

M2 Group Report

Executive Summary

Summary of the project status

→ Mobile

For the mobile part, the project has most of the core functionalities. By sending requests to the backend, through a mobile application, now a user can sign up, login to the app. After the application is opened, the user can see her/his information via Profile Page. For now only username, email, birthdate, location, name, surname are available. Profile picture is not available for now, because we have not decided how to store pictures on the backend yet. Other than that, a user can now see all communities that have been created by all users. And users can create her/his own community. After creating a community, the creator of the community can create data types for posts that will be posted for that community. In those data types, there could be string, int, location and datetime type variables. The users are also able to subscribe to communities, which is a requirement to be able to create posts in that community. Additionally, the users can list all the posts created in the community and view the content of each of them, which includes all of the data type fields of the particular post for now.

→ Frontend

For the frontend application, the project has most of the core functionalities. By sending requests to the backend, through a frontend application, now a user can sign up, login to the app. After the application is opened, the user can see her/his information via Profile Page. For now only username, email, birthdate, location, name, surname are available. Profile picture is not available for now, because we have not decided how to store pictures on the backend yet. Other than that, a user can now see all communities that have been created by all users. Moreover, the creator of the community can create data types for posts that will be posted for that community. In those data types, there could be string, int, location and datetime type variables. The users are also able to subscribe to communities, which is a requirement to be able to create posts in that community. Additionally, the users can list all the posts created in the community and view the content of each of them, which includes all of the data type fields of the particular post for now. Functionality of creating communities have been implemented from profile page but due to the fact that its design in the profile page is yet to be finalized have not been merged into main part of the frontend.

→ **Backend**

We have implemented most of the project. There are lots of functionalities. A user can do basic operations in communities such as creating and viewing a post, creating a data type, joining a community, creating a community, user profile update. We have few functionalities remaining. Moderator permissions in a community and commenting under posts are not implemented. Search posts and tagging also remain unfinished work. We plan to implement search community and post functionality in the coming week. Overall, the project has met its main requirements. We are sure that we have enough time to finish the project.

The changes our team has planned/implemented to improve the development process (based on previous milestone reflections) and how they impacted our process

→ **Mobile**

To be more effective on the mobile part, we started to study earlier for this milestone. And especially in lab meetings, we spent more time discussing the details at the front. Otherwise, especially the api request was subject to change sometimes, and it made our job twice harder. Because once the api response or parameters are changed, we had to make significant changes on our codes. So, based on previous milestone experiences, we tried to be more organised at the inter-team level.

→ **Frontend**

We started to talk more deeply on how the problems can be solved and which way we should solve the problem in the meetings. Frontend team started to make local UI tests before merging the branch to master. Testing has been done manually up to this point but we are planning to automate testing by tests checking the integration with the backend. For instance, we are planning to check with our test whether or not implemented functionality's backend api successfully returned the desired fields. We started briefly describing the assigned tasks in the meetings. We have better UI than the first milestone and we will be better at the final milestone.

→ **Backend**

Other than the lab meetings, we encourage ourselves to contact other teams to ensure some details are planned accordingly. We invested our time in planning more and more. For example, there is an image uploading issue. We as a backend team talked this with other teams and planned how it should be for everyone's sake. Also, we had problems with github. Now before we implement something, we pull from the master branch and start developing after. This way, we don't encounter problems as we did in the previous weeks.

The improvements we have planned for the remainder of our project

→ **Mobile**

For the future, we plan to extend functionalities on mobile. They are;

- Creating posts for communities in one of predefined data types that have been set by creator
- Image selection and uploading for profile page of user
- Searching and filtering communities and posts
- Adding tags for posts

→ **Frontend**

- Backend tasks also valid
- Create community's implementation should be merged with main frontend
- Dynamic data type creation should be implemented
- Update profile should be implemented
- Profile privacy should be implemented
- Search should be implemented
- Mapping should be implemented
- Tests should be implemented

→ **Backend**

There are few remaining major features.

- Uploading images needs rework. Do we really store an image in the database? or how should we return the image as an url, in base64 string or in bytes?
- Searching communities and posts are not implemented yet. Posts can be searched by its datatype's fields or its tags.
- Tags should be implemented.
- Moderator permissions and commenting under a post still remain unimplemented.

Progress based on teamwork

Summary of the work

Name	Work Performed
Mustafa Can Aydin	<ul style="list-style-type: none">- Attended all the meetings.- As a communicator of the team, I led the team in the meetings with allocation of the tasks and flow of discussion.- I did write all the meeting notes- Fixed the bugs that surfaced during development such log in functionality's cache deletion bug, profile page's navigation bar's routing to community page bug, and deployment bugs such as connection refused error when a request is made from frontend to backend.- Added the logo that we agreed to use in a meeting to the repository's img folders to be found easily by everyone. Related commit : https://github.com/bounswe/2021SpringGroup8/commit/3e68c04c32826451873c4b61bb1d120ee5a5733e- Reviewed the backend-frontend integration in the issue : https://github.com/bounswe/2021SpringGroup8/issues/128- Sign up page's update has been reviewed by me in this issue : https://github.com/bounswe/2021SpringGroup8/issues/118- I improved login functionality by the implementation of cache of the logged in user and redirection to the profile page if the browsers cache includes user's information. I implemented log out functionality for the users to be able to use different accounts. I implemented a profile page showing user's information such as name, surname, email, city, etc. Related commit : https://github.com/bounswe/2021SpringGroup8/pull/135/commits

[/fe3842194ff4ed3d86c2d49b1ccbf65450a7be8b](#) , pull request:
<https://github.com/bounswe/2021SpringGroup8/pull/135> and issues :

<https://github.com/bounswe/2021SpringGroup8/issues/130> ,
<https://github.com/bounswe/2021SpringGroup8/issues/134>

- Solved the bug that happened during deployment of the application to AWS EC2 instance due to CORS error. For the solution I had to do research. Also, I reviewed how the deployment has been made and how healthy it is. After the review I let my teammates know about the deployment's state. Issue in which I describe the solution can be found :
<https://github.com/bounswe/2021SpringGroup8/issues/133>
- Merged the frontend implementation with Emre Demir in a meeting we had. Implementation included community page listing all communities, showing details of the community, and subscription to community. I did review the code and during the meeting I merged the pull request with the master. Pull request :
<https://github.com/bounswe/2021SpringGroup8/pull/137>
- I and Mehmet Temizel had a meeting in which I described him the current situation of the project, how we implemented the project using react, specific details about implementation of react such as use of react redux. After the explanation Mehmet had a better understanding of the project and became better equipped for future contribution. Meeting notes :
[https://github.com/bounswe/2021SpringGroup8/wiki/Meeting-1-\(10.12.2021\)](https://github.com/bounswe/2021SpringGroup8/wiki/Meeting-1-(10.12.2021))
- Muhammed Cavus and I made a meeting in which we discuss the design schemes that we

followed and spend time to make both mobile and frontend's designs as compatible as possible so that users can use both apps without any learning curve. Meeting notes :

[https://github.com/bounswe/2021SpringGroup8/wiki/Meeting-11-\(11.12.2021\)](https://github.com/bounswe/2021SpringGroup8/wiki/Meeting-11-(11.12.2021)) and issue :
<https://github.com/bounswe/2021SpringGroup8/issues/149>

