Khmelnytskyi National University

Department of Computer Engineering and Information Systems

**Report**

Laboratory work №1

Discipline: “Object-oriented programming”

Topic: “USE CASE ANALYSIS”

Completed: 1st year student, group CEs-24-1 Maksim Lapko

Name, Surname

Checked: Viacheslav Boiko

Name, Surname

Khmelnytskyi, 2024

Purpose: This laboratory work aims to enable students to understand and apply use case analysis as a part of OOAD in the context of Object-Oriented Programming.

**Task 1**

Provide the short description of the domain you are working in. Provide the   
description of your problem domain, what was included and excluded from   
according to the scope of domain.

#### ****Domain****

The domain of this project is an online social networking platform, where users can interact by creating, editing, and sharing content (posts). The platform also provides moderation features and user management tools for administrators.

#### ****Problem Domain****

The problem domain focuses on providing a structured platform where users can safely create and manage posts, interact with other users, and manage their personal profiles. It also addresses the need for administrative oversight to ensure content compliance with platform guidelines, user safety, and management of roles and permissions.

#### ****Included in the Scope****

* **User Registration and Authentication.** Users can create accounts, log in, and manage their profile information.
* **Post Management.** Users can create, edit, delete, view, and like posts.
* **Reporting and Moderation.** Users can report posts for violations, and admins can review and moderate content (edit, delete posts).
* **User Profile Management.** Users can change their password, avatar, nickname, username (login), and email. Admins can manage these attributes for all users.
* **User and Post Search.** Users can search for other users by name or nickname and search for posts by tags.
* **Admin Rights and Permissions.** Admins have different levels of authority for moderating content and managing users, including banning users, assigning roles, and deleting accounts (for level 3 admins).
* **Account Deletion.** Users can delete their accounts, and admins can delete user accounts as well (level 3).

#### ****Excluded from the Scope****

* **Advanced Social Features.** Features like private messaging, real-time chat, or comment sections are not included.
* **Monetization.** There are no premium accounts, advertisements, or in-app purchases.
* **Third-party Integrations.** Integrations with external services like Google or Facebook for authentication are not included.
* **Notifications and Analytics.** Real-time notifications for post interactions, likes, or in-depth user analytics are not part of the initial scope.

**Task 2**

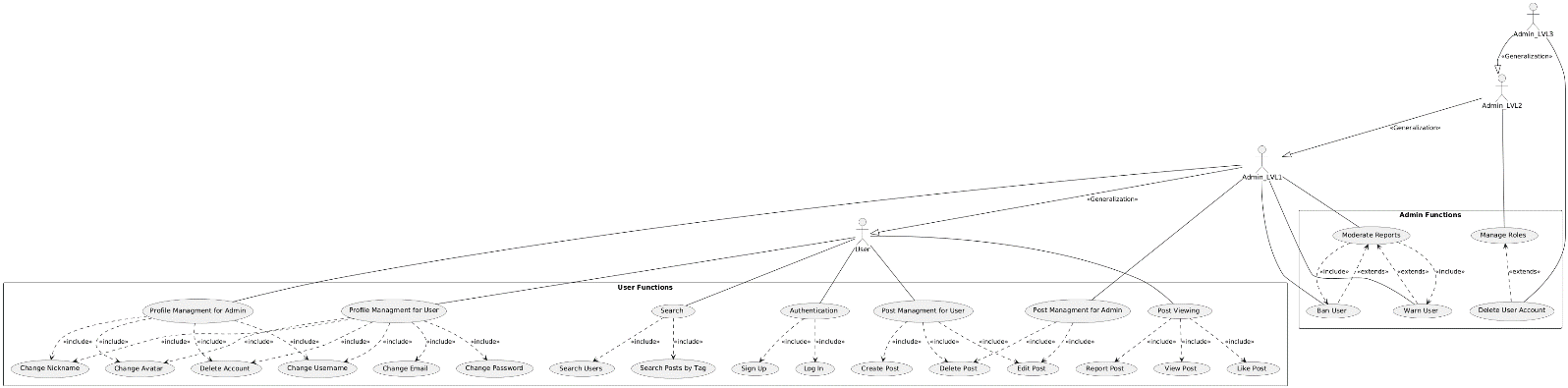
Determine the main notations of your system for use case analysis. Compose

the table with the description of each component.

