

Introduction

1.1 Introduction

The project titled "PIXTREAM" is a visionary social media platform that seeks to revolutionize the way people connect, share, and express themselves in the digital age. As a hub of creativity and interaction, Pixtream aims to empower users to showcase their talents and ideas through a diverse array of multimedia content while fostering meaningful relationships within a vibrant community. In a world increasingly driven by digital communication and innovation, the significance of platforms that prioritize user engagement, creative freedom, and accessibility cannot be overstated. This initiative is dedicated to addressing the evolving needs of content creators, influencers, and social enthusiasts by providing an ecosystem where simplicity meets functionality. Pixtream is not just a platform for posting photos and videos; it is a space where ideas converge, passions thrive, and individuals can curate their unique identities. By offering advanced features such as intelligent recommendations, detailed content categorization, and robust privacy controls, Pixtream ensures that every user has the tools they need to engage, discover, and grow. At its core, Pixtream is committed to enhancing the user experience by blending innovation with inclusivity. It is designed to adapt to diverse interests, enabling users to share their stories, explore global cultures, and build lasting connections. As we delve into the realms of creative expression and technological advancement, our mission is to provide a transformative platform that goes beyond social interaction, fostering a global community of thinkers, creators, and dreamers. Pixtream aspires to redefine the standards of social media by embracing the principles of creativity, collaboration, and accessibility. It is more than a digital space—it is a journey toward realizing potential, sharing experiences, and shaping the future of online engagement. With a focus on intuitive design and comprehensive support, Pixtream stands poised to be a cornerstone in the digital lives of millions, empowering them to connect, inspire, and excel on a global stage.

1.2 Problem statement

In today's digital landscape, social media platforms have become central to how people connect, share ideas, and express creativity. However, the existing platforms often fail to address the diverse needs of their user base effectively. These limitations hinder the potential for meaningful interaction, creative freedom, and community growth. Users face challenges in protecting their privacy, discovering relevant content, and finding like-minded individuals. Additionally, content creators struggle with a lack of visibility and limited tools to organize their work efficiently. These shortcomings not only diminish user satisfaction but also restrict the overall potential of social media as a medium for creativity and connection.

Pixtream seeks to fill this void by providing a user-centric platform that overcomes these challenges. By introducing enhanced privacy controls, intelligent content

categorization, and dynamic recommendation systems, Pixtream offers a solution that fosters creativity, encourages meaningful interactions, and builds an inclusive online community. The project aims to redefine the standards of social media, ensuring that users can express themselves freely, connect authentically, and explore content that truly resonates with their interests.

1.3 Scope and relevance of the project

The scope of Pixtream extends beyond traditional social media functionalities, offering a comprehensive platform that caters to both individual users and content creators. By integrating advanced features such as detailed content categorization, privacy controls, and personalized user recommendations, Pixtream creates an inclusive environment for users to express their creativity, build meaningful connections, and engage with diverse communities.

Scope:

- **Global Connectivity:** Pixtream provides users with the ability to connect with individuals worldwide, fostering a vibrant and diverse user base.
- **Creative Expression:** A dedicated space for sharing multimedia content, including photos, videos, music, and more, categorized into detailed themes and subcategories.
- **Enhanced Interaction:** Tools such as likes, comments, shares, and private messaging create a dynamic and engaging experience for users.
- **Robust Administration:** Features for addressing complaints, handling reports, and maintaining a safe and respectful community.

Relevance:

In an era where digital platforms shape social and cultural interactions, Pixtream is positioned as a relevant solution to existing gaps in the social media ecosystem. It addresses the rising demand for personalized content discovery, secure interactions, and tools that enable users to explore and showcase their creativity. By prioritizing user satisfaction and adaptability, Pixtream aligns with the evolving needs of a digitally connected world, ensuring it remains a valuable tool for both social and creative endeavors.

1.4 Objectives

- Academic

Enrichment Provide students with the opportunity to access courses and academic programs not available at their home institutions. Foster a global

perspective by exposing students to diverse teaching methods, curriculum structures, and academic environments.

- Personal Development

Foster personal growth and independence by challenging students to adapt to new and unfamiliar situations. Encourage self-reflection and the development of resilience, adaptability, and problem-solving skills.

- Professional and career development

Provide opportunities for internships, research projects, or networking events to enhance students' professional skills and build a global professional network. Support students in gaining a competitive edge in the job market by demonstrating cultural awareness, adaptability, and international experience.

System Analysis

2.1 Introduction

Pixtream is a social media web application designed to facilitate user interaction, content sharing, and engagement. The system allows users to create profiles, share posts, follow other users, like and comment on posts, and send private messages. It also includes features such as blocking/unblocking users, reporting inappropriate content, and providing recommendations for users to follow based on shared interests and location. The platform's backend is powered by PHP and MySQL, while the frontend is built using HTML, CSS (via Bootstrap), and JavaScript. The project aims to provide a user-friendly, secure, and scalable platform for social interactions and content sharing.

2.2 Existing system

Social media platforms such as Facebook, Instagram, Twitter (X), TikTok, and Snapchat dominate the industry. They allow users to connect, share content, and engage with each other through features like posts, stories, messaging, and live streaming. These platforms have been instrumental in shaping digital interactions and fostering global communities.

Limitation of existing system

Existing social media platforms face significant limitations that Pixtream aims to address. Many platforms struggle with organizing content effectively, relying on hashtags or general feeds that make it difficult for users to discover specific posts. Pixtream solves this by offering structured categories and sub-options, making content discovery seamless and tailored. Additionally, most platforms limit user control over their experience, using biased algorithms that promote sensational content. Pixtream counters this with a user-focused feed system that allows customization based on preferences, ensuring a more authentic and personalized experience.

Another major issue is the ineffective handling of inappropriate behavior, as current platforms often rely on slow or flawed moderation systems. Pixtream implements a proactive approach, combining automated tools, community-driven moderation, and clear reporting mechanisms to create a safer space for users. Furthermore, while existing platforms are saturated with disruptive ads and lack community-focused features, Pixtream minimizes advertising and fosters interest-based interactions, offering a user-first, inclusive platform.

2.3 Proposed system

Pixtream is a modern social media platform designed to overcome the limitations of existing systems, focusing on user experience, privacy, and meaningful interactions. It provides structured content categories and sub-options to enhance discoverability, allowing users to find specific content effortlessly. The platform empowers users with customizable feed preferences, offering a tailored experience that reduces algorithmic bias and irrelevant content.

Advantages of Pixtream

- **Enhanced Content Organization:** Simplifies content discovery with structured categories.
- **Personalized User Experience:** Tailors feeds through customizable user preferences.
- **Improved Privacy and Security:** Ensures data protection with transparent policies.
- **Proactive Moderation:** Combines tools and community-driven reporting effectively.
- **Minimal Intrusive Advertising:** Reduces ads, enhancing overall user experience.
- **Community Building:** Fosters interest-based, meaningful user connections.
- **Transparency and Trust:** Builds user trust through transparent algorithms.

Disadvantages of Pixtream

- **Initial User Base Development:** Requires time to attract active users.
- **Operational Costs:** Privacy and moderation demand higher expenses.
- **Scalability Challenges:** Ensuring seamless performance as users grow.
- **Competition:** Faces challenges from established social platforms.
- **Ad Revenue Limitation:** Reduced ads may impact early revenues.

2.4 Feasibility study

The feasibility study for **Pixtream** indicates strong potential for success across technical, economic, operational, and legal dimensions. Technically, the project leverages readily available tools such as PHP, MySQL, Bootstrap, and Node.js, ensuring compatibility and scalability. With the developer's expertise in modern web technologies, the system can be implemented effectively to meet its goals. Economically, the use of open-source technologies minimizes initial costs, while monetization through optional premium features and minimal, interest-based ads ensures long-term financial sustainability. The platform's unique features, such as structured content categorization and personalized user experiences, provide a competitive edge, making it attractive to users and enhancing its revenue potential.

Operationally, Pixtream is designed to address the limitations of existing social media platforms, increasing user adoption rates. Its user-friendly interface ensures accessibility, and efficient moderation tools simplify content management and user

safety. From a legal standpoint, the system prioritizes data protection and privacy, aligning with global regulations to ensure compliance and build trust. These factors collectively establish Pixtream as a technically viable, economically sustainable, operationally practical, and legally compliant solution in the competitive social media landscape.

Technical feasibility

Pixtream's development is technically feasible due to the availability of modern tools, frameworks, and expertise. The system is built using technologies like PHP, MySQL, Bootstrap, and Node.js, which are widely used, well-documented, and capable of supporting a scalable architecture. These technologies ensure that the platform can handle a growing user base and provide a seamless experience.

The system architecture has been designed to integrate features like structured content categorization, dynamic suggestions, and real-time interactions, which are achievable using the selected stack. Thus, Pixtream's technical feasibility is well-established, with minimal risks of technology-related constraints.

Operational feasibility

Pixtream's operational feasibility is strong, as the platform is designed to address the challenges of existing social media systems while ensuring ease of use and effective management. The platform's user interface is intuitive, which ensures a smooth onboarding process and user retention, making it accessible to a wide range of users with varying levels of technical proficiency. The system's features, such as content categorization, personalized feeds, and community-driven moderation, are easy to manage through streamlined processes and automated tools, reducing the burden on moderators and administrators.

Furthermore, Pixtream's focus on proactive moderation and real-time reporting ensures that inappropriate behavior is swiftly addressed, fostering a safe environment for users. The combination of user-driven reporting and automated moderation tools allows the platform to handle large-scale interactions effectively, even as the user base grows. Overall, Pixtream is designed with scalable operations in mind, allowing for smooth day-to-day management while offering an enjoyable and secure user experience.

Economic feasibility

Pixtream's economic feasibility is strong, given its low initial development costs and clear paths to revenue generation. The platform utilizes open-source technologies such as PHP, MySQL, and Bootstrap, which significantly reduce initial software and licensing costs. With the developer's existing expertise, the need for external technical resources is minimized, further lowering development expenses.

The primary revenue model for Pixtream includes optional premium features for users who want an enhanced experience and non-intrusive, interest-based advertisements. By minimizing ads and focusing on user-driven content, Pixtream can attract a loyal user base, offering advertisers a more engaged audience for targeted campaigns. Over time, these monetization strategies, combined with Pixtream's unique value proposition, ensure the platform's potential for profitability. As the user base grows, Pixtream can explore additional revenue streams, such as partnerships, sponsored content, or subscription plans, contributing to long-term financial sustainability.

2.5 Software engineering paradigm applied

For the development of Pixtream, the Agile software engineering paradigm is applied to ensure flexibility, iterative progress, and continuous user feedback integration throughout the development lifecycle. Agile methodology focuses on delivering small, functional components of the platform in short cycles (sprints), allowing the development team to adjust based on feedback and changing requirements. This approach supports Pixtream's goal of creating a user-centric, dynamic social media platform, as it facilitates quick adaptations to user needs and market demands.

Additionally, Object-Oriented Programming (OOP) principles are applied to structure the software in a modular, maintainable way. Using OOP enables reusable code components, such as user authentication, post management, and comment systems, ensuring scalability and ease of maintenance as the platform evolves. The combination of Agile for flexible development and OOP for clean, organized code ensures that Pixtream can meet both user expectations and long-term technical goals efficiently.

System Design

3.1 Introduction

System design is a crucial phase in the development of a new system, where the overall solution to the problem is created and detailed. This phase focuses on transforming the feasible system into a structured, functional model that meets user requirements. System design can be divided into two key stages: logical design and physical design. During the logical design phase, the system analyst outlines the system's inputs (sources), outputs (destinations), databases (data storage), and procedures (data flows), ensuring that all elements align with the users' needs.

As the design progresses, the system evolves from a conceptual model in the logical design phase to a detailed, implementable model in the physical design phase. The system analyst must clearly understand the objectives the design aims to achieve, which guides the creation of input data structures and master files (databases) to support the required outputs. The operational and processing aspects are addressed through program construction and testing. The system design includes key components such as:

- **Output Design:** Defining the format and structure of the system's outputs.
- **Database Design:** Structuring the data and its relationships to meet system requirements.
- **Input Design:** Designing the way data will be entered into the system.
- **Form Design:** Creating user-friendly forms for interaction.
- **Architectural Design:** Defining the system's infrastructure, including hardware and software components.
- **System Modules:** Designing functional units or modules within the system for specific tasks.

This comprehensive approach ensures that the system is efficient, scalable, and aligned with the objectives of the proposed solution.

3.2 Database design

Database design is the logical structuring of data storage, where data is organized into records and tables with specific fields that are not directly visible to the user but form the backbone of the system. A database is essentially a collection of data that helps the system efficiently manage and store information, and it is managed by a Database Management System (DBMS). The DBMS imposes constraints like integrity constraints, which include primary keys, unique keys, and referential integrity, ensuring accurate and efficient storage and retrieval of data.

These constraints help in managing data consistency and reliability across the system, particularly when dealing with concurrent data access. Normalization plays a key role in this process by breaking down complex table structures into simpler, more manageable tables, following specific rules to eliminate redundancy and minimize data inconsistencies. This approach reduces disk space usage, enhances

performance, and prevents anomalies such as update, insert, and delete anomalies. Overall, database design and normalization are critical for ensuring that data is stored and accessed efficiently, leading to improved system performance and data integrity.

