uysal
Merve Rabia Barın
Hamza Akyıldız
Onur Can Avcı;Hamza Akyıldız;Abdulkadir Elmacı;İsmail Ata İnan;Merve Rabia Barın;Yağız Efe Şabanoğlu;Umut Kocasarı;Erencan Uysal;Ramazan Bulut
Onur Can Avcı;Hamza Akyıldız;Yağız Efe Şabanoğlu;Ramazan Bulut
Onur Can Avcı;Abdulkadir Elmacı;Yağız Efe Şabanoğlu;Ramazan Bulut
Abdulkadir Elmacı;Erencan Uysal;Hamza Akyıldız;Merve Rabia Barın;Onur Can Avcı;Ramazan Bulut;Umut Kocasarı;Yağız Efe Şabanoğlu;İsmail Ata İnan
Abdulkadir Elmacı;Erencan Uysal;Hamza Akyıldız;Merve Rabia Barın;Onur Can Avcı;Ramazan Bulut;Umut Kocasarı;Yağız Efe Şabanoğlu;İsmail Ata İnan
Abdulkadir Elmacı;Erencan Uysal;Hamza Akyıldız;Merve Rabia Barın;Onur Can Avcı;Ramazan Bulut;Umut Kocasarı;Yağız Efe Şabanoğlu;İsmail Ata İnan
Abdulkadir Elmacı;Erencan Uysal;Hamza Akyıldız;Merve Rabia Barın;Onur Can Avcı;Ramazan Bulut;Umut Kocasarı;Yağız Efe Şabanoğlu;İsmail Ata İnan
Abdulkadir Elmacı;Erencan Uysal;Hamza Akyıldız;Merve Rabia Barın;Onur Can Avcı;Ramazan Bulut;Umut Kocasarı;Yağız Efe Şabanoğlu;İsmail Ata İnan
Abdulkadir Elmacı;Erencan Uysal;Hamza Akyıldız;Merve Rabia Barın;Onur Can Avcı;Ramazan Bulut;Umut Kocasarı;Yağız Efe Şabanoğlu;İsmail Ata İnan
Abdulkadir Elmacı;Erencan Uysal;Hamza Akyıldız;Merve Rabia Barın;Onur Can Avcı;Ramazan Bulut;Umut Kocasarı;Yağız Efe Şabanoğlu;İsmail Ata İnan
Abdulkadir Elmacı
Abdulkadir Elmacı
Onur Can Avcı;Ramazan Bulut
Onur Can Avcı;Ramazan Bulut
Onur Can Avcı;Ramazan Bulut
Abdulkadir Elmacı;Erencan Uysal;Hamza Akyıldız;Merve Rabia Barın;Onur Can Avcı;Ramazan Bulut;Umut Kocasarı;Yağız Efe Şabanoğlu;İsmail

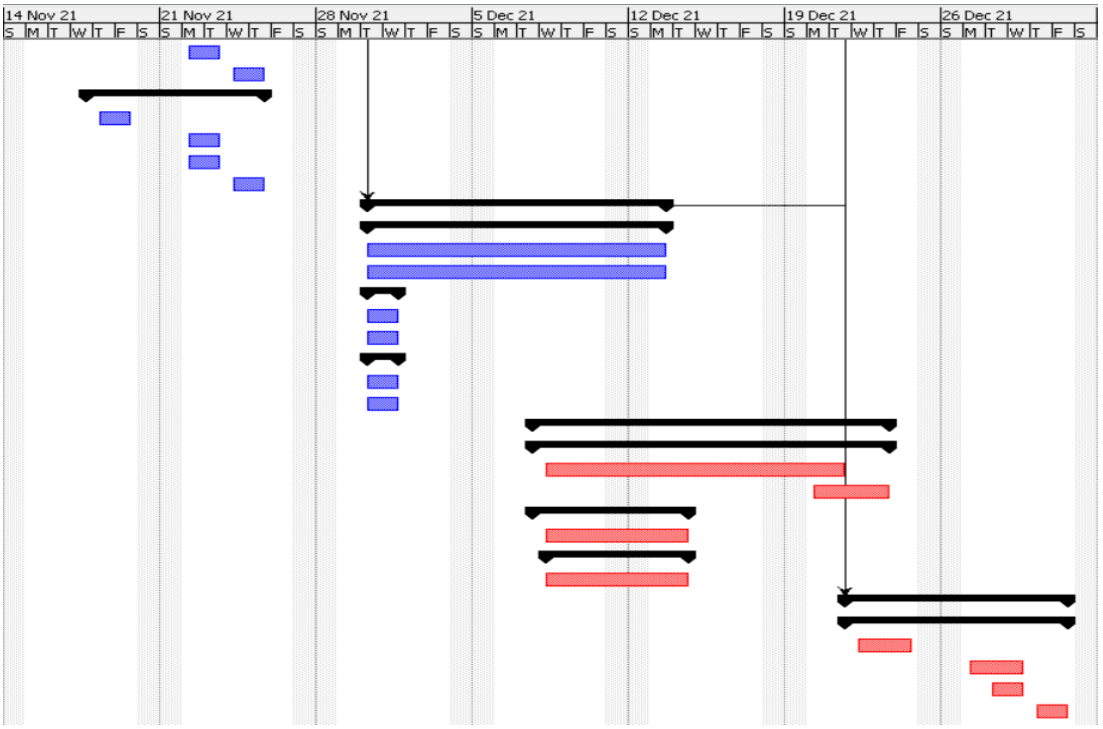
64		**Preparation For Implementation**	7 günler?	**26.10.2021 08:00**	**03.11.2021 17:00**		
65		**Planning**	2,25 günler?	**26.10.2021 08:00**	**28.10.2021 09:59**		
66		Project Plan Updates and Revision	2 günler?	26.10.2021 08:00	27.10.2021 17:00		Yağız Efe Şabanoğlu
67		Distribution of tasks(Backend,Frontend..)	0,25 günler?	28.10.2021 08:00	28.10.2021 09:59		Abdulkadir Elmacı;Erencan Uysal;Hamza Akyıldız;Merve ...
68		**Documentation**	7 günler?	**26.10.2021 08:00**	**03.11.2021 17:00**		
69		Updates	0,25 günler?	26.10.2021 08:00	26.10.2021 09:59		Abdulkadir Elmacı;Erencan Uysal;Hamza Akyıldız;Merve ...
70		Class and Sequence Diagrams	0,25 günler?	29.10.2021 08:00	29.10.2021 09:59		Abdulkadir Elmacı;Erencan Uysal;Hamza Akyıldız;Merve ...
71		Requirements	0,5 günler?	01.11.2021 08:00	01.11.2021 13:00		Abdulkadir Elmacı;Hamza Akyıldız;Merve Rabia Barın;Um...
72		Preparing Issue Templates	2 günler?	02.11.2021 08:00	03.11.2021 17:00		
73		**Building the Infrastructure**	3 günler?	**04.11.2021 08:00**	**08.11.2021 17:00**		
74		**Backend**	2 günler?	**05.11.2021 08:00**	**08.11.2021 17:00**		
75		Create Baseline Development Server	2 günler?	05.11.2021 08:00	08.11.2021 17:00		Hamza Akyıldız;Umut Kocasarı;Yağız Efe Şabanoğlu
76		Create Test Server	2 günler?	05.11.2021 08:00	08.11.2021 17:00		Hamza Akyıldız;Umut Kocasarı;Yağız Efe Şabanoğlu
77		**Frontend**	3 günler?	**04.11.2021 08:00**	**08.11.2021 17:00**		
78		Determining Palette Colors and Theme	2 günler?	04.11.2021 08:00	05.11.2021 17:00		Abdulkadir Elmacı;Erencan Uysal;Merve Rabia Barın
79		Creating Client for communicating with backend	1 gün?	05.11.2021 17:00	08.11.2021 17:00		Abdulkadir Elmacı;Erencan Uysal;Merve Rabia Barın
80		**Mobile**	3 günler?	**04.11.2021 08:00**	**08.11.2021 17:00**		
81		Determining Palette Colors and Theme	2 günler?	04.11.2021 08:00	05.11.2021 17:00		Onur Can Avcı;Ramazan Bulut
82		Creating Client for communicating with backend	1 gün?	05.11.2021 17:00	08.11.2021 17:00		Onur Can Avcı;Ramazan Bulut
83		**Configuration Management/ CI,CD Pipeline**	3 günler?	**09.11.2021 08:00**	**11.11.2021 17:00**		
84		Meeting	2 günler?	09.11.2021 08:00	10.11.2021 17:00		Abdulkadir Elmacı;Umut Kocasarı
85		Planning the Configuration	1 gün	09.11.2021 08:00	09.11.2021 17:00		Abdulkadir Elmacı;Umut Kocasarı
86		Designing an Activity Model	3 günler?	09.11.2021 08:00	11.11.2021 17:00		Umut Kocasarı
87		**Register-Login Page**	2 günler?	**12.11.2021 08:00**	**15.11.2021 17:00**	**73**	
88		**Backend**	2 günler?	**12.11.2021 08:00**	**15.11.2021 17:00**		
89		Register System	2 günler?	12.11.2021 08:00	15.11.2021 17:00		Hamza Akyıldız;Umut Kocasarı;Yağız Efe Şabanoğlu
90		Authentication	1,333 günler?	12.11.2021 08:00	15.11.2021 10:40		Hamza Akyıldız;Umut Kocasarı;Yağız Efe Şabanoğlu
91		LogIn System	1,333 günler?	12.11.2021 08:00	15.11.2021 10:40		Hamza Akyıldız;Umut Kocasarı;Yağız Efe Şabanoğlu
92		**Frontend**	2 günler?	**12.11.2021 08:00**	**15.11.2021 17:00**		
93		Auth Provider System	2 günler?	12.11.2021 08:00	15.11.2021 17:00		Abdulkadir Elmacı;Erencan Uysal;Merve Rabia Barın
94		Session Control System	2 günler?	12.11.2021 08:00	15.11.2021 17:00		Abdulkadir Elmacı;Erencan Uysal;Merve Rabia Barın
95		Register-Login Page Design	2 günler?	12.11.2021 08:00	15.11.2021 17:00		Abdulkadir Elmacı;Erencan Uysal;Merve Rabia Barın
96		**Mobile**	2 günler?	**12.11.2021 08:00**	**15.11.2021 17:00**		
97		Auth Provider System	2 günler?	12.11.2021 08:00	15.11.2021 17:00		Onur Can Avcı;Ramazan Bulut
98		Session Control System	2 günler?	12.11.2021 08:00	15.11.2021 17:00		Onur Can Avcı;Ramazan Bulut
99		Register-Login Page Design	2 günler?	12.11.2021 08:00	15.11.2021 17:00		Onur Can Avcı;Ramazan Bulut
100		**Home Page**	5 günler?	**11.11.2021 08:00**	**17.11.2021 17:00**	**73**	
101		**Backend**	5 günler?	**11.11.2021 08:00**	**17.11.2021 17:00**		
102		Extract following users' recent stories	5 günler?	11.11.2021 08:00	17.11.2021 17:00		
103		**Frontend**	3 günler?	**11.11.2021 08:00**	**15.11.2021 17:00**		
104		Notification System Front	3 günler?	11.11.2021 08:00	15.11.2021 17:00		



103		**Frontend**	3 günler?	**16.11.2021 08:00**	**18.11.2021 17:00**	
104		Notification System Front	3 günler?	16.11.2021 08:00	18.11.2021 17:00	
105		Creating Footer and Header, Navigation	2 günler?	16.11.2021 08:00	17.11.2021 17:00	
106		Creating Widgets for Story, Search Panel	3 günler?	16.11.2021 08:00	18.11.2021 17:00	
107		**Mobile**	3 günler?	**16.11.2021 08:00**	**18.11.2021 17:00**	
108		Notification System Front	3 günler?	16.11.2021 08:00	18.11.2021 17:00	
109		Creating Footer and Header, Navigation	2 günler?	16.11.2021 08:00	17.11.2021 17:00	
110		Creating Widgets for Story, Search Panel	3 günler?	16.11.2021 08:00	18.11.2021 17:00	
111		**Story Page**	4 günler?	**23.11.2021 08:00**	**26.11.2021 17:00**	73
112		**Backend**	4 günler?	**23.11.2021 08:00**	**26.11.2021 17:00**	
113		Like/Dislike Action	4 günler?	23.11.2021 08:00	26.11.2021 17:00	
114		Comment Action	4 günler?	23.11.2021 08:00	26.11.2021 17:00	
115		Report Action	4 günler?	23.11.2021 08:00	26.11.2021 17:00	
116		Edit Story	4 günler?	23.11.2021 08:00	26.11.2021 17:00	
117		**Frontend**	4 günler?	**23.11.2021 08:00**	**26.11.2021 17:00**	
118		Creating Tree based Widget for Comment	4 günler?	23.11.2021 08:00	26.11.2021 17:00	
119		Create and Edit Story Page	4 günler?	23.11.2021 08:00	26.11.2021 17:00	
120		Google Maps API Connection	4 günler?	23.11.2021 08:00	26.11.2021 17:00	
121		Create Comment Page	4 günler?	23.11.2021 08:00	26.11.2021 17:00	
122		**Mobile**	4 günler?	**23.11.2021 08:00**	**26.11.2021 17:00**	
123		Creating Tree based Widget for Comment	4 günler?	23.11.2021 08:00	26.11.2021 17:00	
124		Create and Edit Story Page	4 günler?	23.11.2021 08:00	26.11.2021 17:00	
125		Google Maps API Connection	4 günler?	23.11.2021 08:00	26.11.2021 17:00	
126		Create Comment Page	4 günler?	23.11.2021 08:00	26.11.2021 17:00	
127		**Profile Page**	3 günler?	**29.11.2021 08:00**	**01.12.2021 17:00**	73
128		**Backend**	3 günler?	**29.11.2021 08:00**	**01.12.2021 17:00**	
129		Follow/Unfollow	3 günler?	29.11.2021 08:00	01.12.2021 17:00	
130		Edit Profile Information	3 günler?	29.11.2021 08:00	01.12.2021 17:00	
131		View Shared Stories by User	3 günler?	29.11.2021 08:00	01.12.2021 17:00	
132		View Liked Stories	3 günler?	29.11.2021 08:00	01.12.2021 17:00	
133		Report Action	3 günler?	29.11.2021 08:00	01.12.2021 17:00	
134		**Frontend**	3 günler?	**29.11.2021 08:00**	**01.12.2021 17:00**	
135		Edit and View Only Profile Page Design	3 günler?	29.11.2021 08:00	01.12.2021 17:00	
136		Followers and Following List Design	3 günler?	29.11.2021 08:00	01.12.2021 17:00	
137		Settings Page Design	3 günler?	29.11.2021 08:00	01.12.2021 17:00	
138		Stories Shared and Stories Liked Tab Design	3 günler?	29.11.2021 08:00	01.12.2021 17:00	
139		**Mobile**	3 günler?	**29.11.2021 08:00**	**01.12.2021 17:00**	
140		Edit and View Only Profile Page Design	3 günler?	29.11.2021 08:00	01.12.2021 17:00	
141		Followers and Following List Design	3 günler?	29.11.2021 08:00	01.12.2021 17:00	
142		Settings Page Design	3 günler?	29.11.2021 08:00	01.12.2021 17:00	



142		Settings Page Design	3 günler?	29.11.2021 08:00	01.12.2021 17:00	
143		Stories Shared and Stories Liked Tab Design	2 günler?	29.11.2021 08:00	30.11.2021 17:00	
144		**Search System**	**10 günler?**	**02.12.2021 08:00**	**15.12.2021 17:00**	**73;111;127**
145		**Backend**	**10 günler?**	**02.12.2021 08:00**	**15.12.2021 17:00**	
146		Search Engine	10 günler?	02.12.2021 08:00	15.12.2021 17:00	
147		Filter Search Results	10 günler?	02.12.2021 08:00	15.12.2021 17:00	
148		**Frontend**	**2 günler?**	**02.12.2021 08:00**	**03.12.2021 17:00**	
149		Google Maps API for searching	2 günler?	02.12.2021 08:00	03.12.2021 17:00	
150		Search Page Design (Filters, Sort, etc.)	2 günler?	02.12.2021 08:00	03.12.2021 17:00	
151		**Mobile**	**2 günler?**	**02.12.2021 08:00**	**03.12.2021 17:00**	
152		Google Maps API for searching	2 günler?	02.12.2021 08:00	03.12.2021 17:00	
153		Search Page Design (Filters, Sort, etc.)	2 günler?	02.12.2021 08:00	03.12.2021 17:00	
154		**Explore Page**	**12 günler?**	**07.12.2021 17:00**	**23.12.2021 17:00**	
155		**Backend**	**12 günler?**	**07.12.2021 17:00**	**23.12.2021 17:00**	
156		Recommendation System	10 günler?	07.12.2021 17:00	21.12.2021 17:00	
157		Map Integration	4 günler?	17.12.2021 17:00	23.12.2021 17:00	
158		**Frontend**	**5 günler?**	**07.12.2021 17:00**	**14.12.2021 17:00**	
159		Explore Page Design (tag,)	5 günler?	07.12.2021 17:00	14.12.2021 17:00	
160		**Mobile**	**5 günler?**	**08.12.2021 08:00**	**14.12.2021 17:00**	
161		Explore Page Design (tag,)	5 günler?	08.12.2021 08:00	14.12.2021 17:00	
162		**Admin Page**	**3 günler?**	**30.12.2021 08:00**	**03.01.2022 17:00**	**73;144**
163		**Backend**	**3 günler?**	**30.12.2021 08:00**	**03.01.2022 17:00**	
164		List All Reported Contents	3 günler?	30.12.2021 08:00	03.01.2022 17:00	
165		Filter Reported Contents	3 günler?	30.12.2021 08:00	03.01.2022 17:00	
166		Add to/ Remove from Black List	2 günler?	30.12.2021 08:00	31.12.2021 17:00	
167		Add to Spam Folder	2 günler?	30.12.2021 08:00	31.12.2021 17:00	
168		**Frontend**	**3 günler?**	**30.12.2021 08:00**	**03.01.2022 17:00**	
169		Admin Page Design (general)	3 günler?	30.12.2021 08:00	03.01.2022 17:00	
170		Reported Content Pages (profile, story, commer	3 günler?	30.12.2021 08:00	03.01.2022 17:00	
171		MailBox System and Inbox Page	2 günler?	30.12.2021 08:00	31.12.2021 17:00	
172		Reporting tool	2 günler?	30.12.2021 08:00	31.12.2021 17:00	
173		**Mobile**	**3 günler?**	**30.12.2021 08:00**	**03.01.2022 17:00**	
174		Admin Page Design (general)	3 günler?	30.12.2021 08:00	03.01.2022 17:00	
175		Reported Content Pages (profile, story, commer	3 günler?	30.12.2021 08:00	03.01.2022 17:00	
176		MailBox System and Inbox Page	2 günler?	30.12.2021 08:00	31.12.2021 17:00	
177		Reporting tool	2 günler?	30.12.2021 08:00	31.12.2021 17:00	



