

SWE 573 - SocialHub

Project Report

Git: <https://github.com/ecesari/software-development-practice>

Application URL: <http://ec2-3-239-28-109.compute-1.amazonaws.com:8080/>

Backend Rest API URL: <http://ec2-3-239-28-109.compute-1.amazonaws.com:8081/>

(If a URL change occurs, it will be published on the git page)

Zeynep Ece Sarıoğlu
11/01/2022

| | |
|--|-----------|
| Declaration | 3 |
| Introduction | 3 |
| Software Requirements Specification | 3 |
| Introduction | 3 |
| Purpose | 3 |
| Intended Audience | 3 |
| Scope | 4 |
| Definitions and Acronyms | 4 |
| Overall Description | 5 |
| User Needs | 5 |
| Assumptions and Dependencies | 5 |
| System Requirements | 5 |
| Functional Requirements | 5 |
| Nonfunctional Requirements | 14 |
| Diagrams | 16 |
| Class Diagram | 16 |
| Sequence Diagrams | 16 |
| User Creation | 16 |
| Create Service | 17 |
| Service Request | 17 |
| Request Approval | 18 |
| Use Case Diagram | 18 |
| Mockups | 19 |
| Deployment Status | 27 |
| System manual | 27 |
| User manual | 28 |
| Landing Page | 28 |
| For Unregistered Users | 28 |
| For Registered Users | 29 |
| Registration | 30 |
| Tags | 31 |
| Location | 31 |
| Login | 33 |

| | |
|--|-----------|
| Navbar | 33 |
| Services Navigation | 34 |
| Notifications | 34 |
| User Button | 35 |
| Services Pages | 35 |
| All Services | 36 |
| Created Services | 37 |
| Attending Services | 37 |
| Following User Services | 37 |
| Service Detail Page | 38 |
| Request Buttons | 38 |
| Service Over Button | 39 |
| Requesting a Service | 39 |
| Balance calculation when requesting a service | 40 |
| Completing a Service | 40 |
| Balance Update | 42 |
| User Profiles | 42 |
| Balance on hold | 44 |
| Creating a Service | 44 |
| Tags | 46 |
| Location | 47 |
| Service Request Approval | 48 |
| Approve | 49 |
| Deny | 49 |
| Success messages | 49 |
| Error messages | 50 |
| Testing | 50 |
| Useful Information to Test Deployed Application | 52 |

Declaration

I declare that all work in the project I am submitting is performed by myself. Credits can be given to Creative Tim, which offers free demo websites. The template can be found at: <https://www.creative-tim.com/product/vue-argon-design-system>.

Introduction

SocialHub is a web project that is aiming to create a community by creating and participating on services. Users have a balance based on the services that they have created and participated in. If they offer and successfully complete a service, they can gain credits and they can spend these credits on other credits afterwards. There are many features of the project such as creating services, approval system to join services, handshake system to complete services, location services, tagging system, user following, notification system, listing services, filtering services and so on. These features will be analyzed further in this project report.

Software Requirements Specification

Introduction

Purpose

The purpose of this document is to offer a detailed documentation of the SocialHub application. It explains the purpose and the features of the application.

Intended Audience

The Software Requirements Specification document is intended for the developers, project managers, testers and the stakeholders of the system.

Scope

SocialHub is a web application that serves as an open platform for internet users. The system is designed to create an environment where users can offer services based on their expertise and also participate in others' services.

More specifically, the platform is created to register users, get and display their profile information, and display their balance. list services and filter and sort them, display services, and get notifications for various events. SocialHub facilitates communication between its users as it offers real time events called services. The system contains a relational database containing information on users, services, tags, notifications, user followings and approvals.

Definitions and Acronyms

Database: Collection of all the information maintained by the system.

User: Person(s) who can register through the system.

Profile: A collection of information provided by the user(s).

Tags: A collection of keywords that define characteristics

Service: Events that are offered by users, services can also be consumed by the users.

User Following: Information on which users are following which users.

Request: A request by a user for a service.

Approval: A stage that is present on each request.

Notification: A collection of information provided by the system that is visible to users.

Software Requirements Specification: A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.

Overall Description

User Needs

The system will be open to the internet, and unregistered users will be able to display a sample of services with limited information. Users will be able to display all services, create services, receive and display notifications, request services, approve and deny requests to their services, follow and be followed by other users, display the profile of other users and approve services.

Users will be the primary users and unregistered users will be the secondary users.

Assumptions and Dependencies

SocialHub is deployed on an AWS server. The capacity of the server can be increased or decreased by demand. The system can be run on other service providers if needed. The software developed assumes the use of Docker for containerizing the application. The speed of the system will depend on the server capacity.

System Requirements

Functional Requirements

Registration and Authentication

| Requirement | Completed |
|---|-----------|
| Users can register to the system | yes |
| To sign-up the user must enter their name, password, location and a small biography | yes |

| | |
|--|-----|
| Users shall be able to choose from a predefined list of tags while registering | yes |
| Users shall view the location that they have chosen on the map while registering | yes |
| Users shall display profile information if authorized | yes |
| Users shall display detailed service information if authorized | yes |
| Users shall display filtered and sorted service information if authorized | yes |
| Users shall receive an error when making an unauthorized request | yes |

Users and Following

| Requirement | Completed |
|---|-----------|
| To sign-up the user must enter their name, password, location and a small biography | yes |
| The system has different types of users | yes |
| These users have different privileges | yes |
| The system has admin roles | no |

| | |
|---|-----|
| Users can display their own profiles | yes |
| Users can view other users' profiles | yes |
| Users can create new services | yes |
| Users can request to join services | yes |
| Users shall display the requests made to their services | yes |
| Users shall receive notifications | yes |
| Users can follow other registered users | yes |
| Users can be followed by other registered users | yes |
| Notifications shall be sent to the followed user when a user follows them | yes |
| Users can spend their credits | yes |
| Users shall receive notifications | yes |
| Users can follow other registered users | yes |
| Users can be followed by other registered users | yes |
| Notifications shall be sent to the followed user when a user follows them | yes |
| Users can spend their credits | yes |

| | |
|--|-----|
| Users can gain new credits | yes |
| Users shall display all notifications they have received | yes |
| Users can upload a profile picture | no |
| The users can block other users from following them | no |
| Users can message other users | no |
| Admins can review complaints | no |
| Admins can remove offerings and users | no |
| Users can flag services | no |
| Guests shall view services | yes |

Services

| Requirement | Completed |
|---|-----------|
| Services shall belong to a single user | yes |
| Services can be offered by users only | yes |
| Users shall request services | yes |
| Services shall have a quota for attendees | yes |

| | |
|--|-----|
| Services shall have the date and time information | yes |
| Services must have a description | yes |
| Services can be sorted by the distance to a logged in user | yes |
| Services can be sorted by their created date and service's date | yes |
| Services should be approved by both the created user and the attendees | yes |
| Services can be complete when the created user and an attendee confirms it | yes |
| Services must display a location information | yes |
| Services shall have a map to display the location | yes |
| Services can be flagged by users to be reviewed | no |
| After a service is complete, it can be rated by all attending users | no |

Events

| Requirement | Completed |
|--|-----------|
| Events are created by users | no |
| The users can participate in events | no |
| The system shall display the event information | no |
| The system shall display the guests of an event | no |
| Events shall have a comment section | no |
| Users can comment on the comment section of an event | no |

Request

| Requirement | Completed |
|---|-----------|
| Users can request a service if they have available credits | yes |
| Requests shall be displayed by the service creator | yes |
| Requests shall be approved or denied by the service creator | yes |

| | |
|---|-----|
| Notifications shall be sent to the service creator when a request is made | yes |
|---|-----|

Approval and Denial

| Requirement | Completed |
|---|-----------|
| Users shall approve or deny requests made to their services | yes |
| Notifications shall be sent to the requester when a request is approved or denied | yes |

Balance and Credits

| Requirement | Completed |
|-------------|-----------|
|-------------|-----------|

| | |
|---|-----|
| Credits shall be on hold when a request is made for a service | yes |
| Balance shall be displayed on the user's own profile page | yes |
| The users cannot consume any services when they reach the lower limit of their balance | yes |
| The users cannot offer any services when they reach the upper limit of their balance | yes |
| There is a lower limit of the user balance | yes |
| There is an upper limit of the user balance | yes |
| Providing a service increases the users' balance depending on the credit of the service that was offered | yes |
| Consuming a service decreases the users' balance depending on the credit of the service that was consumed | yes |
| Users initially have 5 credits in their balance | yes |
| Requesting a service shall put credits on hold depending on the credit of the service that was offered | yes |
| Balance shall have credits and credits on hold | yes |
| Balance shall be increased when a service is complete | yes |
| Balance shall have the credit information of a user | yes |
| Users shall have a balance | yes |

Homepage and Feed

| Requirement | Completed |
|--|-----------|
| Homepage should contain a link to user page | yes |
| Homepage should have a feed | yes |
| If the user is signed-in the feed displays the events and services created by the users that they follow | yes |
| If a guest enters the system they see a list of services | yes |
| If a guest enters the system they see a list of events | yes |

Nonfunctional Requirements

Security

| Requirement | Completed |
|---|-----------|
| A token should be used for user authentication | yes |
| Registered and logged in users shall be authenticated | yes |

