Project Save Baltimore

-

System Requirements Specification (SRS)

Version 1.0

Produced For:
Next Century Corporation

Produced By:
Team Indigo (CMSC447):

- Neil Joshi
- Nat Baylon
- Matthew Landon
- Bernie McNamee

Date: February 25, 2016

Project Save Baltimore
System Requirements Specification

Table of Contents

1. Introduction

1.1 Purpose of This Document

The purpose of this document is to provide a comprehensive overview of what *Project Save Baltimore* will do and how it should perform from a user's point of view. This document is targeted towards both Next Century who is the customer of the software, Baltimore City Police Department who will use the system to perform analytics Team Indigo that develops the software.

1.2 References

Next Century Introduction Presentation by John McBeth & Michael Smolyak - 2/2/2016
https://blackboard.umbc.edu/webapps/blackboard/execute/content/file?cmd=view&content_id=_1717281_1&course_id=_23417_1

SRS Template by Susan Mitchell & Michael Grasso - 2/11/2016
http://www.csee.umbc.edu/courses/undergraduate/447/documents/SRS_Template.pdf

Software Requirements Presentation by Dr. Karuna Joshi - 2/11/2016
http://www.csee.umbc.edu/courses/undergraduate/345/spring12/mitchell/lectures/requirements_S12.pdf

1.3 Purpose of the Product

The Baltimore City police department has a database containing information about crimes that have occurred in the area going back a few years. However, it is hard to see and draw meaningful conclusions about a data table only. This project is trying to create a better format for this information to be visualized. The ideal result is for the officers to be able to see trends in the data and be able to react to them. The likely users will be desk employees of the local law enforcement offices. They would look at past trends and gather meaning from them.

This is for analysis of post events, not for real time predictions. This would be separate from in-the-field work. Users would see information up to a certain point in time when the public database was updated.

1.4 Product Scope

The main system will be a public website that will display crime analytics. Users will be able to see past crime information in a number of formats.

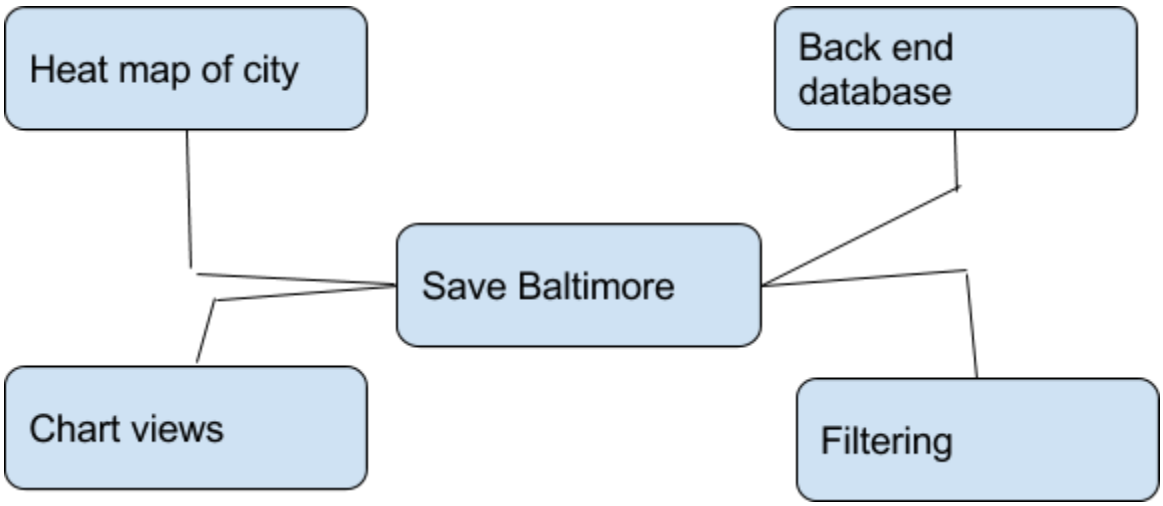
There will be tabs with different ways to visualize the data:

- A map showing a heat map of Baltimore with brighter spots representing high volume crime areas
- A tabular view of the data, to display raw data values
- A charts tab with pie charts, bar graphs, scatter plots, and histogram

