

PART B

Summary Sheet

Title of the problem: Implementation of a Mini Twitter-like Platform Using Data Structures

Problem Statement: Design and implement a simplified Twitter-like platform where users can post tweets, follow/unfollow other users, view a personalized news feed, and search for users by username. The solution should efficiently manage user data, tweets, and relationships using appropriate data structures to ensure scalability and performance.

Methodology :

Step 1: Initialize a **Twitter** object with a predefined number of users and a binary search tree (BST) for managing user data by username.

Step 2: Allow users to post tweets, storing them in linked lists associated with each user.

Step 3: Implement a follow/unfollow mechanism using an adjacency matrix to represent relationships between users.

Step 4: Generate a personalized news feed by combining the tweets of the user and their followees, leveraging a stack for temporary storage.

Step 5: Search for users using the BST for efficient retrieval and display their tweet history.

Step 6: Interface for the platform, leveraging a menu-driven system for console interaction. using GLADE

Step 7: Implement memory cleanup to free dynamically allocated resources and maintain efficiency.

Data Structures used in the solution and the purpose (150 Words):

1. Binary Search Tree (BST):
 - Purpose: Efficiently manage and retrieve user data by username.
 - Functionality: Supports fast insertion and search operations ($O(\log n)$ on average).
2. Linked List:
 - Purpose: Store tweets for each user dynamically.
 - Functionality: Allows quick addition of new tweets and sequential traversal for display.
3. Adjacency Matrix (Static Array):
 - Purpose: Represent and manage follow/unfollow relationships between users.

- Functionality: Enables constant-time updates and lookups for relationships.
- 4. Stack:
 - Purpose: Temporarily store tweets while generating the news feed.
 - Functionality: Allows reverse-order retrieval of the most recent tweets.
- 5. Static Array (User Data Storage):
 - Purpose: Store and directly access user details using indices.
 - Functionality: Simplifies mapping user IDs to their associated data structures.

Interface Design:

1. Console Interface: A menu-driven system to enable user interaction with options for posting tweets, following/unfollowing users, viewing feeds, and searching for users.
2. Glade for the actual interface

The image displays four Glade window mockups for a social media application, arranged in a 2x2 grid. Each window has a title bar and a main content area with a light yellow background.

- window_follow** (top-left): Contains two input fields labeled "Follower ID" and "Followee ID". Below them are two buttons: "Follow" and "Back".
- window_post_tweet** (top-right): Contains three input fields labeled "User ID", "User Name", and "Tweet ID". Below them is a text area labeled "Tweet Message". At the bottom are two buttons: "Post Tweet" and "Back".
- window_feed** (bottom-left): Contains one input field labeled "User ID". Below it are two buttons: "Get Feed" and "Back".
- window_welcome** (bottom-right): Contains four buttons stacked vertically: "Post Tweet", "Follow User", "Get Feed", and "Exit".