Full Design

Justice League Team

Community Watch app

Overview

Description:
We plan to build an application called Community Watch that allows the citizens of Cambridge to stay up to date on local crimes and engage in community discussion on crimes in neighborhoods as well.

Key Purposes:

- **Provide a visual for citizens to understand crimes in neighborhoods**
  An Interactive visual provides benefits like real-time filtering, interactive hoverability for crime statistics, etc. Citizens can now better make out crime patterns in particular neighborhoods by observing the geography of it easily.
- **Provide a platform for community discussion on crimes**
  Citizens can now openly talk about crime with others in a safe space. This safe space is ensured by rules regarding flagging of posts and replies. Having the power to engage in community discussion on crime keeps the community well-informed on what community concerns are.
- **Provide a personal feed for crimes for the neighborhoods you follow**
  Users can follow certain neighborhoods to stay updated on the crimes in those neighborhoods and the community discussion relevant to those neighborhoods as well. This provides a more personal way of knowing what crimes and posts are relevant to you while filtering out the noise of other neighborhoods.

Existing Solutions:

- **Community Discussion via people you know in your community (over phone, in person, etc.)**
  While citizens in a neighborhood can chat with other neighbors they already know about crimes, their reach in understanding how local crimes affect people (including neighbors they don't know) is limited. Having a platform like our app ensures greater reach, so that anyone can freely raise a concern for a neighborhood.
- **Online forums like Reddit for community discussion**
  Online forums like Reddit are great in that the reach is unlimited; however reddit threads can get messy if the only thread that exists is for a neighborhood and crime. If citizens want to quickly learn about different crimes in different neighborhoods, they would have to jump from thread to thread and know what they're looking for, which puts burden on

the user. There's no option to explore crimes freely in their neighborhood. Additionally, the lack of visuals in forums like Reddit make it difficult to understand overall patterns of crimes in neighborhoods.

Conceptual Design

User:

Purpose:

A user is a member of the community using the platform to interact with the rest of the community.

Structure:

User attributes = [userid, username, password]

Actions:

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

Operational Principles:

After creating a user, the user is added to a database.
After deleting a user, the user is deleted from the database. All posts and replies that the user wrote will be deleted.
After updating the username, the user's username is updated in the database. All posts and replies that the user wrote under their previous username will be updated to the new one.
After updating the password, the user's password is updated in the database.
After log-in, the user is ready to perform customized operations (Post, Upvote, Downvote, Flag, Feed).
After log-out, the user can not perform customized operations (Post, Upvote, Downvote, Flag, Feed)

Neighborhood:

Purpose:

A neighborhood is a local community with a given name and location.

Structure:

Neighborhood attributes = [name, location]

Actions:

Create-neighborhood
Delete-neighborhood
*These actions refer to ones in the database only. Users in the application cannot create/ delete crimes on their own.

Operational Principles:

After creating a neighborhood, the neighborhood is added to the database.
After deleting a neighborhood, the neighborhood is deleted from the database.

Crime:

Purpose:

A crime is an event that has happened in a given neighborhood at a certain time.

Structure

Crime attributes = [crime_id, time, neighborhood_id, type]
Set:
Neighborhood = set of neighborhoods = {(nb1), (nb2), …, (nbn)}
Crime = set of crimes = {(c1), (c2), …(cn)}
Relation:
belongs: Crime -> Neighborhood = {(c1, nb1),(c2, nb2) …}

Actions:

Create-crime
Delete-crime

Operational Principles:

After creating a crime, the crime is added to the database.
After deleting a crime, the crime is deleted from the database
*These actions refer to ones in the database only. Users in the application cannot create/ delete crimes on their own.