10. Requirements

Glossary

1. Account Information : A user's personal data such as name, email, username, location, birthday etc.

2. Admin : A user of the system that organizes and manages other users' activities.

3. Banning : Prohibiting a user's access to the system.

4. Black List : The list of users who are not allowed the login the system.

5. Blocking : Event that represents a user(blocker) prohibiting another user's(blocked) interaction to herself.

6. Caption : A title or a brief explanatory text.

7. Comment : Text messages that users leave to a post.

8. Downtime : Time during when the system is unavailable for use.

9. Email : A Message that is sent from a computer user to other user or users via a network electronically.

10. Failure : An unsuccessful event in the system.

11. Follow : Where a user chooses to see another user's or location's posts in feed.

12. Guest : A person in the system that has not logged in.

13. HTTPS : Stands for Hypertext Transfer Protocol Secure.

14. Like : A user showing their approval to another user's activity.

15. Location : A real place in the world that has been added to the system.

16. Maintenance : A scheduled event that is for keeping the system in a good condition.

17. Mute : A user disabling the notifications of a user or a place.

18. Notification : A message sent by the system to a user.

19. Notifications Box : Where all the notifications are kept together.

20. Password : A character string that enables a user to login.

21. Pinning : The way of highlighting the desired content by user.

22. Post : Social media content that is uploaded the system such as photo, video, or story.

23. Privacy Policy : Legal document that states how the system discloses the usage of data gathered from users.

24. Private Account: Only followers who the account approves can see its profile and posts.

25. Profile : It is a unique page for each users showing their personal information.

26. Public Account: Anyone can see the account's post.

27. Recommendation : Recommending the other users or locations according to user's interests.

28. Reply : A function that allows the user to respond to a comment or a reply.

29. Report : Reporting any content that is in violations of the our Terms of Use.

30. Search : Trying to find a location, user or a post.

31. Semantic Search : Searching process with understanding a searcher's intent through contextual meaning.

32. Story : An account of past event in someone's life.

33.Spam Folder : The folder, in which users cannot access the contents, containing spam contents.

34. System: Software of the product.

35. Tag : Keyword added to a social media post to categorize content.

36. Terms of Use : The legal agreements that clarifies the legal obligations of the user, which is declared by service provider, in case of using the service.

37. User : Whoever has an account with which a user can successfully login to the system.

38. Username : A character string that identify a user.

1. Functional Requirements

1.1 User Requirements

- ### 1.1.1 Profile
  - **1.1.1.1** Users shall be able to block other users.
  - **1.1.1.2** Users shall be able to view any of their personal information (email, username and name, location, birthday if provided).
  - **1.1.1.3** Users shall be able to edit any of their personal information (email, username and name, location, birthday if provided).

- **1.1.1.4** Users shall be able to see users that they follow and are followed by.
- **1.1.1.5** Users shall be able to delete their accounts.

- ### 1.1.2 Posting
  - **1.1.2.1** Users shall be able to create posts with a story.
  - **1.1.2.2** Users shall be able to add tags, photo, video and voice to their posts.
  - **1.1.2.3** Users shall be able to add a location with its name or by specifying the location on the map interface.
  - **1.1.2.4** Users shall be able to add multiple locations related to their story.
  - **1.1.2.5** Users shall be able to specify time in their posts to declare exact time or time interval with a maximum of 10 years.
  - **1.1.2.6** Users shall be able to edit their own posts.
  - **1.1.2.7** Users shall be able to delete their own posts.
  - **1.1.2.8** Users shall be able to like the post of their followings.
  - **1.1.2.9** Users shall be able to comment on the post of their followings.
  - **1.1.2.10** Users shall be able to view users who liked or commented to their posts.
  - **1.1.2.11** Users shall be able to delete comments made to their own posts.
  - **1.1.2.12** Users shall be able to reply to comments made by other users with comments.
  - **1.1.2.13** Users should be able to save the posts of their followings.
  - **1.1.2.14** Users should be able to pin comments made to their own posts to the top.

- ### 1.1.3 User to User Interaction
  - **1.1.3.1** Users shall be able to view the posts of their following accounts.
  - **1.1.3.2** Users shall be able to report other users posts, if the user has suspicion about its accuracy.
  - **1.1.3.3** Users shall be able to follow other users.
  - **1.1.3.4** Users shall be able to unfollow other users.
  - **1.1.3.5** Users shall be able to view posts of other users that they follow or have public profile.
  - **1.1.3.6** Users shall be able to view the followers and followings of other users that they follow or have public profile.

- ### 1.1.4 Searching

- **1.1.4.1** Users shall be able search other users by their username.
- **1.1.4.2** Users shall be able search locations by their name.
- **1.1.4.3** Users shall be able to specify the searched location on the map.
- **1.1.4.4** Users shall be able to do semantic search by using tags.
- **1.1.4.5** Users shall be able to filter the search results according to time.

- **1.1.5 Recommendation**
  - **1.1.5.1** Users shall be able to view recommended locations based on their system activity and current location.

1.2 Guest Requirements

- **1.2.1** Guests shall be able to create a new account by choosing a password (email confirmation) and providing email, username (*name*, *location*, *birthday* optionally).
- **1.2.2** Guests shall be able to login to the system by entering username and password if they already have an account.
- **1.2.3** Guests shall be able to only search and view locations.

1.3 Admin Requirements

- **1.3.1** Admins shall be able to login to the system by entering username and password.
- **1.3.2** Admins shall be able to review any content that is reported by users.
- **1.3.3** Admins shall be able to filter reported contents for the types of posts, comments,tags and accounts.

- **1.3.4** Admins shall be able to add users to Black List.
- **1.3.5** Admins shall be able to remove the users from Black List.
- **1.3.6** Admins shall be able to add the content of post, comment, and tags into the spam folder.
- **1.3.7** Admins shall be able to use any search functions that are used by users.

1.4 System Requirements

- **1.4.1 Sign Up**
  - **1.4.1.1** System shall require e-mail confirmation after sign-up.
  - **1.4.1.2** System shall not allow multiple registration with the same e-mail.
- **1.4.2 Login**

    - **1.4.2.1** System shall require a password and a unique username to login.
    - **1.4.2.2** The password shall be at least 8 characters.
    - **1.4.2.3** The password shall consist of alphanumeric characters.
    - **1.4.2.4** The username shall consist of alphanumeric characters.
- **1.4.3 Recommendation**
    - **1.4.3.1** The system should recommend posts and locations to users according to their activity on the system and their current locations.
- **1.4.4 Notifications**
    - **1.4.4.1** The system shall support a notification mechanism.
    - **1.4.4.2** The system shall notify the user if the following events occur via notification box.
        - **1.4.4.2.1** Any activity that happened on a user's post
            - **1.4.4.2.1.1** Post is liked
            - **1.4.4.2.1.2** Post is commented
            - **1.4.4.2.1.2** A comment on the post is replied
        - **1.4.4.2.2** Any activity that happened on a user profile
            - **1.4.4.2.2.1** The user has a new follower
            - **1.4.4.2.2.2** The private user gets following request
            - **1.4.4.2.2.3** The user following request gets accepted

2. Nonfunctional Requirements

- ## 2.1 Portability
    - **2.1.1** The system shall have an Android application and a web page.
- ## 2.2 Accessibility
    - **2.2.1** Language of the Android application and we page shall be English.
- ## 2.3 Performance
    - **2.3.1** The system response should be in 3 second.
    - **2.3.2** The system should handle 500 requests per second.
    - **2.3.3** The system's Android application size should be less than 200 MB.
- ## 2.4 Reliability
    - **2.4.1** The system shall go down for maintenance once a month.
    - **2.4.2** The maintenance shall not take more than 30 minutes.

- **2.4.3** Rest of time, downtime should be less than %1 for failures.

- ## 2.5 Security
  - **2.5.1** The passwords shall be kept encrypted in database.
  - **2.5.2** The web application shall only be accessed by HTTPS protocol.

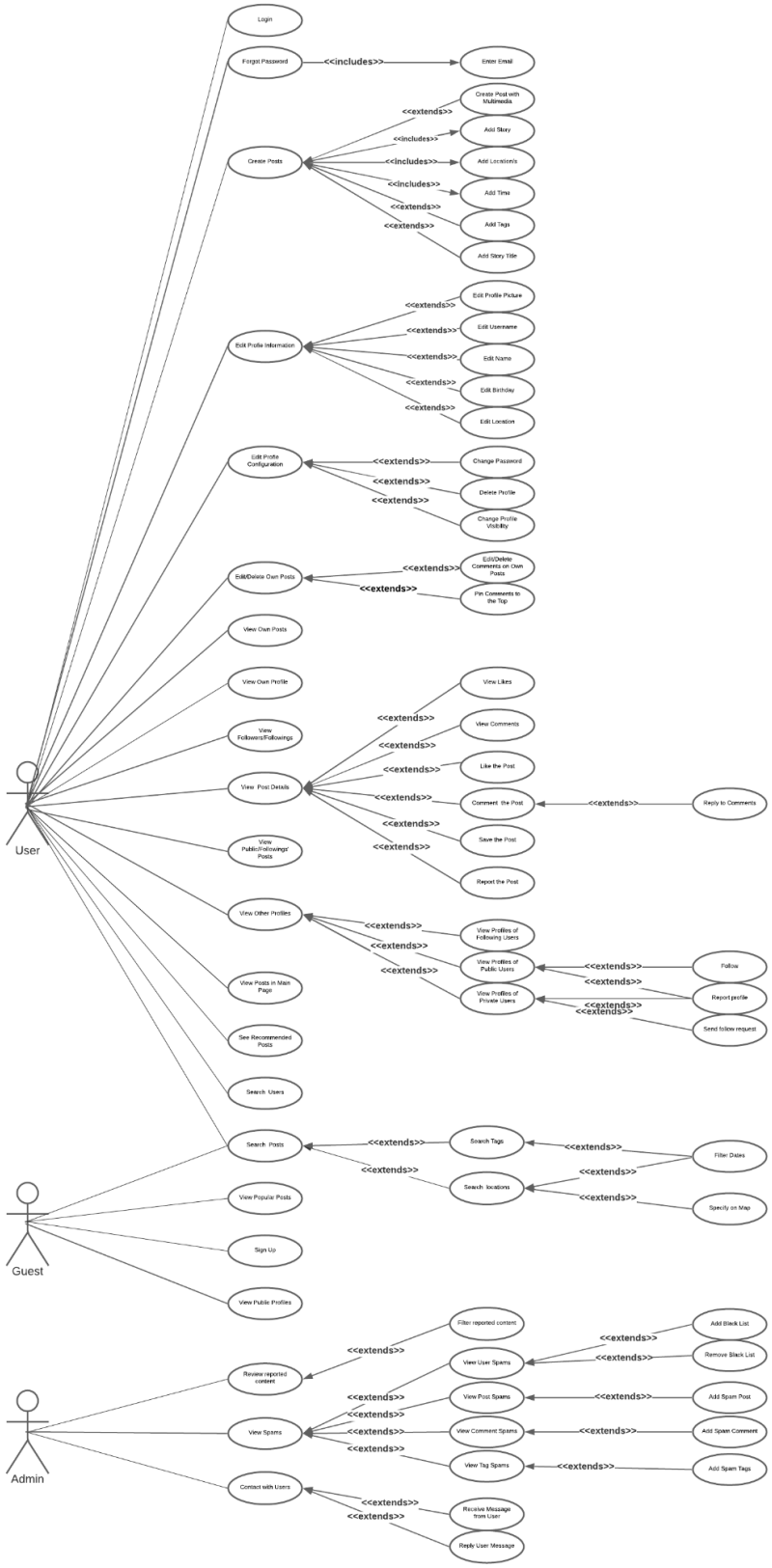
- ## 2.6 Legal and Ethical Issues
  - **2.6.1** Any data collected from users shall be used according to KVKK and GDPR.
  - **2.6.2** New users shall accept "Terms of Use" and "Privacy Policy".

- ## 2.7 Implementation
  - **2.7.1** The system shall use W3C standards and W3C Activity Streams Protocol for subscriptions and notifications.
  - **2.7.2** The tagging system should be supported with Wikidata.org.

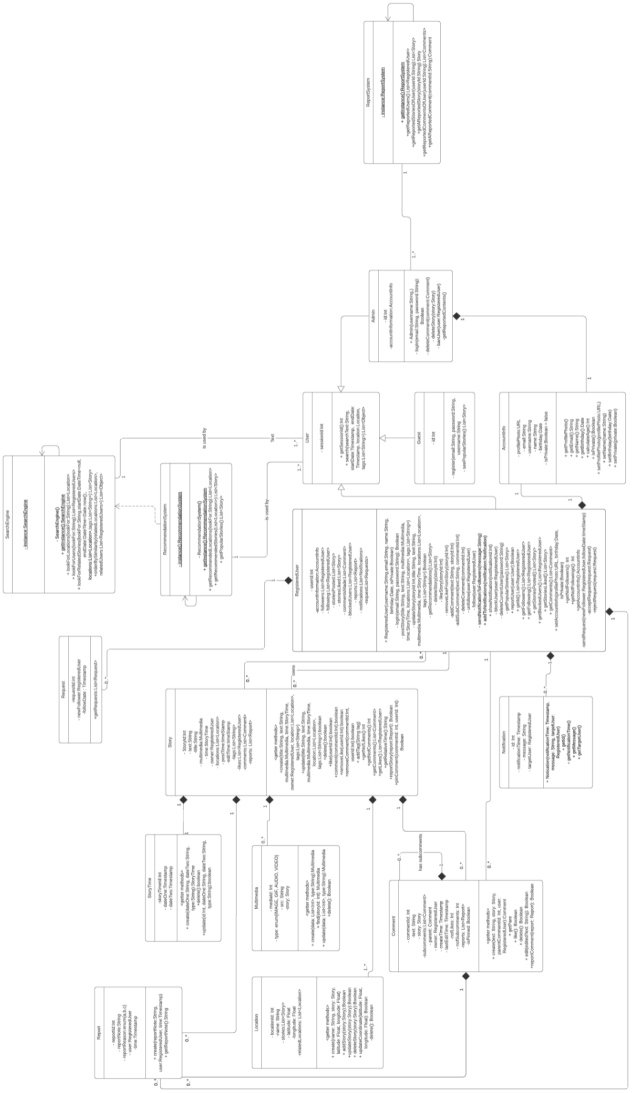
- ## 2.8 Robustness
  - **2.8.1** System should restart the application in 45 minutes after failure.
  - **2.8.2** Server admins should restart application manually in big failures.
  - **2.8.3** Restart after big failures should take up to 4-5 hours.
  - **2.8.4** Less than %2 of events/features should cause system failure.

