

S.A.N.T.A

System Analysing Non-
ethical Transgressions of
Adversaries

SENG202

Deliverable Four

Team 5:

Sam Cassidy
Sarah Jackson
Dominic Jarvis
Emily Price
Vincent Reilly

Submission Date: 06-10-15

Abstract

SANTA (System Analysing Non-ethical Transgressions of Adversaries) is an independent Java application developed for users who wish to view and analyse criminal activity in their area. The application includes functionality for users to view crime trends via graphs, reports and maps, as well as allowing the user to import, export, edit, and create crime records. SANTA is unique to each user as it works with local data stored on the user's computer. This allows each user to only view information which is relevant to them.

SANTA has a wide range of stakeholders, which include not only the end users but also the development team, the suppliers of crime data, investors, and staff responsible for the technology which SANTA will be implementing.

The application has several specific use cases which are combined to create rich user interactions. These use cases fall under the three separate categories of viewing data, analysing data, and managing data. Each of these categories contains many more precise use cases.

The development of software comes with certain risks which have also been analysed. These risks have been assessed to understand how they can best be managed and, if possible, avoided.

Technical diagrams have been created to show how SANTA will function, as well as non-technical diagrams which give an overview of what functionality the application consists of.

Table of Contents

Change Log	4
1. System Context and Relevant Business Information	5
1.1 System Context	5
1.2 Business Vision and Rationale	7
2. Requirements	11
2.1 Stakeholders and Concerns	11
2.2 User Stories and Use Cases	13
2.3 Sequence Diagrams for Major Use Cases	20
2. 4 Functional Requirements	23
2.5 Quality Requirements	24
2.5.1 Commercial	24
2.5.2 Technical	24
3. Domain Model	26
4. Acceptance Tests	26
5. GUI Prototypes	31
6. Risk Assessment	35
7. Project Plan	38
8. Data Model	42
9. Deployment Model	42
10. UML Class Diagram	44
11. Prototype Description	45
12. Testing Procedures	49
13. Future Advice for SENG Students	58
14. References	58
15. Appendix	59

Change Log

The change log is displayed in **Table 1**, and outlines which sections have been modified from Deliverable 2. “N/A” implies it was introduced in this deliverable, and was not in the criteria for deliverable 2.

Section Number	Section Name	Description of Change
1.	System Context and Relevant Business Information	
1.1	System Context	
1.2	Business Vision and Rationale	
2.	Requirements	
2.1	Stakeholders and Concerns	
2.2	User Stories and Use Cases	
2.3	Sequence Diagrams for Major Use Cases	
2. 4	Functional Requirements	
2.5	Quality Requirements	
2.5.1	Commercial	
2.5.2	Technical	
3.	Domain Model	
4.	Acceptance Tests	-irrelevant acceptance tests removed
5.	GUI Prototypes	
6.	Risk Assessment	
7.	Project Plan	
8.	Data Model	

9.	Deployment Model	
10.	UML Class Diagram	-Updated to include new classes that were implemented with the beta release
11.	Prototype Description	-Updated to describe the final version. -Explains all of the features that have been updated and implemented
12.	Testing Procedures	Testing updated to remain consistent with the final release.
13.	Future Advice for SENG Students	NA
14.	References	
15.	Appendix	

Table 1: Describes the updates made to the report from Phase 1

1. System Context and Business Information

1.1 System Context

SANTA (System Analysing Non-ethical Transgressions of Adversaries) is a crime monitoring system that is able to record, monitor and analyse crime data for large cities. It is an open source application promoted by local governments and officials to benefit their communities. The application stores information and provides key statistics of crime committed in selected geographical areas of interest.

The user is able to upload, export, and submit crime data of their interests to perform their own analyses using the various built in features. Some of these features include filtering the data based on key details, mapping where a crime was committed and many more. This allows users to gain insightful knowledge of crime occurrences in their area of residence (and/or other area of interest) to promote safe practice, self-awareness and security.

