

NORTHEASTERN UNIVERSITY
CS5200 - DATABASE MANAGEMENT SYSTEMS

SOCIAL MEDIA MANAGEMENT SYSTEM

GROUP NAME: ShashikumarNPatilP

Submission by:
Neha Shashikumar 002774563
Preksha Patil 002646279

READ ME

Table of Contents

- Project Name
- Description
- Data Functionality
- Installation
- Dependencies
- Configuration
- Run the application

PROJECT NAME - SOCIAL MEDIA MANAGEMENT SYSTEM

DESCRIPTION

Our project aims to develop a comprehensive Social Media Database Management System that serves as the backbone for a social media platform. This system will provide users with the ability to create profiles, share various forms of content (text, photos, videos), engage with other users through likes and comments, and manage their interactions in a secure and efficient manner.

DATA FUNCTIONALITY

In this system, each user is distinguished by a set of distinctive attributes. A user has to create an account in the system to start using the application. The user authentication has its own unique identifier, which includes an email address and a password for logging in. It also records the time of each login. The user profile includes a unique user ID, username, profile photo, personal biography, registration date, account status, and date of birth. Users have the capability to follow other users too. Users are able to create and share multiple posts, each of which is defined by a unique post ID. These posts come with a caption, location, creation date, and last updated date. A post can consist of either photos, videos, or both. For photos and videos there are URL, size specifications, that point to the image, and creation dates. Posts can be interacted with in the form of likes and comments. Each like and comment also has its own unique identifier, and they are linked to the corresponding post. Users are allowed to leave multiple comments on posts, but they are only permitted to like a particular post once.

INSTALLATION

To run this project, you need to have the following installed on your machine:

1. MySQL Database: You must install the MySQL database and MySQL workbench on your laptop. Install MySQL on your machine. You can download it from [MySQL Downloads] (<https://dev.mysql.com/downloads/workbench/>).

2. MySQL Connector for Java: Download the MySQL Connector for Java from [MySQL Connector/J] and include it in your project (<https://dev.mysql.com/downloads/connector/j/>).
3. Java Development Kit (JDK): Install JDK to compile and run the Java code. You can download it from [Oracle JDK](<https://www.oracle.com/java/technologies/javase-downloads.html>).
4. Install an IDE of your choice. We used IntelliJ.
5. Download the MySQL JDBC Driver (mysql-connector-java.jar) from the MySQL website. <https://dev.mysql.com/downloads/connector/j/5.1.html>
6. Connect your IDE to the MySQL using the following steps
 - Open IntelliJ and go to File > Project Structure.
 - In the Project Structure window, go to the Modules tab and click on the Dependencies tab.
 - Click on the green “+” button and select “JARs or directories”.
 - Navigate to the location where you have saved the MySQL Connector JAR file and select it.
 - Click on the “OK” button to add the MySQL Connector to your project.
7. Use the database dump by the name socialmediadump2 to get all the details into your system.

DEPENDENCIES

- MySQL Connector for Java
- Java Development Kit (JDK)

CONFIGURATION

Ensure that your MySQL server is running and accessible.

