**Software Engineering Project – Documentation**

**Threads-Lite**

**MERNA SAMIR – SARAH EL-SHINAWY**

**Software Requirements Specification (SRS)**

# Introduction:

## Purpose

This document defines the requirements for **Threads Lite**, a simplified social media platform designed for lightweight content sharing and user interactions. This project serves as a student-friendly implementation focusing on core functionalities while maintaining scalability and security.

## Scope

Threads Lite is a **full-stack web application** that enables users to connect, share posts, engage with content, and communicate through real-time messaging. The system includes essential features like **authentication, content sharing, notifications, and a follow-based feed**. It is developed using **.NET for the backend** and **React.js for the frontend**.

# Overall Description:

## Product Perspective

Threads Lite is a **standalone web-based social media platform** focused on simplicity. Unlike full-fledged platforms, it omits complex features such as **video feed, group chat, and advanced recommendation systems**, making it easier to develop and maintain while still providing an engaging user experience.

## Product Features

The platform includes the following key features:

* **User Authentication & Authorization**: Secure login, registration, and role-based access control.
* **User Relations**: Users can follow/unfollow other users.
* **Content Creation & Interaction**: Create, edit, delete posts, and engage with content through likes and comments.
* **Feed System**: Displays recent posts from followed users.
* **Search**: Users can search for other users and posts.
* **Chat System**: Real-time one-on-one messaging using SignalR.
* **Notifications**: Alerts for follows, messages, and post interactions.
* **Multimedia Support**: Users can attach images to posts and comments (videos are not supported).

## User Classes and Characteristics

* **Regular Users**: Can create posts, follow users, engage with content, send messages.
* **Moderators (Optional)**: May have privileges such as content moderation.
* **Administrators**: Oversee platform functionality, manage roles, and enforce policies.

# Functional Requirements

## User Authentication & Authorization

* Users can register, log in, and reset passwords securely.
* Role-based access control (Regular User, Admin).

## User Relations

* Users can follow and unfollow other users.
* Users receive notifications when followed.

## Content Management

* Users can create, edit, and delete posts.
* Users can like and comment on posts.
* Users can attach images to posts and comments.

## Feed System

* Displays posts from followed users in chronological order.
* Pagination for content loading.

## Search Functionality

* Users can search for other users and posts.

## Chat System

* Real-time one-on-one messaging with persistent storage.
* Implemented using **SignalR**.

## Notifications

* Alerts for new messages, follows, and post interactions.
* Limited history for past notifications.

# Non-Functional Requirements

## Performance

* Optimized for fast response times in loading content and real-time messaging.

## Scalability

* Supports multiple concurrent users without significant degradation.

## Security

* Encrypted authentication and role-based access control.

## Usability

* User-friendly UI with intuitive design.

## Maintainability

* Modular codebase for easy updates and customization.

## Reliability

* The system ensures uptime with error handling and recovery mechanisms.

# Tools and Technologies

## Languages:

* **HTML, CSS, JavaScript** – Frontend development.
* **C#** – Backend development (.NET Core).
* **SQL** – Database management.

## Frameworks and Libraries:

* **React.js** – Frontend framework.
* **.NET (ASP.NET Core)** – Backend framework.
* **Entity Framework Core** – ORM for database interactions.
* **SignalR** – Real-time chat functionality.

## Development Tools:

* **Visual Studio / Visual Studio Code** – IDEs for development.
* **Postman** – API testing.
* **Browser Developer Tools** – Debugging and performance analysis.

