Sailaja Nallacheruvu, Calvin Wu, Matthew Chung, Aghalya Narayanan
May 3 2023

CS 411 Final Project – Stage 6

1. *Please list out changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).*

   Our project has two major changes from what was originally in our project proposal. The first is that one of our intended functionalities was giving the user the ability to filter through and view data from the database by crime. However, due to the nature of the database being derived from police reports, the values in the database show up with their police crime codes. Since these codes and descriptions are inaccessible to the average user since the crime names are not relatable to users, we scrapped this functionality. Instead, we allowed the users to filter by crime investigations that are currently ongoing, the number of crimes in an area, and by neighborhood.

   Another large change different from our functionality was allowing the user to insert and delete crimes from the database. This is a functionality we added for users so that the database can stay current and relevant to users. Though this was not in our original plan, it allows for more user interaction and ensures the users are aware of the crimes occurring in modern times, making it more useful for users throughout time.

2. *Discuss what you think your application achieved or failed to achieve regarding its usefulness.*

   I believe that our application is useful to users outside of the constraints of this class. The goal of our application, as shown by our project proposal, was to allow users to see which neighborhoods in LA were safe with data available to users to back it up. We accomplished this by allowing users to see the crimes that have occurred in any neighborhood they wish based on user input. We also allow users to set a certain value as a high number of crimes and see which neighborhoods and districts within neighborhoods have too many crimes occurring in them. Also, users can see all the open crimes in the city of LA at any point through our other advanced query. Finally, we allow users to report crimes on their own to our site. This allows users the autonomy to have real time accounting for crimes that are occurring rather than having to wait for crimes to be updated to the parent table of this site. That functionality keeps the app modern, as new crimes can always be updated, allowing users to see what areas in LA are safe and unsafe. As a result of the ability we've provided users to see different types of breakdowns of the crimes occurring, and to add a record of crimes they know have occurred, we allow all users to have accurate data to determine the safety and crime rates of neighborhoods and districts within LA. Thus, our goal was met, making this application useful to users interested in LA.

3. *Discuss if you changed the schema or source of the data for your application.*

   Throughout our project, we did not change the source for our application but we did make adjustments to our schema along the way from the feedback received from our ta. In our original schema, we had some mistakes that were pointed out such as our VictimInfo table having attributes of the report, and the primary keys for some of our tables being wrong. One change that we made was that in our location table, instead of having a primary key DistrictId and a foreign key Address, we changed it to have just one primary key being the address and supplemental attributes including DistrictId, Latitude and Longitude. For the Crime table, we changed the table name from CrimeType to just Crime and added an additional attribute called severity. In the VictimInfo table, instead of having the ReportId being a primary key and foreign key, we made a new primary key called VictimId and made that into the primary key. The last change we made was that we removed the "occurred in" relationship between the AreaInLa table and the Report table.

4. *Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?*

   In order to make our database design more realistic and match up better with the needs of the project, we made the following changes. One of the issues our TA found with our database design was that the VictimInfo table solely contained attributes of the Report Table because the primary keys were all also foreign keys. To solve this issue, we created a unique victimId for each unique victim in the table and stored that in the table as a primary key. This allows us to connect each victim to the crime that occurred to them through the report that was made about that crime. The reason victim and report information are stored in 2 separate tables is to store the information about the people, and the logistical information about the crime while keeping the DB organized. To fix the issue with the primary keys in the Location database, we made the foreign key in the table the primary key of the AreaInLA table, rather than just an attribute. Finally, to make the Crime table have more purpose within the dataset, we added another attribute in the table to represent Severity. Severity will range on a scale from 1-4 where 1 represents a crime of low severity and 4 represents a crime of high severity. Though we tried to find real life datasets that stored information about the severity of different types of crimes, this was hard to accomplish. So, we will be using auto generated data for this attribute for now. In a real life setting, we would compile this information through more hands-on research.