Post: (assoc to neighborhood)

Purpose

enable user to raise comments on a neighborhood

Structure

Set:
User = set of users = {(u1), (u2), …, (un)}
Neighborhood = set of neighborhoods = {(nb1), (nb2), …, (nbn)}
Relation:
posts: User -> Neighborhood = {(u1, nb1), (u2, nb2) …}

Actions

post(u: User, nb: Neighborhood)
u.post += nb

withdrawpost(u: User, nb: Neighborhood)
u.post -= nb

Operational Principle

after post(u1, nb1)
observe nb1 in u1.posts

after withdrawpost(u1, nb1)
observe nb1 no longer in u1.posts

Reply: (assoc to Post)

Purpose

enable user to reply to a post

Structure

Set:

User = set of users = {(u1), (u2), ..., (un)}

Post = set of posts = {(p1), (p2), ..., (pn)}

Relation:

replies: User -> Post = {(u1, p1), (u2, p2) …}

Actions

reply(u: User, p: Post)

u.reply += p

withdrawreply(u: User, p: Post)

u.reply -= p

Operational Principle

after reply(u1, p1)

observe p1 in u1.replies

after withdrawreply(u1, p1)

observe p1 no longer in u1.replies

Upvote: (Post & Reply)

Purpose

enable user to like a post or a reply

Structure

Set:

User = set of users = {(u1), (u2), ..., (un)}

Post = set of posts = {(p1), (p2), ..., (pn)}

Reply = set of replies = {(r1), (r2), ..., (rn)}

Relation:

upvotestoposts: User -> Post = {(u1, p1), (u2, p2), …}

upvotestoreplies: User -> Reply = {(u1, r1), (u2, r2), ..}

Actions

upvotetopost(u: User, p: Post)

u.upvotestoposts += p

withdrawupvotetopost(u: User, p: Post)

u.upvotestoposts -= p

upvotetoreply(u: User, r: Reply)

u.upvotestoreplies += r

withdrawupvotetoreply(u: User, r: Reply)

u.upvotestoreplies -= r

Operational Principle

after upvotetopost(u1, p1)
observe p1 in u1.upvotestoposts

after withdrawupvotetopost(u1, p1)
observe p1 no longer in u1.upvotestoposts

after upvotetoreply(u1, r1)
observe r1 in u1.upvotestoreplies

after withdrawupvotetoreply(u1, r1)
observe r1 no longer in u1.upvotestoreplies

Downvote: (Post & Reply)

Purpose

enable user to dislike a post or a reply

Structure

Set:
User = set of users = {(u1), (u2), ..., (un)}
Post = set of posts = {(p1), (p2), ..., (pn)}
Reply = set of replies = {(r1), (r2), ..., (rn)}
Relation:
downvotestoposts: User -> Post = {(u1, p1), (u2, p2), …}
downvotestoreplies: User -> Reply = {(u1, r1), (u2, r2), ..}

Actions

downvotetopost(u: User, p: Post)
u.downvotestoposts += p

withdrawdownvotetopost(u: User, p: Post)
u.downvotestoposts -= p

downvotetoreply(u: User, r: Reply)
u.downvotestoreplies += r

withdrawdownvotetoreply(u: User, r: Reply)
u.downvotestoreplies -= r

Operational Principle

after downvotetopost(u1, p1)
observe p1 in u1.downvotestoposts

after withdrawdownvotetopost(u1, p1)
observe p1 no longer in u1.downvotestoposts

after downvotetoreply(u1, r1)
observe r1 in u1.downvotestoreplies

after withdrawdownvotetoreply(u1, r1)
observe r1 no longer in u1.downvotestoreplies

Flag: (Post & Reply)

Purpose

enable user to flag a post or a reply for inappropriate languages

Structure

Set:
User = set of users = {(u1), (u2), ..., (un)}
Post = set of posts = {(p1), (p2), ..., (pn)}
Reply = set of replies = {(r1), (r2), ..., (rn)}
Relation:
flagstoposts: User -> Post = {(u1, p1), (u2, p2), …}
flagstoreplies: User -> Reply = {(u1, r1), (u2, r2), ..}