Table design

Table: users

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each user
name	VARCHAR(255)	NOT NULL	User's full name
username	VARCHAR(255)	UNIQUE, NOT NULL	User's unique username
email	VARCHAR(255)	UNIQUE, NOT NULL	User's email address
password	VARCHAR(255)	NOT NULL	User's password
bio	TEXT	NULL	User's biography or description
profile_picture	VARCHAR(255)	NULL	Path to user's profile picture
location	VARCHAR(255)	NULL	User's location
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Account creation time

Table: posts

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each post
username	VARCHAR(255)	NOT NULL	User's username who posted the content
post_path	VARCHAR(255)	NOT NULL	Path to the post content (image/video)
caption	LONGTEXT	NULL	Caption or description for the post
category	SET	NOT NULL	Category of the post, e.g., photos, videos, etc.
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time when the post was created

Table: messages

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each message
sender_id	BIGINT	NOT NULL	ID of the sender
receiver_id	BIGINT	NOT NULL	ID of the receiver
message	TEXT	NOT NULL	Content of the message
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time when the message was sent

Table: follows

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each follow relationship
follower_id	BIGINT	NOT NULL	ID of the user who follows
followed_id	BIGINT	NOT NULL	ID of the user being followed
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time when the follow action occurred

Table: user_blocks

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each block
blocked_by	BIGINT	NOT NULL	ID of the user who blocked
blocked_user	BIGINT	NOT NULL	ID of the user being blocked
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time when the block occurred

Table: likes

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each like
user_id	BIGINT	NOT NULL	ID of the user who liked the post
post_id	BIGINT	NOT NULL	ID of the post that was liked
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time when the like occurred

Table: comments

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each comment
user_id	BIGINT	NOT NULL	ID of the user who commented
post_id	BIGINT	NOT NULL	ID of the post that was commented on
comment	TEXT	NOT NULL	Content of the comment
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time when the comment was posted

Table: shares

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each share
from_user_id	BIGINT	NOT NULL	ID of the user sharing the post
to_user_id	BIGINT	NOT NULL	ID of the user receiving the share
post_id	BIGINT	NOT NULL	ID of the post being shared
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time when the share occurred

Table: saved_posts

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each saved post
user_id	BIGINT	NOT NULL	ID of the user who saved the post
post_id	BIGINT	NOT NULL	ID of the post that was saved
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time when the post was saved

Table: categories

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each category
category_name	VARCHAR(255)	NOT NULL	Name of the category

Table: options

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each option
category_id	BIGINT	NOT NULL	ID of the associated category
option_name	VARCHAR(255)	NOT NULL	Name of the option within the category

Table: user selections

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each user selection
user_id	BIGINT	NOT NULL	ID of the user who made the selection
option_id	BIGINT	NOT NULL	ID of the selected option

Table: admin

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each admin
username	VARCHAR(255)	NOT NULL	Admin's username
email	VARCHAR(255)	NOT NULL	Admin's email address
password	VARCHAR(255)	NOT NULL	Admin's password
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time when the admin account was created

Table: complaints

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each complaint
user_id	BIGINT	NOT NULL	ID of the user filing the complaint
complaint_text	TEXT	NOT NULL	Text content of the complaint
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time when the complaint was made

Table: reports

Field	Datatype	Constraint	Description
id	BIGINT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each report
reported_by	BIGINT	NOT NULL	ID of the user making the report
reported_user	BIGINT	NOT NULL	ID of the reported user
approval	BOOLEAN	DEFAULT FALSE	Approval status of the report
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time when the report was made

3.3 Object Oriented Design-UML diagrams

UML (Unified Modeling Language) is a standard language used in software engineering to specify, visualize, and document the design of a system. It serves as a blueprint for both the design and development of a system, making the entire process easier to understand, modify, and communicate to different stakeholders. UML diagrams allow for the representation of the entities involved in a system and their interactions, helping software engineers plan and model complex systems.

In the context of Pixtream, Object-Oriented Design (OOD) is applied to break down the system into objects representing real-world entities, such as users, posts, messages, and comments. UML diagrams serve as a visual representation of these objects and their relationships. The following UML diagrams are typically used in Object-Oriented Design:

They are as follows:

- Use case diagram
- Activity diagram
- Sequence diagram
- Class diagram

Use case Diagrams:

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor. An actor represents an outside entity- either human or technological. Use case diagrams can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can't do. Use case Diagram consists of use cases and actors and shows the interaction between the use case and actors. The purpose is to show the interactions between use cases and actor. To represent the system requirements from user's perspective. It must be remembered that the use- cases are the functions that are to be performed in the module. An actor could be the end-user of the system or an external system.

Activity Diagram:

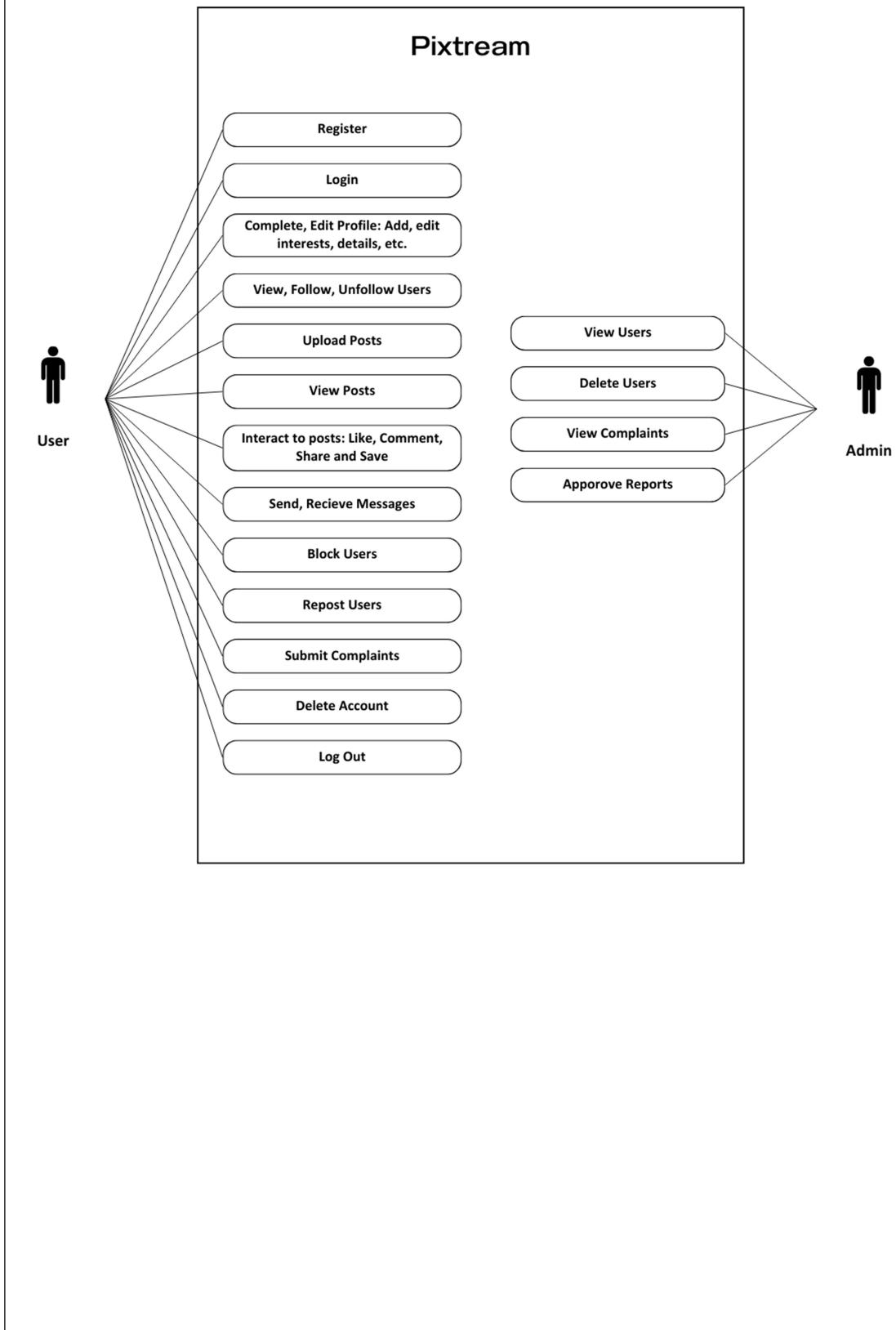
The purpose is to show the activities which the users performed. Activities are shown parallel and sequentially in which order they are performed. Some activities are joined and split according to its flow. Flow of data is represented using arrows.

Sequence Diagram:

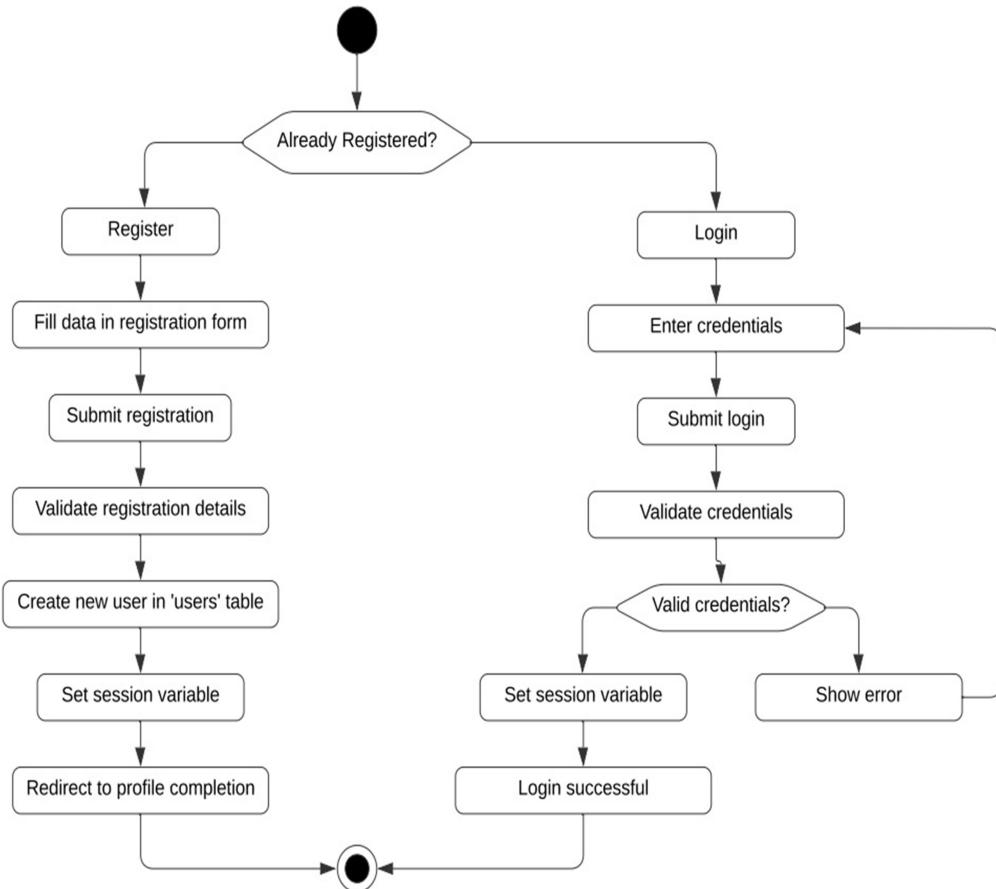
The purpose is to show the sequential flow through of activities. In other Words, we call it mapping processes in terms of data transfers from the actor through corresponding objects. To represent the logical flow of data with respect to a process. It must be remembered that the sequence diagram display objects and not the classes.

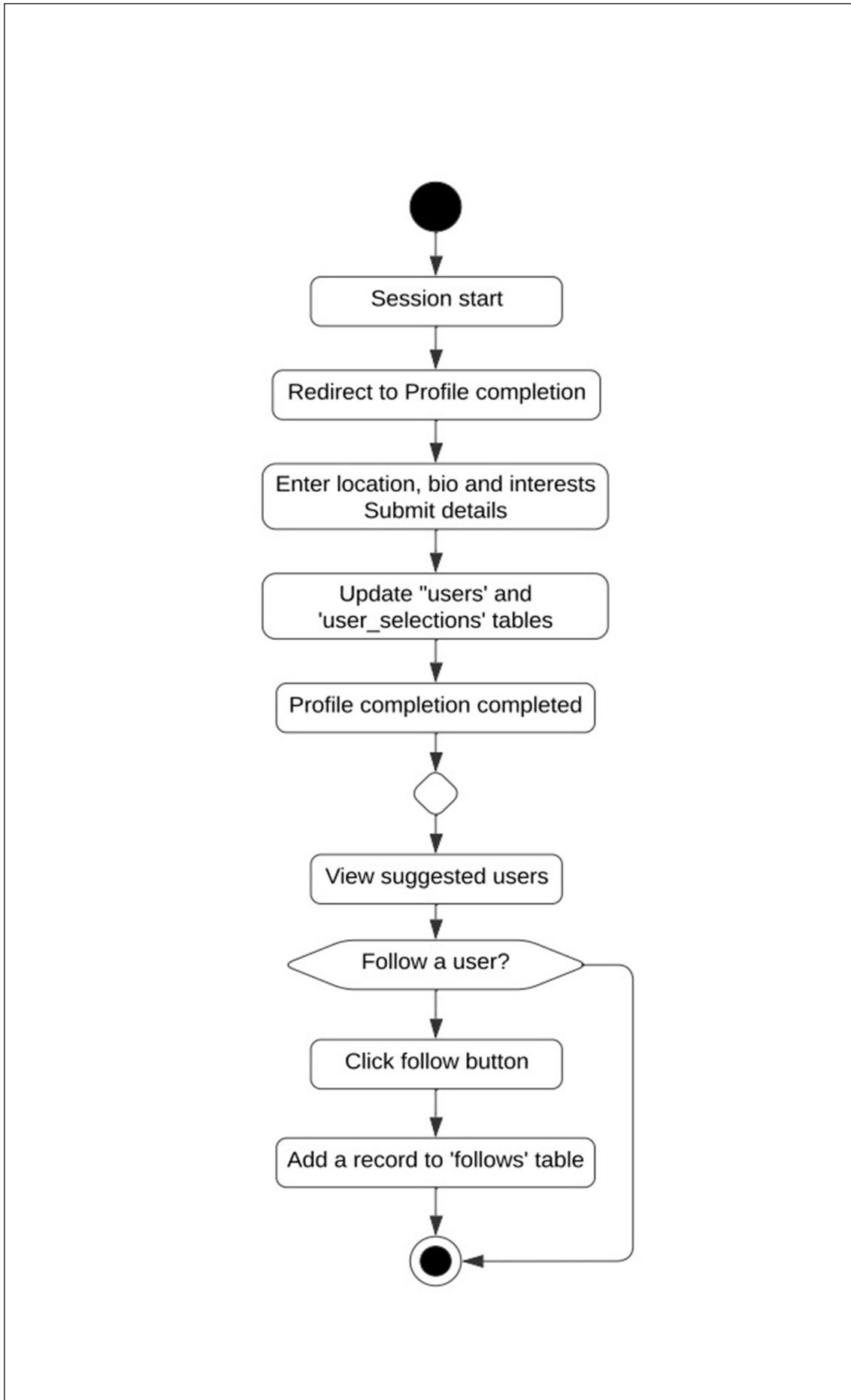
Class Diagram:

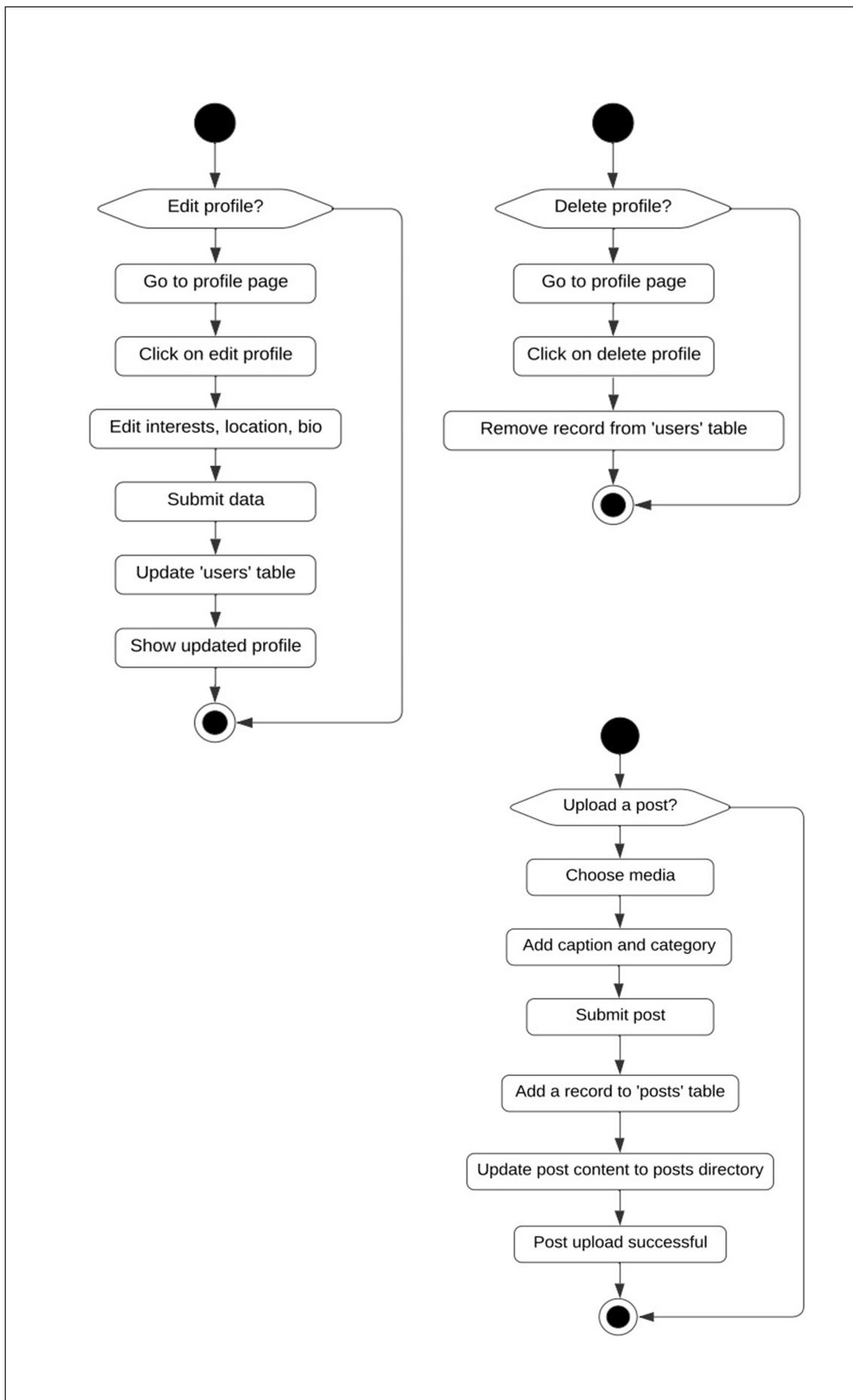
This is one of the most important of the diagrams in development. The diagram breaks the class into three layers. One has the name, the second describes its attributes and the third its methods. The private attributes are represented by a padlock to left of the name. The relationships are drawn between the classes. Developers use the Class Diagram to develop the classes. Analyses use it to show the details of the system. Architects look at class diagrams to see if any class has too many functions and see if they are required to be split.