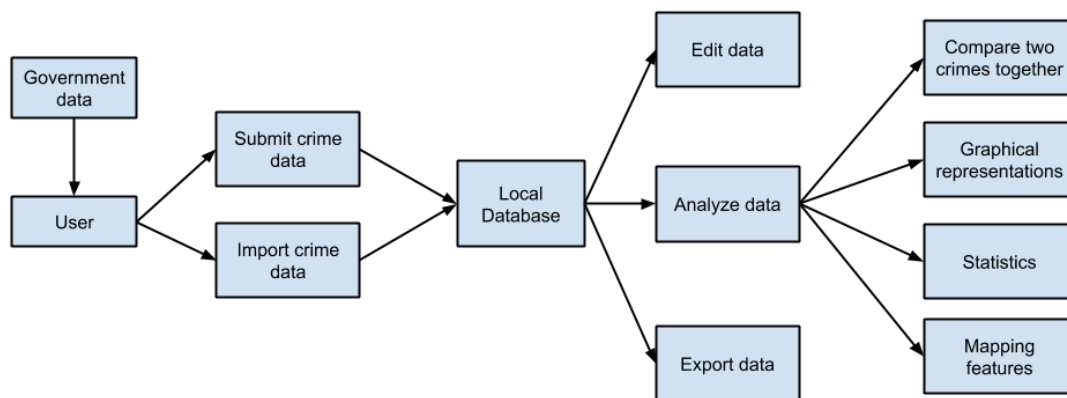


Figure 1: Non-technical diagram of SANTA and its basic features

The application is able to be used by a diverse range of users for their needs and purposes. Here are some examples of users and why they would use it listed below:

- **Residents:** People living in local areas/communities may wish to know crime statistics in their area. Allows them to assess how safe areas are and make decisions accordingly. May wish to know about certain areas out of curiosity. Able to submit crimes they have witnessed in their area, or have heard of from their peers
- **Police Force:** May be able to use it for educational purposes. As they have access to more sensitive information, perhaps they could upload crimes so the public is aware, but without disclosing any confidential information.
- **Local Community Watch Programs:** Able to raise awareness to their community about geographical areas that have statistically more crime. Able to conduct safer practice and utilize various approaches depending on the area and what crimes most often occur there. Able to submit crimes or suspicious activity they have encountered. Members of the Watch could potentially share/export data with other members so they all have the same data set.
- **Other government officials:** Such as politicians interested in crime prevention. Can propose funding to implement programs to reduce crime in areas that are worse than others. Can enforce community protection and other safety practices for areas more predisposed to crime. Can perhaps use it to educate residents and other members of the community
- **Businesses:** May wish to know which areas have more crime and the likelihood on how it may affect their business strategy. For example, a retail store will most likely not have their store in the part of the city

where theft is most abundant, or construction companies with intents of particular communities may choose safer areas where there is statistically less crime.

- **Travellers and potential new residents:** Travellers and new residents may be potentially interested so they can assess which parts of a city are safer than others.
- **Other Statistical Analysts and Media:** Allows other parties to gain insightful information of crime statistics in geographical areas of interest for their intents and purposes. Able to submit crimes they have attained from other sources.
- **Advertising firms:** Can target their markets to certain demographic areas depending on the statistics. For example, in areas where drink driving is statistically higher, preventative drink driving advertisements may be beneficial in that particular area.
- **City Planners:** Can use the application to assess which areas are more predisposed to certain crimes in order to introduce preventative measures, such as installing more cameras and street lighting in locations where there have been statistically more assault cases.

1.2 Business Vision and Rationale

Business Opportunities

Once the application is developed it will open up many business opportunities and has the potential to become very successful. Some business opportunities are listed below:

- As SANTA is intended to be an open data application, and is to be promoted by the government, the likelihood of government funding is very high
- Could potentially build towards government commission. This implies developing the application and selling it to the government (or perhaps another interested party).
- If the application were to remain free to users, money inflow would mainly have to come from advertisements. Perhaps government based advertisements, as they encourage this particular type of application.
- As SANTA functions based on user data, it can theoretically function for any user worldwide. It could be successfully used in any English

speaking country and potentially other languages if multi-lingual options are introduced.