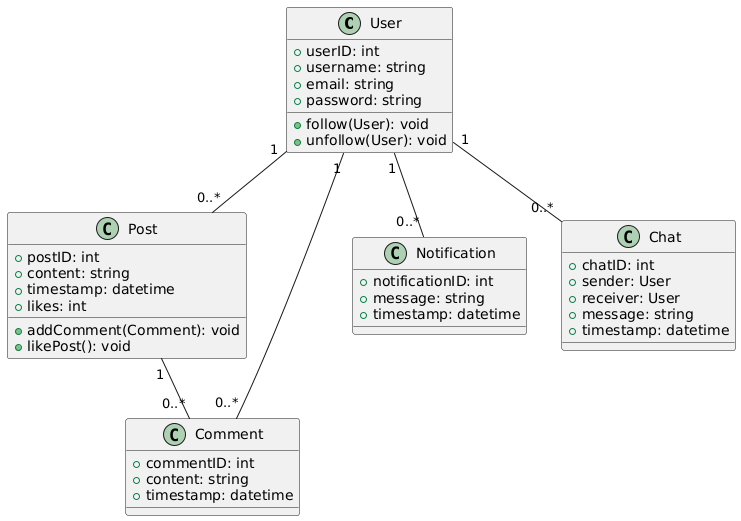
## Version Control:

* **Git & GitHub** – Version control and project collaboration.

# Use case Diagram

## 

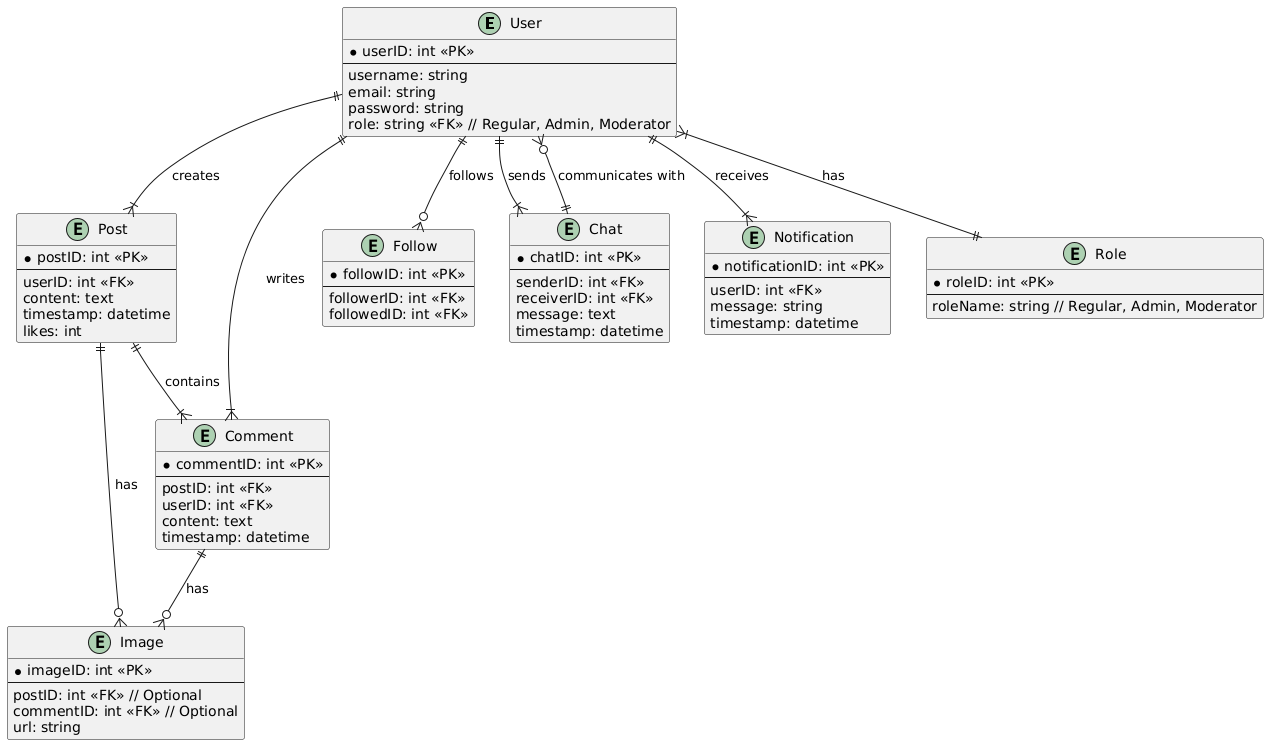
# Class Diagram



# ERD

# 

* **EERD**

****

# Activity Diagram