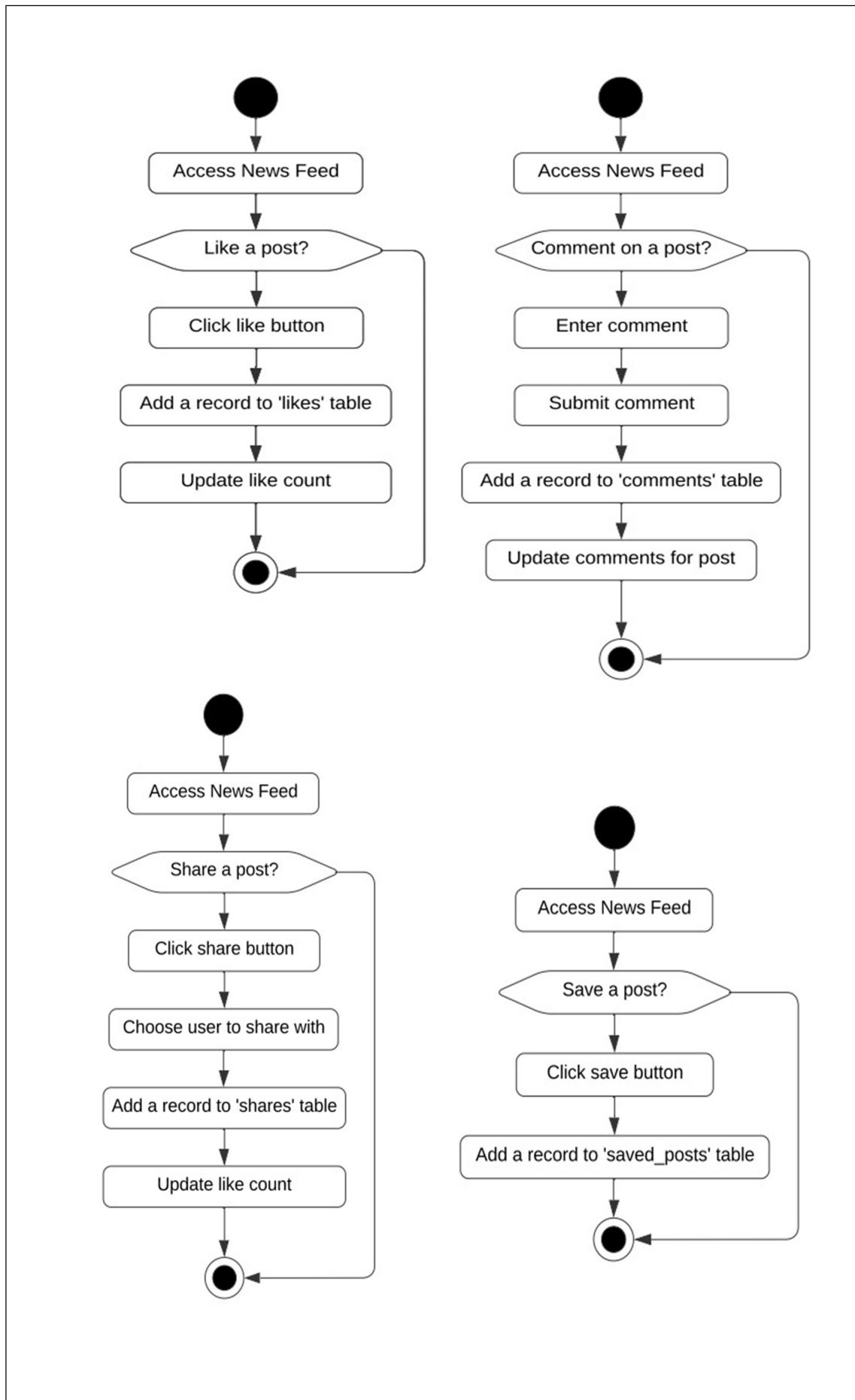
Use case Diagram

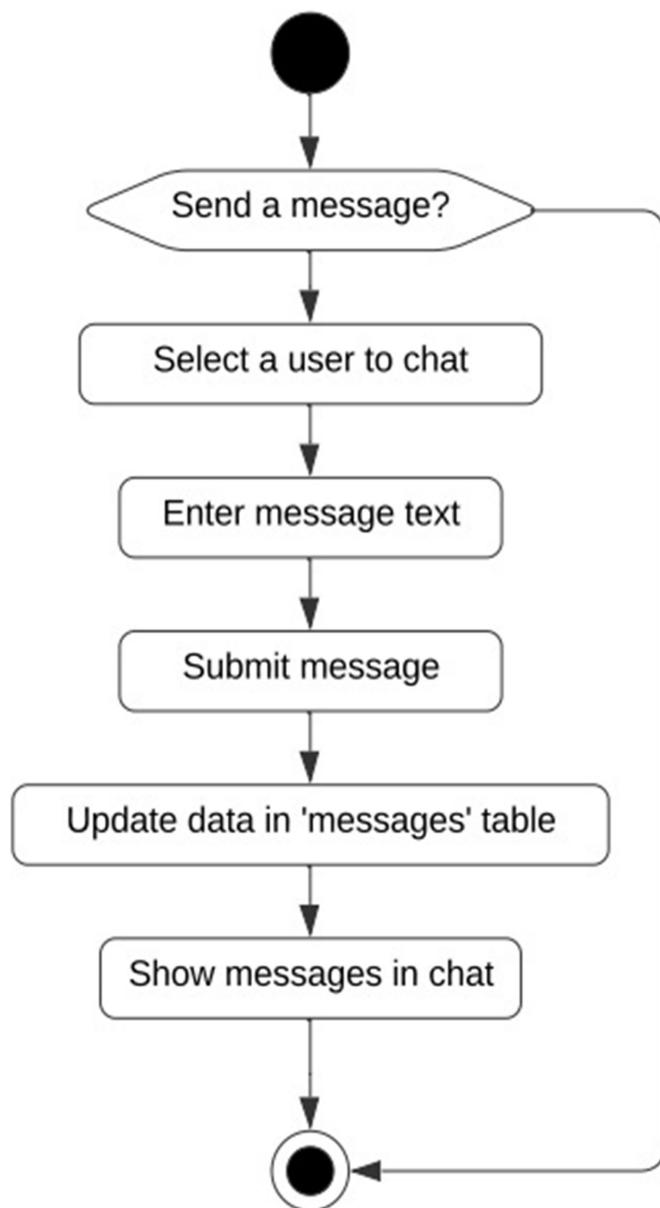
Activity diagram – User

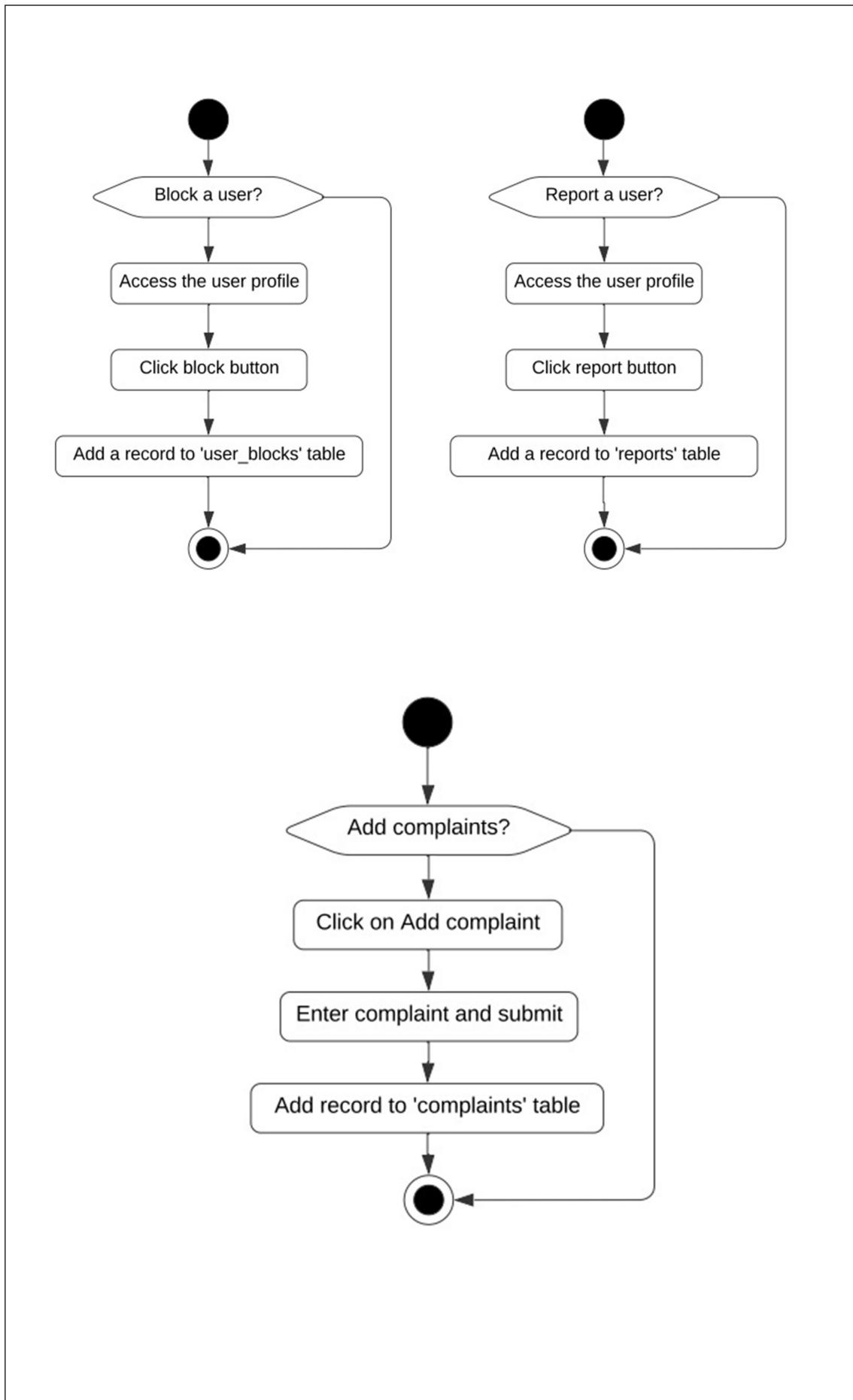


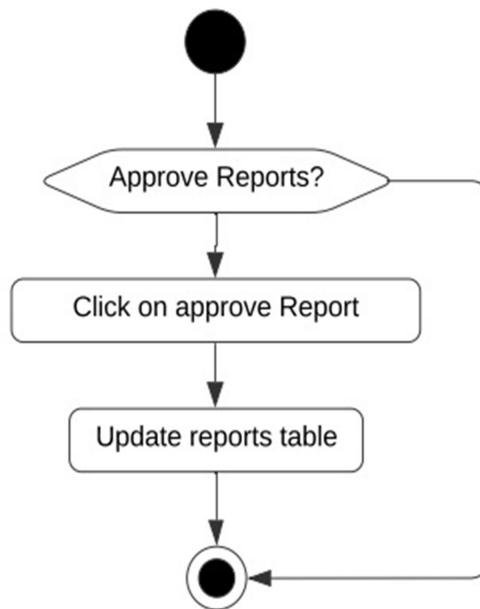
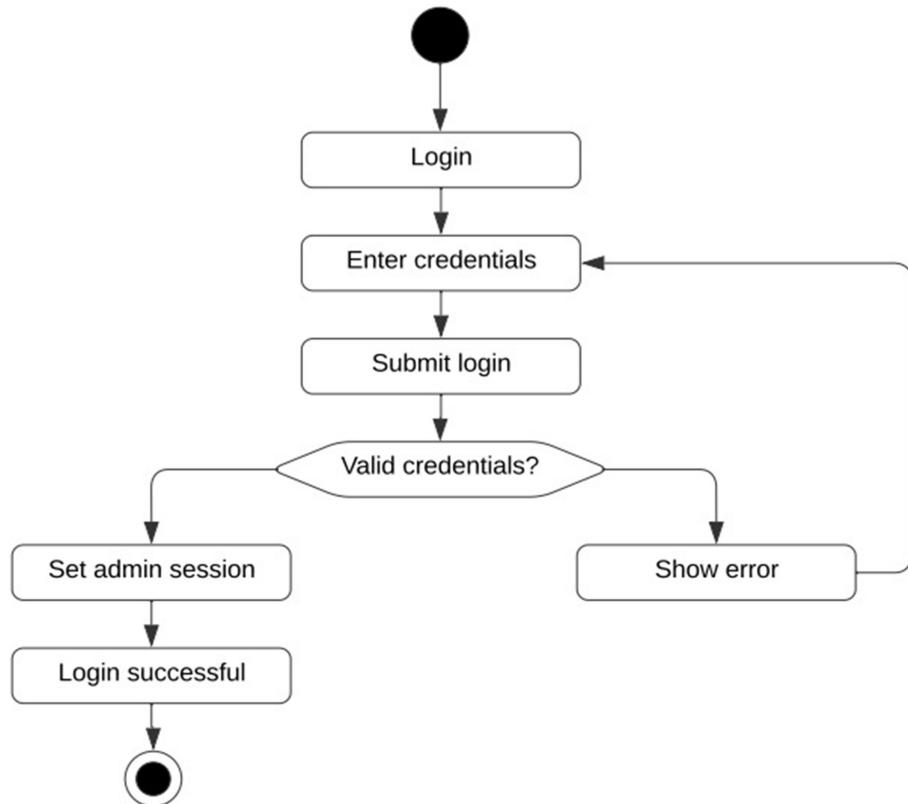


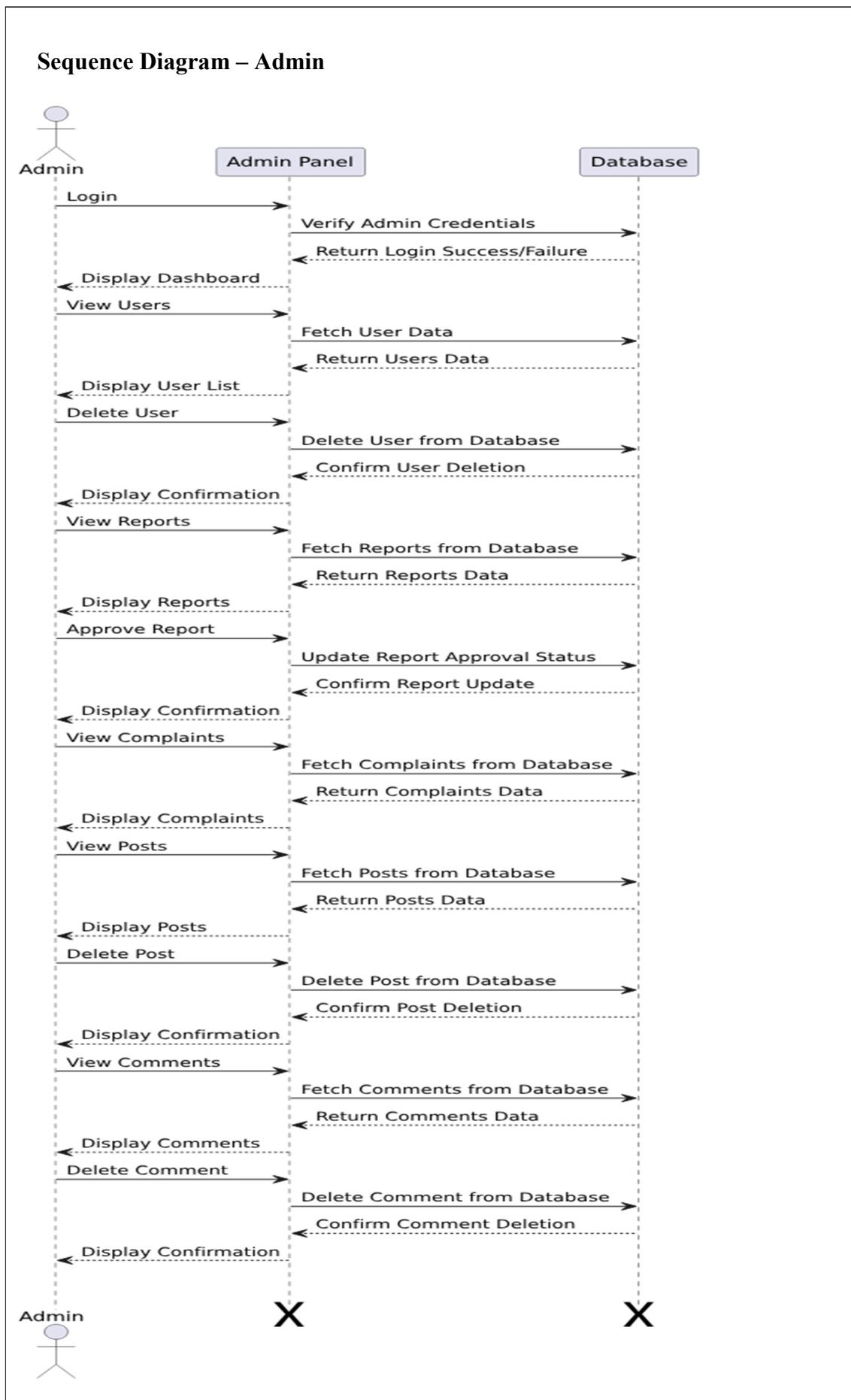




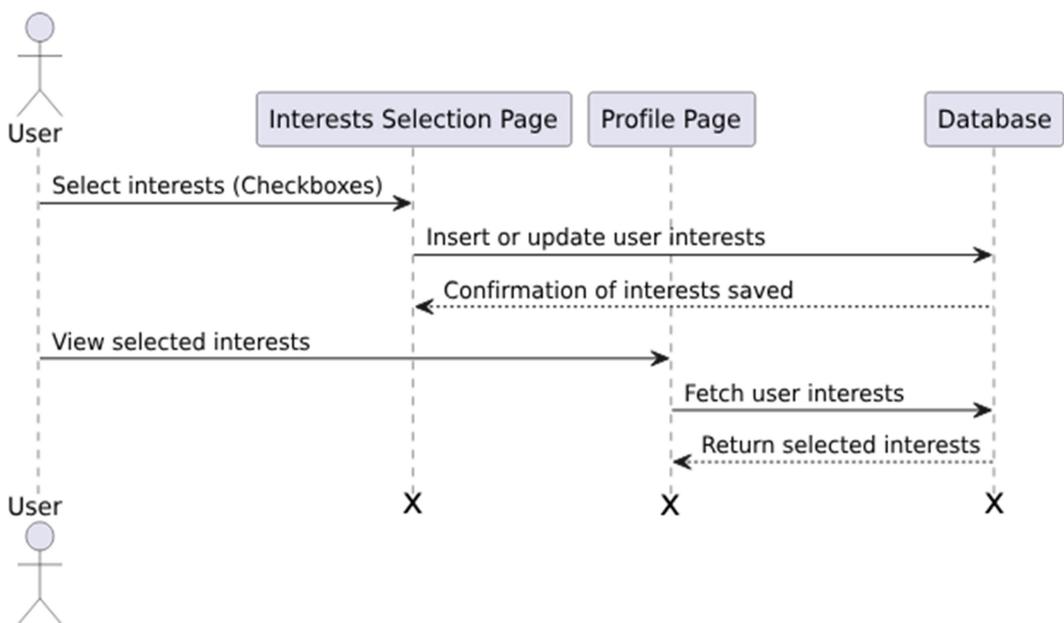
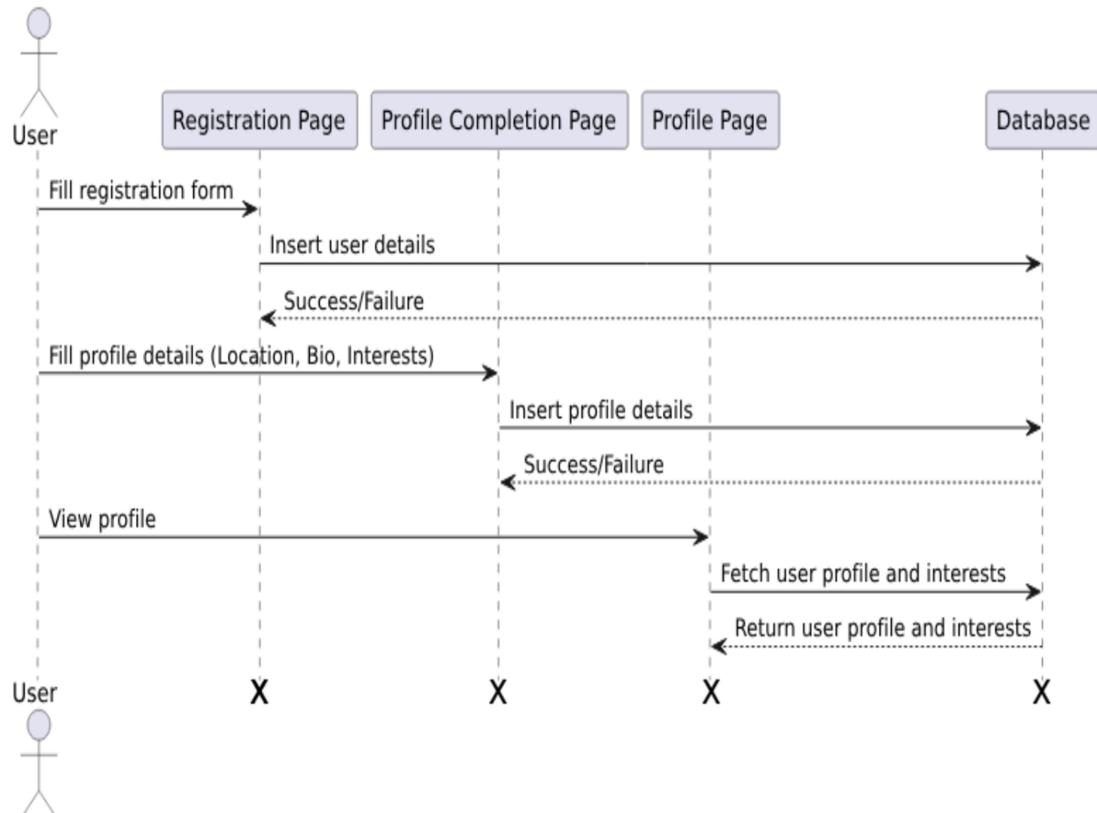


