The ICT University

Spring 2024 Final Semester Exam

School of ICT

Course code: **SEN2241**

Course Title: OBJECT ORIENTED ANALYSIS, DESIGN AND IMPLEMENTATION

Instructor: Eng. Tekoh Palma

Link to GitHub Repository:

Backend: https://github.com/NganeEmmanuel/Socialize.git

Frontend : https://github.com/NganeEmmanuel/socialize-client.git

Deployment: https://socialize-client-production.up.railway.app/

Group Leader: Tchamabe Martine Carole

Group Information

SN	Member's Name	Registration Number	Team Role
1	TIMA CHEGUEM CLAUDE	ICTU20222933	FRONTEND DEV
2	CHOUATEN FOYET YANN RAEL	ICTU20222956	FRONTEND DEV
3	CHI TITI ROSITA LOULIA	ICTU20223194	FRONTEND DEV
4	NDE HURICH DILAN	ICTU20223351	BACKEND DEV
5	YOMEH MARC BRIAN	ICTU20223131	BACKEND DEV
6	AGHUKE DE NGEH BRIYAND	ICTU20222984	FRONTEND DEV LEAD
7	NGANE EMMANUEL	ICTU20222972	PRODUCT OWNER
8	TCHAMABE MARTINE CAROLE	ICTU20223095	SCRUM MASTER

This document contains (51pages)

SOCIALIZE DOCUMENTATION

CHAPTER ONE: INTRODUCTION

1.1General Introduction:

Socialize is a specialized social media web application designed for the ICTU community. Unlike traditional social media platforms, Socialize focuses on fostering connections, collaboration, and engagement among students, Lecturer, and administrative staff within a school environment. The platform aims to create a safe and nurturing digital space where users can share their experiences, seek academic and career guidance, and build meaningful relationships with their peers and educators.

1.2Scope:

The platform's scope covers the core functionalities of a social media application, including user profiles, content creation and sharing, networking, community engagement, and privacy/safety features.

1.3 Aim and Objectives:

The primary aim of Socialize is to enhance the social and academic experience of the school community by providing a dedicated platform for communication, collaboration, and content sharing. The key objectives of the Socialize platform are:

1. To enable students, Lecturers, and staff to connect with each other, share their work, and engage in discussions within a secure and moderated environment.

4. To improve information sharing and timely notification of important events, announcements, and activities within the school community.

5. To provide a user-friendly and responsive platform that is accessible across a variety of devices, ensuring seamless access for all users.

6. To prioritize user privacy and safety by offering robust moderation tools and customizable privacy settings.

7. Participate in discussions and connect with like-minded individuals within interest-based groups

SOCIALIZE DOCUMENTATION

Problem Statement:

Existing social media platforms often lack the specific features and functionality required to cater to the unique needs of a school community. These platforms may not provide the

necessary safeguards, content moderation, and targeted networking opportunities that are essential for a safe and productive educational environment. Additionally, the general nature of these platforms may not adequately address the specific communication, collaboration, and engagement requirements of students, teachers, and school administrators. To address this gap, Socialize aims to develop a tailored social media solution that meets the unique needs of the school community, fostering a secure and enriching digital ecosystem for all users.

Chapter Two: Literature Review

2.1Software Development Methodologies:

There are various software development methodologies that have been widely adopted in the industry. Some of the prominent ones include:

1. **Waterfall Methodology:**

- A linear, sequential approach to software development.
- Emphasis on thorough planning and documentation upfront.
- Rigid, with limited flexibility to accommodate changes during the development process.

2. **Agile Methodology:**

- Iterative and incremental approach to software development.
- Focuses on adaptability, collaboration, and rapid delivery of working software.
- Includes methodologies like Scrum, Kanban, Extreme Programming (XP), and others.

3. **Scrum Methodology:**

- An Agile framework for managing and completing complex projects.
- Emphasizes the use of cross-functional, self-organizing teams.
- Relies on short, time-boxed iterations (sprints) to deliver working software.
- Incorporates regular feedback and adaptation to changing requirements.

4. Kanban Methodology:

SOCIALIZE DOCUMENTATION

- A lean, visual approach to software development.
- Focuses on the continuous flow of work and the optimization of the development process.
- Emphasizes the visualization of the workflow and the limitation of work-in-progress.

Comparison between different Software Development Methodologies:

Each software development methodology has its own strengths, weaknesses, and suitability for different types of projects. The choice of methodology depends on factors such as the project's complexity, the level of uncertainty, the team's expertise, and the organization's culture.

Reason for the choice of Scrum Methodology for Socialize:

The Scrum methodology has been chosen for the Socialize project due to the following reasons:

1. Adaptability: The Scrum framework is well-suited for projects with evolving requirements, such as a social media platform, where user needs and preferences may change over time. This allows the Socialize development team to respond to changes quickly and effectively.

2. Collaboration: Scrum emphasizes the use of cross-functional, self-organizing teams, which aligns well with the collaborative nature of the school community that Socialize aims to serve. This facilitates effective coordination and communication among students, teachers, and administrators.

3. Transparency: Scrum's emphasis on regular feedback, retrospectives, and transparency helps ensure that the development process for Socialize is visible and accessible to all stakeholders, fostering trust and buy-in within the school community.

4. Iterative Delivery: The iterative nature of Scrum, with its short sprints and the ability to deliver working software incrementally, is beneficial for the Socialize project. This allows users to provide feedback and guide the development process, ensuring the platform meets their evolving needs.

SOCIALIZE DOCUMENTATION

5. Proven Track Record: Scrum is a widely adopted and well-established Agile methodology, with a proven track record of successful implementation in a variety of industries, including in the education and social media sectors. This experience provides confidence in the Scrum framework's suitability for the Socialize project.

By adopting the Scrum methodology, the Socialize project team can leverage its strengths to deliver a high-quality, user-centric social media platform that meets the unique needs and requirements of the school community.

Comparison between different Software Development Methodologies

Each software development methodology has its own strengths, weaknesses, and suitability for different types of projects. The choice of methodology depends on factors such as the project's complexity, the level of uncertainty, the team's expertise, and the organization's culture.

Reason for the choice of Scrum Methodology

The Scrum methodology has been chosen for the Socialize project due to the following reasons:

Adaptability: Scrum's iterative and incremental approach is well-suited for a social media platform like Socialize, where user needs and preferences are likely to evolve over time. This allows the development team to respond to changes quickly and effectively.

Collaboration: Scrum's emphasis on cross-functional, self-organizing teams aligns with the collaborative nature of the school community that Socialize aims to serve. This facilitates effective coordination and communication among students, teachers, and administrators.

Transparency: Scrum's focus on regular feedback, retrospectives, and transparency helps ensure that the development process for Socialize is visible and accessible to all stakeholders, fostering trust and buy-in within the school community.

Iterative Delivery: The iterative nature of Scrum, with its short sprints and the ability to deliver working software incrementally, is beneficial for the Socialize project. This allows users to provide feedback and guide the development process, ensuring the platform meets their evolving needs.

Proven Track Record: Scrum is a widely adopted and well-established Agile methodology, with a

SOCIALIZE DOCUMENTATION

proven track record of successful implementation in a variety of industries, including education and social media. This experience provides confidence in the Scrum framework's suitability for the Socialize project.

General review of related concepts with respect to Socialize

The Socialize project involves the development of a social media platform specifically designed for a school community. This introduces several related concepts that should be considered, such as:

- **Educational Technology**: Leveraging technology to enhance learning, collaboration, and communication within the school environment.

- **User-Centered Design**: Ensuring the platform is intuitive, accessible, and tailored to the needs of students, teachers, and administrators.
  Data Privacy and Security: Implementing robust measures to protect sensitive student and school-related information.

- **Community Engagement**: Fostering active participation and collaboration among members of the school community.
  Scalability and Performance: Ensuring the platform can handle the expected user base and traffic without compromising user experience.
  Review of related literature with respect to Socialize

Review of related literature with respect to Socialize

A thorough review of the existing literature on topics related to the Socialize project would provide valuable insights and best practices. This could include research on:

- **Successful implementation of social media platforms in educational settings**: Studies that explore the challenges, best practices, and outcomes of integrating social media into school environments.

- **Challenges and best practices in developing technology-based solutions for schools**: Research that addresses the unique requirements, technical considerations, and organizational factors involved in creating educational technology.

SOCIALIZE DOCUMENTATION

- **The role of collaboration and communication in enhancing learning outcomes**: Literature that examines how social media and collaborative tools can improve student engagement, knowledge sharing, and educational outcomes.

- **Student and teacher perceptions and attitudes towards the use of social media in education:** Investigations into the perspectives and concerns of key stakeholders regarding the integration of social media in the classroom.

- **Strategies for ensuring data privacy and security in educational technology**: Scholarly works that discuss the legal, ethical, and technical approaches to safeguarding sensitive student and school-related data.

- **Lessons learned from the adoption of Agile methodologies, particularly Scrum, in educational technology projects:** Case studies and analyses that highlight the successes, challenges, and best practices in applying Agile frameworks to the development of educational technology solutions.

3.RESEARCH METHODOLOGY AND MATERIALS

3.1 Research Methodology

To provide a thorough understanding of the research methodology for the Socialize project, the following key aspects will be addressed:

1. **Literature Review**:
 - A comprehensive review of the existing literature on related topics, as outlined in the previous section, to identify best practices, challenges, and relevant insights that can inform the development of the Socialize platform.

2. User Interviews and Focus Groups:
 - Engaging with the key stakeholders, including students, teachers, and school administrators, to gather in-depth feedback on their needs, preferences, and concerns regarding the use of a social media platform in an educational setting.

- 3. **Competitive Analysis:**
  - Evaluating existing social media platforms and educational technology solutions to identify their strengths, weaknesses, and the features that could be adapted or

SOCIALIZE DOCUMENTATION

improved upon for the Socialize project.

4. **Prototyping and User Testing**:
- Conducting user testing sessions to assess the platform's usability, functionality, and overall user experience.

5**. Agile Development Methodology**:
- Adopting the Scrum framework to manage the development process, with regular sprint planning, daily stand-ups, and retrospectives to foster adaptability and continuous improvement.

3.1.1 System Requirements

The social features for the Socialize platform include the following key requirements:

- **User Profiles and Connections:**
  Ability for users to create and manage their own profiles
  Functionality to connect with other users and build a social network

- **Social Feeds and Activity Streams:**
  Real-time updates and notifications of user activities and content sharing

- **Content Sharing and Discussions:**
  Capability to share various content types (images) with other users
  Threaded discussions and commenting features to facilitate collaboration and exchange of ideas

- **Notifications and Alerts:**
  Timely notifications to users about relevant activities, updates, and mentions
  Customizable notification preferences and settings

3.1.2 Feasibility Study for Socialize (TELOS)

Key Findings

Feasibility Analysis

SOCIALIZE DOCUMENTATION

Technical Feasibility:

The proposed social features is built using standard web application development practices and technologies, such as user account management, real-time data processing, and notification systems.

Economic Feasibility:

A detailed cost-benefit analysis is required to assess the economic feasibility and ensure alignment with the overall business strategy and financial goals of the Telos platform. Our development was cost free

Legal Feasibility:

Compliance with data privacy regulations is essential for the collection, storage, and usage of user data.

Content moderation policies and processes must be established to ensure appropriate user-generated content.

Adherence to intellectual property laws and guidelines for content sharing and distribution is necessary.

Operational Feasibility:

The social features should be designed to integrate seamlessly with the existing Telos platform features and workflows.

Effective community management strategies, including user engagement, scalability, and support, are crucial for the successful implementation and adoption of the social features.

Scheduling Feasibility:

The development and deployment of the social features were carefully scheduled, with a phased implementation approach (SCRUM) to manage complexity and risk.

3.2 System Requirements (Functional and Non-Functional)

The system requirements for the Socialize project are derived from the user stories and the research methodology described above. These requirements are divided into functional and non-functional categories.

3.2.1 Functional Requirements

SOCIALIZE DOCUMENTATION

3.2.1.1. **User Registration and Authentication:**

- Users (students, teachers, and administrators) can register and create accounts on the Socialize platform.
- Users can log in securely using their credentials.
- Users can reset their passwords and update their profile information.

3.2.1.2. **Social Networking Features**:

- Users can create and maintain personal profiles, including basic information, profile pictures, and bios.
- Users can connect with other members of the school community by sending and accepting friend requests.
- Users can post updates, share content (images, videos), and engage with posts through likes, comments, and shares.
- Users can form and participate in discussion groups or clubs based on shared interests or activities.
- Users can send private messages to individual friends or group members.

3.2.1.3. **Content Moderation and Reporting:**

- Users can interact with other users through comments and posts
- Users can report inappropriate or offensive content, which will be reviewed and addressed by the moderation team.
- Administrators can suspend, ban, or restrict user accounts based on violations of the platform's terms of service.

3.2.2 Non-Functional Requirements

3.2.2.1. **Usability and Accessibility**:

- The Socialize platform should be intuitive, user-friendly, and easy to navigate for users of all technical abilities.
- The platform should adhere to accessibility guidelines to ensure it is inclusive and accommodates users with diverse needs and abilities.

3.2.2.2. **Performance and Scalability**:

- The platform should be able to handle the expected user base and traffic without compromising response times or user experience.
- The platform should be designed to scale efficiently as the user base grows.

3.2.2.3. **Security and Privacy**:

SOCIALIZE DOCUMENTATION

- The platform should implement robust security measures to protect user data and prevent unauthorized access or data breaches.
- The platform should comply with relevant data privacy regulations and ensure the confidentiality of sensitive student and school-related information.

3.2.2.4. **Reliability and Availability**:
- The platform should maintain a high level of uptime and availability, with minimal downtime or service interruptions.
- The platform should have a comprehensive backup and disaster recovery plan to ensure business continuity.

3.2.2.5. **Interoperability and Integration**:
- The platform should be able to integrate with existing school systems, such as student information systems, learning management platforms, and communication tools.

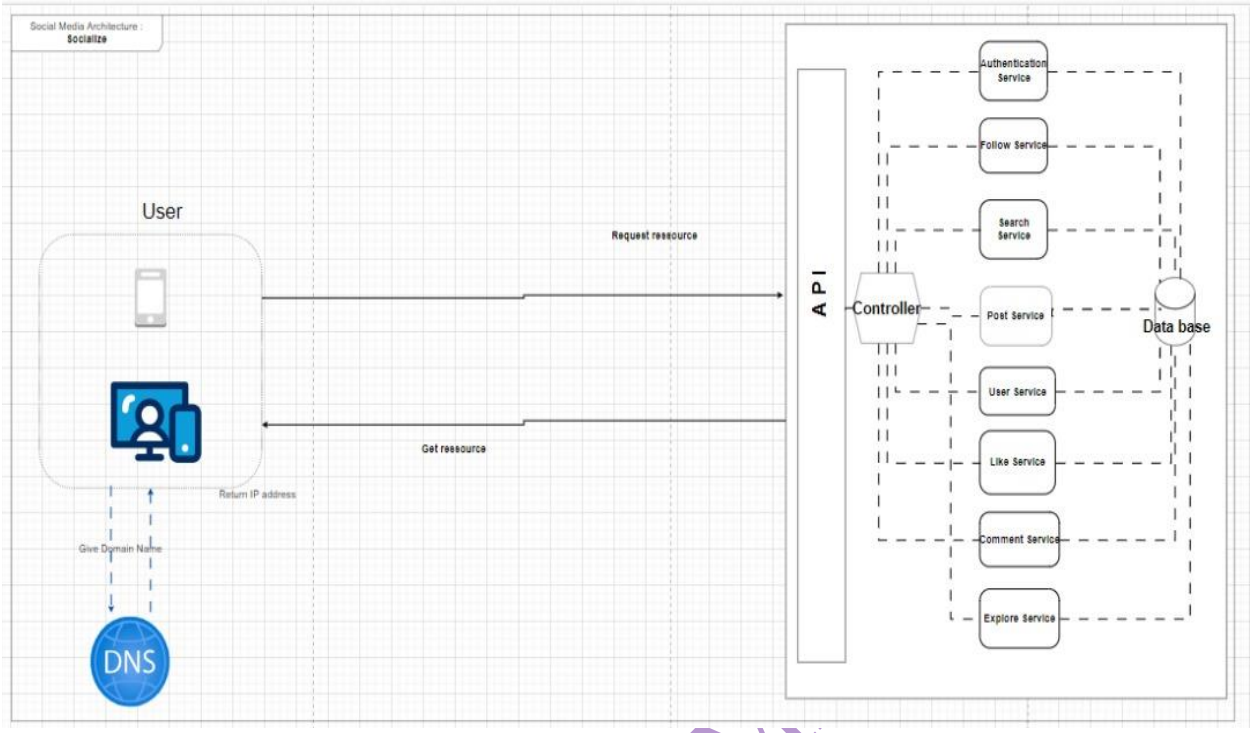
3.2.2.6. **Maintainability and Extensibility**:
- The platform is designed with a modular architecture to facilitate easy maintenance, updates, and the addition of new features in the future.
- The codebase adhered to industry best practices and coding standards to ensure long-term maintainability.