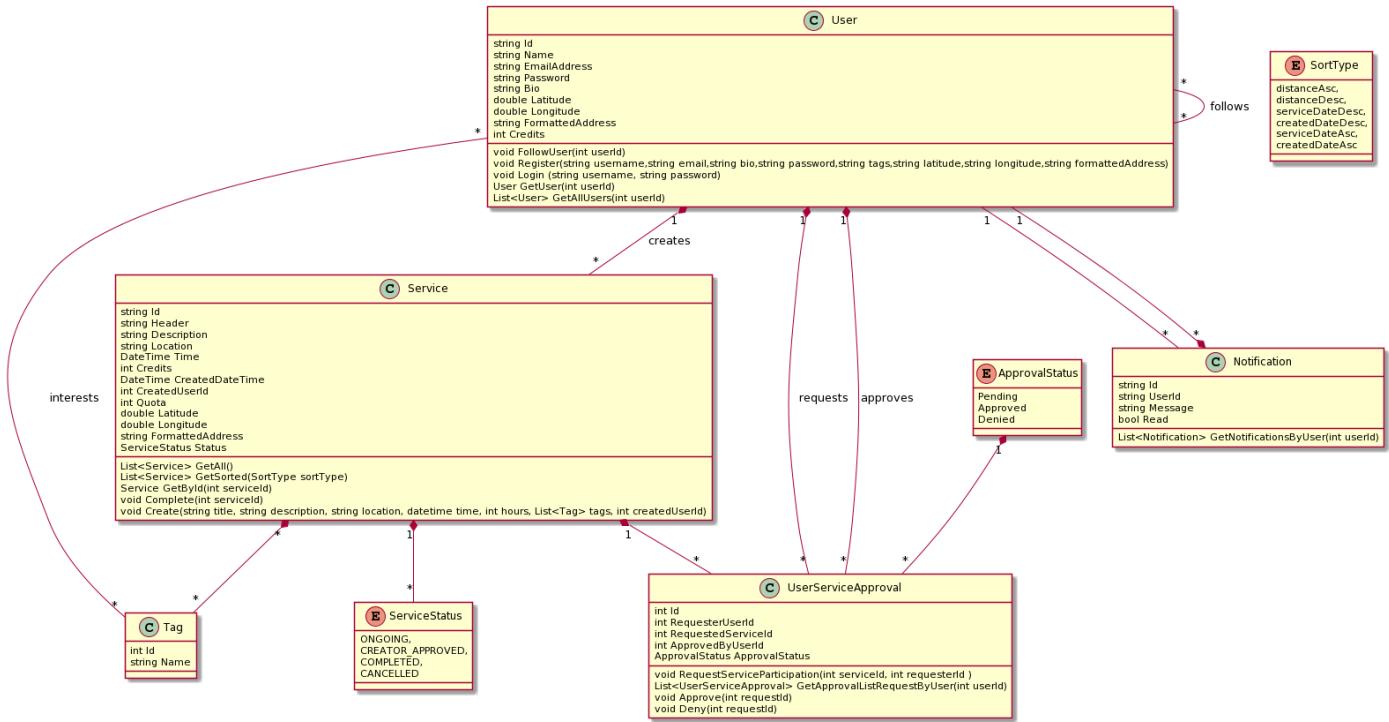
| | |
|---|-----|
| A balance belonging to the user shall be visible to them only | yes |
|---|-----|

System

| Requirement | Completed |
|--|-----------|
| The system shall have a frontend and backend application | yes |
| The backend shall be a REST api | yes |
| The frontend shall be a web application | yes |
| Frontend shall consume the api | yes |
| The system shall be in English | yes |
| The system should be accessible on web | yes |
| The system should be deployed on cloud | yes |

