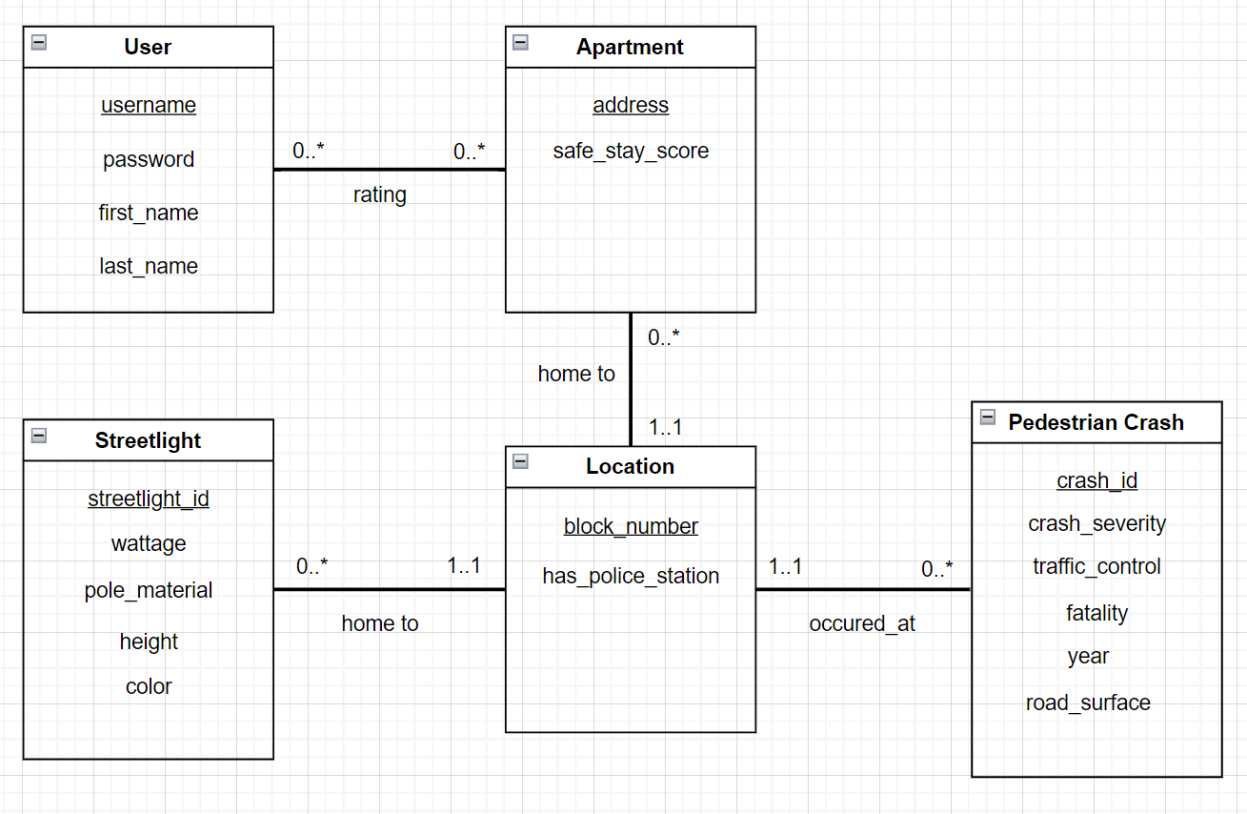
UML Diagram



Explanation of Entities

Our UML diagram contains 6 entities.

1. User
    a. Attributes include username, password, first_name, last_name
    b. Represents individuals who interact with our website. Each user has unique credentials in the form of a username, which is the primary key. Each User will also be required to input their first and last name, as well as a password to login and rate apartments. User makes sense as its own entity because it really encapsulates the necessary information one needs when making an account on the platform/website.
2. Apartment
    a. Attributes are address, and safe_stay_score, has_police_station
    b. Represents the properties in Champaign that a user can rate. It contains the address which is a unique identifier for each apartment. Each apartment also has its own SAFESTAY score which an algorithm we

