

# System Requirements Specification

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

*Judgment Utility Space Time Intensive Crime Evaluator*

**Client**

Michael Smolyak

**Team 1 - Aequitas**

Gregory Mayer

Andrew McLamb

Arjun Saini

Nicholas Sorauf

2/19/2018

JUSTICE  
System Requirements Specification

Table of Contents

<b>1. Introduction</b>	<b>3</b>
<b>2. Functional Requirements</b>	<b>5</b>
<b>3. Use Case Tests</b>	<b>18</b>
<b>4. Non-Functional Requirements</b>	<b>21</b>
<b>5. User Interfaces</b>	<b>23</b>
<b>6. Deliverables</b>	<b>24</b>
<b>7. Open Issues</b>	<b>25</b>
<b>Appendix A - Agreement Between Customer and Contractor</b>	<b>26</b>
<b>Appendix B - Team Review Sign-off</b>	<b>27</b>
<b>Appendix C - Document Contributions</b>	<b>28</b>

# 1. Introduction

## 1.1 Purpose of This Document

This document details the features for the JUSTICE web application, its functions, and the preconditions required for operation. The intended audience is the Baltimore Police Department.

## 1.2 References

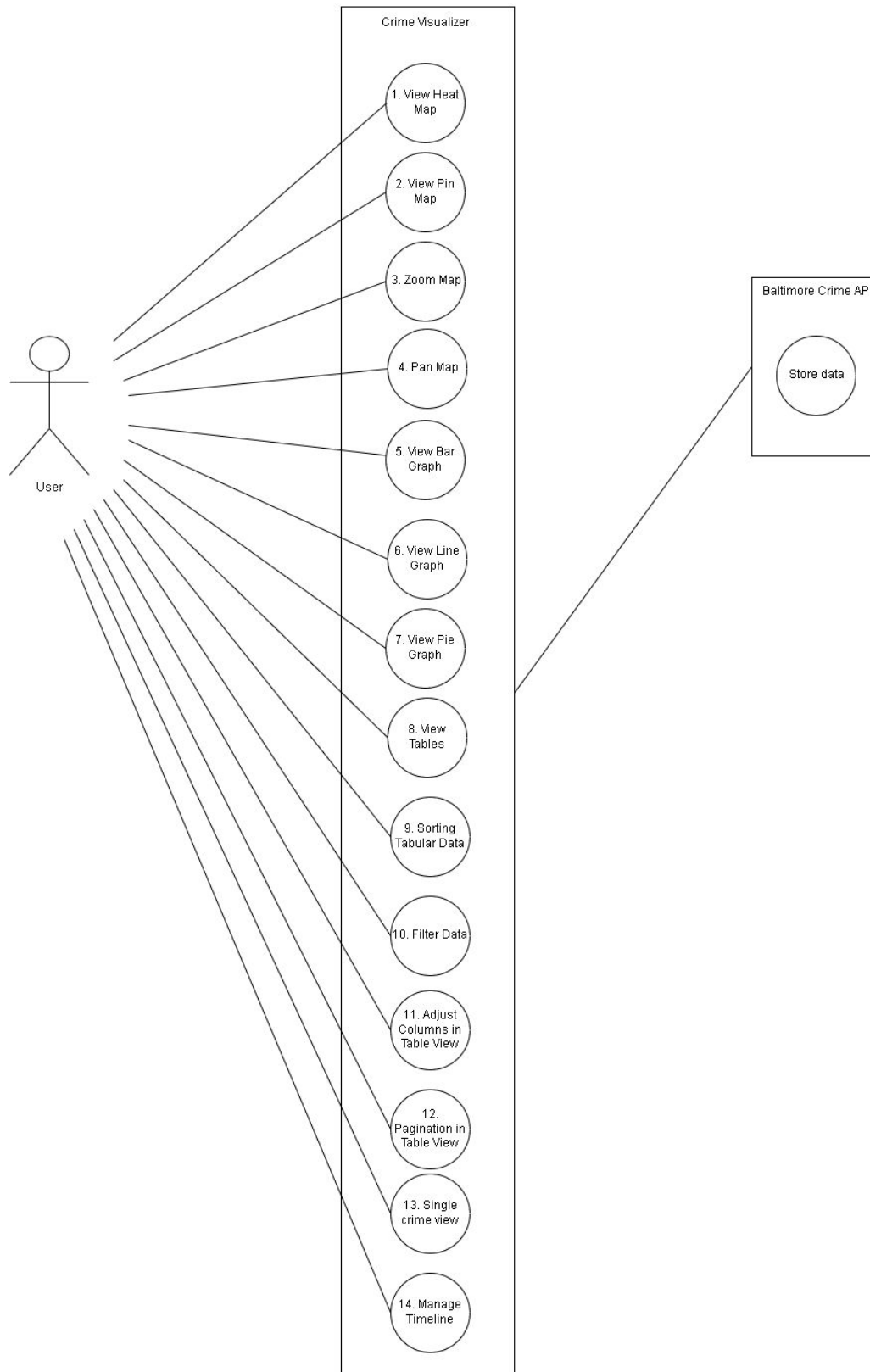
None so far

## 1.3 Purpose of the Product

JUSTICE is designed to provide a convenient way for the Baltimore Police Department to visualize the victim based crime data published on the Open Baltimore website. Doing so will allow them to analyze possible trends and help them to make more informed decisions with regard to allocating department resources.

## 1.4 Product Scope

The JUSTICE web application consists of multiple use cases including, but not limited to: Viewing maps, charts, and tables, filtering and sorting data, adjusting the timeline, and viewing crimes individually. Please refer to figure 1 below for further understanding of the actions available to users.



**Figure 1: Use Case Diagram**

## 2. Functional Requirements

Please refer to the use case diagram, figure 1, above for an overview of each use case.

### 2.1 Use Case 1

Number	1	
Name	View Heat Map	
Summary	User views their chosen data visualized as a heat map.	
Priority	5	
Preconditions	The user has filtered and chosen the data that they want to view as a heat map.	
Post-conditions	The heat map displays using the chosen data.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	User clicks on view heat map button.	
Main Scenario	Step	Action
	1	User selects the data to be used.
	2	System displays the heat map.
Extensions	Step	Branching Action