- Property investors may take a particular interest in the application. Certain geographical areas will have a higher value than others depending on the crime rate. May use the application so they have an advantage in the market and use it to decide when and where to invest by observing trends over time.
- Police departments and other crime officials may take a particular interest in the application, because they could use it to educate the public and promote awareness of crime frequency in particular areas without disclosing sensitive information.
- Community watch and residents may contribute/donate to the costs of SANTA to maintain the services it provides.
- There are various business opportunities in big cities, where there is a demand for statistical analysis on crime – this information can be used by various parties for various reasons.

High Level Services Provided:

- SANTA will be easy to use and navigate because the GUI (graphical user interface) will have informative and intuitive labels. All of the main tools and features will be easy to find using the labels.
- A user will be able to import or submit crime data manually for their own analyses. A user will also be able to export from their data set to share with others.
- SANTA will provide an in-depth raw data viewer, where a user can view and filter information on their crime data sets. It summarizes your data set to enable easy analysis.
- SANTA can be used to generate reports and graphs on selected data sets to help users understand crime trends and frequencies of crimes in selected areas.
- Using Google Maps, SANTA allows users to visualize on a map where crime(s) have occurred in a particular area.

Unique Selling points and Competitors:

SANTA is unique with how a user can download the application and personalize the data sets accordingly to their own interests and preferences. Unlike other competitors, such as NZ.stat and City of Chicago, a user can load selected crime data based on their interests, rather than having large pre-sets of data taking up excess memory. Not only can officials submit

information, but residential users can too. These submissions can be exported from their application if they wish.

NZ.stat does not provide graphical representations of the data, so frequencies of crimes or trends over selected periods can not be plotted. SANTA provides the viewing of crime statistics, specified by selected fields and parameters, in graphs and mapping features.

Because the application is going to be coded in Java, it has the potential to be modified for release on various platforms, such as mobile phones or tablets. Other crime databases such as NZ.stat and City of Chicago only offer online webpage services. There are similar applications such as Crime Spy UK, but these applications will only cover a pre-set geographical area. SANTA enables the user to graphically visualize on a map any crime data set they have submitted/imported to their application (if the data set is provided with a location). This gives SANTA the potential to be used worldwide, and gives the user flexibility to view data that they are interested in.

Positive and Negative Aspects

SANTA has various positive aspects that make it unique to other competitors. While this is the case, we have also tried to consider as many potential downsides, so we can work to either completely eliminate or minimise the effects of these downsides. These aspects are displayed below in **Table 1**.

Positive	Negative
SANTA will have a much greater influx of exportable crime data, as all users can upload their own submissions.	Submissions to the database may not be accurate without an official report overseeing it. Some users may conduct dishonest practice or insert incorrect details, and export the data.
As SANTA uses open data, anyone can export information at their own whim.	As SANTA uses open data, some users may download information that may not be accurate that has been exported by other sources.
SANTA is able to be personalized with the data that the user is interested in. This enables the application to potentially be used worldwide in geographical areas that have recorded	If SANTA can be used worldwide, there may be implications with quality and maintenance over different servers. Also, other nations may differ in policies with information they wish to disclose to

crime data.	the public.
The application will have various filters and features to view and analyze crime datasets of interest presented in ways such as profiles, tables, graphs and trendlines.	The more features, the more complex the application becomes. As more features are introduced, keeping it as user friendly as initially intended may be unattainable.
SANTA is orientated being easy to use. The application intends to be user friendly by being easy to navigate	Various users may have different wants or needs. It may not be possible to create an interface that will satisfy every user. Some users may not utilize/need all of the features.
Because the SANTA application will be coded in Java, it has the potential to be used on a variety of platforms and OS's, such as computers, mobile phones and tablets. It is not restricted to a website unlike some competitors.	This requires the application to be functional on various OS platforms. Alterations or different versions of the application may have to made so it can be supported on certain devices. This will also increase the required maintenance of the application.
SANTA can be used offline to analyse pre-loaded crime data.	Some features won't be available in offline mode. Users won't be able to import/export data from the internet, and some mapping features may be unavailable.