By defining these functional and non-functional requirements, the Socialize project team can ensure that the platform is designed and developed to meet the specific needs of the school community, while also considering broader technical and operational considerations.

SOCIALIZE DOCUMENTATION

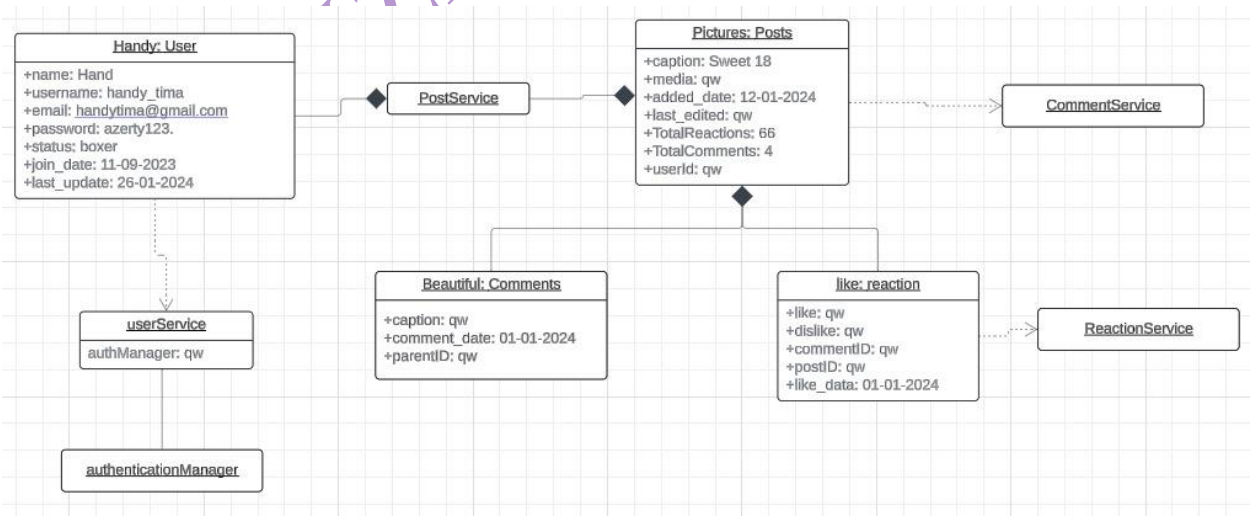
3.3. System Design

3.3.1. System's Architecture



3.3.2. UML Diagram

3.3.2.1 Object Diagram



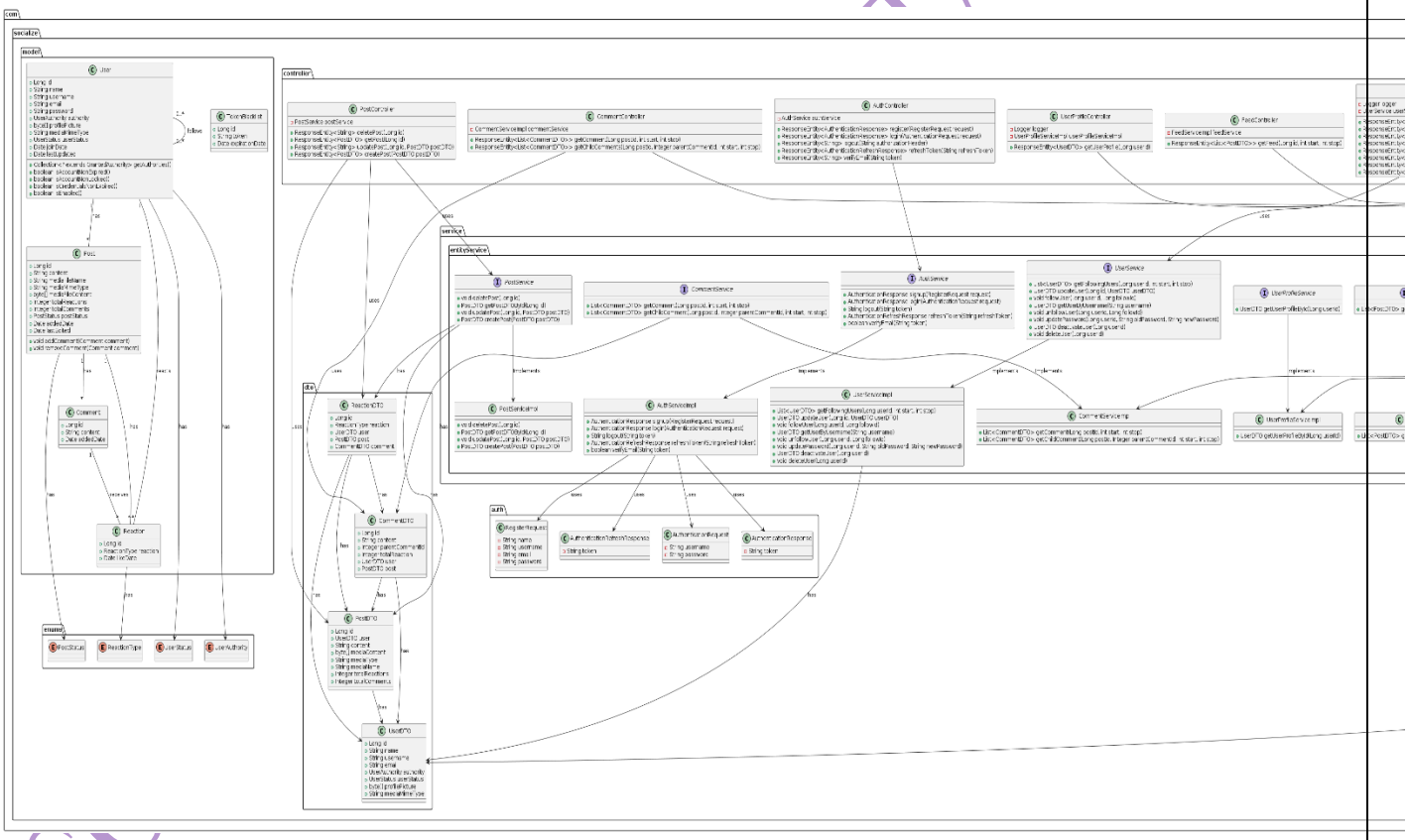
SOCIALIZE OBJECT DIAGRAM

3.3.2.2 Class Diagram

The Class Diagram represents the static structure of the system, including the classes, their attributes, and the relationships between them.

It helps in helps to design the data model and the underlying entities of the system.

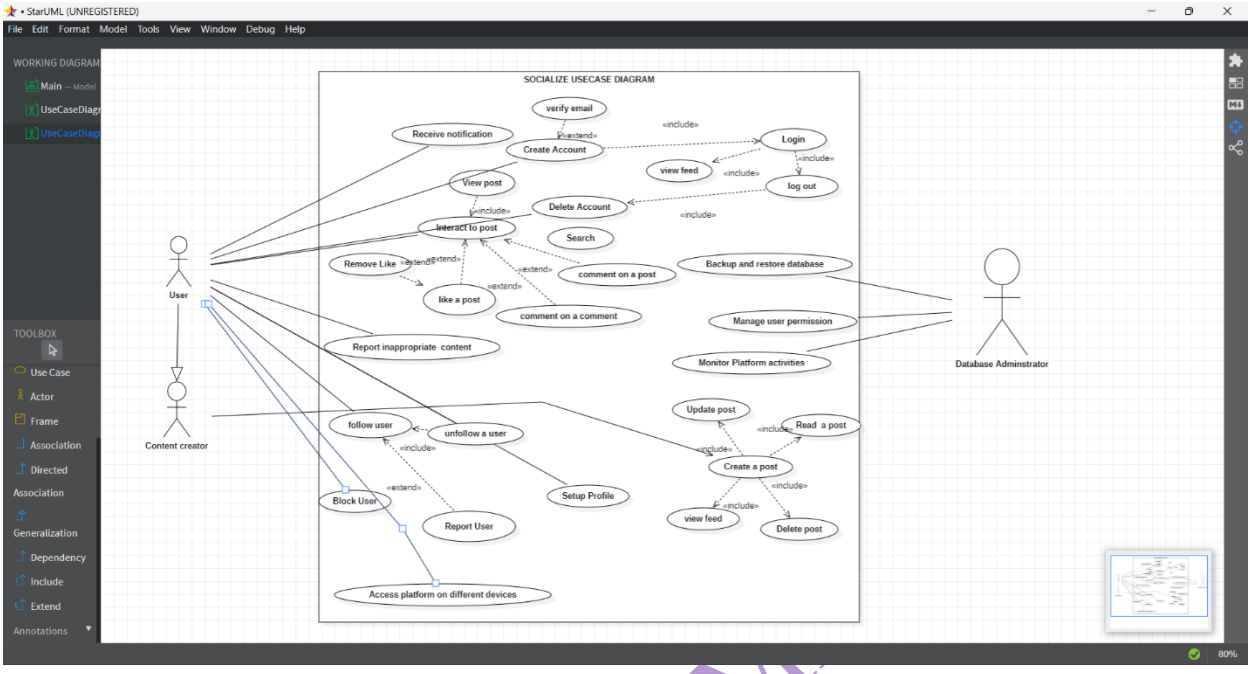
Also Supports the development of the system's database schema and the implementation of the business logic.



Socialize Class Diagram

3.3.2.3 Use Case Diagram

The use case diagram depicts the main functionalities of the system and the interactions between the actors (Content Creator, User, Database Administrator) and the system.



SOCIALIZE USECASE DIAGRAM

Use Case Description:

1.Create Account: The User can create a new account on the platform, which requires email verification.
2.**Login/Logout**: The User can log in and log out of their account.
3.**Delete Account**: The User can delete their account on the platform.
Create, Read, Update, and Delete Posts: The User can create, read, update, and delete their own posts.
4.**Set Up Profile**: The User can set up their profile on the platform.
5.**Search for Users and Posts**: The User can search for other users and posts on the platform.
6.**Remove Likes**: The User can remove their likes from posts.

7.View Existing Posts: The Content Creator can view a list of all the existing posts on the platform.
8.**Receive Notifications**: The User receives notifications about activities on the platform, such as new posts, comments, and messages.
9.**Interact with Posts**: The User can like, comment, and share posts.
10.**Report Inappropriate Content**: The User can report any content they consider inappropriate or in violation of the platform's guidelines.
11.**Block/Unfollow Users**: The User can block or unfollow other users to prevent interactions.
12.**Access Platform on Different Devices**: The User can access the platform on various devices,

SOCIALIZE DOCUM

and their account information is synchronized.
13.**Backup and Restore Database**: The Database Administrator can backup and restore the platform's database.
14.**Manage User Permissions**: The Database Administrator can manage user permissions on the platform.
15.**Monitor Platform Activity**: The Database Administrator can monitor the activity on the platform.

3.3.2.4 Sequence Diagrams

The sequence diagram models the dynamic behavior of the system, showing the interactions between the different components and the sequence of messages exchanged.

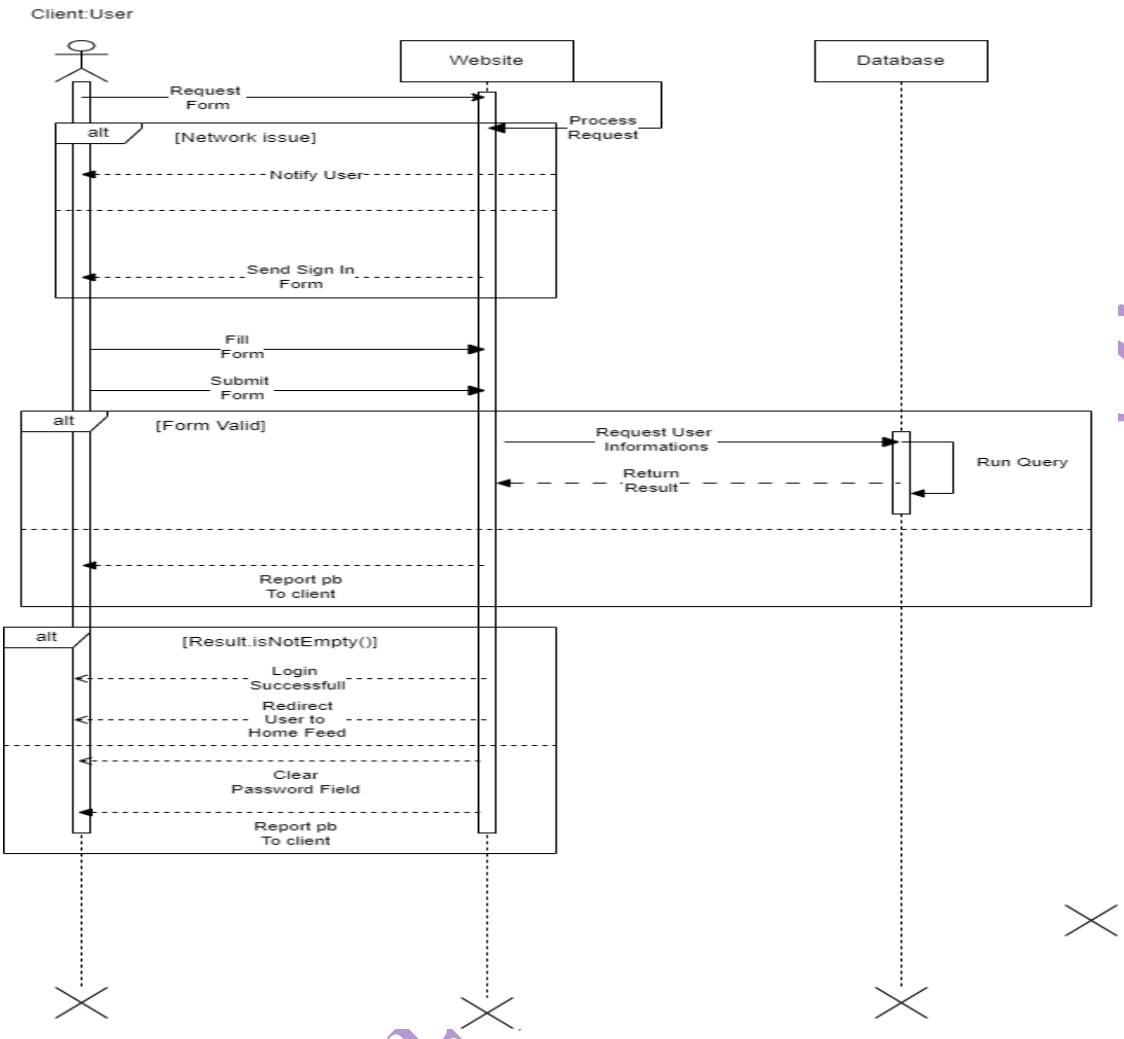
Supports the design of the system's API and the implementation of the communication mechanisms.

We used 5 use cases to model the behavior of 5 different sequence diagrams.