	2a	The user may decide to use zooming and panning features while looking at the map.
Open Issues	None	

## 2.2 Use Case 2

Number	2	
Name	View Pin Map	
Summary	User views their chosen data visualized as a pin map.	
Priority	5	
Preconditions	The user has filtered and chosen the data that they want to view as a pin map.	
Post-conditions	The pin map displays using the chosen data.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	User clicks on view pin map button.	
Main Scenario	Step	Action
	1	User selects the data to be used.
	2	System displays the pin map.
Extensions	Step	Branching Action
	2a	The user may decide to use zooming and panning features while looking at the map.
	2b	The user may click on a specific pin on the map to see its data in single crime view.
Open Issues	None	

### 2.3 Use Case 3

Number	3	
Name	Map Zooming	
Summary	User adjusts their map view by zooming in or out.	
Priority	5	
Preconditions	The user is already viewing a map.	
Post-conditions	The map is zoomed in or out to allow the user to view the map in different ways.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	The user scrolls up on their mouse wheel to zoom in or down to zoom out.	
Main Scenario	Step	Action
	1	The user scrolls their mouse wheel to zoom in or out.
	2	The system updates the map by zooming in or out.
Extensions	Step	Branching Action
	2a	The user may continue zooming further.
	2b	The user may pan the map around to focus on a specific area.
Open Issues	None	

## 2.4 Use Case 4

Number	4	
Name	Map Panning	
Summary	User adjusts their map view by panning around.	
Priority	5	
Preconditions	The user is already viewing a map	
Post-conditions	The map is panned to allow the user to view the map in different ways.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	The user click and holds a part of the map and moves their mouse in a direction and the map will pan in the opposite direction.	
Main Scenario	Step	Action
	1	The User clicks and holds a part of the map.
	2	While still holding the click, the user moves their mouse in the opposite direction they want to pan.
	3	The system updates the map by panning in the intended direction.
Extensions	Step	Branching Action
	3a	The user may release the mouse button and repeat the steps to continue panning.
	3b	The user may decide to zoom in or out after they have panned to a specific area.
Open Issues	None	



