Conceptual Sketch

Justice League team:
Fernando Herrera
Stuti Vishwabhan
Wenbo Shi
Sanja Simonovikj

Key Purposes and Social Needs

We want to design a website to increase police accountability through increased citizen oversight of crime reports. We also want to allow citizens to be actively engaged in their local community safety. Using our website citizens would be able to filter different crimes by time, location, or type and easily visualize them on an interactive map. They'll also be able to submit their own crime reports and comment on existing crime reports to share knowledge with other citizens.

Concepts

Crime

Purpose:
A crime is an event that has happened in a given neighborhood at a certain time. Crimes can be one of several predefined types, such as shoplifting, harassment, missing person etc. A crime can be sourced from an official database from the Cambridge Municipal Government or added by a user in the community. To avoid false reports, the community will moderate validity through the concept of upvoting/downvoting and commenting (explained later).
Actions:

- AddCrime
- UpdateCrime
- DeleteCrime

Structure

Crime = {crimeid[number], time, location, type[], userReported [String]}

Operational Principle:
After a user adds a crime it is added to our database. Alternatively, the crime is sourced from the official dataset and then added to our internal database. Updating the crime will update one of its fields. Deleting it will remove the crime.

User

Purpose:

A user is a member of the community using the platform to interact with the rest of the community. A user can alert and comment on crimes, as defined in those concepts. A user can be an ordinary user or a government official. Government officials have more power over ordinary users and they can perform more actions (TBD).

Structure:

user = {userid, username, password, is_official}

Actions:

Create-user
Update-username
Update-password
Delete-user
Log-in
Log-out

Operational Principal:

After creating a user, the user is added to a database.
Once an user is registered, it can log-in and interact with the application following the other concepts. After logging out, the session will end until logging in again.
A user can update its information, and then change will be recorded in the database.
If a user deletes its account, it is then unable to log-in again with that username.

Alerting

Purpose:
Allow users to stay up to date on crimes in neighborhoods of interest to them
Structure:

Set: User, Neighborhood
Relation: alerts: User -> Neighborhood

Action:
Request Alert (u: User, n: Neighborhood)
De-request Alert (u: User, n: Neighborhood)
Operational Principles:

- After requesting alert, user should see list of crime alerts for the neighborhood they requested in tab on screen
- After de-requesting alert, user should not see list of crime alerts for the neighborhood they de-requested in tab on screen

Comments

Purpose: enable user to comment on a specific crime
Structure:

Set: User, Crime
Relation: comments: User -> Crime

Actions:

comment (u: User, c: Crime, comments, a: anonymous_flag)
u.comments += (c, comments)

editcomment(u:User, c: Crime, newcomments)
u.comments = (c, newcomments)

withdrawcomment (u: User, c: Crime, comments)
u.comments -= (c, comments)

Operation Principles:

after comment (u1, c1, comment1),

observe (c1, comment1) in u1.comments

after editcomment (u1, c1, newcomment1),
observe (c1, newcomment1) in u1.comments

after withdrawcomment (u1, c1, comment1),
observe (c1, comment1) no longer in u1.comments

Neighborhood

Purpose:
A neighborhood is a local community with a given name and location.
Structure:
Neighborhood = {name, location}
Actions:
Create-neighborhood
Delete-neighborhood
Operational Principles:
After creating a neighborhood, the neighborhood is added to a database.
After deleting a neighborhood, the neighborhood is deleted from the database.

Upvoting/Downvoting

Purpose: Upvoting and downvoting crimes serve as a way to establish the veracity of a crime report. This is especially important for crimes added by users. Crimes that have been downvoted a lot such that the total score is less than 0 (upvotes-downvotes = total score), could be flagged to indicate that they might not be accurate.

Structure:
Upvote ={user, crime}
Downvote = {user, crime}
Actions:
UpvoteCrime
DownvoteCrime
Operational Principle:
After a crime is upvoted the total score/count for the crime is increased by 1, downvoting decreases it by 1. If a crime has been downvoted by many users (score < 0) then the crime will be flagged as incorrect and potentially removed by moderators or users with higher access (gov officials)

Interesting conceptual design work

We will have a map with some heat visuals for reported crimes, and a community-run platform that allows for discussion and updates to the information that this map presents. There is engagement from the community (e.g., encouraging comments on crime to support the neighborhood to mitigate crime eventually), and there is interesting information that can be presented in unique ways no other side is currently presenting.