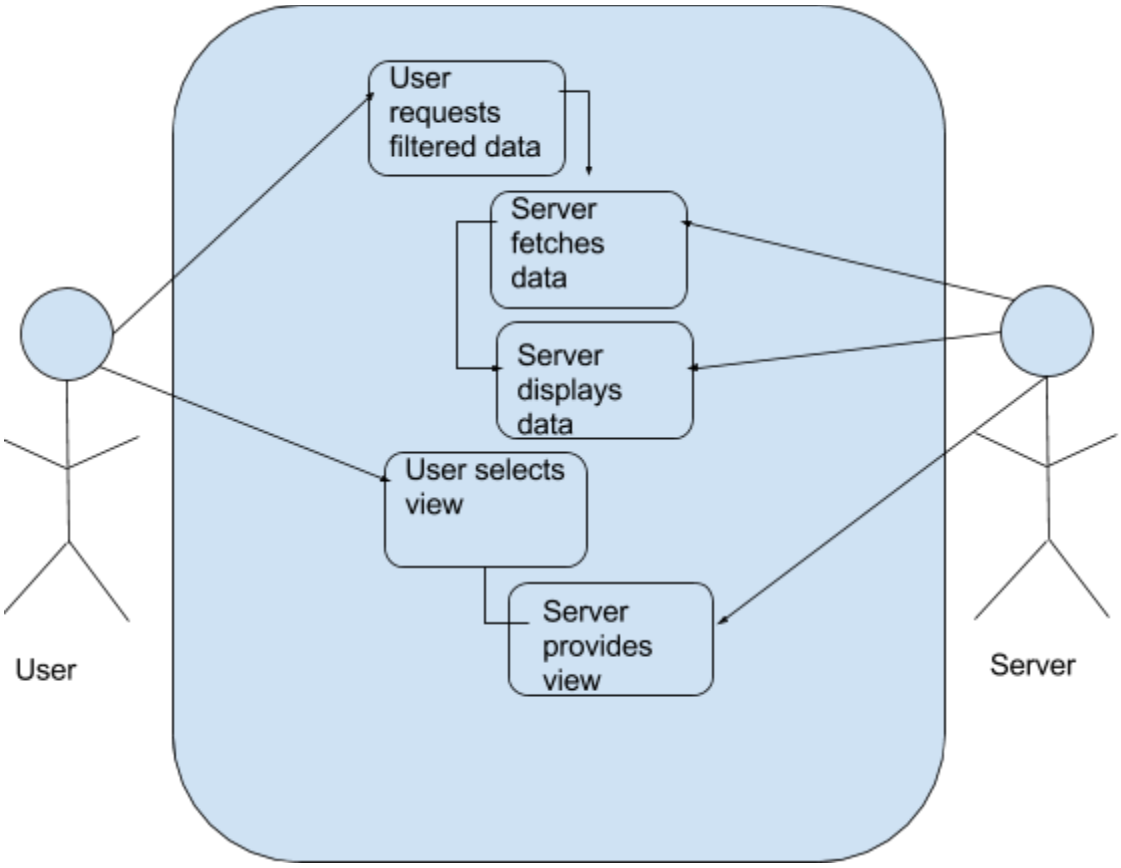
There will be a filter section with filters that persist on the views, no matter which tab is used, and the different views of the data will be filtered by what the user selects in the filters. The filter terms will be street name, crime type, weapon, district, neighborhood, and time interval. The time interval will be separate from the other filters, and will be below the data view.

The dataset is downloaded by the administrators, who run scripts on it to clean it and update the database when they would like to update the dataset.