11. UML Software Designs

11.1. Use Case Diagrams



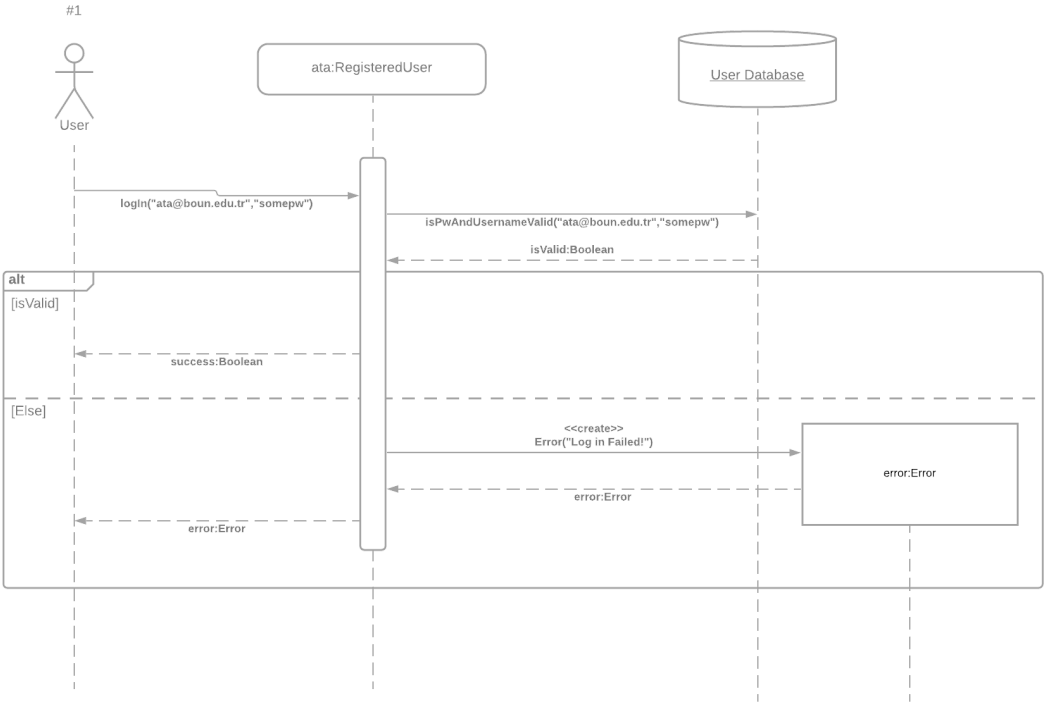
11.2. Class Diagrams



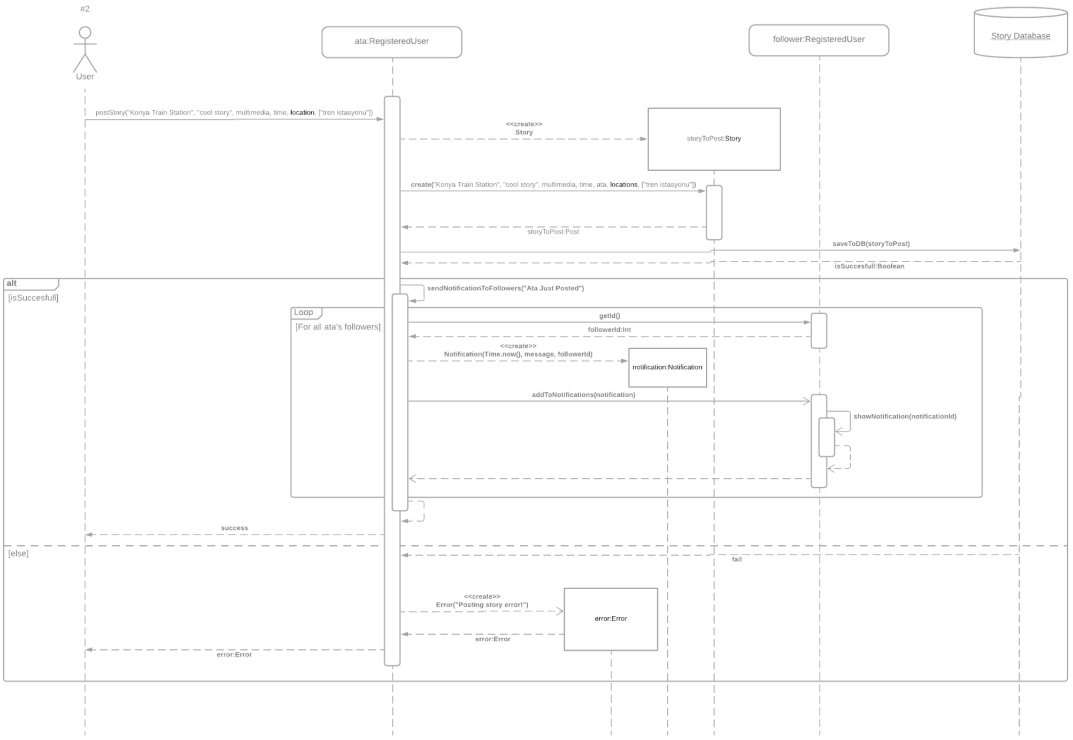
https://lucid.app/lucidchart/68303da9-e502-47b1-93e1-25292586df97/edit?shared=true&page=.Q4MUjXso07N#

11.3. Sequence Diagrams

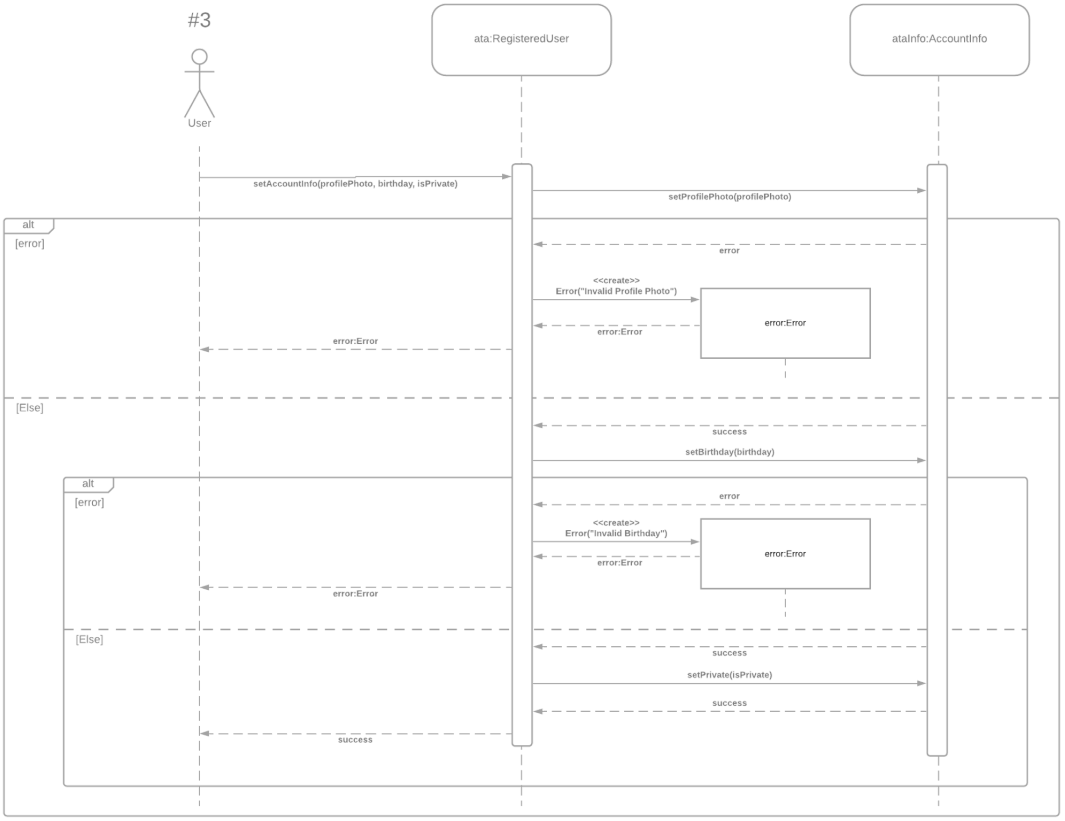
1. Sign In



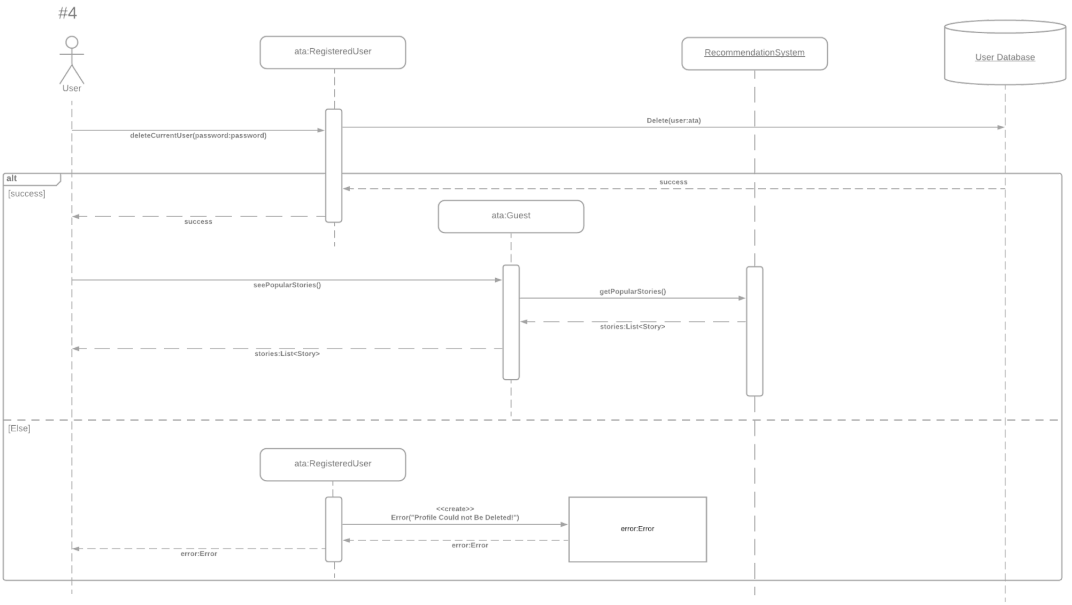
2. Post Creation



3. Edit Profile Information

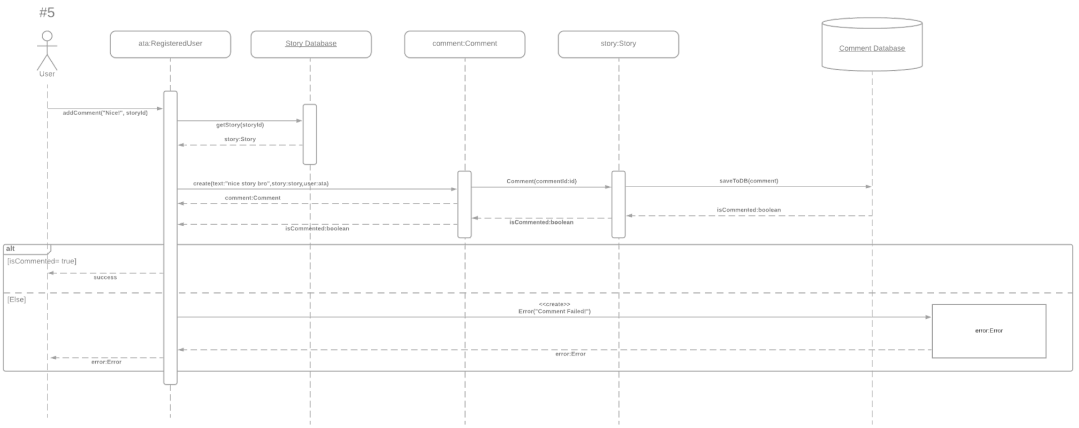


4. Delete Own Profile

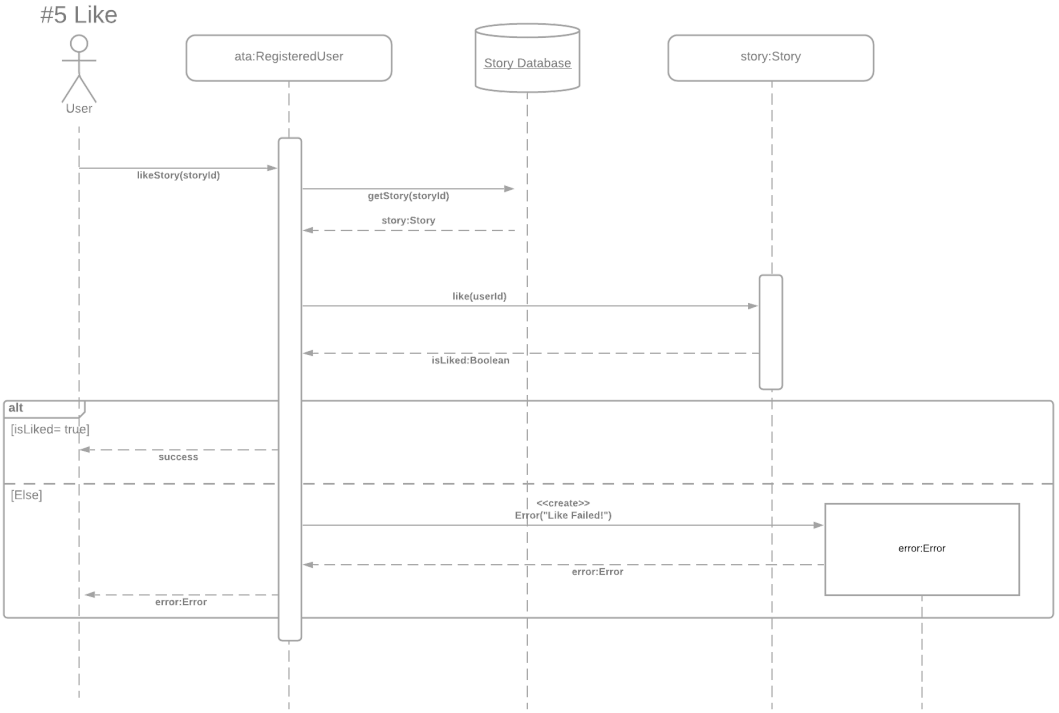


5. Like and Comment a Story

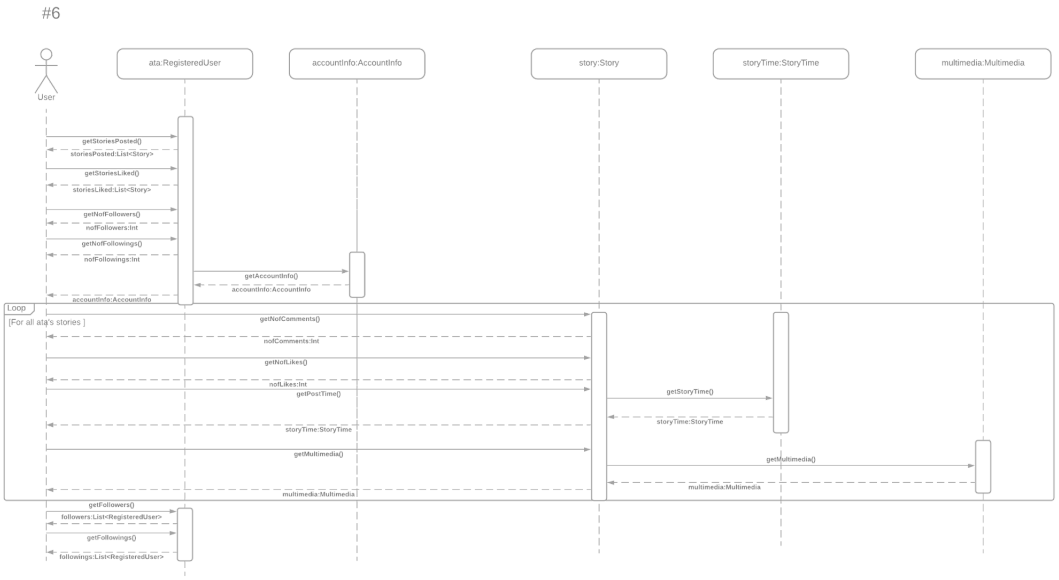
a. User comments on a story in the application.



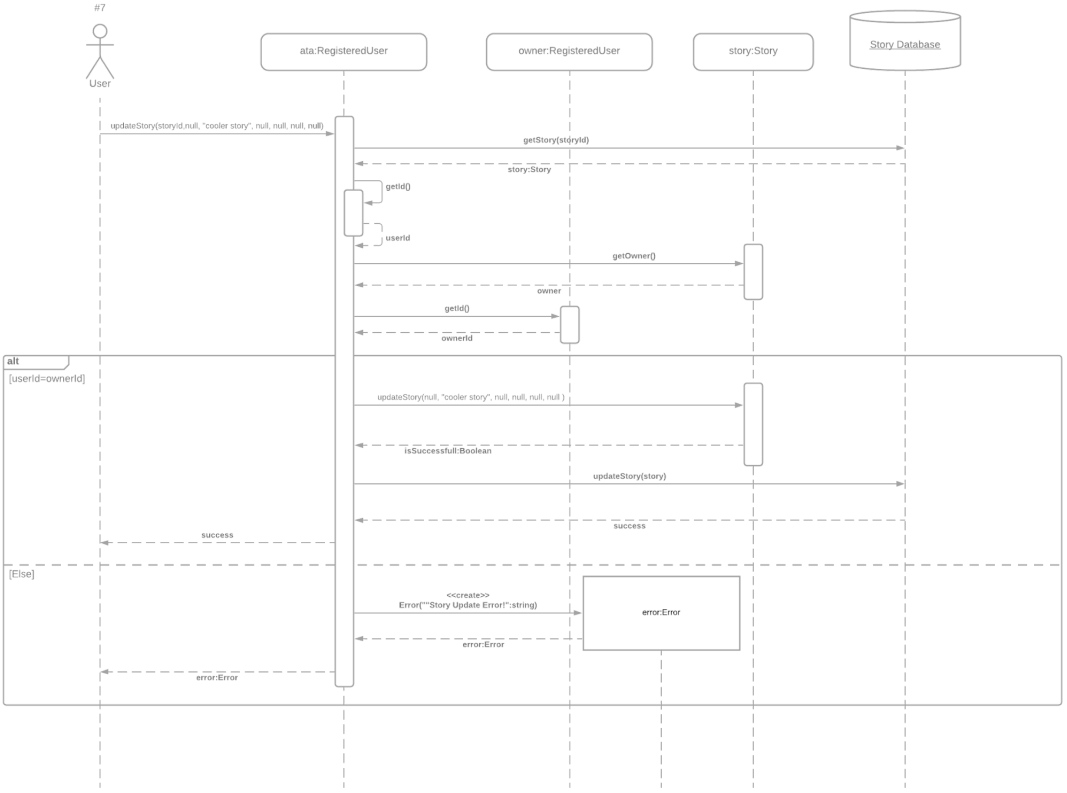
b. User likes a story in the application.