develop will calculate. We decided to model Apartments as entities since users can interact with them by providing ratings.
    c. Police stations can span multiple blocks so we decided for it to be part of the location entity, and their presence/ absence can impact safety ratings of nearby apartments.
3. Location
    a. Attributes are block_number, and has_police_station
    b. The location entity is used to tie together Streetlight, Police Station, Pedestrian Crash tables. We decided to model this as an entity because this abstraction will allow for shared data like streetlights and crashes to be linked to specific geographic blocks rather than having redundant location data in each entry.
4. Streetlight
    a. Attributes are streetlight_id, wattage, pole_material, height, and pole color
    b. Each streetlight contains unique attributes such as wattage and height. We modeled this as an entity because streetlights can influence various safety metrics, and its attributes such as wattage can be used to calculate apartment and location safety.
5. Pedestrian Crash
    a. Attributes are crash_id, crash_severity, traffic_control, fatality, year, road_surface
    b. We decided to model a pedestrian crash as an entity because multiple pedestrian crashes can occur in a certain location, and its specific properties (like crash severity) are relevant to understanding the area's security.

Relationships and Their Cardinalities

1. “Rating” Relationship between User and Apartment Entities
    a. The point of this relationship is because users can interact with apartments through safety ratings.
    b. This represents a many-to-many relationship. A "User" is very flexible in that they can rate zero or many apartments, and each apartment can receive ratings from zero or many users. The flexibility allows for multiple users to rate the same apartment while allowing a single user to rate multiple apartments as well.
2. “Home to” Relationship between Apartment and Location Entities

    a. This relationship was created to categorize apartments geographically and allow for easy grouping of apartments based on their physical location.
    b. Each apartment must be tied to exactly one specific location but a location can be home to 0 or many apartments. Therefore the relationship between apartment and location is a many to one relationship.
3. “Home to” Relationship between Location and Streetlight Entities
    a. Similar to the apartment-location relationship, this relationship models the distribution of streetlights across different locations.
    b. A Streetlight must be associated with one location but a location can have 0 to many streetlights. Therefore the relationship between streetlight and location is also a many to one relationship.
4. “Occurred at” Relationship between Location and Pedestrian Crash Entities
    a. This relationship represents the association between pedestrian crashes and their locations, which helps to track crash frequency and severity at different locations, and to identify high-risk areas.
    b. A pedestrian crash can only occur at one location but a particular location can be associated with 0 or multiple crashes over time. Therefore, the relationship between location and pedestrian crashes is a one to many relationship.

Normalization

Our schema already adheres to BCNF.

- User: The only functional dependency is Username → {First Name, Last Name, Password}, where Username is a superkey. Hence, User is in BCNF.
- Rating: The only functional dependency is (Username, Apartment) → Rating, where (Username, Apartment) is a superkey. Hence, Rating is in BCNF.
- Apartment: The only functional dependency is Address → {Block Number}, where Address is a superkey. Hence, Apartment is in BCNF.
- Location: The only functional dependency is Block Number → {Has Police Station}, where Block Number is a super key. Hence, Location is in BCNF.
- Streetlight: The only functional dependency is Streetlight ID → {Block Number, Wattage, Pole Material, Height, Color}, where Streetlight ID is a superkey. Hence, Streetlight is in BCNF.
- Pedestrian Crash: The only functional dependency is Crash ID → {Block Number, Crash Severity, Traffic Control, Fatality, Year, Road Surface}, where Crash ID is a superkey. Hence, Pedestrian Crash is in BCNF.

Schema

User(username: VARCHAR(255) [PK], password: VARCHAR(255), first_name: VARCHAR(255), last_name: VARCHAR(255))

Rating(username: VARCHAR(255) [PK, FK to User.username], address: VARCHAR(255) [PK, FK to Apartment.address], rating: INT)

Apartment(address: VARCHAR(255) [PK], block_number: INT [FK to Location.block_number], latitude: FLOAT, longitude: FLOAT, safe_stay_score: FLOAT(3,2))

Streetlight(streetlight_id: INT [PK], block_number: INT [FK to Location.block_number], wattage: INT, pole_material: VARCHAR(255), height: INT, color: VARCHAR(255), latitude: FLOAT, longitude: FLOAT)

Location(block_number: INT [PK], has_police_station: BOOL)

Pedestrian_Crash(crash_id: INT [PK], block_number: INT [FK to Location.block_number], crash_severity: INT, traffic_control: VARCHAR(255), year: INT, road_surface: VARCHAR(255), latitude: FLOAT, longitude: FLOAT)