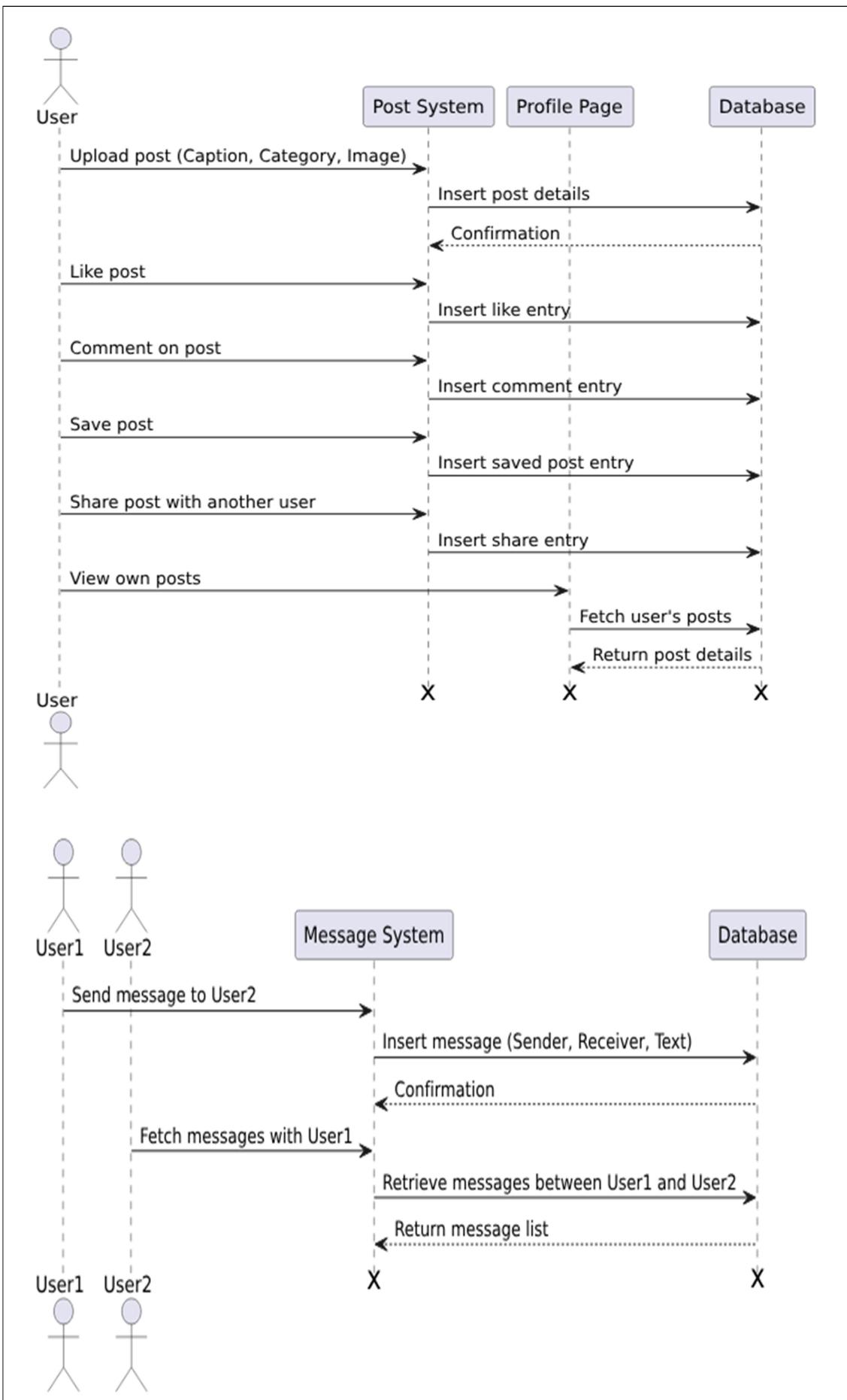


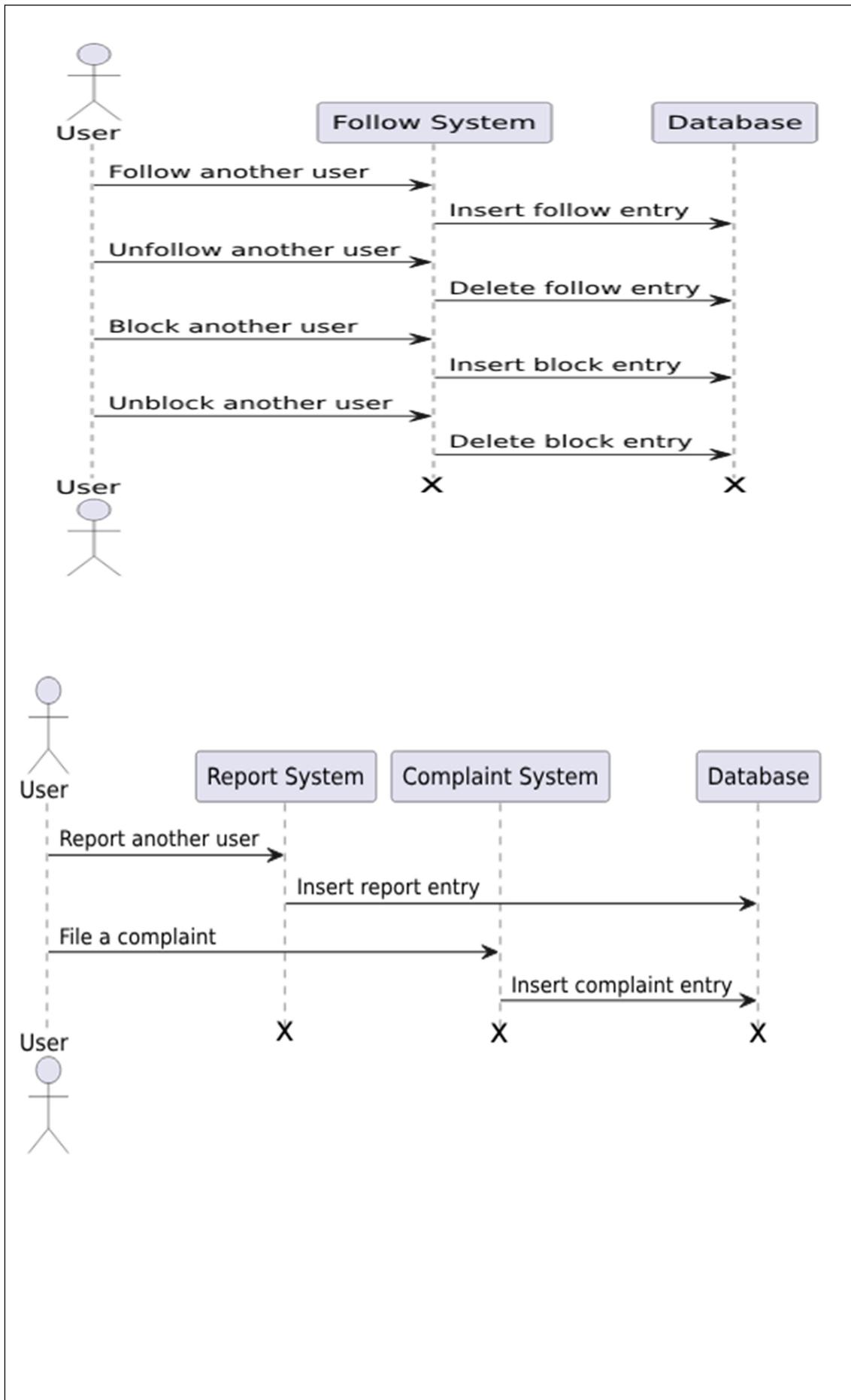
Activity Diagram – Admin



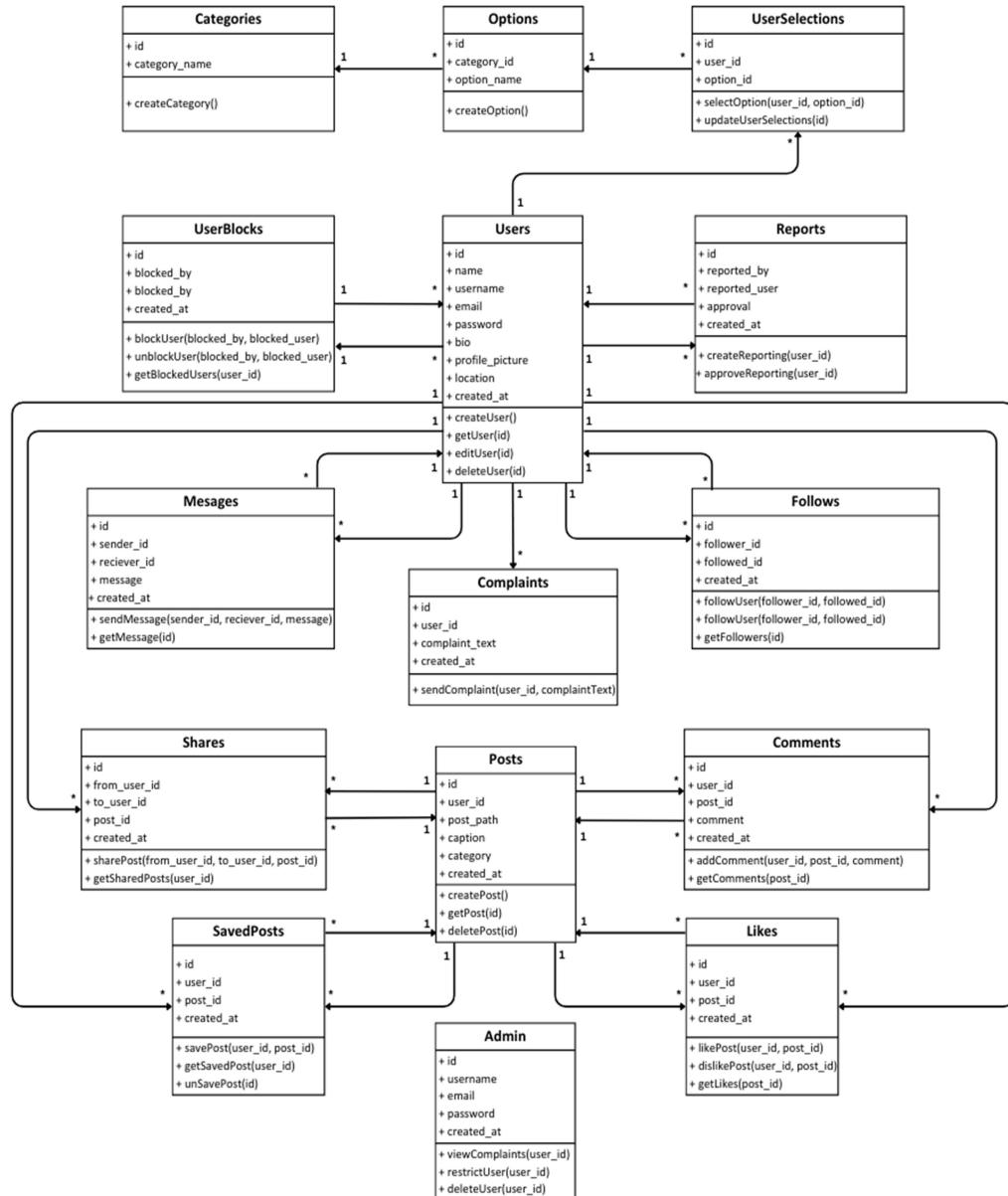
Sequence Diagram - User







Class Diagram



5.4 MODULES DESCRIPTION

A software system is always divided into several sub systems that makes it easier for the development. A software system that is structured into several subsystems makes it easy for the development and testing. The different subsystems are known as the modules and the process of dividing an entire system into subsystems is known as modularization or decomposition. The system under consideration has been divided into several modules taking in consideration the above-mentioned criteria. Modules are

User

The User Module is responsible for managing user-related functionalities within Pixtream, allowing users to interact with the platform. This module includes user registration, login, profile management, content creation, and social interactions. The user is able to upload posts, follow/unfollow other users, send and receive messages, and engage with posts through likes, comments, and shares. The module also allows for managing blocked users and reporting inappropriate content.

- User Registration: Allows new users to sign up by providing personal details such as name, email, username, password, and location.
- User Login: Provides authentication for registered users to access their accounts.
- Profile Management: Enables users to update their profile information, including bio, profile picture, and personal preferences.
- Post Management: Users can upload, edit, and delete posts, including photos, videos, and other types of content.
- Social Interactions: Follow and unfollow other users, send and receive private messages, like and comment on posts, share posts, and save posts.
- Block/Unblock Users: Manage relationships with other users by blocking or unblocking them, preventing unwanted interactions.
- Report Content: Allows users to report inappropriate or offensive content to the admin for further action.

Admin

The Admin Module is designed for administrative management of the platform, allowing authorized personnel to oversee user activity, content moderation, and system management. This module enables the admin to perform tasks such as user management, content management, managing complaints and reports, and maintaining system integrity.

- User Management: Admin can view, edit, and delete user accounts, including resetting passwords, updating profile information, and suspending accounts for violations.
- Content Moderation: Admin has the ability to review and remove offensive or inappropriate content (posts, comments, etc.), ensuring the platform adheres to its guidelines.
- Complaints Management: Admin can view and manage complaints submitted by users about other users or content.

- Reports Management: Admin can review and approve or reject reports submitted by users regarding inappropriate behavior or content.
- System Monitoring: Admin can monitor system performance, view user statistics, and handle maintenance tasks to ensure the platform runs smoothly.

3.4 Input design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing data entry. The activity of putting data into the computer for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system.

- What data should be given as input?
- The dialogue to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error Occur.

For User

- **Registration**
Users input their name, email, username, password, and location to create an account.
- **Login**
Users provide their username and password to access the system.
- **Upload Post**
Users upload media files, captions, and select a category for their posts.
- **Comment on Posts**
Users input text to comment on posts.
- **Send Messages**
Users type messages to communicate with others in the messaging feature.
- **Search Users or Posts**
Users input search queries to find specific users or posts.
- **Report Content or Users**
Users provide reasons for reporting posts or other users.