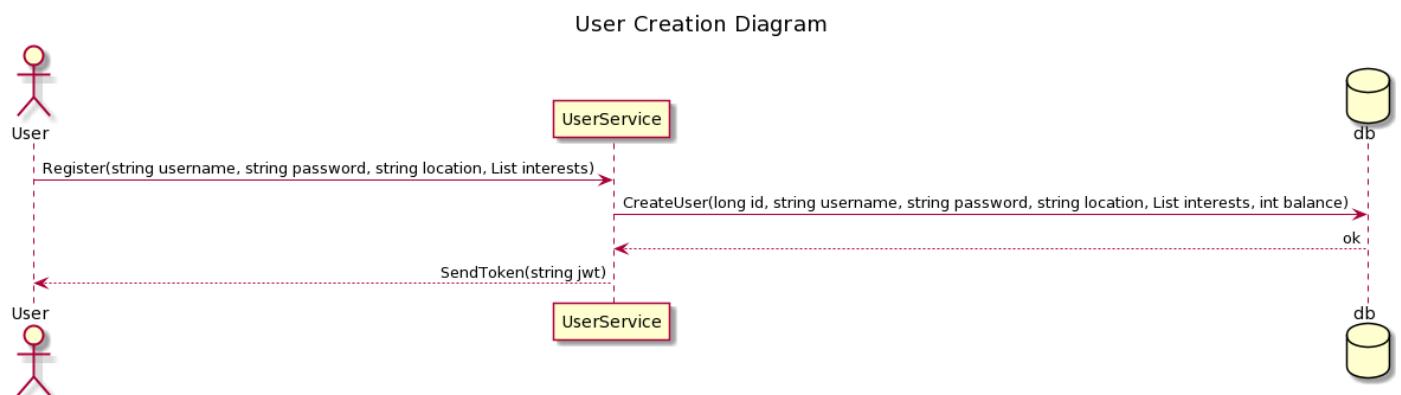
Diagrams

Class Diagram

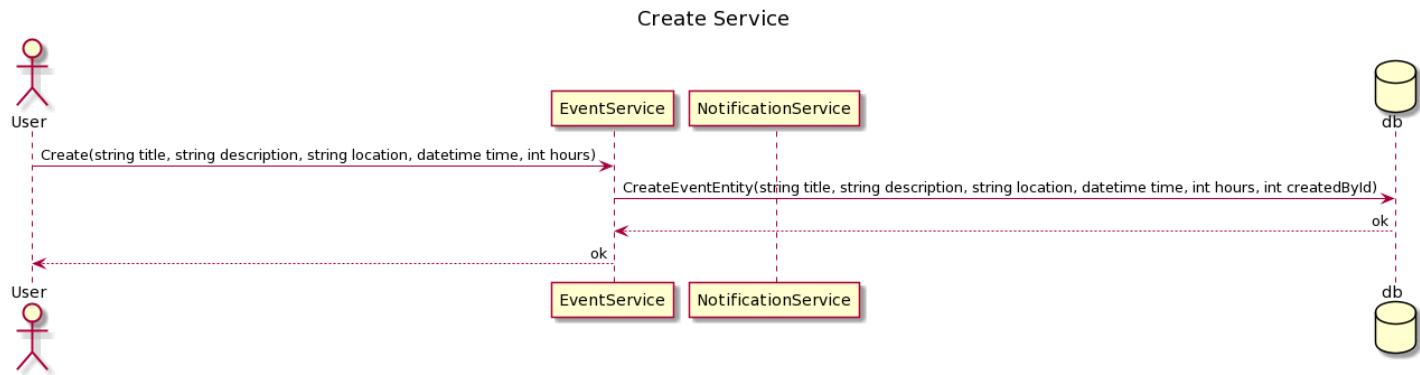


Sequence Diagrams

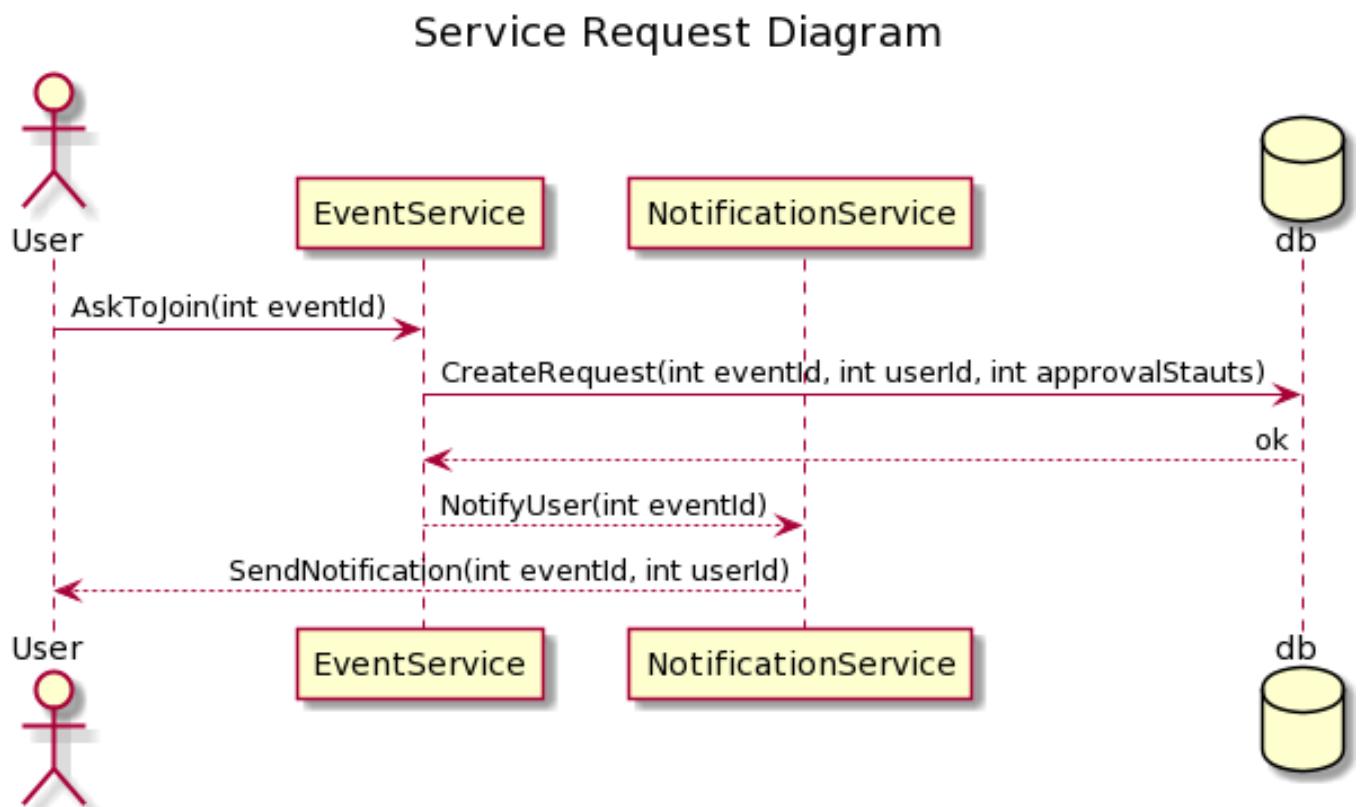
User Creation



Create Service

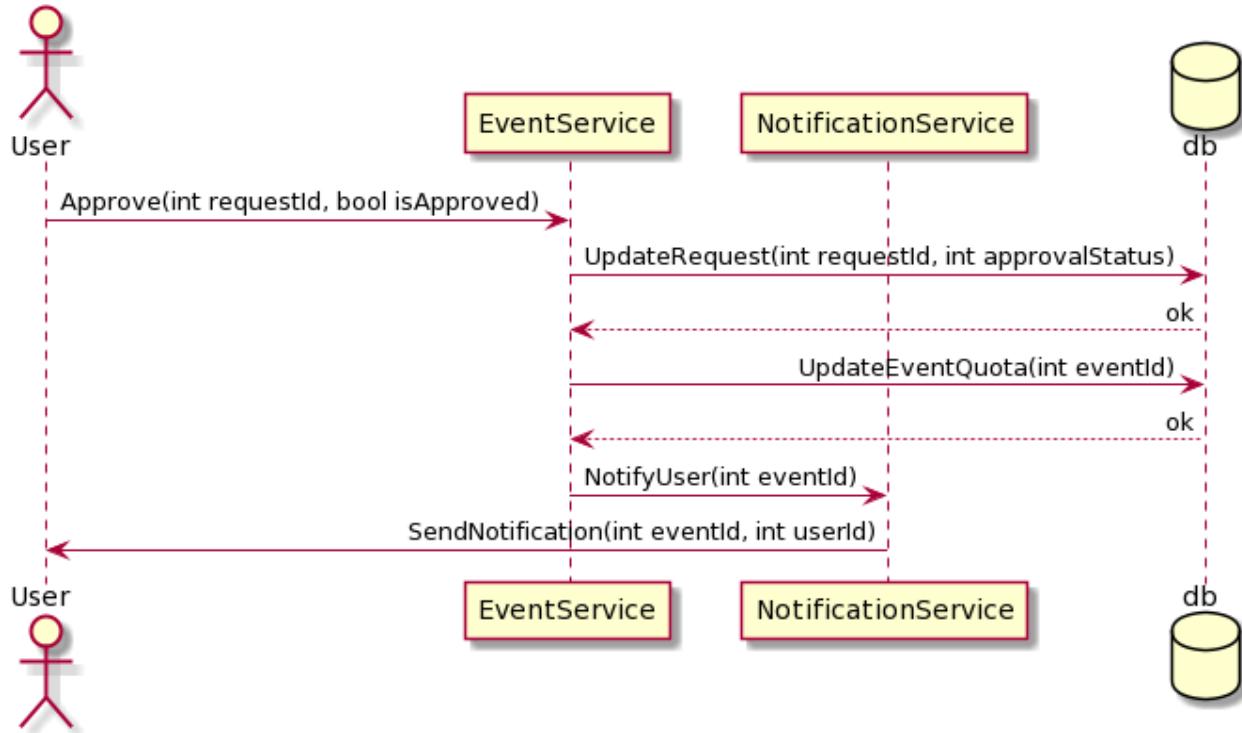


Service Request

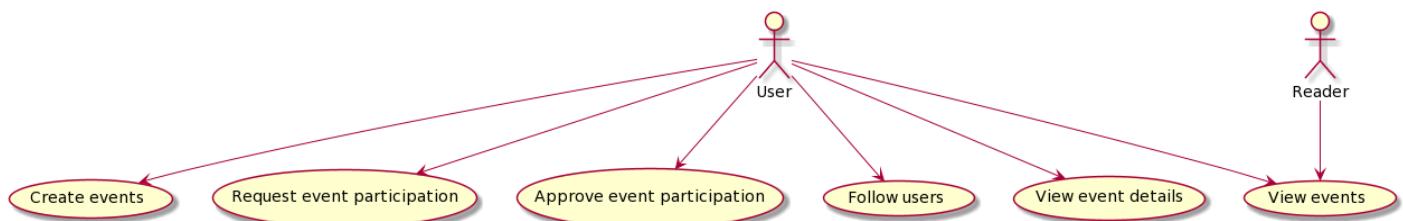


Request Approval

Request Approval Diagram



Use Case Diagram



Mockups



Untitled Project a Subtitle

REGISTER

LOG-IN

My Profile

Services

Services Offered

| Name | Time | Attendees | Waiting For Approval |
|-----------|------------------|-----------|----------------------|
| Service 1 | 17.11.2021 15:30 | 4/5 | 1 |
| Service 2 | 30.11.2021 20:30 | 1/5 | 0 |

Services Attending

| Name | Time | Approval Status |
|-----------|------------------|-----------------|
| Service 3 | 19.11.2021 11:00 | Approved |
| Service 4 | 05.02.2022 19:30 | Denied |
| Service 5 | 13.11.2021 11:00 | Denied |
| Service 6 | 19.11.2021 11:00 | Pending |

Balance: 14 Hours

Users

Following

| |
|--------|
| user-1 |
| user-2 |
| user-3 |

Followers

| |
|--------|
| user-4 |
|--------|

A sidebar menu with a search bar at the top. Below the search bar is a section titled "Menu" with a red horizontal line underneath. The menu items are: "HOMEPAGE", "EXPLORE" (with a dropdown arrow), "MY PROFILE", "USERS", and "CREATE SERVICE". Below this is a section titled "Get in touch" with a red horizontal line underneath. It contains three contact entries: an email link with an envelope icon, a phone number link with a phone icon, and an address link with a house icon. At the bottom of the sidebar is a copyright notice.

Search

Menu

HOME PAGE

EXPLORE

MY PROFILE

USERS

CREATE SERVICE

Get in touch

[information@untitled.tld](#)

[\(000\) 000-0000](#)

[1234 Somewhere Road #8254
Nashville, TN 00000-0000](#)

© Untitled. All rights reserved. Demo Images:
Unsplash. Design: [HTML5 UP](#).



Create Service

Title

Description

Tags:

Sports Arts Music Theater Movies Books

Photo:

No file chosen

Hours:

Search

Menu

[HOMEPAGE](#)

[EXPLORE](#)

[MY PROFILE](#)

[USERS](#)

Get in touch

Sed varius enim lorem ullamcorper dolore aliquam aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin sed aliquam facilisis ante interdum. Sed nulla amet lorem feugiat tempus aliquam.

information@untitled.tld

(000) 000-0000

1234 Somewhere Road #8254
Nashville, TN 00000-0000

© Untitled. All rights reserved. Demo Images: [Unsplash](#). Design: [HTML5 UP](#).

Untitled Project
a Subtitle

[REGISTER](#)
[LOG-IN](#)

A Sample Event

created by user-ece

in Istanbul

at 14.12.2021 13:00

Donec eget ex magna. Interdum et malesuada fames ac ante ipsum primis in faucibus. Pellentesque venenatis dolor imperdiet dolor mattis sagittis. Praesent rutrum sem diam, vitae egestas enim auctor sit amet. Pellentesque leo mauris, consectetur id ipsum sit amet, fergiat. Pellentesque in mi eu massa lacinia malesuada et a elit. Donec urna ex, lacinia in purus ac, pretium pulvinar mauris. Curabitur sapien risus, commodo eget turpis at, elementum convallis elit. Pellentesque enim turpis, hendrerit.

Donec eget ex magna. Interdum et malesuada fames ac ante ipsum primis in faucibus. Pellentesque venenatis dolor imperdiet dolor mattis sagittis. Praesent rutrum sem diam, vitae egestas enim auctor sit amet. Pellentesque leo mauris, consectetur id ipsum sit amet, fersapien risus, commodo eget turpis at, elementum convallis elit. Pellentesque enim turpis, hendrerit tristique lorem ipsum dolor.

Search 

Untitled Project a Subtitle

[REGISTER](#) [LOG-IN](#)

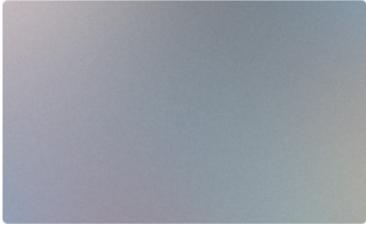
Events

Search



Interdum aenean
in Istanbul
Aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin aliquam facilisis ante interdum. Sed nulla amet lorem feugiat tempus aliquam.

[MORE](#)



Nulla amet dolore
in Istanbul
Aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin aliquam facilisis ante interdum. Sed nulla amet lorem feugiat tempus aliquam.

[MORE](#)



Tempus ullamcorper
in Istanbul
Aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin aliquam facilisis ante interdum. Sed nulla amet lorem feugiat tempus aliquam.

[MORE](#)



Sed etiam facilis
in Istanbul
Aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin aliquam facilisis ante interdum. Sed nulla amet lorem feugiat tempus aliquam.

[MORE](#)



Feugiat lorem aenean
in Istanbul
Aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin aliquam facilisis ante



Amet varius aliquam
in Istanbul
Aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin aliquam facilisis ante

Menu

- HOME PAGE
- EXPLORE
- MY PROFILE
- USERS

Get in touch

 information@untitled.tld

 (000) 000-0000

 1234 Somewhere Road #8254
Nashville, TN 00000-0000

© Untitled. All rights reserved. Demo Images: [Unsplash](#). Design: [HTML5 UP](#).

The image shows a web page layout with a sidebar on the left and a main content area on the right.

Left Sidebar (Menu):

- Search bar with a magnifying glass icon.
- Three horizontal red bars icon.
- Menu** (underlined)
- [HOMEPAGE](#)
- [EXPLORE](#) (with a dropdown arrow)
- [MY PROFILE](#)
- [USERS](#)

Main Content Area:

Login

Name

Password

Submit (red button)

Get in touch (underlined)

Sed varius enim lorem ullamcorper dolore aliquam aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin sed aliquam facilisis ante interdum. Sed nulla amet lorem feugiat tempus aliquam.

Email: information@untitled.tld

Phone: (000) 000-0000

Address: [1234 Somewhere Road #8254](#)
Nashville, TN 00000-0000

© Untitled. All rights reserved. Demo Images:
[Unsplash](#). Design: [HTML5 UP](#).

Search 

Untitled Project a Subtitle

[REGISTER](#) [LOG-IN](#)

Services

Search



Interdum aenean

@user-ece

Aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin aliquam facilisis ante interdum. Sed nulla amet lorem feugiat tempus aliquam.

[MORE](#)



Nulla amet dolore

@user-ece

Aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin aliquam facilisis ante interdum. Sed nulla amet lorem feugiat tempus aliquam.

[MORE](#)



Tempus ullamcorper

@user-ece

Aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin aliquam facilisis ante interdum. Sed nulla amet lorem feugiat tempus aliquam.

[MORE](#)



Sed etiam facilis

@user-ece

Aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin aliquam facilisis ante interdum. Sed nulla amet lorem feugiat tempus aliquam.

[MORE](#)



Feugiat lorem aenean

@user-ece

Aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin aliquam facilisis ante



Amet varius aliquam

@user-ece

Aenean ornare velit lacus, ac varius enim lorem ullamcorper dolore. Proin aliquam facilisis ante

Menu

- Homepage
- Explore
- My Profile
- Users

Get in touch

 information@untitled.tld

 (000) 000-0000

 1234 Somewhere Road #8254
Nashville, TN 00000-0000

© Untitled. All rights reserved. Demo Images: [Unsplash](#). Design: [HTML5 UP](#).

Deployment Status

The application is dockerized and hosted on EC2. The URL is:

<http://ec2-3-239-28-109.compute-1.amazonaws.com:8080/>

It is possible to reach the backend rest api which can be found at:

<http://ec2-3-239-28-109.compute-1.amazonaws.com:8081/>

URL's can change as EC2 instances renew DNS addresses when rebooted, current URL's can be found at the git page.

Backend and frontend applications each have a Dockerfile. Docker-compose has three images; app image that is the backend image, db image with mysql image and ui image.

System manual

To run the system, either the repo must first be cloned to the local machine or the release version should be downloaded.

