

Project Reflection Report

CS 411

Team 077 RandomName

Yu-Ju Chang
Rui Zhou
Nidhi Chaudhari
Aniket Gargya

Project Video Demo:

<https://drive.google.com/file/d/1I2Ql8E2hizHX45TlJPf83ytfp49HpZBL/view?usp=sharing>

In terms of changes in our project from the original proposal in Stage 1, we were able to achieve most of our initial proposed goals and our project looks very similar to the way we described it in our proposal. We had mentioned in our proposal that our website will support “users being able to see which locations in the city would be particularly unsafe to visit at a particular time, being able to report crimes, and being able to search up any crime in LA from the last few years”, and our project supports all this. We had also mentioned that “Other than this our website offers Police officials to create their profile and handle information pertaining to a case”. This is one change that we did not implement. There was no “change of direction” as such, but more of a time issue due to which we decided to not implement this feature.

What our application successfully achieved in terms of usefulness : civilians can use our app to report a crime if they encounter one to help the police immediately learn the basic information like the location, the victim, and the weapon used by the perpetrator. Users can also decide where in the city they would feel safer living based on where we see more or less crimes occurring. (by referring to the UI map)

This application would also be useful to the police because they can decide which areas would require increased patrolling, and it can also help them keep track of all crimes occurring across the city. Based on our proposal, we seem to have achieved all use cases that we touched upon.

Regarding the schema, we did change many columns’ variable types. For one, there was a “Date & Time” type in the original data, which we changed to “VARCHAR” in our database because the format in the original data was not supported by MySQL. Time, area code, crime code, age, etc. were stored as “Plain Text” in the original data; we changed the data type to “INT” for our schema so that non-numeric characters could not be inserted. Lastly, longitude and latitude had a “Number” type which we changed to “REAL” to store floating/double in our schema. Aside from that, we broke the source table into multiple small tables, which was described in our Stage 2 ER diagram, so the application did not have to touch the large source table every time.

To convert the ER diagram to our final implemented table, we did not change the ER diagram too much. The only thing we altered was the primary keys for each table. We needed to add more columns to some tables to be able to make primary keys. For instance, we originally used “Longitude” and “Latitude” as the primary key for the Location table. However, we later realized that “Longitude” and “Latitude” alone cannot uniquely identify each entry because there were different area codes associated with the same “Longitude” and “Latitude”, which was due to police reporting’s logistical issues. A more suitable design, in this case, would be to prune the data that was corrupted by human errors. In the end, we did not choose to do so because the

information regarding certain codes was nowhere to be found. We figured that we shouldn't randomly pick a code to use since it would further corrupt the data.

We added basic CRUD functionalities and the advanced queries. For CRUD, we added them because our application is one that can benefit the police which makes their job more efficient. As a result, with our app, the police can easily add a crime to the database, remove a crime from the database, update a crime from the database and delete a crime from the database. Regarding advanced queries, we added queries that help the police to find out how many crimes happened in one place. In this way, the police can allocate their forces based on local safety situations. Also, we create a query to help the police find that crime that involves strong armor which needs to be solved immediately.

For advanced database programs, we create a trigger that prevents adding crimes that are located outside LA, which helps clean irrelevant data from the LA crime database when adding new crime to the LA crime database. We also created a procedure to help the police. The procedure helps lay out all the crimes that are urgent and more urgent. The urgent crimes are the crimes that involve severe violence while more urgent crimes are the ones still in investigation. In this way, the police will know which crime they need to solve first.

The next four paragraphs describe the technical challenges we each encountered.

Yu-Ju: One main challenge would be using Google's GCP. Although educational credits were offered to everyone, the credit ran out quickly even when the database was not queried. One might argue that there should be enough credits because there were four people in a group, and each had \$50 credits. However, this was not the case. At the beginning of the semester, when the project was not started, almost everyone redeemed the credit to their own account. This posed a challenge because a reasonable way to manage a group project was to keep only one database in a certain group member's account. After some investigation, I believed that each member could be added to the same project. Then from their account, they could choose to apply the educational credit to that certain project. Another tip for future students who are doing this project is to only let one person in the group, e.g. the captain, redeem the coupon, and all other members should provide him with their coupons.

Rui Zhou: One main challenge is uploading the database to GCP. When we first started the project, we found that we could not directly upload the database to GCP because the database has many NULL values and undefined values. However, we had already created sub-tables based on the main database. As a result, we took a long time to define new sub-tables based on modified main database, which is low-efficient and time consuming. So, for future students, my advice is they need to review the main database and appropriately modify it when they first have it. This could make the whole project moving smoothly.

Aniket Gargya: One challenge was that it was initially difficult for us to debug our application, as we did not have easy access to all the data through our application, because we were still in the development process. To overcome this hurdle, we created a manual query running in our application that could run SQL commands and it would give us the output all through the web application. This allowed for us to easily visualize what we were doing without having to constantly connect to the server through GCP. We did not include this in our production version because that would obviously be a large security concern.

Nidhi Chaudhari: One challenge that we faced was working with GCP and connecting to the database. Since it was a relatively new concept to us we did not know how to fix our issues and we could not find too much documentation online to help us. We found ourselves stuck many times because we were unable to connect to the database on GCP, or because our credits ran out, or we were not sure exactly how to link GCP to our code in Python. A combination of not having enough knowledge and a lack of sufficient documentation was what got us stuck a couple of times.

Regarding the differences between our original proposal and the final application, there were no other changes to the original proposal than the one difference mentioned above. However, we did reduce the number of functionalities due to time limitations.

For future works, rather than allowing users to input any weapon or crime type, we should only allow them to select one of the pre-existing types, or have a way for them to suggest a new type that will be allowed after it is approved by the LAPD or application moderators. This will allow the data to be sorted and organized in an improved manner. We could also allow for advanced querying, so the users can write queries that allow results that at least one of the keywords they specify using OR, or all the keywords they specify using AND. This way users do not need to be database experts (or even have database access) but can complete complicated tasks.

We split the work of developing our application into multiple parts, including setting up the backend, the frontend, each of the backend CRUD endpoints, corresponding frontend pages, and the stored procedure, and trigger. Each member took upon a task based off of their prior experience, as well as what they wanted to learn. This worked well as everyone was happy doing their tasks, and with frequent and effective communication, we were able to ensure our components fit together seamlessly.