## 2.5 Use Case 5

Number	5	
Name	View Bar Graph	
Summary	The user views their selected data based on filters chosen as a bar graph.	
Priority	3	
Preconditions	The user has selected data they want to view as a bar graph.	
Post-conditions	A bar graph displays using the chosen data.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	The user clicks on the view bar graph button.	
Main Scenario	Step	Action
	1	The user must specify which part of their data set will fit to the X and Y axes. For example, CrimeDate on the X axis and Total Incidents on the Y axis.
	2	The system displays the bar graph using the given inputs.
Extensions	Step	Branching Action
	None	
Open Issues	None	

## 2.6 Use Case 6

Number	6	
Name	View Line Graph	
Summary	The user views their selected data as a line graph.	
Priority	3	
Preconditions	The user has selected data based on filters chosen they want to view as a line graph.	
Post-conditions	A line graph displays using the chosen data.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	The user clicks on the view line graph button.	
Main Scenario	Step	Action
	1	The user must specify which part of their data set will fit to the X and Y axes. For example, CrimeDate on the X axis and Neighborhoods (i.e. the number of crimes that happen in a specific Neighborhood) on the Y axis.
	2	The system displays the line graph using the given inputs.
Extensions	Step	Branching Action
	1a	The user may decide to pick and chooses the number of lines (Neighborhoods in this example) they want to see represented on the line graph.
Open Issues	None	

## 2.7 Use Case 7

Number	7	
Name	View Pie Chart	
Summary	The user views their selected data based on filters chosen as a pie chart.	
Priority	3	
Preconditions	The user has selected data they want to view as a pie chart.	
Post-conditions	A pie chart displays using the chosen data	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	The user clicks on the view pie chart button.	
Main Scenario	Step	Action
	1	The user must specify which category to use regarding their data set. For example, choosing Description creates a pie chart based on types of crime.
	2	The system displays the pie chart using the given inputs
Extensions	Step	Branching Action
	None	
Open Issues	None	

## 2.8 Use Case 8

Number	8	
Name	Table View	
Summary	The table view showcases the current data set based on filters chosen.	
Priority	5	
Preconditions	None	
Post-conditions	The data set is displayed with each data entry as a row.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	The user visits our web application.	
Main Scenario	Step	Action
	1	The System displays the current data set in a table view.
Extensions	Step	Branching Action
	1a	The user can sort the data alphabetically or numerically depending on which column header the user clicked on.
	1b	The user may adjust column sizes to focus more on specific parts of the data entries.
	1c	The user may alter the table view by choosing to only display so many entries per page or having all entries displayed on one page.
	1d	The user may click on a single entry to view the data in the single crime view.
Open Issues	None	

## 2.9 Use Case 9

Number	9	
Name	Sorting Tabular Data	
Summary	The user sorts the data alphabetically or numerically depending on which column header the user clicked on. Data can be sorted by date or time independently.	
Priority	4	
Preconditions	The user is currently in a table view.	
Post-conditions	The table updates to have the list of entries sorted as desired.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	The user clicks on a column header in the table view	
Main Scenario	Step	Action
	1	The system updates the table view with the entries sorted alphabetically or numerically depending on which column header the user clicked on. For example clicking on Neighborhood will sort the entries alphabetically by neighborhood (from A-Z).
Extensions	Step	Branching Action
	1a	The user may click the same column header to sort again. Doing so with Neighborhood will sort from Z-A.
Open Issues	None	