The steps to clone the repo are as follows:

1. Open Git Bash
2. Go to the directory of where the files will be installed
3. Enter command `git clone`
<https://github.com/ecesari/software-development-practice.git>
4. Press enter

The steps to directly download the release version are as follows:

1. Go to
<https://github.com/ecesari/software-development-practice/releases/tag/publish>
2. Go to the assets tab
3. Click on one of the release files provided

After successfully getting the code, docker-compose can be run to deploy the system on docker. The steps to deploy on docker are as follows:

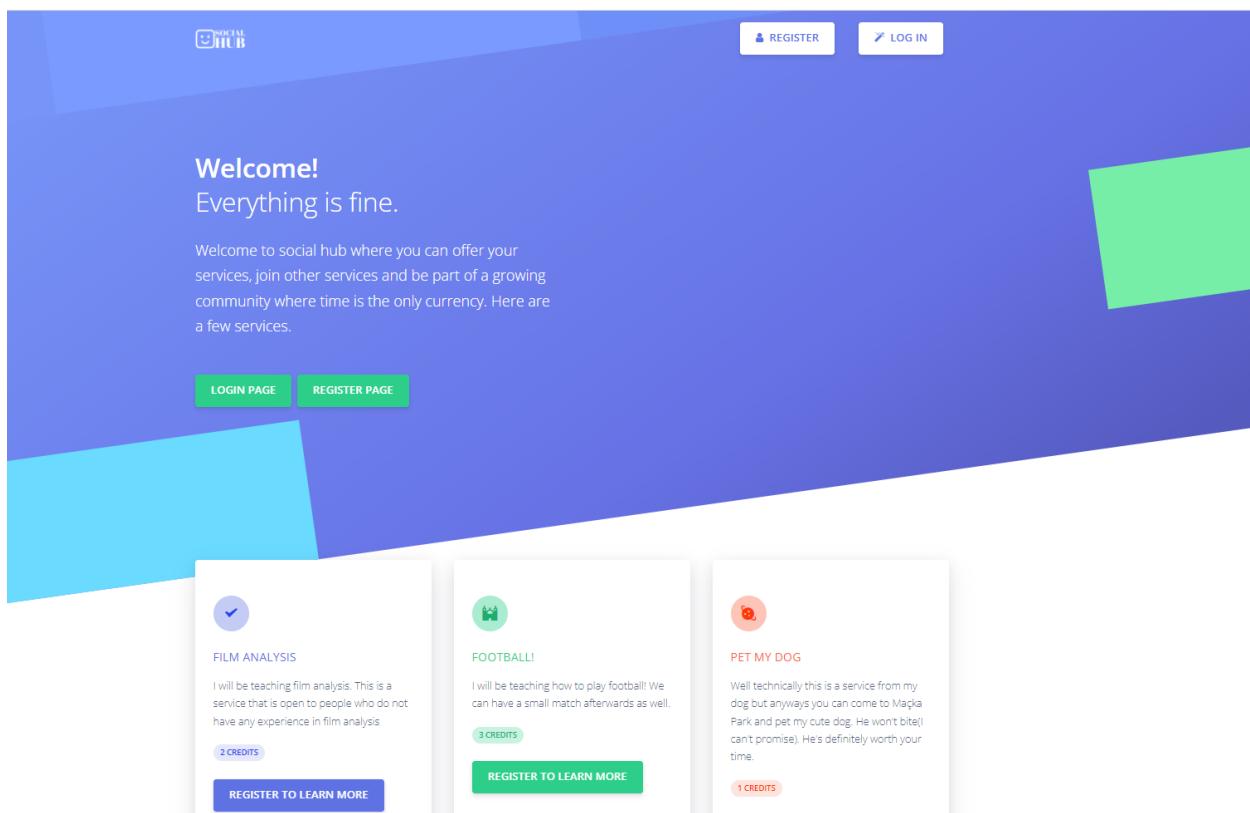
1. Go to the backend folder with the following command on your terminal
2. **cd software-development-practice/backend**
3. Enter the command docker-compose up –build

After following all the steps, the system should successfully run on your computer. Frontend application that is visible to users will be at <http://localhost:8080/> and the backend api will be at <http://localhost:8081/>

User manual

Landing Page

For Unregistered Users



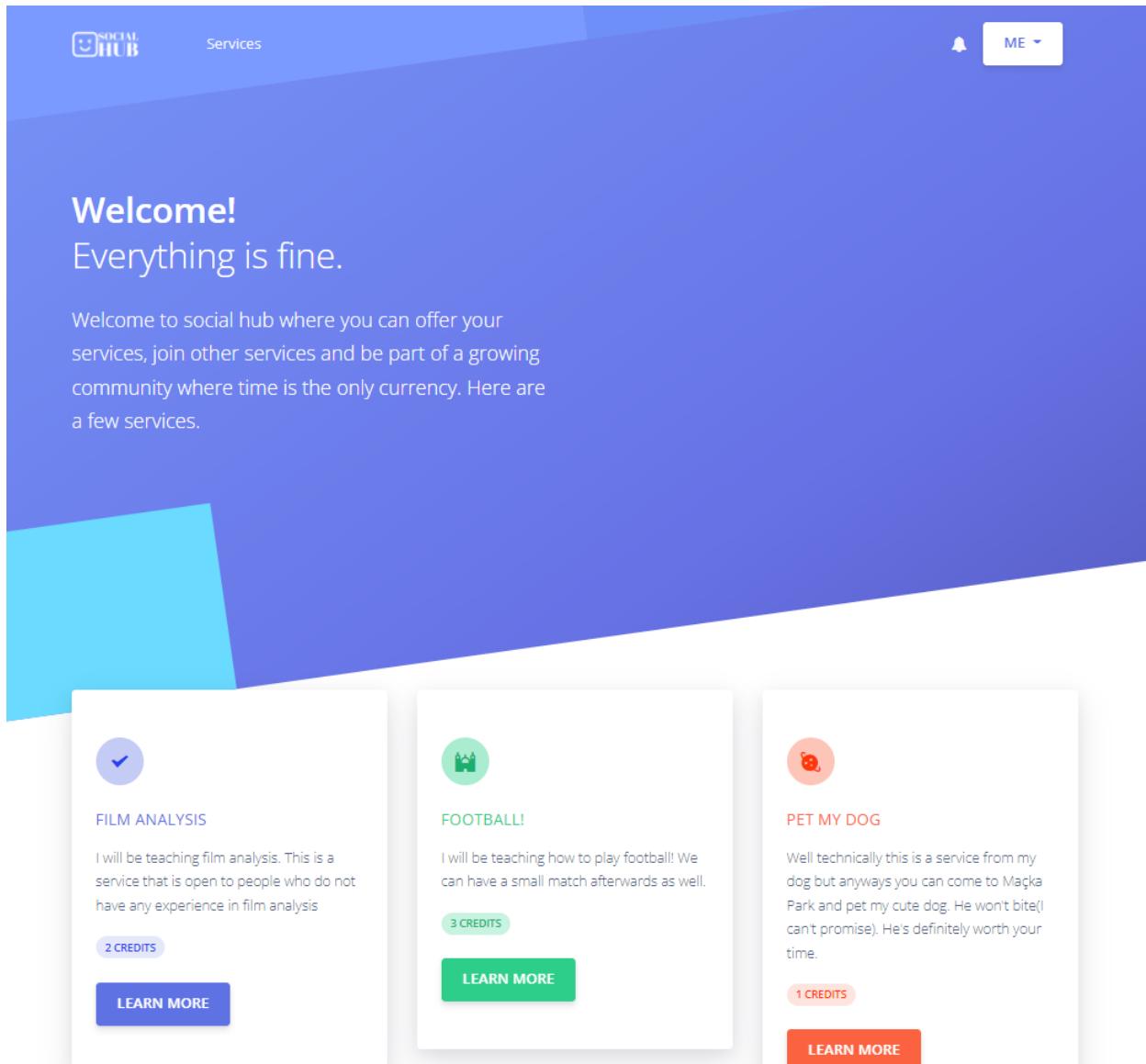
Homepage of the application for unregistered users

The application's landing page offers limited information for unregistered users. Three services are displayed on the main page with title, description and credit information.

The user has to register to view more information. If unauthenticated, only the registration and login buttons will be available to the user.

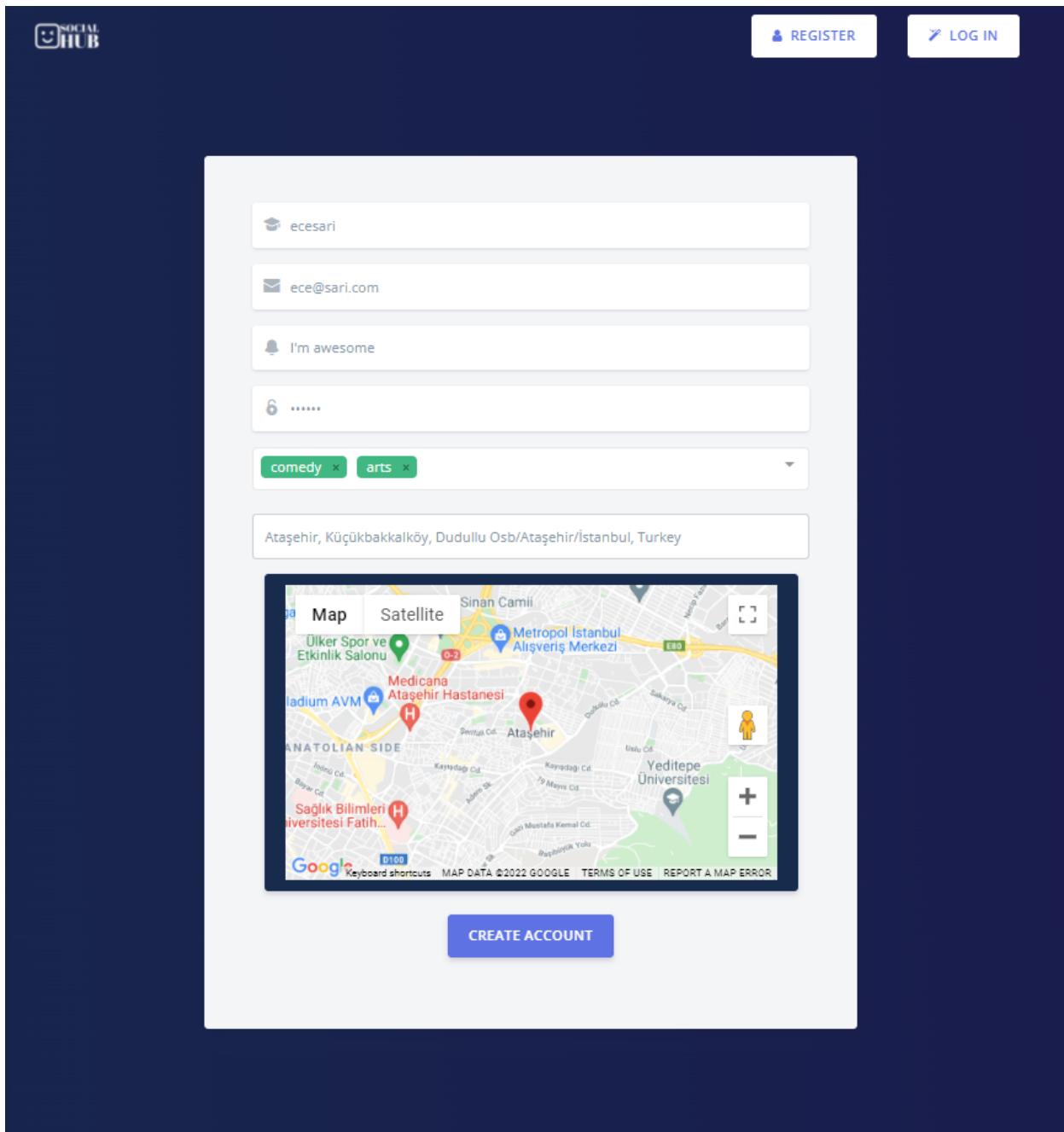
For Registered Users

The landing page changes when the user is logged in. Services now have links to the detail page and the navigation bar displays different information