1.User Authentication

A-Sign-IN

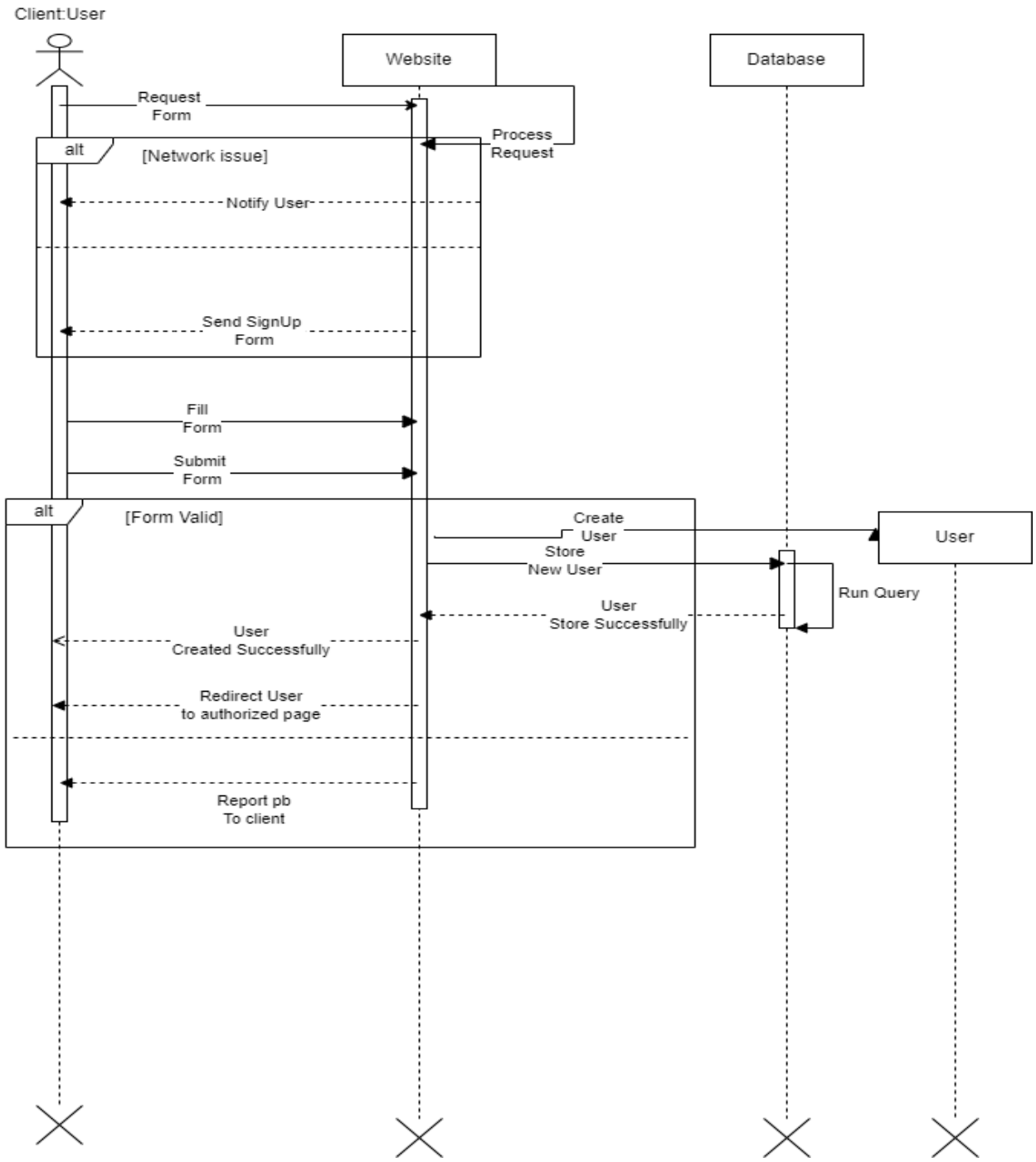
SOCIALIZE DOCUMENTATION



SIGNIN SEQUENCE DIAGRAM

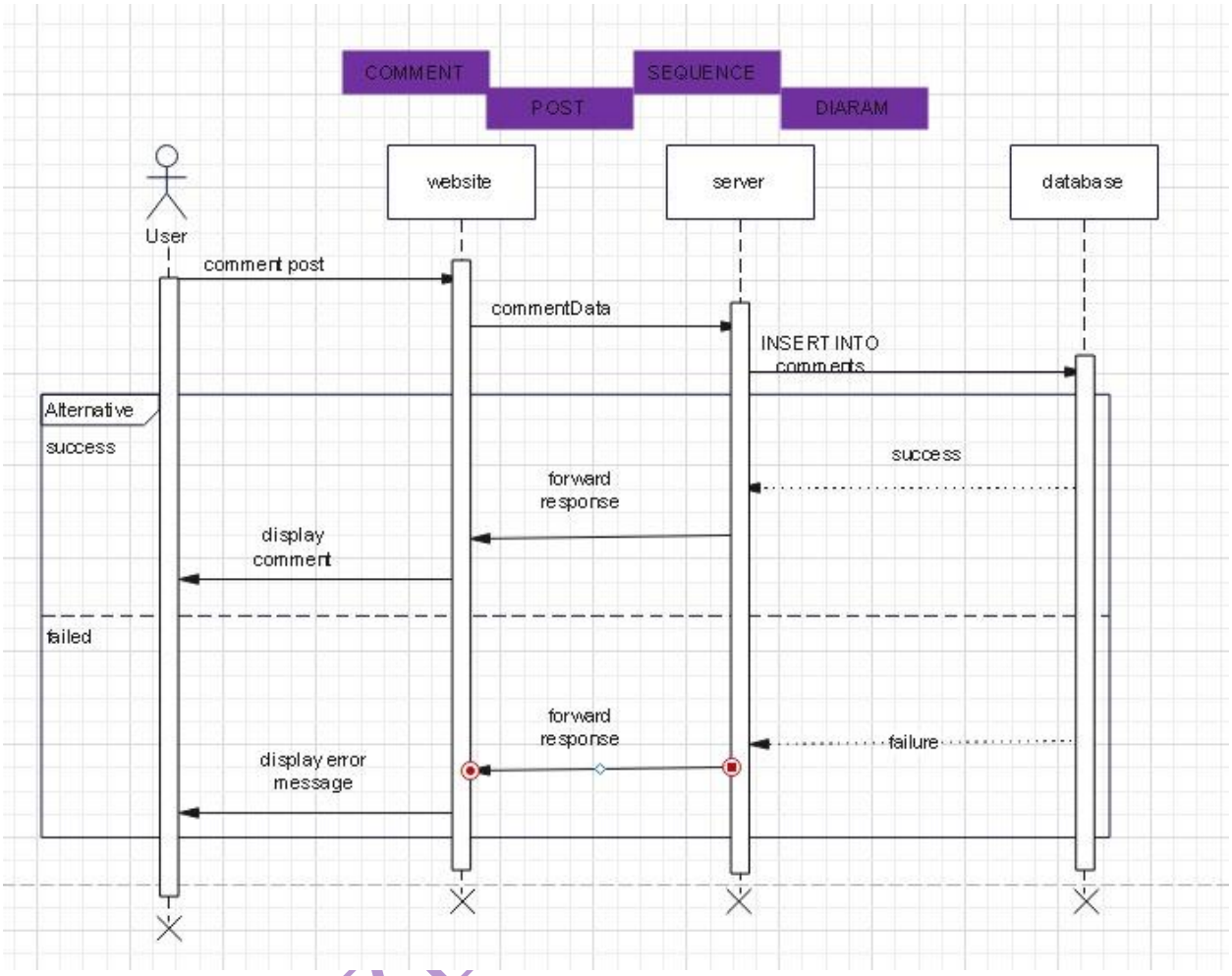
SOCIALIZE

B-Sign-Up



SIGNUP SEQUENCE DIAGRAM

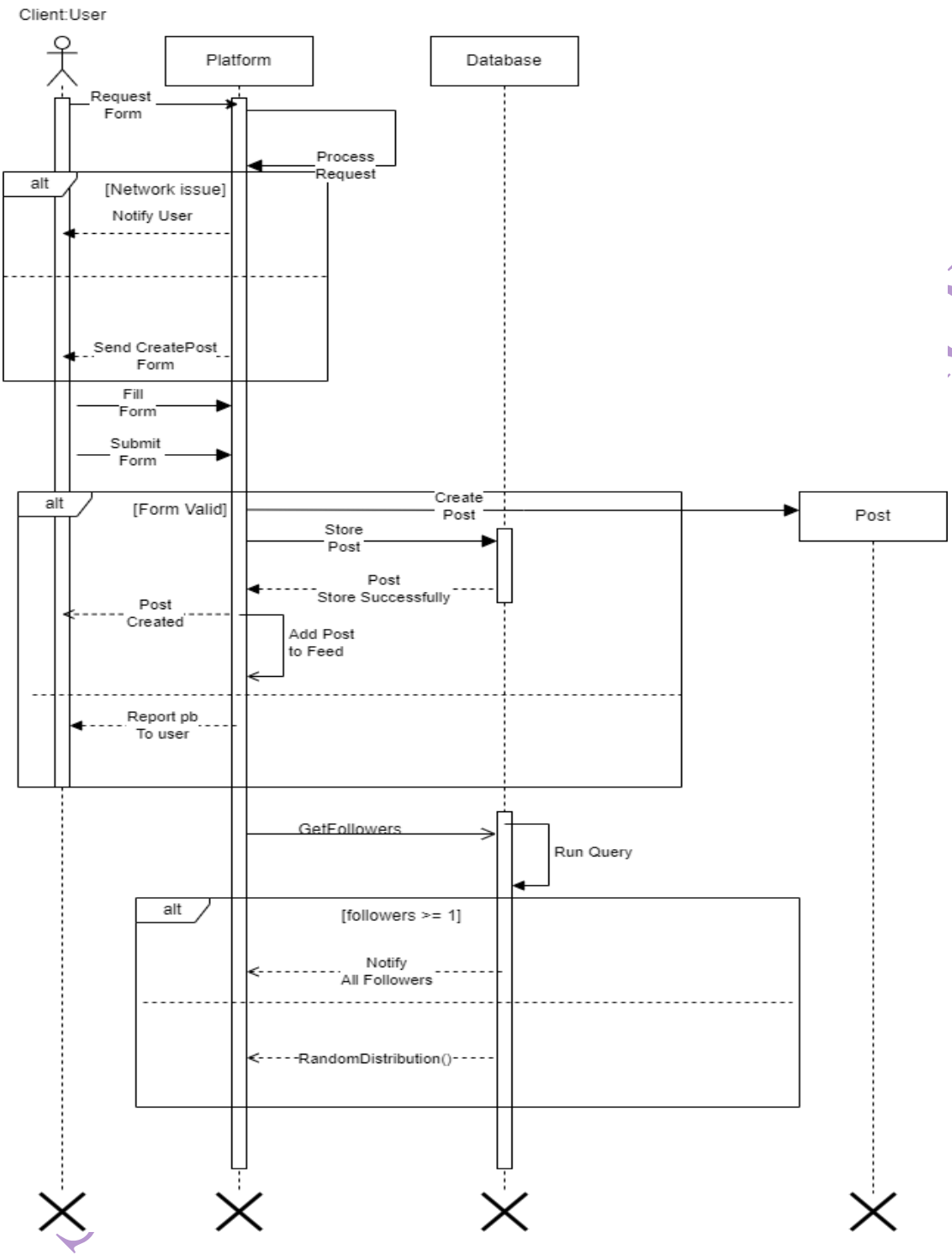
2-Comment on a post



COMMENT ON A POST SEQUENCE DIAGRAM

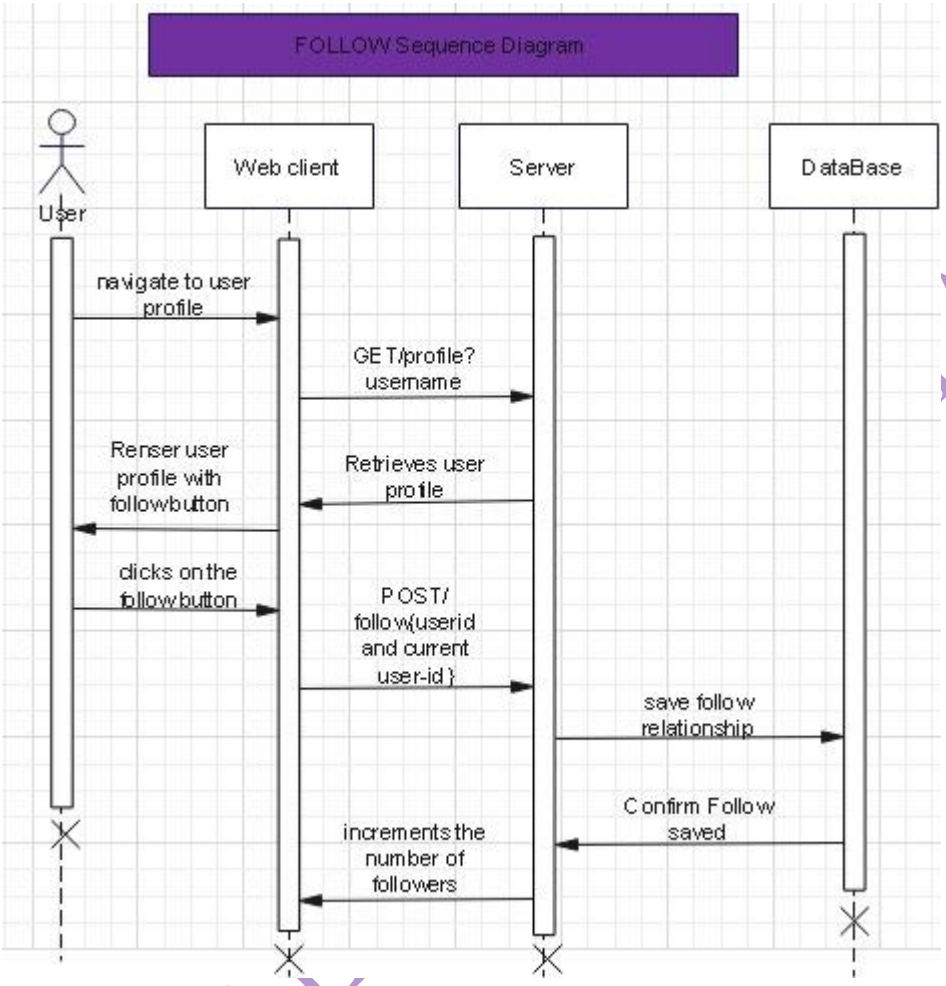
SOCIALIZE

3-Create Post



CREATE POST SEQUENCE DIAGRAM

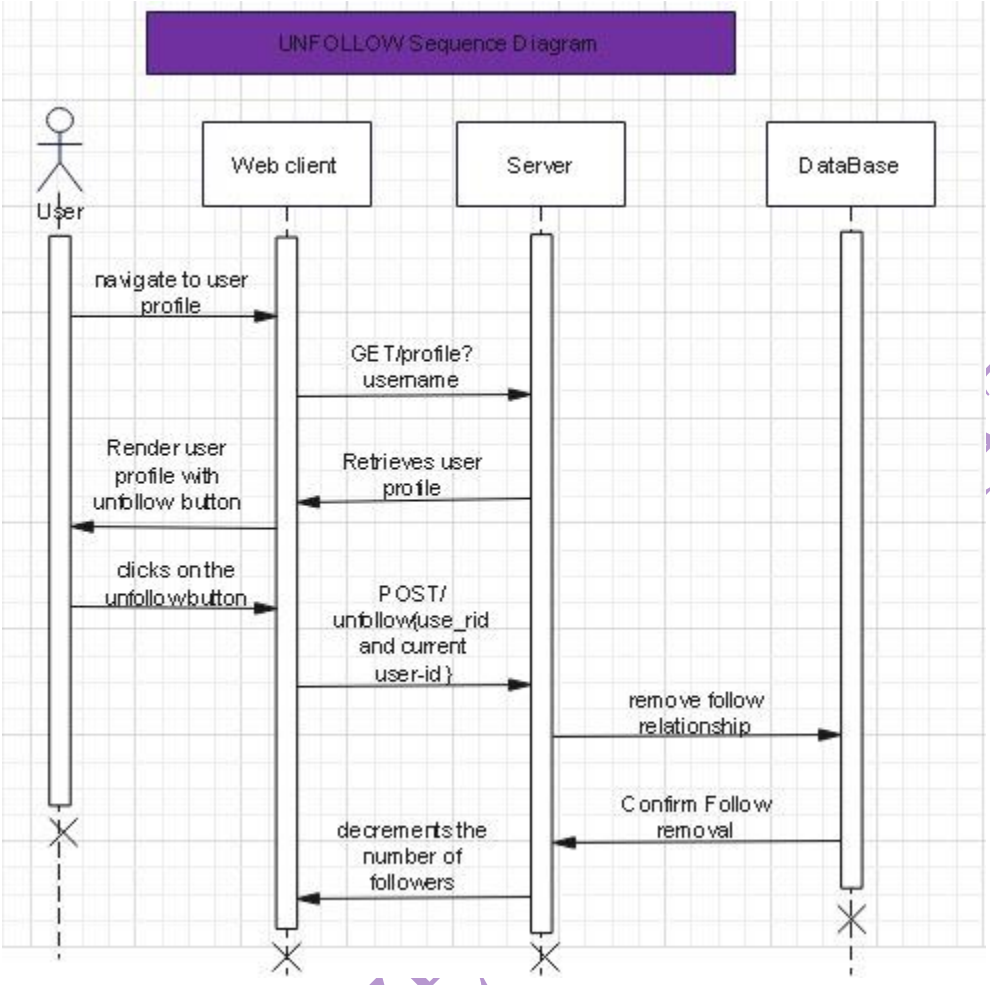
4-Follow a User



***Follow a User* sequence diagram**

SOCIALIZATION

5-Unfollow a user



Unfollow a User Sequence Diagram

3.4. Application of Scrum

3.4.1. Team Organization

The team was structured as follows:

Scrum Master: Reviewed and Assigned tasks to the entire team
Front-end Development Team: Led by a Front-end Lead

SOCIALIZE DOCUMENTATION

Back-end Development Team: Led by the Product Owner
Product Owner: Initiated the project, shared tasks based on user stories, reviewed changes, and merged them when approved

Testing: Conducted by the Scrum Master, Product Owner, then entire team (Each team member had to test what they were working on at the end of each assigned task)

- ✓ **3.4.2. workflow management**
  The team adopted an agile methodology, with the following workflow:

- ✓ Product Owner initiated the project and shared tasks with the entire team based on user stories.

- ✓ The Front-end and Back-end Development teams worked on their respective tasks.

- ✓ The Scrum Master viewed assigned tasks to the entire team and tracked the progress.

- ✓ The Product Owner assigned tasks and reviewed the changes made by the teams and merged them when approved.

- ✓ The Scrum Master, Product Owner, and the entire team conducted testing.

- ✓ The team utilized project management tools (Teams, Jira) to track tasks, deadlines, and progress.

3.4.3. Conflict Resolution:

The team encountered the following conflicts and resolved them:

The Product Owner had issues with a Back-end Developer due to poor communication.
The Scrum Master had problems with the former Front-end Lead due to poor communication which led to the delayed progress of the entire team.
To resolve these conflicts, the team implemented the following rules:

- ✓ Encouraged open and constructive dialogue to address any disagreements or tensions.
- ✓ Focused on finding win-win solutions that considered the perspectives of all team members.
- ✓ Send some backend members to work with the frontend to speed up progress.
- ✓ Scrum meetings were mandatory to ensure everyone is up to date with the project and better understand the difficulties the team was facing, trying to amend

SOCIALIZE DOCUMENTATION

.

