

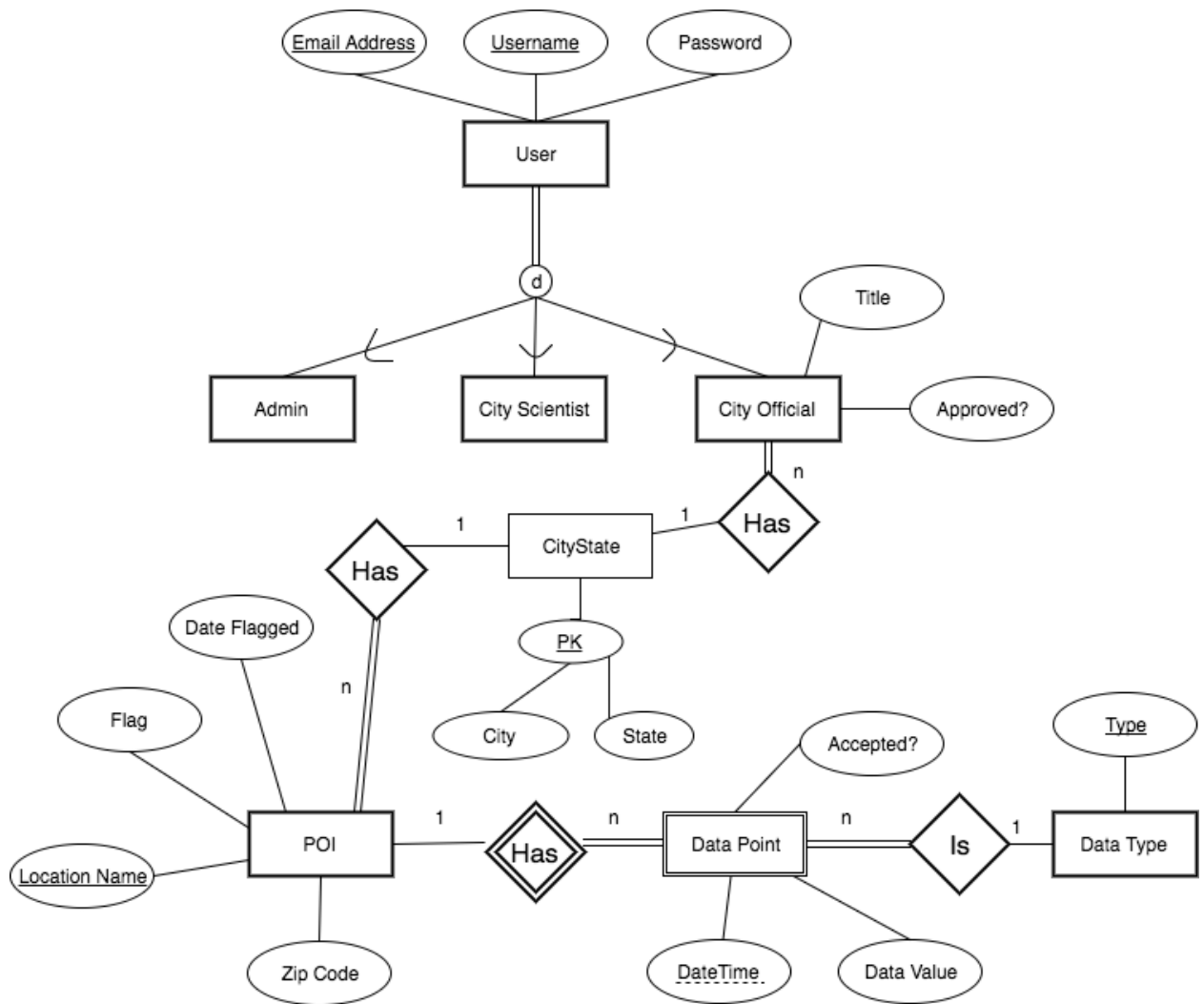


[GROUP 35]

