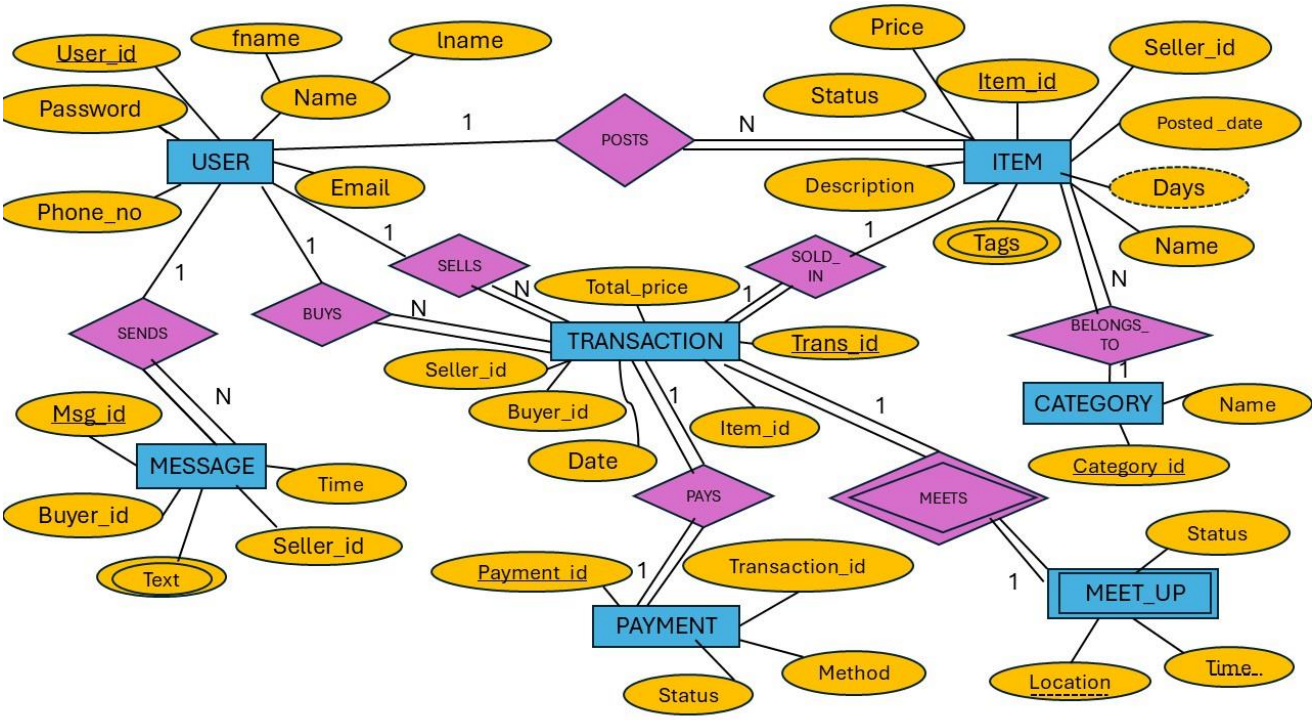
CS 3380 PROJECT PHASE 1

The **university marketplace (Truman's List)** is a database system where college students can buy and sell items such as textbooks, electronics, and furniture. As far as I know, there is no specific platform where Mizzou students can buy and sell items from other students. I have often seen students posting rental places, furniture, and electronic equipment on social media platforms like Snapchat. However, this application will allow users (students) to post items for sale, browse available items, communicate with sellers, and complete transactions, including managing meet-up locations and times for item exchanges, easily and efficiently.

This database system is essential because it will organize and manage the large amount of data generated by the marketplace in a structured and efficient way. Key features such as tracking users, transactions, messages, and meet-ups require a well-designed database to ensure data consistency, quick retrieval, and security. The database will allow users to filter items by category, manage their buying/selling history, and automate notifications for communication between buyers and sellers. Without a robust database, the platform would struggle to maintain accurate records of transactions, manage user interactions, and handle the dynamic nature of a university marketplace.



This database system is designed to manage various entities and relationships to bring a seamless experience for the users.

Entities :

Users: Users are students who buy and sell items. Each user has personal attributes such as a unique user ID (Student_ID), name, email, phone number, and password. Users can be both buyers and sellers.

Items: These are the products listed for sale on the platform. Each item has an associated item ID, name, description, price, status (available, sold, etc.), and is posted by a user. Items also belong to specific categories, which help in filtering and organization. Also, sellers can add tags that can help users/customers to help searching an item. For example, you can add tags like, I phone, 2022, 14 when selling your phone. The derivative attribute days gives the number of days since the item has been posted. It can be calculated using the posted date and the current date

Transactions: This tracks the details of purchases. A transaction includes the transaction ID, buyer ID, seller ID, item ID, total price, and the date of the transaction.

Messages: This entity stores the communication between buyers and sellers. Each message includes a message ID, sender (buyer), receiver (seller), the text of the message, and the time it was sent.

Categories: Categories organize items into groups such as textbooks, electronics, or furniture. Each category has a category ID and a name.

Payments: This entity tracks payment details for each transaction, including payment ID, payment method (cash, credit, etc.), status (completed, pending), and the related transaction ID.

Meet-ups: This weak entity records the meet-up locations and times for the exchange of items between buyers and sellers. It includes the meet-up location, and time, and is linked to a specific transaction through the transaction ID.

Assumptions:

- Each user can post multiple items for sale. Every item can be listed under one or more categories (e.g., an item can belong to both “Furniture” and “Electronics”).
- A transaction is initiated when a buyer purchases an item. Each transaction involves exactly one buyer, one seller, and one item. Once an item is sold, its status is updated, and it can no longer be listed for sale.
- Meet-ups are scheduled based on the agreement between the buyer and seller. The meet-up entity does not have a separate primary key but is associated with a unique Transaction_id and includes the Location and Time to uniquely identify the meet-up
- Every item needs to belong to a category, and categories may exist without items.

- A payment is required for every transaction, meaning both participation and cardinality are 1:1. This ensures each transaction has a corresponding payment record.
- every transaction requires a delivery (meet up)
- each item can only be sold in one transaction.

Functions;

1. Item listing and search
   The system will allow users to post items for sale, and potential buyers will be able to search for items based on various attributes (e.g., category, price, tags). Users can filter listings by category.

   User: Represents the seller posting the item and the buyer searching for items.
   Item: Contains the details of the items being listed, such as name, description, price, and status (available or sold).
   Category: Used to classify items (e.g., electronics, books).

2. Transaction processing
   When a buyer decides to purchase an item, the system processes the transaction by recording the buyer, seller, the item being sold, and the transaction details such as the price and date.

   User: Represents both the buyer and the seller in the transaction.
   Transaction: Records the details of the transaction, including the buyer, seller, item, total price, and transaction date.
   Item: The item being sold is marked as "sold" in the system after the transaction is complete.
   Payment: Captures payment details, including the method and status of the payment.

3. Meet-up scheduling
   After the transaction Buyer and the Seller can communicate via messages and schedule a time and a place to complete the sale. The system records this information.
   User: Both the buyer and seller are involved in the meetup scheduling.
   Message: Allows communication between the buyer and seller to arrange the meeting.
   Meetup: Records the agreed-upon time and location for the transaction to take place, where the buyer and seller meet to complete the sale and status updates to “ completed’’