Challenges Encountered and How They Were Overcome:

- Backend: Application unable to run caused by the database connectivity
  Used Xampp and/or MYSQL server
- FrontEnd: Uploading an image (post by a user)
- Making the Screens responsive

3.4. Scrum Artifacts:

3.4.1. product backlog

PRODUCT BACKLOG						
StoryID	User Stories	Acceptance Criteria	priority	initial estimate	adjustment factor	final estimate
1	As a content creator, I want to be able to see a list of alredy existing post as to follow the creators	As a content creator, I want to be able to see a list of already existing posts to follow the creators	8	21	1	21
2	As a professional looking to network, I want to be able to connect with other professionals in my field and see their work experience through a dedicated profile section	As a professional, I can connect with other professionals in my field and see their work experience through a dedicated profile section	7	21	0.666666667	14
3	As a user, I want to receive timely notifications about friend requests, likes, comments, and other relevant activities so that I can stay informed.	As a user, I receive timely notifications about friend requests, likes, comments, and other relevant activities.	5	21	5.714285714	120
4	As a user, I want to like, comment, and share blog posts from other users to show appreciation for their content	As a user, I can like, comment on, and share blog posts from other users.	4	21	0.666666667	14

MENTATION

	and engage in meaningful discussions.					
5	As a user, I want to report inappropriate content or block users to ensure a safe and positive online experience. (Safety and moderation features)	As a user, I can report inappropriate content or block users	9	21	1	21
6	As a user, I want the platform to be accessible on a variety of devices (desktop, mobile) so I can access it from anywhere. (Responsiveness for different screen sizes)	As a user, I can access the platform on a variety of devices (desktop, mobile).	10	21	1.142857143	24
7	As a user, I want to be able create and login to an account to join the community and connect with friends.	As a user, I can create and login to an account to join the community and connect with friends	1	21	1	21
8	As a user I want to be able to log out and delete my account in the situation where I no longer wish to be on the platform.	As a user, I can log out and delete my account.	11	21	0.666666667	14
9	As a user I want to be able to (CRUD) a post so as to share my thoughs with my network	As a user, I can create, read, update, and delete (CRUD) a post to share my thoughts with my network	3	21	3.428571429	72
10	As a user, I want to set up my profile so that I can share my details with the community	As a user, I can set up my profile with my details.	2	21	3.428571429	72
11	As a user, I want to search for other users and posts, so that I can find content and connections relevant to me	As a user, I can use the search function to find users and posts	6	21	1	21

3.4.2. Sprint Backlog

Release	Sprint	ID of User Stories	Period/hrs
Release 1: Rest API, frontend, backend deployment	SPRINT 1	7	21
	SPRINT 2	9,10	21
	SPRINT 3	4,3	21
	SPRINT 4	11,2	21
	SPRINT 5	1,5	21
	SPRINT 6	6,8	21
	SPRINT 7	6,7	21
	SPRINT 8	8,10	21

3.5 Test Case Document

3.5.1 Functionality: **User Registration**

Name: Verify user can register a new account
Preconditions:
- User is not currently logged in to the platform.

Test Steps:
1. User navigates to the registration page.
2. User enters a unique username.
3. User enters a valid email address.
4. User enters a secure password.
5. User enters a name.
6. User clicks the "Register" button.

Expected Result:
- The user's account is successfully created else the user is requested to try again.
- The user is logged in to the platform.
- The user receives a “Successfully registered! Redirecting to login…” message and is redirected to the platform's homepage after 3 seconds.

Actual Result:
- The user's account was created and they were logged in successfully.

Pass/Fail: Pass

SOCIALIZE DOCUMENTATION

3.5.2 Functionality: **User Login**

Name: Verify user can login to their account
Preconditions:
- User has a registered account on the platform.

Test Steps:
1. User navigates to the login page.
2. User enters their registered Username.
3. User enters their password.
4. User clicks the "Login" button.

Expected Result:
- The user is authenticated and logged in to the platform if credentials of login = credentials of Register, else the User is requested to try again.
- The user is redirected to the platform's feed.

Actual Result:
- The user was logged in successfully.

Pass/Fail: Pass

3.5.3 Functionality: **Create a Post**

Name: Verify user can create a new post
Preconditions:
- User is logged in to the platform.
- User has permission to create posts.

Test Steps:
1. User navigates to the "Create Post" section of the platform.
2. User enters a title for the post.
3. User enters the content for the post.
4. User optionally uploads an image or video to accompany the post.
5. User clicks the "Post" button.

Expected Result:
- The post is successfully created and displayed in the user's feed.

SOCIALIZE DOCUMENTATION

- The post is visible to the user's followers.

Actual Result:
- The post was created and displayed correctly.

Pass/Fail: Pass

3.5.4 Functionality: **User Interaction (Like, Comment on Post, Comment on a Comment)**

Name: Verify user can like a post
Preconditions:
- User is logged in to the platform.
- User has permission to interact with posts.

Test Steps:
1. User navigates to a post in their feed.
2. User clicks the "Like" button on the post.

Expected Result:
- The user's like is registered, and the post 's like count is incremented.

Actual Result:
- The like was registered correctly, and the post 's like count was updated.

Pass/Fail: Pass

3.5.5 Functionality: **Comment on a post**

Name: Verify user can comment on a post
Preconditions:
- User is logged in to the platform.
- User should be able to view the post
Test Steps:
1. User navigates to a post in their feed.
2. User enters a comment in the comment section.
3. User clicks the "Comment" on the post body.

Expected Result:

SOCIALIZE DOCUMENTATION

- The user's comment is displayed under the post.
- The post's comment count is incremented.

Actual Result:
- The comment was posted successfully, and the post's comment count was updated.

Pass/Fail: Pass

3.5.6 Functionality: **Comment on a comment**

Name: Verify user can comment on a comment
Preconditions:
- User is logged in to the platform.
- User can view post and comment.

Test Steps:
1. User navigates to a post in their feed.
2. User clicks on an existing comment to expand it.
3. User enters a reply comment in the comment section.
4. User clicks the "Post Comment" button.

Expected Result:
- The user's reply comment is displayed as a nested comment under the original comment.
- The comment's reply count is incremented.

Actual Result:
- The reply comment was posted successfully, and the comment's reply count was updated.

Pass/Fail: Pass

3.5.7 Functionality: **Notification**

Name: Verify user receives notifications for new activities
Preconditions:
- User is logged in to the platform.
- User has permission to receive notifications.

Test Steps:
1. User performs various actions, such as creating a post, receiving a like or comment, or being mentioned in a comment.
2. User checks the notifications section of the platform.

SOCIALIZE DOCUMENTATION

Expected Result:
- The user receives notifications for the various activities they performed or were involved in through Gmail.

Actual Result:
- The user received the expected notifications, and they were displayed correctly in the notification feed.

Pass/Fail: Pass

3.5.8 Functionality: **Reporting and Moderation Mechanism**

Name: Verify user can report inappropriate content
Preconditions:
- User is logged in to the platform.
- User has permission to report content.

Test Steps:
1. User identifies a post or comment that they consider inappropriate.
2. User clicks the "Report" button on the content.
3. User selects the appropriate reason for reporting the content.
4. User submits the report.

Expected Result:
- The reported content is flagged for review by the moderation team.
- The user receives a confirmation that the report has been submitted.

Actual Result:
- The report was submitted successfully, and the user received the expected confirmation.

Pass/Fail: Pass

3.5.9 Functionality: **Multi-Device Accessibility**

Name: Verify user can access the platform on different devices
Preconditions:

SOCIALIZE DOCUMENTATION

- User has a registered account on the platform.

Test Steps:
1. User logs in to the platform on their desktop computer.
2. User performs various actions, such as creating a post, liking a post, and commenting on a post.
3. User logs out of the platform on the desktop computer.
4. User logs in to the platform on their mobile device.

Expected Result:
- The user's account information, including their posts, interactions, and settings, are synchronized across all devices.
- The user can seamlessly continue their activities on the platform from any device.

Actual Result:
- The user's account information was correctly synchronized, and they were able to continue their activities on the mobile device.

Pass/Fail: Pass

This test case document covers user registration, user login, creating a post, user interaction (like, comment on post, comment on comment), notification, reporting and moderation mechanism, and multi-device accessibility. Each test case includes the, name, preconditions, test steps, expected results, and the actual results. The pass/fail status is also provided for each test case.

The Actual results will be represented below for each endpoint.

3.6 Proposed algorithms

3.6.1 User Registration:

register user (username, email, password, name)
Check if the provided username is unique.
Validate the email format.
Check if the password meets the security requirements (e.g., minimum length, complexity).
Create a new user account with the provided information.
Log the user in to the platform.
Display a success message and redirect the user to the platform's homepage.

SOCIALIZE DOCUMENTATION

3.6.2 User Login:
login user (username, password)
Retrieve the user's account information based on the provided username.
Verify the provided password against the stored password.
If the credentials are valid, log the user in to the platform.
Redirect the user to the platform's feed.
If the credentials are invalid, display an error message and prompt the user to try again.

3.6.3 Create a Post:
create post (title, content, media file=None)
Validate the provided title and content.
If a media file is provided, upload and associate it with the post.
Create a new post with the provided information.
Add the post to the user's feed and make it visible to their followers.
Display a success message to the user.

3.6.4 User Interaction:
like post (post_id, user_id)
Retrieve the post based on the provided post_id.
Associate the user's like with the post.
Increment the post 's like count.
Update the post's data in the database.

3.6.5 comment_on_post (post_id, user_id, comment_text)
Retrieve the post based on the provided post_id.
Create a new comment with the provided comment_text and associate it with the post.
Increment the post's comment count.
Update the post's data in the database.

3.6.6 comment_on_comment (comment_id, user_id, reply_text)
Retrieve the comment based on the provided comment_id.
Create a new reply comment with the provided reply_text and associate it with the original comment.
Increment the comment's reply count.
Update the comment's data in the database.

3.6.7 Notifications: Send_notification (user_id, notification_type, notification_data)
Retrieve the user's notification preferences based on the user_id.

SOCIALIZE DOCUMENTATION

Determine the appropriate notification channel (e.g., email, in-app notification) based on the user's preferences.
Construct the notification message with the provided notification_type and notification_data.
Send the notification to the user using the selected channel.

3.6.8 Reporting and Moderation: report_content (content_id, user_id, reason)
Retrieve the reported content based on the provided content_id.
Create a new report with the provided reason and associate it with the content.
Flag the reported content for review by the moderation team.
Notify the moderation team about the new report.
Provide a confirmation to the user who submitted the report.

3.6.9 Multi-Device Accessibility: sync_user_data (user_id, device_id)
Retrieve the user's account information and activity data based on the user_id.
Synchronize the user's data across all devices associated with the user_id.
Update the user's data on the current device to reflect the latest changes.
Ensure a seamless transition for the user when switching between devices.

3.7 Materials and technologies used

3.7.1Design

- UML Tool:
- Star UML
- plant UML
- Lucid chart
- Architecture: Lucid chart

3.7.2 Implementation

For Frontend,

- Programming Language: Reactjs
- Web Framework: React vit
- Session Management: JSON Web Tokens
- Version Control Platform: GitHub

For Backend,

- Programming Language: Java Spring boot
- Database: MySQL
- Authentication Library: bcrypt.js

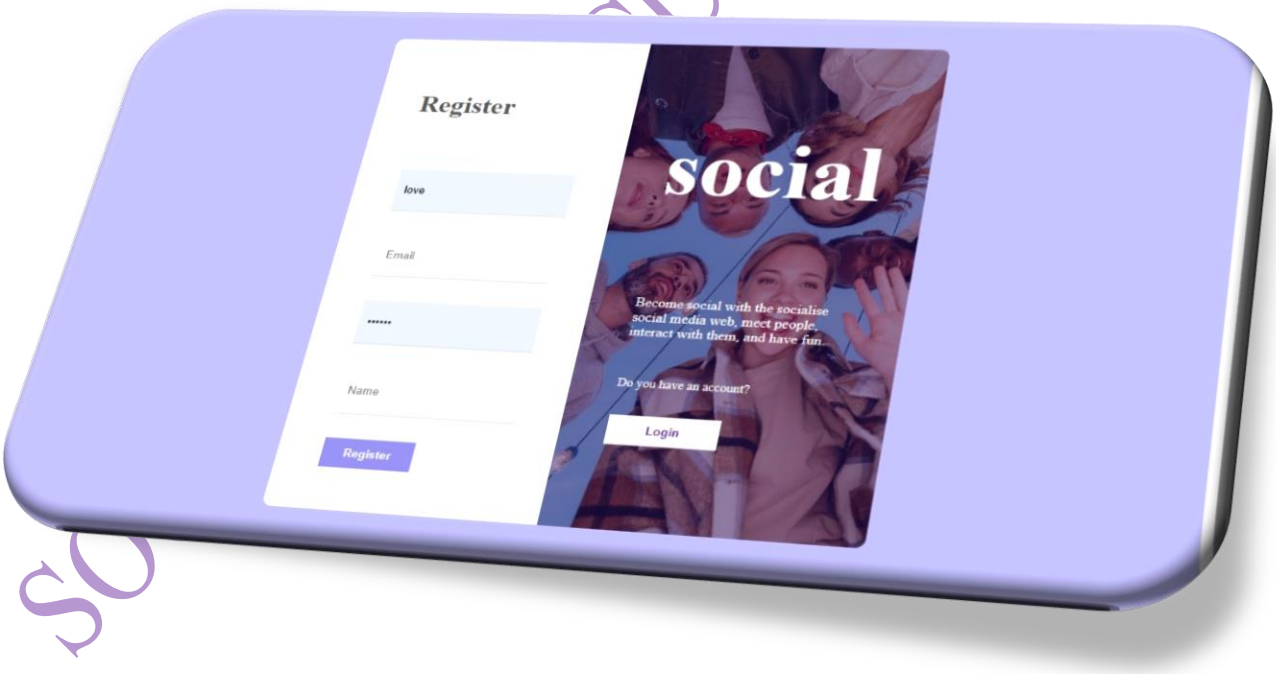
SOCIALIZE DOCUMENTATION

- API documentation: Rest API
- Testing Framework: Junit test
- Version Control Platform: GitHub

4.RESULTS AND DISCUSSIONS

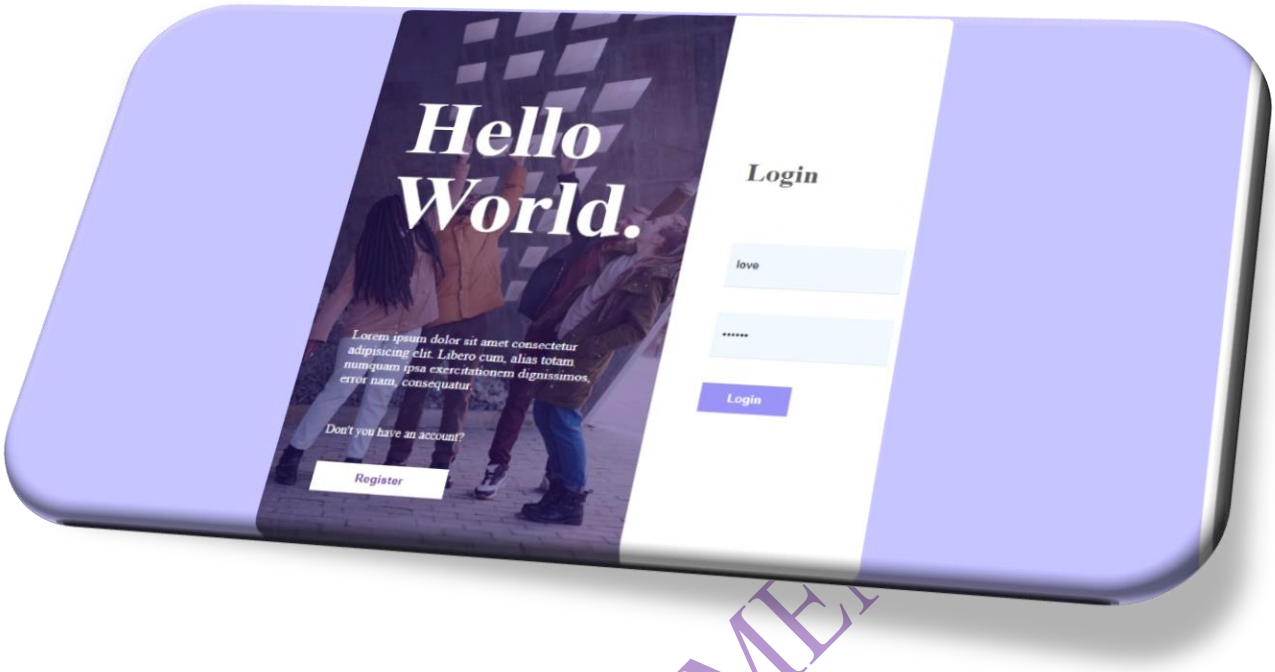
4.1. Screenshots of various application scenarios:

4.1.1 Register page

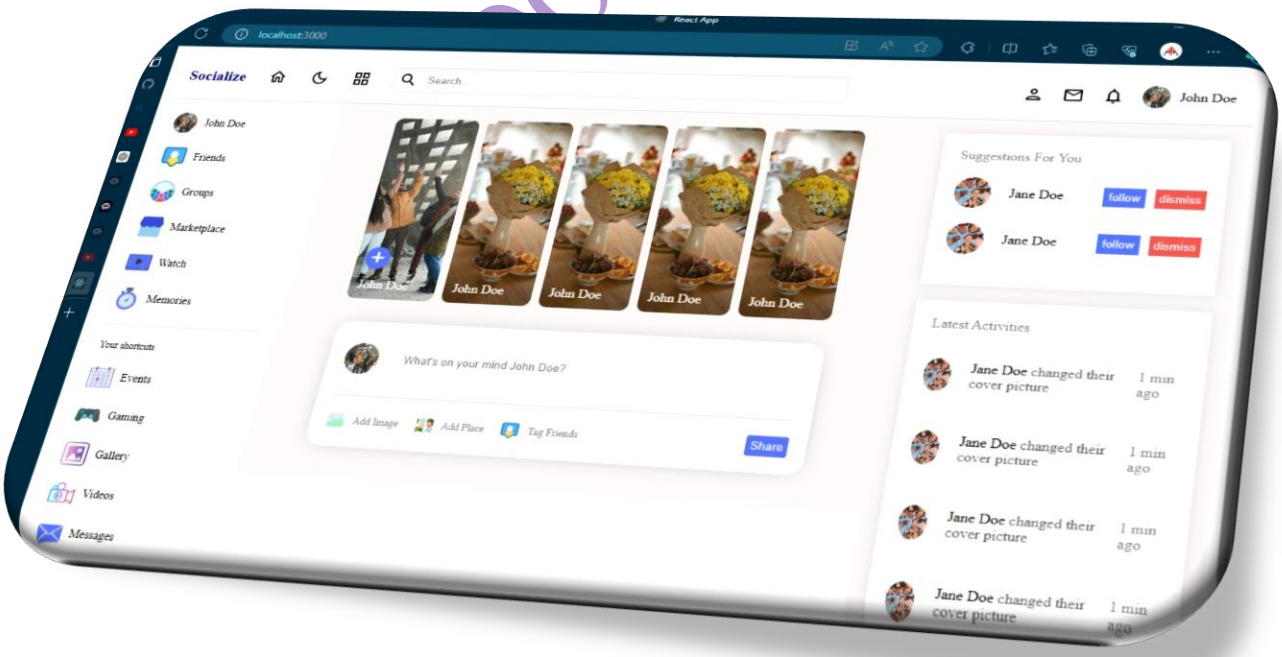


4.1.2 Login Page

SOCIALIZE DOCUMENTATION

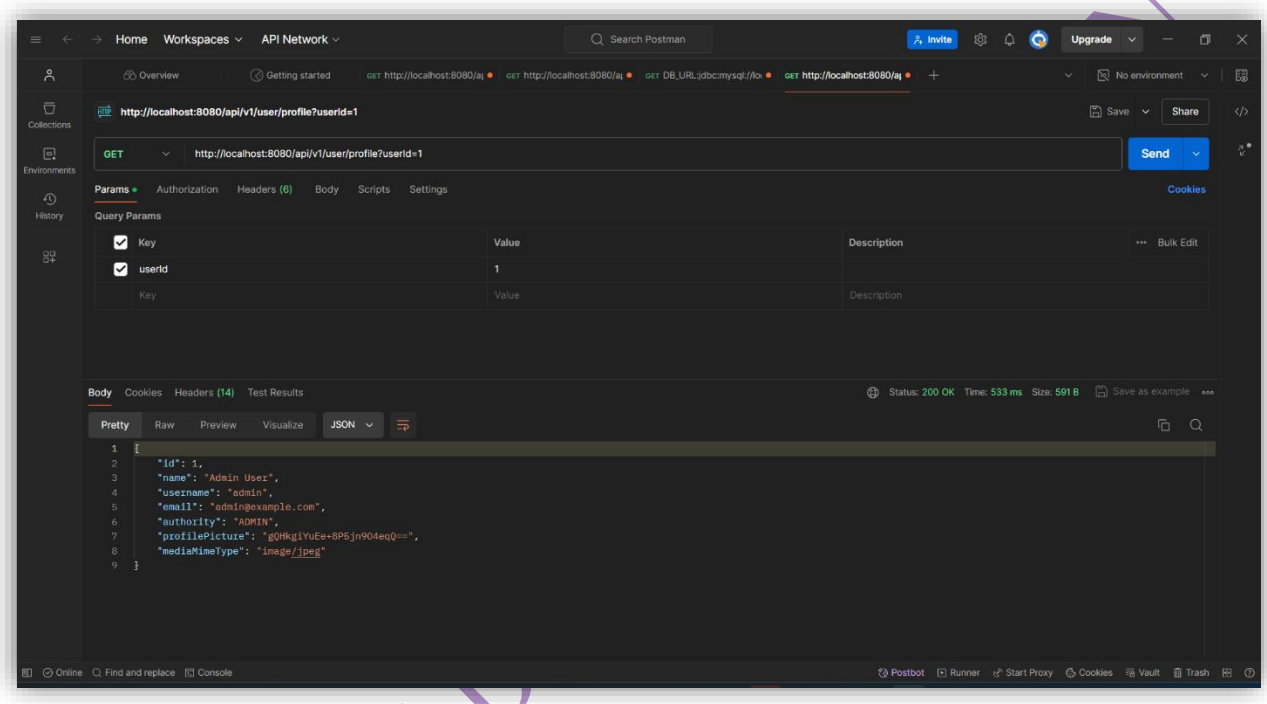


4.1.3 feed screen



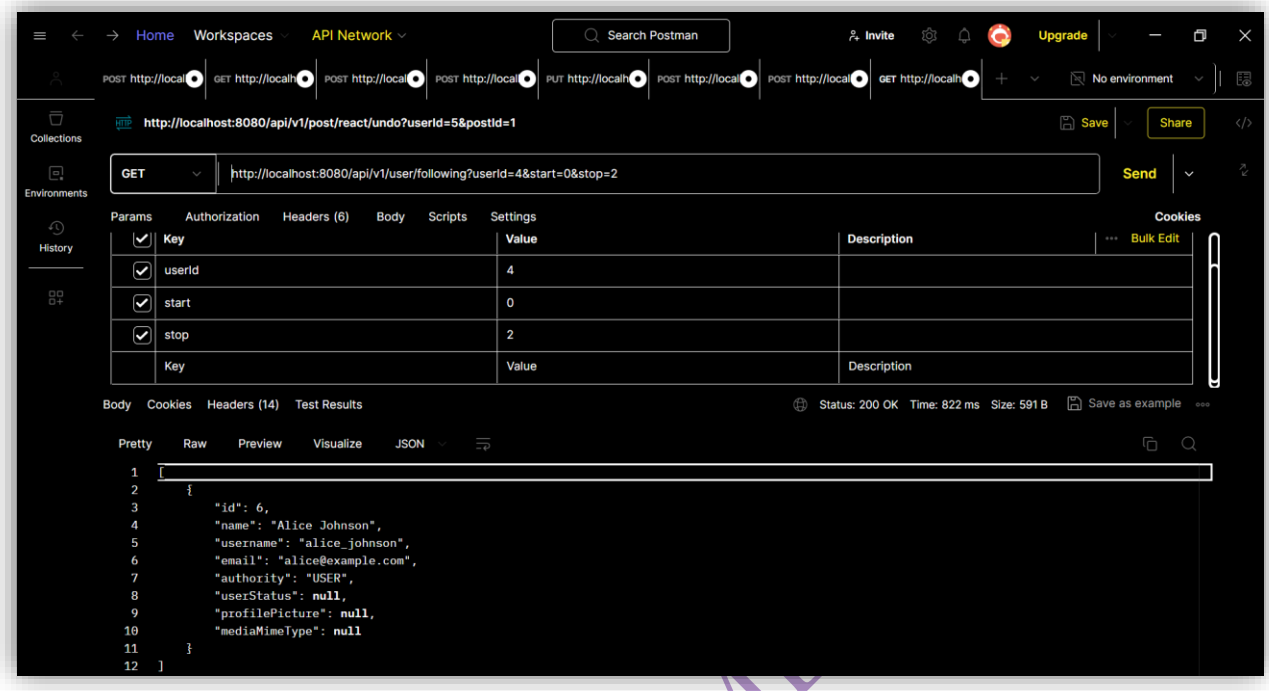
4.2. Screenshots of various API Request/Response:

4.2.1.api/v1/user/profile

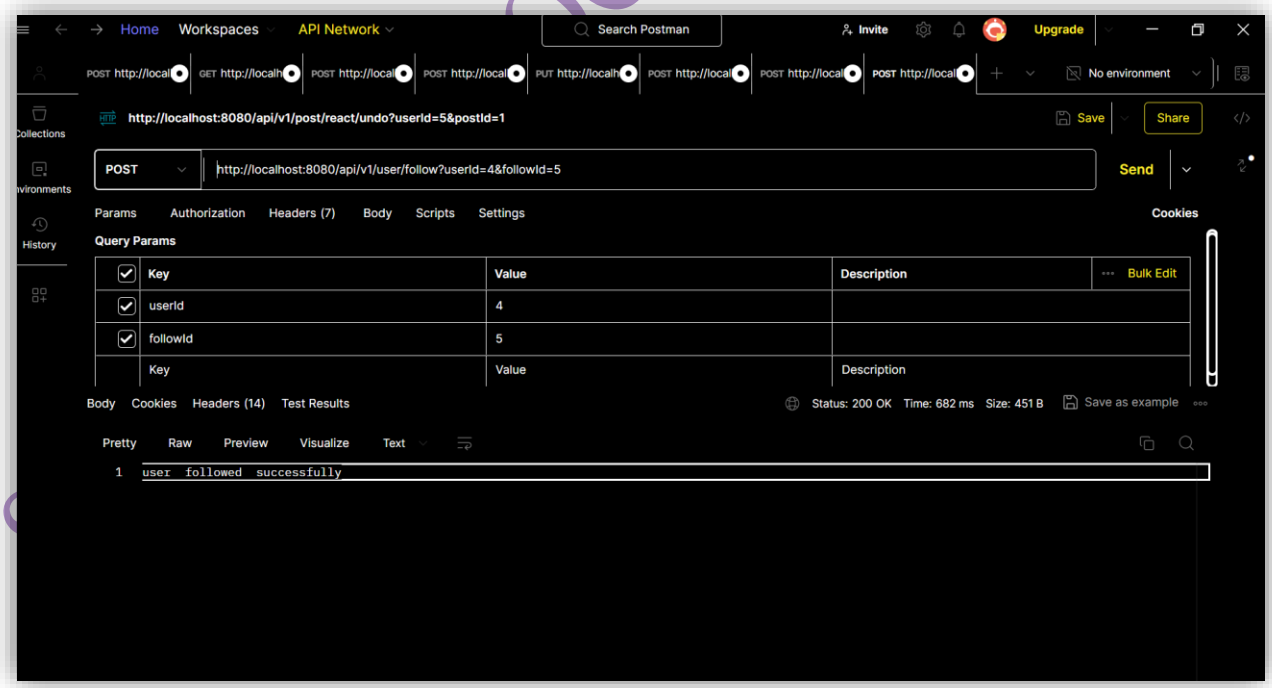


4.2.2. /api/v1/user/following

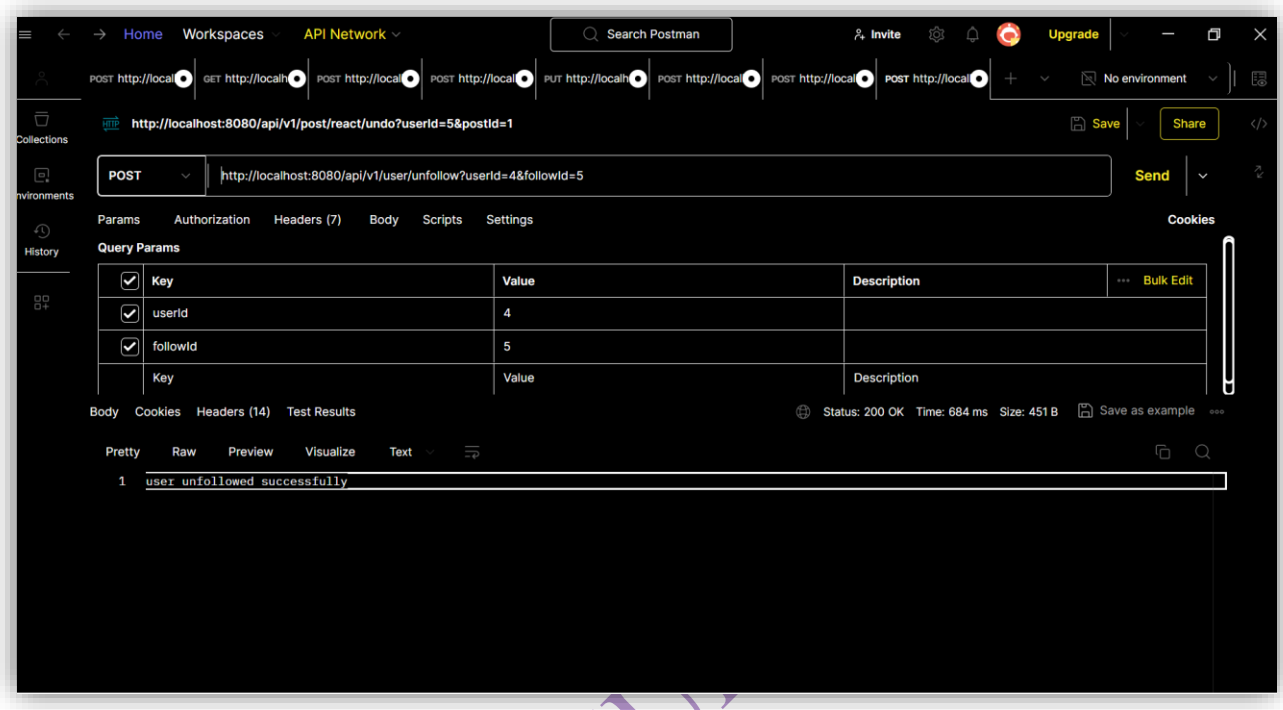
SOCIALIZE DEMO



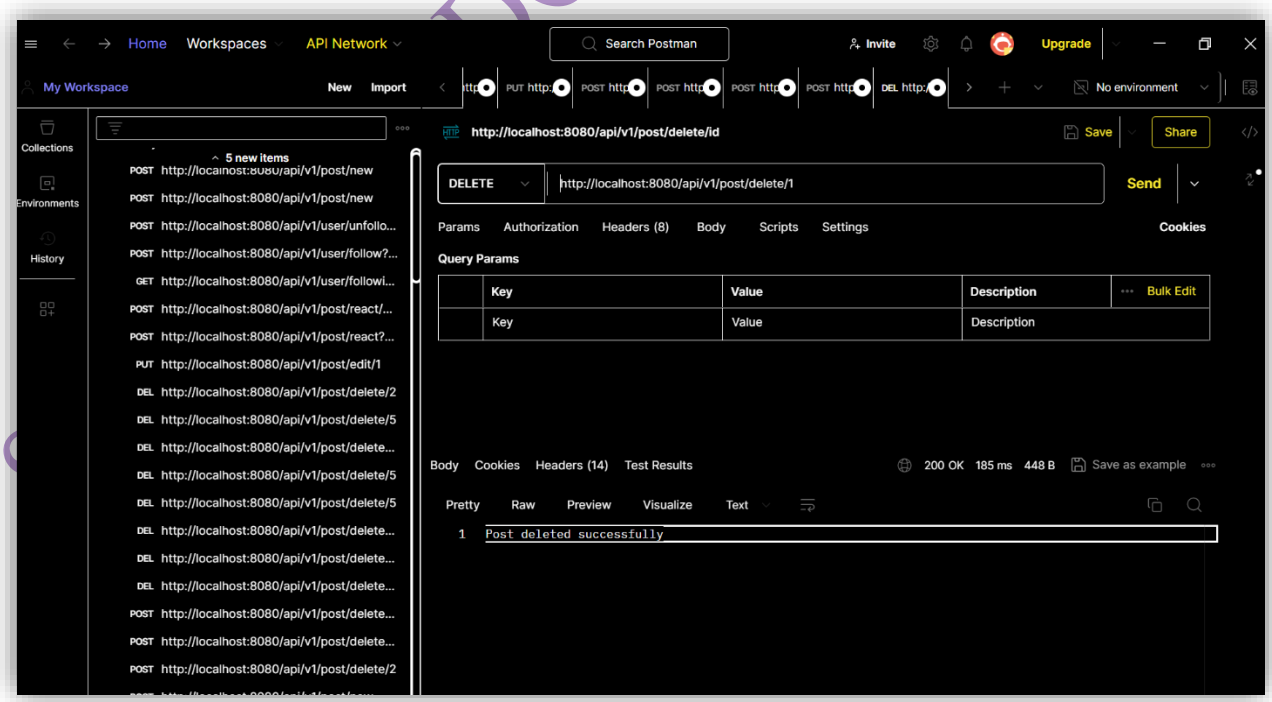
4.2.3. /api/v1/user/follow



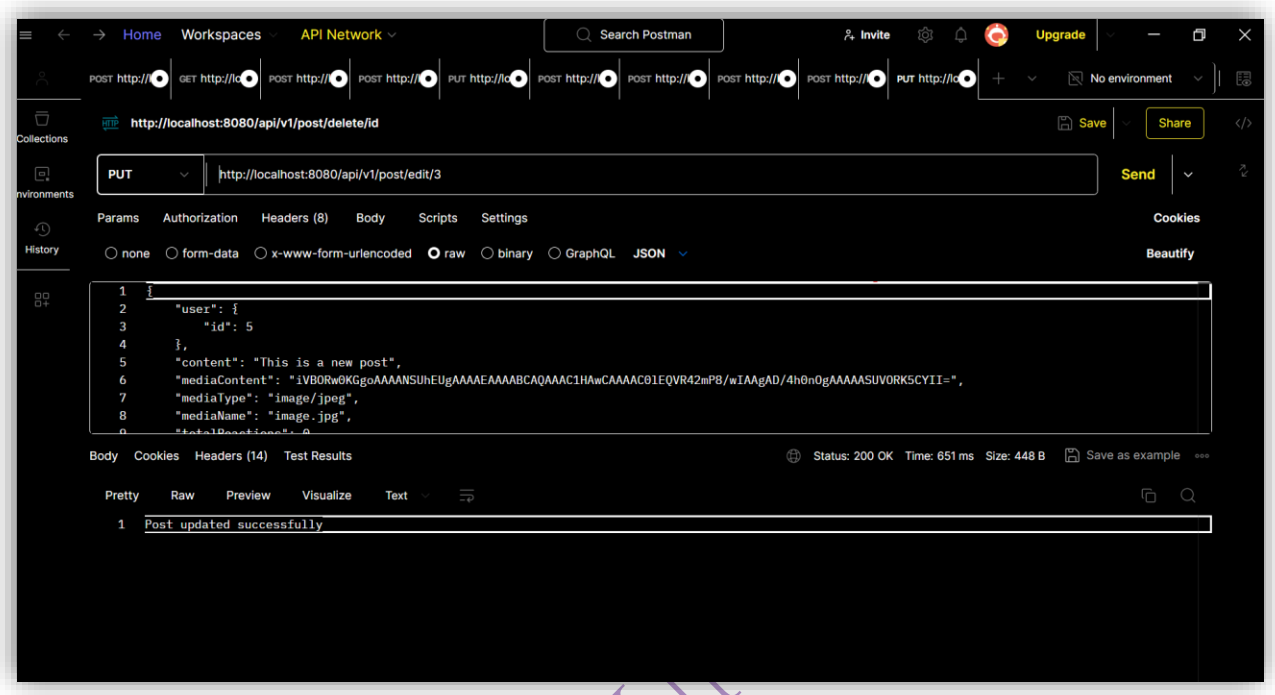
4.2.4./api/v1/user/unfollow



4.2.5./api/v1/post/deletePost

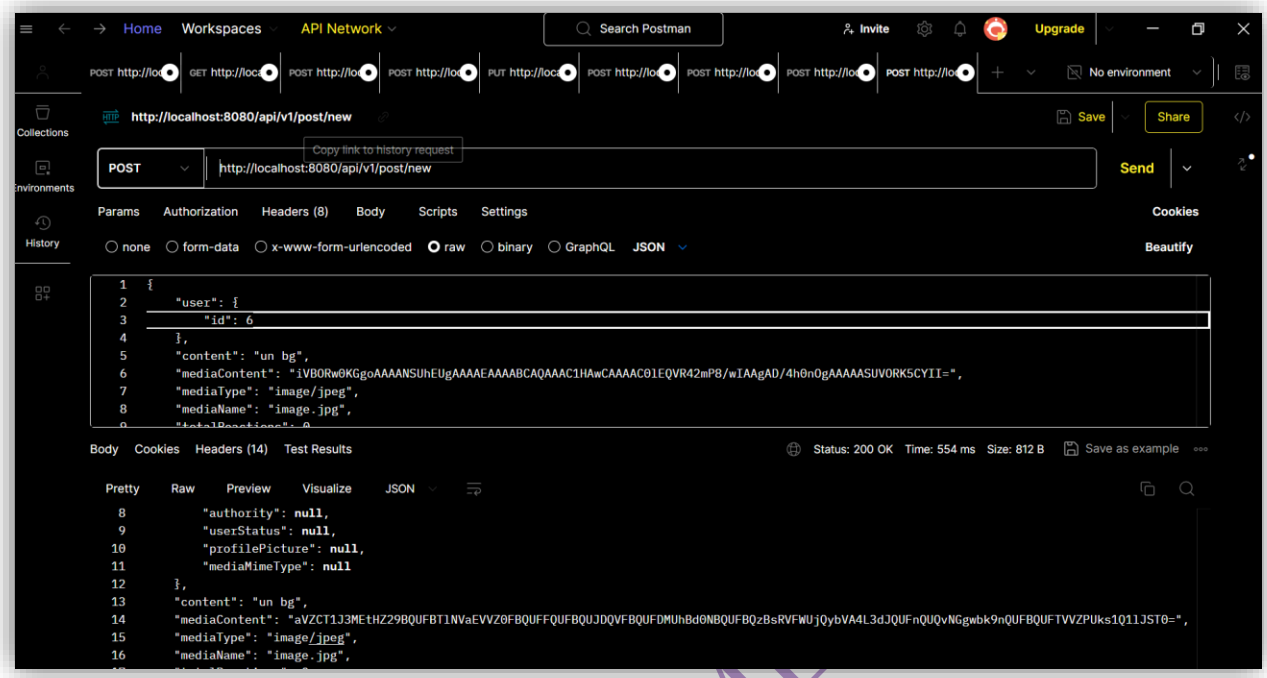


4.2.6./api/v1/post/editPost

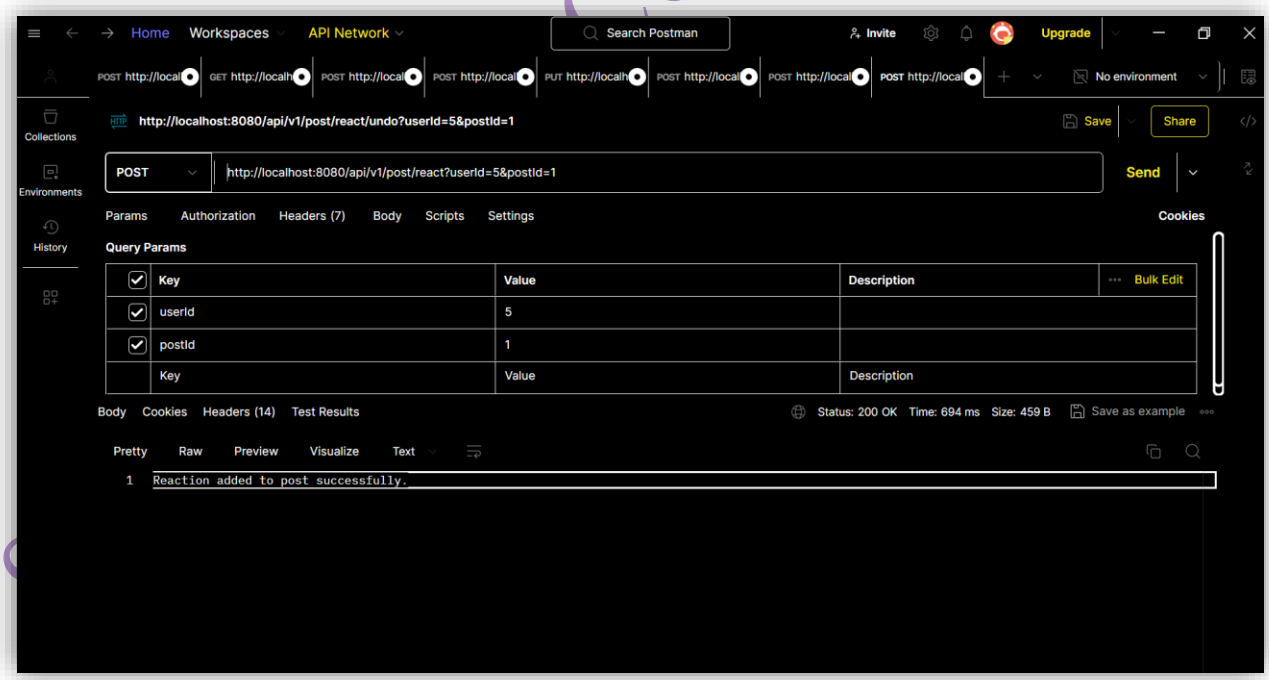


4.2.7./api/v1/post/new

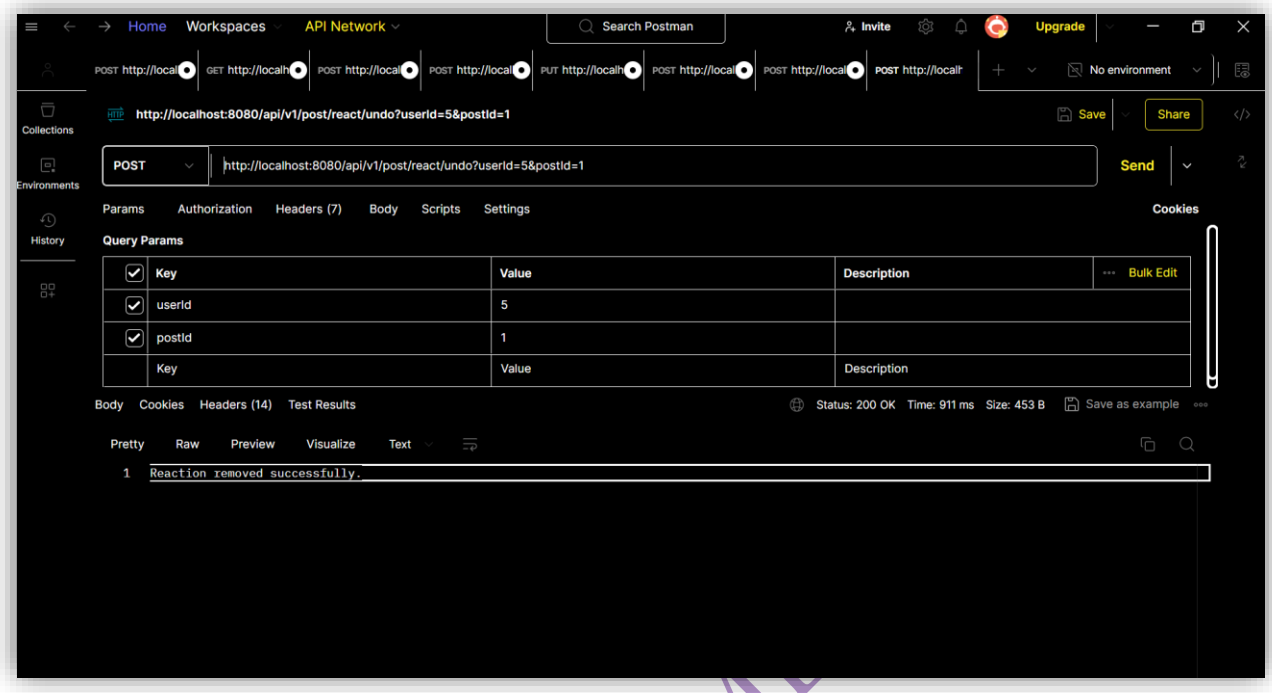
SOCIALIZE DOCU



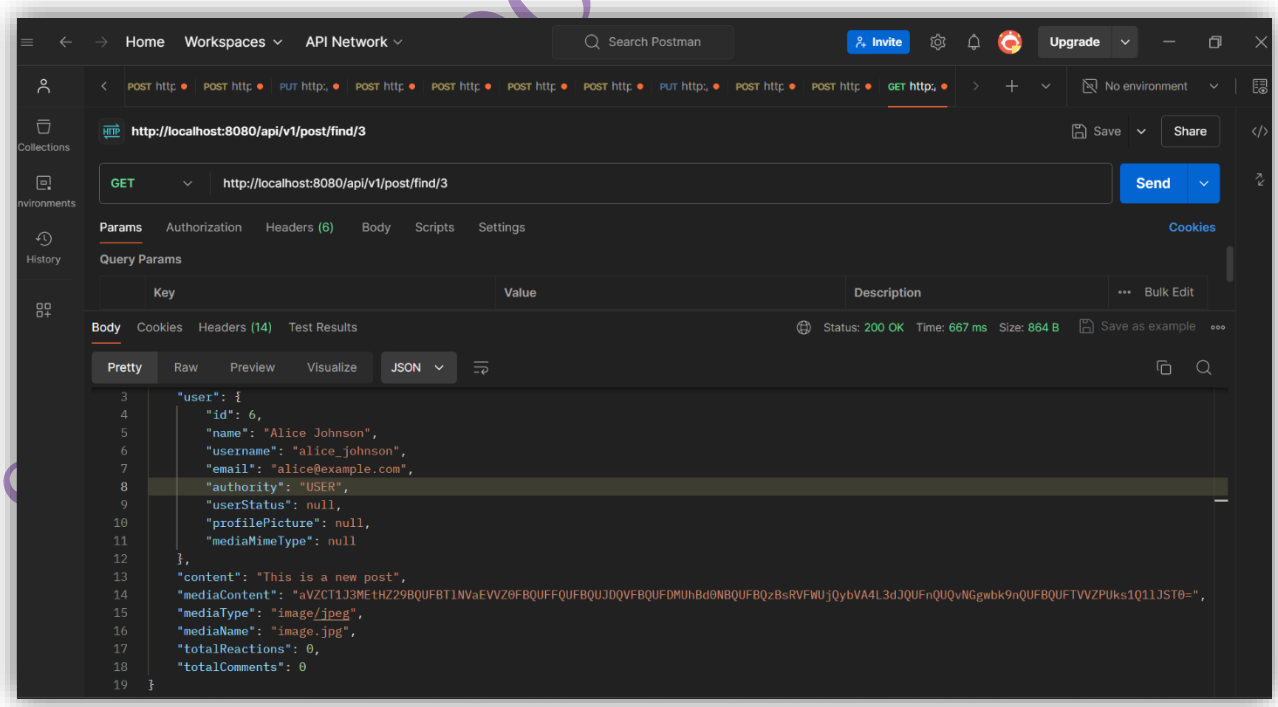
4.2.8. /api/v1/post/react



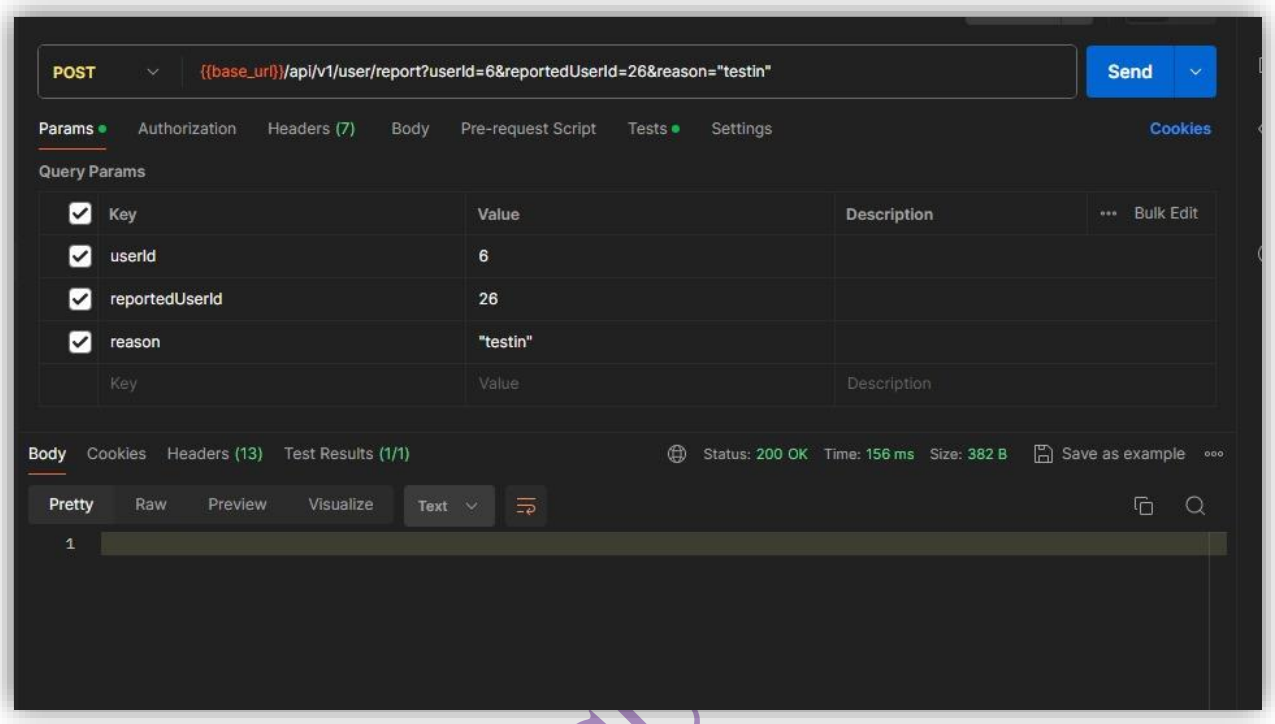
4.2.9/api/v1/post/react/undo



4.2.10: /api/v1/post/find

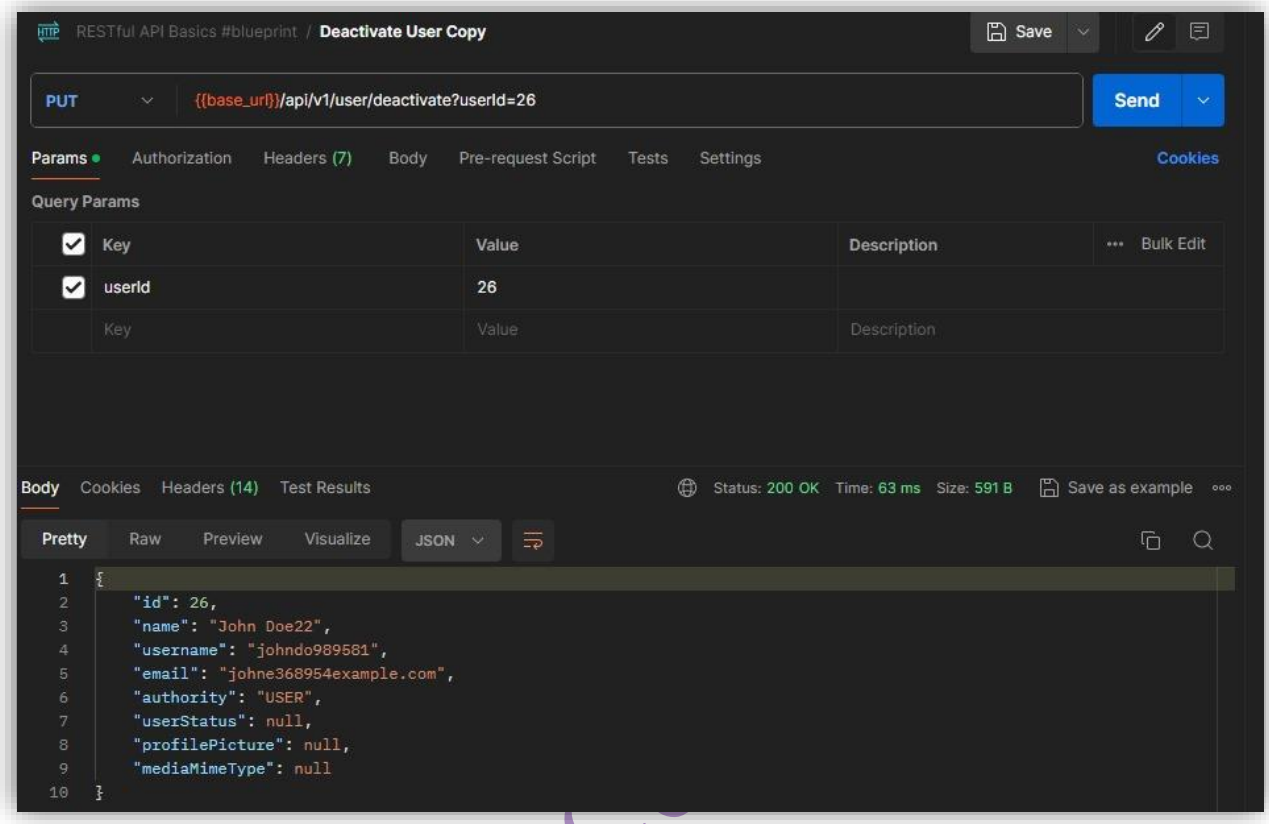


4.2.11: api/v1/report/user

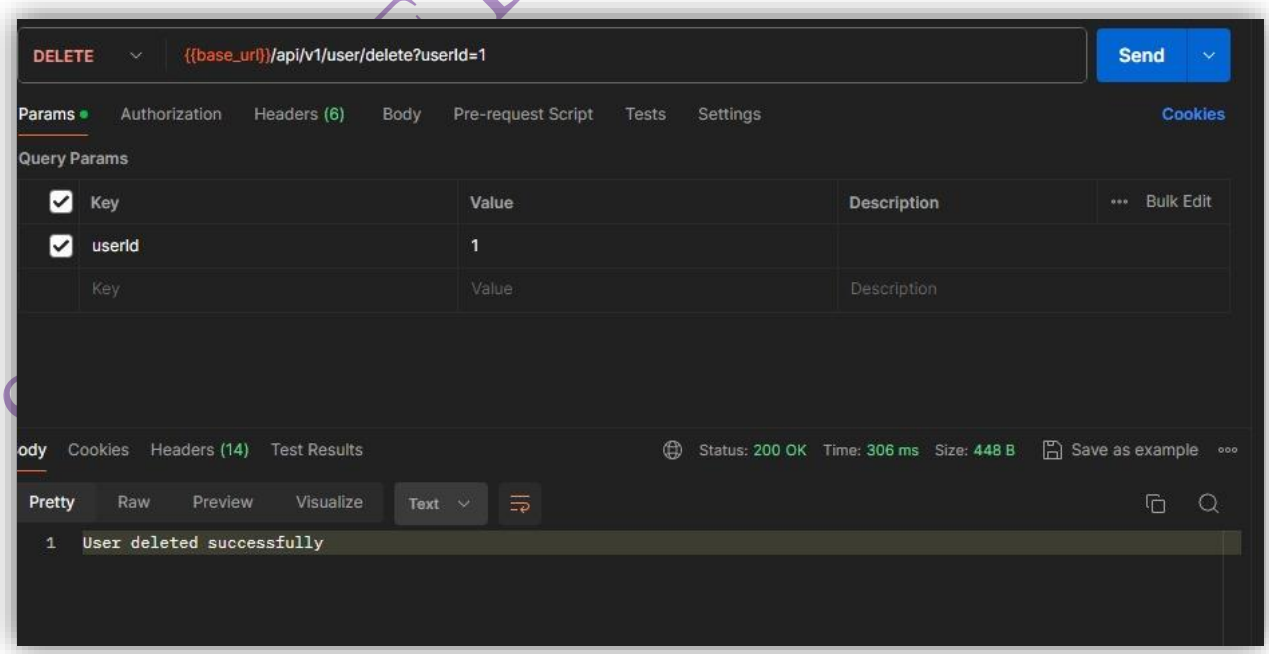


4.2.12: /api/v1/user/deactivate

SOCIALIZE DOCU



4.2.13: api/v1/user/delete



4.2.14:/api/v1/comment/parent

localhost:8080/api/v1/comment/parent?postId=1&start=1&stop=4

Nouveau dossier | IET Digital Library: e... | Join ResearchGate f... | ChatGPT | Home - Canva | Livres

Impression élégante


```
[
  {
    "id": 4,
    "content": "Comment 4 content",
    "parentCommentId": null,
    "totalReaction": 0,
    "user": {
      "id": 1,
      "name": "User1",
      "username": "user1",
      "email": "user0@example.com",
      "authority": "USER",
      "profilePicture": "cGF0aC90by9wcm9maWxl",
      "mediaMimeType": null
    },
    "post": {
      "id": 1,
      "user": {
        "id": 1,
        "name": "User1",
        "username": "user1",
        "email": "user0@example.com",
        "authority": "USER",