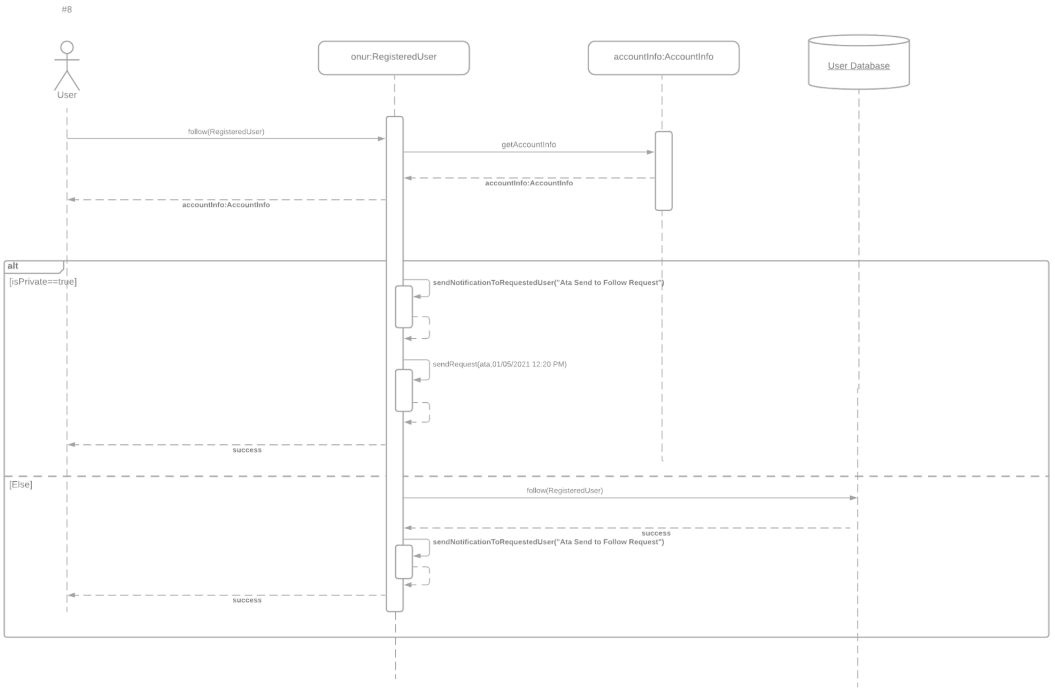
6. View Followers/Followings



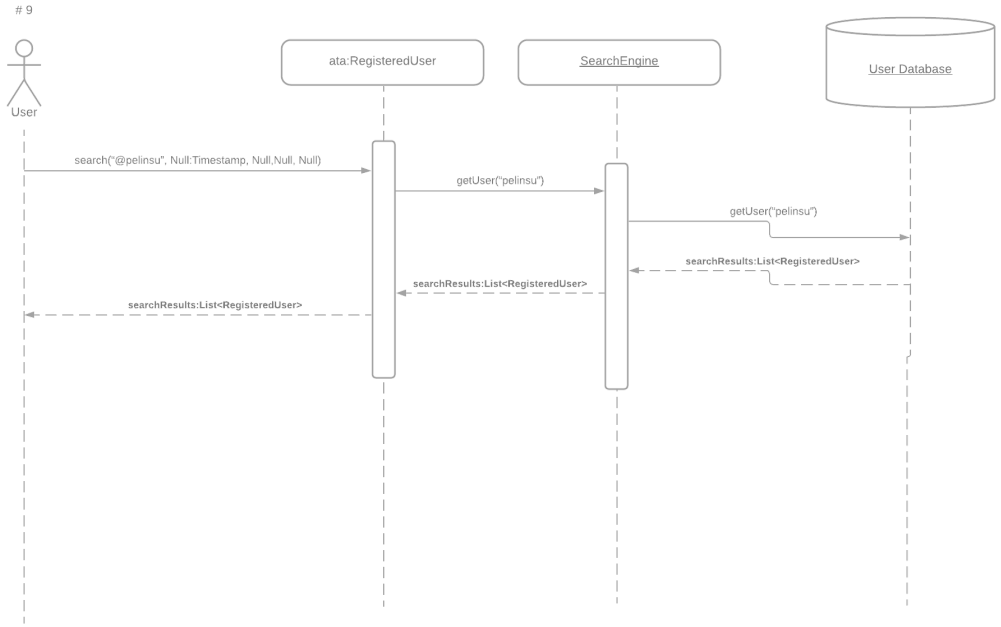
7. Edit Own Post



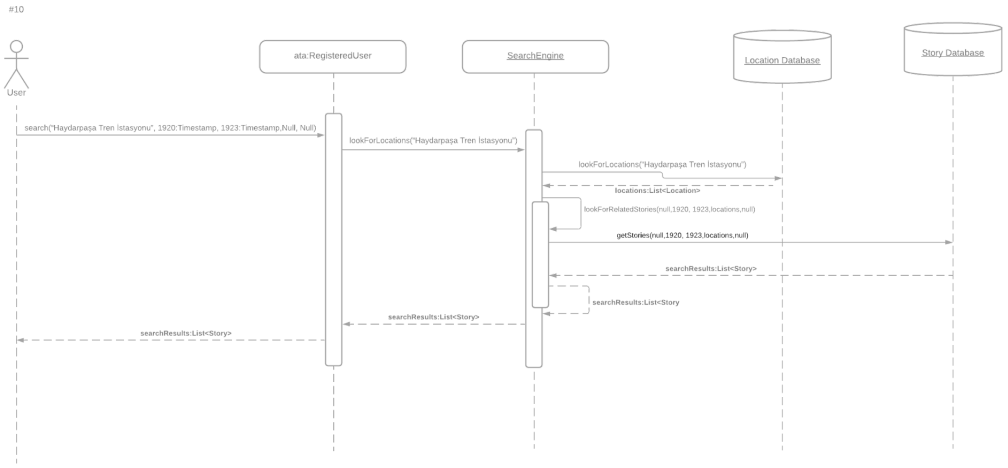
8. Follow Other Users



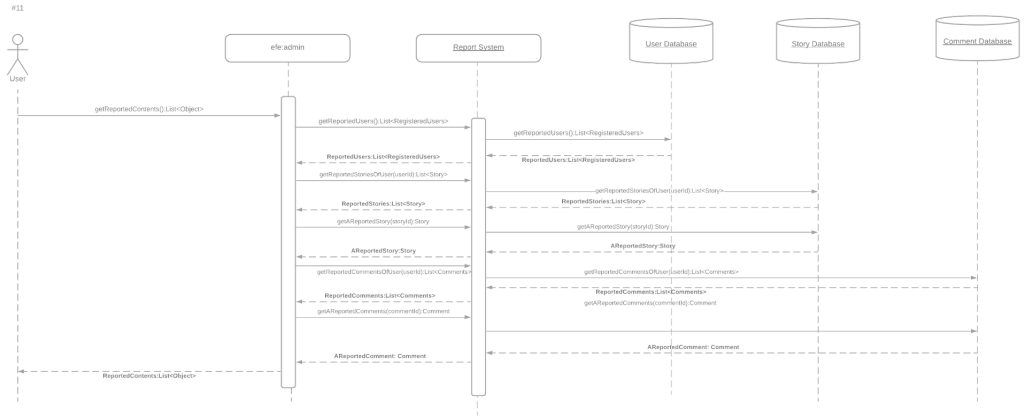
9. Search Other Users



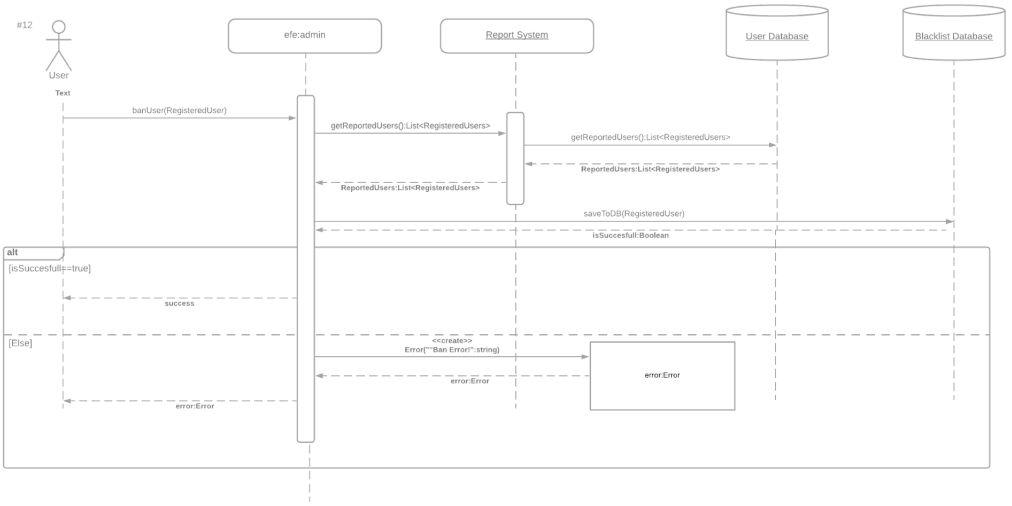
10. Search Posts by Location with Filtering by Date



11. Review Reported Contents



12. Add Someone to Blacklist



1. Guest Scenario
    a. Guest in Android Scenario

Persona

- Pelinsu Tilki is a third year university student and she studies Computer Engineering.
- She is 23 years old.
- She likes music and dancing especially salsa.
- She liked going clubs with her friends but she could hardly find time for it because of the work load of the courses.

Story

- Pelinsu is a computer engineering student at Bogazici University.
- Due to the pandemic, she generally stays home with her family in İzmir.
- She can not stay in İstanbul for long since she does not have a accomodation place.
- She misses her friends and school too much.
- She wants to feel like she is in South Campus by reading some recent stories taking place in South Campus.

Preconditions

- She is not a currently registered user of the application.

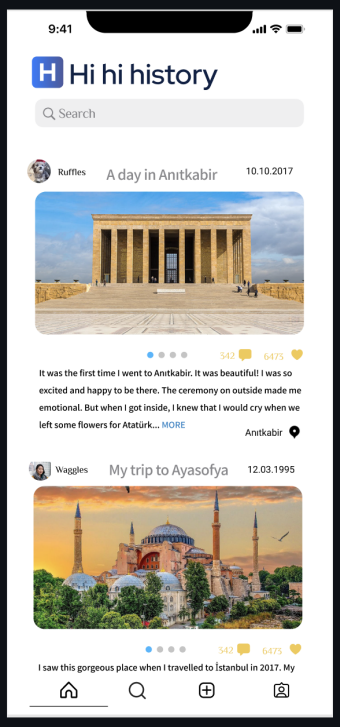
Goals

- She wants to do a search to see the posts taking place in Albert Long Hall.
- She wants to read a recent story about Albert Long Hall.
- She wants to see photos of Albert Long Hall since she misses her school in the pandemic.

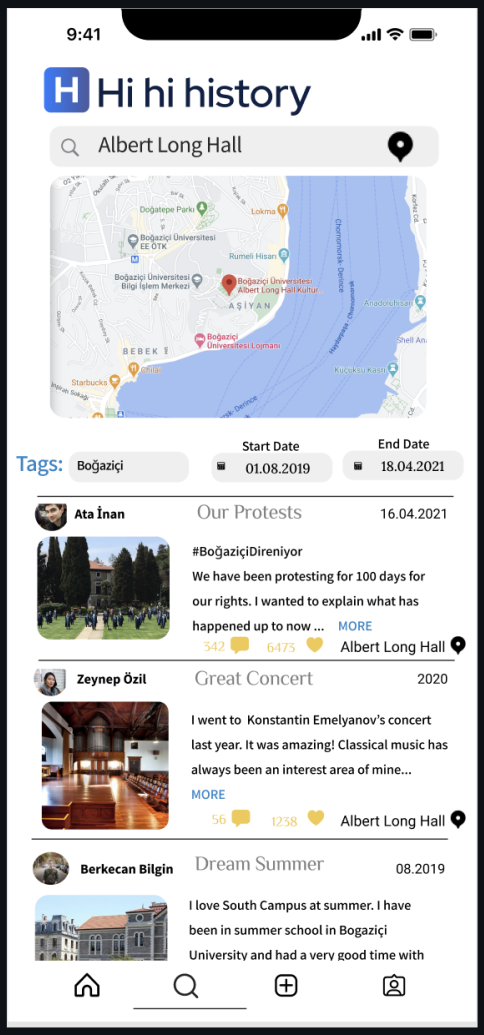
Acceptance Criteria

- **1.2.1** Guests shall be able to create a new account by choosing a password (email confirmation) and providing email, username (name, location, birthday optionally).
- **1.2.2** Guests shall be able to login to the system by entering username and password if they already have an account.
- **1.2.3** Guests shall be able to search and view locations, shall not be able to search or view users.

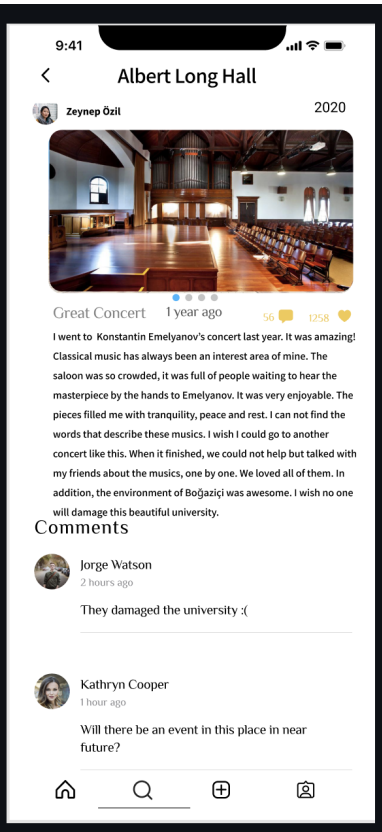
Scenario

- 

1. She enters the application without registering.
2. She comes to the main page.
3. She sees a search bar at the top.\



4. She clicks to the search bar.
5. She writes "Albert Long Hall".
6. She clicks the search button.\

- 

7. She specifies a tag named "Boğaziçi".
8. She sees posts about “Albert Long Hall” in Boğaziçi in chronological order.
9. She chooses a post to see it in more detail.\
10. She scrolls through the photos in the post.
11. She tries to write a comment to that however a screen appears, which requires her to sign in to application.

b. Guest in Web Scenario

Persona

- Pelinsu Tilki is a third year university student and she studies Computer Engineering.
- She is 23 years old.
- She likes music and dancing especially salsa.
- She liked going clubs with her friends but she could hardly find time for it because of the work load of the courses.

Story

- Pelinsu is a computer engineering student at Bogazici University.
- Due to the pandemic, she generally stays home with her family in İzmir.
- She can not stay in İstanbul for long since she does not have a accomodation place.
- She misses her friends and school too much.
- She wants to feel like she is in South Campus by reading some recent stories taking place in South Campus.

Preconditions

- She is not a currently registered user of the application.

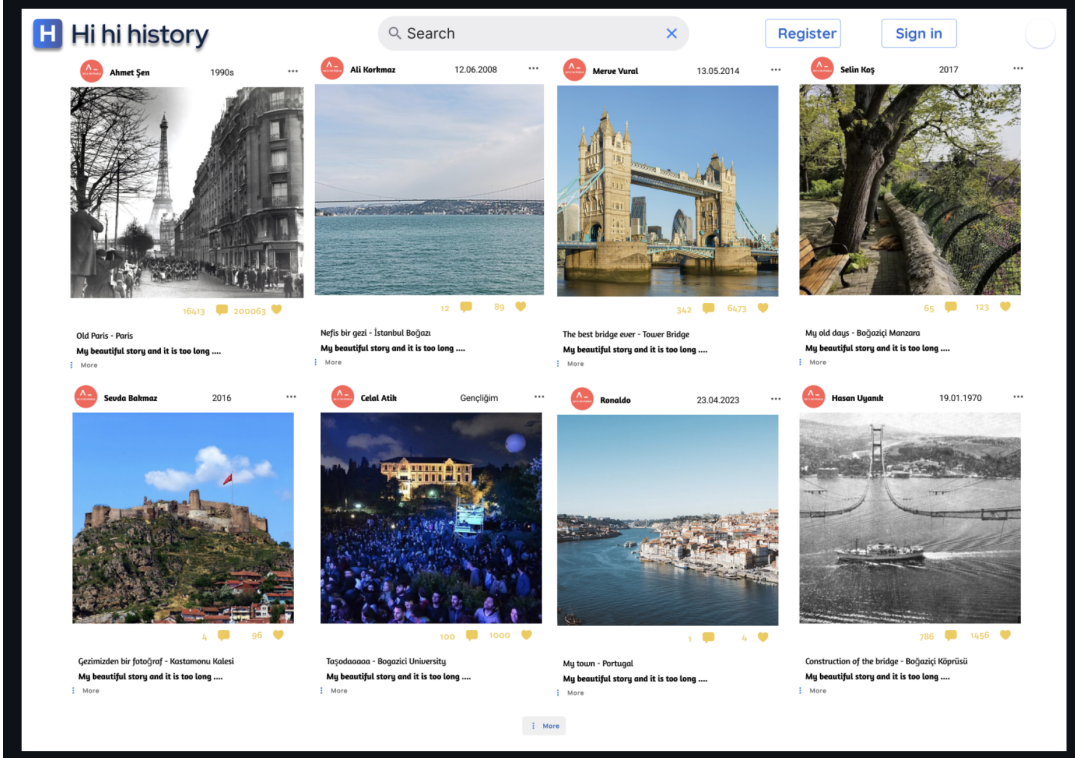
Goals

- She wants to do a search to see the posts taking place in Albert Long Hall.
- She wants to read a recent story about Albert Long Hall.
- She wants to see photos of Albert Long Hall since she misses her school in the pandemic.

