**SYSTEM REQUIREMENT SPECIFICATION**

PROJECT TITLE: **INTER UNIVERSITY SOCIAL MEDIA**

APPLICATION NAME: **UNIGRAM**

* PROJECT SCOPE:
  + - Our project scope encompasses essential features and functionalities that provide core management value to users.
    - Inter University Social Media(Unigram) will be used to enable university students to communicate easily and share information, ideas and inspiration among other students.
* PROJECT REQUIREMENTS:

1. ***FUNCTIONAL FEATURES***
2. **User Registration and Authentication:**

* **Functionality**:

Allows users to register for the application and authenticate themselves.

* **Description**:

Users can create an account by providing their names, email and password. Secure authentication mechanisms ensure user data is protected.

* **Input**:

User-provided names, email and password.

* **Process:**
  + When a user registers, their email and password are validated.
  + User credentials (email) are stored securely in the database.
  + The password is hashed before storage.
  + Authentication tokens or session data are generated and stored in the database, associating them with the user’s account.
* **Database Interaction:**
  + **Users** table: Stores user account details, including names, email, hashed password, and authentication tokens or session data.
* **Output**:
  + Successful registration or authentication.
  + Access tokens or session information for authenticated users.

**2. Profile Creation:**

* **Functionality:**

Allows users to create and personalize their user profiles.

* **Description**:

Users can customize their profiles by adding profile pictures, display names, and optional information to make their presence unique.

* **Input**:

User-provided profile information, such as profile picture, display name, and optional details.

* **Process:**

User-provided profile information, such as profile picture, display name, and optional details, is stored in the database.

* **Database Interaction:**

**users** table: Contains user-specific data, including profile information.

* **Output**:

User profiles with personalized information.

**3. Friend/Contact Management:**

* **Functionality**:

Provides users with tools to manage their friends and contacts.

* **Description**:

Users can add, remove, or interact with their friends or contacts, enabling social connections within the app.

* **Input**:

User actions, such as adding or removing friends.

* **Process:**
  + When a user adds a friend/contact, the relationship is recorded in the database.
  + When a user removes a friend/contact, the relationship is updated or deleted in the database.
* **Database Interaction:**
  + **contact** table: Manages relationships between users, indicating who is connected with whom.
* **Output**:
  + Updated friend/contact lists for users.

**4. Chatting and Messaging:**

* **Functionality**:

Implements real-time chat functionality for users.

* **Description**:

Users can send text messages in real-time to their contacts, providing a seamless communication experience.

* **Input**:

User-provided text messages, recipient information.

* **Process:**
  + When a user sends a message, it's stored in the database with details like content, sender, recipient, timestamp, and read status.
  + Messages are associated with a specific conversation using the **conversation\_id** foreign key.
* **Database Interaction:**
  + **message** table: Stores individual messages, including content, sender, timestamp, and read status.
  + **conversation** table: Helps organize messages by grouping them based on the conversation they belong to.
* **Output**:

Real-time chat messages displayed to users.

Message delivery and read statuses.

**5. Notifications:**

* **Functionality**:

Notifies users about important activities within the app.

* **Description**:

Users receive notifications about new messages and other relevant events, enhancing user engagement.

* **Input**:

Triggered events, such as new messages or friend requests.

