CS348 Project Milestone 1

Abdullah Bin Assad Chandana Sathish Lukman Mohamed
Vikram Subramanian Dhvani Patel

June 21, 2020

Application description

Application User

We are creating an interactive web app for people interested in exploring crime data in the Greater Toronto Area. People who wish to assess how safe a neighbourhood is before investing in a new property or novice drivers trying to avoid accident-prone roads or just people curious about crime rates around them can greatly benefit from our application.

Interaction with the app

A user would simply search for a query (as seen on the demo application) using the drop-down on certain words in the question to tailor the search to their needs. For example, "I want to explore bicycle theft crimes in 2019 (year) citywide on a bar chart" can be one of the many queries a user selects. Alternatively, a user can also pick from a list of predefined queries (not in the demo yet) if they are unsure about where to start. Once the user clicks "OK", the page updates to display the requested query and allows them to change the representation/visualization of the data as well as further refine their search with more options. We also plan on adding additional features as follows.

Key features

1. Filter crime/traffic events (starter question with drop-down for filtering different columns as seen in demo) and show results on a table
2. Display crime data
    - bar chart, line chart, and pie chart
    - map, heat map
3. Create predefined complex queries in the form of question and answer
    - Example - Where should I park by bike in this neighbourhood, where should I drive as a new driver, is there a correlation between education/demographic and crime?
4. Provide users with information on closest police division or which neighborhood they are situated in based on the address they provide us with

5. Report a crime
6. Interactive "How well do you know your city" feature
    - Let the user guess certain values from a query they select, then show them the actual results and tell them how close they were
    - Store user guess average and tell them how they compare against other users who tried guessing the same value

Data

The tables we get from the Toronto Police Open Data (`https://data.torontopolice.on.ca/pages/open-data`) are a little difficult to work with. Therefore, we create a data parser (in code.zip) to convert it to .csv files that fulfill our requirements. Sample data is part of the code.zip as csv files (ex. Neighbourhood.csv, CrimeEvent.csv).

System support

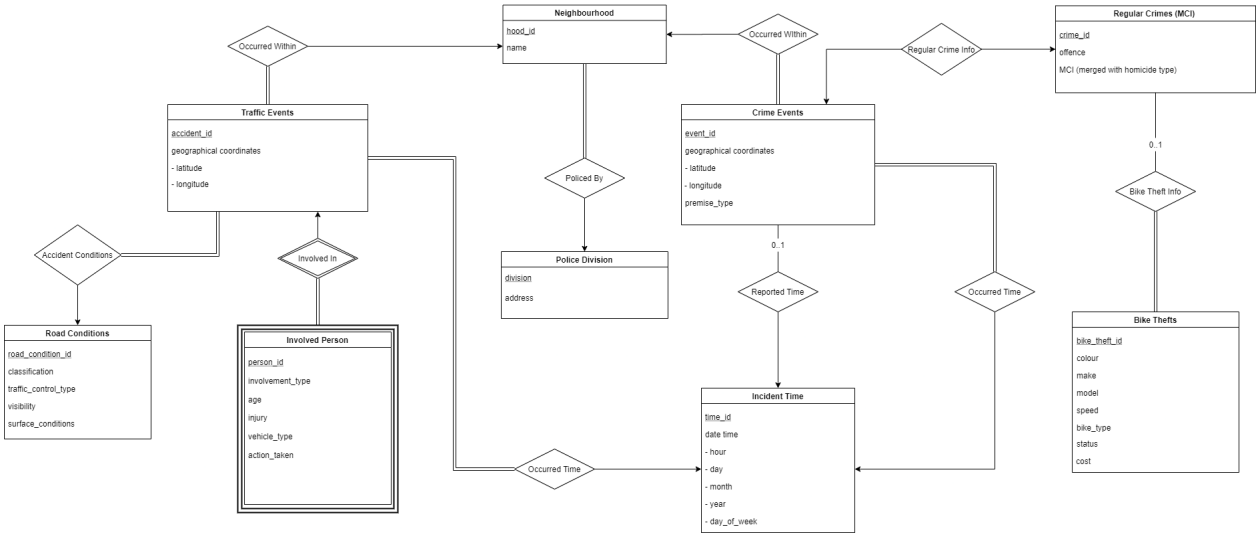
Our interface will consist of an interactive web-app. We will be using Azure to host our database and run a MySQL instance. This setup is ideal and efficient for building a web app such as ours. Azure and MySQL are modern, scalable, efficient and flexible. We will be using Node.js with an Express.js server as our back-end to interact with the database. For the front-end, we will be using the React library. Thus, our web application code mostly consists of JavaScript. Members of our group have experience with Node.js and React which gives us an excellent basis for our project.

Design database schema

List of assumptions

- all reported times of incidents are either in or after 2014
- for regular crimes and traffic accidents, the time and place of the incident are recorded accurately and not left out (not NULL mostly)
- the level and standard of reporting crimes is consistent in all neighbourhoods (required for useful analysis and comparison among neighbourhoods)
- the cause of the traffic accident is taken from testimonies of all parties involved and is not one sided (although hard to avoid survivorship bias)
- the police arrive at the scene of an accident within reasonable time to be able to assess the road conditions leading up to the cause of the accident
- all parties involved in an accident are accounted for (driver, passengers, pedestrians)
- the status of stolen bikes is updated as best as possible (most bikes are never recovered)

E/R diagram



Note: Check the standalone image of the ER diagram if this one is too small.

Relational Data Model

- IncidentTime
  - time_id
  - hour
  - day
  - month
  - year
  - day_of_week
- PoliceDivision
  - division
  - address
- Neighbourhood
  - hood_id
  - name
  - division (Foreign Key Referencing PoliceDivision(division))
- BikeTheft
  - bike_theft_id
  - colour
  - make
  - model
  - speed

  - bike_type
  - status
  - cost

- RegularCrime
  - crime_id
  - offence
  - MCI

- CrimeEvent
  - event_id
  - occurrence_time_id (Foreign Key Referencing IncidentTime(time_id))
  - reported_time_id (Foreign Key Referencing IncidentTime(time_id))
  - crime_id (Foreign Key Referencing RegularCrime(crime_id))
  - bike_theft_id (Foreign Key Referencing BikeTheft(bike_theft_id))
  - hood_id (Foreign Key Referencing Neighbourhood(hood_id))
  - latitude
  - longitude
  - premise_type

- InvolvedPerson
  - accident_id (Foreign Key Referencing TrafficEvent(accident_id))
  - person_id
  - involvement_type,
  - age
  - injury
  - vehicle_type
  - action_taken

- RoadCondition
  - road_condition_id
  - classification
  - traffic_control_type
  - visibility
  - surface_condition

- TrafficEvent
  - accident_id
  - occurrence_time_id (Foreign Key Referencing IncidentTime(time_id))

- road_condition_id (Foreign Key Referencing RoadCondition(road_condition_id))
- hood_id (Foreign Key Referencing Neighbourhood(hood_id))
- latitude
- longitude