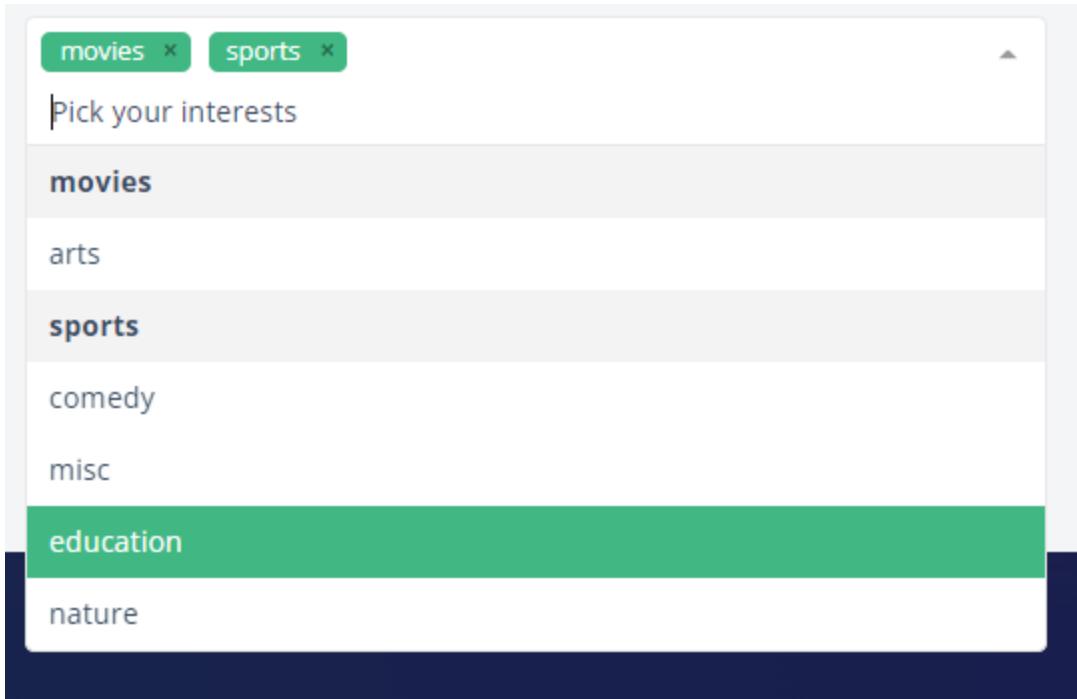
Registration

Users can register to the page by following the buttons on the home page. The placeholders will give information on which information the user should enter.



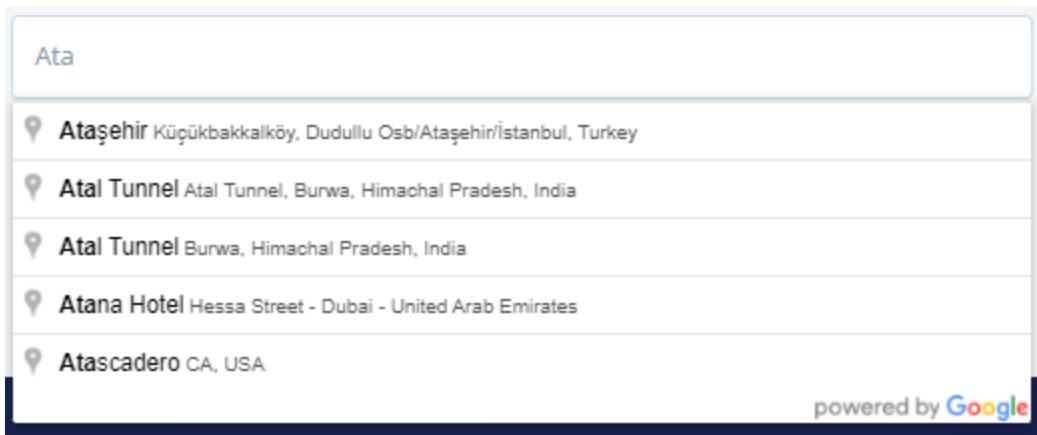
Filled registration form

Tags

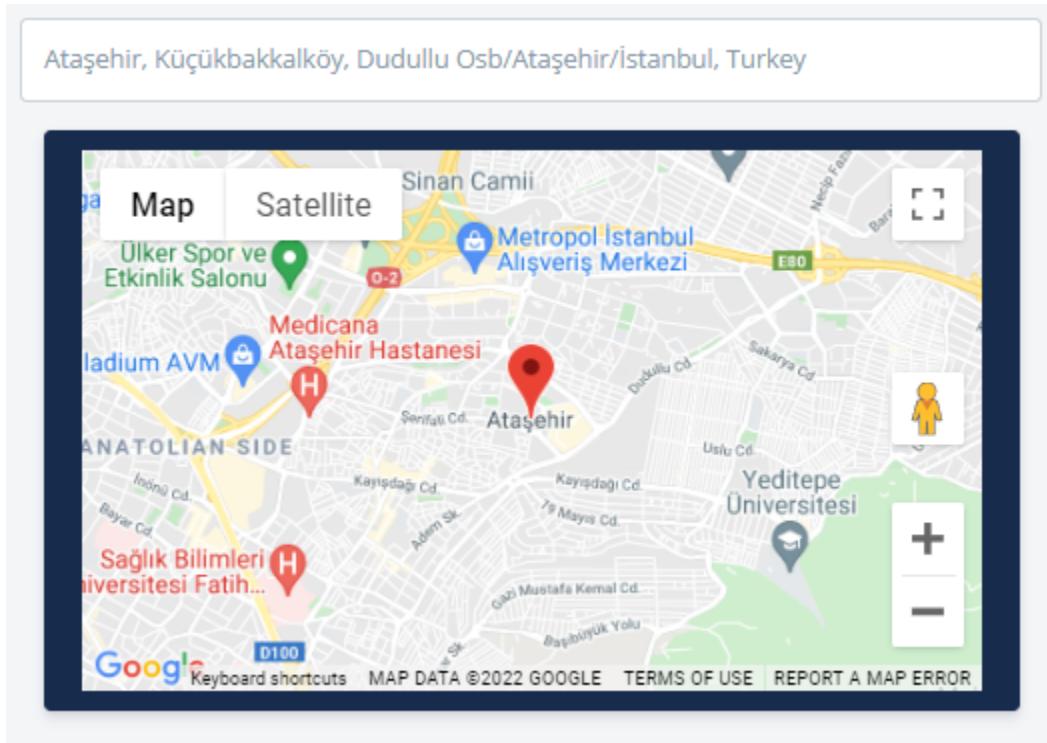


On the registration page, the user has the option to choose from multiple tags that are predefined by the system. Users can choose multiple tags.

Location

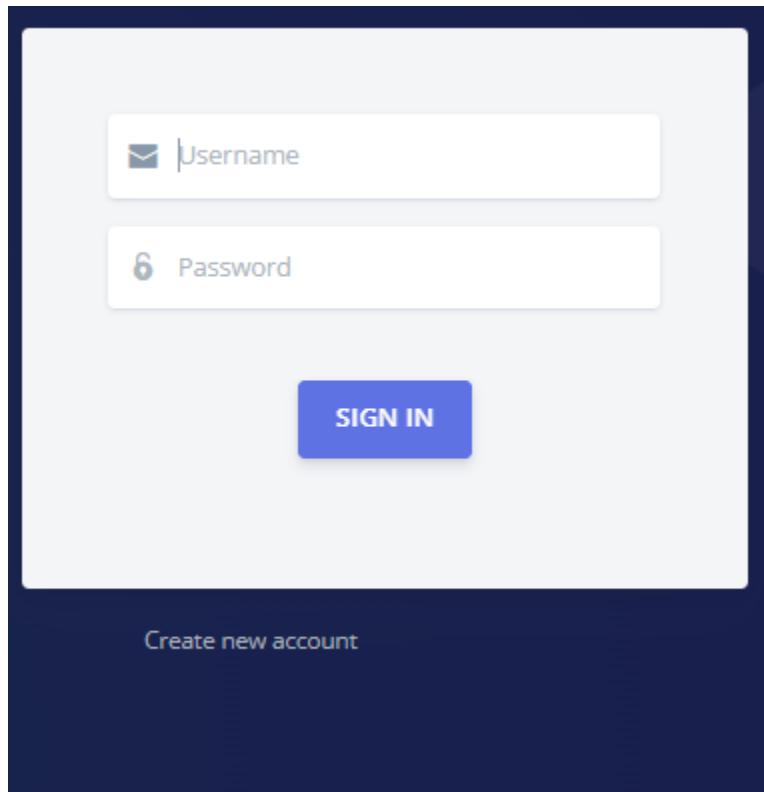


The user can enter their location information into the textbox. After selecting their location, their selected location will be displayed on the map.



After entering all the information and clicking the create button. The application will register the user and the api will return a jwt token unique to the user. The token will then be stored on the system and the user will be redirected to the homepage. The registered will have 5 credits on their balance which can be displayed on the profile page.

