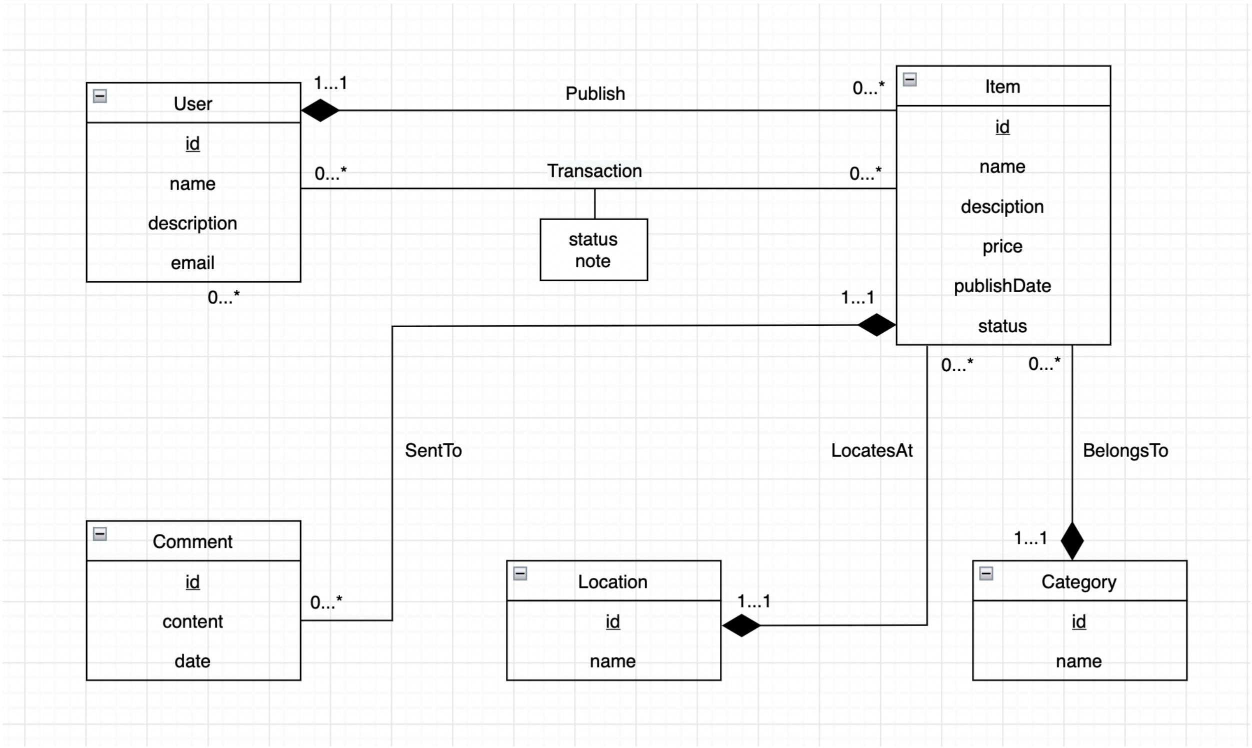
UML Diagram:



Description:

Entity

- User: This entity includes all users of our project including sellers and buyers. We use userId as primary key, and useName, userDescription, and email as other attributes.
- Item: This entity consists of all the items which are published by buyers and sold on the platform. The item id is primary key, which also includes item's name, description, price, publishdata, and status.
- Comment: This entity consists of comments to each item. We use commentId as primary key, content as the content of comment, and data shows the day the user sends comment.
- Location: Our project has several fixed locations to make transactions. This entity includes several possible locations. We use locationId as primary key, and the name of location represents the address.
- Category: To make buyers easily find the product they are looking for, we use categories to define each product. The id of each category is primary key, and the category name is other attributes.

Relationship

- User can publish many items, but each item could only be published by one user
- Each item can have multiple comments, but each comment could only be attached to one item
- Each item could only be traded at one location, where each location can hold multiple trades
- Each item can only belongs to one category, where each category could be assigned to multiple items

Schema:

```
CREATE TABLE User (
  id INTEGER NOT NULL PRIMARY KEY,
  name TEXT,
  email TEXT
);

CREATE TABLE Category (
  id INTEGER NOT NULL PRIMARY KEY,
  name TEXT
);

CREATE TABLE Location (
  id INTEGER NOT NULL PRIMARY KEY,
  name TEXT
);

CREATE TABLE Item (
  id INTEGER NOT NULL PRIMARY KEY,
  userId INTEGER NOT NULL,
  locationId INTEGER NOT NULL,
  categoryId INTEGER NOT NULL,
  name TEXT,
  description TEXT,
  price REAL,
  publishDate DATE,
  status VARCHAR(20),
  FOREIGN KEY (userId) REFERENCES User(id),
  FOREIGN KEY (locationId) REFERENCES Location(id),
  FOREIGN KEY (categoryId) REFERENCES Category(id)
);

CREATE TABLE Comment (
  id INTEGER NOT NULL PRIMARY KEY,
  itemId INTEGER NOT NULL,
  content TEXT,
  date DATE,
  FOREIGN KEY (itemId) REFERENCES Item(id)
);

CREATE TABLE Transaction (
  userId INTEGER NOT NULL,
  itemId INTEGER NOT NULL,
  status VARCHAR(20),
  PRIMARY KEY (userId, itemId),
  FOREIGN KEY (userId) REFERENCES User(id),
  FOREIGN KEY (itemId) REFERENCES Item(id)
);
```