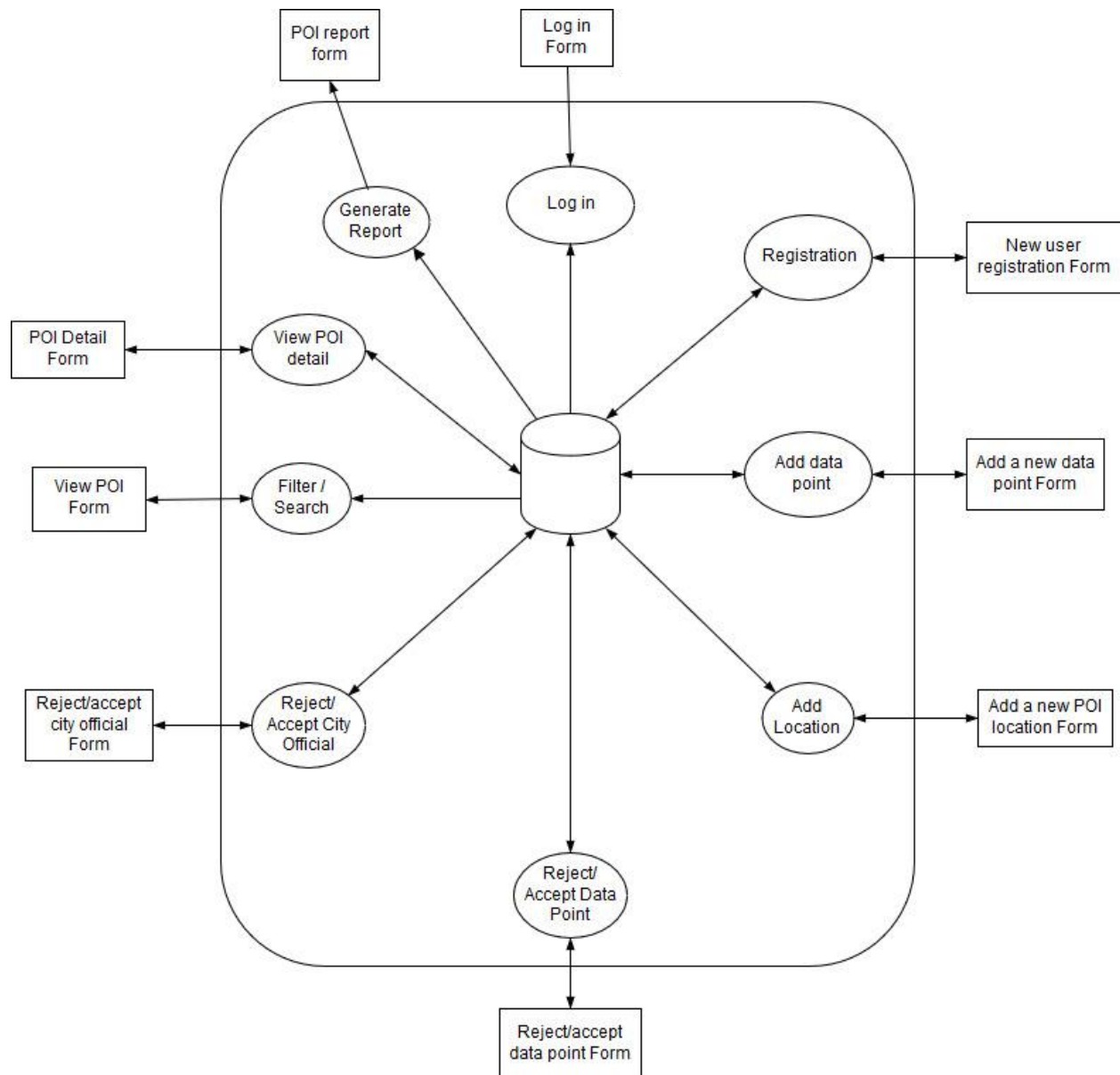
[Phase 2]