Table 2: Describes the positive and negative aspects of SANTA

Why is it worth developing?

SANTA is worth developing because the overall intent is to benefit all members of the community. It will:

- collectively store crimes so they can be analysed and interpreted easily. It will enable trends over time to be observed, and which areas are most predisposed to crime.
- to enable users access to statistics presented in an insightful manner, and to provide information on crime in geographical areas of interest
- assist officials, neighbourhood watch and residents to improve certain communities if need be.
- raise awareness to users to potentially reduce the risk/chance of a crime occurring, and encourage safe practice. It allows users to make decisions affecting their livelihood and current situation based on the provided statistics and analysis

- to enable users to filter through certain crime information to benefit their interests

2. Requirements

2.1 Stakeholders and Concerns

For the SANTA project a stakeholder is anyone who is affected in some way by the project. The table below lists SANTA's stakeholders, and shows the priority and concerns of each. Concern priority is ranged from 1 (low importance) to 3 (critical to the project).

ID	Priority	Stakeholder	Concerns	Concern Priority
SH1	3	Developers	Time	3
			Feasibility	3
			Maintainability	1
			Code Quality	2
			Documentation	2
SH2	1	Raw Data Suppliers	Misuse of data	1
SH3	3	Residential Users	Usability	3
			Unique features	2
			Technical errors	2
			Correct Calculations	3
			Responsiveness	2
			Trend Identification	3

SH4	3	Government Users	Trend Identification	3
			Usability	2
			Responsive	2
SH5	3	One Off User	Usability	2
			Responsive	2
			Trend Identification	3
SH6	1	Investors	Project Failure	1
SH7	1	UC Tech support	Assistant Time	1
SH8	2	SENG202 staff	Plagiarism	2
			Project Failure	3

Table 3: Shows the stakeholders and their concerns

Stakeholder Descriptions:

Developers: SENG 202 team 5 development team.

Raw Data Supplier: Local Governments that release crime data to the public.

Residential Users: Individual residents who use SANTA to monitor crime in their area.

Government Users: Users who want to understand crime trends in specific areas to educate residents or make decisions about resource allocation (streetlights and security cameras). These people include police departments and crime officials, city planners and neighbourhood watch groups.

One-off Users: Anyone who uses the application for single-use to get a quick understanding of crime trends in a city. These people include travellers, businesses, other statistical analysts, potential new residents, media, advertising firms and property investors.

Investors: People or companies who may be interested in investing in SANTA in the future.

UC Tech support: The university staff managing the version control systems SANTA's developers will utilise.

SENG202 staff: The lecturers and tutors that run the SENG202 course and oversee the student activity.

End-users have been categorized into 3 main groups; government users, residential users and one-off users. These are based on our assumptions of

what the average individual of each group would use the app for. Each end-user will have access to the same features of the application, and will use them accordingly for their intents and purposes. Some end-users may or may not use the same features as others.

Concerns:

Time: Time required to complete the project.

Feasibility: The likelihood of the promised features being completed by the deadline.

Maintainability: The ease of adding new features and extending existing features.

Code Quality: SANTA's source code consistently follows a set of coding rules and conventions for readability.

Documentation: Code is explained throughout the development process through use of JavaDoc.

Misuse of data: Where the data provided is not used as intended within the application. e.g. A user submits/updates a data file with incorrect information

Usability: The difficulty of using and/or navigating the application and its features.

Unique features: The program provides new services as well as improving upon existing features from similar applications.

Technical errors: The application contains errors to do with functionality e.g the program may force quit without prompt when attempting to perform a simple instruction.

Correct Calculations: The calculated data displayed in the system is correct and accurate.

Responsiveness: The program has a fast action-reaction time in most cases, and provides feedback to the user when it will take a long time completing certain actions.

Trend Identification: How well the program summarizes crime trends and conveys them to the end users.

Profit: How much revenue the program may potentially make.

Assistant Time: How much time is required to fix technical issues.