For Admin

- **Login**
Admins provide their username and password to access the admin panel.
- **Moderate Posts**
Admins input decisions (approve, delete, or warn) for flagged posts.

- **Handle Complaints**
Admins input resolution statuses (resolved, rejected) and notes for user complaints.
- **Manage Users**
Admins input actions (deactivate, warn) for user accounts.
- **Add Categories/Options**
Admins input new categories or options for posts and user selections.

3.5 Output design

Computer output is the most important and direct information source to the user. Output design is a process that involves designing necessary outputs in the form of reports that should be given to the users according to the requirements. Efficient, intelligible output design should improve the system's relationship with the user and help in decision making. So, while designing output the following things are to be considered.

- Determine what information to present
- Arrange the presentation of information in an acceptable format
- Decide how to distribute the output to intended receipts.

For User

- **View Posts**
Users can view posts from their network or the explore feed, along with like, comment, and share options.
- **View Profile**
Users can see their own or others' profiles, including bio, posts, followers, and following.
- **View Notifications**
Users receive notifications for new likes, comments, follows, or shared posts.
- **View Messages**
Users access their messages in a chat interface with timestamps.
- **View Saved Posts**
Users can see a gallery of posts they have saved.

For Admins

- **View Users**
Admins see a list of all registered users with details like email, registration date, and profile information.

- **View Complaints**
Admins can view complaints, including the user's complaint text, submission date, and status.
- **View Reports**
Admins review reports with details about the reporting user, the reported user, and the type of issue.
- **Dashboard Overview**
Admins access a dashboard showing summarized platform statistics, including active users, total posts, complaints resolved, and flagged content.
- **View System Logs**
Admins monitor system logs for activities like logins, reports, or content moderation actions.

System Environment

4.1 Introduction

It encompasses the hardware, software, data, personnel, and networks that collectively create the context within which a software system operates. Understanding and carefully managing the system environment is essential for ensuring optimal performance, reliability, and scalability of the software. This introduction provides an overview of the significance of the system environment in the realm of software projects.

4.2 Software requirement specification

Operating System : Microsoft Windows 11 Home Single Language

Front End : PHP, HTML, JavaScript,CSS

Back End : MySQL

Server : Apache

Software Used : Visual Studio Code, WAMP

4.3 Hardware requirements specification

Processor : Intel(R) Core(TM) i5 or higher

Primary : 512.0MB RAM or higher

Secondary : 2.0GB hard disc or higher

Monitor : CRT or TFT or higher

Keyboard : 104 K

Pointing device : 2 or 3 button mouse

4.4 Tools, Platforms

Front end tool

The front-end of an application is distinctly human. It's what the user sees, touches and experiences. In this respect, empathy is a required characteristic of a good front-end developer. The front-end of an application is less about code and more about how a user will interpret the interface into an experience. That experience can be the difference between a billion-dollar company and complete collapse. If you were Myspace user in 2004, you were probably content with the experience. But once you started to use Facebook, you almost certainly had a better experience. You realized that you could socialize with a simpler design, no flashing banner ads, easy-to-find friends, etc. Facebook and Myspace had a lot of differences under the hood as well (back-end), but at least part of Facebook's triumph can be attributed to a better front-end and user experience.

The technologies used in front-end development commonly include:

HTML – All code in a web application is eventually translated to HTML. It's the language that web browsers understand and use to display information to users. A

web developer's understanding of HTML is analogous to a carpenter's understanding of a screwdriver. It's so important and necessary that it's often assumed for employment.

CSS – By itself, HTML is quite plain. HTML does provide some basic style options, but to build a good front-end, developers must have experience with CSS. CSS provides the paint, templates, glitter, buttons, tassel, lights, and many other things that can be used to improve the presentation of a web page. CSS is so commonly used that languages have been built to make writing CSS easier. These languages – like Sass and LESS – are also known as CSS pre-compilers, but they are simply used to write more efficient and manageable CSS code.

JavaScript – If you could only learn one language in your lifetime, you'd be well advised to choose JavaScript. Though it's not exclusively a front-end language, that's where it's most commonly used. JavaScript is a language that is run on a client machine, i.e., a user's computer. This means that JavaScript can be used to program fast, intuitive and fun user experiences, without forcing a user to refresh their web page. Drag-and drop, infinite-scroll and videos that come to life on a web page can all be programmed with JavaScript. JavaScript is so popular that entire frameworks have been built just to make building application front-ends easier. Frameworks like Angular, Ember, React and Backbone are all very widely used for JavaScript-heavy front-ends.

Back end tool

The back-end of a web application is an enabler for a front-end experience. An application's front-end may be the most beautifully crafted web page, but if the application itself doesn't work, the application will be a failure. The back-end of an application is responsible for things like calculations, business logic, database interactions, and performance. Most of the code that is required to make an application work will be done on the back-end. Back-end code is run on the server, as opposed to the client. This means that back-end developers not only need to understand programming languages and databases, but they must have an understanding of server architecture as well. If an application is slow, crashes often, or constantly throws errors at users, it's likely because of back-end problems.

PHP

PHP is an acronym for "PHP: Hypertext Pre-processor". PHP is a widely-used, open-source scripting language. PHP scripts are executed on the server. PHP is free to download and use. PHP is an amazing and popular language. It is powerful enough to be at the core of the biggest blogging system on the web. It is deep enough to run the largest social network. It is also easy enough to be a beginner's first server-side language. Files can contain text, HTML, CSS, JavaScript, and PHP code. PHP code are executed on the server, and the result is returned to the browser as plain HTML. PHP

files have extension ".php". PHP can generate dynamic page content. PHP can create, open, read, write, delete, and close files on the server. PHP can collect form data. PHP can send and receive cookies. PHP can add, delete, and modify data in your database. PHP can be used to control user-access. PHP can encrypt data. With PHP you are not limited to output HTML. You can output images, PDF files, and even flash movies. You can also output any text, such as XHTML and XML.

MYSQL

MySQL is the most popular Open-Source Relational SQL Database Management System. MySQL is one of the best RDBMS being used for developing various based software applications. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. This tutorial will give you a quick start to MySQL and make you comfortable with MySQL programming.

WAMP SERVER

WampServer is a Web development platform on Windows that allows you to create dynamic Web applications with Apache2, PHP, MySQL and MariaDB. WampServer automatically installs everything you need to intuitively develop Web applications. You will be able to tune your server without even touching its setting files.

Operating system

WINDOWS 11

Windows 11 is the latest major release of Microsoft's Windows NT operating system, released in October 2021. It is a free upgrade to its predecessor, Windows 10 (2015), and is available for any Windows 10 devices that meet the new Windows 11 system requirements. Windows 11 features major changes to the Windows shell influenced by the cancelled Windows 10X, including a redesigned Start menu, the replacement of its "live tiles" with a separate "Widgets" panel on the taskbar, the ability to create tiled sets of windows that can be minimized and restored from the taskbar as a group, and new gaming technologies inherited from Xbox Series X and Series S such as Auto HDR and Direct Storage on compatible hardware. Internet Explorer (IE) has been replaced by the Chromium-based Microsoft Edge as the default web browser, like its predecessor, Windows 10, and Microsoft Teams is integrated into the Windows shell. Microsoft also announced plans to allow more flexibility in software that can be distributed via the Microsoft Store and to support Android apps on Windows 11 (including a partnership with Amazon to make its app store available for the Function). Citing security considerations, the system requirements for Windows

11 were increased over Windows 10. Microsoft only officially supports the operating system on devices using an eighth-generation Intel Core CPU or newer (with some minor exceptions), a second-generation AMD Ryzen CPU or newer, or a Qualcomm Snapdragon 850 ARM system-on-chip or newer, with UEFI secure boot and Trusted Platform Module (TPM) 2.0 supported and enabled (although Microsoft may provide exceptions to the TPM 2.0 requirement for OEMs). While the OS can be installed on unsupported processors, Microsoft does not guarantee the availability of updates. Windows 11 removed support for 32-bit x86 CPUs and devices that use BIOS firmware. Windows 11 received a mixed reception at launch. Pre-release coverage of the operating system focused on its stricter hardware requirements, with discussions over whether they were legitimately intended to improve the security of Windows or as a ploy to upsell customers to newer devices and over the e-waste associated with the changes. Upon release, it was praised for its improved visual design, window management, and stronger focus on security, but was criticized for various modifications to aspects of its user interface that were seen as worse than its predecessor, as an attempt to dissuade users from switching to competing applications.

System Implementation

5.1 Introduction

It is a critical phase in the software development life cycle where the designed system is put into action. It involves translating the specifications and plans developed during earlier stages into a functioning and operational software system. This process encompasses the actual coding, configuration, testing, and deployment of the software. In this introduction, we will delve into the significance of system implementation and provide a general description of the key activities involved.

5.2 Coding

Sample codes

register_process.php

```
<?php
// Database connection
include 'dbconfig.php';
session_start();

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $conn->real_escape_string($_POST['name']);
    $username = $conn->real_escape_string($_POST['username']);
    $email = $conn->real_escape_string($_POST['email']);
    $password = $conn->real_escape_string($_POST['password']);

    // Check if username or email already exists
    $check_user = "SELECT * FROM users WHERE username='$username' OR
                  email='$email' LIMIT 1";
    $result = $conn->query($check_user);

    if ($result->num_rows > 0) {
        $_SESSION['error'] = "Username already exists!";
        header("Location: authen.php");
        exit();
    } else {
        $sql = "INSERT INTO users (name, username, email, password) VALUES
               ('$name', '$username', '$email', '$password')";

        if ($conn->query($sql) === TRUE) {
            // Set session variable for the username
            $_SESSION['username'] = $username;
        }
    }
}
```

```

        // Redirect to user details page
        header("Location: user_details.php");
        exit();
    } else {
        $_SESSION['error'] = "Error: " . $sql . "<br>" . $conn->error;
        header("Location: authen.php");
        exit();
    }
}

$conn->close();
?>

```

login_process.php

```

<?php
// Database connection
include 'dbconfig.php';
session_start();

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $conn->real_escape_string($_POST['username']);
    $password = $conn->real_escape_string($_POST['password']);

    // Fetch user
    $sql = "SELECT * FROM users WHERE username='$username' LIMIT 1";
    $result = $conn->query($sql);

    if ($result->num_rows == 1) {
        $user = $result->fetch_assoc();

        // Check if the user has an approved report
        $reportCheckSql = "SELECT approval FROM reports WHERE
                           reported_user = " . $user['id'] . " AND approval = TRUE LIMIT 1";
        $reportCheckResult = $conn->query($reportCheckSql);

        // If user has an approved report, show a suspension message
        if ($reportCheckResult->num_rows > 0) {
            $_SESSION['error'] = "Your account is suspended for 1 day due to a report.
                                  Please try again later.";
        }
    }
}

```

```

        header("Location: authen.php");
        exit();
    }

    // Verify password (ensure to use password hashing in real scenarios)
    if ($password == $user['password']) {
        // Start session and set session variables
        $_SESSION['username'] = $user['username'];
        $_SESSION['user_id'] = $user['id']; // Store user ID for future use
        // Redirect to dashboard or home page
        header("Location: dashboard.php");
        exit();
    } else {
        $_SESSION['error'] = "Invalid username or password!";
        header("Location: authen.php");
        exit();
    }
} else {
    $_SESSION['error'] = "No user found with that username!";
    header("Location: authen.php");
    exit();
}
}

$conn->close();
?>

```

post_upload.php

```

<?php
include 'dbconfig.php';
session_start();

if (!isset($_SESSION['username'])) {
    header('location:authen.php');
    exit();
}

$username = $_SESSION['username'];

// Retrieve user ID based on the username
$user_query = "SELECT id FROM users WHERE username='$username';";
$user_result = mysqli_query($conn, $user_query);
$user_info = mysqli_fetch_assoc($user_result);
$user_id = $user_info['id'] ?? null;

```

```

if (!$user_id) {
    $_SESSION['upload_message'] = 'User ID not found.';
    header('location: dashboard.php');
    exit();
}

// Check if form data is received and a file is uploaded
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_FILES['media'])) {
    error_log("File upload initiated."); // Log upload initiation

    $caption = mysqli_real_escape_string($conn, $_POST['caption']);
    $category = mysqli_real_escape_string($conn, $_POST['category']);

    // Allowed file types for upload
    $allowed_types = ['image/jpeg', 'image/png', 'image/gif', 'video/mp4', 'video/x-msvideo',
    'video/x-flv', 'video/ogg', 'video/webm'];
    $upload_dir = 'post_uploads/';

    // File information
    $file = $_FILES['media'];
    $file_name = $file['name'];
    $file_tmp_name = $file['tmp_name'];
    $file_type = $file['type'];
    $file_error = $file['error'];
    $file_size = $file['size'];

    // Log file details
    error_log("Uploaded file: $file_name, Type: $file_type, Size: $file_size bytes");

    // Validate file type
    if (!in_array($file_type, $allowed_types)) {
        error_log("Invalid file type: $file_type");
        $_SESSION['upload_message'] = 'Invalid file type. Only images (JPEG, PNG, GIF) and supported video formats are allowed.';
        header('location: dashboard.php');
        exit();
    }

    // Check for upload errors
    if ($file_error !== UPLOAD_ERR_OK) {
        switch ($file_error) {
            case UPLOAD_ERR_INI_SIZE:
            case UPLOAD_ERR_FORM_SIZE:
                $_SESSION['upload_message'] = 'File size exceeds the allowed limit.';
                break;
            case UPLOAD_ERR_PARTIAL:
                $_SESSION['upload_message'] = 'File was only partially uploaded.