- Updated the welcome page of the application with enhanced navigation bar, main body including logo of the app and pictures along with inspirational and informative quotes about the app encouraging users to use the app, and a footer for detailed navigation. Related commit :
<https://github.com/bounswe/2021SpringGroup8/commit/4f73dbc12a7508725c6f5ff2dbe5555b3009215e> , issue :
<https://github.com/bounswe/2021SpringGroup8/issues/147> and pull request :
<https://github.com/bounswe/2021SpringGroup8/pull/159>
- Some color changes, image size, and concealing fields only important to development has been implemented in the commit :
<https://github.com/bounswe/2021SpringGroup8/commit/d82552618c627f69e6ae328e36c4f6164898c94c>
- I resolved the bug on the button of the navigation bar of the profile page which routes from profile page to community page. I became functional with the changes I made in this commit :
<https://github.com/bounswe/2021SpringGroup8/commit/2110cd45ac0ddf994120a6c77d89cf8f7347e643>
- Implemented creation of community functionality from profile page. Related commit can be found :
<https://github.com/bounswe/2021SpringGroup8/tree/Frontend/Cr>

	Create community profile page
Emre	<ul style="list-style-type: none"> - Attended lab meetings, followed the plan while implementing the frontend application. - Tried to fix bugs at every commit. Details can be found in commits done by me (Physiclar, Animega Development, Bleudot -Dev) (I had to use different computers during the implementations) - Noticed a register api design issue and shared with team members (https://github.com/bounswe/2021SpringGroup8/issues/122) - Team logo was not accessible by all team members so I create and issue to make it structurally accessible for all team mates (https://github.com/bounswe/2021SpringGroup8/issues/123) - Completed frontend and backend integration and create working request model examples for frontend to backend to make other team members implementation easier (https://github.com/bounswe/2021SpringGroup8/issues/128) - Communities page is implemented (https://github.com/bounswe/2021SpringGroup8/commit/5d36ad236bd178dec148b40d028d829301ba31d1) - Community page and its functionalities are implemented. <ul style="list-style-type: none"> - Commits (https://github.com/bounswe/2021SpringGroup8/commit/5d36ad236bd178dec148b40d028d829301ba31d1) (https://github.com/bounswe/2021SpringGroup8/commit/c0d9ce6954b995e48108f854db725053aa79f9ec) (https://github.com/bounswe/2021SpringGroup8/commit/f54b7ac828a6c2e99ff9fb1b22794ca1868fa)

[90d](https://github.com/bounsw/2021SpringGroup8/commit/3b73da952622f19796d3a7f41604dd3e8f270f8))
(<https://github.com/bounsw/2021SpringGroup8/commit/01be725602c35c11c07b6fb4f8df9080588ee37>)

(<https://github.com/bounsw/2021SpringGroup8/commit/efea3aec187da54eb15d3f1fa33fd61030375533>)

(<https://github.com/bounsw/2021SpringGroup8/commit/4c708e631744d792f9a63bcf8b6cc38b9597f68f>)

(<https://github.com/bounsw/2021SpringGroup8/commit/67d743a0e4fb3261216cd3752f9cb3c97f1ee8aa>)

- Issues
(<https://github.com/bounsw/2021SpringGroup8/issues/129>)
(<https://github.com/bounsw/2021SpringGroup8/issues/138>)
(<https://github.com/bounsw/2021SpringGroup8/issues/163>)
(<https://github.com/bounsw/2021SpringGroup8/issues/164>)
(<https://github.com/bounsw/2021SpringGroup8/issues/165>)

- Pull Requests
(<https://github.com/bounsw/2021SpringGroup8/pull/137>)
(<https://github.com/bounsw/2021SpringGroup8/pull/158>)

- Created Amazon instance and deployed backend and frontend to cloud with Muhammet Çavuş
<https://github.com/bounsw/2022>

	<p>1SpringGroup8/issues/133</p> <ul style="list-style-type: none"> - Created an amazon instance before customer demo due to lack of memory for web app to run in the instance created by Muhammet Çavuş - Lead the mobile team and allocate and implement tasks among team members - Presented the frontend in the customer demo - Contributed to solve cors problem - Reviewed pull requests and issues (https://github.com/bounswe/2021SpringGroup8/pull/135) (https://github.com/bounswe/2021SpringGroup8/pull/148) - Merged the frontend implementation with Mustafa Can Aydın in a meeting we had. Implementation included community page listing all communities, showing details of the community, and subscription to community. I did review the code and during the meeting I merged the pull request with the master. Pull request : https://github.com/bounswe/2021SpringGroup8/pull/137 - Attended lab meetings, followed the plan while implementing the backend side. - Fixed bugs at backend side (Backend Tasks · Issue #124 · bounswe/2021SpringGroup8 (github.com)) - Implemented data type - Changed create post endpoint - Implemented data fields such as bool, int, location (for maps), string, datetime. - Changed city field to location field in user profile. Now, a user profile can contain a location object with real word coordinates. This is optional field for users. - Updated signup endpoint
Faruk	<ul style="list-style-type: none"> - Attended lab meetings, followed the plan while implementing the backend side. - Fixed bugs at backend side (Backend Tasks · Issue #124 · bounswe/2021SpringGroup8 (github.com)) - Implemented data type - Changed create post endpoint - Implemented data fields such as bool, int, location (for maps), string, datetime. - Changed city field to location field in user profile. Now, a user profile can contain a location object with real word coordinates. This is optional field for users. - Updated signup endpoint

	<ul style="list-style-type: none"> - Updated update profile endpoint - Prepared most of the backend side paragraphs in milestone 2. - Created issues: <ul style="list-style-type: none"> - Backend Tasks · Issue #124 · bounswe/2021SpringGroup8 (github.com) - Can we add return to communities page button in community viewing page? · Issue #160 · bounswe/2021SpringGroup8 (github.com) - Assigned issues: <ul style="list-style-type: none"> - Token field should be added to Register Api · Issue #122 · bounswe/2021SpringGroup8 (github.com) - Profile page update · Issue #130 · bounswe/2021SpringGroup8 (github.com) - Frontend and Backend need to be deployed to AWS and CORS error · Issue #133 · bounswe/2021SpringGroup8 (github.com) - Signup Birthdate Parameter Bug · Issue #143 · bounswe/2021SpringGroup8 (github.com) - New Logo suggestion · Issue #144 · bounswe/2021SpringGroup8 (github.com)
Muhammet Çavuş	<ul style="list-style-type: none"> ● Attended lab meetings. ● Fixed and re-designed the Sign Up page and Communitie Page, added “create community” pop up page https://github.com/bounswe/2021SpringGroup8/issues/139 ● Implemented lower navigation bar, which includes go to Home Page, Refresh Page, Create Community and LogOut buttons https://github.com/bounswe/2021SpringGroup8/issues/139 ● Birthday field in signup was

- faulty on backend side and I helped backend team to fix it
<https://github.com/bounswe/2021SpringGroup8/issues/142>
- Profile Page and Home Page for Android App is implemented
<https://github.com/bounswe/2021SpringGroup8/issues/142>
 - Maps Functionaliy is implemented for Android App
<https://github.com/bounswe/2021SpringGroup8/issues/136>
 - Logo for our project is created and presented to our team
<https://github.com/bounswe/2021SpringGroup8/issues/144>
 - Location selection API is changed on backend after discussion and I re-designed the code
<https://github.com/bounswe/2021SpringGroup8/issues/150>
 - Data types are implemented on mobile app
<https://github.com/bounswe/2021SpringGroup8/issues/151>
 - Helped my teammate to fix some problems
<https://github.com/bounswe/2021SpringGroup8/issues/155>
 - Provided necessary page(Inside of a community page) and necessary list to list post previews
 - Participated a meeting with Mustafa Can aydın to compare web and mobile apps
<https://github.com/bounswe/2021SpringGroup8/issues/149>
 - Created Amazon instance and deployed backend and frontend to cloud
<https://github.com/bounswe/2021SpringGroup8/issues/133>
 - Wrote Executive Summary mobile and Implementation of core functionality parts of this Milestone Group Report
- Pull Requests:**
- <https://github.com/bounswe/2021SpringGroup8/pull/140>
- <https://github.com/bounswe/2021SpringGroup8/pull/145>