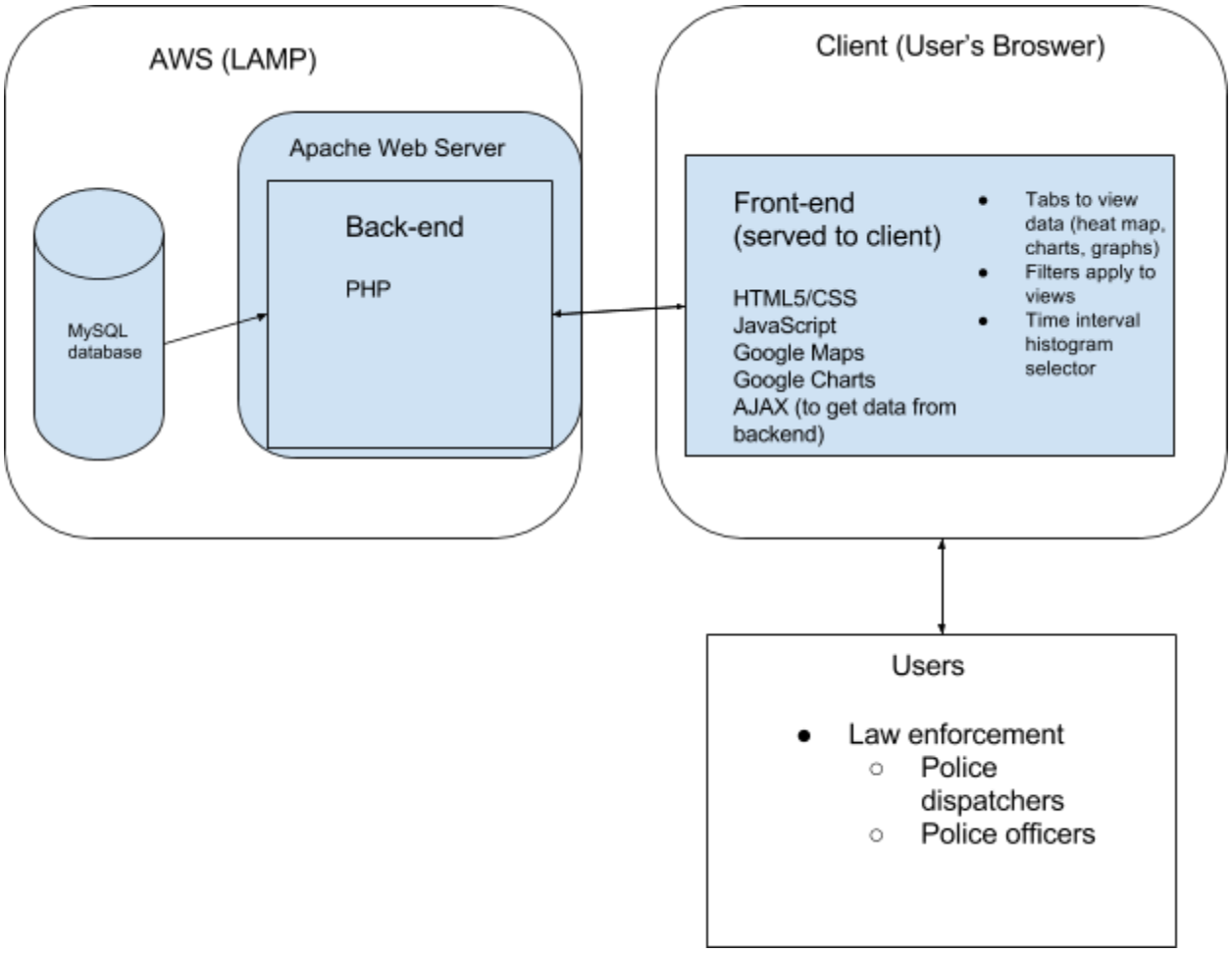
Context Document



Top Level Use Case



Scope Definition Diagram



2. Functional Requirements

Number	**1**	
Name	User Requests Filtered Data	
Summary	The user selects a filter term, be it temporal or geographical information	
Priority	5/5	
Preconditions	The components of the site are in working order	
Postconditions	The current view displays the filtered data the user requests	
Primary Actor	User	
Secondary Actors	Web Server	
Trigger	The user changes either the time slide-bar or a dropdown filter item	
Main Scenario	**Step**	**Action**
	1	User selects filter term
	2	Server fetches the filtered data
	3	Server provides the filtered data in all views
Extensions	**Step**	**Branching Action**
	1a	Filters by district, neighborhood, street name, crime type with drop down checkboxes
	1b	Filters by date (histogram time slider). User drags the min/max date, server responds with filtered data

Number	**2**	
Name	View Baltimore Heat Map	
Summary	User opens to the Google map tab	
Priority	4/5	
Preconditions	Must have properly formated lat/long data to be able to mark	
Postconditions	Plots crime data with brighter spots showing heavy crime and less color showing little crime	
Primary Actor	User, database	
Secondary Actors	Map service	
Trigger	User opens map tab	
Main Scenario	**Step**	**Action**
	1	Click map tab
	2	Page shows map with hotspots
	3	Zooming and scrolling by map
Extensions	**Step**	**Branching Action**
	1a	Applying different filters will bring up more targeted map

Number	**3**	
Name	View charts (see 1.4: Product scope) showing break-downs of events by the filter terms	
Summary	User opens the specific chart tab	
Priority	4/5	
Preconditions	Need to have enough crimes with the requested field to be able to show meaningful analysis	
Postconditions	Creates different charts showing a breakdown of data through the different filters the user selects.	
Primary Actor	User, database	
Secondary Actors	Google Chart (API) Service	
Trigger	User opens one of the tab	
Main Scenario	**Step**	**Action**
	1	Click desired chart tab
	2	Chart is made with requested field

Number	**4**
Name	Tabular view of the Dataset
Summary	The user opens the tabular view to look at the clean raw data
Priority	4/5
Preconditions	Data is in the Mysql database
Postconditions	User sees the records in a table view