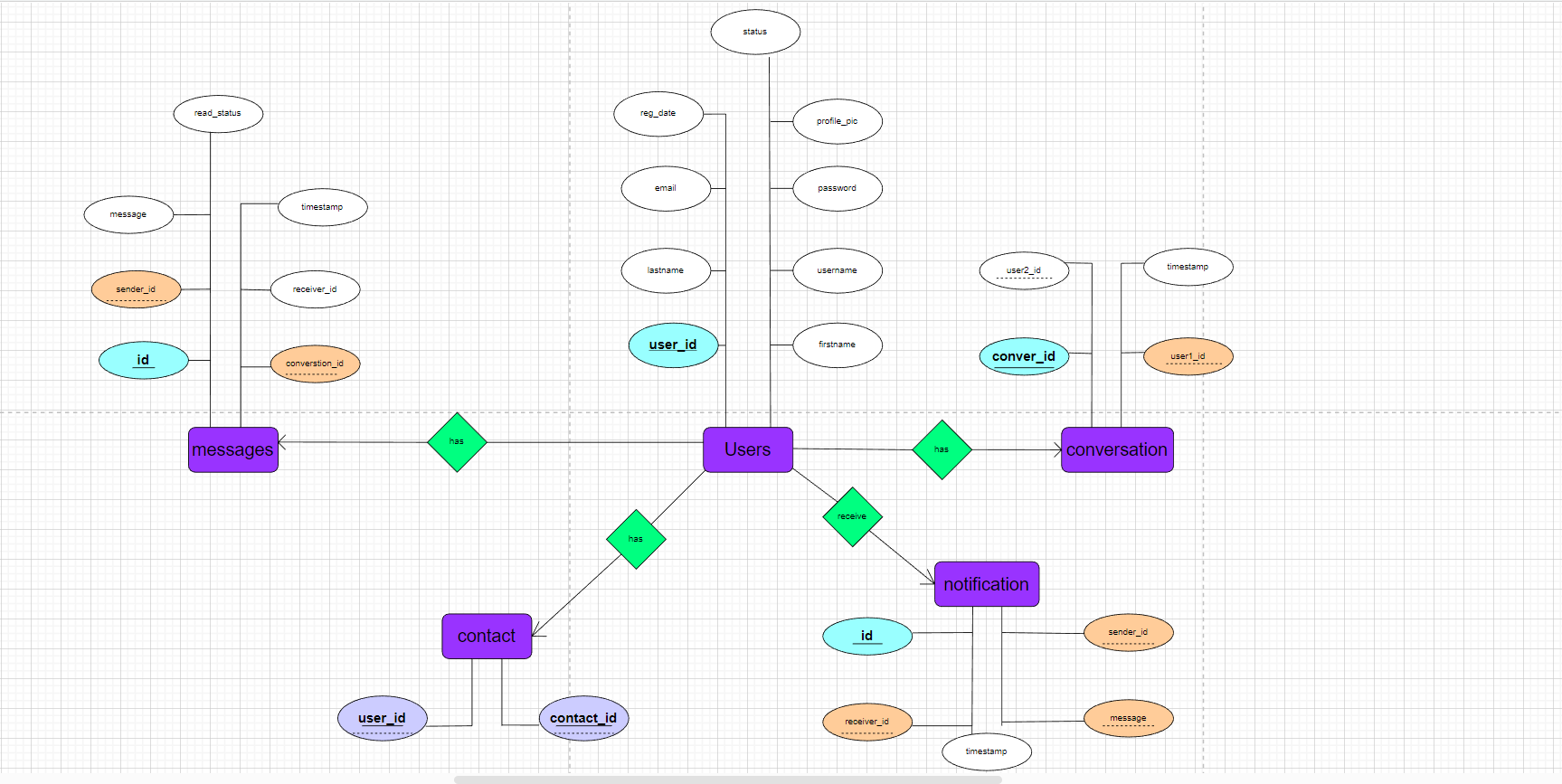
* **Process:**
  + Notifications are generated based on specific events like new messages or friend requests.
  + These notifications are stored in the database, associating them with the user(s) they are intended for.
* **Database Interaction:**
  + **notification** table: Contains notification data, including the recipient, sender, message, and timestamp.
* **Output**:

Notifications delivered to users for real-time updates.

