Shopping List App

1. Project Description and Conceptual Model

The Shopping List application allows users to create and manage multiple shopping lists. Each list can contain various items, and users can invite others to collaborate. The owner can add or remove members and delete lists. Both owners and members can add, remove, and mark items as completed.

Conceptual Model:

- **User**: Represents an individual using the application.
- **ShoppingList**: A list created by a user to manage shopping items.
- **Item**: Represents a product or object to be purchased, with a status indicating whether it is completed.
- **ListMember**: Connects users to shopping lists, allowing shared access with roles ('owner' or 'member').

2. Relational Schema

Tables and Relationships:

1. **User**
   - user_id (PK)
   - username
   - email
   - password_hash
2. **ShoppingList**
   - list_id (PK)
   - list_name
   - owner_id (FK references User.user_id)
   - created_at
3. **Item**
   - item_id (PK)
   - item_name
   - list_id (FK references ShoppingList.list_id)

    - is_completed (BOOLEAN)

4. **ListMember**
    - list_id (FK references ShoppingList.list_id)
    - user_id (FK references User.user_id)
    - role (e.g., 'owner', 'member')
    - PRIMARY KEY (list_id, user_id)

3. Physical Model

- **User**: Stores essential user information.
- **ShoppingList**: Links to the user who owns the list.
- **Item**: Each item is associated with a specific list and can be marked as completed.
- **ListMember**: Manages collaborations, mapping users to lists with defined roles.