Login



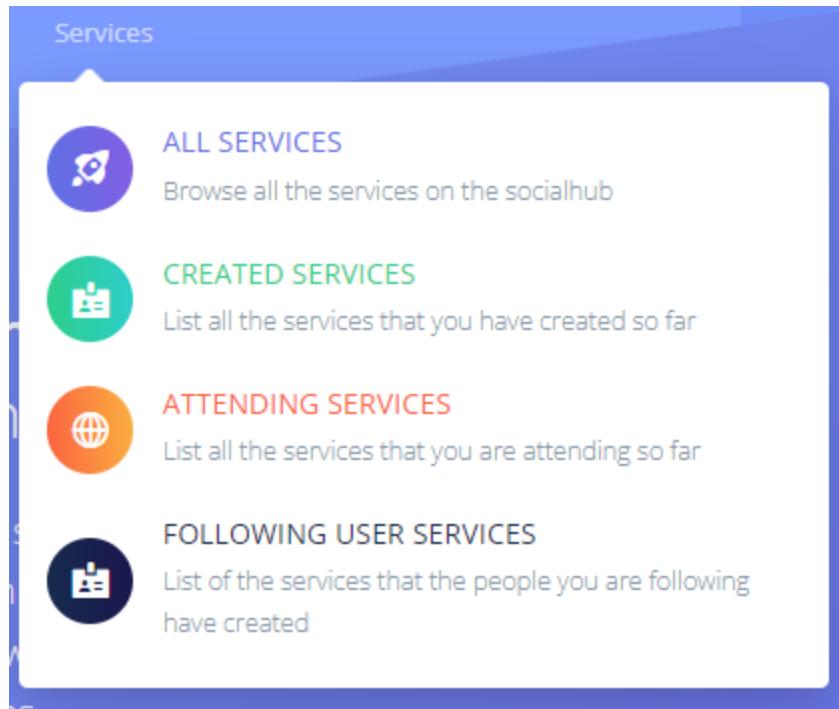
Users can log in to the system by clicking the login buttons on the homepage. Entering a username and password is sufficient to enter the system.

Navbar

Navigation bar has user information navigation on the right and the service links to the left and also the logo for the application.



Services Navigation



The Services button is a dropdown button. When hovered over, it displays a menu with detailed information on the links. The user can choose between different pages to navigate to.

Notifications

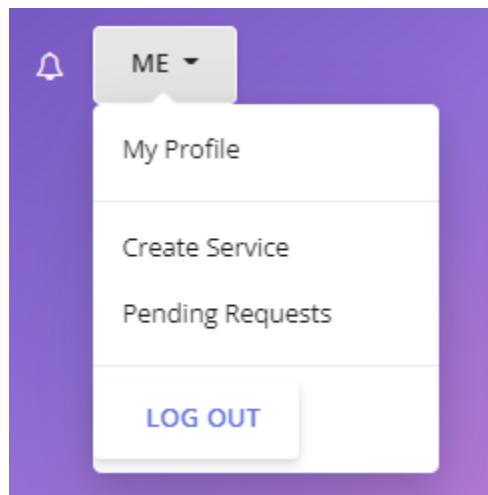
Navigation on the right has the notification button. If there are unread notifications, the bell icon will be filled and the number of new messages will be displayed on the tooltip.

The icon and the tooltip will change if all notifications are read.



User Button

If clicked, the user profile titled “me” will open a menu. The menu has links to user’s own profile page, new service and pending requests.



Services Pages

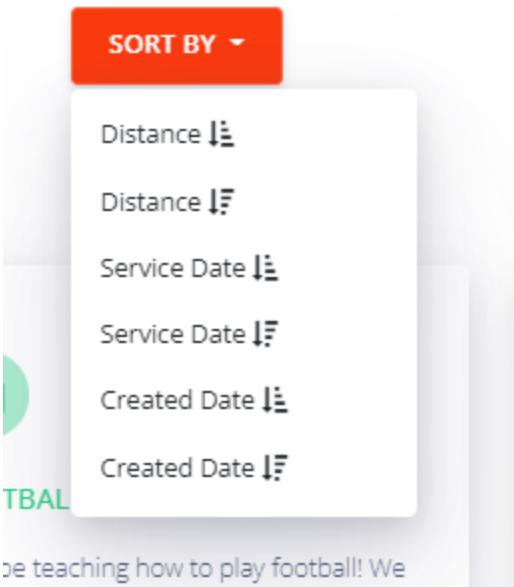
Different service viewing pages can be found at the navigation bar’s services dropdown button.

All Services

The screenshot shows the 'All Services' page with a header featuring a circular icon with a building and the text 'All Services'. Below it is a message: 'Here you can find all the active services. Go explore!' with a small potted plant icon. A 'SORT BY' button is visible. Three service cards are listed:

- FILM ANALYSIS**: Description: 'I will be teaching film analysis. This is a service that is open to people who do not have any experience in film analysis.' Credits: 2 CREDITS, Distance: 0.18 KM. Button: LEARN MORE.
- FOOTBALL!**: Description: 'I will be teaching how to play football! We can have a small match afterwards as well.' Credits: 3 CREDITS, Distance: 19.69 KM. Button: LEARN MORE.
- PET MY DOG**: Description: 'Well technically this is a service from my dog but anyways you can come to Maćka Park and pet my cute dog. He won't bite(I can't promise). He's definitely worth your time.' Credits: 1 CREDITS, Distance: 5.56 KM. Button: LEARN MORE.

Clicking the “all services” button on the navbar will redirect the user to the all services pages. The services will now display the distance to the user along with the title, description and the credit information. The user can sort the list with different sorting filters. All **future services** will be listed in this page.



When a sorting filter is clicked, the list will be re-ordered.

Created Services

Created services page will display all services that have been created by the user regardless of their date.

Attending Services

Created services page will display all services that are being attending by the user regardless of their date.

Following User Services

Following user services page will display all services that are being offered by the user's following users.

Service Detail Page

Service detail page holds information on the service. The information provided by the service page are; number of approved participants, service date, total quota, number of credits, title, description, tags, location and the creator's link.

The screenshot shows a service detail page with the following elements:

- Participants:** 4
- Quota:** 20
- Credits:** 2
- SEND REQUEST TO JOIN** button (highlighted in blue)
- Title:** Film Analysis by joshua
- Map:** A Google Map showing the location of the service in Istanbul, specifically in the Bebek area. Key landmarks visible include the Istanbul Sapphire AVM, Profilo AVM, Zorlu Center, and the Rumeli Hisarı Temporaril. The map also shows the Bosphorus and various bridges.
- Date and Time:** Wed, 19 Jan 2022 18:00
- Description:** I will be teaching film analysis. This is a service that is open to people who do not have any experience in film analysis.
- Tags:** MOVIES, ARTS

Request Buttons

The button on the upper right corner changes behavior based on the service. If there are no approved or pending requests made by the user, the button will be “send request” if a request is sent, the button will read “a request has already been sent” and if the user is the creator there will be no button. Clicking on the button will initiate the service request procedure.

[SEND REQUEST TO JOIN](#)

[ALREADY REQUESTED TO JOIN](#)

Service Over Button

There are two cases where a service over button is displayed.

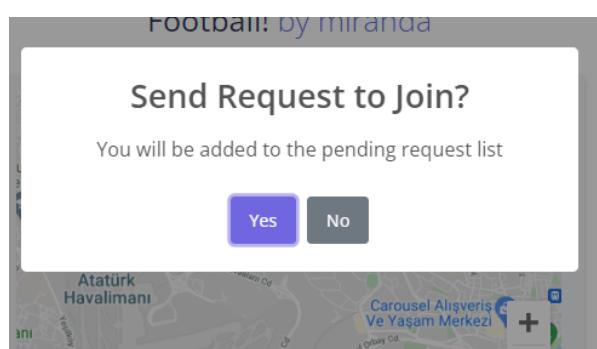
- The service is in the past, the service is in the “ONGOING” state and the creator of the service is logged in.
- The service is in the past, the service is in the “APPROVEDBYCREATOR” state and the attendee is logged in.

[SERVICE IS OVER?](#)

Clicking the link will prompt the user to confirm that the service is over.

Requesting a Service

Requesting to join a service will prompt the user to confirm that they would like to join the service. If the user clicks yes the request will be sent. If there is a balance available, a success message will be displayed that the request has been made and the page will be refreshed. A notification will be sent to the creator of the service at the end of the transaction.



Balance calculation when requesting a service

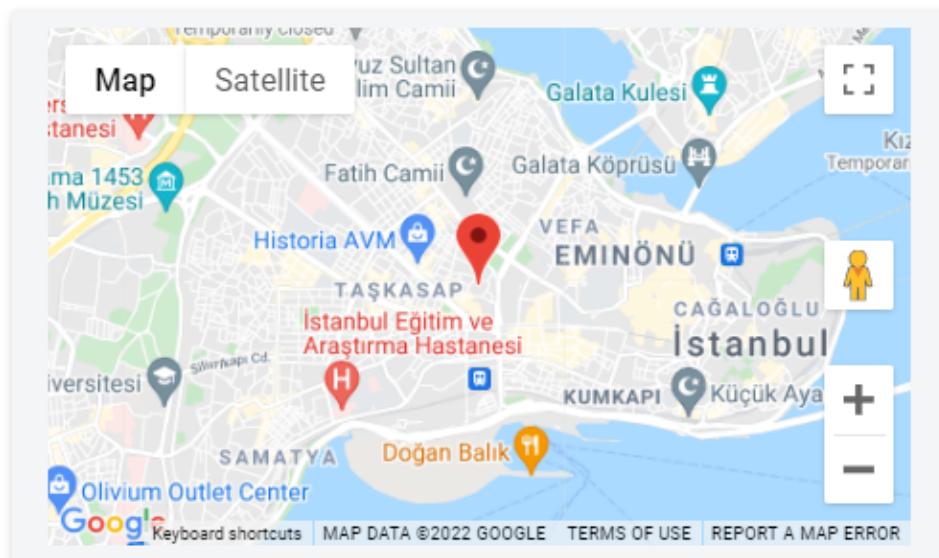
A calculation is made on the server side when a user requests a service. The calculation is made by adding the current balance and the balance of offered service and subtracting requested services' balances. Balances that would be gained or removed are on hold by the system. These can be displayed on the profile page.

| | | | |
|-----------|-----------|----------|-----------------------|
| 0 | 0 | 5 | 3 |
| Following | Followers | Credits | Credits on Hold |

Completing a Service

There is a handshake when completing a service. After the time of a service has passed, the show service button will be displayed to the creator on the service detail page.