	<p>https://github.com/bounswe/2021SpringGroup8/pull/153</p>
Mehmet Selahaddin Şentop	<ul style="list-style-type: none"> Attended weekly meetings, except when not available and reviewed the notes when not attended the meeting. Fixed some bugs in mobile implementation. Implemented listing the posts in a community. Implemented subscribing to a community. Implemented viewing inside of a post while displaying its fields. Implemented adding menu bar to all pages to provide returning to home page. Prepared the User Interface/User Experience part of this report for mobile part as well as contributed to executive summary for mobile part and made some corrections in other subtitles. <p>Issues:</p> <ul style="list-style-type: none"> https://github.com/bounswe/2021SpringGroup8/issues/132 -created https://github.com/bounswe/2021SpringGroup8/issues/155 -created https://github.com/bounswe/2021SpringGroup8/issues/144 -assigned <p>Pull requests:</p> <ul style="list-style-type: none"> https://github.com/bounswe/2021SpringGroup8/pull/157 -created, all of my implementations are included in this branch https://github.com/bounswe/2021SpringGroup8/pull/153 -merged <p>Commits:</p> <ul style="list-style-type: none"> https://github.com/bounswe/2021SpringGroup8/tree/mobile_posts -all of my commits are done in this branch that I created.

	<p>They can be traced back in the history tab.</p>
Gokhan	<ul style="list-style-type: none"> ● Attended lab meetings. ● Fixed some bugs ● at backend side(database related ones) (Backend Tasks - Issue #124) ● bounswe/2021SpringGroup8 (github.com) ● Implemented data type functionality. ● The post architecture is changed at database. ● Changed architecture of user profile. Now, a user profile can contain a location object with real word coordinates. This is optional field for users. ● Update profile functionality is added to database. ● Prepared Api call examples and status of requirements part of Milestone2. <p>Issues:</p> <ul style="list-style-type: none"> ● https://github.com/bounswe/2021SpringGroup8/issues/122: Token field should be added to Register Api ● https://github.com/bounswe/2021SpringGroup8/issues/124: Backend Tasks ● https://github.com/bounswe/2021SpringGroup8/issues/130: Profile page update ● https://github.com/bounswe/2021SpringGroup8/issues/133: Frontend and Backend need to be deployed to AWS and CORS error ● https://github.com/bounswe/2021SpringGroup8/issues/143: Signup Birthdate Parameter Bug ● https://github.com/bounswe/2021SpringGroup8/issues/144: New Logo suggestion

Mehmet	<ul style="list-style-type: none"> Attended almost all meetings(1 missing meeting may be) Checked some meeting notes Gave feedback related to an issue(https://github.com/bounsw/e/2021SpringGroup8/issues/144) that is about new logo design. Tried to catch up on the project. Made contribution to development process part of Milestone 2. Attended one-to-one meeting(https://github.com/bounsw/e/2021SpringGroup8/wiki/Meeting-1-(10.12.2021)) with communicator of the team. Checked some functionalities of the project on local computer. Checked the commit on this issue(https://github.com/bounsw/e/2021SpringGroup8/issues/128)
Bugra	

Status of the requirements

Name	Backend	Frontend	Mobile
Functional User Requirements			
Sign Up			
Guests shall be able to sign up as a customer by providing their name, surname, e-mail address, choosing a password and a unique user name.	completed	Completed	Completed
Each user shall have a unique username and password- that contains letters, digits, underscore and symbols-	completed	Completed	Completed
Sign In			

Users will use this unique username and password combination to log in system.	completed	completed	completed
Users shall be able to reset their password in case they forget their password.	In Progress	Not started	Not started
Profile			
Users shall have a profile page.	completed	Completed	Completed
Each user should be able to choose the communities that they are interested in.	completed	Completed	Completed
Each user should create a profile about them, that contains information about them.	completed	In Progress	In Progress
Each user should reveal as much information about them as they want.	In Progress	Not Started	In Progress
User Interactions			
Each user should be able to follow other users.	Not Started	Not Started	Not Started
Each user shall decide if they want to receive notifications or not.	Not Started	Not Started	Not Started
Communities			
Each user shall create a community, join an existing community or exit a joined community as they want.	Completed	In Progress	Completed
Users need to be accepted to a community according to a community-defined acceptance policy. (A community might be open to all users or it can accept only the users that got invitation from a member of the community.)	In Progress	Not Started	Not Started
Each community has an admin and can have multiple moderators.	In Progress	In Progress	In Progress

Community admin or moderators can create special Data Types for the community, which can contain fields of string, number, location or image.	Completed	Completed	Completed
If a user wishes to join an invitation-only community, their application must be approved by an administrator.	Not Started	Not Started	Not Started
Admins of a community should be able to monitor the posts against unethical and/or illegal contents and take actions accordingly (e.g. banning the member responsible for the post from the community).	In Progress	Not Started	Not Started
Functional System Requirements			
Notifications			
System shall provide notifications for users.	Not Started	Not Started	Not Started
Search			
System shall have search box in order for users to search communities or other users for interaction.	In Progress	In Progress	In Progress
Community Authorization			
System shall prohibit users from joining an invitation-only community without an approval.	Not Started	Not Started	Not Started
System shall prevent users from accessing a community if their access rights were to be suspended.	Not started	Not Started	Not Started
Recommendation			
System shall be able to	Not Started	Not Started	Not Started

recommend communities and other users to follow based on the interest of the users of the application.			
Non-Functional Requirements			
Standards			
Implementation of the system shall be in line with guidelines of W3C Activity Stream Standart 2.0 .	Completed	Completed	Completed
The semantic tagging shall supported with wikidata.org .	Not Started	Not Started	Not Started
Security			
User Information security			
Password length shall be in the predefined range (8-16) and passwords must not contain non-ASCII characters.	Completed	Completed	Completed
Passwords shall be encrypted with SHA256 hash-algorithms before being stored in the database.	Completed	Completed	Completed
Passwords shall include at least one upper case letter.	Completed	Completed	Completed
Account Security			
Respond of confirmation mail is necessary for activating account.	Not Started	Not Started	Not Started

New logins from unrecognized devices shall be sent to the user's email account in order to ensure that the user's account is not stolen and that login is done by the owner of the account.	Not Started	Not Started	Not Started
In case of a password change request, the user shall be informed about the request via given contact details and the change is allowed only after the user approves the request.	Not Started	Not Started	Not Started
Performance			
Response			
System response time shall be under 1 second in order for the user's flow of thought to stay uninterrupted.	Completed	Completed	Completed
System shall respond to the user in a maximum of 10 seconds. For longer delays, users may want to perform other tasks while waiting for the system to finish so the system should give feedback to users indicating when the system expects it to be done.	Completed	Completed	Completed
If a delay in response occurs during an operation using a third-party application, then the user shall be informed about that.	Not Started	In Progress	In Progress
Efficiency			
The system should be able to serve up to 1000 concurrent users.	In Progress	Completed	In Progress
At least, %20 of the system's processing capacity will be available during peak load hours.	Not Started	Not Started	Not Started