```

```

        Please try again.';
        break;
    case UPLOAD_ERR_NO_FILE:
        $_SESSION['upload_message'] = 'No file was uploaded. Please choose a file.';
        break;
    case UPLOAD_ERR_NO_TMP_DIR:
        $_SESSION['upload_message'] = 'Temporary folder is missing.
        Please contact support.';
        break;
    case UPLOAD_ERR_CANT_WRITE:
        $_SESSION['upload_message'] = 'Failed to write file to disk.';
        break;
    case UPLOAD_ERR_EXTENSION:
        $_SESSION['upload_message'] = 'A PHP extension stopped the file upload.';
        break;
    default:
        $_SESSION['upload_message'] = 'An unknown error occurred.
        Error code: ' . $file_error;
        break;
    }
    header('location: dashboard.php');
    exit();
}

// Check if the file size exceeds server limits (100MB)
if ($file_size > 104857600) { // 100 MB limit
    $_SESSION['upload_message'] = 'File size exceeds the allowed limit of 100MB.';
    header('location: dashboard.php');
    exit();
}

// Generate a unique file name
$file_extension = pathinfo($file_name, PATHINFO_EXTENSION);
$new_file_name = uniqid('post_') . '.' . $file_extension;
$upload_path = $upload_dir . $new_file_name;

// Ensure the upload directory exists and has the correct permissions
if (!is_dir($upload_dir)) {
    if (!mkdir($upload_dir, 0777, true)) {
        $_SESSION['upload_message'] = 'Failed to create upload directory.';
        header('location: dashboard.php');
        exit();
    }
}

// Move file to upload directory
if (move_uploaded_file($file_tmp_name, $upload_path)) {

```

```

$sql = "INSERT INTO posts (user_id, post_path, caption, category, created_at)
        VALUES ('$user_id', '$upload_path', '$caption', '$category', NOW())";
if (mysqli_query($conn, $sql)) {
    $_SESSION['uploaded_post'] = [
        'path' => $upload_path,
        'caption' => $caption,
        'category' => $category,
    ];
    $_SESSION['upload_message'] = 'Post uploaded successfully!';
} else {
    $_SESSION['upload_message'] = 'Error uploading post to the database: ' .
        mysqli_error($conn);
}
} else {
    $_SESSION['upload_message'] = 'Error moving file to upload directory.
        Please check permissions.';
}
} else {
    $_SESSION['upload_message'] = 'No file uploaded.';
}

// Redirect back to the dashboard with the message
header('location: dashboard.php');
exit();
?>

```

follow_unfollow.php

```

<?php
include 'dbconfig.php';
session_start();

error_reporting(E_ALL);
ini_set('display_errors', 1);

if (!isset($_SESSION['username'])) {
    header('Location: authen.php');
    exit;
}

$followerUsername = $_SESSION['username'];
$followedUsername = $_POST['username'];
$action = $_POST['action'];

// Get the follower's ID
$followerQuery = "SELECT id FROM users WHERE username='$followerUsername'";
$followerResult = mysqli_query($conn, $followerQuery);

```

```

$followerRow = mysqli_fetch_assoc($followerResult);
$followerId = $followerRow['id'];

// Get the followed user's ID
$followedQuery = "SELECT id FROM users WHERE username='$followedUsername'";
$followedResult = mysqli_query($conn, $followedQuery);
$followedRow = mysqli_fetch_assoc($followedResult);
$followedId = $followedRow['id'];

if ($action === 'follow') {
    // Ensure both IDs are valid before inserting
    if ($followerId && $followedId) {
        $sql = "INSERT INTO follows (follower_id, followed_id) VALUES
            ('$followerId', '$followedId')";
        if (mysqli_query($conn, $sql)) {
            header('Location: ' . $_SERVER['HTTP_REFERER']);
            exit;
        } else {
            echo "Failed to follow user.";
        }
    } else {
        echo "Invalid follower or followed user.";
    }
} elseif ($action === 'unfollow') {
    if ($followerId && $followedId) {
        $sql = "DELETE FROM follows WHERE follower_id = '$followerId' AND
            followed_id = '$followedId'";
        if (mysqli_query($conn, $sql)) {
            header('Location: ' . $_SERVER['HTTP_REFERER']);
            exit;
        } else {
            echo "Failed to unfollow user.";
        }
    } else {
        echo "Invalid follower or followed user.";
    }
} else {
    echo "Invalid action.";
}

mysqli_close($conn);
?>

```

```

messaging.php
<?php
include 'dbconfig.php';
session_start();

if (!isset($_SESSION['username'])) {
    header('location:authen.php');
    exit();
}

$username = $_SESSION['username'];

if (isset($_POST['searchUser'])) {
    $searchQuery = mysqli_real_escape_string($conn, $_POST['searchUser']);
    $searchedUsers = [];

    if (!empty($searchQuery)) {
        $searchSQL = "
            SELECT u.*
            FROM users u
            WHERE (u.username LIKE '$searchQuery%' OR u.name LIKE '$searchQuery%')
            AND u.username != '$username';

        $searchResult = mysqli_query($conn, $searchSQL);

        while ($user = mysqli_fetch_assoc($searchResult)) {
            $profilePic = !empty($user['profile_picture']) ?
                $user['profile_picture'] : 'profile_picture/default.png';
            echo '
                <a href="chat_screen.php?user_id=' . $user['id'] . '"
                    style="text-decoration: none; display: flex; align-items: center;">
                    <div class="dropdown-item user-item d-flex align-items-center">
                        
                        <div class="user-info">
                            <div class="name">' . htmlspecialchars($user['name']) . '</div>
                            <div class="username text-muted"
                                style="font-size: 0.85em;">' . htmlspecialchars($user['username']) . '</div>
                        </div>
                    </div>
                </a>';
        }
    } else {
        echo '<div class="dropdown-item text-center">No users found</div>';
    }
    exit();
}

```

```

}

// Send message
if (isset($_POST['send-msg'])) {
    $message = $_POST['message'];
    $receiver_id = $_POST['receiver_id'];

    if (!empty($message) && !empty($receiver_id)) {
        // Get sender's user ID
        $senderSQL = "SELECT * FROM users WHERE username='\$username'";
        $senderResult = mysqli_query($conn, $senderSQL);
        $senderData = mysqli_fetch_assoc($senderResult);
        $sender_id = $senderData['id'];

        // Insert message into messages table
        $insertSQL = "INSERT INTO messages (sender_id, receiver_id, message, created_at)
                      VALUES ('$sender_id', '$receiver_id', '$message', NOW())";
        echo $insertSQL;
        $insertResult = mysqli_query($conn, $insertSQL);
        if ($insertResult) {
            $_SESSION['msg-success'] = 'Sent!';
            header("location:chat_screen.php?user_id=" . urlencode($receiver_id));
        } else {
            $_SESSION['msg-fail'] = 'Failed!';
            header("location:chat_screen.php?user_id=" . urlencode($receiver_id));
        }
        exit();
    }

// Fetch messages between the logged-in user and selected user
if (isset($_POST['fetchMessages'])) {
    $partnerUsername = mysqli_real_escape_string($conn, $_POST['partnerUsername']);

    // Get sender's user ID
    $senderSQL = "SELECT id FROM users WHERE username='\$username'";
    $senderResult = mysqli_query($conn, $senderSQL);
    $senderData = mysqli_fetch_assoc($senderResult);
    $sender_id = $senderData['id'];

    // Get receiver's user ID
    $receiverSQL = "SELECT id FROM users WHERE username='\$partnerUsername'";
    $receiverResult = mysqli_query($conn, $receiverSQL);
    $receiverData = mysqli_fetch_assoc($receiverResult);
    $receiver_id = $receiverData['id'];
}

```

```

// Fetch messages between the two users

$messagesSQL = "
    SELECT m.*, u.name, u.username
    FROM messages m
    JOIN users u ON u.id = IF(m.sender_id = '$sender_id', m.receiver_id, m.sender_id)
    WHERE (m.sender_id = '$sender_id' AND m.receiver_id = '$receiver_id')
        OR (m.sender_id = '$receiver_id' AND m.receiver_id = '$sender_id')
    ORDER BY m.created_at ASC";
$messagesResult = mysqli_query($conn, $messagesSQL);

while ($message = mysqli_fetch_assoc($messagesResult)) {
    echo '
<div class="message' . ($message['sender_id'] == $sender_id ? 'sent' : 'received') . '">
    <strong>' . htmlspecialchars($message['username']) . ':</strong>
    <strong>' . htmlspecialchars($message['message']) . '</strong>
    <span class="timestamp">' . date('Y-m-d H:i:s', strtotime($message['created_at'])) . '</span>
</div>';
}
exit();
}
?>

```

admin_dashboard.php

```

<?php
include 'dbconfig.php';
session_start();

error_reporting(E_ALL);
ini_set('display_errors', 1);

// Check if the user is logged in
if (!isset($_SESSION['id'])) {
    header('location: admin.php'); // Redirect to login if not logged in
    exit();
}

// Fetch all users from the database
$sql = "SELECT * FROM users";
$result = $conn->query($sql);

$sql_user_count = "SELECT COUNT(*) AS user_count FROM users";
$result_user_count = $conn->query($sql_user_count);

```

```

$user_count = 0; // Initialize in case no users are found

if ($result_user_count && $result_user_count->num_rows > 0) {
    $row = $result_user_count->fetch_assoc();
    $user_count = $row['user_count'];
}

// Fetch total number of complaints
$sql_complaints_count = "SELECT COUNT(*) AS complaints_count FROM complaints";
$result_complaints_count = $conn->query($sql_complaints_count);
$complaints_count = 0;
if ($result_complaints_count && $result_complaints_count->num_rows > 0) {
    $row = $result_complaints_count->fetch_assoc();
    $complaints_count = $row['complaints_count'];
}

// Fetch total number of reports
$sql_reports_count = "SELECT COUNT(*) AS reports_count FROM reports";
$result_reports_count = $conn->query($sql_reports_count);
$reports_count = 0;
if ($result_reports_count && $result_reports_count->num_rows > 0) {
    $row = $result_reports_count->fetch_assoc();
    $reports_count = $row['reports_count'];
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PIXTREAM - Administrator</title>
    <link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="library-files/fontawesome/css/all.min.css">
    <link rel="stylesheet" href="assets/css/me.css">
    <link rel="icon" type="image/x-icon" href="assets/img/LOGO_tab.svg" />

    <style>
        /* Your CSS styles here */
    </style>
</head>
<body>

    <!-- Toggle Button -->

```

```

<button class="btn mybtn-outline toggle-btn" id="sidebarToggle"><i class="fa-solid fa-bars"></i></button>

<div class="container-fluid d-flex admin-2divs">
    <!-- Sidebar -->
    <div class="sidebar" id="sidebar">
        <div class="text-center mb-4">
            
        </div>
        <ul class="nav flex-column">
            <li class="nav-item">
                <a class="nav-link active-now" href="#">
                    <b>Dashboard</b>
                </a>
            </li>
            <li class="nav-item">
                <a class="nav-link link1" href="admin_users.php">
                    <b>Manage Users</b>
                </a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="admin_complaints.php">
                    <b>View Complaints</b>
                </a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="admin_reports.php">
                    <b>View Reports</b>
                </a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="logout.php">
                    <b>Log Out</b>
                </a>
            </li>
        </ul>
    </div>
    <!-- Sidebar -->

    <!-- Main Content -->
    <div class="content" id="mainContent">
        <!-- Dashboard Content -->
        <div class="container admin-dash">
            <p class="mt-5 fs-5"><b>Welcome to the Pixtream Admin Dashboard.
                Manage users, view complaints, and oversee reports here.</b></p>
        </div>
    </div>
</div>

```

```

<!-- Example Cards for Quick Stats -->
<div class="row mt-5">
    <!-- Total Users Card -->
    <div class="col-lg-4 col-md-6 col-12 mb-4">
        <a class="link1" href="admin_users.php">
            <div class="card text-center admin-card">
                <div class="card-body">
                    <h5 class="card-title">Total Users: <?php echo $user_count; ?></h5>
                    <p class="card-text mt-4">Pixtream has a total of
                        <strong></strong> users registered.
                    Stay updated on the growth of your platform as new users sign
                    up.</p>
                </div>
            </div>
        </a>
    </div>

    <!-- Complaints Card -->
    <div class="col-lg-4 col-md-6 col-12 mb-4">
        <a class="link1" href="admin_complaints.php">
            <div class="card text-center admin-card">
                <div class="card-body">
                    <h5 class="card-title">Complaints: <?php echo $complaints_count;
                    ?></h5>
                    <p class="card-text mt-4">There are currently complaints pending
                    review. Please address these issues promptly to maintain user
                    satisfaction.</p>
                </div>
            </div>
        </a>
    </div>

    <!-- Reports Card -->
    <div class="col-lg-4 col-md-6 col-12 mb-4">
        <a class="link1" href="admin_reports.php">
            <div class="card text-center admin-card">
                <div class="card-body">
                    <h5 class="card-title">Reports: <?php echo $reports_count; ?></h5>
                    <p class="card-text mt-4">You have new reports from users
                    concerning inappropriate content or behavior. Ensure to
                    investigate and take necessary actions.</p>
                </div>
            </div>
        </a>
    </div>
</div>

