Feature: Neighborhood Crime Alert
Developer: David Chan
Date Submitted: April 15th, 2023
Reviewer: Sierra Harris
Review Date: April 16th, 2023

Major Positives and Major Negatives

Top Qualities

- The design shows input validation at every layer per the HLD
- The design correctly follows the flow of data as describe in the High level design and includes all layers.
- The frontend design includes all necessary input boxes for a crime alert per the BRD.

Top Flaws

- The design does not show how crimes alerts over 25 miles from the users location would be filtered out.
- The design does not show nor include the necessary objects to carry out success cases such as the CrimeAlert object and Result object.
- It is unclear what is in the crime alert database and what is selected to perform success case because of the use of select *.
- The design does not show how a alert owner will be validated when they try to delete a alert, how a crime alert will be deleted or updated. The design does not include a SQL query to show this process.

Unmet Requirements

- The frontend nor backend design includes the character count limitations per the BRD such as the title limit of 50 characters and the description limit of 150 characters.
- The design does not show the failure case of how the system would behave if the the system allows a user who was not the creator to edit the post.

- The design does not show the business rule of crime alerts being automatically deleted after 24 hours of posting per the BRD.

Design Recommendations

I recommend checking if the alert owner matches the user that wants to delete an alert at the entry point validation function. Since the crime alert object is already in view the crime alert object owner property should be to. You can use this to see if the usernames match. This would keep the system from unnecessarily going to the data store and improve performance.

I also recommended adding the SQL queries to the design so other developers know what exactly is the query that the data access object is carrying out when it reaches the data access layer.

I also recommend including how you plan to send an email notification into the design instead of implicitly stating Send Email Notification". How is this process carried out and how would the system behave for a failed or successful attempt

Lastly, I recommend adding a models section to show what the crime alert object looks like and help determine what should be in the database.

Test Recommendations

One recommendation is to create a front-end end-to-end test that shows the fields of the crime alert being filled with correct and required information and a failure test for the wrong and incomplete crime alert to ensure that:

1. Users are only allowed to enter the correct values aligning with the BRD such as an absolute location and time on the format "00:00" and,
2. Users cant leave required fields blank and,
3. User can not input any more characters past the limits set in the BRD

I also recommend integration test to test the DAO and database to ensure the correct information is being pulled from the database for all the queries.
Lastly, I recommend a unit test for filtering out crime alerts that are over 25 miles away that were pulled from the database.

Feedback

Ultimately the design aligns with our High-level design, each layer is correctly displayed and the flow of data aligns as well. The design also covers the majority of the functional and non-functional requirements from the BRD and coding standards. However, I think other developers would have questions regarding the process of viewing, editing and deleting a crime alert, since part of processes are not explicitly designed. For instance, the design does not show the what columns grabbed from the database, the crime alert object model, the filtering function and is missing SQL queries. Furthermore, I dont believe that the design would work if new requirements were introduced because there is a lot not explicitly stated or designed.