 ***Tweet-Tok***

**Database Design Document (DDD)**

**Version *1.00.00***

**Prepared by: *Evangelista, John Mikael D.***

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Introduction 3](#_Toc1698141376)

[1.1 Document Objectives 4](#_Toc501146363)

[1.2 Intended Audiences 4](#_Toc1382327303)

[1.3 References 4](#_Toc903257328)

[2 Detailed Database Design 5](#_Toc183215146)

[2.1.1 Data dictionary 5](#_Toc1172039121)

[2.1.1.1 Data dictionary for Element: adminUsers 5](#_Toc1665640972)

[2.1.1.2 Data dictionary for Element: tiktokUsers 5](#_Toc1299497928)

[2.1.1.3 Data dictionary for Element: tiktokUsersPost 6](#_Toc1444700551)

[2.1.1.4 Data dictionary for Element: tiktokUserFollow 6](#_Toc2046547168)

[2.2 MySQL database design (Relational database) 7](#_Toc606145078)

[2.2.1 Conceptual diagram 7](#_Toc410882953)

[2.2.2 Description 7](#_Toc122104620)

[2.2.3 Purpose of Tables 8](#_Toc800428340)

[2.2.3.1 Purpose of adminUser Table 8](#_Toc1933560191)

[2.2.3.2 Purpose of tiktokUsers Table 8](#_Toc811240682)

[2.2.3.3 Purpose of tiktokUsersPost Table 8](#_Toc903616028)

[2.2.3.4 Purpose of tiktokUserFollow Table 8](#_Toc1731950957)

[2.2.4 Relations 9](#_Toc1933511333)

[3 References 10](#_Toc1956539387)

[4 Appendix 1 – XML Schema 10](#_Toc886540697)

# Introduction

The section introduces the Database Design Document (DDD) for LocAdoc to its readers.

## Document Objectives

This Database Design Document (DDD) for the Tweet-Tok software has the following objectives:

* ***Describe the Database Design***: To provide a comprehensive description of the relational database structure that supports the Tweet-Tok social media platform. This includes defining key entities such as adminUser, tiktokUsers, tiktokUsersPost, and tiktokUserFollow, along with their attributes (e.g., UserNameID, PostID, FollowID), relationships (e.g., users posting content, following other users), and constraints (e.g., primary and foreign keys). The document will outline how data is organized and managed within a Database Management System (DBMS) to enable efficient storage, retrieval, and manipulation of user profiles, posts, and follow relationships.
* ***Explain Data Access and Manipulation***: To detail the mechanisms and software units (e.g., SQL queries, stored procedures, or application logic) used to access and manipulate the database. This includes describing operations such as Create, Read, Update, Delete (CRUD) for entities like user profiles and posts, as well as specific actions like following or unfollowing users, viewing posts, and administrative oversight, all of which are reflected in the ERD’s operational nodes.
* ***Serve as the Basis for Implementation***: To act as a definitive guide for implementing the database, ensuring that the design aligns with the Tweet-Tok platform’s functional requirements (e.g., user interactions, content management) and non-functional requirements (e.g., performance, scalability). The document will include schema definitions, indexing strategies, and data integrity rules to facilitate accurate database creation.
* ***Provide Visibility for Support and Evaluation***: To ensure transparency into the database design for stakeholders such as acquirers, technical reviewers, and support teams. This visibility enables evaluation of the design’s quality, security, and scalability, while also supporting ongoing maintenance, troubleshooting, and potential enhancements of the Tweet-Tok platform.

## Intended Audiences

This DDD is intended for the following audiences:

* Programmers: Developers tasked with implementing the database schema, indexes, and related software units. They will rely on this document to build a functional database that supports the Tweet-Tok platform’s features, such as user management and post retrieval.
* Testers: Quality assurance professionals who will develop and execute test cases to validate the database design. They will use the document to confirm data integrity, performance under load, and correct execution of operations like posting content, following users, and administrative tasks.

## References

This DDD refers to the following references:

* CTINFMGL Project Specifications

# Detailed Database Design

This section describes the actual design of different databases at varying levels of abstraction. A subsection for each of conceptual, internal, logical and physical levels.