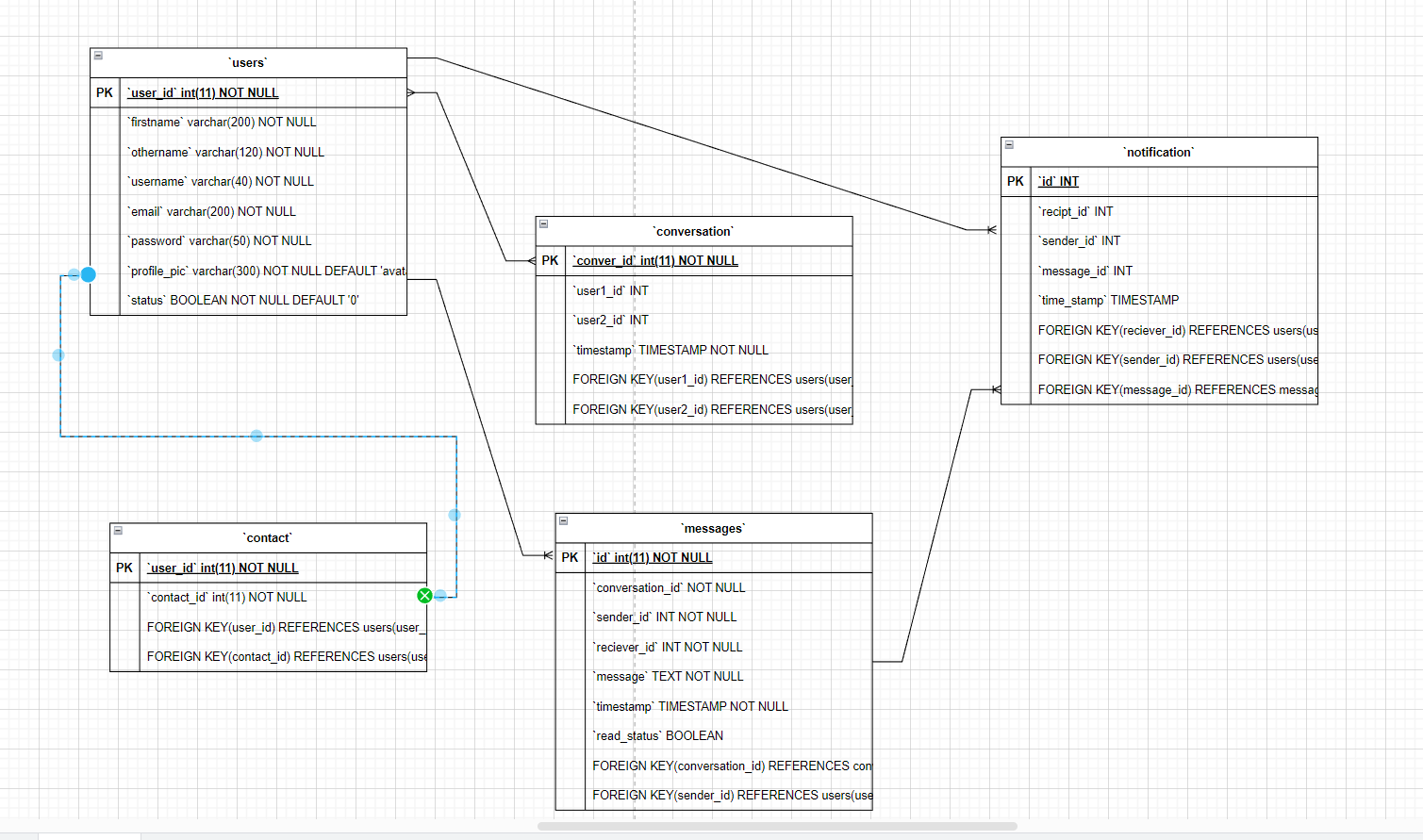
1. ***NONE FUNCTIONAL***

* User Interface Design
  + Home Page
  + Chat List
  + Contacts List
  + Chatting Interface
* Compatibility
  + Multiple Browsers
  + Responsiveness
* Useability
  + Ease To Use And Engage With The App
* Accessibility:
  + Design the app with accessibility features for users with disabilities.
  + Ensure that the app is usable by a diverse range of users.

