

# **Design and Implementation of a Desktop-Based Social Networking System**

**TeamCgram**

**Course Project Report**

**Submitted By:**

**Team C**

Hikmatyar (Team Lead)  
Asma Saqib  
Almir Ayham  
Sidra Noor  
Muhammad Ayaz  
Muhammad Saleh  
Muhammad Mansoor  
Meher Asif  
Farhan Hussain  
Daniyal Hikmat

**Department of Software Engineering  
UIInstitute of Management Sciences (IMSciences), Peshawar**

**2026**

# Project Team

Role	Name
Team Lead	Hikmatyar
Member	Asma Saqib
Member	Almir Ayham
Member	Sidra Noor
Member	Muhannad Ayaz
Member	Muhammad Saleh
Member	Muhammad Mansoor
Member	Meher Asif
Member	Farhan Hussain
Member	Daniyal Hikmat

# Contents

<b>Project Team</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
<b>2 System Architecture</b>	<b>4</b>
<b>3 Data Structures and Algorithms</b>	<b>5</b>
3.1 User Management Using BST . . . . .	5
3.2 Post and Comment System . . . . .	5
3.3 Notification System . . . . .	5
<b>4 Follow System</b>	<b>6</b>
<b>5 Graphical User Interface</b>	<b>7</b>
5.1 Screen Architecture . . . . .	7
5.2 Implemented Screens . . . . .	7
<b>6 Testing and Validation</b>	<b>8</b>
6.1 Testing Strategy . . . . .	8
6.2 Integration Testing Summary . . . . .	8
<b>7 Performance Analysis</b>	<b>9</b>
<b>8 Limitations</b>	<b>10</b>
<b>9 Future Enhancements</b>	<b>11</b>
<b>10 Conclusion</b>	<b>12</b>

# Chapter 1

## Introduction

Social networking systems require efficient data handling, scalable design, and responsive user interfaces. This project presents a desktop-based social networking application developed using **C++** and **SFML**. The system focuses on essential features including user management, post interaction, notifications, and graphical user interfaces.

The primary objective is to demonstrate the application of data structures, object-oriented programming, and event-driven GUI design within an academic scope.

# Chapter 2

## System Architecture

The application follows a modular architecture consisting of the following components:

- User Management Module
- Post and Comment Module
- Follow and Notification System
- Graphical User Interface (SFML)

Each module is independently designed to improve maintainability and scalability.

# Chapter 3

## Data Structures and Algorithms

### 3.1 User Management Using BST

Users are stored using a Binary Search Tree (BST) indexed by usernames, allowing efficient insertion and lookup.

---

#### Algorithm 1 BST User Search

```
SearchUsernode, username node = null null node.username = username node.user
username < node.username SearchUsernode.left, username SearchUsernode.right,
username
```

---

### 3.2 Post and Comment System

Posts are stored using linked lists, while comments are implemented using singly linked lists to ensure fast insertion and minimal memory overhead.

Listing 3.1: Comment Structure

```
1 struct Comment {
2     int userID;
3     std::string text;
4     time_t timestamp;
5     Comment* next;
6 }
```

### 3.3 Notification System

Notifications are stored in a linked list with recent notifications inserted at the head. Unread notifications are tracked using a boolean flag.

Listing 3.2: Notification Types

```
1 enum class NotificationType {
2     FOLLOW,
3     LIKE,
4     COMMENT
5 }
```

# Chapter 4

## Follow System

Each user maintains two lists:

- Followers
- Following

Duplicate follow relationships are prevented through validation checks.

# Chapter 5

## Graphical User Interface

The graphical interface is implemented using **SFML**, providing real-time rendering and event-driven interaction.

### 5.1 Screen Architecture

All screens inherit from a common abstract base class to ensure consistency.

Listing 5.1: Screen Base Class

```
1 class Screen {
2 public:
3     virtual void handleEvent(const sf::Event&) = 0;
4     virtual void update(float) = 0;
5     virtual void render() = 0;
6 };
```

### 5.2 Implemented Screens

- Login and Registration Screen
- Feed Screen with Scrolling Support
- Notification Screen

# Chapter 6

## Testing and Validation

### 6.1 Testing Strategy

The system was tested using unit testing, integration testing, and user acceptance testing to ensure correctness and reliability.

### 6.2 Integration Testing Summary

Table 6.1: Integration Test Results

Scenario	Status
User Registration	Passed
Post Creation	Passed
Like and Comment	Passed
Notification Delivery	Passed

# Chapter 7

## Performance Analysis

Table 7.1: Performance Benchmarks

Operation	100 Users	500 Users	1000 Users	Complexity
User Search	0.002ms	0.005ms	0.008ms	$O(\log n)$
Post Creation	0.001ms	0.001ms	0.001ms	$O(1)$
Feed Load	15ms	78ms	165ms	$O(n)$
Add Comment	0.006ms	0.009ms	0.012ms	$O(n)$

# **Chapter 8**

## **Limitations**

- File-based data persistence
- No password encryption
- Limited comment nesting
- No real-time updates

# Chapter 9

## Future Enhancements

- Database integration
- Secure authentication mechanisms
- Media sharing support
- Real-time notifications
- Mobile-friendly interface

# **Chapter 10**

## **Conclusion**

This project successfully demonstrates the design and implementation of a desktop-based social networking system using C++ and SFML. The system applies efficient data structures, modular design principles, and thorough testing, fulfilling all academic requirements.

# Bibliography

- [1] SFML Documentation.  
<https://www.sfml-dev.org>
- [2] Cormen et al., *Introduction to Algorithms*, MIT Press.
- [3] Bjarne Stroustrup, *The C++ Programming Language*.