Actions

flagtopost(u: User, p: Post)
u.flagstoposts += p

withdrawflagtopost(u: User, p: Post)
u.flagstoposts -= p

flagtoreply(u: User, r: Reply)
u.flagstoreplies += r

withdrawflagtoreply(u: User, r: Reply)
u.flagstoreplies -= r

Operational Principle

after flagtopost(u1, p1)
observe p1 in u1.flagstoposts

after withdrawflagtopost(u1, p1)
observe p1 no longer in u1.flagstoposts

after flagtoreply(u1, r1)
observe r1 in u1.flagstoreplies

after withdrawflagtoreply(u1, r1)

observe r1 no longer in u1.flagstoreplies

Feed: (assoc to neighborhood/**crime**/user)

Purpose

enable user to follow the crime status of a neighborhood

Structure

Set:

User = set of users = {(u1), (u2), ..., (un)}

Neighborhood = set of neighborhoods = {(nb1), (nb2), ..., (nbn)}

Relation:

feeds: User -> Neighborhood = {(u1, nb1), (u2, nb2) …}

Actions

feed(u: User, nb: Neighborhood)

u.feeds += nb

unfeed(u: User, nb: Neighborhood)

u.feeds -= nb

Operational Principle

after feed(u1, nb1)

observe nb1 in u1.feeds

after unfeed(u1, nb1)

observe nb1 no longer in u1.feeds

Wireframes

https://app.moqups.com/ckK6rVvycl/view/page/a1971e51e

Design Commentary

1. **Separating Feed onto a separate tab vs having it on home page**
   We decided to put the personalized feed with posts/crimes in the neighborhoods a user follows in a new tab, as opposed to keeping it on the home page. This design choice was preceded by a thorough discussion related to how the user interface should look. Some team members felt that it's more intuitive to have the personalized feed in the homepage upon logging in (as done by many social media apps), while others argued that this would cause too much clutter and would cause confusion considering the fact

that we also want to show the list of crimes and posts that are filtered and shown on the map. While we realized there are pros/cons to each choice, we found a middle ground by having a special view of the homepage which only shows posts/crimes for the neighborhoods a user follows, and then have a generic “exploration” view which contains the filters to show only crimes and posts that satisfy the selection criteria. This choice was made partially due to the fact that Vue.js can dynamically show content (related to the Single Application Page paradigm).

2. **Reframing the concept of Alerting as a concept of a Feed**
   The initial concept of alerting consisted of users being somehow notified of new crimes in the neighborhood they are concerned about (by “following” the neighborhood). This would have required some sort of notification/alert system in place, where a user would acknowledge they have received the alert after which the alert would be obsolete. After some thorough consideration of this and other related concepts and after introducing the concept of Posts we realized it would be better to reframe the Alerting concept as a Feed, where we would be able to get a feed for both crimes and posts for the neighborhoods we are interested in. This not only simplifies the design, but it allows us to bridge the concepts of Crime and Posts in a meaningful manner that also nicely relates to the visualization portion of our design (Crimes/Posts shown for the filtered items on the dynamic map in the “explore” view and crimes/posts shown for the neighborhoods a user follows on the static map in the “feed” view).

3. **Allowing users to discuss topics on a neighbourhood level as opposed to a crime level**
   Initially we thought it would be nice if users can comment on particular crimes as that would allow for more transparency and it might even help the crime case to be resolved easier. However, after a more thorough discussion we realized this constrains the users to talk about specific topics only and it also opens the doors for controversy. Instead we decided to introduce the concept of a Post, which would be associated with a neighborhood at a higher level. This allows users to have more flexibility on what to discuss and simplifies the complex ethical issues associated with crime commenting.

4. **Users cannot edit their posts and replies**
   We initially thought that users’ posts and replies could be changed, but ultimately decided against it due to the complications it introduces. For example, if a user writes a post, other users can reply to the post with their own thoughts. But, if the original poster changes the content of their post later, then all the replies to the original post become void. The number of downvotes/ upvotes/ flags should also become void. This introduces complications since we would like individuals’ opinions to be heard through the replies, but we want to preserve the original context since it provides authenticity to the opinions. Instead, if we don’t allow users to edit their posts and replies (only delete them), then we force the user to carefully consider the content they are posting before they post it.

