Khmelnytskyi National University

Department of Computer Engineering and Information Systems

**Report**

Laboratory work №8

Discipline: “Software Requirements and Quality Analysis”

Topic: “Technical task”

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

Name, Surname

Checked: Oksana Onyshko

Name, Surname

Khmelnytskyi, 2024

***Purpose of work:*** To form practical skills from development technical task for design software according to the template of the relevant standards.

**8.1** **Tasks for laboratory work**

1. Develop a technical task according to the selected template

2. Complete the work.

3. Carry out job protection.

**8.2 Completion of the task**

**1. General Information**

* **Project Name**: Social Network Y
* **Customer**: HNU, Department of Software Engineering
* **Developer**: Лапко Максим Олексійович
* **Project Start Date**: [Start Date]
* **Deadline**: [Deadline Date]

**2. Purpose and Objective**

* **Purpose**: To develop and implement a social networking platform with basic communication and interaction features for users.
* **Objective**: Provide a secure, user-friendly, and efficient social platform that enables user interaction through posts, comments, and likes, while allowing administrators to moderate content.

**3. Product Description**

**3.1 Functionalities for Users**

* **Registration and Authentication**: User sign-up and login with email or username, ensuring uniqueness and security.
* **Post Management**: Ability for users to create, edit, and delete posts.
* **Interaction with Posts**: Users can like and comment on posts.
* **Profile Customization**: Users can update their password, avatar, nickname, username, and email.
* **Account Deletion**: Users have the option to delete their accounts permanently.
* **Search Functionality**: Users can search for other users and posts by tags.

**3.2 Functionalities for Administration**

* **Content Moderation**: Administrators can review, edit, and delete user posts and comments.
* **Violation Handling**: Administrators can view and process reports of violations.
* **User and Admin Management**: Ability to manage user roles and permissions, and oversee other administrators.
* **Profile Update for Users**: Administrators can change any user’s avatar, nickname, username, and email.
* **User Banning and Account Deletion**: Admins (level 3) can ban users and delete user accounts.

**4. Functional Requirements**

**4.1 User Functionalities**

* **Registration**:
  + Unique username or email validation during sign-up.
* **Authentication**:
  + User login with JWT token issuance for secure access to system functions.
* **Post Operations**:
  + Users can create, edit, delete, and view their posts.
  + Users can comment and like posts.
* **Profile Updates**:
  + Users can update their password, avatar, nickname, username, and email.
* **Account Deletion**:
  + Users can delete their account permanently.

**4.2 Administrator Functionalities**

* **Post and Comment Moderation**:
  + Administrators can edit or delete any user’s posts and comments.
* **Violation Processing**:
  + Administrators can view and address reports on content violations.
* **Role and Permission Management**:
  + Admins can manage user and other admin roles, with Level 3 administrators able to handle role assignments and bans.
* **User Profile Management**:
  + Admins can update users' profile details, such as avatar, nickname, username, and email.
* **User Ban and Deletion**:
  + Level 3 admins can ban users or delete user accounts.

**5. Non-Functional Requirements**

**5.1 Performance**

* **Reliability**: System uptime should be maintained at 99% to ensure consistent user access.
* **Scalability**: The system should handle concurrent users efficiently, adapting to varying traffic loads without degradation in performance.

**5.2 Usability**

* **User Interface**: The interface should be intuitive and easy to navigate, with clear buttons for main actions and a responsive design across devices.

**5.3 Security**

* **Data Protection**: Ensure user data is encrypted and securely stored.
* **Authentication and Authorization**: Implement JWT-based authentication for secure access and role-based access control for admin functions.

**6. Interface Requirements**

**6.1 User Interface**

* The main menu should include buttons for registration, login, and navigation to posts and profile management sections.
* **Screen Layouts**:
  + Consistent design with standard font and color scheme.
  + Clear visual indicators for accessible functions (e.g., like, comment, report).

**6.2 Hardware Interface**

* The system should support standard desktop and mobile device configurations.

**6.3 Software Interface**

* **Database**: MySQL for data storage, with necessary CRUD operations supported.
* **Backend**: REST APIs for interaction between frontend and backend.

**6.4 Communication Interfaces**

* **API Security**: Secure HTTP (HTTPS) for data transmission.
* **Data Format**: JSON format for API data exchange, with JWT for session management.

**7. Operational Environment**

* **Hardware**: Compatible with general server configurations (minimum 8 GB RAM, 4-core CPU).
* **Operating System**: Linux-based servers (Ubuntu preferred).
* **Development Platform**: Go for backend, React for frontend.

**8. Limitations and Constraints**

* **Compliance**: Follow industry standards for data privacy and security.
* **Language**: English and Ukrainian support in the user interface.
* **Browser Compatibility**: Ensure compatibility with the latest versions of Chrome, Firefox, and Safari.

**Conclusion:**

In this laboratory work, I gained practical skills from development technical task for design software according to the template of the relevant standards like DSTU GOST 34.603–92.