

SafeCity Design Document

Eric He (ehe)
Kavin Sivakumar (ks16)
Stefan Keselj (skeselj) – Liason

March 15, 2016

1 OVERVIEW

SafeCity is a web application which informs people about the crimes happening in different areas. Its principal component is a map which displays crimes that happened in the area selected as color coded bubbles. It also features a control panel, which visualizes the crime data in different graphs, displays impressions users wrote about different areas, and links to articles about the safety of different areas. It will be structured as a three tiered application, with a MariaDB database, a PHP and Python logic layer, and Bootstrap and KeenIO presentation.

2 REQUIREMENTS AND TARGET AUDIENCES

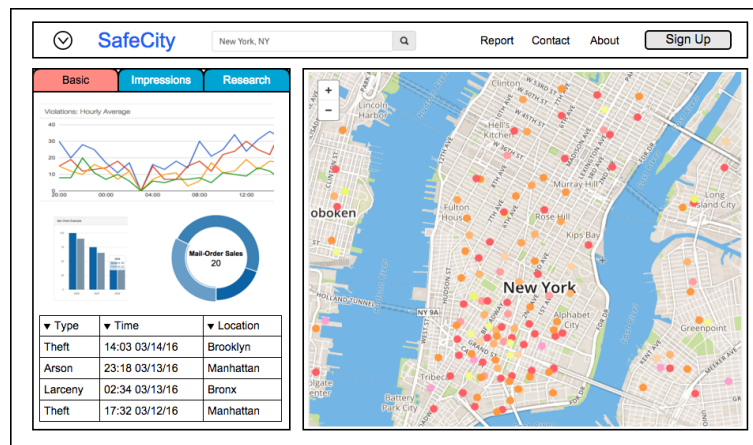
There is a massive knowledge gap between the authorities and local residents that know information about crime and the people that need it. Police stations release vast amounts of crime data, but it is meaningless to people looking for insights in such an unprocessed form. People living in areas where crimes happen know all there is to know about the true character of their community, but have no medium to communicate it to interested people. SafeCity will solve this problem by delivering analytics based on police reports and centralizing the impressions people have of areas. It will serve anyone interested in knowing more about the safety of their community, so virtually all people at one point or another.

There have been multiple attempts at a tool like this, but none have managed to achieve the two attributes we think will differentiate our product: clear design and comprehensive information. CrimeMapping has sparse data, almost exclusively in Los Angeles (there are two

crimes in the last month in New York). CrimeReports covers many cities, but the vast majority of crimes it reports are sexual assaults. SpotCrime has passable density over a fair number of cities, but the user can only see the crimes within a circle of a few miles in radius around a reference point, and zooming out does not display any more information.

3 FUNCTIONALITY

The main focus of this application is to connect users with information about crimes happening in their cities. Each of the possible scenarios will be similar: a user will pan and select different areas on the map to the right and then interact with one of three different data representations on the control panel to the left. What differentiates these users is the type of data they will be interacting with, which comes in three forms: graphical, verbal, and quantitative.



3.1 CASUAL PASSERBY

This user represents the people who stumble upon our site after having a transient interest in crime rates while surfing the web. The user will load the front page to find a large map taking up the rightmost two thirds of the screen, displaying all the crimes that happened in Manhattan in the last year as color coded bubbles. There will be a slider on the bottom of the map which will allow the user to play around with the time frame displayed on the map (maximum last ten years, minimum last week). The user can mouse over the map to find out details of each crime: time, type, and any people involved. To the left there will be a control panel summarizing the information in the map in different ways, which we hope will catch the user's eye for a few seconds. It will contain a line graph displaying the total number of crimes over time, a pie chart displaying all the different types of crimes, and a scrollable table containing all the crimes sorted chronologically. Once the user gets bored with Manhattan, he or she will type a different place, most likely his or her hometown, up in the search bar at the top of the screen and go through the same process, and then leave.

3.2 ACTIVE MEMBER

This user represents the people who have more lasting involvement with our application because they care about how the world views their community. This person will want the full benefits of being a member, so at some point they will click on the "Sign up" button on the top right and enter their username, password, full name, email address, and postal code (the last two will be verified). From then on they will either sign in through a login page or automatically sign in through their browser. This type of user has already explored the basic summaries mentioned above, and will spend more time on the second tab of the control panel: the impressions page. This page will be something like a Twitter feed, containing geo-tagged tweets involving safety or crime in an area, and a stack of 200 character testimonies people wrote about the safety of an area. Active members will be the primary contributors and curators of this information. They will write the testimonies about the areas in their city that they know, and they will upvote or downvote content as they feel fit to ensure that the most representative snippets are noticed. Periodically, these members will receive an email digest of what has been said about their area to invite them back.

