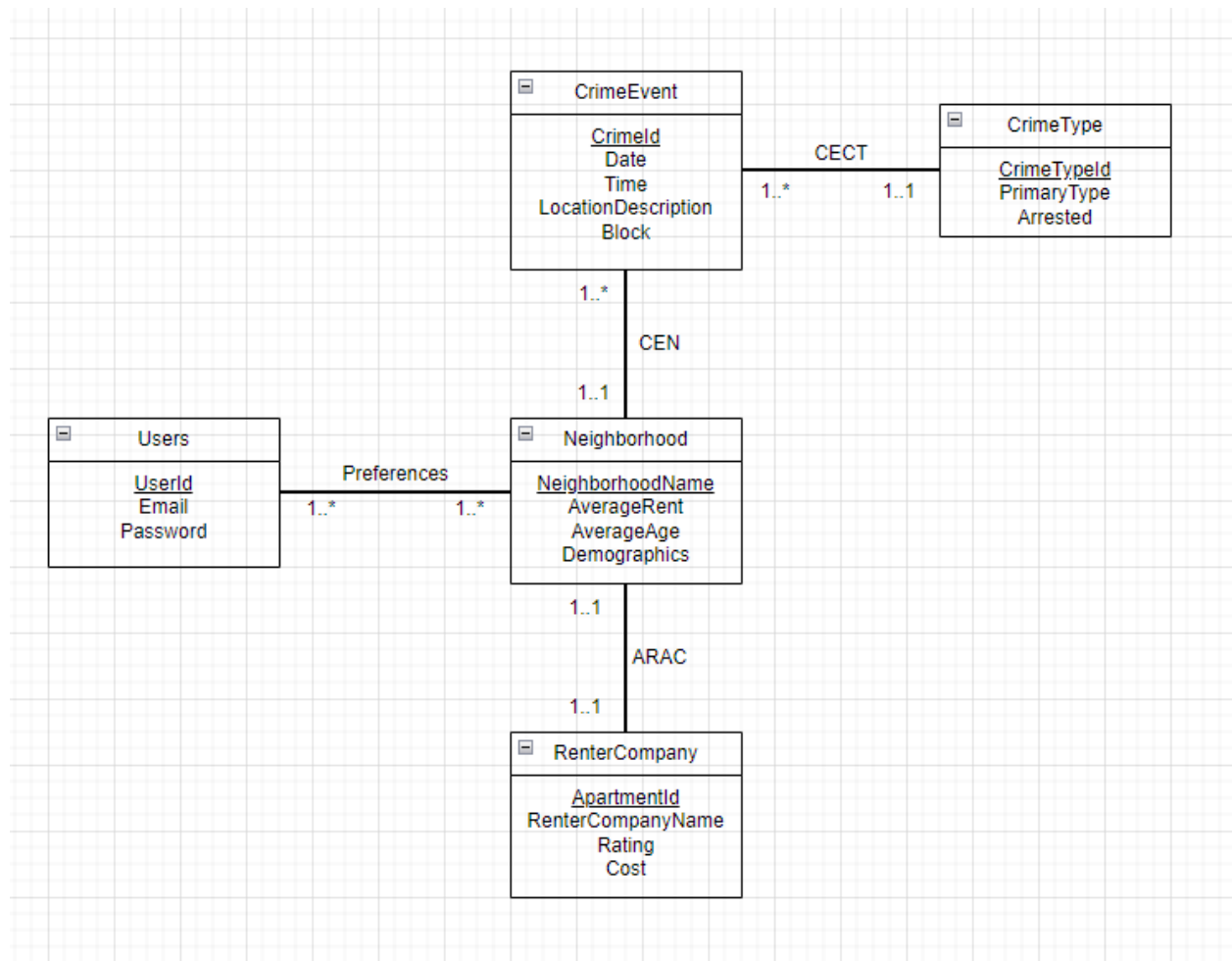


Stage.2: Database Design

UML Diagram



Entities

1. User Account Info

This entity stores the user's information, with 3 attributes.

- **UserId**: an Id that will identify every user based on a unique string, using a string attribute. This will be the primary key.
- **Email**: a string attribute that will be the username to login to the application.

- Password: a string attribute that will be the password to login to the application.

The entity is designed with the following assumptions:

1. Every user will have an auto-generated UserId that cannot be modified. It will be used internally to identify different users and will not be shown publicly.
2. Every user will have an associated Email and Password that they are able to change and update as needed.

2. CrimeEvents

This entity stores all crime events and the related information, with 5 attributes.

- Crimeld: an Id that will identify every crime event based on a unique integer. This will be the primary key and will be of an integer attribute.
- Date: a date attribute, representing the date of the crime.
- Time: a time attribute, representing the specific time of the crime.
- Location Description: a string attribute, representing the location description of the crime.
- Block: a string attribute, representing the street name where the crime took place.