### Data dictionary

#### Data dictionary for Element: adminUsers

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data Type** | **Constrain** | **Description** |
| **id (Primary Key)** | Integer | Auto Increment | The ID of the admin account. |
| **adminUser** | String |  | The username of the admin account. |
| **adminPassword** | Integer |  | The password of the admin account. |
| **account\_created** | String Time Stamp |  | The date of the admin account created. |

#### Data dictionary for Element: tiktokUsers

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data Type** | **Constrain** | **Description** |
| **UserNameID (Primary Key)** | String |  | The UserNameID of the user. |
| **FirstName** | String |  | The First Name of the user. |
| **LastName** | String |  | The Last Name of the user. |
| **Email** | String | UNIQUE | The Email of the user. |
| **UserPassword** | String |  | The Password of the user. |
| **Pronoun** | String |  | The Pronoun of the user. |

#### Data dictionary for Element: tiktokUsersPost

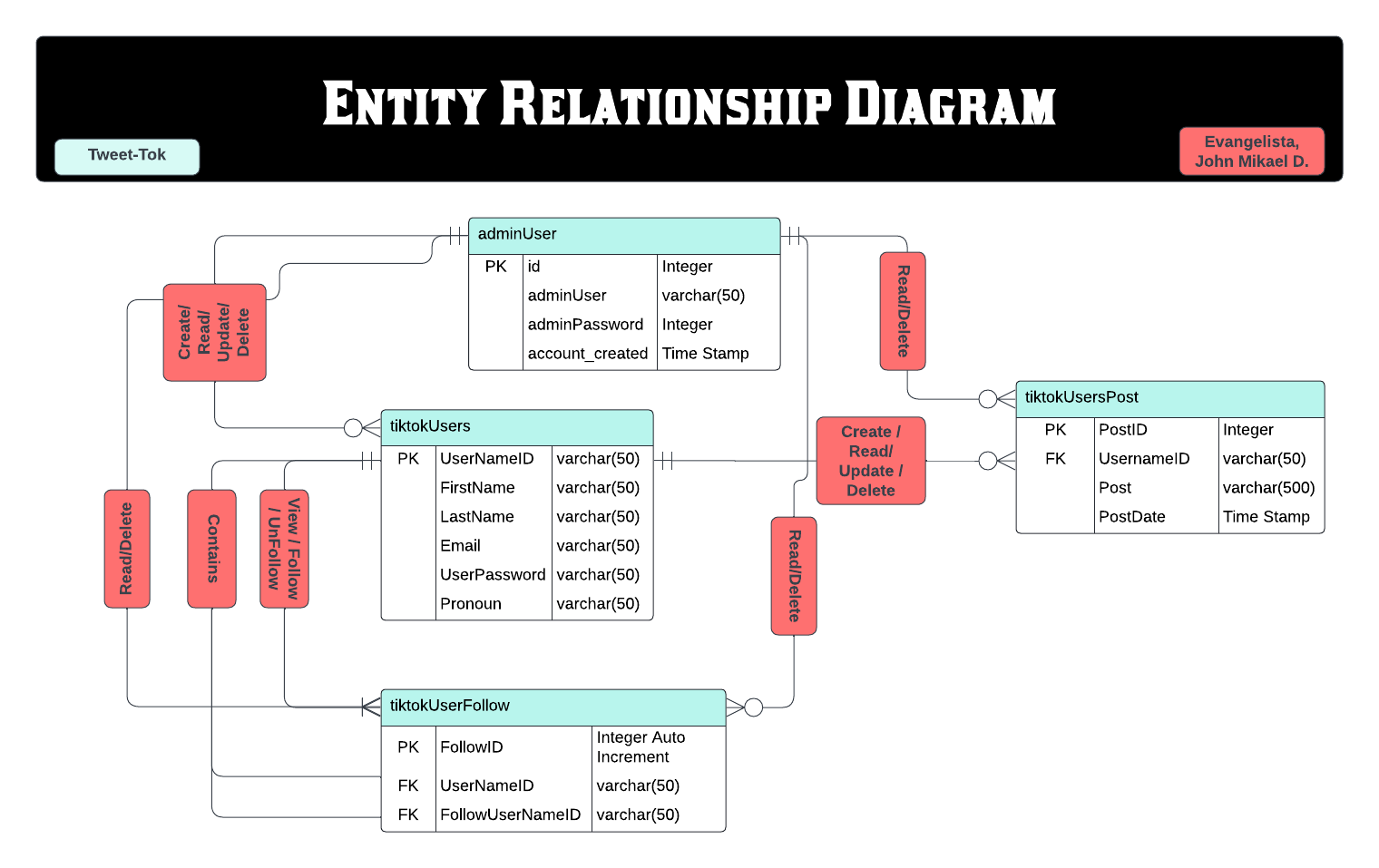
|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data Type** | **Constrain** | **Description** |
| **PostID (Primary Key)** | Integer | Auto Increment | The ID of the user Post. |
| **UserNameID (Foreign Key)** | String |  | The UserNameID of the user. |
| **Post** | String |  | The post of the user. |
| **PostDate** | String Time Stamp |  | The date of the post created. |