Ethics Protocol Analysis

1. Envision possible futures

Possible future #1

We release the Community Watch app. We consistently update the database with the newest crimes in the Cambridge area and with enough marketing, the app becomes popular among the residents. They actively use the app to become informed of the latest crimes, they monitor the safety in the neighborhoods they are interested in and they stay engaged with the community by opening threads about various crime-related issues. Due to widespread community engagements, unsolved crimes are resolved faster, crime rates become lower in neighborhoods with higher community engagement, and the residents use our platform to organize themselves to make meaningful changes to improve the safety of the community.

Possible future #2

We release the Community Watch app. At first there is a hype around it and many residents sign up, but bootstrapping engagement from the community proves to be difficult, especially as users are not sure if there would be any legal/criminal consequences of them sharing their crime-related opinions/claims publicly. Eventually users forget the app exists and they continue engaging with other, more generic apps to organize themselves and stay up to date with general news (Facebook, Twitter, Reddit etc).

Possible future #3

We release the Community Watch app. Many residents don't hear about the app at all and the app becomes popular only when a big crime happens and activists need a platform to discuss their ideas. They become aware of our app and decide to promote it and use it for community engagement. However, there is a lot of controversy about the big crime that initiated the engagement and users start spreading false information and act in ways against our implied code of conduct. Our mechanisms to deal with false information are not developed and robust enough, and as a consequence, many innocent people get defamed or associated with crimes they have little to no relation to. The app is taken down by the authorities as a dangerous platform that does not satisfy some strict and obscure laws.

2. Identify stakeholders
    - Residents of Cambridge
    - The local government
    - Criminals
    - The Police
    - People in power
    - Minority groups

- Black people
- City newcomers
- Political activists
- The media
- Business owners

3. Identify values at play
    - **Outcome lens** - in what ways our design makes it better or worse for certain stakeholders
    - **Process lens** - how did the process treat stakeholders?
    - **Structure Lens** - how are the outcomes distributed among different stakeholders? What are the differences in how different stakeholders were treated by the process?
4. Identify value-laden design-choices

Possible choice	**Values promoted**	**Values demoted**
Users can add/report crimes themselves	*Structure lens*: users get power to share crimes that might not be reported to the police	*Outcome lens*: bad actors can prank the system and report non-existent crimes
Users can comment on specific crimes	*Process lens:* users get to contribute to the discussion related to a given crime and might even give important details that could help the authorities	*Outcome lens:* users can spread false information, defame or cause harm to innocent stakeholders and hinder potential judicial processes
User's posts or replies with many downvotes are hidden	*Outcome lens:* users have easy access only to content that is considered truthful according the crowdsourced moderation mechanism	*Structure lens:* unpopular opinions or opinions shared by a minority group are more likely to be censored even if they are truthful and useful to the discussion

5. Choose and Justify
    a. <u>How are crimes identified?</u>
    Choice: Crimes are added by periodically updating the app's database from the official city of Cambridge crime database; users cannot report crimes themselves
    Justification: The cons outweigh the pros of allowing users to report crimes themselves. If a crime is serious/real a user can actually go and report it in the police, after which it will eventually land in our database. However, if a user falsely reports a crime it might cause unnecessary panic, cause harm to innocent stakeholders and the users could potentially face legal consequences.

b. How can users engage with the community?
   Choice: Users can write posts/comments for a particular neighbourhood as opposed to a specific crime.
   Justification: Allowing users to write posts on a neighbourhood-level as opposed to crime-level gives more flexibility as of what can be discussed on our platform and potentially avoids conspiracies as users are not implicitly forced to discuss specific crimes but crime-related topics for a given neighbourhood.
c. How do we ensure our platform creates a safe, transparent and authentic discussion space?
   Choice: Hide comments/posts that are downvoted by many users and allow users to flag certain content as inappropriate.
   Justification: Hidden content will still be available for users to view, but they would be aware that the content might be fraudulent as they would need to explicitly click to uncover it. This helps in mitigating the spread of false information but it also prevents complete censorship. Flagging content as inappropriate is reserved for cases where an inappropriate language is used and users would be aware of content that is flagged as inappropriate and it would be at their discretion to read it or not.