Reliability			
Response Reliability			
Supported number of concurrent users(1000) shall be scalable as the registered user numbers grow to (100 000).	Not Started	Completed	Not Started
If Crash happens the system shall response with error message.	Completed	Completed	Completed
Accessibility			
System will be available to access from desktop or mobile devices.	Completed	Completed	Completed
System shall be accessible using modern browsers.	Completed	Completed	Not Started
System will be responsive to aspect ratios of devices used to access the system.	Work should done in front-end and mobile	In Progress	In Progress
Platform shall support ASCII and UNICODE characters.	Completed	Completed	Completed

API endpoints

Provide the following:

- API documentation

Link to the documentation

<https://github.com/bounswe/2021SpringGroup8/blob/master/backend/Endpoints/Readme.md>

The screenshot shows a GitHub repository page for 'bounswe / 2021SpringGroup8'. The repository is public. The 'Code' tab is selected, showing the 'master' branch. The main content is the 'Readme.md' file, which contains the following JSON-like object definitions:

```
{  
    "_isoformat": "2021-11-13T15:30:16.263000"  
}
```

Objects Definitions

Date.Object

```
{  
    "@type": "Date.Object",  
    "locname": str,  
    "longitude": double,  
    "latitude": double  
}
```

Location.Object

DataType.Object

```
{  
    "@type": "DataType.Object",  
    "name": str,  
    "fields": {  
        "fieldname1": "str",  
        "fieldname2": "int",  
        "fieldname3": "bool",  
        "fieldname4": "location",  
        //fieldname5: "image",  
        "fieldname6": "datetime"  
    }  
}
```

🔗 User.Object

```
{  
    "@type": "User.Preview",  
    "id": str,  
    "username": str,  
    "name": str,  
    "surname": str,  
    "loc": Location.Object,  
    "birthdate": Date.Object,  
    "pplink": str,  
}
```

🔗 Community.Object

```
{  
    "@type": "Community.Object",  
    "id": str,  
    "communityTitle": str,  
    "description": str,  
    //moderators: [list of User.Preview],  
    "dataTypes": [list of DataType.Object],  
    "subscribers": [list of User.Preview],  
    "posts": [list of Post.Objects],  
    "creationTime": Date.Object,  
    "createdBy": User.Preview,  
}
```

🔗 Post.Object

```
{
```

```
"postTitle": str,  
"dataTypeName": str,  
"fieldValues": { (this is an example field values, it depends on the data type)  
    "a": "asdasd",  
    "b": 412,  
    "c": {  
        "locname": "Istanbul",  
        "longitude": 1.5,  
        "latitude": 1.7,  
        "@type": "Location.Object"  
    }  
},  
"creationTime": Date.Object,  
"postedAt": Community.Preview,  
"postedBy": User.Preview,  
"id": str,  
"@type": "Post.Object"  
}  
  
◀ ▶
```

🔗 View Definitions

🔗 User.Preview

```
{  
    "@type": "User.Preview",  
    "id": str,  
    "username": str,  
    "pplink": str,  
}
```

🔗 Community.Preview

```
{  
    "CommunityTitle": str,  
    "id": str,  
    "creationTime": {  
        "_isoformat": str  
    },  
    "createdBy": User.Preview,  
    "@type": "Community.Preview"  
}
```

🔗 Post.Preview

```
{  
    "@type": "Post.Preview",  
    "postTitle": str,  
    "id": str,  
    "postedBy": User.Preview,  
    "creationTime": {  
        "_isoformat": str  
    }  
}
```

🔗 User Functionalities

🔗 Update Profile

- Url: <http://localhost:8080/updateprofile>
- Method: POST
- Request:
 - Header:
 - Body (all fields are optional except @usertoken):
 - "@usertoken": str
 - "email": str
 - "name": str
 - "surname": str
 - "loc": Location.Object
 - "birthdate": ISODATE
 - "pplink": str
- Response:
 - Header:
 - Body:

```
{  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "@type": "User.UpdateProfile",  
    "@success": "True" or "False",  
    "@error": str => if @success is "False",  
}
```

🔗 Sign Up

- Url: <http://localhost:8080/signup>
- Method: POST
- Request:
 - Header:
 - Body:
 - "username": str
 - "email": str
 - "password": str
 - "name": str
 - "surname": str
 - "loc": Location.Object (optional)
 - "birthdate": ISODATE (optional)
 - "pplink": str (optional)
- Response:
 - Header:
 - Body:

```
{
    "@context": "https://www.w3.org/ns/activitystreams",
    "@type": "User.SignUp",
    "@success": "True" or "False",
    "@return": User.Object => if @success is "True",
    "@error": str => if @success is "False",
    "@usertoken": str,
}
```

🔗 Login

- Url: <http://localhost:8080/login>
- Method: POST
- Request:
 - Header:
 - Body:
 - "username": str
 - "password": str
- Response:
 - Header:
 - Body:

```
{  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "@type": "User.Login",  
    "@success": "True" or "False",  
    "@usertoken": str,  
    "@return": User.Object if @success is "True",  
    "@error": str => if @success is "False",  
}  
}
```

🔗 Get My Profile

- Url: <http://127.0.0.1:8080/getmyprofile>
- Method: POST
- Request:
 - Header:
 - Body:
 - "@usertoken": str
- Response:
 - Header:
 - Body:

```
{  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "@type": "User.GetMyProfile",  
    "@success": "True",  
    "@return": User.Object  
}
```

🔗 Get User Preview

- Url: <http://127.0.0.1:8080/getuserpreview>
- Method: POST
- Request:
 - Header:
 - Body:
 - "userid": str

- Response:

- Header:

- Body:

```
{  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "@type": "User.GetPreview",  
    "@success": "True",  
    "@return": User.Preview  
}
```

🔗 Community Functionalities

🔗 Create Community

- Url: <http://localhost:8080/createcommunity>

- Method: POST

- Request:

- Header:

- Body:

- "@usertoken": str
 - "communityTitle": str,
 - "description": str,

- Response:

- Header:

- Body:

```
{  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "@type": "Community.Create",  
    "@success": "True" or "False",  
    "@return": Community Object => if @success is "True",  
    "@error": str => if @success is "False",  
}
```

🔗 Subscribe To Community

- Url: <http://localhost:8080/subscribetocommunity>

- Method: POST
- Request:
 - Header:
 - Body:
 - "@usertoken": str
 - "communityId": str
- Response:
 - Header:
 - Body:

```
{  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "@type": "Community.SubscribeTo",  
    "@success": "True" or "False"  
}
```

🔗 Unsubscribe Community

- Url: <http://localhost:8080/unsubscribecommunity>
- Method: POST
- Request:
 - Header:
 - Body:
 - "@usertoken": str
 - "communityId": str
- Response:
 - Header:
 - Body:

```
{  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "@type": "Community.Unsubscribe",  
    "@success": "True" or "False"  
}
```

🔗 Get All Communities

- Url: <http://localhost:8080/getallcommunities>
- Method: POST
- Request:

- Header:
- Body:
- Response:
 - Header:
 - Body:

```
{  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "@type": "Community.GetAll",  
    "@success": "True",  
    "@return": [list of Community.Preview]  
}
```