|  |  |  |
| --- | --- | --- |
| ****Notation Type**** | ****Name**** | ****Description**** |
| **Actor** | **User** | A regular user who can register, log in, create/edit/delete/like posts, report posts, manage their profile, search users and posts, and delete their account. |
| **Actor** | **Admin** | Administrators with levels 1-3 who can moderate posts, process reports, manage users and admins, and perform user account management tasks like editing or deleting posts. |
| **Use Case** | **Sign Up** | A user creates an account using email or username with uniqueness verification. |
| **Use Case** | **Log In** | A user logs in and receives a JWT token for future actions. |
| **Use Case** | **Create Post** | A user creates and shares a post. |
| **Use Case** | **Edit Post** | A user edits their post. Admins can edit posts for all users. |
| **Use Case** | **Delete Post** | A user deletes their post. Admins can delete posts for any user. |
| **Use Case** | **Like Post** | A user likes a post. |
| **Use Case** | **View Post** | A user views posts shared by others. |
| **Use Case** | **Report Post** | A user reports a post for violations. |
| **Use Case** | **Change Profile** | A user changes their password, avatar, nickname, username (login), or email. Admins can also change these details for any user. |
| **Use Case** | **Delete Account** | A user deletes their account. Admins at level 3 can delete any user's account. |
| **Use Case** | **Search Users** | A user searches for other users by name or nickname. |
| **Use Case** | **Search Posts by Tag** | A user searches posts by tags. |
| **Use Case** | **Moderate Reports** | Admins review and take action on reported posts (delete/edit posts, warn users). |
| **Use Case** | **Warn User** | Admins can issue warnings to users for guideline violations. |
| **Use Case** | **Ban User** | Admins can ban users from the platform. |
| **Use Case** | **Manage Roles** | Admins at level 2 and above can manage user/admin roles (assign, promote, demote). |
| **Use Case** | **Delete User Account** | Admins at level 3 can delete any user’s account. |

**Task 3**

Create a detailed Use Case diagram which may contain all previously described notations.



**Diagrams code:**@startuml

actor User

actor Admin\_LVL1

actor Admin\_LVL2

actor Admin\_LVL3

rectangle "User Functions" {

usecase "Authentication"

usecase "Sign Up"

usecase "Log In"

usecase "Post Managment for User"

usecase "Post Managment for Admin"

usecase "Create Post"

usecase "Edit Post"

usecase "Delete Post"

usecase "Post Viewing"

usecase "View Post"

usecase "Like Post"

usecase "Report Post"

usecase "Profile Managment for User"

usecase "Profile Managment for Admin"

usecase "Change Username"

usecase "Change Email"

usecase "Change Password"

usecase "Change Nickname"

usecase "Change Avatar"

usecase "Delete Account"

usecase "Search"

usecase "Search Users"

usecase "Search Posts by Tag"

}

"Authentication" ..> "Sign Up" : <<include>>

"Authentication" ..> "Log In" : <<include>>

"Post Managment for User" ..> "Create Post" : <<include>>

"Post Managment for User" ..> "Edit Post" : <<include>>

"Post Managment for User" ..> "Delete Post" : <<include>>

"Post Managment for Admin" ..> "Edit Post" : <<include>>

"Post Managment for Admin" ..> "Delete Post" : <<include>>

"Post Viewing" ..> "View Post" : <<include>>

"Post Viewing" ..> "Like Post" : <<include>>

"Post Viewing" ..> "Report Post" : <<include>>

"Profile Managment for User" ..> "Change Username" : <<include>>

"Profile Managment for User" ..> "Change Email" : <<include>>

"Profile Managment for User" ..> "Change Password" : <<include>>

"Profile Managment for User" ..> "Change Nickname" : <<include>>

"Profile Managment for User" ..> "Change Avatar" : <<include>>

"Profile Managment for User" ..> "Delete Account" : <<include>>

"Profile Managment for Admin" ..> "Change Username" : <<include>>

"Profile Managment for Admin" ..> "Change Nickname" : <<include>>

"Profile Managment for Admin" ..> "Change Avatar" : <<include>>

"Profile Managment for Admin" ..> "Delete Account" : <<include>>

"Search" ..> "Search Users" : <<include>>

"Search" ..> "Search Posts by Tag" : <<include>>

rectangle "Admin Functions" {

usecase "Moderate Reports"

usecase "Warn User"

usecase "Ban User"

usecase "Manage Roles"

usecase "Delete User Account"

}

Admin\_LVL1 --|> User : <<Generalization>>

Admin\_LVL2 --|> Admin\_LVL1 : <<Generalization>>

Admin\_LVL3 --|> Admin\_LVL2 : <<Generalization>>

User -- "Authentication"

User -- "Post Managment for User"

User -- "Post Viewing"

User -- "Profile Managment for User"

User -- "Search"

Admin\_LVL1 -- "Post Managment for Admin"

Admin\_LVL1 -- "Profile Managment for Admin"

Admin\_LVL1 -- "Moderate Reports"

Admin\_LVL1 -- "Warn User"

Admin\_LVL1 -- "Ban User"

Admin\_LVL2 -- "Manage Roles"

Admin\_LVL3 -- "Delete User Account"

"Moderate Reports" <.. "Warn User" : <<extends>>

"Moderate Reports" <.. "Ban User" : <<extends>>

"Moderate Reports" ..> "Warn User" : <<include>>

"Moderate Reports" ..> "Ban User" : <<include>>

"Manage Roles" <.. "Delete User Account" : <<extends>>

@enduml

**Conclusions**

We successfully defined the interactions between users, admins, and the core functionalities of the Social Network Y platform. By organizing the system into clear actors and use cases, we ensured that both user and admin roles are well-structured, with a clear distinction of permissions.