# **User Requirements Specification for YansTribe**

By Petar Yankov

Contents

[User Requirements Specification for YansTribe 1](#_Toc197052534)

[1. Introduction 2](#_Toc197052535)

[2. Functional Requirements 2](#_Toc197052536)

[FR-01: User Registration & Authentication 2](#_Toc197052537)

[FR-02: User Profile Management 2](#_Toc197052538)

[FR-03: Post Creation 2](#_Toc197052539)

[FR-04: Post Viewing 2](#_Toc197052540)

[FR-05: Post Editing 2](#_Toc197052541)

[FR-06: Post Deletion 3](#_Toc197052542)

[FR-07: Post Reactions 3](#_Toc197052543)

[FR-05: Comment Section 3](#_Toc197052544)

[FR-06: Comment Creation 3](#_Toc197052545)

[FR-07: Comment Editing 3](#_Toc197052546)

[FR-08: Comment Deletion 3](#_Toc197052547)

[FR-09: Comment Reactions 3](#_Toc197052548)

[FR-10: Stories 3](#_Toc197052549)

[FR-11: Friend Management 3](#_Toc197052550)

[FR-12: Messaging System 3](#_Toc197052551)

[FR-13: Privacy & Security 4](#_Toc197052552)

[FR-14: Admin Controls 4](#_Toc197052553)

[3. Use Cases 4](#_Toc197052554)

[Use Case ID UC-001 4](#_Toc197052555)

[Title: User Registration Actor 4](#_Toc197052556)

[*MSS* 4](#_Toc197052557)

[Use Case ID UC-002 4](#_Toc197052558)

[Title: Create Post Actor 4](#_Toc197052559)

[Use Case ID: UC-003 4](#_Toc197052560)

[Title: Send Friend Request Actor 4](#_Toc197052561)

[Use Case ID: UC-004 4](#_Toc197052562)

[Title: Send Message Actor 4](#_Toc197052563)

[Use Case ID: UC-005 4](#_Toc197052564)

[Title: Delete Comment (Admin) 4](#_Toc197052565)

[4. Non-Functional Requirements 4](#_Toc197052566)

[5. Conclusion: 4](#_Toc197052567)

## Introduction

This web application is a social media platform that integrates features from booth Facebook and Instagram, allowing users to interact through stories, posts, comments, messages, and friendships. Users can personalize their experience by modifying their settings, privacy options, and profile details.

## Functional Requirements

### FR-01: User Registration & Authentication

* Users must be able to sign up, log in and log out securely.
* Password encryption and account verification must be implemented.

### FR-02: User Profile Management

* Users can view and edit their personal details including name, bio, profile picture and password.
* Users can set their account visibility as public or private.

### FR-03: Post Creation

* Users can create posts containing text, images, or videos.
* Users must be able to preview the post before publishing it.
* The preview should display the post exactly as it would appear once published.
* After previewing, users can choose either to go back and edit the content or confirm the post by clicking a **"Post"** button.
* Once confirmed, the post is published and becomes visible based on the user’s privacy settings.

### FR-04: Post Viewing

* Users can view posts from their own profile, friends, and public accounts.
* Posts should be displayed in chronological order or sorted by relevance (e.g., most recent, most liked).
* Users can interact with posts directly from the feed (e.g., like, open comments).
* Users can hide unwanted posts from their feed.

### FR-05: Post Edit

* Users can edit their own posts.
* The system must maintain a record of the last modified timestamp.
* Edited content should update without creating duplicates.

### FR-06: Post Deletion

* Users can delete their own posts.
* Deleted posts must be permanently removed.
* Admins can delete posts that violate community guidelines.

### FR-07: Post Reactions

* Users can like or unlike any post they have access to.
* The system must prevent duplicate likes by the same user.

### FR-05: Comment Section

* Users must be able to see comment section after post.
* Users must be able to hide unwanted comments.

### FR-06: Comment Creation

* Users can add comments to any post they have access to.

### FR-07: Comment Editing

* Users can edit their own comments.
* An “Edit” option must be visible for the user's own comments within the modal.

### FR-08: Comment Deletion

* Users can delete their own comments.
* Admins can delete any comment that violates platform rules.
* A “Delete” option must be available for the comment owner (and admins).

### FR-09: Comment Reactions

* Users can like or unlike any comment they have access to.

### FR-10: Stories

* Users can upload stories that disappear after 24 hours.
* Users can view stories from their friends and public accounts.

### FR-11: Friend Management

* Users can send, accept, and decline friend requests.
* Users can remove friends from their list.

### FR-12: Messaging System

* Users can send private messages to their friends.
* Users can view their message history.

### FR-13: Privacy & Security

* Users can block other users.
* Users can change their password.

### FR-14: Admin Controls

* Admins can manage users, posts and comments.
* Admins can delete accounts violating platform rules.

## Use Cases

### Use Case ID UC-001

### Title: User Registration Actor

#### User Preconditions:

* User is not registered.
* User has a valid email.

#### Trigger:

the user initiates the registration process.

#### Postcondition:

* The user account is created successfully.

#### MSS:

* The system prompts for user details (email, username, password, etc.).
* The user enters the details and submits.
* The system verifies and creates the account.

### Use Case ID UC-002

### Title: Create Post Actor

#### User Preconditions:

* The user is logged in.
* The user has an active account.

#### Trigger:

The user selects the option to create a post.

#### Postcondition:

The post is published successfully.

#### MSS:

1. The system displays the post creation interface.
2. The actor enters the post content (text, image, video, etc.).
3. The actor confirms the post.
4. The system saves and displays the post.

### Use Case ID: UC-003

### Title: Send Friend Request Actor

#### User Preconditions:

* The user is logged in.

#### Trigger:

The user selects another user another user to send a friend request.

#### Postcondition:

Friend request is sent.

#### MSS:

1. The system displays the user profile page.
2. The actor selects button “Send friend request”.
3. The system savaes the friend request in the database.

### Use Case ID: UC-004

### Title: Send Message Actor

#### User Preconditions:

* The user is logged in.
* The recipient is a friend or has allowed messages.

#### Trigger:

The user selects a friend and types a message.

#### Postcondition:

The message is sent successfully.

#### MSS:

1. The system displays the chat interface.
2. The actor types a message and sends it.
3. The system delivers messages in real time.

### Use Case ID: UC-005

### Title: Delete Comment (Admin)

#### Administrator Preconditions:

* The administrator is logged into the admin portal.
* The comment exists in the system.

#### Trigger:

The administrator selects a comment to delete.

#### Postcondition:

The comment is removed.

#### MSS:

1. The system displays the comment list.
2. The actor selects a comment, and choses “Delete”.
3. The system removes the comment from the database.

## Non-Functional Requirements

* **Performance:** The system must handle at least 10, 000 concurrent users.
* **Security:** Implement secure authentication and data encryption.
* **Scalability:** The application must be scalable to accommodate growth.
* **Usability:** The UI must be intuitive and responsive.

## Conclusion:

This document outlines the key functional and non-functional requirements for the social media web app, ensuring a structured approach to development and implementation.