3.3 IN-DEPTH RESEARCHER

This user represents the people who are familiar to our site and come to it for more detailed and technical forms of information than those mentioned above. This user will occasionally check out the two tabs mentioned previously and be a member like the case above, but will spend most of their time viewing and downloading data from the third section of the control panel: the research tab. In this tab, the user will be able to download plain text files containing all the data used to make our application: all the times, types, and locations of crimes in an area and all the testimonials written about an area. The hope is that this user will use our information to gather further insights into how and why crime is happening. We will offer the option for writers to provide links to their work or that of others, so interesting articles or reports will be available to anyone else who is curious. Additionally, we will display our own take on using our data to research crime: a k-nearest-neighbors crime prediction algorithm which estimates the likelihood that certain crimes will happen in certain areas. This data will be overlaid to the right as a heatmap. This use case involves very technical features with limited marginal benefit, we consider it our stretch goal.

4 DESIGN

SafeCity will be implemented as a classic three-tier system, with the data tier (a MariaDB MySQL database with bash update scripts), logic tier (querying through PHP and Python preprocessing modules), and presentation tier (Bootstrap layout, KeenIO charts, and Google Maps API) all mutually independent. In general, the user will engage in a small action that prompts our business logic to request crime or user information from the database, which will be processed and then displayed for the user in a responsive format.

4.1 DATA

This is a data driven application, so the data tier is arguably the most essential to its success. These days NoSQL databases like MongoDB are popular, but we have chosen to keep it simple with MariaDB relational database because we think our data will have a simple and uniform structure. There will be four tables: user, crime, impressions, and possibly research. The user table will store the name, username, password (md5 hashed), email, and zip code of each user. crime table will store longitude, latitude, type, and time of each crime. The impressions table will store the longitude, latitude, content, and score of each impression. The research table will store the longitude, latitude, and link to each research item.

The users table will start out empty but every time a user signs up a row of information will be added. We will initialize the crime tables using data dumps of historical data, and then update them every day by scraping the webpages of different police and news stations with bash scripts that use wget or possibly BeautifulSoup. To maintain the impressions tab, we will both store any impressions written by our users and log geotagged tweets from Twitter's API. The research tab will be updated by people who add links onto our website for the most part, but we might add a few on our own to get the ball rolling. No values in the database will ever be deleted because each entry in our databases takes only on the order of 10 bytes.

4.2 LOGIC

The role of the logic tier is to respond to requests from the presentation tier by retrieving information from the data tier and then process it before giving it to the presentation tier. A user based query will happen when a user tries to login or sign up, and it will involve reading and writing to the user table using PHP. If the user exists, the username and zip code will be sent back to the presentation tier as a tuple, otherwise nothing will be returned. A location based request will come whenever the user pans on the map, selects an area of it, or enters a location in the search bar (which puts the map in that location). These location based requests have the form of four floats: the minimum and maximum longitude and latitude of a rectangular area of the map. Through PHP select requests, we will query the database to pull all crimes, impressions, and articles with longitude and latitude between the minimum and maximum longitude and latitude. It is not yet clear how we will make this operation quick when the amount of data we have gets very large, but for now we will likely allow the select operation to traverse all the crimes. Once these crimes, impressions, or article links are obtained, a Python script will package them as tuples and concatenate them into a list and then send it to the presentation layer.

4.3 PRESENTATION

The presentation tier is where the magic happens, our quick and intuitive way of presenting information will be an important differentiator from our competition. The overall layout will be a KeenIO template for an analytics dashboard, which we chose to keep the different sections organized. KeenIO is compatible with Bootstrap (in fact it was implemented with it),

which will allow us to use Bootstrap's elegant buttons, search bar, and tables. Most importantly, both Bootstrap and KeenIO are responsive frameworks, so our website can be viewed on screens of different sizes (such as phone screens, our competitors cannot do this). Finally, the driver of the application is the map, which will be a window from the Google Maps API displaying all the crimes given to it by the logic layer. Every time the user pans across the map, selects a region of it, or enters a new address, the maps API will return the center coordinates, length, and width of the current window (we think), which will then be sent to the logic layer in four tuple form for information retrieval.

5 TIMELINE

5.1 PRODUCT CHECKPOINTS

Prototype A:

This is the first checkpoint we will be working towards. Prototype A will have the infrastructure needed so that the app can access crime data and then work with the data in some way, either printing it out or populating a map with each crime. Prototype A will only have a map for New York City, but there will be options that categorize the crimes. For instance, you can select different options that show only crimes that happened in the past year, or in past years.

Prototype B:

The second checkpoint, Prototype B, will allow for a crime map of any area chosen by the viewer through an address and Google Maps. At this point, all data analytics for the app has been completed. Prototype B also includes a login and signup feature which will allow the user to create his or her own account for our application, as well as the impressions page.