Camila Bianchi di Carcano, cbdc3, Section A
Parth Sehgal, psehgal3, Section A
Justin Ashtiani, jashtiani3, Section A
Yang Zhang, yzhang3026, Section A

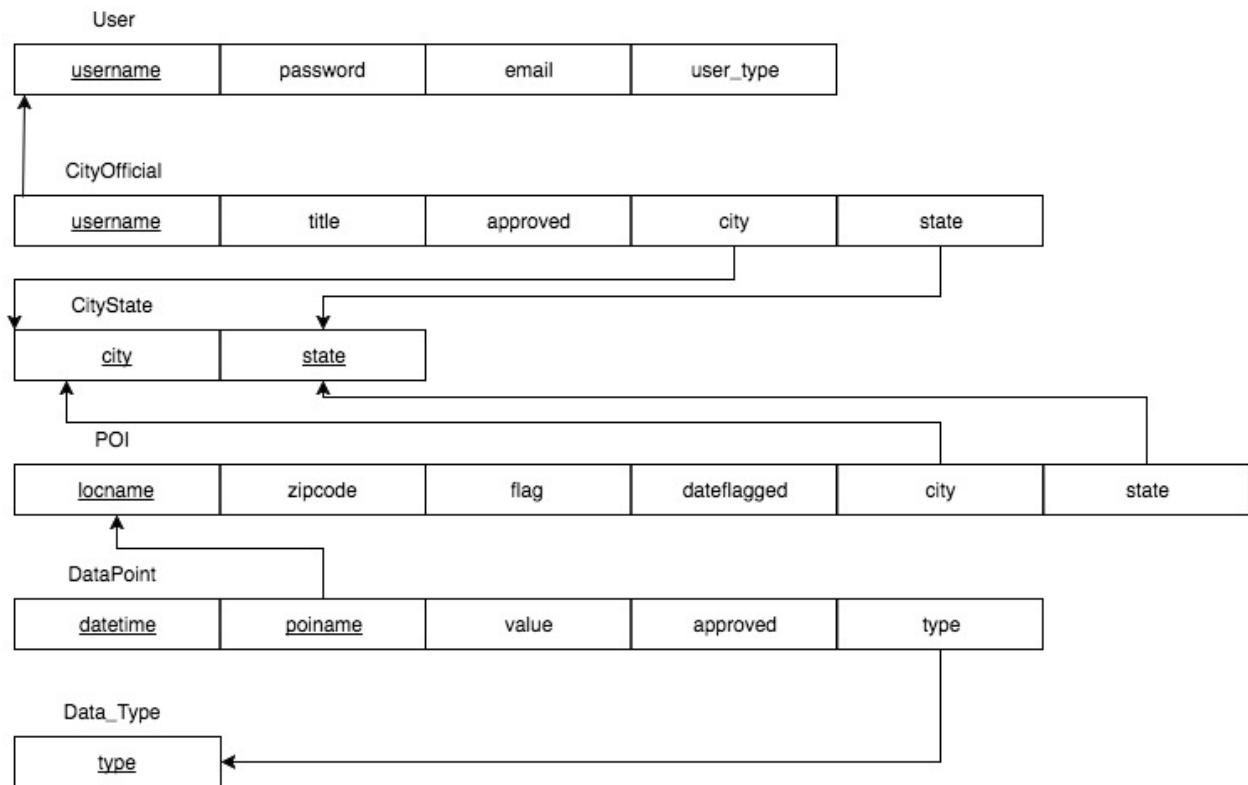
EER Diagram



Information Flow Diagram



Relational Schema Diagram



Note: We created two tables for users. User contains each type of user (city officials, city scientists, and admins). There is also a city official table (CityOfficial). When a city official is added to the user table, code in our application will add the city official to the CityOfficial table as well.