Plagiarism: Where a group or individual's work has been taken from another source, and they declare it as their own original work.

Project Failure: The project is not completed by the due date, or does not perform to its expectations.

2.2 User Stories and Use Cases

Use cases provide an overview of how a program is expected to function. Each use case may have several different flows, or 'paths' the actors can take to reach the end goal. The use cases for SANTA have been split into three high-level sections: View data, analyse data, and manage data. These are

individually split into smaller, more specific use cases and it is likely that these use cases will be combined by SANTA's users to create relevant interactions.

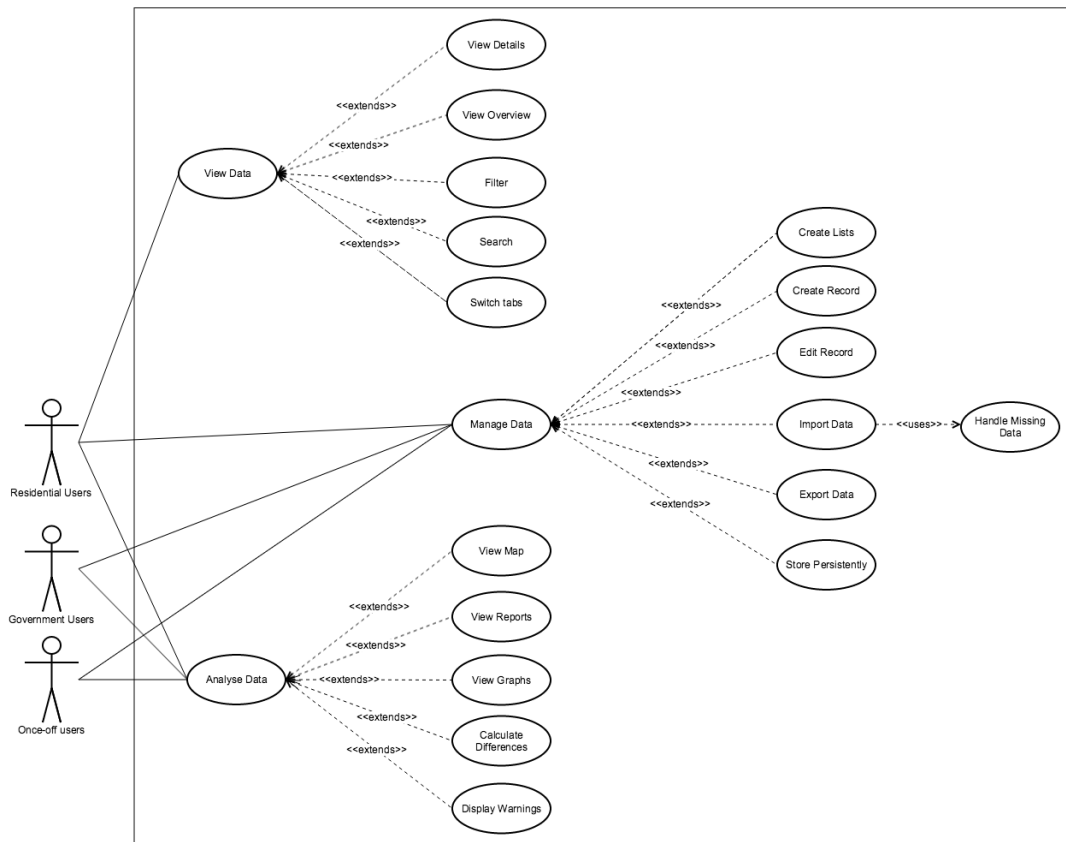


Figure 2: High-level use case diagram
View Data

UC1: View Overview of Crime Data:

Actors: End-user: Residents.

Goal: View all crime record data in a single list

Precondition: An appropriate data file exists on the user's computer

Basic flow: User opens program > Imports .csv data file into program > Shown a loading screen while data imports > Crime records from the file displays in the viewing panel

Alternative Flows:

1. User opens program > Data is still loaded into program from a previous session > Data from previously session displays in a viewing panel
2. User opens program > Loads a non-compatible file type > Error Message displayed

Postcondition: Data has been loaded and is viewable

UC2: View Expanded Record:

Actors: End-user: Residents.