## 2.10 Use Case 10

Number	10	
Name	Filter Data	
Summary	<p>The user can globally filter the data into a subset by using categories such as: CrimeCode, Location, Description, Inside/Outside, Weapon, Post, District, Neighborhood, Longitude, Latitude, Location1 (Latitude, Longitude), Premise (Type of Location), or Total Incidents.</p> <p>*Note the user will not be able to filter by CrimeDate and CrimeTime because time related filtering will be accomplished using the timeline.</p>	
Priority	4	
Preconditions	None	
Post-conditions	Everything updates to only show the subset of data based off of the filtering choices.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	The user clicks on the apply filters button in the filtering sidebar.	
Main Scenario	Step	Action
	1	The user is prompted to choose which category to filter by and to what extent they want to filter using that category. For example, filtering by Neighborhood can create a subset of data for

		only one neighborhood or a few specific neighborhoods.
	2	Every view updates to show the new subset of data.
Extensions	Step	Branching Action
	2a	The user may decide to filter further by using other categories.
Open Issues	None	

## 2.11 Use Case 11

Number	11	
Name	Adjust Column Size in Table View	
Summary	The user adjusts column sizes to focus more on specific parts of the data entries.	
Priority	1	
Preconditions	The user is in the table view.	
Post-conditions	The table updates to show the adjusted column sizes.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	The user clicks and holds on the edge of a column.	
Main Scenario	Step	Action
	1	The user drags their mouse to the right or left to increase or decrease the width of a particular column.
	2	The system updates the table view to show the adjusted column sizes.
Extensions	Step	Branching Action
	2a	The user can repeat the above steps to adjust the sizes of other columns.
Open Issues	None	



## 2.12 Use Case 12

Number	12	
Name	Pagination in Table View	
Summary	The user alters the table view by choosing to only display so many entries per page or having all entries displayed on one page.	
Priority	1	
Preconditions	The user is in table view.	
Post-conditions	The table view is updated based on how the user wants to view the many data entries in the current data set.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	The User clicks on a page view button.	
Main Scenario	Step	Action
	1	The user is prompted to choose how many data entries are to be view per page or if all entries are to be displayed on one page.
	2	The system updates the table view to reflect the user's choice.
Extensions	Step	Branching Action
	None	
Open Issues	None	

## 2.13 Use Case 13

Number	13	
Name	Single Crime View	
Summary	All information of a single data entry is displayed in an individual view.	
Priority	2	
Preconditions	The user is either in a table view or is in a pin map view.	
Post-conditions	The information of a single data entry is displayed in an individual view.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	In a table view, the user clicks on a specific data entry to focus on it. In a pin map view, the user clicks on a specific pin to see the pins information.	
Main Scenario	Step	Action
	1	The system displays the information that the specific data entry has.
	2	The user closes the single crime view to continue using the view they were in before the single crime view window was created.
Extensions	Step	Branching Action
	None	
Open Issues	None	

## 2.14 Use Case 14

Number	14	
Name	Manage Timeline	
Summary	The user can adjust two separate sliders to filter data by time. The first slider controls dates and the second slider controls the 24 hours in a day. Each slider can be adjusted on either end. For example, the first slider can be adjusted to view data in 2013 and the second can be adjusted to view data from Noon to Midnight.	
Priority	5	
Preconditions	None	
Post-conditions	The dataset is filtered by the user chosen constraints involving time.	
Primary Actor	User	
Secondary Actor(s)	None	
Trigger(s)	The User manipulates one of the timeline sliders.	
Main Scenario	Step	Action
	1	The user manipulates one or both ends of either slider to choose a specific time interval for the data to be filtered to.
Extensions	Step	Branching Action
	1a	The user can click on the center of an interval on one of the sliders to move the interval around. For example, if the date slider interval is all of 2013, it can be dragged to be all of 2015 or if the hour slider has a 3 hour interval,

		that interval can be dragged to look at a 3 hour interval during a different time of day.
Open Issues	None	

## 3. Use Case Tests

Use case testing is a vital part of product validation and verification. The final product should be bug free and work how it is intended to work. These tests outline how each use case will be tested during the testing phase of the development process.

### 3.1 Use Case 1 Test - View Heat Map

Choose a subset of data, via filtering, to view as a heat map and click the view heat map button. Confirm that the map displays the information that was chosen. Check to see if further filtering of the map can be done via the heat map view. See Testing Report for further details and results.