Create table Statements:

```
CREATE TABLE User(  
    username varchar(25) NOT NULL,  
    password varchar(25) NOT NULL,  
    email varchar(50) NOT NULL,  
    user_type char(20) NOT NULL,  
    CONSTRAINT check_user_type CHECK (user_type in ('city_official', 'city_scientist',  
'admin')),  
    PRIMARY KEY (username),  
    UNIQUE (email)  
)ENGINE=InnoDB;
```

```
CREATE TABLE CityState(  
    city varchar(50) NOT NULL,  
    state varchar(50) NOT NULL,  
    PRIMARY KEY (city, state)  
)Engine=InnoDB;
```

```
CREATE TABLE CityOfficial(  
    username varchar(25) NOT NULL,  
    city varchar(50) NOT NULL,  
    state varchar(50) NOT NULL,  
    title varchar(50) NOT NULL,  
    FOREIGN KEY (city, state) REFERENCES CityState(city, state)  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (username) REFERENCES User(username)  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    PRIMARY KEY (username)  
)ENGINE=InnoDB;
```

```
CREATE TABLE POI(  
    locname varchar(150) NOT NULL,  
    zipcode varchar(5) NOT NULL,  
    flag BOOLEAN DEFAULT FALSE,  
    dateflagged DATE DEFAULT NULL,  
    city varchar(50) NOT NULL,  
    state varchar(50) NOT NULL,  
    FOREIGN KEY (city, state) REFERENCES CityState(city, state)  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    PRIMARY KEY (locname)  
)ENGINE=InnoDB;
```

```
CREATE TABLE DataType(  
    type varchar(25) NOT NULL,  
    CONSTRAINT check_type CHECK (type in ("air quality", "mold")),  
    PRIMARY KEY (type)  
)Engine=InnoDB;
```

```
CREATE TABLE DataPoint(  
    datetime TIMESTAMP NOT NULL,  
    poiname varchar(150) NOT NULL,  
    value DOUBLE NOT NULL,  
    approved BOOLEAN DEFAULT FALSE,  
    type varchar(25) NOT NULL,  
    PRIMARY KEY (datetime, poiname),  
    FOREIGN KEY (type) REFERENCES DataType(type)  
        ON DELETE CASCADE    ON UPDATE CASCADE,  
    FOREIGN KEY (poiname) REFERENCES POI(locname)  
        ON DELETE CASCADE    ON UPDATE CASCADE  
)Engine=InnoDB;
```

Constraints:

- Only city scientists can add new data points.
- Only city scientists can create new POI Locations.
- Only admins can approve/reject data points.
- Only admins can approve/reject new city official accounts.
- City official accounts only need to be approved by one admin.
- Date flagged only stores the most recent date a data point was flagged.
- Only city officials can add/remove flags.
- No attribute of USER can be NULL
- No attribute of LOCATION can be NULL
- No attribute of DATA POINT can be NULL

Assumptions:

- All users must belong to exactly one of the three different types of users.
- The database doesn't need to keep track of which city scientist added a data point.
- The database doesn't need to keep track of which city scientist added a location.
- The database doesn't need to keep track of which admin approved a data point.
- The database doesn't need to keep track of which admin approved a city official.
- The database doesn't need to keep track of which city official flagged a data point.
- The only possible values for the Approved? Attribute both in city official and in data point are PENDING or APPROVED. If an admin rejects a city official or data point that record is deleted but the same record can be added again later and wait to be approved. In the case of a city official- an individual can attempt to re-register with the previously rejected email address or username. That is, once rejected data points or city officials aren't 'banned' from future registration attempts.
- Since the database stores information on locations and POI locations we assume that it must fetch this information when forms are generated that allow data points or city officials to add the city and state. And these forms are in turn used to add new data to the database (i.e. city officials or data points) we have indicated this by showing double arrows for these forms on our information flow diagram.