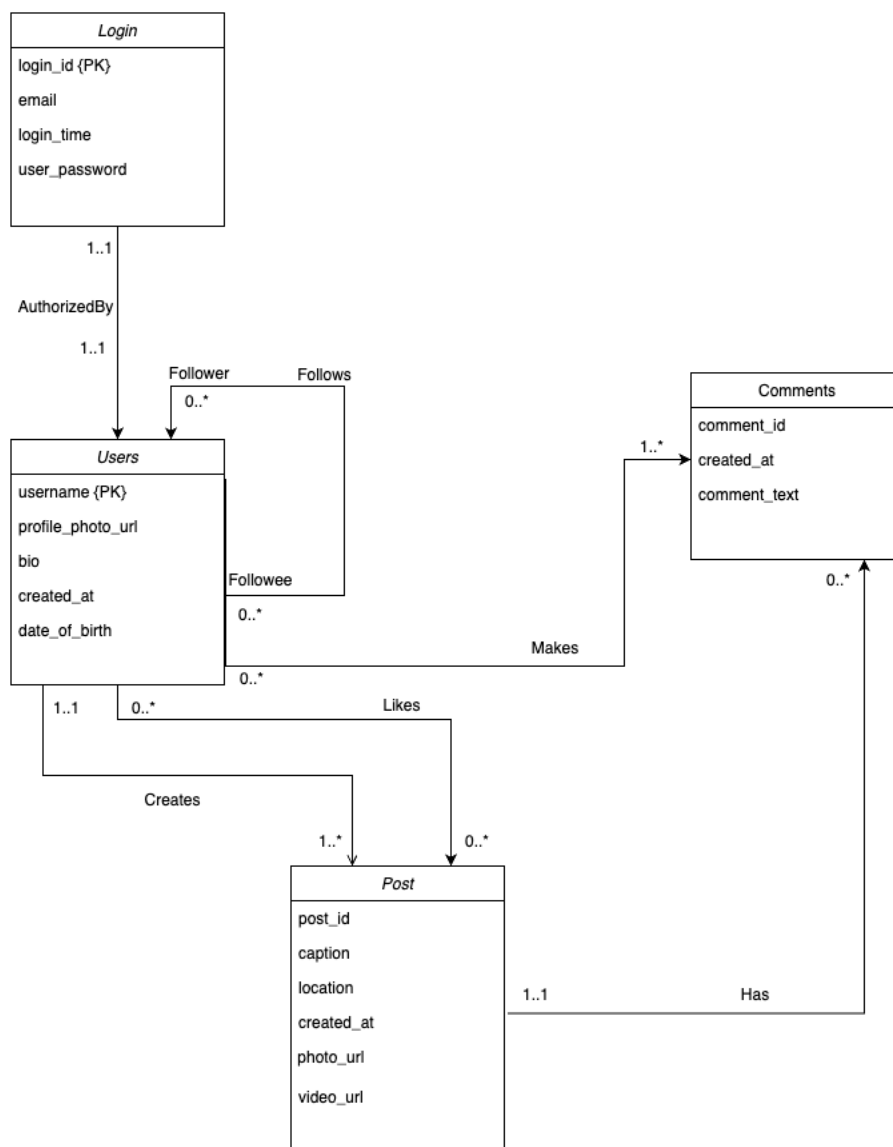
A db.properties file has been created and present in the root folder. Please update the file with the following contents:

```
db.username=[Your MySQL username]
db.password=[Your MySQL password]
db.serverName=[Database server name, usually localhost]
db.portNumber=[MySQL port, default is 3306]
db.dbName=socialmedia
```