        "profilePicture": "cGF0aC90by9wcm9maWxl",
        "mediaMimeType": null
      },
      "content": "first post",
      "mediaContent": "bWVkaWFGaWxlTmFtZQ==",
      "mediaType": "mediaFileName",
      "mediaName": "mediaFileName",
      "totalReactions": 0,
      "totalComments": 0
    }
  },
```


4.2.15:/api/v1/ comment/child

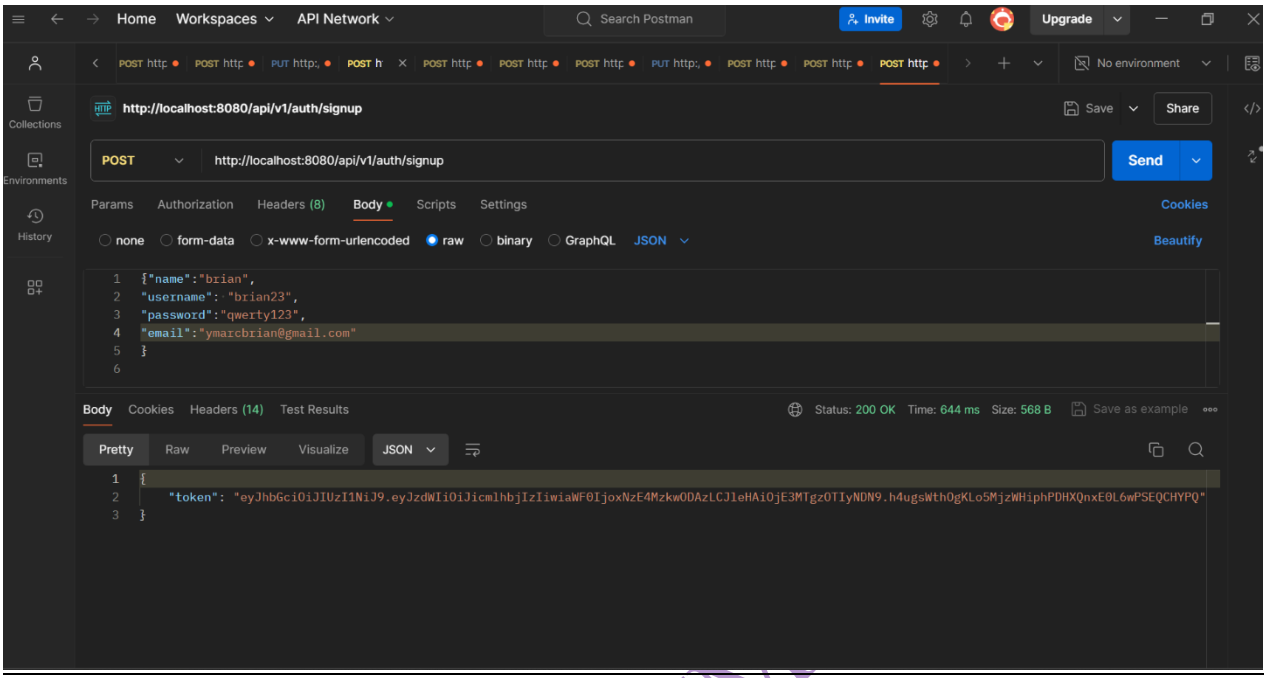
localhost:8080/api/v1/comment/child?parentCommentId=1&postId=1&start=1&stop=4

Nouveau dossier | IET Digital Library: e... | Join ResearchGate f... | ChatGPT | Home - Canva | Livres | Nouveau dossier | PyQt5

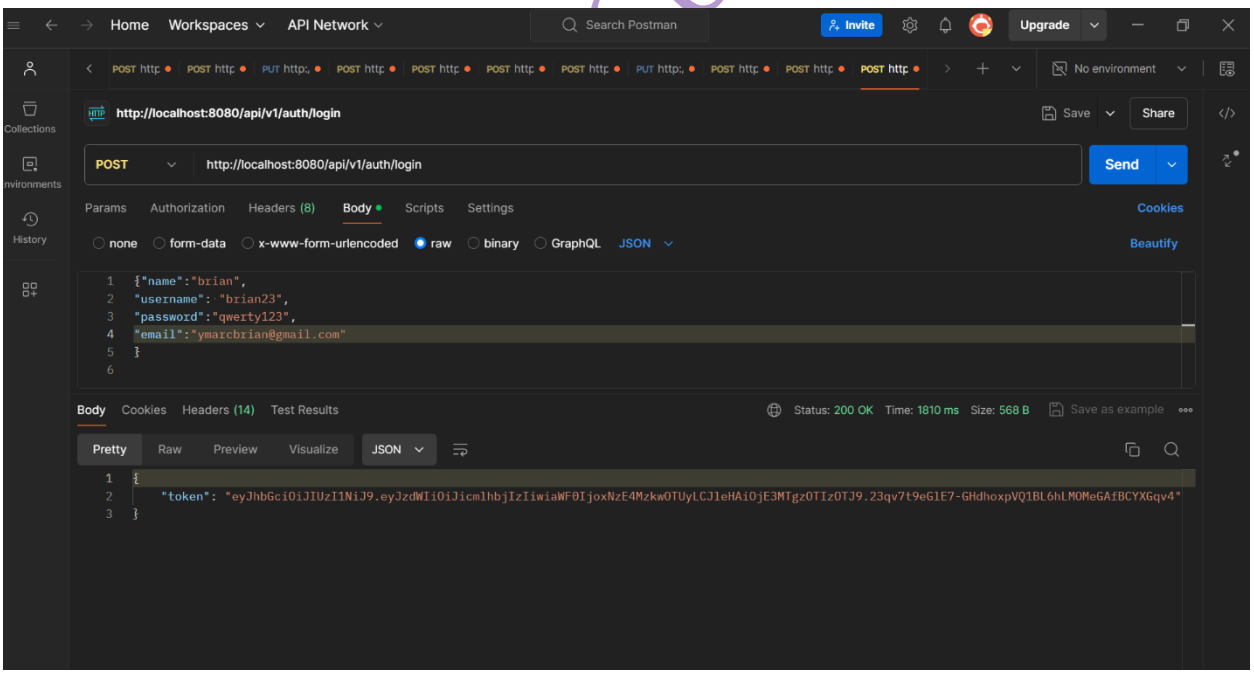
Impression élégante


```
[
  {
    "id": 14,
    "content": "Child comment 1 for comment 1",
    "parentCommentId": 1,
    "totalReaction": 0,
    "user": {
      "id": 1,
      "name": "User1",
      "username": "user1",
      "email": "user0@example.com",
      "authority": "USER",
      "profilePicture": "cGF0aC90by9wcm9maWxl",
      "mediaMimeType": null
    },
    "post": {
      "id": 1,
      "user": {
        "id": 1,
        "name": "User1",
        "username": "user1",
        "email": "user0@example.com",
        "authority": "USER",
        "profilePicture": "cGF0aC90by9wcm9maWxl",
        "mediaMimeType": null
      },
      "content": "first post",
      "mediaContent": "bWVkaWFGaWxlTmFtZQ==",
      "mediaType": "mediaFileName",
      "mediaName": "mediaFileName",
      "totalReactions": 0,
      "totalComments": 0
    }
  },
  {
    "id": 15,
    "content": "Child comment 2 for comment 1",
    "parentCommentId": 1,
    "totalReaction": 0,
    "user": {
      "id": 1,
```