🔗 Get Community

- Url: <http://localhost:8080/getcommunity>
- Method: POST
- Request:
 - Header:
 - Body:
 - "communityId": str
- Response:
 - Header:
 - Body:

```
{  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "@type": "Community.Get",  
    "@success": "True",  
    "@return": Community.Object  
}
```

🔗 Delete Community

- Url: <http://localhost:8080/deletecommunity>
- Method: POST
- Request:
 - Header:
 - Body:
 - "@usertoken": str
 - "communityId": str

- Response:
 - Header:
 - Body:

```
{  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "@type": "Community.Delete",  
    "@success": "True" or "False"  
}
```

🔗 Create Data Type

- Url: <http://localhost:8080/createdatatype>
- Method: POST
- Request:
 - Header:
 - Body:
 - "@usertoken": str
 - "communityId": str
 - "datatypepname": str
 - "datatypefields": Json | example { "a": "str", "b": "int", "c": "location" },
- Response:
 - Header:
 - Body:

```
{  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "@type": "Community.CreateDataType",  
    "@success": "True" or "False"  
}
```

🔗 Post Functionalities

🔗 Submit Post

- Url: <http://localhost:8080/submitpost>
- Method: POST

- Request:
 - Header:
 - Body:
 - "@usertoken": str
 - "communityId": str
 - "title": str
 - "datatypepname": str
 - "datatypevalues": Json | example => {"a": "asdasd", "b": 412, "c": { "locname": "Istanbul", "longitude": 1.5, "latitude": 1.7}}

- Response:

- Header:
- Body:

```
{
  "@context": "https://www.w3.org/ns/activitystreams",
  "@type": "Post.Submit",
  "success": "True",
  "@return": Post.Object
}
```

🔗 View Post

- Url: <http://localhost:8080/viewpost>
- Method: POST
- Request:
 - Header:
 - Body:
 - "postId": str
- Response:
 - Header:
 - Body:

```
{
  "@context": "https://www.w3.org/ns/activitystreams",
  "@type": "Post.View",
  "success": "True",
  "@return": Post.Object
}
```

🔗 Delete Post

- Url: <http://localhost:8080/deletepost>
- Method: POST
- Request:
 - Header:
 - Body:
 - "@usertoken": str
 - "postId": str
- Response:
 - Header:
 - Body:

```
{  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "@type": "Post.Delete",  
    "success": "True" or "False"  
}
```

- Link to API

IP address for API: <http://3.145.120.66:8080/>

You can reach out the endpoint links from here:

<https://github.com/bounswe/2021SpringGroup8/blob/master/backend/Endpoints/Readme.md>

Example of two endpoints:

<http://3.145.120.66:8080/login>

<http://3.145.120.66:8080/signup>

3 examples of API calls corresponding to core functionality of our application:

1. SignUp functionality:

To successfully makes these api calls you need to fill requests body with necessary informations in documentation.

Api call to <http://3.145.120.66:8080/signup>

```
// 20211217192530
// http://localhost:8080/signup

{
  "@context": "https://www.w3.org/ns/activitystreams",
  "@type": "User.SignUp",
  "@success": "True",
  "@return": {
    "username": "backendTest2",
    "email": "backend@gmail.com",
    "name": "Gökhan Uysal",
    "surname": "Uysal",
    "birthdate": null,
    "loc": null,
    "pplink": null,
    "subscribes": [
      ],
    "createdCommunities": [
      ],
    "posts": [
      ],
    "id": "61bcb9facbe3047eabfd6845",
    "@type": "User.Object"
  },
  "@usertoken": "qwkspwpezvzxodso"
}
```

2. Create Community functionality:

Api call to: <http://3.145.120.66:8080/createcommunity>

```

// 20211217192941
// http://localhost:8080/createcommunity

{
  "@context": "https://www.w3.org/ns/activitystreams",
  "@type": "Community.Create",
  "@success": "True",
  "@return": {
    "communityTitle": "backendTestCommunity",
    "description": "test Community for backend",
    "creationTime": {
      "_isoformat": "2021-12-17T19:29:41.442000"
    },
    "posts": [
      ],
    "subscribers": [
      ],
    "dataTypes": [
      ],
    "createdBy": {
      "username": "backendTest2",
      "id": "61bcb9facbe3047eabfd6845",
      "pplink": null,
      "@type": "User.Preview"
    },
    "id": "61bcbaf5cbe3047eabfd6846",
    "@type": "Community.Object"
  }
}

```

3. Create Post Functionality:

Api call to : <http://3.145.120.66:8080/submitpost>

```
// 20211217194001
// http://localhost:8080/submitpost

{
  "@context": "https://www.w3.org/ns/activitystreams",
  "@type": "Post.Submit",
  "success": "True",
  "@return": {
    "postTitle": "backend test Post",
    "dataTypeName": "interested instrument",
    "fieldValues": {
      "name of the instrument": "guitar",
      "years of experience": 4,
      "want to play in a band?": true
    },
    "creationTime": {
      "_isoformat": "2021-12-17T19:40:00.658000"
    },
    "postedAt": {
      "CommunityTitle": "Music Community",
      "id": "61b79c6f596357c065841558",
      "creationTime": {
        "_isoformat": "2021-12-13T19:18:07.327000"
      },
      "createdBy": {
        "username": "mss",
        "id": "61b79ba1596357c065841557",
        "pplink": "asd",
        "@type": "User.Preview"
      },
      "description": "Do you like playing/listening to music?",
      "@type": "Community.Preview"
    },
    "postedBy": {
      "username": "backendTest2",
      "id": "61bcb9facbe3047eabfd6845",
      "pplink": null,
      "@type": "User.Preview"
    },
    "id": "61bcbd60cbe3047eabfd6847",
    "@type": "Post.Object"
  }
}
```

Mapping functionality

Describe the status of this functionality.

Frontend →

In frontend map functionality is yet to be implemented. Frontend will catch up with the mobile in this functionality. We just implemented the latitude and longitude coordinate input fields.

Backend →

We have implemented a location object which stores a name, longitude and latitude. A user profile contains a location object as well as a data type can contain. We store the data coming from front-end and mobile teams.

Mobile →

On the mobile app, a user can set his/her location by choosing from a map. A button is added on mobile app so that users can open Google Maps.

The work that has been completed regarding this functionality

Mobile:

In the mobile app, while a user sign up, the user needs to set his/her location. To do that, he/she needs to open GoogleMaps to choose a location. When the Maps is opened, the user can choose locations in 3 different ways.

- By searching a place with a search bar.
- By dragging a marker to a custom location.
- By clicking anywhere on the GoogleMaps.

After a location is chosen, the name of the city that location resides in, longitude and latitude values are stored in the backend. So that when a user goes to Profile Page, she/he can see the exact location again on Profile Page.

Frontend:

Mapping functionality is yet to be implemented in the frontend. Latitude and longitude coordinate input fields have been implemented only.

The work that remains for this functionality

Mobile:

For mobile app, we will implement the suggested places feature. For now when a user searches a place, only the best fit result is set as the new location. But in the feature release, when a user searches something with a search bar, there will be multiple possible results and the user will be able to choose one of them.

And, the same functionality that we used for signup, will be added to posts. Since we already implemented the Maps, it will be easy to apply the same things on posts.