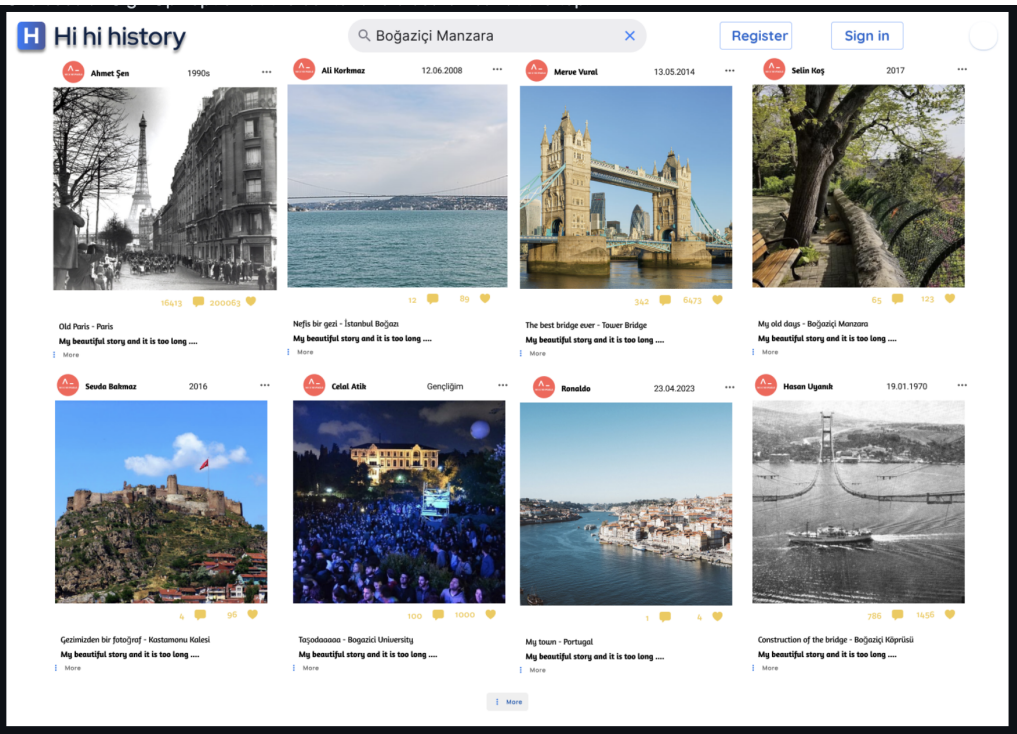
Acceptance Criteria

- **1.2.1** Guests shall be able to create a new account by choosing a password (email confirmation) and providing email, username (name, location, birthday optionally).
- **1.2.2** Guests shall be able to login to the system by entering username and password if they already have an account.
- **1.2.3** Guests shall be able to search and view locations, shall not be able to search or view users.

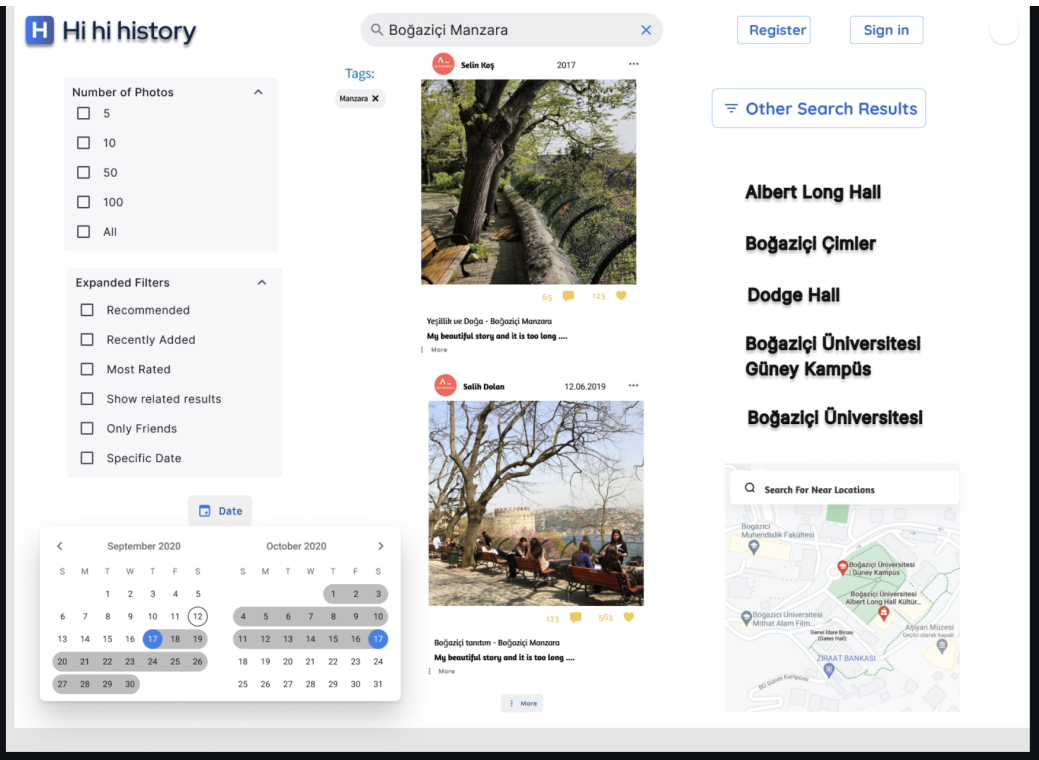
Scenario



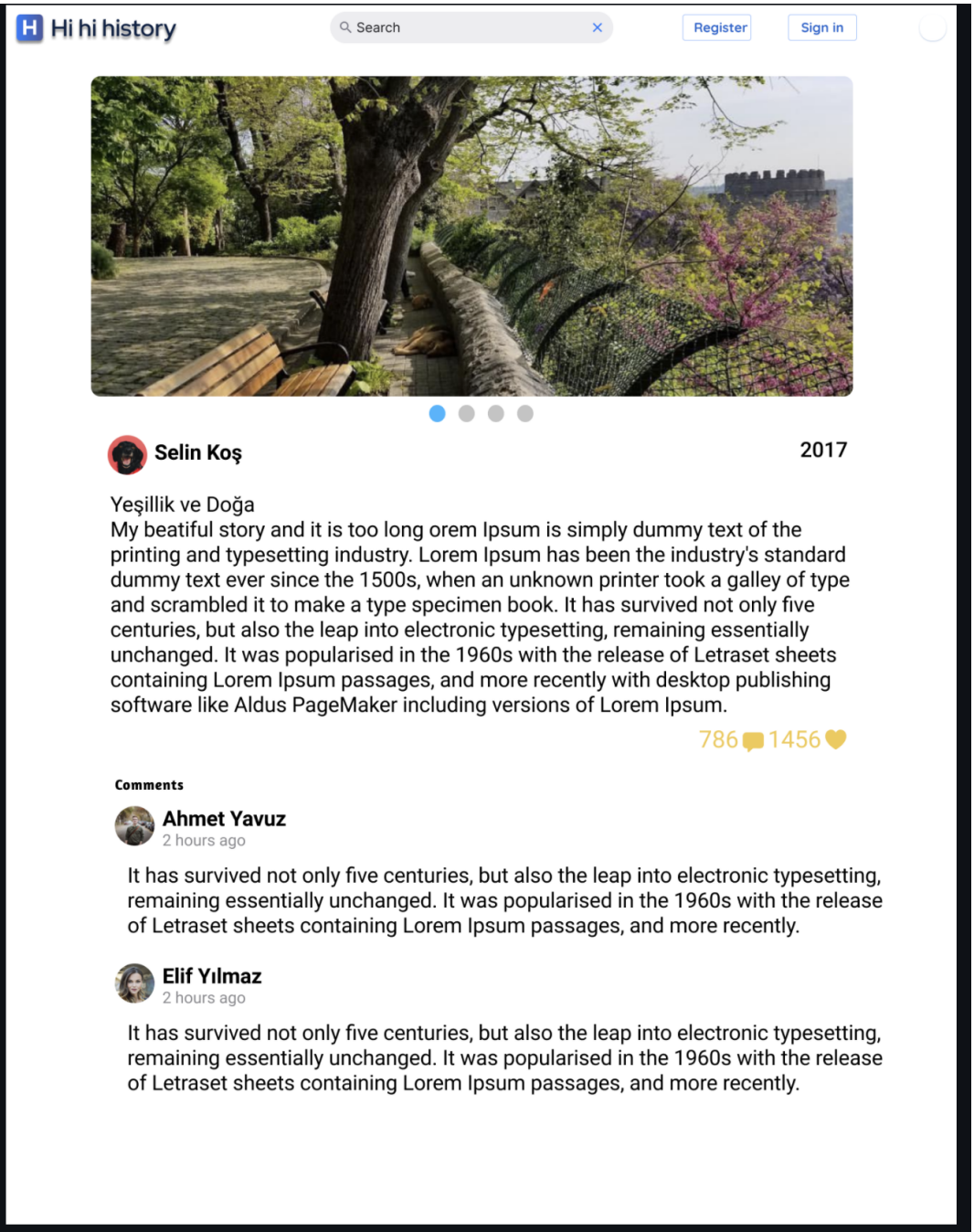
1. She enters the application without registering.
2. She comes to the main page.
3. She sees a “Sign Up” option at the center and a search bar at the top.



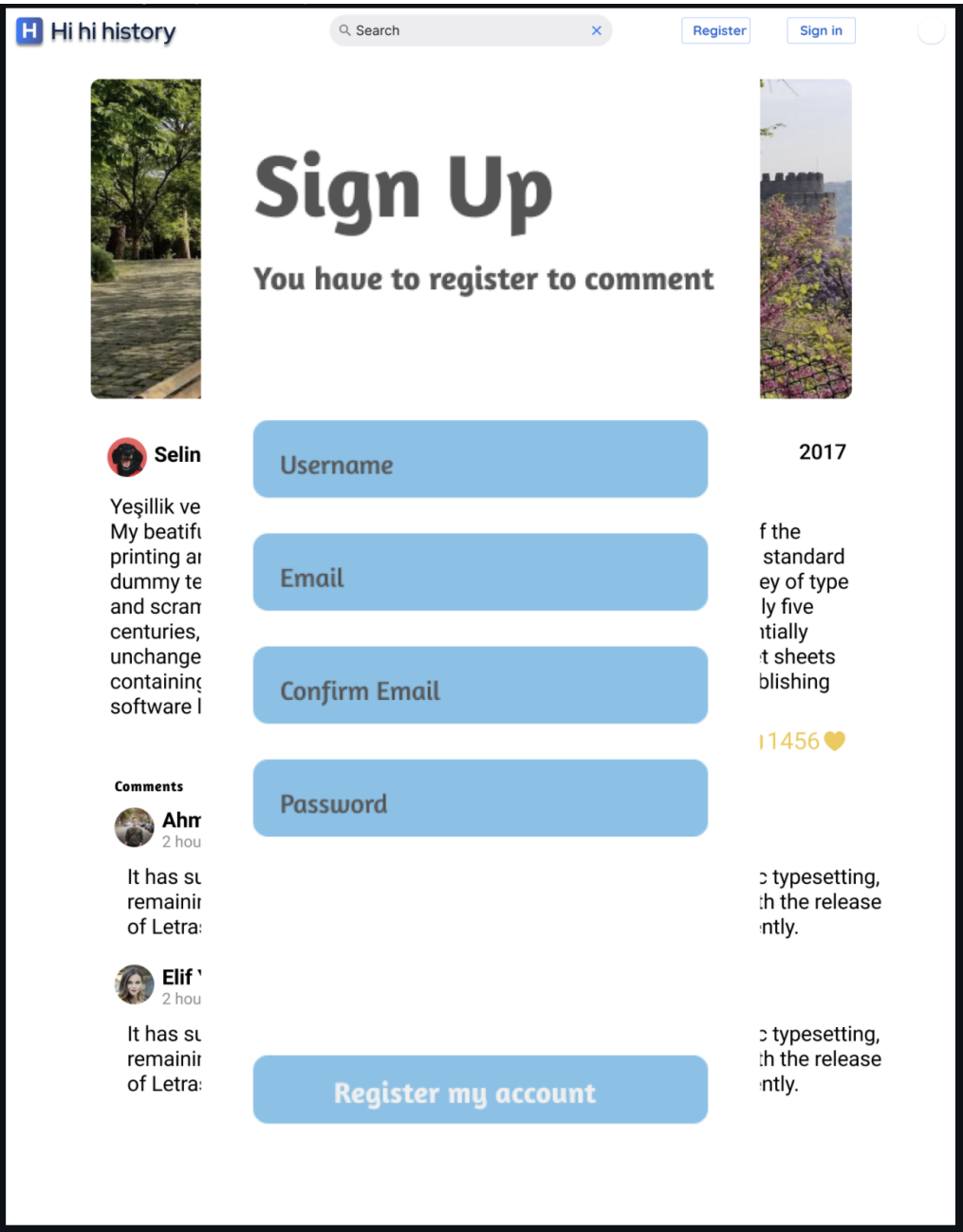
4. She clicks to the search bar.
5. She writes “Boğaziçi Manzara”.



6. She clicks the search button.
7. From the search results, she selects the first option which corresponds to “Boğaziçi Manzara” in Boğaziçi University.



8. She sees posts about “Boğaziçi Manzara” in chronological order.
9. She chooses a post to see it in more detail.
10. She scrolls through the photos in the post.



11. She tries to write a comment to that however a screen appears, which requires her to sign in to application.

2. User Scenario
    a. User in Android Scenario
        i. User Sharing & Editing Scenario

User

- Salih YILMAZ

Story

Salih will be retired in a few years. Lately he has become nostalgic by thinking all the years that he worked as a railroad officer. He remembers a close friend of his who runned a restaurant in Konya Train Station, also where he worked for 5 years in 90's. He wants to tell his friend's contribution to the Konya Train Station to other people.

Precondition

- He is already logged in to the application.
- He posted some stories and photos before.
- He is following other users.

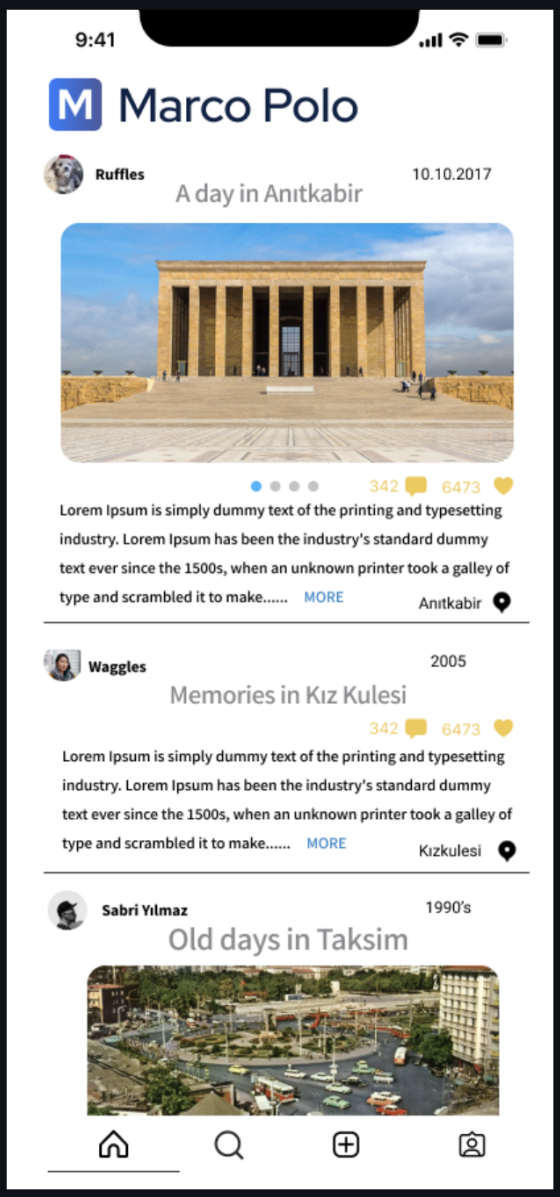
Goals

- He wants to share a story and photo from the 90's about Konya station.
- He wants to edit the post that he shared to add new information or to fix typos.

Acceptance Criteria

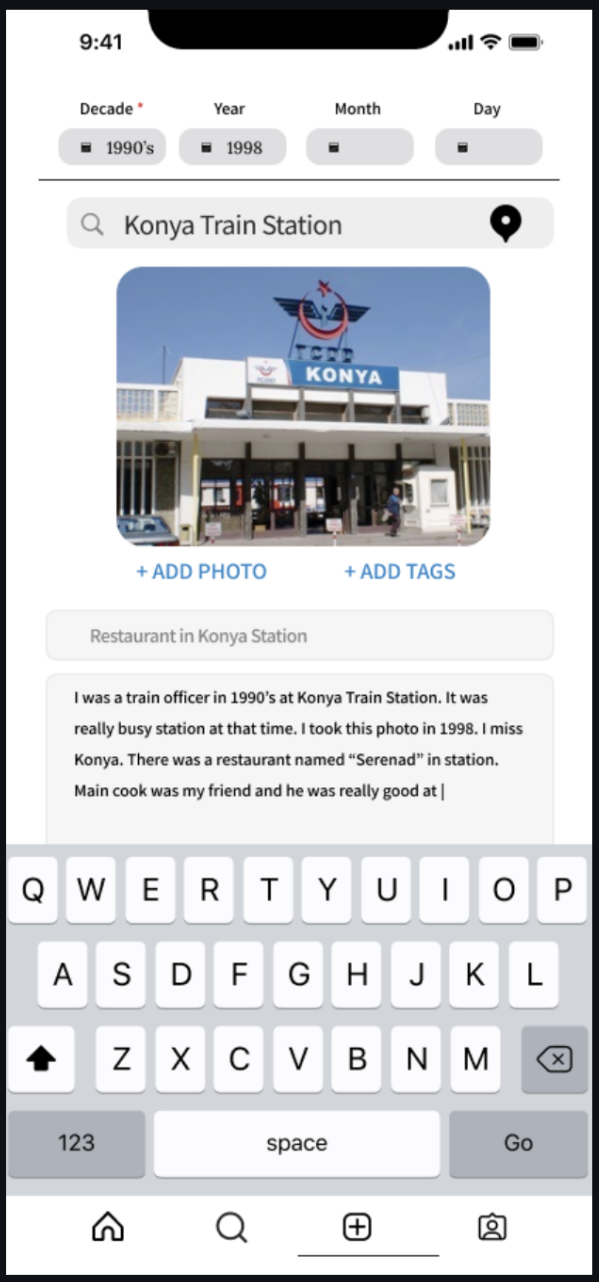
- **1.1.1.4** Users shall be able to view the posts that they posted.
- **1.1.2.1** Users shall be able to create posts with a photo or video.
- **1.1.2.2** Users shall be able to add tags and stories to their posts.
- **1.1.2.3** Users shall be able to specify time in their posts to declare when the photo or video in the post is filmed.
- **1.1.2.4** Users shall be able to edit and delete their own posts.