```

```
<!-- Add more content here -->
</div>
<!-- End of Dashboard Content -->
</div>
<!-- Main Content -->
</div>

<!-- JS files -->

<script>
    document.getElementById('sidebarToggle').addEventListener('click', function() {
        const sidebar = document.getElementById('sidebar');
        const mainContent = document.getElementById('mainContent');
        sidebar.classList.toggle('collapsed');
        mainContent.classList.toggle('collapsed');
    });
</script>

<script src="node_modules/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

System Testing

6.1 Introduction

System testing includes both verification and validation and serves as a major quality control measure during software development. Testing a large system is complex and is divided into smaller activities to ensure efficiency. In the Pixtream project, all modules were tested individually to identify coding errors and ensure proper functionality. Special test data were used to examine the output, ensuring the software behaves as per its specifications and user expectations. This constitutes unit testing.

Dynamic testing approaches can detect the presence of errors but may not specify their exact nature. Reference documents, such as requirement specifications, were used to guide testing efforts. Testing primarily focuses on the external behavior of the system rather than the internal logic. In Pixtream, the system was divided into small modules, tested separately, integrated, and then tested with live data to ensure optimal functionality. The types of testing employed include:

- Unit testing
- Integration testing
- Validation testing
- Alpha testing
- Beta testing

6.2 Unit testing

Unit testing is the first level of testing, focusing on the smallest unit of software design. Each module in the Pixtream project was tested during the coding phase. For example, the login and registration modules were tested for proper data handling. Module interfaces were verified to ensure smooth information flow, and data consistency checks were performed to maintain integrity throughout algorithm execution. All independent code paths were examined to ensure every statement was executed at least once.

6.3 Integration Testing

Integration testing involves systematically constructing the program structure while testing module interactions to identify interfacing errors. The objective is to combine unit-tested modules and verify their collective performance as dictated by the design. For example, in Pixtream, user registration details such as username and password

were verified to be stored correctly in the database. Integration testing was conducted across both the admin and user modules to ensure seamless interaction between components.

6.4 System testing

System testing is a comprehensive phase where the entire software system is tested as a unified whole to confirm adherence to specified requirements. This phase follows integration testing and precedes acceptance testing. The primary aim is to validate both functional and non-functional aspects of the system in a fully integrated environment.

In Pixtream, system testing included examining the platform under various scenarios to identify and resolve issues. This ensured that the system meets the performance, security, and usability standards. Successful system testing provides confidence in the system's robustness, paving the way for a smooth acceptance testing phase and reliable deployment.

System maintenance

7.1 Introduction

Even though the changeover of the system is fully correct and complete, it is not the end of the matter. The system should be given proper security and maintenance in order to keep them efficient and up-to-date.

7.2 Maintenance

This software can be modified as need occurs. Maintenance includes all the activities after installation of the software that is performed to keep the system operational. The process of maintaining involves:

- Understanding the existing software
- Understand the effect of change
- Test for satisfaction

Maintenance can be done to this project by simply adding the new requirements that are the form of database the system can be modified. The maintenance process also helps to remove an error that raise in the system even after testing process. The complexity of the maintenance task coupled with the neglect of the maintenance concerns during development makes maintenance the most costly activity in life of the system project. The system security is for protection against fraud and disaster. To avoid unauthorized access, password protection is highly recommended while running this new system. The password has to be maintained directly and files have to be kept very confidential.

Future enhancement and scope of further development

8.1 Introduction

The Pixtream system is designed with scalability and adaptability in mind, ensuring ease of integration for future enhancements. The system accommodates necessary updates and modifications without extensive rework. It is built to anticipate future trends and user demands, ensuring its relevance and usability over time.

8.2 Merits of the system

- The modular design facilitates easy integration of additional features.
- Built using PHP and MySQL, the system leverages modern web technologies, making upgrades straightforward.
- The current system includes core functionalities such as user interactions, post management, and administrative controls, providing a strong foundation for expansion.

8.3 Limitations of the system

- The current system is focused on basic social media functionalities and lacks advanced features such as AI-driven recommendations or automated moderation.
- Integration with third-party services, such as payment gateways or external APIs, is not yet implemented.
- The system does not currently support live streaming or real-time notifications, which could enhance user engagement.

8.4 Future Enhancement of the system

To ensure continued growth and relevance, several enhancements can be considered for the Pixtream system:

- **Advanced Search and Filtering:** Allow users to filter posts and users based on specific criteria, such as location, category, or interests.
- **Real-Time Features:** Introduce live streaming, real-time chat, and push notifications for enhanced interactivity.
- **Gamification:** Add badges, rewards, or achievement systems to encourage user participation.
- **Integration with AI:** Implement AI-driven user suggestions, content moderation, and personalized recommendations.
- **Monetization Features:** Include options for premium subscriptions, advertisements, or in-app purchases.
- **Mobile App Development:** Extend the platform's accessibility with dedicated mobile applications for Android and iOS.
- **Accessibility Features:** Enhance the system's usability for differently-abled users by integrating accessibility standards.

These enhancements will not only increase user satisfaction but also position Pixtream as a competitive and feature-rich social media platform.

Conclusion

CONCLUSION

In today's digital age, computers are integral to all facets of life, playing a critical role in enhancing productivity, accuracy, and efficiency across various sectors. From business to education, social media platforms like Pixtream make managing personal connections, content sharing, and communication seamless and effortless.

The Pixtream project was completed successfully within the set timeline. All modules were tested individually and then integrated into the final system. After thorough testing with real user data, the system functioned as expected and met all the defined objectives.

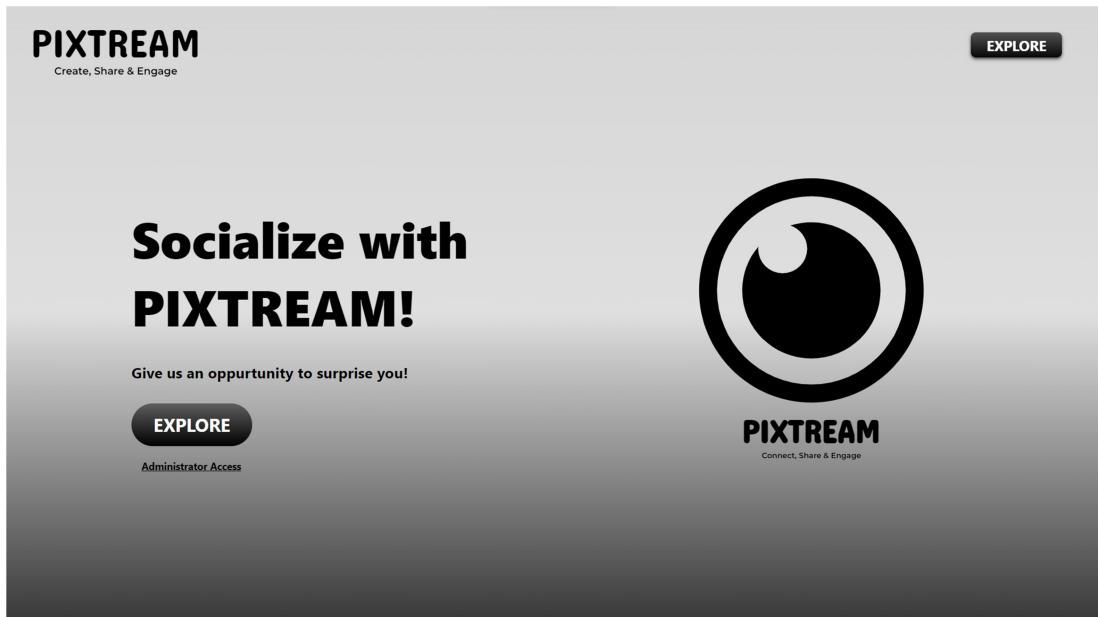
Pixtream was developed to address the needs of modern social media users, providing a reliable platform for interaction, content sharing, and user management. The system's modular design ensures that future improvements, such as enhanced AI features, real-time interactions, and mobile app development, can be easily integrated. As Pixtream continues to evolve, it has the potential to transform into a more robust and feature-rich platform to cater to a broader audience.

Bibliography

- Software Engineering (Third Edition), K K Aggarwal, Yogesh Singh, New Age International Publications.
- Software Engineering (7th Edition), Ian Sommerville, Pearson Education
- An Integrated approach to Software Engineering (Second Edition), Pankaj Jalote,
- www.chatgpt.com for web designing

Appendix

Screenshots



The image shows a post detail page from the PIXTREAM platform. At the top left is the PIXTREAM logo. At the top right are links for "Home", "People", "Chat", and "Sain Sabraj". The main content area features a post by user "Aiden Miller" (Dec 10 2024) showing a photograph of a small wooden cabin in a green, mountainous landscape. The post has the categories "Nature" and "Photography". Below the photo are three interaction buttons: "Liked" (black), "Comment" (gray), and "Share" (gray). The background of the page is white.

The image displays two screenshots of the PIXTREAM application interface.

Screenshot 1: New Post Screen

This screenshot shows the 'New Post' dialog box over a blurred background image of a forest path. The dialog box contains:

- A 'Choose file' button with a camera icon.
- An image preview of a colorful autumn forest path.
- A 'Caption' field containing "Nature ❤️".
- A 'Category' dropdown menu set to "Photography".
- A 'Upload' button at the bottom.

The background shows a user profile for "Aiden Miller" with a placeholder profile picture, a post count of 1, and follower/following counts of 3 each. Below the post are buttons for "Liked", "Comment", and "Share".

Screenshot 2: User Profile Screen

This screenshot shows the profile page for "Oliver King". The profile includes:

- A placeholder profile picture.
- The name "Oliver King".
- Statistics: Posts (1), Followers (3), Following (3).
- Buttons: "Follow" (black) and "Message".
- A bio: "Social media expert and digital marketer".
- A location: "New York, USA".
- Interest categories: Technology, Lifestyle, Short Films, Relatable Memes, Dark Humor.
- A "Posts - Oliver King" section with a thumbnail image of a city skyline.
- A context menu for the profile: "About this Account", "Report", and "Block User".

The URL in the browser address bar is: `http://localhost/dashboard/PIXTREAM-master/user_profile.php?username=oliver_king`.

The image shows two screenshots of the PIXTREAM application interface.

Top Screenshot: A post by user "Aiden Miller" (Dec 10 2024) featuring a landscape photo. An "Add Comment" overlay is open, showing a comment from "jack_brown" (Dec 11 2024) with a heart and a thumbs-up emoji. Below the photo are categories: "Nature" and "Photography". At the bottom are "Liked", "Comment", and "Share" buttons.

Bottom Screenshot: A search results page titled "Suggested For You" with a search bar and a "Clear" button. It lists "Harper Allen" (harper_allen) with a "Follow" button. Below that is a section titled "People Near You" listing users: "Lily Roberts" (lily_roberts), "Victoria Harris" (victoria_harris), "Oliver King" (oliver_king), "Aiden Walker" (aiden_walker), "Noah Carter" (noah_carter), and "Alexander Wright" (alexander_wright). Each user entry includes a "Follow" or "Following" button.

Profile View:

Profile Details:

- User: David Adams
- Followers: 6
- Posts: 1
- Last Seen: 4:44 AM

Message History:

- 1. Hello... How are you? 4:44 AM
- 2. Doing good! 4:44 AM

Edit Profile View:

Categories and Sub-options:

- ART & DESIGN:** Digital Art, Painting, Sketching
- ARTICLES:** Health & Wellness, Lifestyle, Technology
- FOOD & COOKING:** Culinary Tips, Food Reviews, Recipes
- MEMES:** Dark Humor, Relatable Memes, Wholesome Memes
- MUSIC:** Cover Songs, Instrumental, Original Compositions
- OTHERS:** Miscellaneous, Random Thoughts
- PHOTOGRAPHY:** Abstract, Nature, Urban Photography
- PHOTOS:** Landscapes, Portraits, Street Photography, Wildlife
- STORIES:** Historical Narratives, Personal Journeys, Travel Stories
- TRAVEL & ADVENTURE:** Backpacking, Nature Exploration, Road Trips
- VIDEOS:** Documentaries, Short Films, Tutorials, Vlogs

Profile Settings:

- STATE: New York, USA
- ADD A BIO: (Empty input field)
- Buttons: Skip, Submit

Welcome to the Pixtream Admin Dashboard. Manage users, view complaints, and oversee reports here.

Total Users: 94

Pixtream has a total of users registered. Stay updated on the growth of your platform as new users sign up.

Complaints: 2

There are currently complaints pending review. Please address these issues promptly to maintain user satisfaction.

Reports: 0

You have new reports from users concerning inappropriate content or behavior. Ensure to investigate and take necessary actions.

- Dashboard
- Manage Users
- View Complaints
- View Reports
- Log Out

Manage Users

Name	Username	Email	Location	Joined Date	Actions
Ella King	ella_king	ella.king@example.com	San Francisco, USA	December 2024	<button>View</button>
Lily Roberts	lily_roberts	lily.roberts@example.com	New York, USA	December 2024	<button>View</button>
Ethan Walker	ethan_walker	ethan.walker@example.com	Toronto, Canada	December 2024	<button>View</button>
Chloe Adams	chloe_adams	chloe.adams@example.com	London, UK	December 2024	<button>View</button>
Lucas Harris	lucas_harris	lucas.harris@example.com	Los Angeles, USA	December 2024	<button>View</button>
Zoe Lee	zoe_lee	zoe.lee@example.com	Paris, France	December 2024	<button>View</button>
Benjamin Clark	benjamin_clark	benjamin.clark@example.com	Berlin, Germany	December 2024	<button>View</button>
Maya Wilson	maya_wilson	maya.wilson@example.com	San Francisco, USA	December 2024	<button>View</button>
Ryan Harris	ryan_harris	ryan.harris@example.com	Toronto, Canada	December 2024	<button>View</button>
Olivia Miller	olivia_miller	olivia.miller@example.com	Vancouver, Canada	December 2024	<button>View</button>
David Adams	david_adams	david.adams@example.com	Miami, USA	December 2024	<button>View</button>
Harper Scott	harper_scott	harper.scott@example.com	Los Angeles, USA	December 2024	<button>View</button>
Sophia Evans	sophia_evans	sophia.evans@example.com	Berlin, Germany	December 2024	<button>View</button>

The screenshot shows a web-based application interface for managing complaints. On the left, there is a sidebar with a navigation menu:

- Dashboard
- Manage Users
- View Complaints** (highlighted in grey)
- View Reports
- Log Out

The main content area is titled "Manage Complaints". It displays two complaint entries in a table format:

Complaint by:	Date Filed
Sain Saburaj	10 December 2024
Sain Saburaj	10 December 2024

Each row contains the complainant's name and the date the complaint was filed. The first row is labeled "Technical Issue" and the second row is labeled "UI problem".