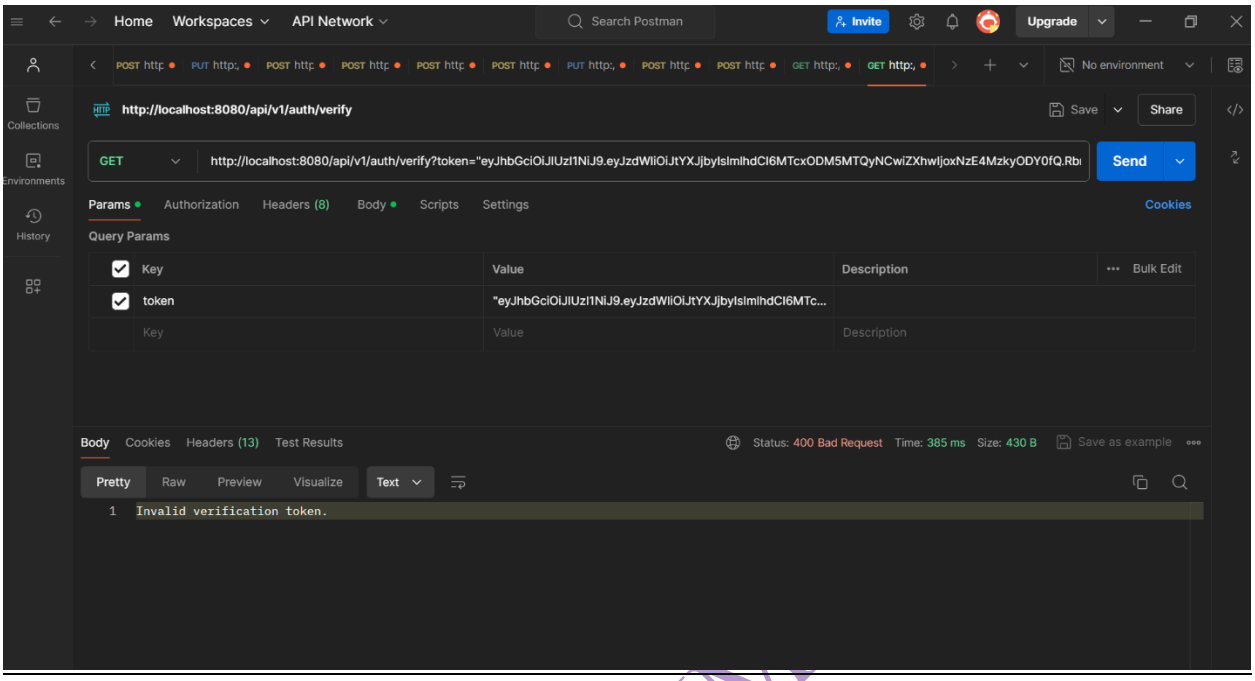
4.2.16:/api/v1/auth/signup



4.2.16:/api/v1/auth/login



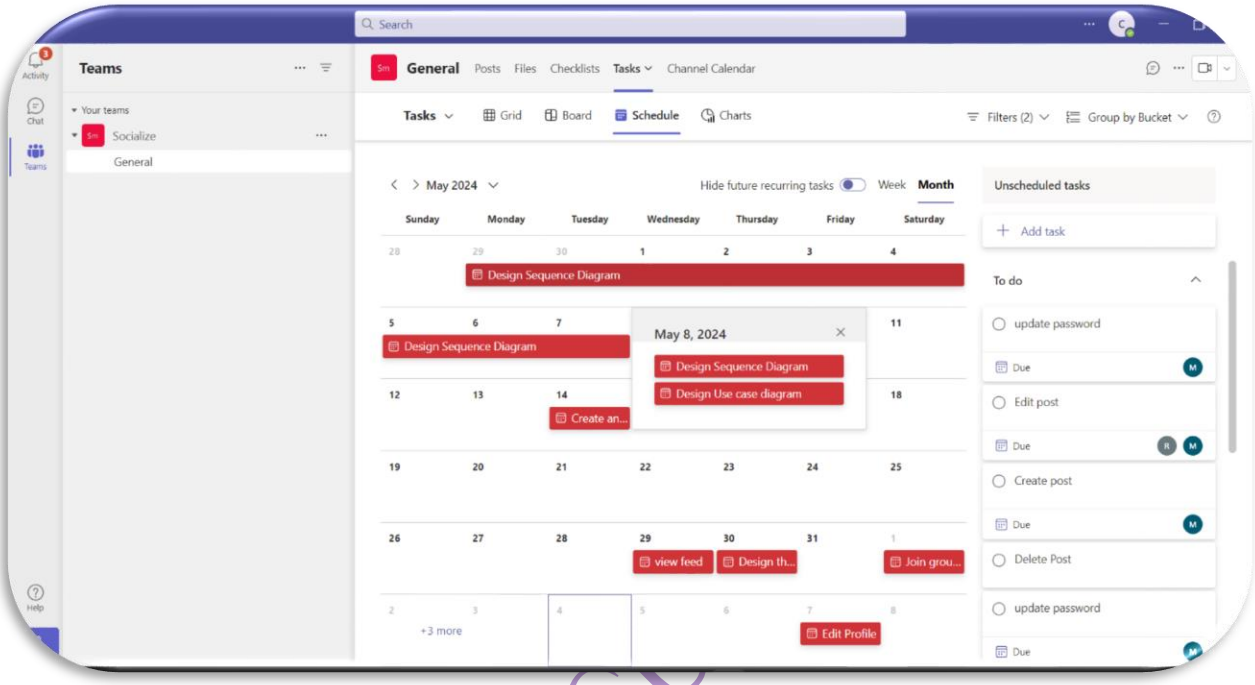
4.2.16:/api/v1/auth/verify-email

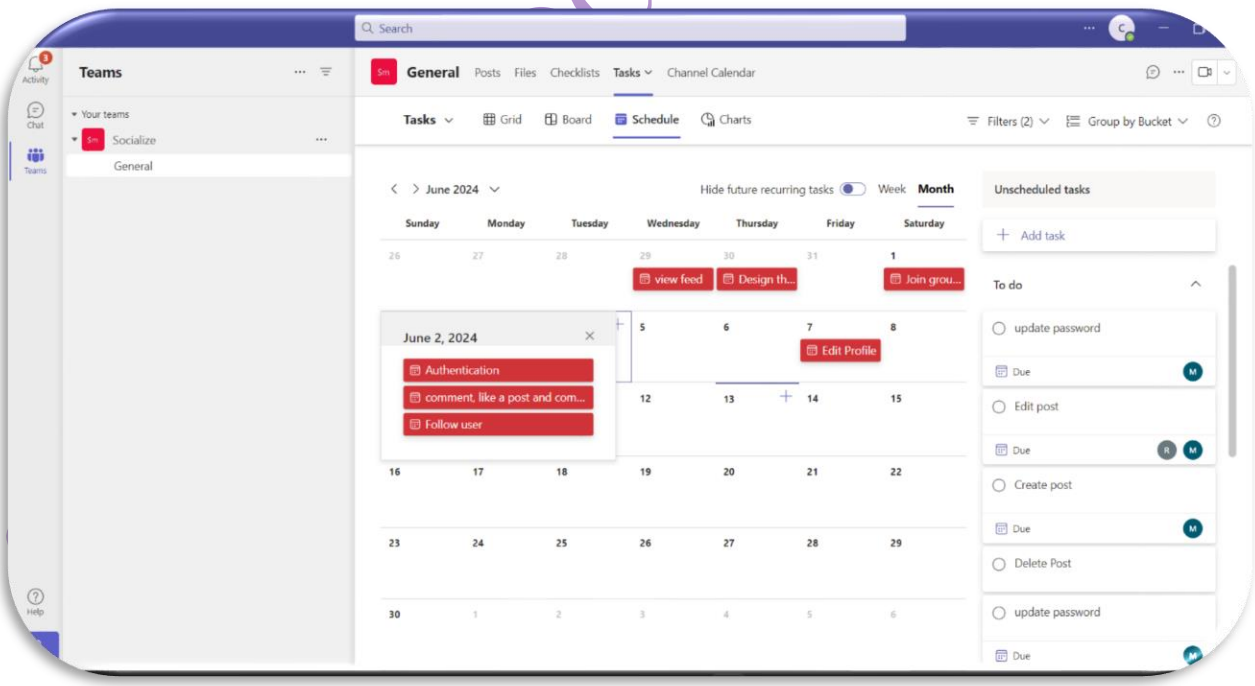


SOCIALIZE DOCUM

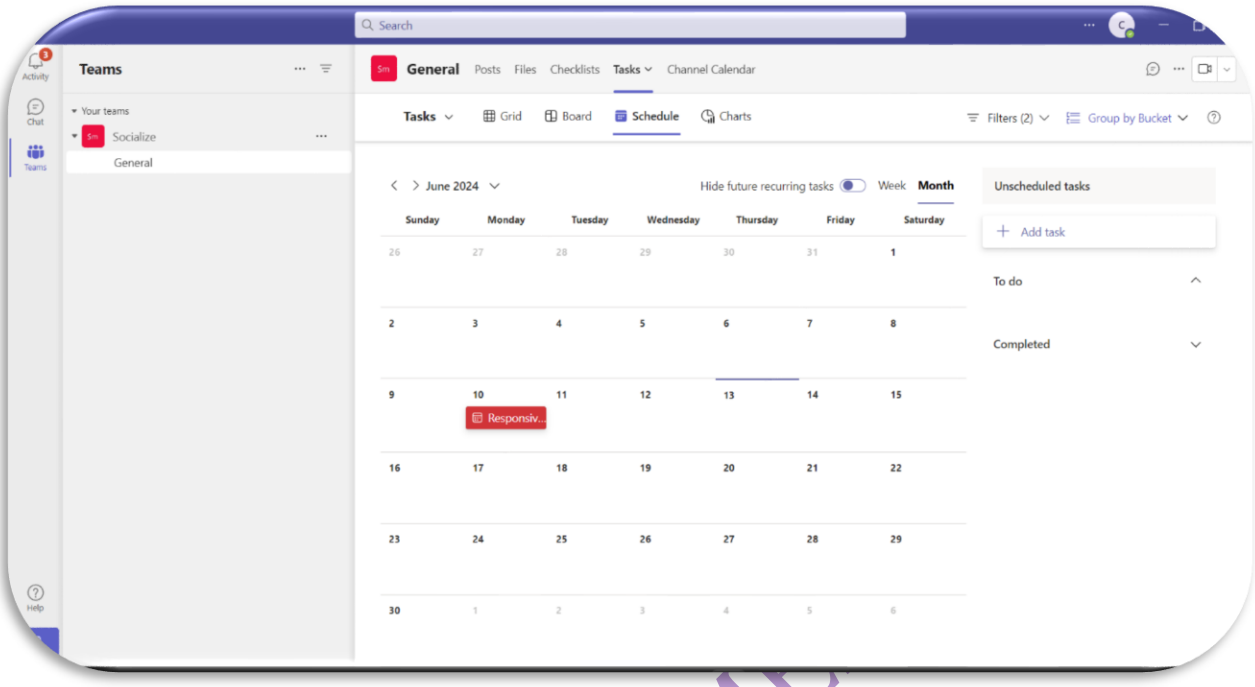
4.3. Tasks (posts and ToDos) on Microsoft Team:

Todos

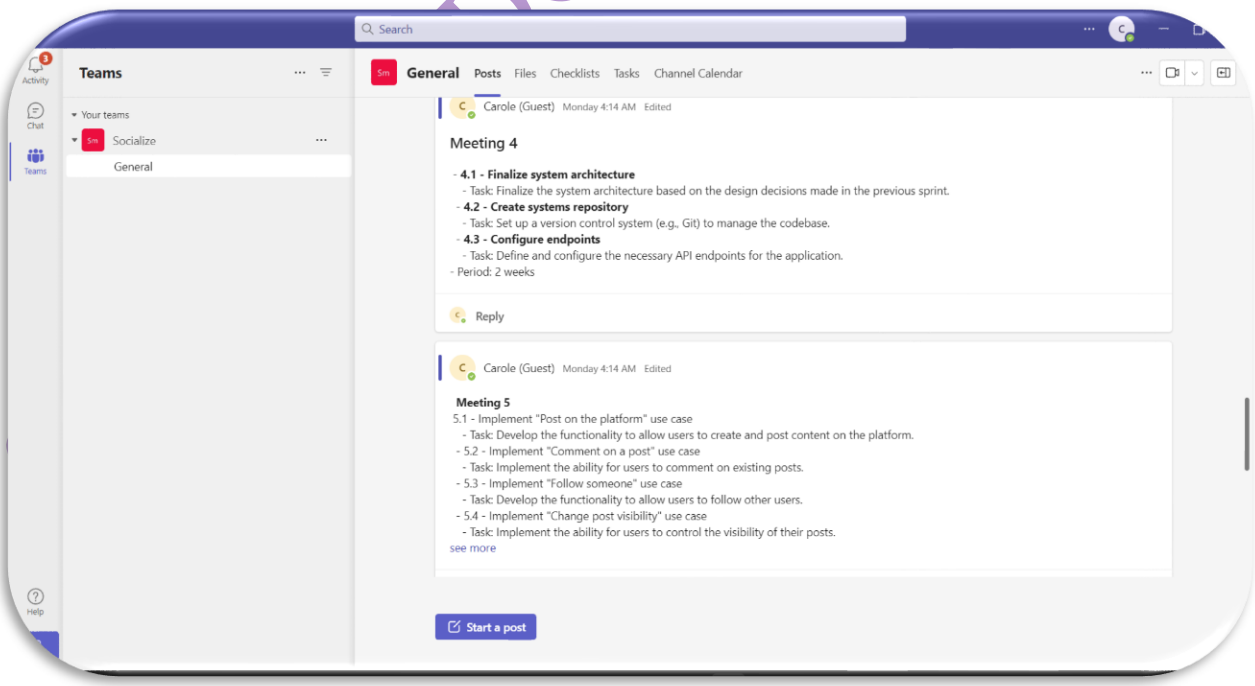




Todos left



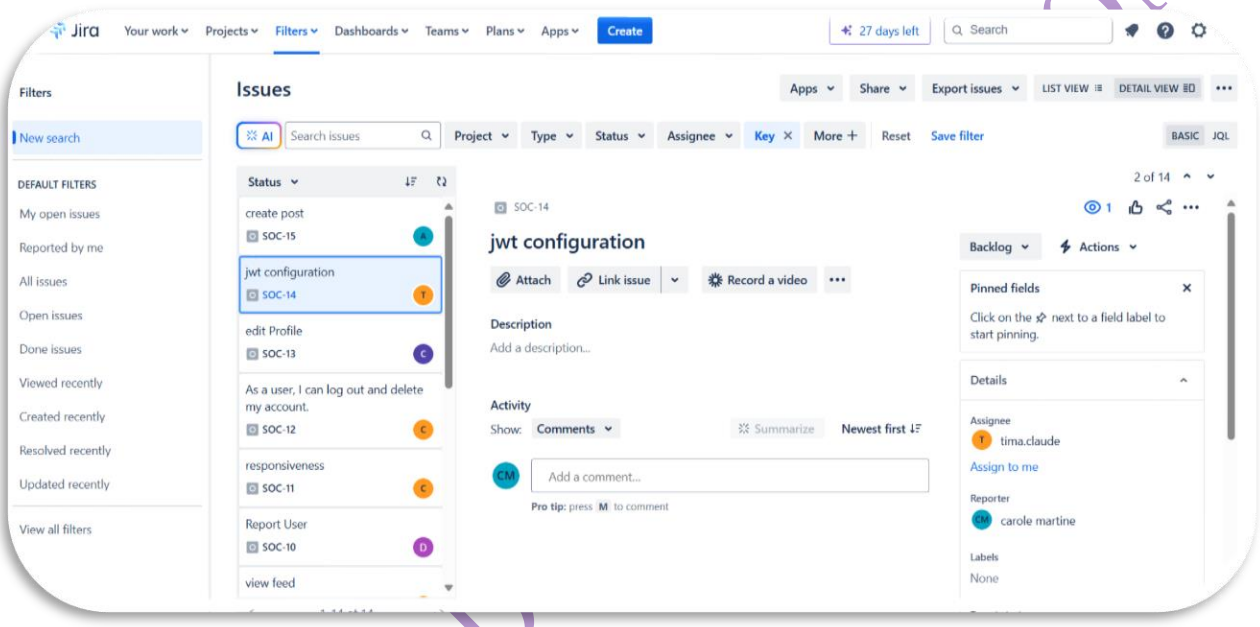
Post/tasks assigned



Link to the conversation:

https://teams.microsoft.com/l/channel/19%3Aqu0alA5OYKuA3B4xxXlS_8V0zhy58fIfttWuv5y0qHw1%40thread.tacv2/General?groupId=734da85f-02c7-4d08-8c48-067f5d68fc2e&tenantId=e4c61d21-2baa-4a7c-910d-cf0bd4bb1208

Jira kanban board:



5.RECOMMENDATIONS AND CONCLUSION

Recommendations:

1. Phased Implementation Approach:

- Adopt a phased implementation approach to manage complexity and risk.

SOCIALIZE DON

- Start with the core social networking functionality, including user profiles, connections, content sharing, and activity streams.

- Gradually expand the feature set based on user feedback and platform needs, such as advanced content moderation, analytics, and monetization capabilities.

2. Seamless Integration:

- Ensure the social features are tightly integrated with the existing Telos platform, providing a cohesive and seamless user experience.

- Leverage the platform's existing infrastructure, APIs, and user base to minimize development efforts and maximize adoption.

3. Community-Centric Design:

- Prioritize the needs and preferences of the Telos community throughout the development process.

- Engage with users, gather feedback, and continuously iterate on the social features to enhance user satisfaction and platform engagement.

4. Scalability and Performance:

- Architect the social features with scalability in mind, considering the anticipated growth in user base and content volume.

- Optimize system performance, reliability, and responsiveness to provide a high-quality user experience.

5. Governance and Moderation:

- Establish clear governance policies and moderation processes to maintain a healthy and productive community.

- Implement robust content moderation tools and user reporting mechanisms to address inappropriate or harmful content.

6. Monetization Strategies:

SOCIALIZE DOCUMENTATION

- Explore monetization opportunities that align with the Telos community's needs and preferences, such as premium features, sponsored content, or platform-specific tokens.

- Ensure any monetization efforts are transparent and do not compromise the user experience or community-driven ethos of the platform.

General Conclusion:

The implementation of social features on the Socialize platform presents a significant opportunity to enhance user engagement, foster a vibrant community, and drive the overall growth and success of the Telos ecosystem.

The feasibility analysis has demonstrated that the proposed social features are technically feasible, economically viable, legally compliant, operationally achievable, and schedulable within the project timeline.

By carefully planning and executing the project, the Socialize team can successfully integrate the social features, creating a more immersive and engaging platform for developers, creators, and users. This, in turn, can lead to increased adoption, stronger brand recognition, and a more sustainable business model for the Telos platform.

With a strategic, user-centric, and phased approach to development and deployment, the Socialize social features project has the potential to become a key differentiator and a valuable addition to the platform's offerings.

SOCIALIZE DOCUMENTATION

SOCIALIZE DOCUMENTATION