The entity is designed with the following assumptions:

1. We will record the Crimeld, Date, Time, Location Description, and Block in this entity set. This will be important identifiers on where the crime was located.
2. Using this we can identify the specific time and place allowing us to make more precise recommendations.

3. CrimeTypes

This entity stores the crime type Id and the corresponding crime type, with 3 attributes.

- CrimeTypeId: a unique identifier of a crime type. This is the primary key and will be of an integer attribute.

- **PrimaryType:** a string attribute representing the name of a specific type of crime.
- **Arrested:** A boolean attribute representing if the crime led to an arrest or not.

Each Crime Event has their own specific type of crime. Each type also has a different associated level of danger as Battery is less severe to Homicide.

4. Neighborhood

This entity will store information about each location, with 4 attributes.

- **NeighborhoodName:** a unique identifier of location. This is the primary key and will be of a string attribute.
- **AverageRent:** an integer value representing the average rent of an apartment in that neighborhood.
- **AverageAge:** an integer value representing the average age of people in that area.
- **Demographic:** a string value representing the most common ethnicity demographic in that area.

We will ultimately be using the neighborhood entity as the middleman to calculate metrics and suggestions for our users.

5. RenterCompany

This entity stores the Renter Company name and the corresponding average apartment information, with 4 attributes.

ApartmentId: an integer attribute representing a unique Id corresponding to a specific property as well as the Primary Key.

RenterCompanyName: a string attribute representing the company name.

Rating: an integer attribute representing the average rating of the company.

Cost: an integer attribute representing the average cost of their apartments.

The entity is designed with the following assumptions:

1. We will record the Company Name and rating to give a general sense of satisfaction previous customers reported about the company.

2. From this, we will be able to provide stronger recommendations based on the costs and rating.

Relations

1. Preferences (Users and Neighborhoods)

Preferences is a relationship between the Users and Neighborhoods where it will save the users preferred neighborhoods. This is a many to many relationship as the user can have a saved preference of many different neighborhoods and a neighborhood can have the preference of many different users.

2. CEN (CrimeEvents and Neighborhood)

This represents the connection between Crime Events and their respective neighborhoods. It is a many-to-one relationship, meaning each crime event is associated with only one neighborhood as its location, while a neighborhood can encompass multiple crime events.

3. CECT (CrimeEvents and CrimeTypes)

This represents the connection between crime events and crime types. This is a many-to-one relationship where each crime event has just one crime type, but a crime type can be tied to many crime events.

4. ARAC (Average Rent and Apartment Cost)

This is a relation between the average rent of an apartment in a neighborhood and average apartment cost of an apartment from the leasing company. This is a one to one relation since each apartment has one cost from the leasing company which is tied to the average rent value of an apartment in a given neighborhood.

Relational Schema

We'll structure the database using five normalized tables. Each table adheres to the principles of the 3rd Normalization Form (3NF). In simpler terms, for any nontrivial dependency $A_1, A_2, \dots, A_n \rightarrow B$ in a table, either $\{A_1, A_2, \dots, A_n\}$ must be a super-key, or B is a part of a key. This criterion is satisfied in our tables.

1. Users

Users(

 UserId VARCHAR(255) [PK],

 Email VARCHAR(255),

 Password VARCHAR(255),
)

- UserId is the primary key.
- All attributes are directly dependent on UserId.
- No transitive dependencies.

2. CrimeEvents

CrimeEvents(

Crimeld INT [PK],

Date DATE,

Time TIME,

LocationDescription VARCHAR(255)

Block VARCHAR(255)

)

- Crimeld is the primary key.
- All attributes are directly dependent on Crimeld.
- No non-prime attribute determines another non-prime attribute. Thus, no transitive dependencies.

3. CrimeTypes

CrimeTypes(

Crimeld INT [PK],

PrimaryType VARCHAR(255),

Arrested BOOLEAN

)

- Crimeld is the primary key.
- All attributes are directly dependent on Crimeld.
- No non-prime attribute determines another non-prime attribute. Thus, no transitive dependencies.

4. Neighborhoods

Users(

NeighborhoodName VARCHAR(255) [PK],

AverageRent INT,

AverageAge INT,

Demographic VARCHAR(255)

)

- NeighborhoodName is the primary key.
- All attributes depend solely on NeighborhoodName.
- No transitive dependencies

5. RenterCompany

Users(

ApartmentId INT [PK],

RenterCompanyName VARCHAR(255),

Rating INT,

Cost INT

)

- ApartmentId is the primary key.
- All attributes depend solely on ApartmentId .
- No transitive dependencies