### 3.2 Use Case 2 Test - View Pin Map

Choose a subset of data, via filtering, to view as a pin map and click the view pin map button. Confirm that the map displays the information that was chosen. Check to see if further filtering of the map can be done via the pin map view. See Testing Report for further details and results.

### 3.3 Use Case 3 Test - Map Zooming

Within a map view, attempt both zooming in and out as much as possible, within the hard coded limit. Check to make sure the map displays correctly at each zoom step and do this test on both the heat and pin maps. See Testing Report for further details and results.

### 3.4 Use Case 4 Test - Map Panning

Within a map view, pan the map around as much as possible, within the hard coded limit. Check to make sure the map displays correctly after small and large pans alike and do this test on both the heat and pin maps. See Testing Report for further details and results.

### 3.5 Use Case 5 Test - View Bar Graph

Choose a subset of data, via filtering, to view as a bar graph and click the view bar graph button. Specify which part of the data subset will fit to the X and Y axes, then confirm that the bar graph displays showing the correct data and uses the correct inputs for the two axes. Do this test multiple times using different data subsets and choosing different X and Y axes. See Testing Report for further details and results.

### **3.6 Use Case 6 Test - View Line Graph**

Choose a subset of data, via filtering, to view as a line graph and click the view line graph button. Specify which part of the data subset will fit to the X and Y axes and the number of trend lines to view based off of the chosen Y axis. Confirm that the line graph displays showing the correct data and uses the correct inputs for the two axes along with the chosen number and type of trend lines. Do this test multiple times using different data subsets and choosing different X and Y axes along with a different number of trend lines. See Testing Report for further details and results.

### **3.7 Use Case 7 Test - View Pie Graph**

Choose a subset of data, via filtering, to view as a pie graph and click the view pie graph button. Specify which part of the data subset will be the category to base the pie graph on e.g. Description leads to a pie graph based on type of crime. Confirm that the pie graph displays showing the correct data and uses the correct categorical input. Do this test multiple times using different data subsets and choosing different categories. See Testing Report for further details and results.

### **3.8 Use Case 8 Test - View Tables**

Confirm that one can enter into a table view. Confirm that the table displays data properly. See Testing Report for further details and results.

### **3.9 Use Case 9 Test - Sort Tabular Data**

Within a table view, click on the header of a column to sort the data alphabetically or numerically depending on the chosen column. Confirm that an alphabetical sort based on a specific column works by displaying the data sorted from A to Z. Click again on the same column to resort from Z to A. Using a different column, confirm that a numerical sort works by displaying the data sorted from least to greatest. Click again on the same column to resort from greatest to least. Do this test with every available column. See Testing Report for further details and results.

### **3.10 Use Case 10 Test - Filter Data**

Using checkboxes in a sidebar, globally filter the data into a subset by using categories such as: CrimeCode, Location, Description, Inside/Outside, Weapon, Post, District, Neighborhood, Longitude, Latitude, Location1 (Latitude, Longitude), Premise (Type of Location), or Total Incidents. Be thorough by testing each and every category individually and do various tests using multiple or potentially all of the filter options. Confirm that the data subset updates based off of the filters that are active. Note that filtering based on time is done via the timeline. See Testing Report for further details and results.

### **3.11 Use Case 11 Test - Adjust Columns in Table View**

Within a table view, click and drag on the lines dividing columns in order to resize them. Attempt to make columns smaller and larger. Test the limits of how small or how large a column can be. See Testing Report for further details and results.

### **3.12 Use Case 12 Test - Pagination in Table View**

Within a table view, click on the button that enables pagination and select how many data entries are to be shown on a single page from the given options. Confirm that the table view updates to only show the chosen number of data entries and do this test for each option. See Testing Report for further details and results.

### **3.13 Use Case 13 Test - View Single Crime**

While viewing a pin map, click on a single pin or while viewing a table, click on a single row entry to see the details of that specific crime. Confirm that clicking on those elements brings up accurate details about the crime. See Testing Report for further details and results.