5. *Discuss what functionalities you added or removed. Why?*

   We implemented the "Report Crime" functionality on our website not only to meet the CRUD requirements of our database, but also for its numerous benefits. Firstly, it allows users to actively participate in making their community safer by reporting any crimes they witness or experience. This information can help law enforcement agencies identify crime hotspots and allocate resources accordingly. Furthermore, it can provide a more comprehensive and accurate view of crime data in a particular area. Official crime statistics are often based on reported crimes, but many incidents go unreported for various reasons. By enabling users to report crimes directly on our website, we can obtain a more complete picture of crime in the area. Finally, this functionality can help to build trust between the community and law enforcement agencies. By providing a direct channel for community members to report crimes, it can foster a sense of cooperation and collaboration in keeping the community safe.

Besides, we added various search functionalities, including "Neighborhood Search", "Number of ongoing crimes by crime type", "District safety index", and "Crime rate comparison". This can improve the user experience by allowing users to quickly find the specific information they need without having to sift through large amounts of data manually. Categorizing crimes by type and providing a comparison of crime rates can also help users gain a better understanding of the crime landscape in a particular area, making it easier for them to make informed decisions.

We decided to remove the functionality of providing detailed crime information on our website. This is because our website aims to provide users with an overview of the overall crime situation in LA and help them make informed decisions about where to live or visit. Detailed crime information is not necessary in achieving this goal. Additionally, obtaining all the necessary details and information about a crime is challenging as they can only be found at the police station, which we do not have access to.

6. *Explain how you think your advanced database programs complement your application.*

Our SafeLA is complemented by advanced queries that can provide users with specific and detailed information about the crime situation in LA. For instance, we implemented an advanced query that shows the total number of ongoing crime investigations categorized by type. This information can help users gain a better understanding of the types of crimes that are prevalent in certain areas, which can be useful for making informed decisions about where to live or visit. Additionally, we have a query that shows the AreaName and DistrictId where more than N crimes have occurred in a district. This can be particularly useful for those who are concerned about the safety of a specific district, as they can quickly find out the number of crimes that have occurred in that area.

In addition to the advanced queries we implemented in our SafeLA, we also added a trigger that allows users to add new addresses into the location database when reporting a crime. This enhances the user experience by streamlining the process of reporting crimes and adding new locations to the database. Additionally, we implemented a stored procedure that compares the severity of crimes in LA. Overall, these advanced database programs provide our users with a more customized and efficient experience and help them make informed decisions about their safety.

7. *Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.*
   Sailaja – One technical challenge I faced was connecting multiple get (read of CRUD) requests to the same handle_request function in the backend. For both of our advanced queries, and the home page, we were outputting information from the database based on user input. Since all of these were "get" requests that used different values for user inputs (the home page required a neighborhood input and one of the queries required a value, for example), I had to figure out how to manage all of these in one request. To overcome this problem, I manually assigned dummy values to variables that were irrelevant to some of the functions of our page. This allowed each separate get request to output the necessary values based on each page's function.

   Calvin – One technical challenge I met was making connections between our react frontend and backend using python and axios. Since I didn't have any experience before, I took a lot of time to search tutorials and videos to learn this. Also when I was writing the backend, I struggled dealing with the cross-origin resource sharing (CORS), which is a security mechanism used by web browsers to restrict web pages from accessing resources that are located outside their domain. Always remember to add this line to allow the access, `response.headers.add ('Access-Control-Allow-Origin', '*').`

   Matthew – One technical challenge I faced was when creating the stored procedure. Inside the loop, I forgot to add the case to end the loop once the cursor has finished so this created a large table inside our database. This used up a lot of memory inside our gcp and if we hadn't noticed it, it might have caused our gcp to use up more credits than we had available.

   Aghalya – One technical challenge I faced was when working on the user interface of the application. I was struggling to help connect the SQL backend to the frontend using a Python flask application. By reading about how flask applications worked and specific endpoint requests using the official Python documentation (and GeeksforGeeks), I was able to help implement this.

