Design Document

Part A: General Overview of the System

Our system is designed to manage user accounts, tweet management, and search functionalities. It provides a streamlined user experience by efficiently handling social media data using **Python** and **SQLite**.

System Overview

The application integrates multiple components that work together to:

- Authenticate users securely.
- Enable searching for tweets and users.
- Facilitate the composition and posting of new tweets.

A high-level data flow diagram illustrates how the components interact with the database, ensuring smooth information exchange between user inputs, processing logic, and the database.

User Guide

1. **System Setup**
   - Ensure **Python 3.x** and **SQLite** are installed on your system.
   - Place the provided SQLite database (`prj-sample.db`) in the working directory.
2. **User Login**
   - Upon launching, the system prompts for a **username** and **password**.
   - If incorrect credentials are entered, an error message is displayed.
   - Successful login grants access to the main menu.
3. **Functionalities**
   - **Search Tweets:** Enter keywords or hashtags to display matching tweets.
   - **Search Users:** Input a username or partial name to find user profiles.
   - **Compose Tweets:** Create a tweet within the character limit, which is then added to the database.
4. **Navigation**
   - The menu-driven interface allows easy movement between functionalities.
   - Error handling ensures user-friendly guidance for invalid inputs.

Part B: Detailed Design of the Software

The design is modular, focusing on the major functions of the application. Below is an overview of the primary components, their responsibilities, and their interfaces:

1. **Login and Logout Component**
    - **Developer:** Alex
    - **Responsibility:** Manage user authentication.
    - **Interface:** Accepts username/password, validates against the database, and handles errors for invalid credentials.
2. **Search Tweets Component**
    - **Developer:** Timi
    - **Responsibility:** Query the database for tweets matching keywords/hashtags and display results.
    - **Interface:** Accepts search input, retrieves matching data, and presents it to the user.
3. **Search Users Component**
    - **Developer:** Nick
    - **Responsibility:** Search for users by name and display their details.
    - **Interface:** Accepts input criteria and queries the database for matching users.
4. **Compose Tweet Component**
    - **Developer:** Max
    - **Responsibility:** Create and post tweets while ensuring constraints are met.
    - **Interface:** Accepts user input, validates the tweet, and inserts it into the database.
5. View Followers Component
    - Developer: Alex
    - Responsibility: Allow a user to view their followers
    - Interface: Queries database for followers and allows user to follow back

Structure and Relationships

- All components share access to the SQLite database through a common database handler.
- Each component interacts with the database via SQL queries embedded in Python functions.
- Functions follow standard conventions for reusability and maintainability.

Part C: Testing Strategy

General Approach

Our testing strategy ensures that individual components and the fully integrated system meet functional requirements.

1. **Individual Component Testing**
    - Conducted by each developer for their respective component.
    - **Tools Used:**

        - Print statements for debugging.
        - Try-except blocks for error handling.
    - **Scenarios Covered:**
        - Valid and invalid user inputs.
        - Database edge cases (e.g., non-existent users, empty searches).
2. **Integration Testing**
    - Conducted collaboratively after individual testing.
    - **Scenarios Covered:**
        - Smooth data flow between components.
        - Correct handling of user sessions and menu transitions.
3. **Test Coverage**
    - Each functionality tested under various conditions, including edge cases.
    - Results showed minimal bugs, mostly related to user input formatting, which were resolved during testing.

Part D: Group Work Breakdown Strategy

Team Composition

- **Alex:** Login and Logout Component, and followers component
- **Timi:** Search Tweets Component.
- **Nick:** Search Users Component.
- **Max:** Compose Tweet Component.

Work Allocation

- **Rationale:** Tasks assigned based on expertise with similar systems or functionalities.
- **Time Estimates:**
    - Each member spent approximately **4 hours** developing their component.
    - The group collectively spent an additional **5 hours** on integration and debugging.

Coordination Strategy

- Set up meetings through Google Meet to track progress and resolve issues.
- Shared a GitHub repository to manage code contributions and version control.
- Discussed and documented any deviations from the project specification or additional coding decisions.

Notable Decisions

- Added functionality for better error handling in user input.
- Modified the search interface to include partial name matching for user searches.