Frontend:

Firstly we are planning to implement a map that users can use to pick which city they are from. This map component will be embedded in profile page and will be seen to the user when he wants to enter information about his/her city. Also, user will be able to add city information through this map component when they are creating posts. Map component shall save the latitude and longitude information to be send to backend and let users to have location information in the posts that they created. Map component shall include functionalities of

- Selecting a location by clicking places on the map component
- Dragging the marker on the component to select/change location
- By entering the name of the location to the component's search field it will select from suitable locations
- When clicked to a predefined location's name component will pop out to show it on the google maps view considering

User Interface/User Experience

[Link to the code in the project repository](#)

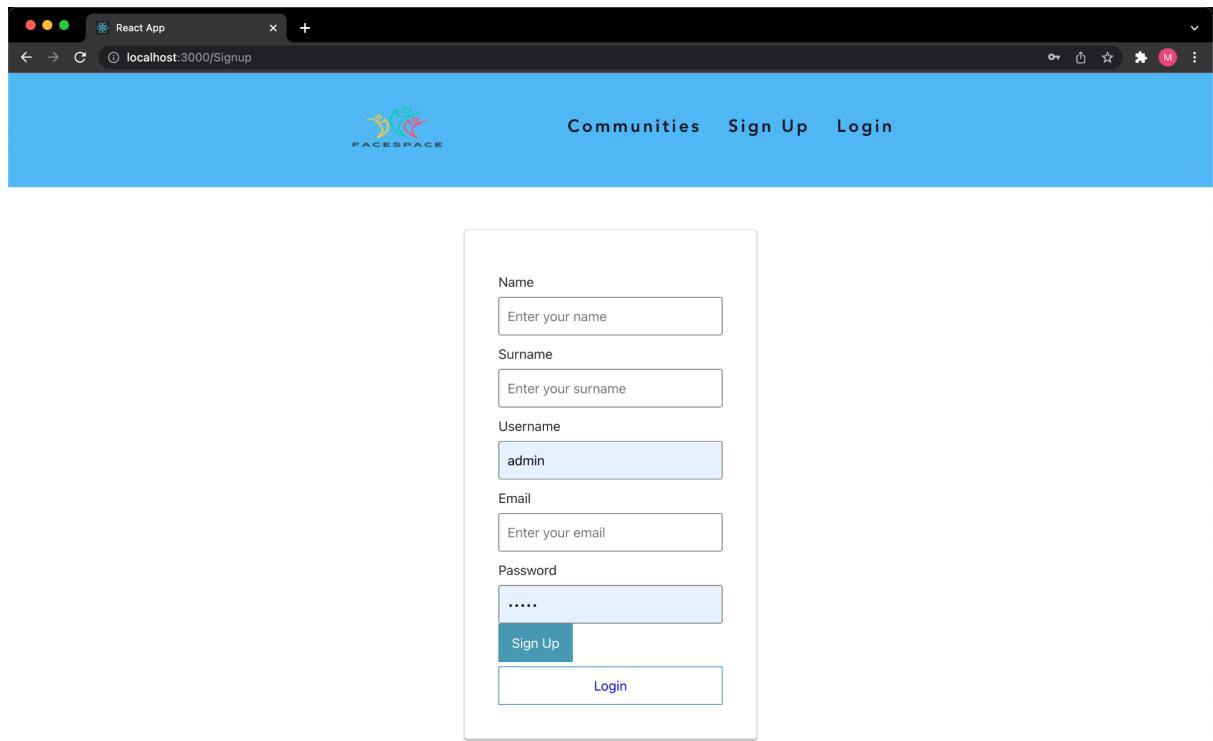
Mobile ->

<https://github.com/bounswe/2021SpringGroup8/tree/master/mobile/FaceSpace/app/src>

Frontend ->

<https://github.com/bounswe/2021SpringGroup8/tree/master/frontend>

Screenshot of all implemented Web interfaces



Sign Up page

A screenshot of a web browser showing a form for creating a new community. The title bar says "React App" and the address bar shows "localhost:3000/createPost/61b79b8b596357c065841556". The page has a blue header with the "FACESPACE" logo and navigation links for "Communities", "Sign Up", and "Login". The main content area contains fields for "Title" (filled with "Chess Community"), "Data Type" (set to "datatypepeyeni2123"), three text input fields labeled "a", "b", and "c", and four location-related fields: "Location Name", "Longitude", "Latitude", and a "Submit" button.

Title
Chess Community

Data Type
datatypepeyeni2123

a

b

c

Location Name

Longitude

Latitude

Submit

Post creation page

A screenshot of a web browser showing a community profile. The title bar says "React App" and the address bar shows "localhost:3000/community/61b79b8b596357c065841556". The top navigation bar includes "Communities", "Keyy", and "Log Out". The main content area shows a "Soccer Community" with an "UNSUBSCRIBE" button. It features two posts by user "emre": one titled "title" with a sunset image and another titled "Great Final" with a similar sunset image. To the right, there are sections for "About" (described as a community for soccer fans), "Tags" (soccer football), and "Moderators" (emre). A blue circular "+" button is located at the bottom right.