Scenario

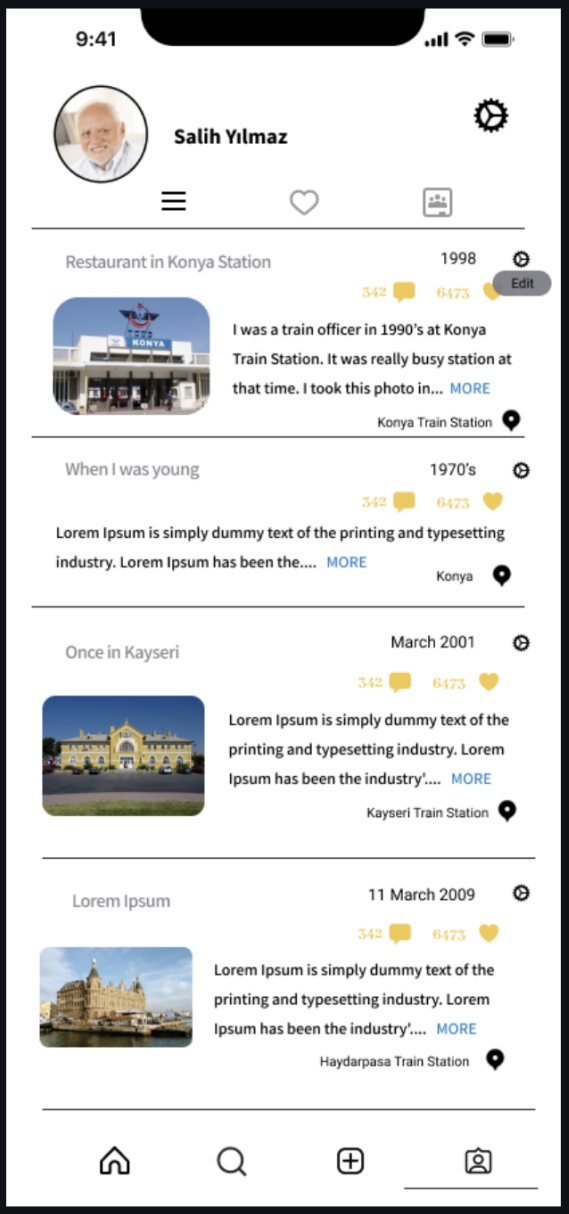


1. He is already logged in.
2. He is at the main page.
3. He sees the recent posts about the people and places he followed.

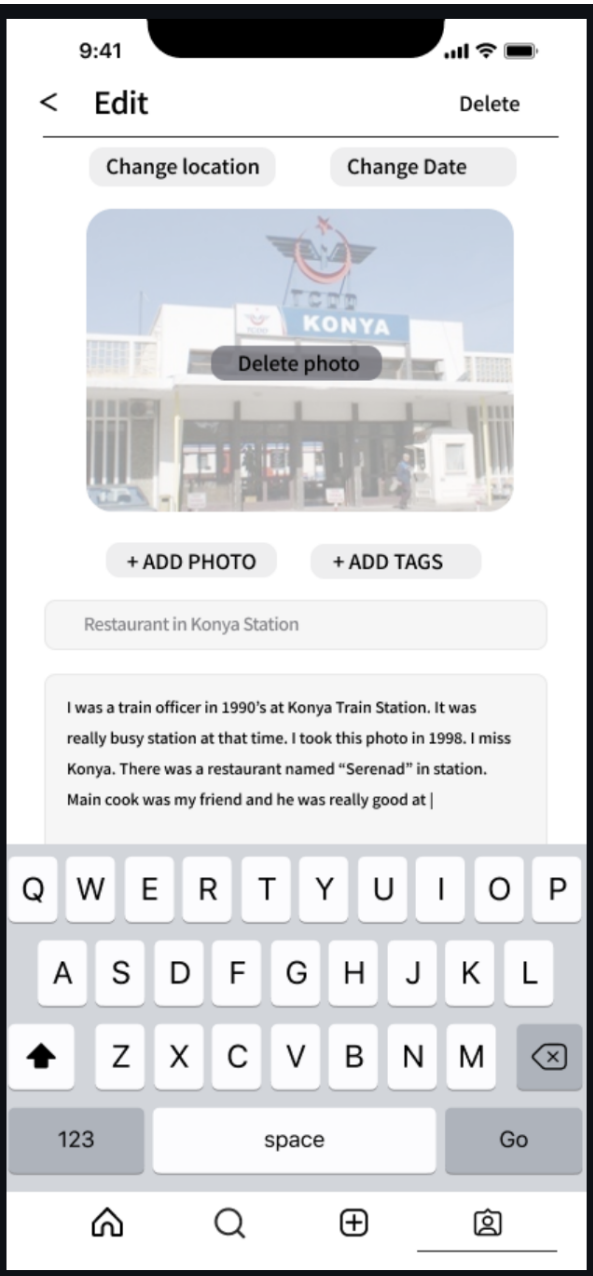
3. He sees the recent posts about the people and places he followed.



4. He clicks to share a new post button.
5. He writes a new story about the Konya Train Station.
6. He adds a photo, date, location and tags describing the story.
7. Then he posts the story.



- **8.** He remembers an additional info about the story that he posted about Konya Train Station.
9. He goes to his profile page.
10. He selects the edit button on the post about Konya Train Station.



11. He adds additional information that he remembered.
12. He saves the post.

ii. User Searching, Filtering & Interaction Senario

User

- Salih YILMAZ

Story

Salih has met with his close friend recently. While talking about interesting things that they witnessed, his friend mentioned a cat he saw in Haydarpasa Train Station in his last visit to Istanbul. Salih wants to learn this cat's story in Haydarpasa Train Station.

Precondition

- He is already logged in to the application.
- He posted some stories and photos before.
- He is following other users.

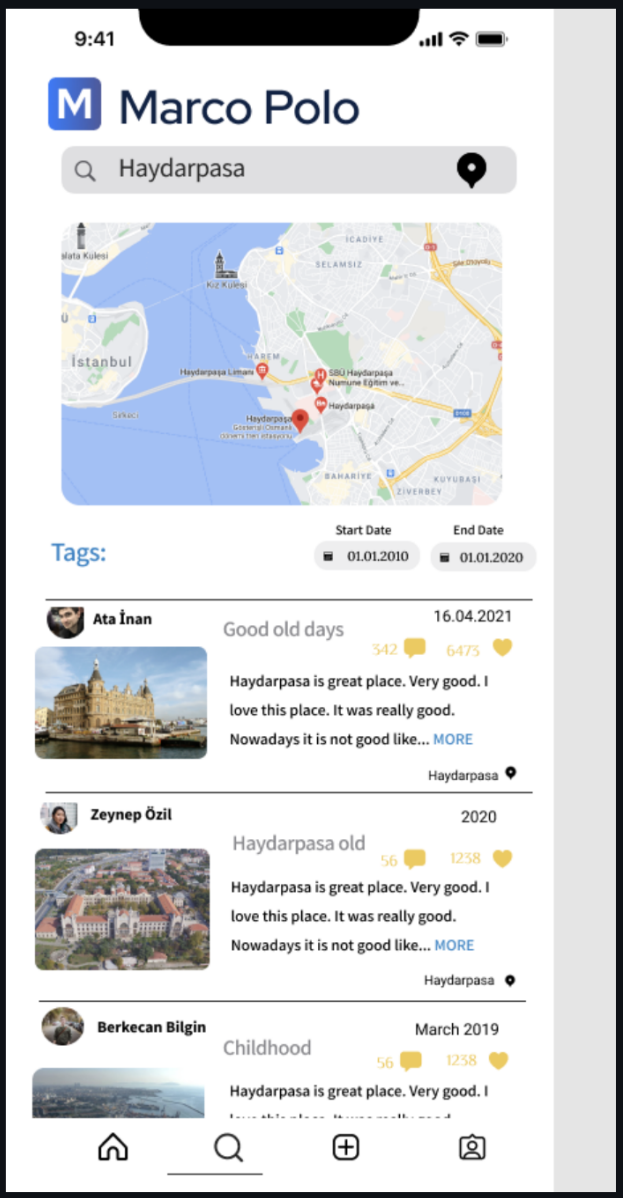
Goals

- He wants to search for cat stories of Haydarpasa Train Station at 2010's and view top stories.
- He wants to contribute a story by commenting and liking.

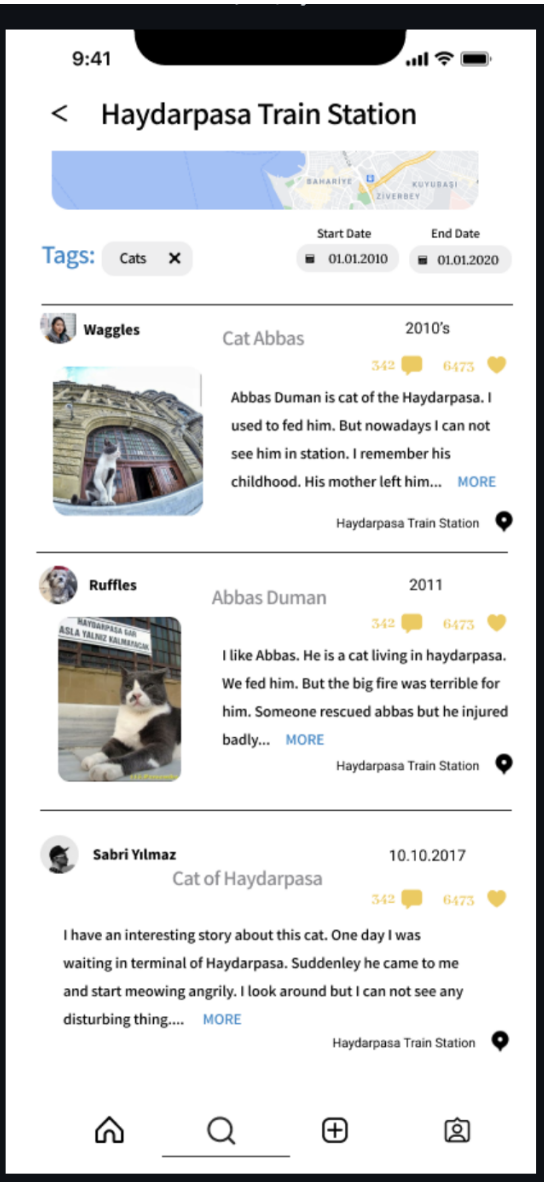
Acceptance Criteria

- **1.1.3.12** Users should be able to see the liked posts and comments of other users that they follow or users that have public profile
- **1.1.4.1** Users shall be able search locations by their name.
- **1.1.4.2** Users shall be able to do semantic search by using tags.
- **1.1.4.3** Users shall be able to filter the posts that were shared from a location by time.

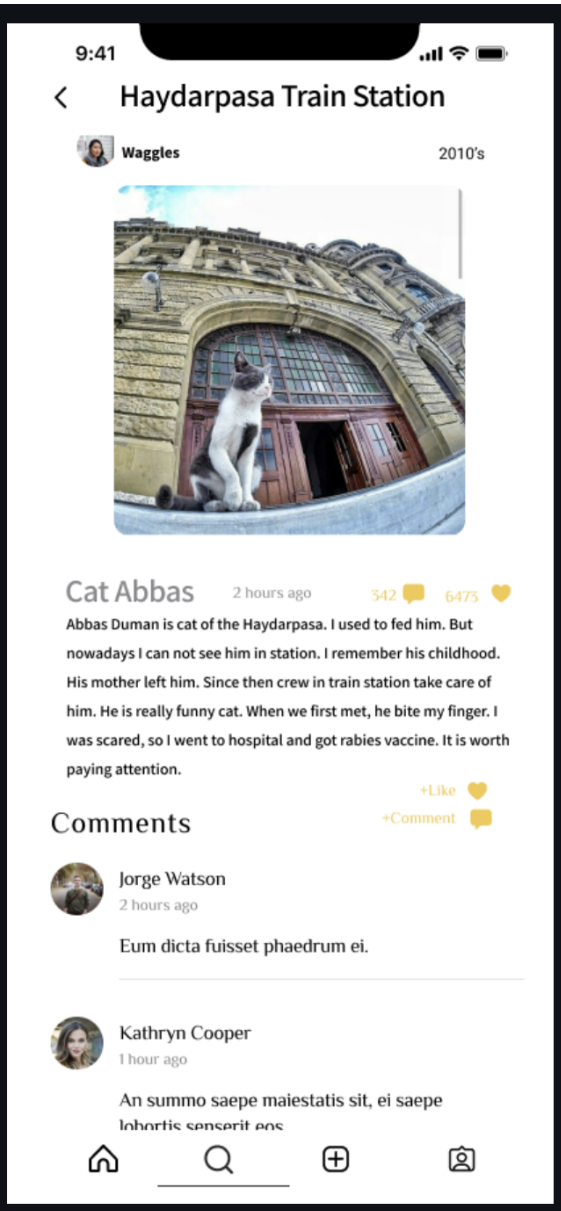
Scenario



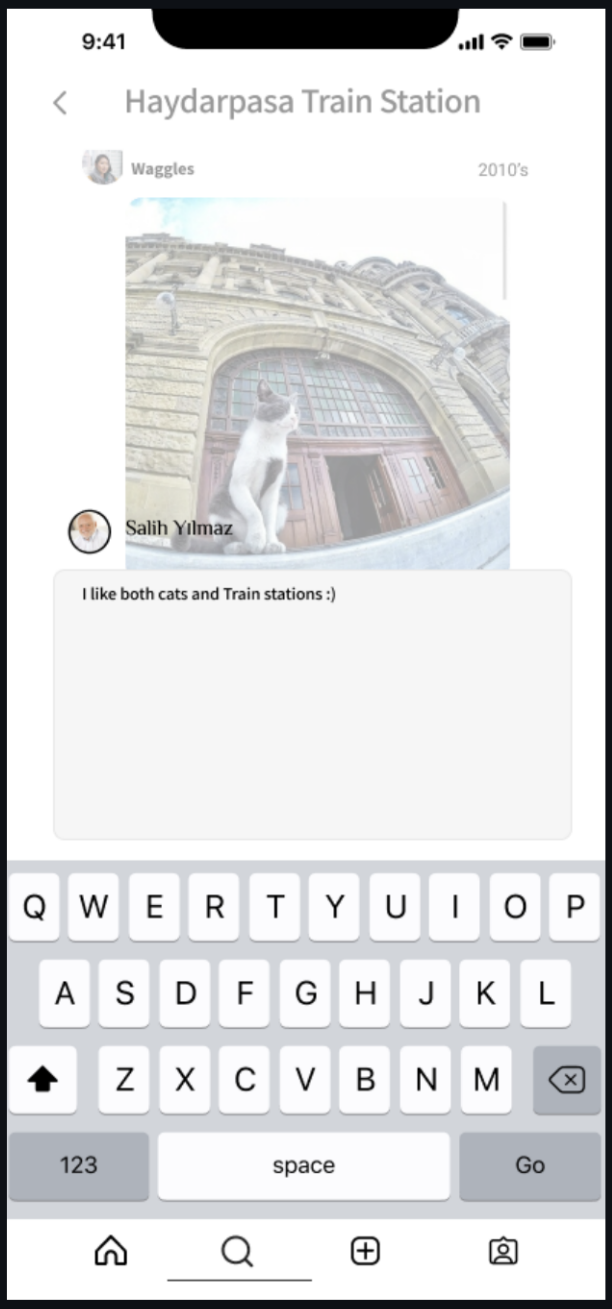
1. He is already logged in.
2. He goes to the search page.
3. He searches about Haydarpaşa.



- **4.** He searches about Haydarpaşa train station to read about the stories of the past.
5. He filters the posts to the 2010's and filters the tags to cats.



- 6. He sees the top liked posts from the 2010's which are about cats in Haydarpaşa Train Station.
7. He selects the top liked post about a cat named "Cat Abbas".
8. He likes the story.



- **9.** He comments on the story.

iii. User Recommendation Scenario

User

- Salih YILMAZ

Story

Salih has worked in train stations for about 30 years and he has a lots of memories from these places. He is quite curious about other stations' stories all around the world and memories that people saved in those stations.

Precondition

- He is already logged in to the application.
- He posted some stories and photos before.
- He is following other users.

Goals

- He wants to see recommended places to explore the stories about these places which are recommended by his interest.

Acceptance Criteria

- **1.1.5.1** Users shall be able to view recommended locations based on their system activity and current location.

b. User in Web Scenario

i. User Sharing & Editing Scenario

User

- Salih YILMAZ

Story

Salih will be retired in a few years. Lately he has become nostalgic by thinking all the years that he worked as a railroad officer. He remembers a close friend of his who runned a restaurant in Konya Train Station, also where he worked for 5 years in 90's. He wants to tell his friend's contribution to the Konya Train Station to other people.

Precondition

- He is already logged in to the application.
- He posted some stories and photos before.
- He is following other users.