Final product:

The final product will finally have a more polished version of the impressions tab and be linked to Twitter. If we have time, it will also include the research tab. Any timing issues with the database will have been resolved. Moreover, all formatting and aesthetic decisions will be taken place here so the application looks clean and pleasing to the eye.

5.2 WEEKLY CHECKPOINTS

Week 1 (3/20-3/26): Setup

Draw up on paper different layouts of our application and how the final product will look like. Setup the Github account for our project and make sure all logistics in that avenue are set up. Build project website as specified by the course website also by 3/21. Setup the MariaDB database so that we can read and write from the command line. Download crime data for NYC.

Week 2 (3/27-4/3): Prototype

Build an application that takes in crime data and colors in a map with the individual entries,

this will be the prototype specified by the course website that is due this week.
As mentioned above, the map will only be of NYC and the only analytics we will have is basic categorization of crimes. We will have a pie chart that showcases this.

Week 3 (4/4-4/10): Other Analytics

By this week, we will finish the categorization of crimes option and start adding the two other analytics components: the line graph of crimes versus time and the scrolling table of crimes. By the standards outlined above, Prototype A should be done by this week.
This week we will complete our infrastructure for automatically updating existing crime data. By our standards, our Prototype A should be done by this week.

Week 4 (4/11-4/17): Expanding Data

Demo Alpha version.

Allow any location in the US to be searched and the crime data for it displayed.
Have automatic scraping and updating scripts for all these locations.
Ensure that reading and writing from the database with all these different locations is fast.
Data analytics should be done at this point, so any remaining bugs should be fixed by now.

Week 5 (4/18-4/24): User Interaction

Gather feedback from our friends to see if they would use our application.
Complete the login and signup feature, test it thoroughly ourselves.
Have some form of an impressions page, the Twitter link is not necessary but by the end of this week a user must be able to write an impression for an area and we must save it.
At this point Prototype B should be completed.

Week 6 (4/25-4/31): Touchup and Stretch

Demo Beta version.

Work on aesthetics of application and fix any remaining bugs before anything else.
Integrate NLP better with Twitter if the impressions tab completed above is lacking it.
Add another layer to the database for the research tab and start adding a few sample articles.

Week 7 (5/1-5/5): Final Demo

Demo final product.

Work on final documentation to be handed in on Dean's Date.
(Hopefully) relax and have fun playing with our website.

6 RISKS AND OUTCOMES

6.1 SOCIAL

When dealing with the publication and visualization of crime data, something that ultimately involves the safety of people's lives, there will undoubtedly be many social risks that need to be carefully considered. The sensitivity of this topic has hindered many other people trying to tackle crime analytics, and we would like to be clear that our intention is only to present

the data as it is, not defame neighborhoods or ruin property values.

Our goal is to have data on crimes easily and widely available to anybody that decides to use our app. Searching for recent activity in a certain borough in New York City should immediately pull up all relevant crimes with additional details if desired. However, due to the negative nature of crimes, such information may trigger an over-exaggerated reaction to a situation that, in reality, needs context to understand properly. For example, a user looking for a place to travel may search up crime data on Queens, and if he or she finds a single entry on crime, might deem the area unsafe to visit. This is a completely natural and understandable reaction, but also biased. Is it possible for an area with a population of over two million people to be entirely devoid of crime? This bias will probably exist, and may skew local economics in unintended directions based upon misjudgements. However, our role as developers is to provide this data in an organized, but raw and unbiased form. The way the data is interpreted is out of our control.

In fact, we hope that the outcome of this project is exactly the aforementioned bias, but viewed in a more positive light. Our goal is not to discourage business in local stores or devalue houses in areas where crime rates are high, but to allow users to use their biases to make more informed judgements. If a user is extremely risk averse and will not tolerate a single crime in a certain area, well so be it, but at least he or she can use our app to easily make that judgement. Similarly, if a user doesn't mind a few crimes per couple hundred thousand people, so be it as well, but once again, at least he or she can use our app to easily make that judgement. Our end goal is to quickly and effectively deliver information about crime, information that is important to the safety of the user.

6.2 TECHNICAL

- Google API might not provide as much information about the current state of the map as we think (i.e. it might only give the coordinates for searches, not for panning)
Mitigation: we will include a refresh button which sends our server the coordinates, or our own arrows which expand the window and tell us the new coordinates.
- Twitter API might not allow us to display their tweets alongside our impressions
Mitigation: we will include two separate windows in the impressions tab: one for testimonies our users generated and one for tweets about the area
- KeenIO might not allow us to display crimes as color coded bubbles in the exact way we want, and might force us to use the large, ugly tags they have by default
Mitigation: we could switch to a different library which does allow us to do things the way we want, use JavaScript to overlay dots on Google Maps directly, or use the tags
- MariaDB might be slow because it's not a fancy NoSQL database
Mitigation: this probably won't happen because our data set is relatively small, but we could make search more efficient by sorting or indexing on our longitude and latitude coordinates