Eminönü Tour by miranda



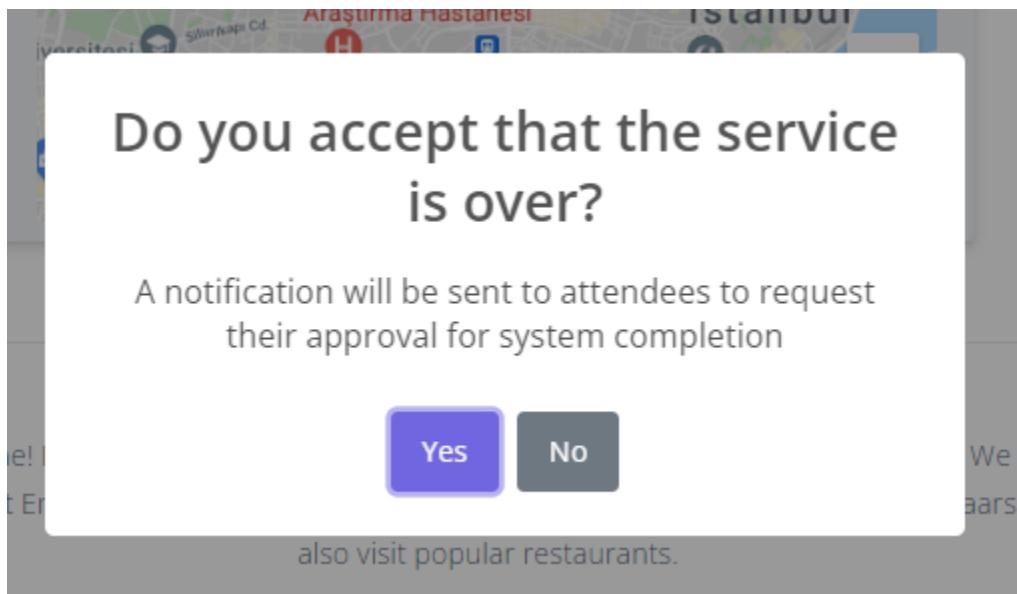
⌚: Wed, 15 Dec 2021 12:00

Hey everyone! I'm a professional tourist and I would like to give you a tour of Eminönü. We will start and finish at Eminönü Meydan. We will be visiting many historical places as well as bazaars. We will also visit popular restaurants.

MISC

SERVICE IS OVER?

If the button is clicked, the user will be prompted to confirm that the service is over.



Clicking yes will call the approve service method. This method will put the service on “APPROVEDBYCREATOR” state. **Pending user approvals will automatically be denied** by the system and a notification will be sent to requestees. Approved users will receive a notification that the creator has completed the service and that they need to confirm that the service is over.

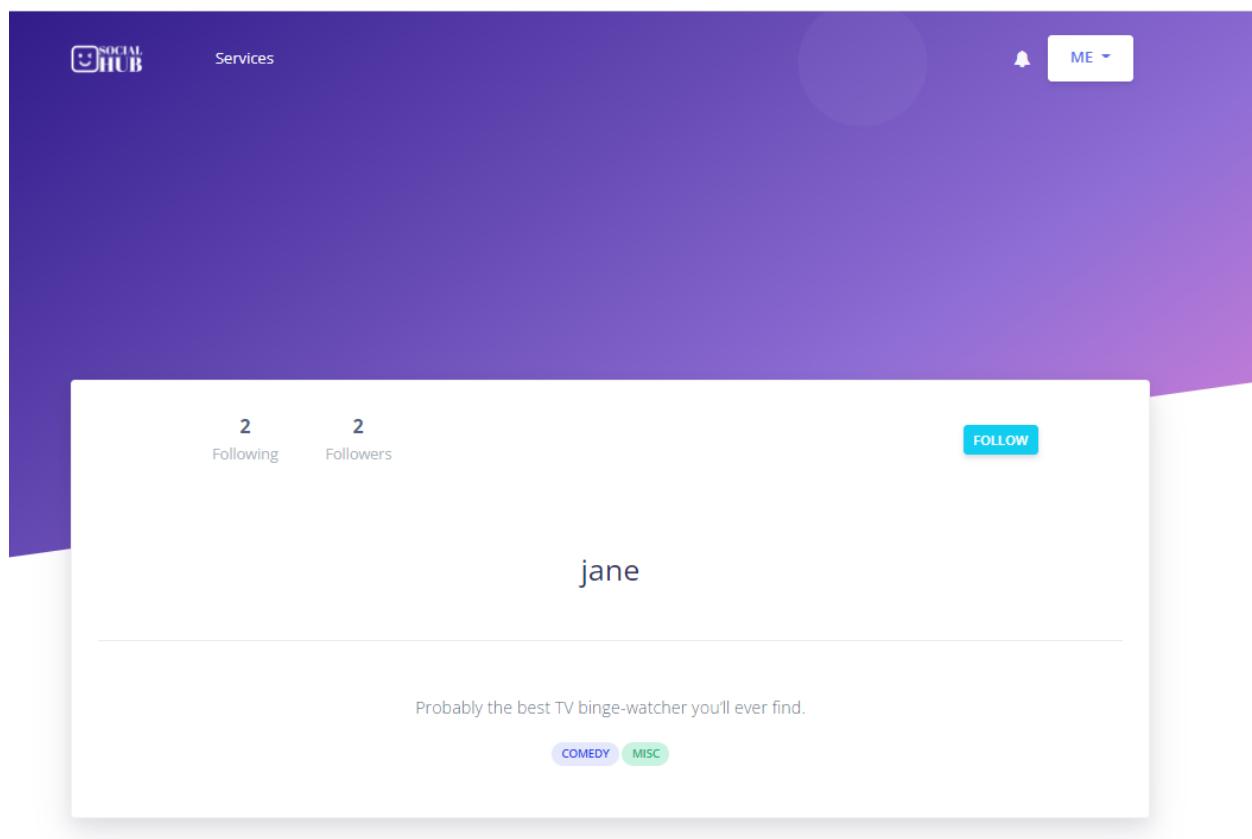
Same button will be visible to the attendees of the service if the creator has approved that the service was over. Users will be prompted to confirm that the service is over if they click the button.

Balance Update

After an attendee confirms the service is over, a notification is sent to the creator and the balances are updated. The service’s credit will be reduced from the balance of all attendees and the service credit amount will be added to the creator’s balance.

User Profiles

User pages can be viewed by all registered users. Users will see different information on their own profile page and others’ profile page.



Thank you for visiting socialhub!
You can visit the source code from link

The user page will display information about how many people are following the user and how many the user is following, their username, their bio and their interests. If the logged in user is not following the user, a button to follow them will be displayed on the page. If they are already following the user, a label stating that they are already following them will be displayed instead.

1 Following 4 Followers 6 Credits 1 Credits on Hold

miranda

Gamer. Award-winning music buff. Social media maven. Zombie fan. Student. Professional internet fanatic. Thinker. Freelance baconaholic.

ARTS MISC

If the user is displaying their own profile page, their credit will be displayed on their page on top of the basic information. No following button or label will be displayed.

Balance on hold

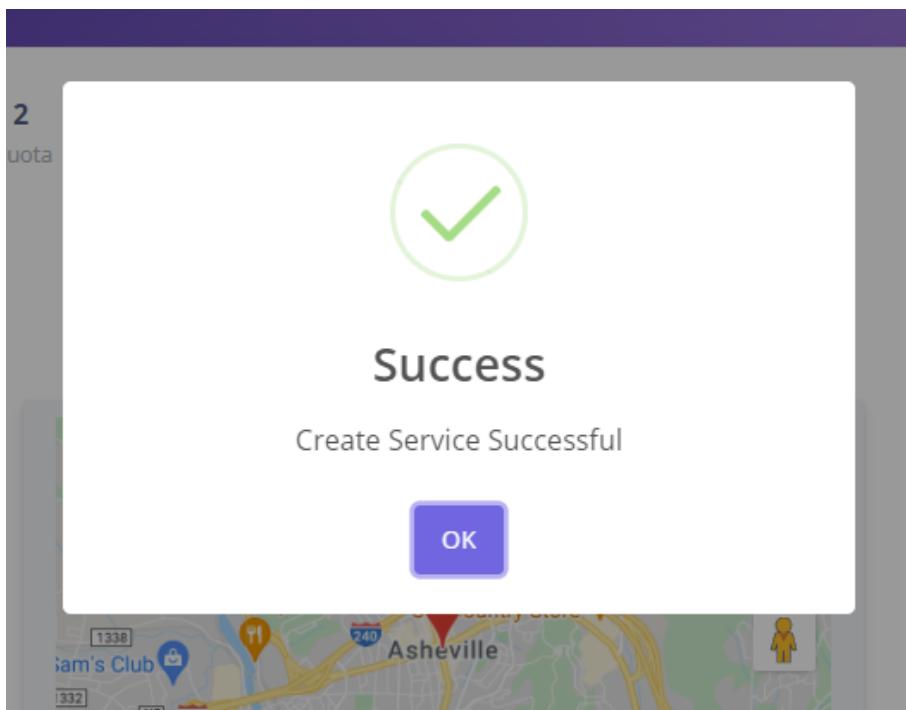
System holds the balance of the users if they offer or request a service. The calculation is made by adding the current balance and the balance of offered services and subtracting requested services' balances.

Creating a Service

Users can create services by clicking the create service button under the profile button on the right of the navbar.

The screenshot shows a web application interface for creating a service. At the top, there is a header with the SocialHub logo, a 'Services' link, a notification bell icon with '44' notifications, and a 'ME' dropdown menu. The main title 'Create a Service' is displayed prominently, followed by the subtitle 'contribute to socialhub'. A descriptive text explains that users can create services to contribute to the community and gain time balance based on service duration. Below this, there are several input fields: 'Header' (with a placeholder 'Write a description of the service here ...'), 'Quota' (with a date input field showing '2018-01-01'), 'Credits' (empty), 'Pick a tag' (a dropdown menu), and 'Enter a location' (empty). A large green 'CREATE SERVICE' button is located at the bottom center. The background features a blue gradient with white diagonal stripes.

A service header, description, date, quota, credit amount, tags and location can be entered into the system. Creating a system will return a success message and the user will be forwarded to the created service's page.



Tags

A screenshot of a tag selection interface. At the top, there are two green buttons with white text: 'movies x' and 'sports x'. Below them is a search bar with the placeholder 'Pick your interests'. A scrollable list of tags follows, with some items highlighted in green. The visible tags are: movies, arts, sports, comedy, misc, education, and nature. The 'education' tag is currently selected, as indicated by the green background.

On the registration page, the user has the option to choose from multiple tags that are predefined by the system. Users can choose multiple tags.

Location

The screenshot shows a search interface for locations starting with 'Ata'. The search bar at the top contains the text 'Ata'. Below the search bar is a list of five results:

- 📍 Ataşehir Küçükbakkalköy, Dudullu Osb/Ataşehir/İstanbul, Turkey
- 📍 Atal Tunnel Atal Tunnel, Burwa, Himachal Pradesh, India
- 📍 Atal Tunnel Burwa, Himachal Pradesh, India
- 📍 Atana Hotel Hessa Street - Dubai - United Arab Emirates
- 📍 Atascadero CA, USA

A small 'powered by Google' logo is visible in the bottom right corner of the interface.