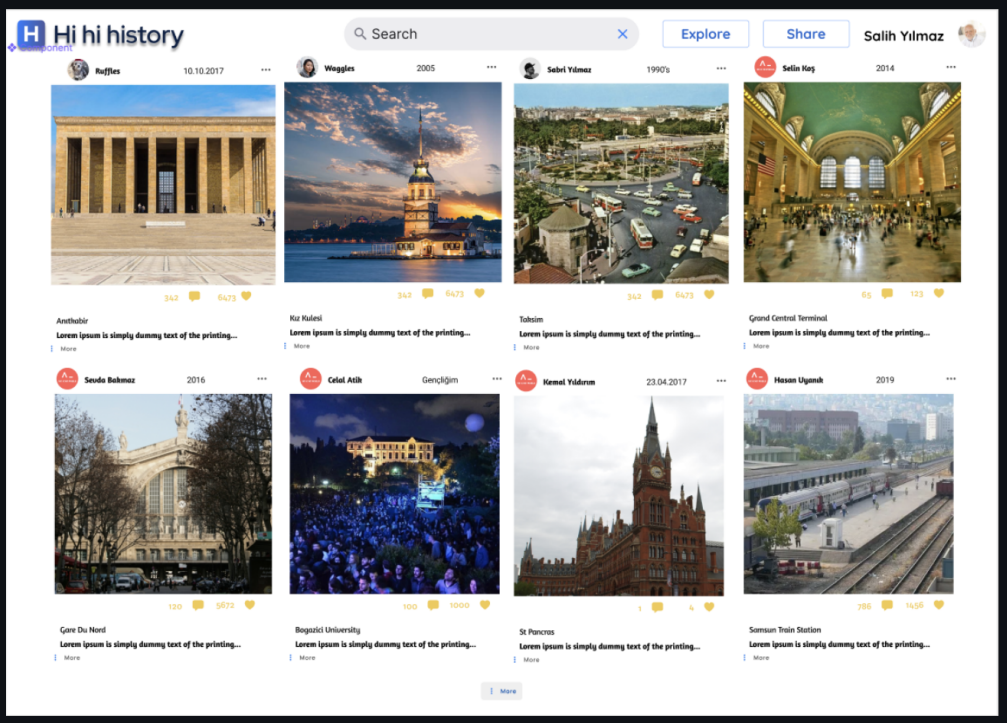
Goals

- He wants to share a story and photo from the 90's about Konya station.
- He wants to edit the post that he shared to add new information or to fix typos.

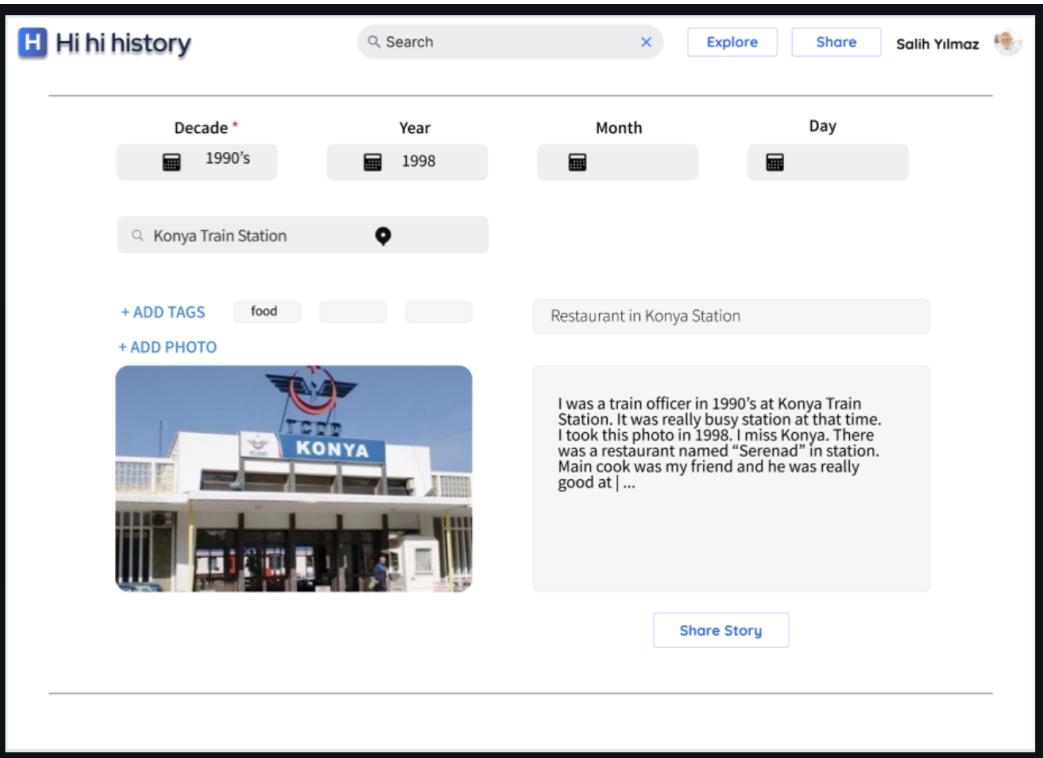
Acceptance Criteria

- **1.1.1.4** Users shall be able to view the posts that they posted.
- **1.1.2.1** Users shall be able to create posts with a photo or video.
- **1.1.2.2** Users shall be able to add tags and stories to their posts.
- **1.1.2.3** Users shall be able to specify time in their posts to declare when the photo or video in the post is filmed.
- **1.1.2.4** Users shall be able to edit and delete their own posts.

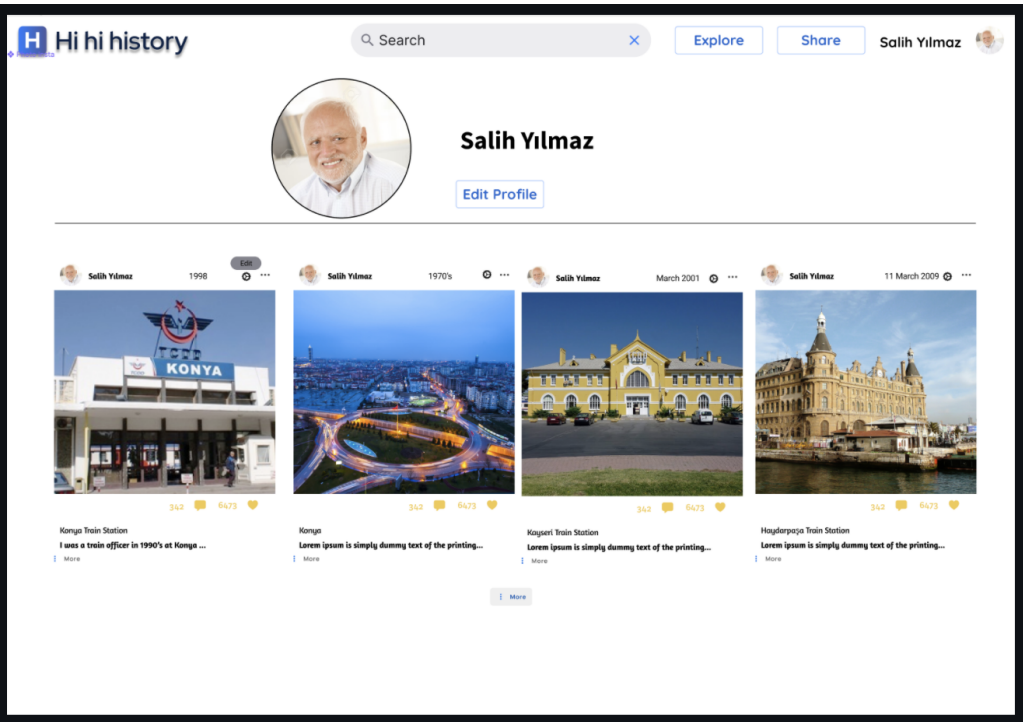
Scenario



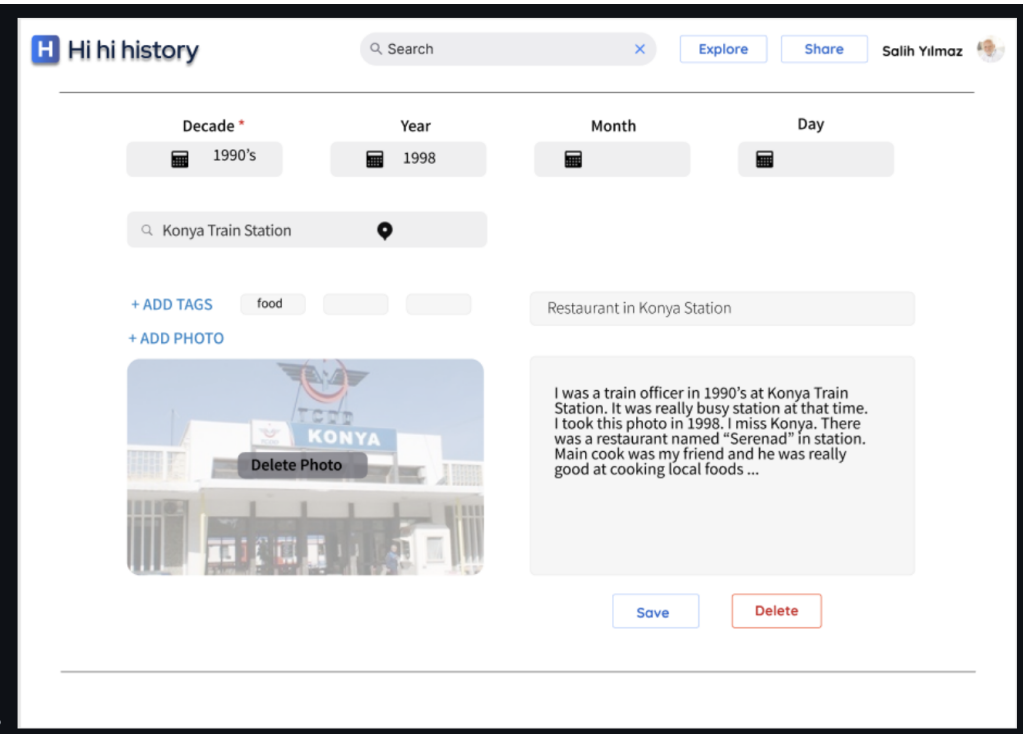
1. He is already logged in.
2. He is at the main page.
3. He sees the recent posts about the people and places he followed.



4. He clicks to share a new post button.
5. He writes a new story about the Konya Train Station.
6. He adds a photo, date, location and tags describing the story.
7. Then he posts the story.



8. He remembers an additional info about the story that he posted about Konya Train Station.
9. He goes to his profile page.
10. He selects the edit button on the post about Konya Train Station.



11. He adds additional information that he remembered.
12. He saves the post.

ii. User Searching, Filtering & Interaction Senario

User

- Salih YILMAZ

Story

Salih has met with his close friend recently. While talking about interesting things that they witnessed, his friend mentioned a cat he saw in Haydarpasa Train Station in his last visit to Istanbul. Salih wants to learn this cat's story in Haydarpasa Train Station.

Precondition

- He is already logged in to the application.
- He posted some stories and photos before.
- He is following other users.

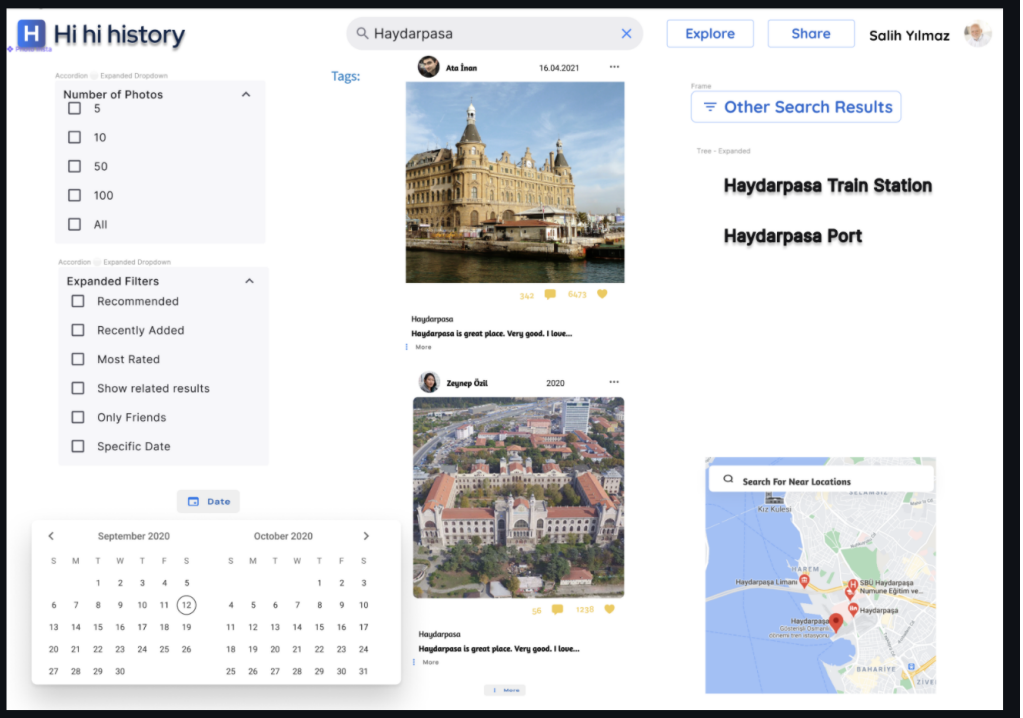
Goals

- He wants to search for cat stories of Haydarpasa Train Station at 2010's and view top stories.
- He wants to contribute a story by commenting and liking.

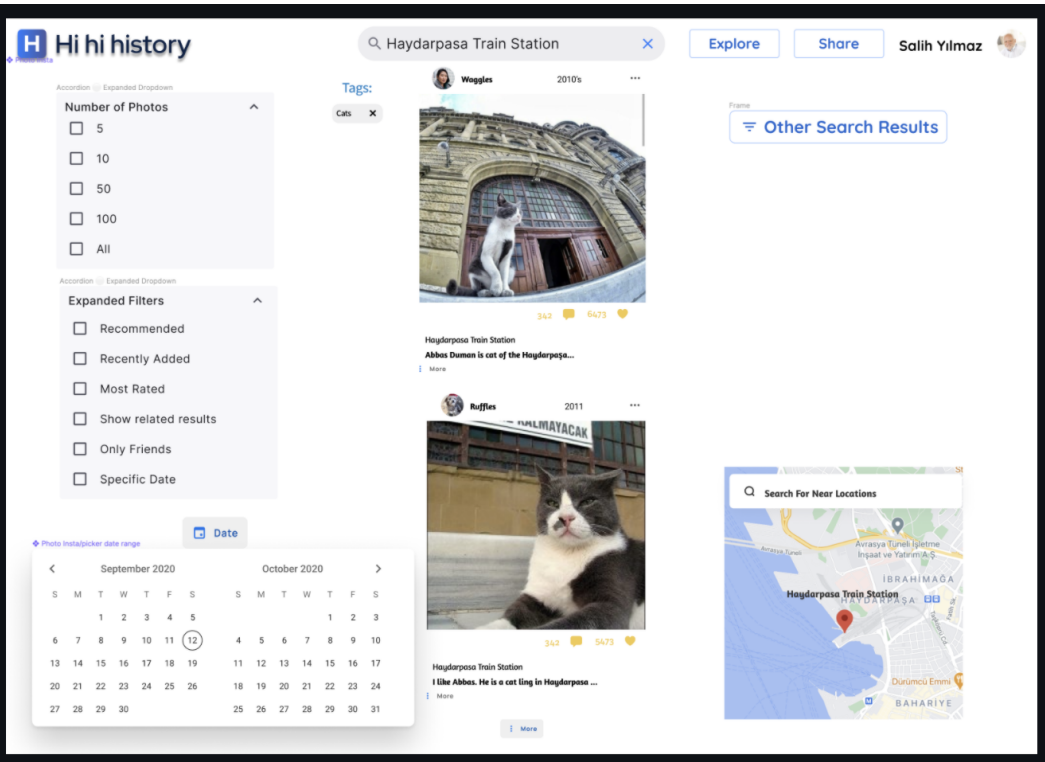
Acceptance Criteria

- **1.1.3.12** Users should be able to see the liked posts and comments of other users that they follow or users that have public profile
- **1.1.4.1** Users shall be able search locations by their name.
- **1.1.4.2** Users shall be able to do semantic search by using tags.
- **1.1.4.3** Users shall be able to filter the posts that were shared from a location by time.

