

J.U.S.T.I.C.E.

Judgment Utility Space Time Intensive Crime Evaluator

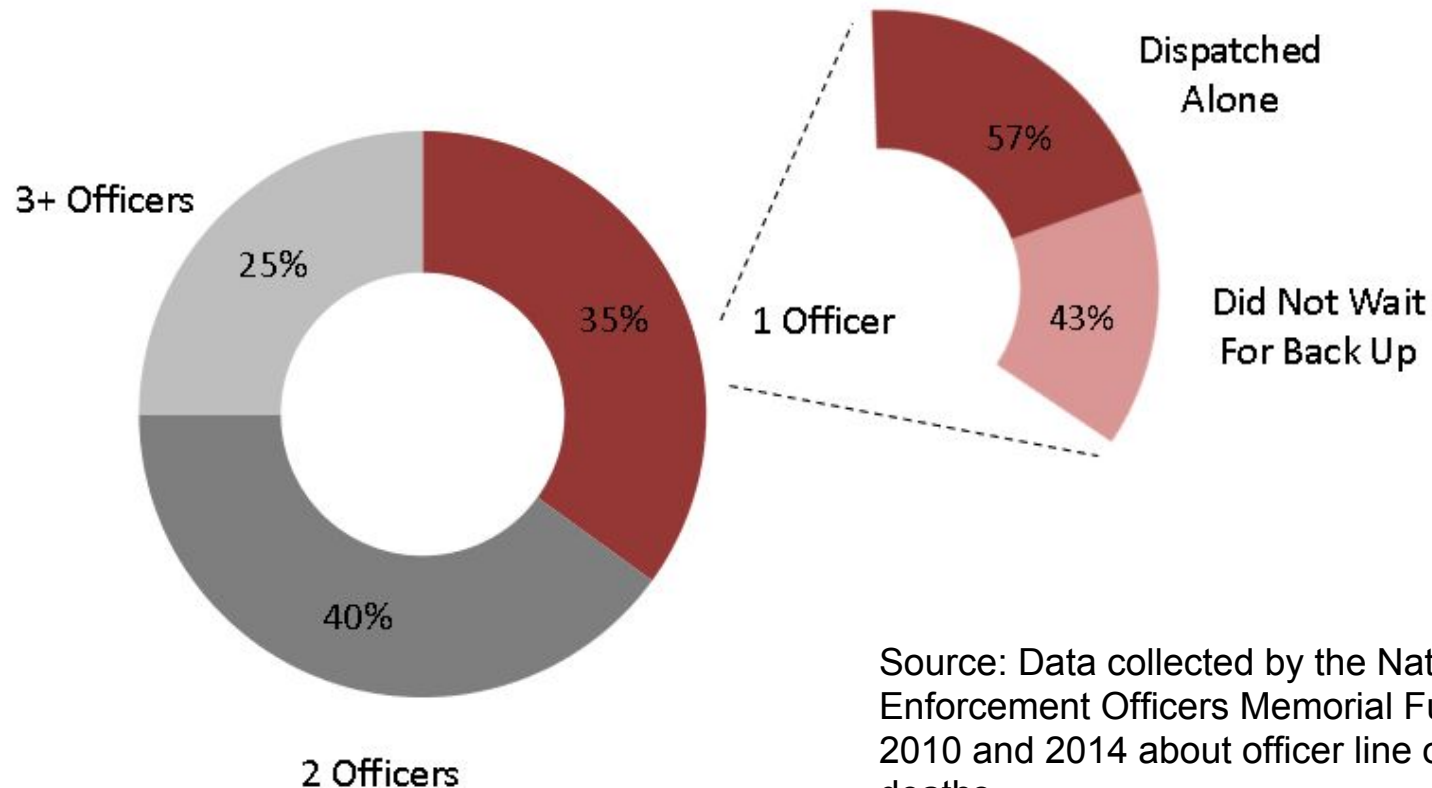
Team 1: Aequitas

Andrew McLamb, Arjun Saini, Gregory Mayer, Nicholas Sorauf

Project Background/Motivation

- Our requirements were to develop a crime data visualizer which displays Baltimore city crime statistics according to publicly available data on Open Baltimore.
- The visualizer would display data in a variety of methods, such as a heat map and graphs (line, bar, pie chart), as well as in a tabular format.
- The user would also have the ability to filter and sort data according to various parameters.
- The goal of our crime visualizer is to save officers lives by providing strong data analysis in the hands of every police officer in the field.