Soccer Community

UNSUBSCRIBE

emre @emre title

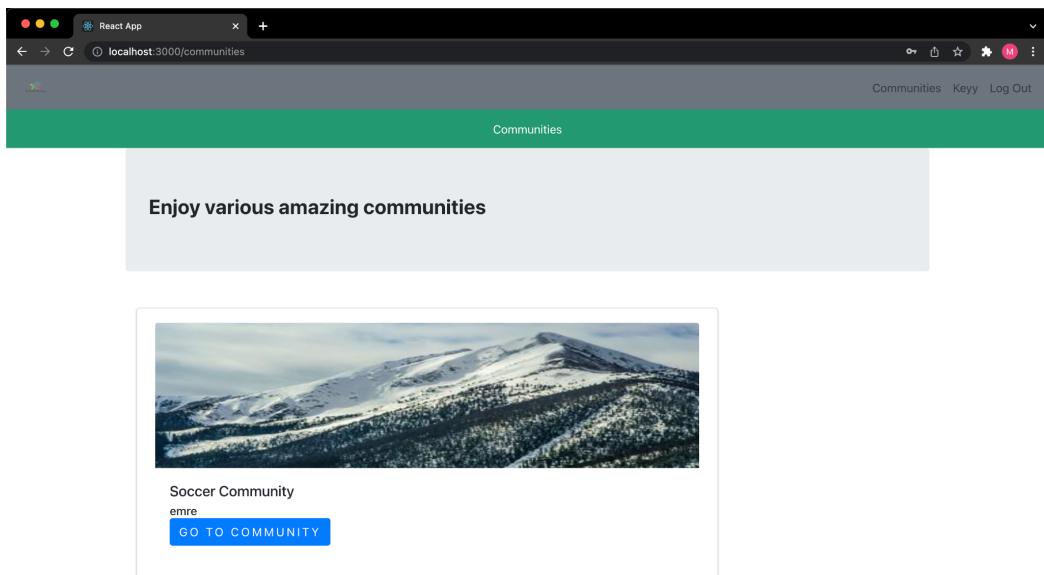
emre @emre Great Final

About
This community is for soccer fans

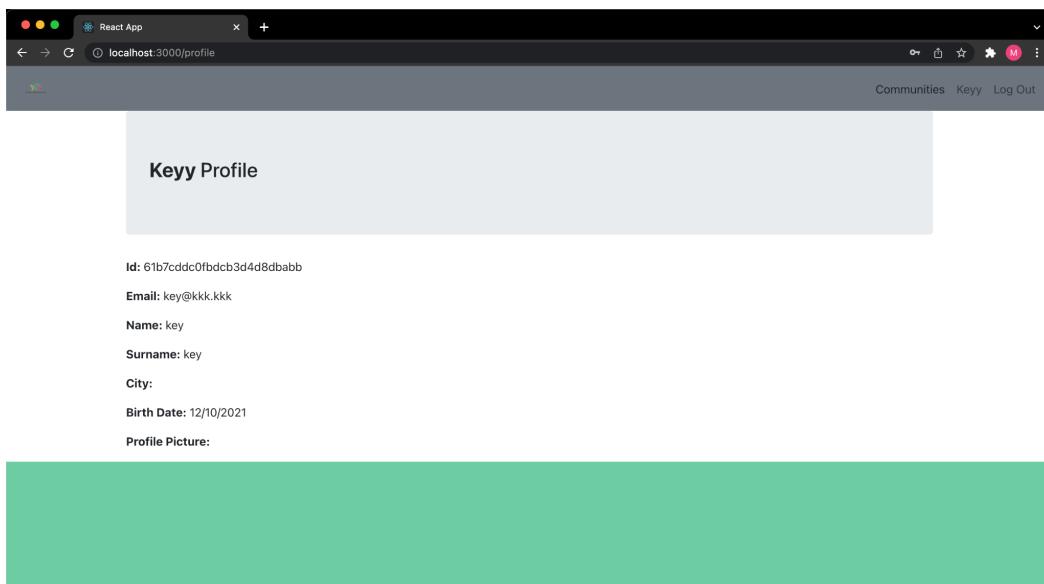
Tags
soccer football

Moderators
emre

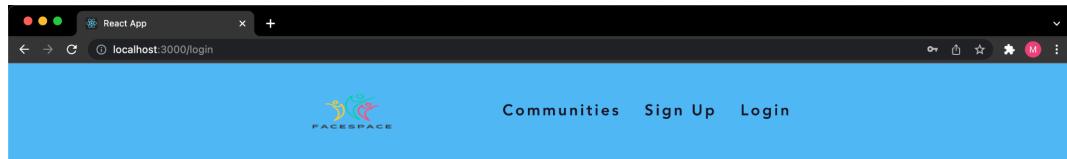
Detailed view of the community



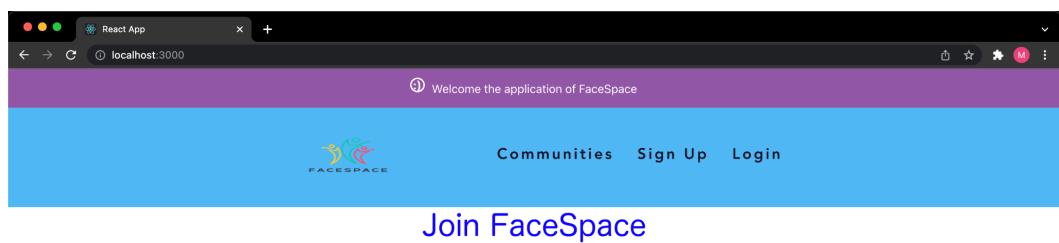
Page listing all the communities



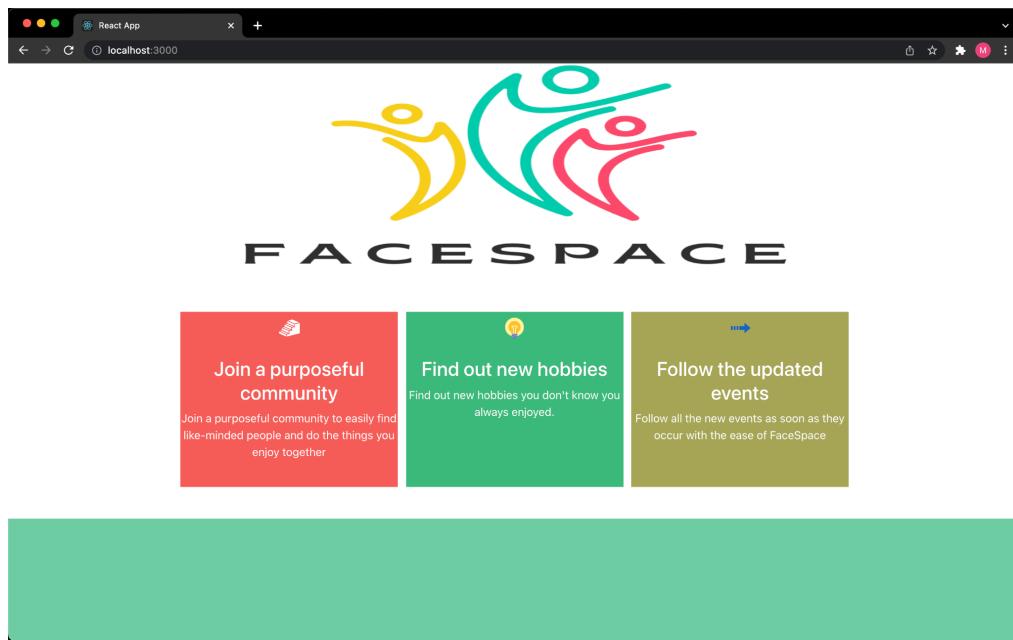
User profile page



Login Page



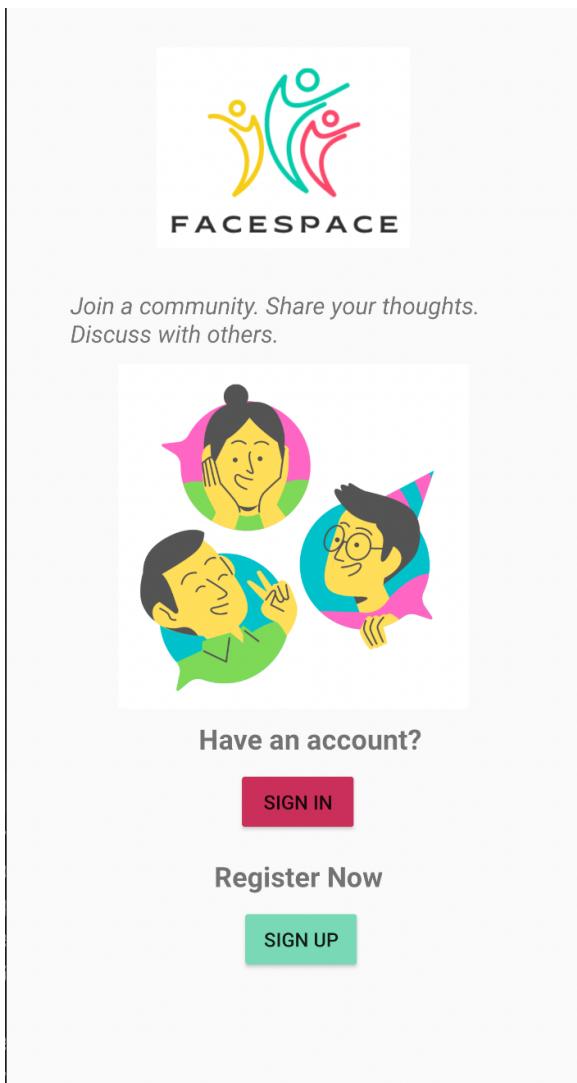
Welcome page (upper view)



Welcome page (lower view)

Screenshot of all implemented Mobile Interfaces

Main Page



Sign In Page

The Sign In Page features the Facespace logo at the top left. Below it is a "Sign In" heading followed by the text "Hi there! Nice to see you again." There are two input fields: one for "Username" with placeholder text "enter username" and another for "Password" with placeholder text "enter your password". A large green "Sign In" button is centered below the input fields. At the bottom are two smaller buttons: "Forgot Password?" and "Sign Up".