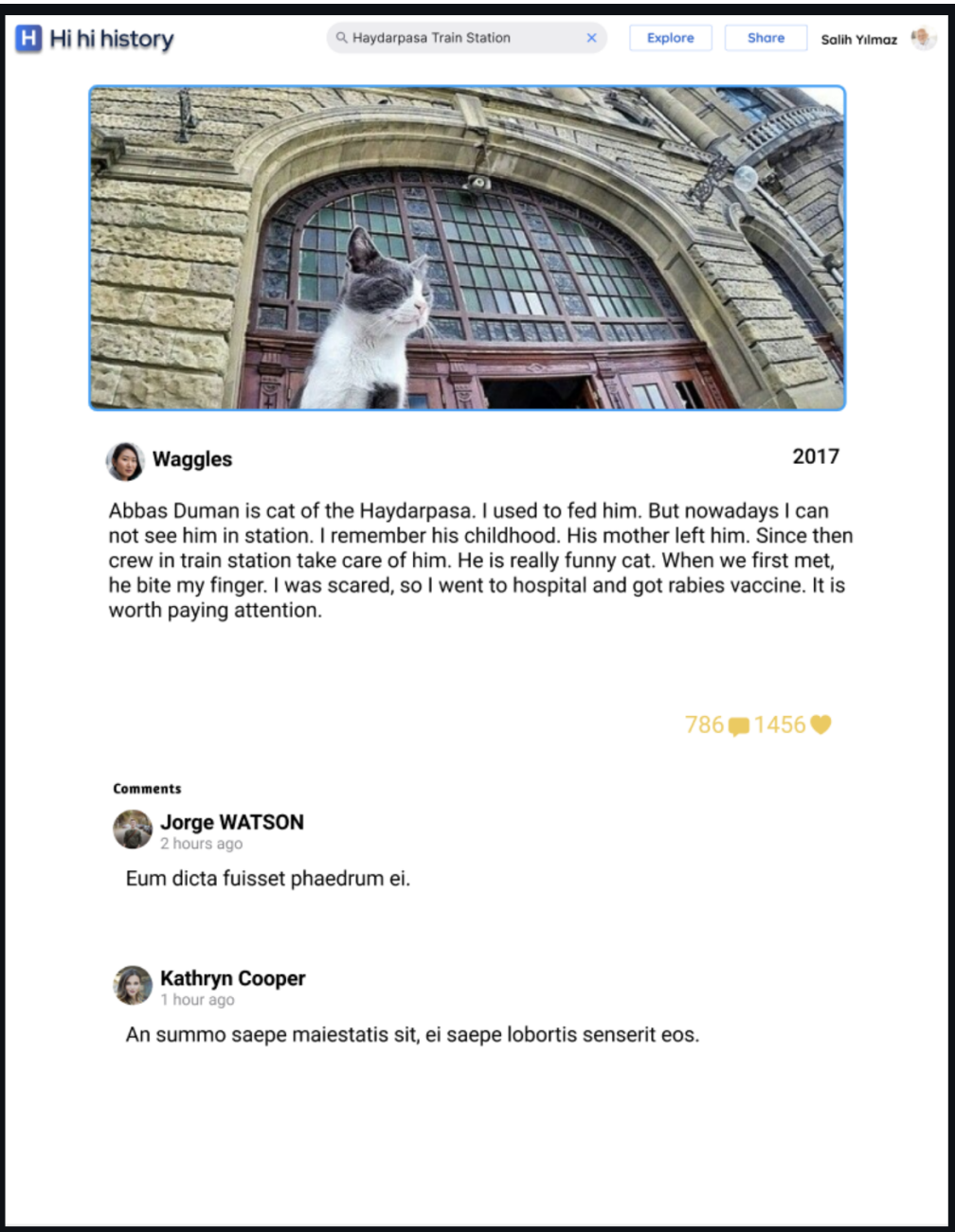
Scenario



1. He is already logged in.
2. He goes to the search page.
3. He searches about Haydarpaşa.



4. He searches about Haydarpaşa train station to read about the stories of the past.
5. He filters the posts to the 2010's and filters the tags to cats.



- **6.** He sees the top liked posts from the 2010's which are about cats in Haydarpaşa Train Station.
- **7.** He selects the top liked post about a cat named "Cat Abbas".
- **8.** He likes the story.

Hi hi history

Haydarpasa Train Station

Explore Share Salih Yılmaz

Waggles 2017

Abbas Duman is cat of the Haydarpasa. I used to fed him. But nowadays I can not see him in station. I remember his childhood. His mother left him. Since then crew in train station take care of him. He is really funny cat. When we first met, he bite my finger. I was scared, so I went to hospital and got rabies vaccine. It is worth paying attention.

786 1456

Comments

Salih Yılmaz

I like both cats and train stations :)

Share Cancel

Jorge WATSON
2 hours ago

Eum dicta fuisset phaedrum ei.

Kathryn Cooper
1 hour ago

An summo saepe maiestatis sit, ei saepe lobortis senserit eos.

9. He comments on the story.

iii. User Recommendation Scenario

User

- Salih YILMAZ

Story

Salih has worked in train stations for about 30 years and he has a lots of memories from these places. He is quite curious about other stations' stories all around the world and memories that people saved in those stations.

Precondition

- He is already logged in to the application.
- He posted some stories and photos before.
- He is following other users.

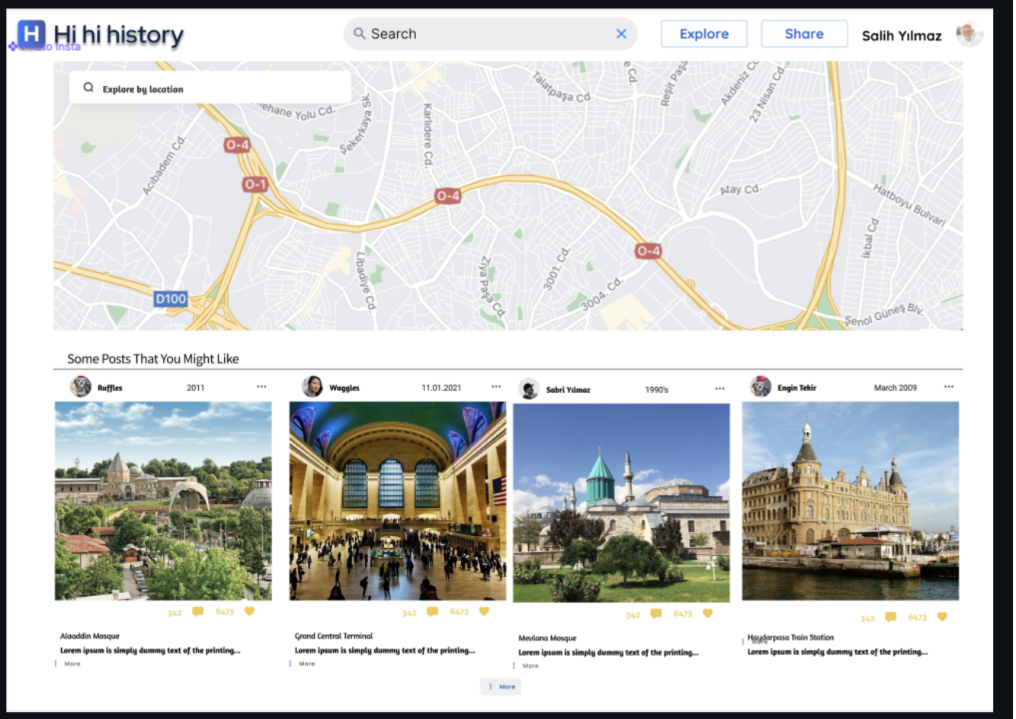
Goals

- He wants to see recommended places to explore the stories about these places which are recommended by his interest.

Acceptance Criteria

- **1.1.5.1** Users shall be able to view recommended locations based on their system activity and current location.

Scenario

- 

  1. He goes to the search page to see recommended places.
  2. He sees recommended railroad stations.(Haydarpaşa Tren İstasyonu, Grand Central Terminal,..)
  3. He sees a post about Grand Central Terminal.
  4. He likes the post.

3. Admin Scenario

Persona

- Regina Phalange was born in the USA. She is 22 years old. She moved to Turkey after her mom's suicide. She is a senior Computer Engineering Student at Bogazici University.
- She writes a blog about his adventures about the time she was living in the streets . She loves to travel and goes on adventures.
- She is an admin user on the platform. She works in 5-hour periods on working days. She is free to choose her working time. She works four days in a week.

Story

She wants to help people to have a good experience on learning the places' history while she can make some pocket money. She decided to join the "platform" team as an admin user. She is responsible for keeping the platform safe according to Terms of Use agreement. And she needs to check the admin mailbox to see if any user has a problem with the platform.

Precondition

- She is already logged into the system with the account's credentials that is authorized as an admin.
- She has been an admin user quite a while.

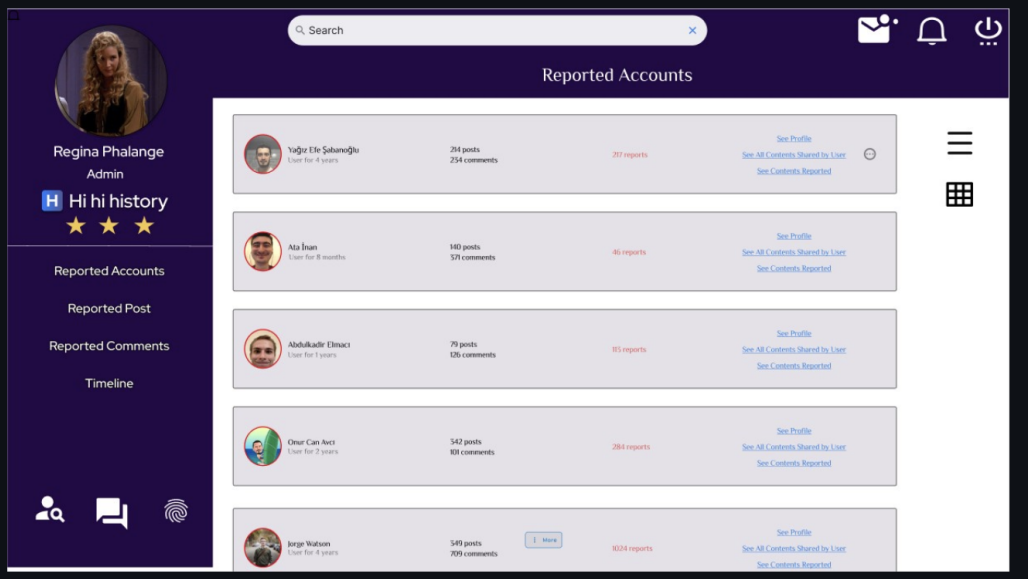
Goals

- She wants to delete unwelcome contents as described in Terms of Use.
- She wants to ban accounts which harm the other user experience on the platform.
- She wants to review the users' questions about the platform via the mailbox.
- She wants to remove the accounts bans if the user has a valid excuse that does not violate the Terms of Use.

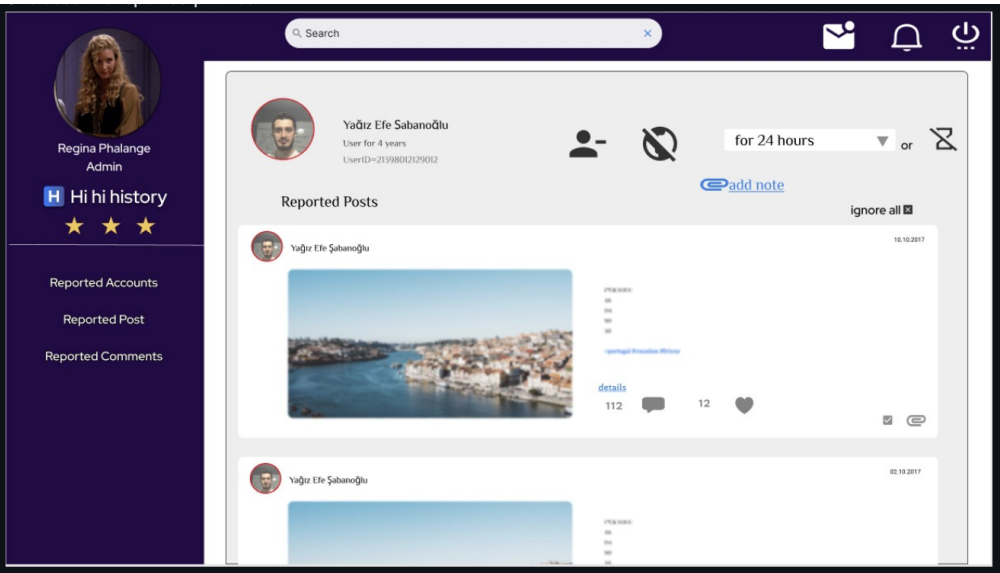
Acceptance Criteria

- She shall be able to delete contents successfully so that non-admin users cannot see it.
- She shall be able to ban accounts so that accounts will not have any function which users have.
- She shall be able to remove the ban from the accounts so that accounts will not lose any interaction.
- She shall be able to reply to the messages in the mailbox so that the user will get the message successfully.

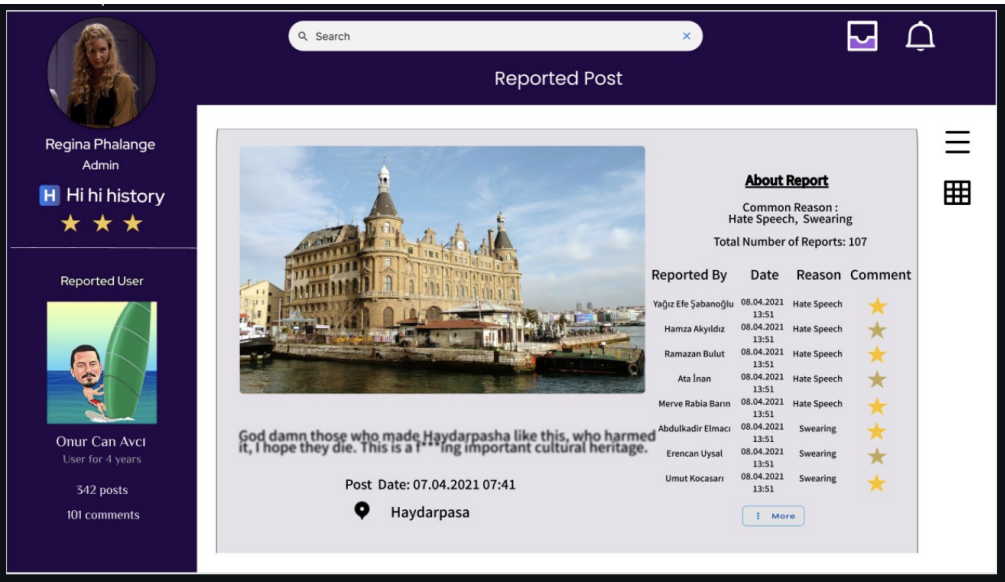
Scenario



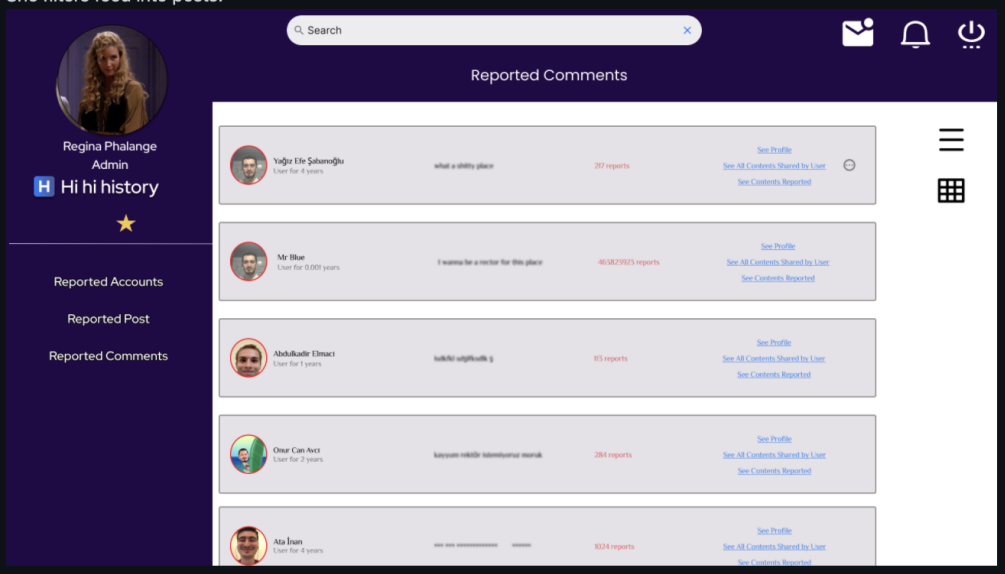
- She is already logged in.
- She is at the main page of admin users.
- She sees the reported profiles.



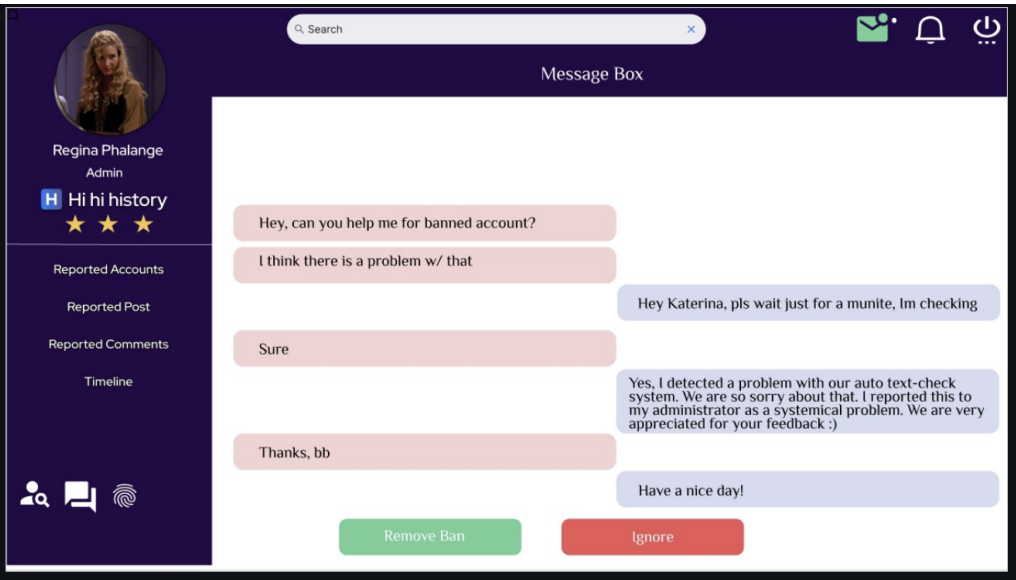
- She clicks the profile that is reported.
- She sees a content that violates the Terms of Use
- She decides to ban the account.
- She attaches a note about the violation.
- She bans the account.
- She returns to the main page.
- She sees a post.



- She decides to check reported posts.
- She filters feed into posts.



- She decides that the post does not violate the Terms of Use.
- She ignores the report.
- She returns to the main page.
- She decides to go over on reported comments.
- She sees a comment that violates the Terms of Use
- She deletes the comment and attaches a note about the violation.



- She decides to look at users' messages.
- She goes to the mailbox.
- She opens a message.
- A banned user explains himself in a way that the action that he took for banning is not a violation to Terms of Use.
- She reads the note about the violation and user excuse.
- She decides to remove the ban.

https://github.com/bounswe/2021SpringGroup7/wiki/Design