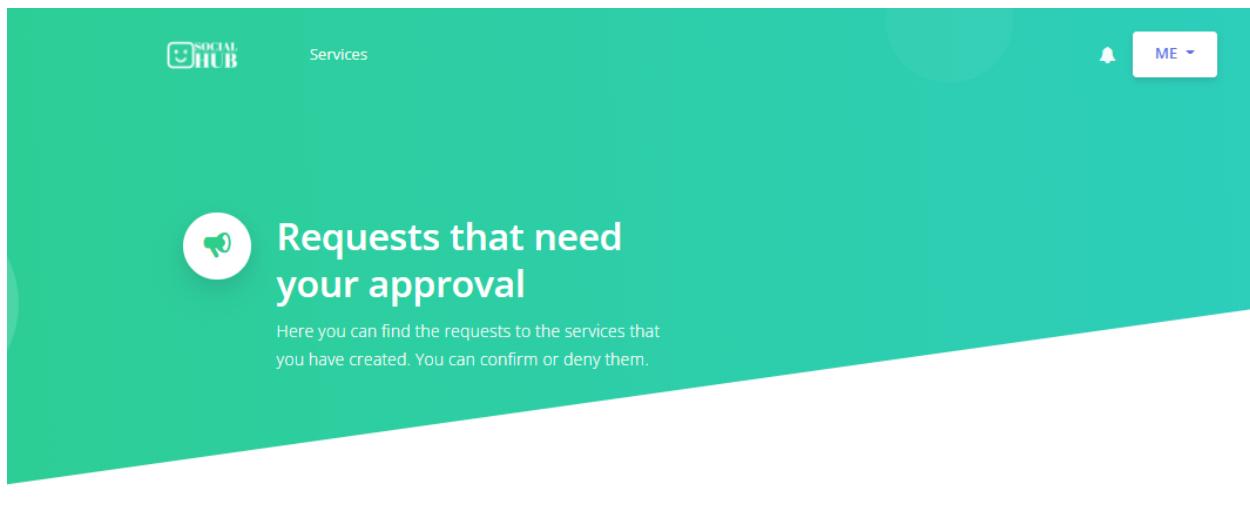
The user can enter their location information into the textbox. After selecting their location, their selected location will be displayed on the map.

The screenshot shows a Google Maps interface with the following details:

- The search bar at the top displays the selected location: "Ataşehir, Küçükbakkalköy, Dudullu Osb/Ataşehir/İstanbul, Turkey".
- The map view shows the area around Ataşehir, featuring landmarks such as "Metropol İstanbul Alışveriş Merkezi", "Medicana Ataşehir Hastanesi", and "Ülker Spor ve Etkinlik Salonu".
- The map includes standard controls for zooming in and out (+/-) and a location marker icon.
- At the bottom of the map, there is a footer with links: "Google", "Keyboard shortcuts", "MAP DATA ©2022 GOOGLE", "TERMS OF USE", and "REPORT A MAP ERROR".

Service Request Approval

Requests made to a service will be displayed on the creator's page. Users can see the requests made to all their services on the pending requests page. Clicking on the pending requests button under the profile button will redirect the user to pending requests page.



The page will list the header of the service, requester's name and the buttons to approve or deny the request. Clicking on the buttons will send a request to the server and an alert will be displayed on the page.

Approve

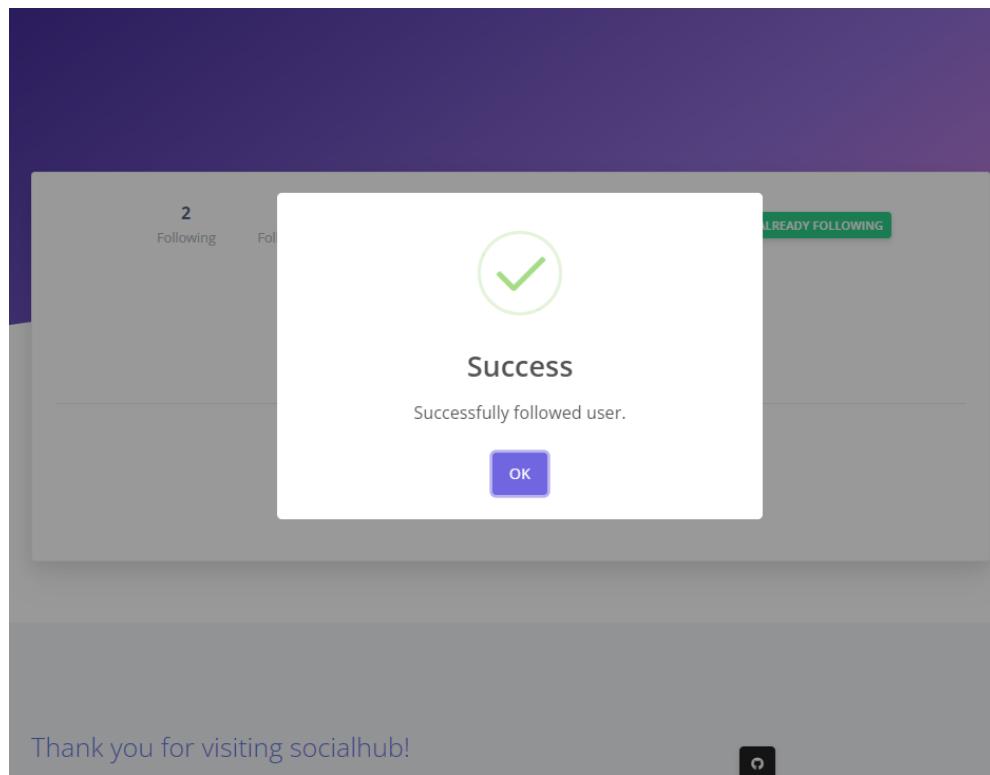
If the creator approves the system, the status will be updated to "APPROVED". A notification will be sent to the requester and their balance on hold will increase.

Deny

If the creator approves the system, the status will be updated to "DENIED". A notification will be sent to the requester.

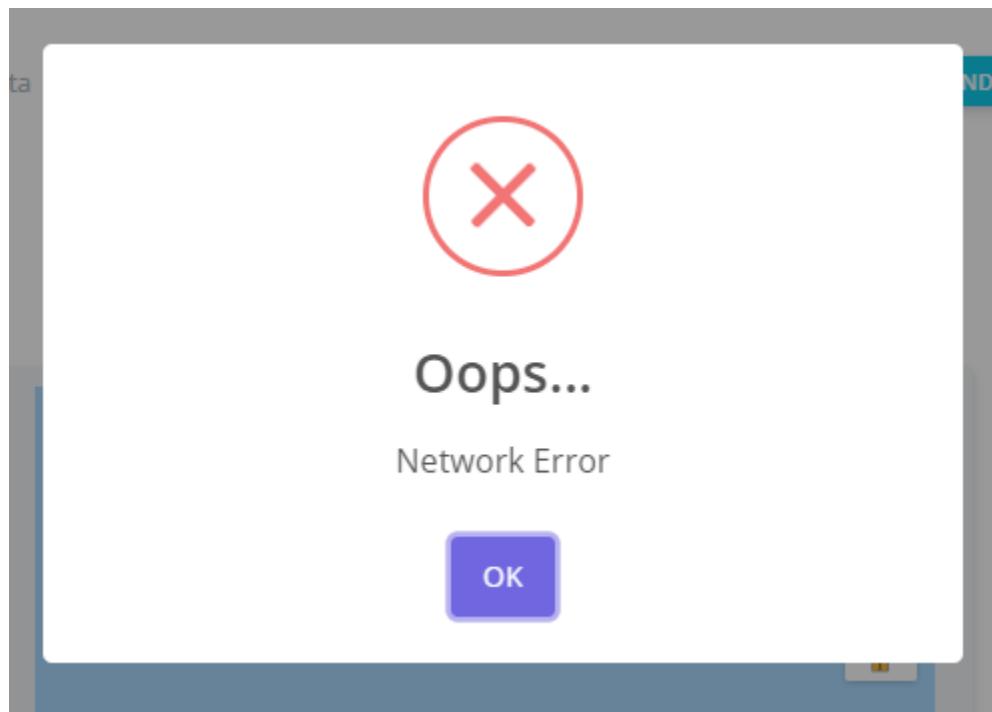
Success messages

Success messages will be displayed on a popup message



Error messages

Errors will be displayed on a popup message



Testing

There are unit tests, manual tests. Detailed postman collections can be used to view request details. The postman collection documentation is very useful as it lists all requests with its headers and it can be found at:

<https://documenter.getpostman.com/view/7873476/UVXesdff>

The screenshot shows a Postman collection interface with the following structure and endpoints:

- SWE573-api**
 - Service**
 - GET GetAllServices
 - GET GetServiceById
 - POST CreateService
 - GET GetByUser
 - GET GetServiceFiltered
 - GET GetAllServicesSorted
 - Tags**
 - POST New Tag
 - GET Get Tags
 - DEL Delete Tag
 - User**
 - GET Get All Users
 - POST Login
 - GET Get User Service Details
 - GET GetUserServiceDetails
 - GET GetProfile
 - GET GetProfile By Id
 - Approval**
 - GET SendRequest
 - Notification**
 - GET GetNotificationByUser
 - POST Register

Postman collection

Unit test results are as follows:

| | |
|--|--------|
| ✓ socialhub (com.swe573) | 214 ms |
| ✓ NotificationTest | 64 ms |
| ✓ mapToDto_ReturnsSameProperties() | 53 ms |
| ✓ contextLoads() | 11 ms |
| ✓ ApprovalServiceUnitTests | 58 ms |
| ✓ UserServiceApproval_ShouldThrowError_IfCreditE | 58 ms |
| ✓ UserServiceUnitTests | 63 ms |
| ✓ FollowUser_ShouldReturnEntity() | 21 ms |
| ✓ MapToDto_ShouldReturnSameFields() | 7 ms |
| ✓ FollowUser_ShouldThrowError_IfAlreadyFollowing | 5 ms |
| ✓ contextLoads() | 4 ms |
| ✓ Service_GetBalance_ShouldReturn_BalanceToBe() | 8 ms |
| ✓ Register_ShouldReturnEntity() | 7 ms |
| ✓ Register_ShouldThrowError_WhenDataIsInvalid() | 11 ms |
| ✓ ServiceServiceUnitTests | 29 ms |
| ✓ Service_ShouldThrowError_IfCreditAboveThresho | 25 ms |
| ✓ contextLoads() | 4 ms |

Unit test results

Useful Information to Test Deployed Application

All predefined users' passwords are 1. The user "miranda" has a lot of seed data that could make testing easier.

"Miranda" has a service called "Eminönü Tour" that was in the past which could be useful to test the handshake mechanism. The user "jane" is one of the attendees.