Sign Up Page

Sign Up

Username:

enter username

Email:

enter your email

Password:

enter your password

Name:

enter your name

Surname:

enter your surname

City:

City



Birthday:

Name:

enter your name

Surname:

enter your surname

City:

City



Birthday:

Date



Profile Picture Link:

enter profile picture link

I have read and accept everything.

Sign Up

Have an account?

Sign In

Home Page

Welcome to FaceSpace

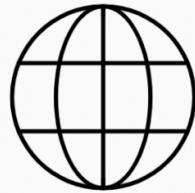
Hello, mss!

Here, you will share thoughts and find out what others think.

My Profile



My Communities



All Communities



Log Out



<http://3.145.120.66:8080/login>
login successful. Your Token is :
henrfghulpfraxxd

Profile Page

MY PROFILE



@mss

ms@s.com



Back to
home



Surname Name

s
ms



Date of Birth

1966-12-13



City

Ä°stanbul

mss wants to go to his/her profile.

All Communities Page

Soccer Community by emre
This community is for soccer fans
Since: 2021-12-13 [Details](#)

Music Community by mss
Do you like playing/listening to music?
Since: 2021-12-13 [Details](#)

Kotlin Community by gravewalker
We learn kotlin here.
Since: 2021-12-13 [Details](#)

CMPE451 Lovers by mss
Everything about software engineering
Since: 2021-12-13 [Details](#)

sasad by Keyy

[Home](#) [Refresh](#) [Create New Community](#) [Logout](#)

Create New Community Page

Since: 2021-12-13

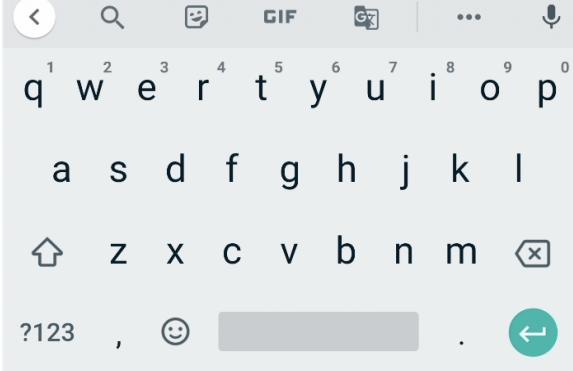
New Community

Title:

Description:

[Create](#) [Cancel](#)

Home Refresh Community Logout



Community Main Page (Subscribed)

shadow of war

Talion Forever

2021-12-13 Created by: gravew

[UNSUBSCRIBE](#)



Home



Refresh



Create
New Post



Logout

Community Main Page (Unsubscribed)

shadow of war

Talion Forever

2021-12-13 Created by: gravew

[SUBSCRIBE](#)



User with id henrfghulpfraxxd unsubscribed successfully!

Create
New Post

Logout

Create New Data Type

Create Data Type

Starting _____

Name	Type
bowBrand	: str

Name	Type
bowLength	: int

× +

Variable Name Select Type ADD

bowLength int RESET

bowLength | bowlength | mic

1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p
a s d f g h j k l
z x c v b n m x

?123 , smile . ✓

Creation of datatype is True

Choose a Created Data Type

Archery

Archery is a passion for me.

2021-12-13 Created by: mss CREATE DATATYPE

UNSUBSCRIBE

arrows
bows
Archers
Starting (cursor)

Home Refresh + Create New Post Logout

View Post Page

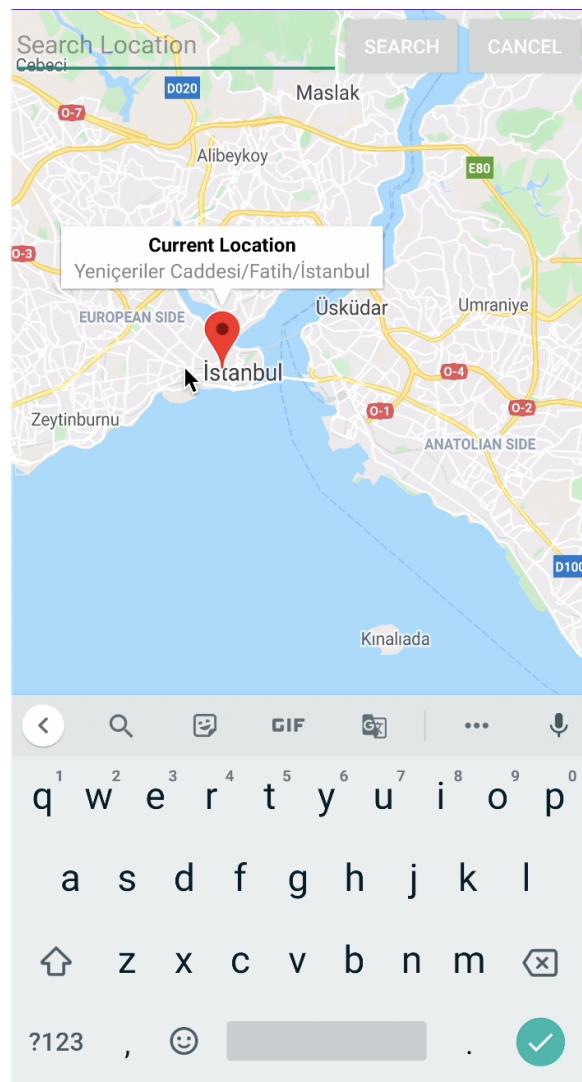
title Posted by: emre

Which : halisaha,
When : 19,
Where : {"locname":"naturel","longitude":1.13
,"latitude":3.23,"@type":"Location.Object"},

Date: 2021-12-14

Home **Refresh** **Create New Post** **Logout**

Map Interface



Standards

Explain the work you have completed towards the application of the W3C Activity Stream 2.0 recommended standard and Wikidata to your project.

Our backend apis are very similar to activity streams. Some of the features might not be compatible in terms of naming. However, the general structure is the same. We have schemas, objects and object previews. According to those, we return the objects. For example, a post object contains community and user object previews.

Example post object:

'Post.Object

```
{
```

```
"postTitle": str,  
"dataTypeName": str,  
"fieldValues": { (this is an example field values, it depends on the data type)  
    "a": "asdasd",  
    "b": 412,  
    "c": {  
        "locname": "Istanbul",  
        "longitude": 1.5,  
        "latitude": 1.7,  
        "@type": "Location.Object"  
    }  
,  
    "creationTime": Date.Object,  
    "postedAt": Community.Preview,  
    "postedBy": User.Preview,  
    "id": str,  
    "@type": "Post.Object"  
}
```

Example api call:

```
// 20211217192530
// http://localhost:8080/signup

{
  "@context": "https://www.w3.org/ns/activitystreams",
  "@type": "User.SignUp",
  "@success": "True",
  "@return": {
    "username": "backendTest2",
    "email": "backend@gmail.com",
    "name": "Gökhan Uysal",
    "surname": "Uysal",
    "birthdate": null,
    "loc": null,
    "pplink": null,
    "subscribes": [
      ],
    "createdCommunities": [
      ],
    "posts": [
      ],
    "id": "61bcb9facbe3047eabfd6845",
    "@type": "User.Object"
  },
  "@usertoken": "qwkspwpezvzxodso"
}
```