8. *Are there other things that changed comparing the final application with the original proposal?*

   As we discussed in the first question for this reflection, in comparing the final application with the original proposal, we identified two major changes in our project. For example, one significant alteration involves the functionality for filtering and viewing data from the database. In the original plan, we intended to provide users with the ability to filter data by crime type. However, we realized that the database contained police crime codes that are not relatable to the average user, making the original plan impractical. Therefore, we modified this functionality to allow users to filter data by ongoing crime investigations, the number of crimes in a specific area, and by neighborhood.

   Additionally, another significant change we made to the original proposal was to allow users to insert and delete crimes from the database. Although this functionality was not initially planned, we added it to encourage more user interaction and ensure the database remains relevant to users. By enabling users to insert and delete crimes, they can keep the database up-to-date with the latest crime incidents, making it more useful over time. This change also enhances the user experience by providing a more engaging platform that enables them to contribute to the database.

   Finally, because of these changes, we altered our user interface significantly from the original plan. We originally decided to have three pages. The home page would just have a search bar and a possible implementation of the Google Maps API to allow users to interact with and view the neighborhoods they were investigating. The following page would have crime information about the actual neighborhood that the user searched for, and the last page would have additional information about the weapon used and the crime itself. However, to account for adding more SQL features, such as the user being able to update, delete, or insert new crime information, we needed to remodel our user interface application, having four pages at completion of the project.

9. *Describe future work that you think, other than the interface, that the application can improve on.*

   There are a few things that the application can improve on besides the organization of the user interface. First, we only utilize one database for the application, so with more time, we could implement more databases to provide a greater amount of information for users to have access to. This would allow users to be able to discover a more cohesive amount of information, as well, so that they can learn as much as possible about different crimes and their reports in all the various regions in Los Angeles.

Another improvement that the application can implement is further datasets from other cities, rather than having information for simply Los Angeles. If the application is more widespread in the locations the databases provide information for, users from areas outside of Los Angeles would also be able to find use out of the application. For instance, if we implemented a database from Urbana-Champaign, this would be highly useful for UIUC students, as they would be able to view regions of Urbana-Champaign with high crime rates in order to be able to avoid these if possible. They also would be able to update the application by inserting their own reports of crimes in real time, which would allow other students and residents to be more aware of crimes occurring in their neighborhoods and maintain track of them.

Also, the application can be improved in terms of our implementation of the Google Maps API. We unfortunately did not have the time to spend, but the interaction with the user and application would have been better had we been able to allow users to view the specific neighborhood that they searched for immediately. Currently, we have a map of Los Angeles that allows users to zoom in on neighborhoods and view them in satellite or terrain mode, which is helpful for them to view exactly the region that they want to search for. However, this can be improved by having the map zoom in by itself onto the neighborhood the user searches for, which would allow them to view where it is in relation to other neighborhoods without them having to zoom in and search for a neighborhood themselves.

10. *Describe the final division of labor and how well you managed teamwork.*

In general, our team managed division of labor well, especially considering we began this project not knowing each other, our work styles, or our creative preferences at all. We were able to divide the work depending on each of our strengths, and we also worked as a group communally on many aspects of the project. For instance, Aghalya and Sailaja were most proficient in front end programming, so they focused on organizing the user interface of the application. Sailaja also worked on the advanced queries to make them more user interactive. Calvin and Matthew shared a large portion of implementing the SQL queries, including the update, insert, delete, and trigger processes. Considering we did not know each other going into this project, we communicated incredibly well throughout the project, ensuring that everyone was on the same page about our goals for the application and agreeing about the creative and technical decisions that we needed to make. We also utilized both Zoom and WhatsApp to regularly communicate our ideas, divide the aspects of the project, and organize meeting times to verbally discuss issues or decisions to be made.