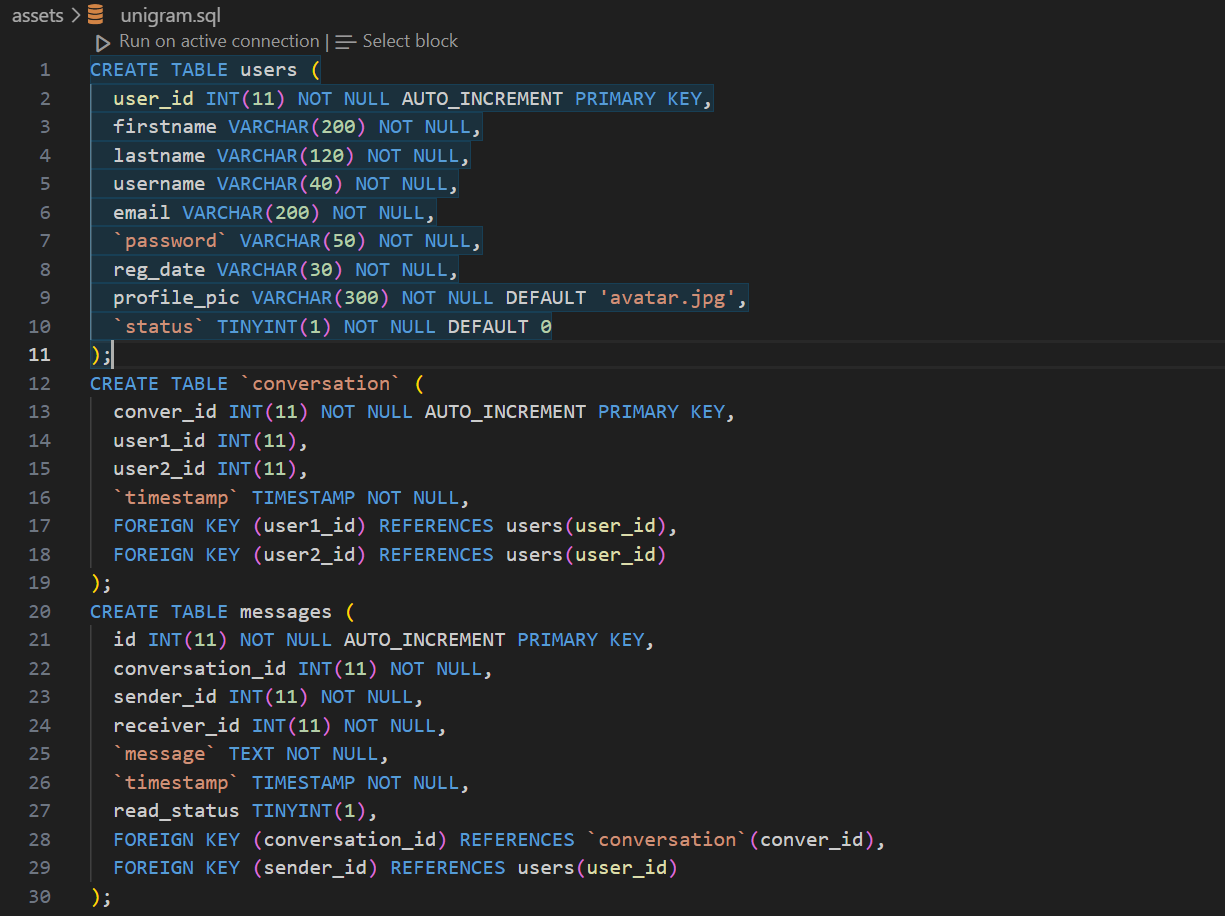
**ERD FOR UNIGRAM DATABASE**

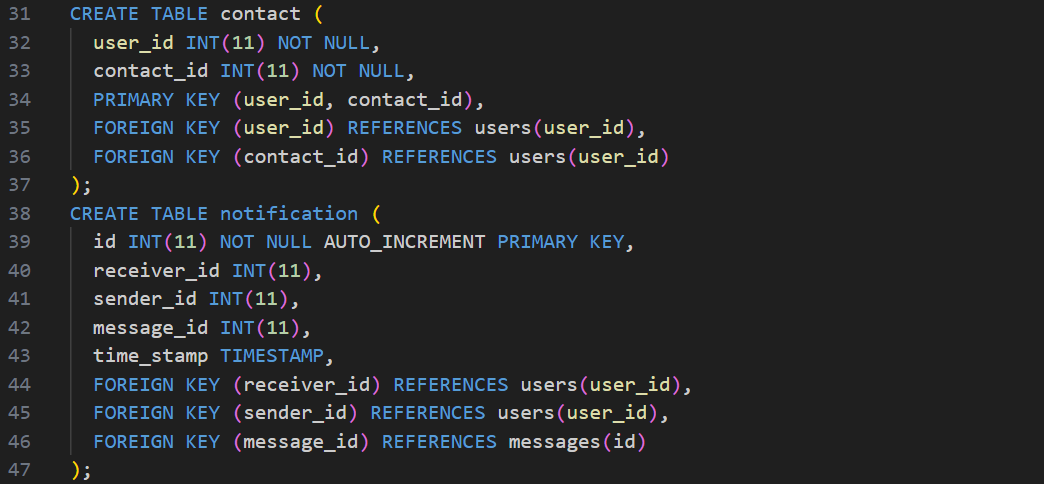
****

**LOGICAL SCHEMA FOR UNIGRAM DATABASE**

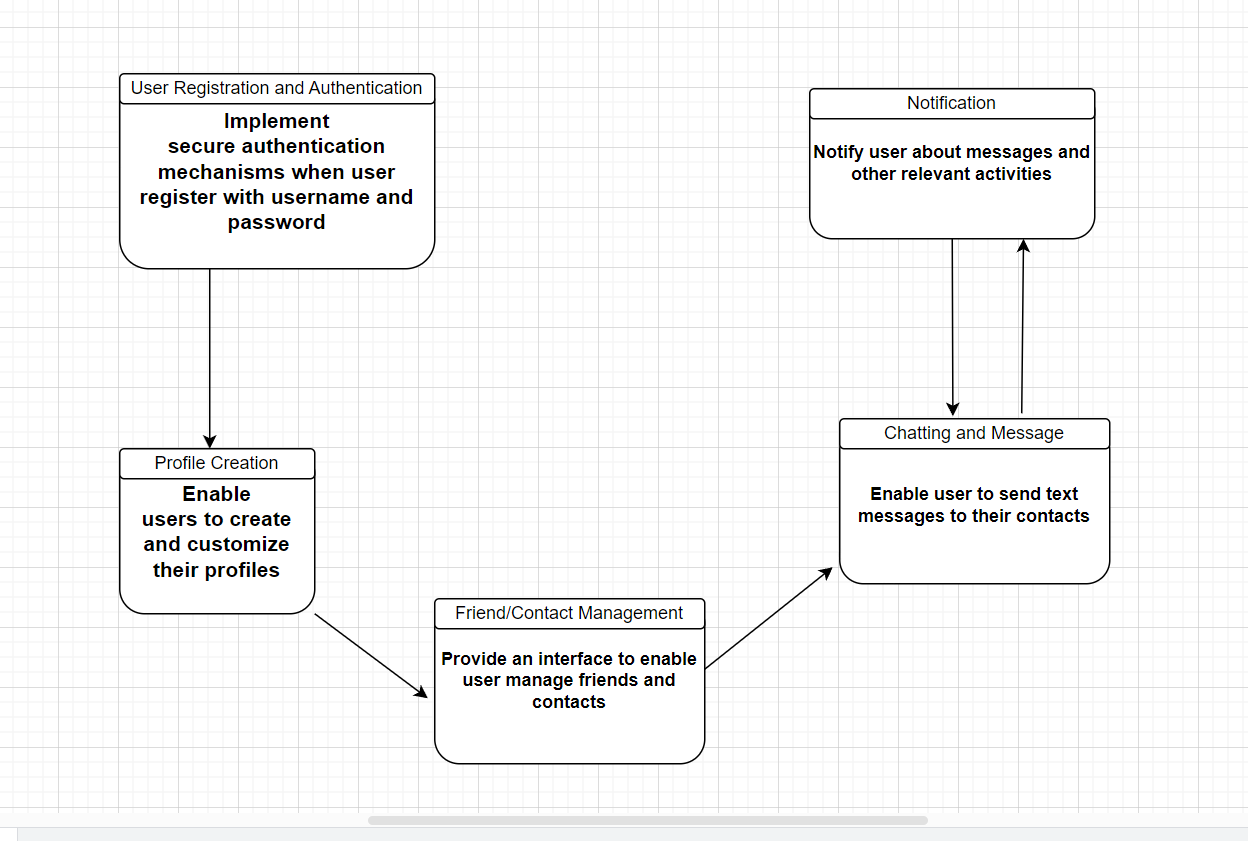
****

**PHYSICAL SCEHAM FOR UNIGRAM DATABASE**

****

****

**FFBD FOR UNIGRAM DATABASE**

****

**TEAM MEMBERS:**

|  |  |
| --- | --- |
| NAMES | REGISTRATION NO |
| SALEHE NYATU | T22-03-10342 |
| SHABAN JUMA MASENZA | T22-02-03577 |
| YASRI JUMA NTETERE | T22-03-11731 |
| HONEST A ASSEY | T22-02-02824 |
| KELVIN J ASSENGA | T22-03-11783 |
| DENIS A MREMA | T22-03-08920 |
| ERICK K SHAYO | T22-03-09516 |
| PAUL MSALU | T22-02-02887 |