6. Build

   N/A

Updates for the MVP

Since our proof-of concept submission we have not made any significant design decisions changes, besides not allowing users to edit their posts (we allow them to delete only), which was done for security and authenticity reasons. We have updated our design decisions section to reflect this. Additionally, we decided to use markers on the map to showcase crimes, as opposed to creating a heatmap to show the density of crimes in areas. This was not only easier for implementation, but also gives the user the flexibility to hover over a crime to see basic info about it, while still being able to infer the density of crimes based on the density of markers.

Usability heuristics

#1: Visibility of system status

The user has access to informative messages after performing certain actions (ex. Sign-in, sign-up) and also has access to a navigation bar that helps the user navigate easily. The

informative titles of the individual views help the user figure out where in the system they are. Additionally the user can figure out which tab they are looking at (Crimes or Posts) as it will be highlighted.

#2: Match between system and the real world

We use clear and common English keywords to inform the user about the action that will happen once they interact/click a certain place/button. For example words like follow, post, sign in, sign up, delete etc should be very intuitive and understandable for the average user.

#3: User control and freedom

In many cases users can easily recover after they do something wrong. As mentioned, the navigation bar allows them to access any page from any other page. As an additional flexibility users can change their username and password at any time. They can delete their posts after posting and unfollow neighborhoods they have followed. Some actions are not reversible, such as the “delete account” action, although this is clearly marked with a red button as a danger zone.

#4: Consistency and standards

We use the same keywords and icons throughout our design, and we do not have very similar icons that represent a different thing. We follow the conventions by using similar naming and icons like those of other social media. For example we use the typical keywords “sign in”, “sign up”, “follow”, “unfollow”, “upvote”, “edit”, “delete”, “refreet” (similar to retweet, repost etc). The same holds for the icons, for example, to get to your profile you would have to click on an icon that has a small human-like avatar on it, just like other social media sites.

#5: Error prevention

We warn users at places where the action has potentially severe consequences, such as deleting an account. We do not display actions that can be performed only if a use is logged in, if the user isn't logged in, for example following/unfollowing, posting etc. We set reasonable defaults, such as the map filters being set to show “all” types of crimes and neighborhoods.

#6: Recognition rather than recall

We used dropdown menus for crime type option and neighborhood option when the user wants to set filters on crimes. With the dropdown menu with a full list of available options, users don't need to memorize any information on those two critical and categorical fields.

#7: Flexibility and efficiency of use

The crimes are both displayed in the form of markers in the Map and as list items. This flexibility allows users to review from two perspectives whichever is comfortable for the users.
In addition, we have a "My Neighborhood" tab, which provides personalization by tailor contents (only crimes of the following neighborhood) and functionality for individual users.

#8: Aesthetic and minimalist design

Our app has the content and visual design only focused on the key concepts (crime report, crime alert, and crime comments) of the app. One map to visualize the crime cases, One tab for allowing users to post comments on certain neighborhoods. No unnecessary UI component to distract users.

#9: Help users recognize, diagnose, and recover from errors

We used error messages to guide users to log in. For example, certain functionalities (follow/unfollow a neighborhood, write a post on certain neighborhood) requires the user to log in first. In those cases, either error messages (e.g., not sign in yet) are used or corresponding buttons (e.g., post comment button) are hidden to help users recognize the misoperation.

#10: Help and documentation

We don't provide a specific documentation, listing the concrete steps to that needs to be carried out. However, due to the simplicity of how we presenting the key concepts (filter for crimes in the Map, Post Tab) in the app, matching between system and real world, consistency, appropriate error message, we think the setting of the app should be easy for user to learn quickly and efficiently.