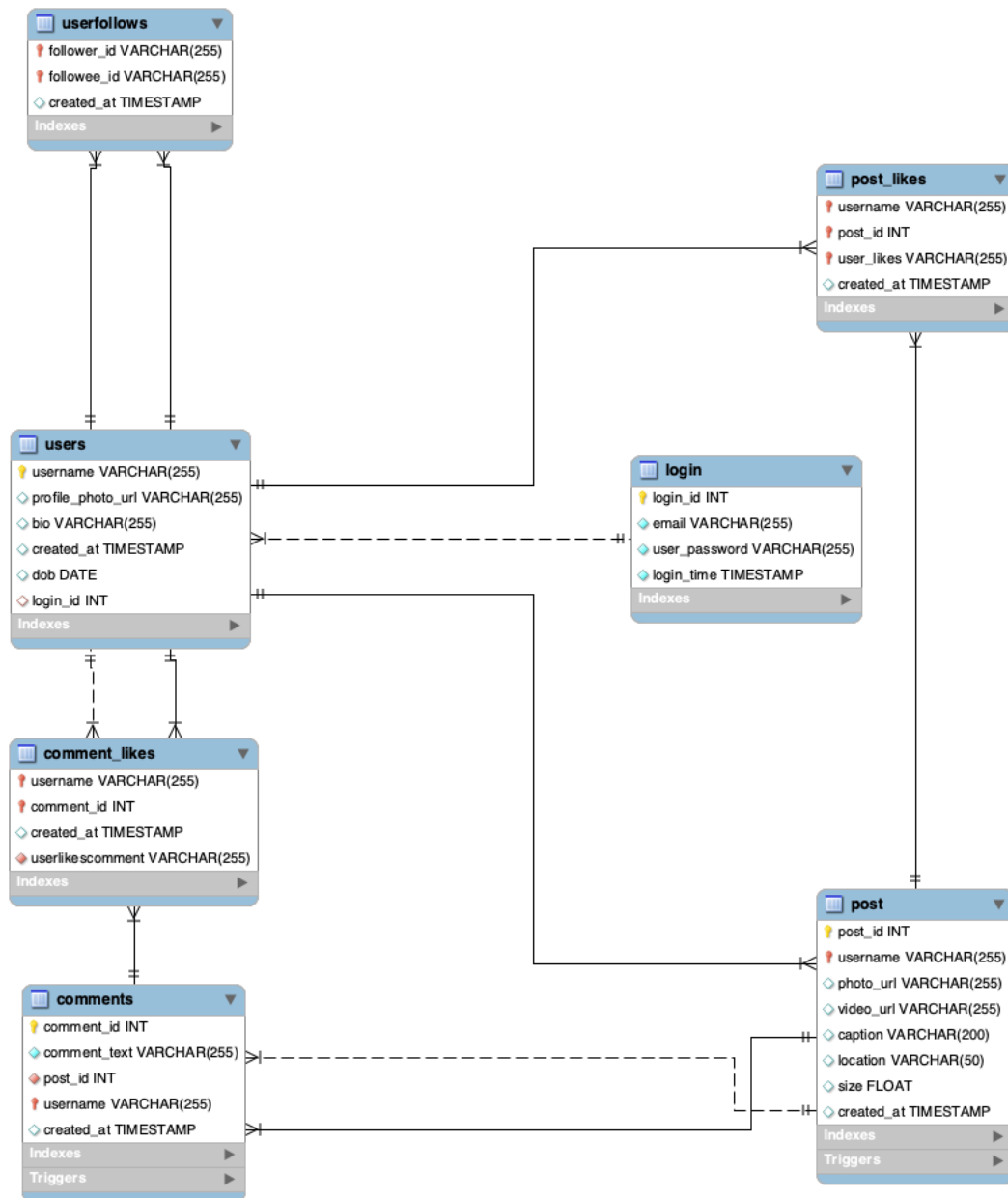
RUN THE APPLICATION

Compile and run the Java code in your preferred IDE or using command-line tools. Ensure that the MySQL server is running before executing the application.

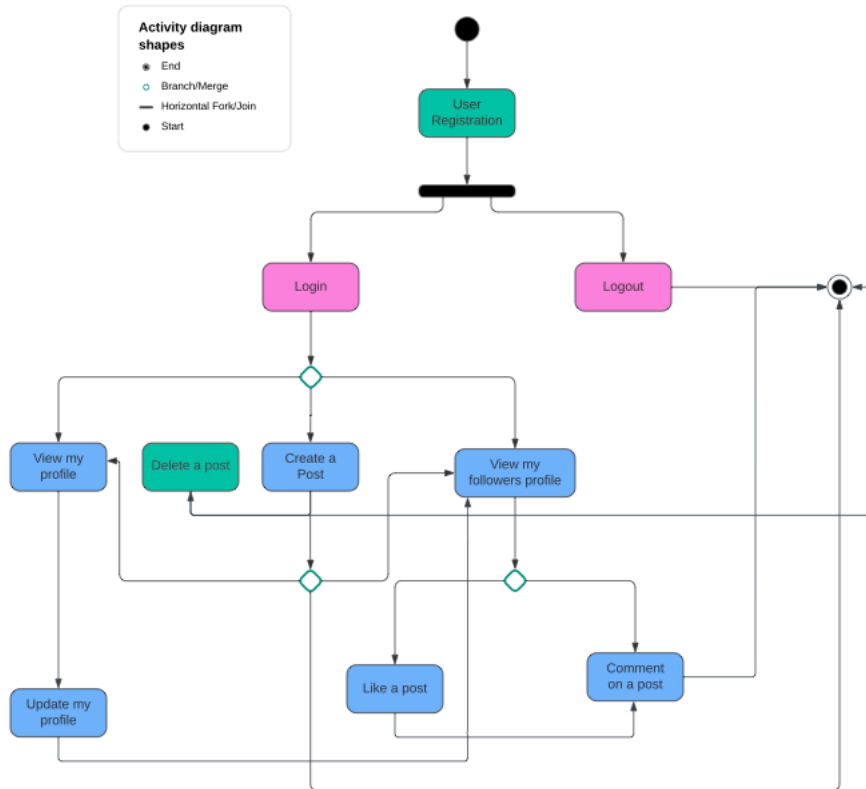
CONCEPTUAL DESIGN/UML DIAGRAM



LOGICAL DESIGN/REVERSE ENGINEERED SCHEMA



FINAL USER FLOW/ACTIVITY DIAGRAM



TECHNICAL SPECIFICATIONS:

1. **Java Version:**
The project is developed using Java programming language.
Recommended Java version: Java 8 or later.
2. **MySQL Database:**
The project uses the MySQL database to store and manage data.
The MySQL Connector/J library is employed for Java-MySQL connectivity.
3. **Database Design:**
The MySQL Workbench is utilized for designing and managing the database schema.
The database schema includes tables for users, posts, comments, and other relevant entities.
4. **JDBC (Java Database Connectivity):**
The JDBC API is used to interact with the MySQL database.
The SocialMediaCLI class manages database connection, disconnection, and provides a connection object for data access.
5. **User Interface:**
The project includes a simple command-line interface for user interaction. The SocialMediaCLI class handles user input, displays menus, and calls methods based on user choices.

LESSONS LEARNED

1. **Technical Expertise Gained:**
Throughout the development of the project, the team acquired valuable technical expertise in various areas. The use of Java as the programming language provided a robust foundation for building the application. The integration of the MySQL Connector allowed seamless communication between the Java application and the MySQL database. Additionally, working with MySQL Workbench enhanced our skills in database design and management.
2. **Insights and Time Management:**
One significant insight gained during the project was the importance of effective time management. Clear project milestones and deadlines were essential to ensure that each phase of development was completed on schedule. Regular team meetings and task prioritization played a crucial role in achieving project goals within the specified time frame.
3. **Data Domain Insights:**
Working on a social media application exposed the team to the intricacies of data modeling for user-related functionalities, such as user posts, comments, and interactions. Understanding the relationships between different entities in the database schema was critical for designing an efficient and scalable system.
4. **Realized or Contemplated Alternative Design/Approaches:**
During the project, we encountered situations where alternative design approaches were considered. For instance, the decision to use a relational database (MySQL) over a NoSQL database (MongoDB) was carefully evaluated based on the specific

requirements of the application. While a relational database offered better support for complex relationships between entities, a NoSQL database could provide advantages in terms of scalability and flexibility.

5. **Documenting Code Challenges:**

The development process also involved overcoming challenges in the code. Documenting these challenges and the solutions applied is crucial for future reference. For instance, dealing with foreign key constraints and ensuring data consistency required careful handling, and documenting these experiences will serve as a valuable resource for troubleshooting similar issues in the future.

FUTURE WORK

1. **Enhanced User Interface:** Improve the user interface to make it more user-friendly and visually appealing. Consider implementing a web-based or graphical user interface for a better user experience.
2. **Advanced Search and Filters:** Extend the search functionality to allow users to filter and search for specific content based on various criteria, such as location, date, or user-specific preferences.
3. **Mobile Application:** Develop a mobile application to provide users with on-the-go access to the platform. This could involve creating native apps for iOS and Android devices.
4. **Notifications System:** Implement a notification system to alert users about new comments, likes, or followers. This could enhance user engagement and interaction.
5. **Data Analytics and Insights:** Integrate data analytics tools to gather insights into user behaviour, popular content, and trends. This data can be used to improve the platform and tailor it to user preferences.
6. **Security Enhancements:** Strengthen the security measures by implementing features such as two-factor authentication, encryption of sensitive data, and regular security audits to ensure the protection of user information.
7. **Integration with External Platforms:** Allow users to share their posts on other social media platforms or integrate with external services for a more connected experience.
8. **Multimedia Support:** Extend the platform to support a wider variety of multimedia content, such as GIFs, interactive images, or live streaming.
9. **Gamification Elements:** Introduce gamification elements, such as badges, achievements, or a point system, to encourage user engagement and loyalty.
10. **Collaborative Features:** Enable users to collaborate on content creation, allowing multiple users to contribute to a single post or participate in shared albums or events.

CONCLUSION

In conclusion, this project has been a comprehensive endeavour aimed at developing a social media database management system. Throughout the development process, we encountered various challenges and learned valuable lessons that have contributed to our growth as developers.

One of the primary objectives of this project was to an application where a user is able to upload photos, videos and comment on other users post .We successfully achieved this goal by developing a command line argument user interface. This project allowed us to apply theoretical knowledge gained during our coursework to a real-world scenario, providing a hands-on experience in software development.