Number of Officers on Scene at Time of Line of Duty Death



Source: Data collected by the National Law Enforcement Officers Memorial Fund between 2010 and 2014 about officer line of duty deaths.

Breakdown of 91 Line of Duty Deaths by Dispatched Call Types

Source: Data collected by the National Law Enforcement Officers Memorial Fund between 2010 and 2014 about officer line of duty deaths.

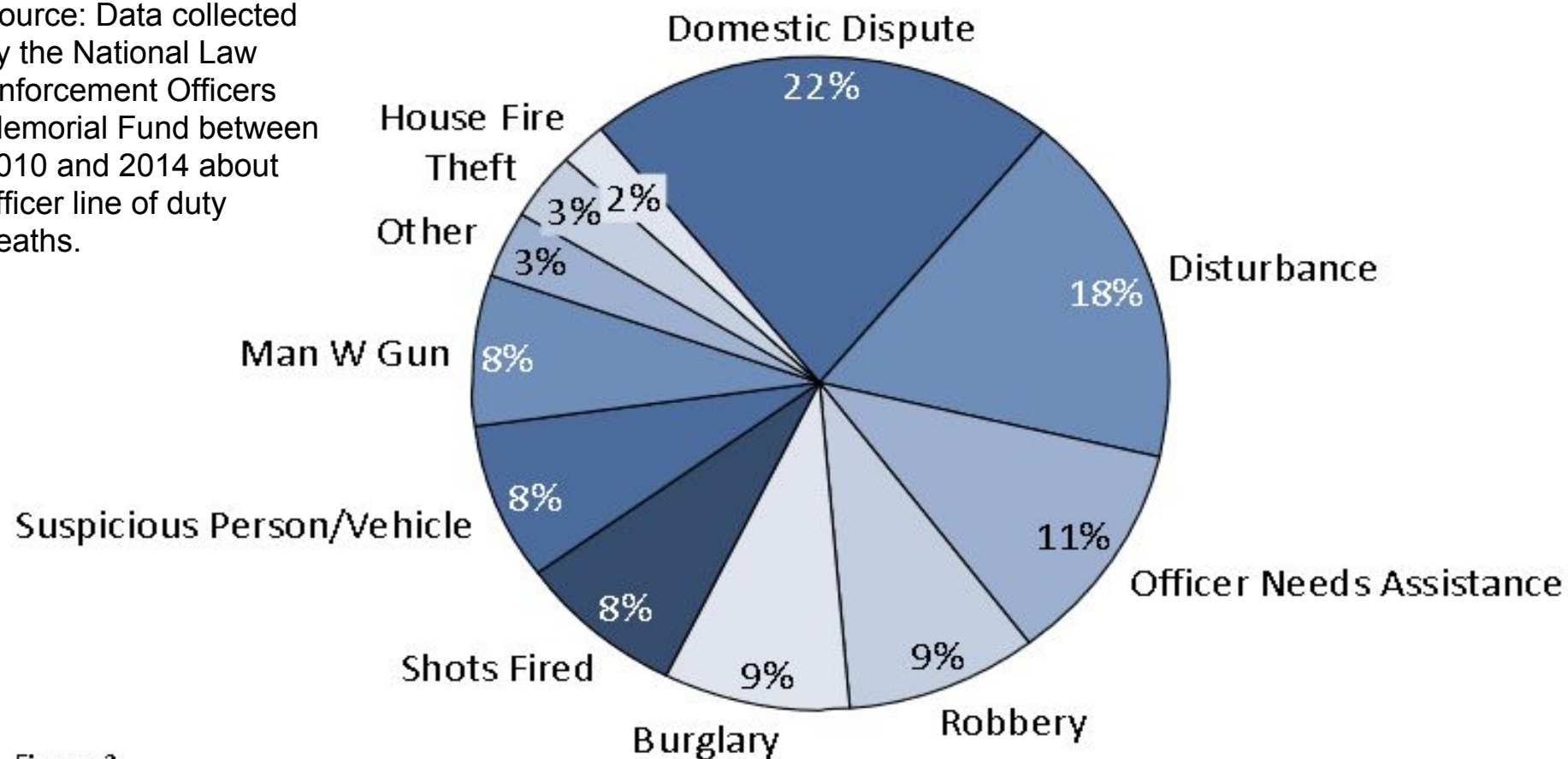


Figure 2

Prototype

<https://swe.umbc.edu/~nsorauf1/447/>

Timeline

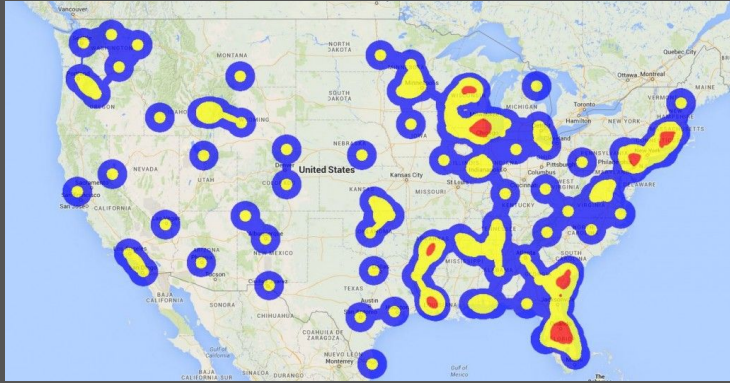
Date range selector



Time range selector



Heat Maps and Pin Maps



Source:
<https://www.espatial.com/articles/big-data-the-insurance-business-and-mapping-software>



Source: <https://www.espatial.com/features/pin-map>

Heat maps allow one to visualize different concentration levels of crime, while the Pin maps allow for more focused analysis.

Filter and Sort Options

- Timeline
- CrimeCode
- Location
- Description
- Inside/Outside
- Weapon
- Post
- District
- Neighborhood
- Longitude, Latitude
- Premise (Type of Location)
- Total Incidents
- Threat Level

Features

- Dashboard
- Timeline
- Heat map and pin map
- Bar graph and line graph
- Pie chart
- Single crime view

Advanced Features

- Displaying crime characteristics that are the most common
- Filters (applying multiple filters at once)
- Sorting data
- Applying changes on the page instantly
- Data processing
- Data analytics

Post-Release Features

- Saving user preferences
 - Most frequently used filters, allow users to change default settings
- Statistical analysis
 - I.e. is there a correlation between a specific neighborhood and a certain type of crime?
- Mobile application
 - Have our application dynamically adjust to the type of device it is being used on and adjust based on portrait or landscape orientation for mobile devices.

Functional Requirements

- View heat map
- View pin map
- Map zooming
- Map panning
- View bar graph
- View line graph
- View pie chart
- Table view
- Sorting tabular data
- Filter data
- Adjust column size in table view
- Pagination in table view
- Single crime view
- Manage timeline

Non-Functional Requirements

- Front end shall be in JavaScript & HTML
- Shall use a code style checker
- Shall have a package manager
- Should handle up to 500,000 entries
- Back end shall be in node.js
- Shall be available through the web
- Shall be hosted on a linux server
- Shall have revision control
- Should be compatible with Firefox and Chrome
- Shall use MongoDB
- Webpage shall render within 5 seconds

Architecture design