Primary Actor	User, database	
Secondary Actors		
Trigger	User clicks tabular view tab	
Main Scenario	**Step**	**Action**
	1	Tabular view is opened
	2	A page with the html table and data is shown
Extensions	Step	Branching Action
	1	User clicks on column header, table sorts by that column

Number	**5**
Name	Dataset is updated with cron job
Summary	Cron script fetches data, runs scripts, and updates the database
Priority	3/5
Preconditions	Original source is updated (monthly)

Postconditions	Database has current crime data	
Primary Actor	Cron scheduler	
Secondary Actors	Source, database	
Trigger	First day of the month comes, cron job is executed	
Main Scenario	**Step**	**Action**
	1	Bash script downloads dataset in a csv
	2	Bash script executes a python script that reads the csv and updates the dataset

Test 1: Click on a filter option. The desired result should be that the map/visual display updates.

Test 2/3/4: Click on a data view (map, various charts, tabular) tab. This should change the view to whatever the user selected (e.g. Heat Map, various types of charts/graphs, or table).

Test 5: Admin checks database after the first of the month to see if it has new crime information

3. Non-Functional Requirements

Number	Name	Description	Tests	Priority
1	**Browser Compatibility**	Website is compatible with latest browsers	Try different browsers: Chrome, Firefox, Microsoft Edge	3/5
2	**Responsive UI**	When a user clicks on an option, the interface	Click on different filters and tabs, try resizing the	4/5

		shows the result in less than two seconds	page or using different resolutions	
3	**Convenient Map**	Map should be intuitive (dragging gestures to pan, scroll to zoom)	Try using the map: zoom, drag, labels on roads/areas	4/5
4	**Stable System**	System should not crash, regardless of what the user is trying to do, intended or not.	Try many different combinations of input, let the site run for a long time	4/5
5	**Friendly UI**	User interface is simple, easy to use, and legible	Ask customer and people not involved with the project to try out the UI	3/5
6	**Site Always Up**	Whenever a user wants to access the site, it is available for use	Try the site during different times of day, try allowing many users	4/5
7	**Site Security**	Website should not allow any user the possibility to modify the database	Try changing the URL for malicious intent, ask friends to hack the site and attempt to modify data	1/5
8	**Data Updates**	New data are accessible and updated to the database	Check if database is updated after the first of the month, or look at the tabular view and sort by date-time field	1/5
9	**Maintenance**	System should be maintained throughout its use, in case any new issues arise	Make sure that software is compatible with current browsers. And all applications are at their latest version	2/5
10	**User Profiles**	A login system should be used to further improve security and allow expandability to more secure datasets	Sign in to the system, store information about the user's' past searches	1/5

4. **User Interface**

See "User Interface Design Document for *Project Save Baltimore*."

5. Deliverables

All deliverables will be given on Product Delivery and Demo Day at Next Century: May

Hard copies of each of the following:

- Systems Requirement Specification
- System Design Document
- User Interface Design Document
- User Manual
- Administrator Manual
- Copies of all Bi-weekly Status Reports

A CD (or electronic copy in a ZIP file) containing the following:

- Systems Requirement Specification
- System Design Document
- User Interface Design Document
- User Manual
- Administrator Manual
- All source code
- A link to the website containing the product
- Any other software required for installation and execution of the delivered program

6. Open Issues

1) None

Appendix A – Agreement Between Customer and Contractor

This document defines what the end product will contain. Each use case explains a capability of the system. By signing off on this section, we agree that we are expected to deliver a program with each of the above features. Also, we agree to be open to changes from the customer and integrate them into the system.

When changes are requested, there is a procedure that should be used to more easily integrate them into the system. First, the customer expresses interest. If it is a major alteration, the team and clients will meet to discuss all aspects of the new addition and gather requirements of the new capability. Then, the team will talk offline and formulate final requirements and show them to the customer. We will make any changes until both are happy. Finally, the team will design and implement the feature.

Customer Comments:

Appendix B – Team Review Sign-off

By signing below, I agree that I have reviewed this document and I am content on its contents. I assisted in the writing and formalizing of the document.

Neil Joshi	Signature	Neil Joshi 3/7/16
Nathaniel Baylon	Signature	Nathaniel Baylon 3/7/16
Matthew Landen	Signature	Matthew Landen 3/7/16
Bernie McNamee	Signature	Bernard McNamee 3/7/16

Member Comments:

Appendix C – Document Contributions

Neil (~25%): Foundational outline, cover page, document purpose, NFRs, document cleanup

Nat (~25%): Use-case diagram, use-case table #1&4, help with NFRs, references

Matthew (~25/%): Context diagram, scope definition paragraph, use cases, help on NFRs

Bernie (~25%): Created the UML/Use case diagram and contributed to NFRs