### **3.14 Use Case 14 Test - Manage Timeline**

Confirm that both timeline sliders can be manipulated via both ends and that a chosen interval can be dragged around to maintain the given interval, but to change the time being considered for filtering. Be thorough by checking various different intervals. Also check the extremes by attempting the largest and smallest interval possible on each slider. While the above is being tested, confirm that the data subset updates based off of the different time intervals being used as filters. See Testing Report for further details and results.

## 4. Non-Functional Requirements

#	Item	Priority 1 (Lowest) to 5 (Highest)
1	Application front end shall be written in JavaScript and HTML.	4
2	Application shall use a code style checker.	4
3	Application shall have a package manager	4
4	Application should be able to manage large amounts of data up to 500,000 entries	4
5	Applications back end shall be written in node.js	4
6	Application shall be available through the web	2
7	Application shall be hosted on a linux server	2
8	Code shall have revision control	3
9	Application should be compatible with firefox and chrome	4
10	Application shall use MongoDB document database	3
11	Webpage shall render within 5 seconds	4



## 5. User Interfaces

See “User Interface Design Document for Crime Visualization System”

## 6. Deliverables

All content delivered to the client will be electronically delivered digital copies. Hard copies of the deliverables will only be upon request.

### Deliverables:

- Systems Requirement Specification.
- System Design Documents.
- User Interface Design Document.
- Source Code.
- Software required for installation of the application.
- Test Reports
- Copies of all Bi-weekly Status Reports.
- Administrator Manual.

### Delivery Date:

03/05/2018  
04/04/2018  
04/04/2018  
04/25/2018  
04/25/2018  
04/25/2018  
05/07/2018  
05/07/2018

### Addition Items that will be delivered:

- User Manual.

05/07/2018

## 7. Open Issues

The Application currently does not have any open issues because it is not in the implementation phase yet.

# Appendix A - Agreement Between Customer and Contractor

The customer agrees to a crime visualization system that allows the user to filter data in a variety of forms (maps, graphs, tables) relative to a chosen time frame. Use cases are included in the functional requirements section above detailing the behavior between the system and user.

When future changes to this document occur a drafted new document shall be created. An electronic copy of both versions will be presented to the client for review. Upon approval, the draft will be finalized and signed off by both parties.

## Client

Name Michael Smolyak Date 03/05/2018

Print



Name \_\_\_\_\_ Date 03/05/2018

Signature

## Team

Name \_\_\_\_\_ Date \_\_\_\_\_

Print

Name \_\_\_\_\_ Date \_\_\_\_\_

Signature

Name \_\_\_\_\_ Date \_\_\_\_\_

Print

Name \_\_\_\_\_ Date \_\_\_\_\_

Signature

Name \_\_\_\_\_ Date \_\_\_\_\_

Print

Name \_\_\_\_\_ Date \_\_\_\_\_

Signature

Name \_\_\_\_\_ Date \_\_\_\_\_

Print

Name \_\_\_\_\_ Date \_\_\_\_\_

Signature

## Appendix B - Team Review Sign-off

This document has been collaboratively written by all members of the team. In addition, all team members have reviewed this document and agree on both the content and the format. Any disagreements or concerns are addressed in team comments below.

### Team

Name \_\_\_\_\_ Date \_\_\_\_\_

Print

Name \_\_\_\_\_ Date \_\_\_\_\_

Signature

Comments

---

---

Name \_\_\_\_\_ Date \_\_\_\_\_

Print

Name \_\_\_\_\_ Date \_\_\_\_\_

Signature

Comments

---

---

Name \_\_\_\_\_ Date \_\_\_\_\_

Print

Name \_\_\_\_\_ Date \_\_\_\_\_

Signature

Comments

---

---

Name \_\_\_\_\_ Date \_\_\_\_\_

Print

Name \_\_\_\_\_ Date \_\_\_\_\_

Signature

Comments

---

---

## Appendix C - Document Contributions

Gregory Mayer was the leader on writing this document contributing the use case diagram and the non-functional requirements, and the deliverables sections. Andrew McLamb wrote the introduction section of the requirements. Proofreading and formatting was done by Nicholas Sorauf. The team divided up, between all members, the use cases and testing to ensure agreement on the requirements. The team also reviewed the overall layout of the document.