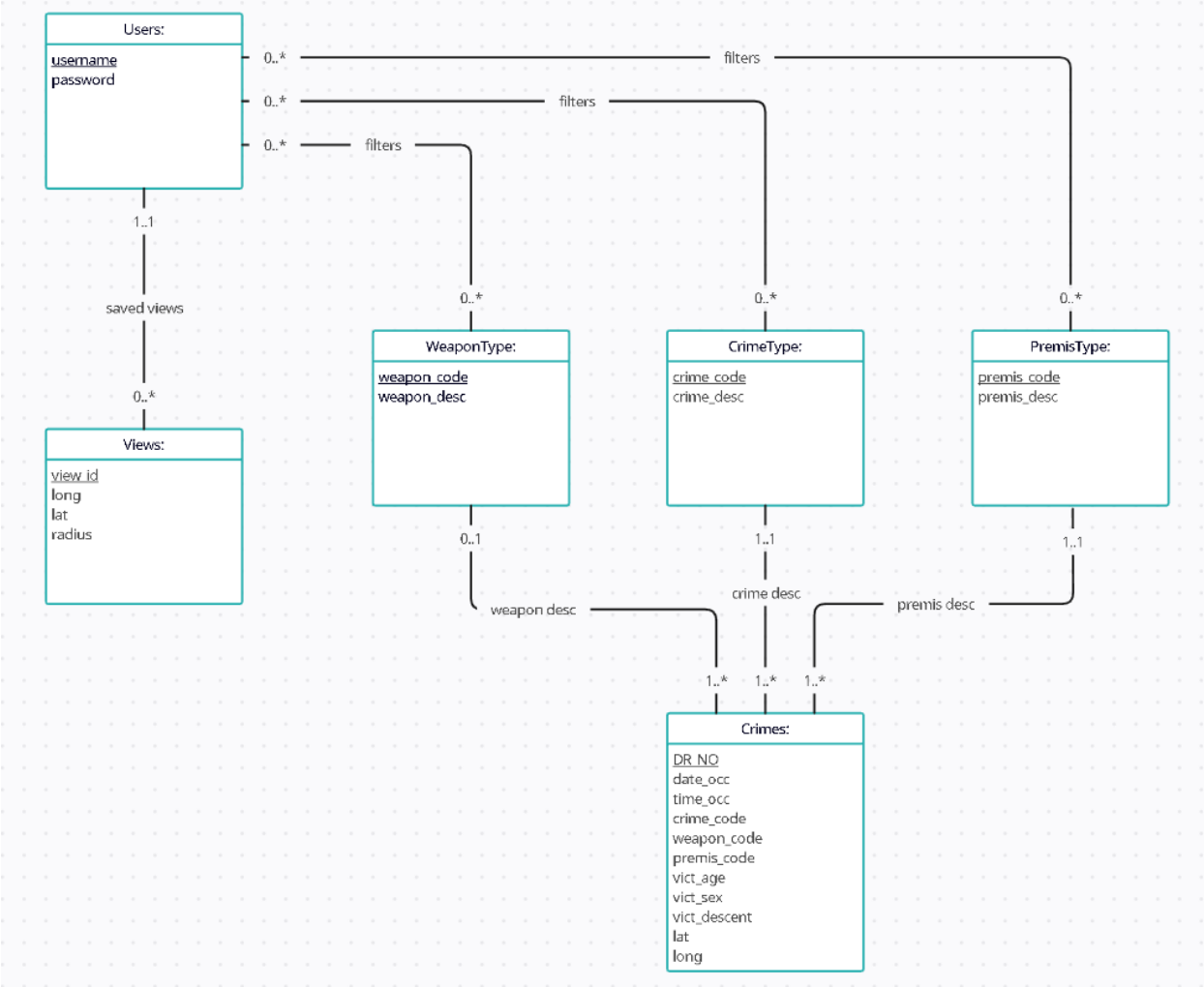
Conceptual UML



Description of Assumptions and Cardinality:

- We assume each user will have one unique username and password. In addition, we think each person will only have 0 or 1 filter at any time, but each filter can only be associated with one user. We assume that each user will be able to filter out multiple different types of crime, weapons, or premise locations to allow for more finetuning and specialized searching when using the website. Because of this, there is a many to many relationship between each filter and the crime type, weapon type, and premise type, where a user can also decide not to filter for any types of crimes, weapons, or premises. Users' relationship with views are explained below.
- We assume each view will have one person associated with it, and that each view will require a location (represented by longitude and latitude) as well as a radius. We think that a person should be able to save multiple views, and therefore a user would associate from 0 to many views, but each view would only be associated with one user.
- We assume that the filter will contain unique IDs relating to how it will filter out different crime, weapon, and premise types.
- We assume that each crime has a specific ID associated with it (the DR_NO), a date that it occurred, the time it occurred, a specific code corresponding to the type of crime, weapon (this can be null), and premise, and that it will contain information regarding the victim (such as age, sex, and descent). Finally, we will also have the location with it, represented as longitude and latitude. We assume that each crime will have a crime type associated with it, and a premise type associated with it. And a single crime/premise type can be associated with crime entities multiple times, creating a many to one relationship. This is a similar case with weapon type, but a crime can be made with no weapon, so allow for the possibility that there is no weapon associated with the crime.
- We assume that each crime type will contain the ID that uniquely identifies it and a description of what that crime is (e.g. vandalism, assault, etc). The relationships with other entities is explained above.
- We assume that each weapon type will contain the ID that uniquely identifies it and a description of what that weapon is (e.g. machete, bat, etc). The relationships with other entities is explained above.
- We assume that each premise type will contain the ID that uniquely identifies it and a description of what that premise is (e.g. single family home, etc). The relationships with other entities is explained above.

Normalization:

FD = {

weapon_code -> weapon_desc;

crime_code->crime_desc;

premis_code -> premis_desc;

dr_no -> weapon_code, crime_code, premis_code, date_occ, time_occ vict_age, vict_sex, vict_descent, lat, long;

view_id-> long, lat, radius

username -> weapon_filter, crime_filter, premis_filter;

}

Minimal basis:

Step1+2:

weapon_code -> weapon_desc

crime_code ->crime_desc

premis_code -> premis_desc

dr_no -> weapon_code

dr_no -> crime_code

dr_no -> premis_code

dr_no -> date_occ

dr_no -> time_occ

dr_no -> vict_age

dr_no -> vict_sex

dr_no -> vict_descent

dr_no -> lat

dr_no -> long

view_id -> long

view_id -> lat

view_id -> radius

username -> weapon_filter

username -> crime_filter

username -> premis_filter

weapon_code+ {weapon_code}

crime_code+ {crime_code}

premis_code+ {premis_code}

dr_no+ {dr_no}

view_id+ {view_id}

username+ {username}

dr_no, view_id, weapon_code, crime_code, premis_code and username are superkeys, premis_code, crime_code, and weapon_code are foreign keys.

These relationships comply with BNCF because every relation has a left hand side that is superkey.
We are using BNCF because it minimizes the redundancy in the functional dependencies of the schema. BCNF removes certain types of redundancies and all redundancies based on FDs are removed. BCNF Decomposition avoids information loss and you can construct the original relation instance from the decomposed relations' instances. This allows for more efficiency compared to 3NF.

Relational Schema:

Crimes(DR_NO: INT [PK], date_occ: DATE, time_occ: TIME,
crime_code: INT [FK to CrimeType.crime_code],
weapon_code: INT [FK to WeaponType.weapon_code],
premis_code: INT [FK to PremisType.premis_code],
vict_age: INT, vict_sex: VARCHAR(255), vict_descent: VARCHAR(255),
lat: REAL, long: REAL)

CrimeType(crime_code: INT [PK] , crime_desc: VARCHAR(255))

WeaponType(weapon_code: INT [PK], weapon_desc: VARCHAR(255))

PremisType(premis_code: INT [PK], premis_desc: VARCHAR(255))

Users(username: VARCHAR(255) [PK], password: VARCHAR(255))

Views(view_id: INT [PK], long: REAL, lat: REAL, radius: REAL)

CrimeFilter(username: VARCHAR(255) [PK][FK], crime_code: INT [PK][FK])

WeaponFilter(username: VARCHAR(255) [PK][FK], weapon_code: INT [PK][FK])

PremisFilter(username: VARCHAR(255) [PK][FK], premis_code: INT [PK][FK])