#### Data dictionary for Element: tiktokUserFollow

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data**  **Type** | **Constrain** | **Description** |
| **FollowID (Primary Key)** | String |  | The ID of the user follow. |
| **UserNameID (Foreign Key)** | String |  | The UserNameID of the user. |
| **FollowUserNameID (Foreign Key)** | String |  | The UserNameID of the other user. |

## MySQL database design (Relational database)

### Conceptual diagram



### Description

The Entity Relationship Diagram (ERD) titled "Entity Relationship Diagram" represents the database structure for a social media platform named "Tweet-Tok," as indicated by a light blue label in the top left corner. Created by Evangelista, John Mikael D. The diagram outlines four main entities: adminUser, tiktokUsers, tiktokUsersPost, and tiktokUserFollow. These entities are interconnected through relationships that define their interactions, with specific operations associated with each entity, depicted in red boxes.

The ERD for "Tweet-Tok" defines a database structure for a social media platform with four key entities:

* **adminUser**: Manages and Oversees the system with full CRUD operations.
* **tiktokUsers**: Represents users who can (CRUD) profiles, can (CRUD) posts, can follow and unfollow other users, and can search other user's profile.
* **tiktokUsersPost**: Stores user's post.
* **tiktokUserFollow**: Tracks follow relationships between users.

Relationships connect adminUser to tiktokUsers for administrative functions, tiktokUsers to tiktokUsersPost for content creation, and tiktokUsers to tiktokUserFollow for social interactions. The diagram employs standard ERD notation to clearly illustrate the schema, supporting user management, content sharing, and following features.

### Purpose of Tables

#### Purpose of adminUser Table

This table facilitates secure administrative access and oversight of the platform. It authenticates admin users and enables them to manage user accounts, posts, and follow relationships. Sensitive data across the database (excluding primary and foreign keys) is encrypted using the admin’s password and decrypted upon request, ensuring secure administrative operations.

#### Purpose of tiktokUsers Table

This table manages user authentication, personal details, and session handling for logged-in users. It supports personalization and social interactions within the platform. If set, the user’s password encrypts sensitive data (e.g., posts and follow relationships) across the database, which is decrypted when the user accesses their profile or related content. An admin area within the platform allows users to update account details, such as their password. All data, except keys, is stored encrypted for security.

#### Purpose of tiktokUsersPost Table

This table enables users to create, read, update, and delete their posts, supporting core content management features. Each post is encrypted using the creator’s password and decrypted only for the user or authorized viewers, ensuring data privacy. The foreign key UserNameID establishes a one-to-many relationship with tiktokUsers, allowing one user to have multiple posts.

#### Purpose of tiktokUserFollow Table

This table manages social connections by recording who follows whom, supporting the platform’s social networking features. Follow relationships are encrypted using the follower’s password and decrypted when the user interacts with their follow list or views followed users’ content, maintaining privacy. The dual foreign keys create a many-to-many relationship between users via the tiktokUsers table.

### Relations

|  |  |  |
| --- | --- | --- |
| **From Table** | **To Table** | **Relation** |
| adminUser | tiktokUsers | A admin can CRUD tiktokUsers |
| adminUser | tiktokUsersPost | A admin can Read/Delete tiktokUsersPost |
| adminUser | tiktokUserFollow | A admin can tiktokUserFollow |
| tiktokUsers | tiktokUsersPost | A user can CRUD posts |
| tiktokUsers | tiktokUserFollow | A user can follow and unfollow users. |
| tiktokUsers | adminUser | A user can be CRUD by an admin |
| tiktokUsersPost | adminUser | A post can be read and delete by an admin |
| tiktokUsersPost | tiktokUsers | A post can be CRUD by an user |
| tiktokUserFollow | adminUser | A follow can be read and delete by an admin |
| tiktokUserFollow | tiktokUsers | A user can follow and unfollow other users |

# References

1. " CTINFMGL Project Specifications "<https://believed-bongo-319.notion.site/CTINFMGL-Project-Specifications-19296450aad180aea9ebf27987415f4c>

# Appendix 1 – XML Schema