Goal: Load an individual data record in its own frame/window

Precondition: A loaded data file is being displayed in data viewer

Basic flow: User selects a record from raw data viewer > empty fields in a side window displaying detailed information about the record > user clicks the close button on the popup to get back to data view.

Alternative Flows: User is already viewing expanded data viewer > navigates to next record by navigation bar > next record is shown in expanded viewer

Postcondition: User has viewed data relating to specific record and returned to previous screen

UC3: Filter Data:

Actors: End-users: Residents, police department, city planners, property investors, travellers.

Goal: View data grouped by specific information

Precondition: A loaded data file is being displayed in the program's data viewer

Basic flow: User selects a filter type > Data meeting that criteria is displayed > user can expand view of individual records

Alternative Flows: User selects a filter from a drop down box > There is no data meeting that criteria > Error message displayed

Postcondition: Appropriate results for the user's selected filters are shown on screen

UC4: Search Data:

Actors: End-users: Residents, police departments

Goal: View data based on search terms

Precondition: Appropriate data is displayed in the program, each individual crime record has a unique ID

Basic flow: User enters an ID to search> the crime record relating to the searched ID is displayed in the expanded viewing mode > user selects a "close search" option

Alternative flows: User enters an ID to search > there is no data related to that ID > the user is shown a message suggesting they try different search criteria

Postcondition: Record matching searched ID is shown on screen in expanded viewer

UC5: Switch Tabs:

Actors: End-users: Residents, police departments, city planners, property investors, travellers.

Goal: To move from viewing one tab of the application to another (where each tab contains a different function)

Precondition: Crime data has been loaded into the system

Basic flow: User is viewing one tab > selects another tab with mouse click > view swaps to new tab instantly

Alternative flows: -

Postcondition: User is viewing the tab they selected to switch to

Analyse Data

UC6: Draw Graph:

Actors: End-users: Residents, police departments, city planners, property investors, travellers.

Goal: To draw a graph based on user selected data

Precondition: Appropriate data file has been loaded into program

Basic flow: User selects data to compare > selects statistics section of program > fills in appropriate text fields > submits request> graphs displayed on screen

Alternative flows:

1. User selects statistics section of program > selects data to compare > fills in appropriate text fields > submits request > graphs displayed on screen
2. User selects statistics section of program > attempts to submit request > is prompted to select data and appropriate text
3. User selects statistics section of program > fills in text fields > selects data > submits request > graphs displayed on screen

Postcondition: Graphs are displayed on screen

UC7: View Crime Data on Map:

Actors: End-users: Residents, neighbourhood watch, city planners, property investors, travellers.

Goal: Display specific crime records on a map

Precondition: Application has appropriate data file loaded, data file contains latitudes and longitudes of crime records, and the application has internet access.

Basic flow: User selects data to display > selects map section of program> map is displayed on screen, each data point representing a single crime record

Alternative flows: User opens record in expanded viewer > selects “to map” function > single data point is shown on the map, showing the location of the selected crime record

Postcondition: Data is shown on a map using Google Maps API

UC8: View Crime Statistics:

Actors: End-users: Residents, police departments, city planners, property investors, travellers.

Goal: To create a report of crime statistics

Precondition: Application has data file loaded

Basic flow: User selects data to display > selects “statistics” option > fills in appropriate text files > selects “report” function > presses “GO” > report displayed on screen

Alternative flows:

1. User selects statistics section of program > selects data to display > fills in appropriate text fields > submits request > report displayed on screen
2. User selects statistics section of program > attempts to submit request > is prompted to select data and appropriate text
3. User selects statistics section of program > fills in text fields > selects data > submits request > report displayed on screen

Postcondition: Data is shown in a report format

UC9: Calculate Differences in Data:

Actors: End-users: residents

Goal: Difference between two subsequent data points is calculated correctly and displayed

Precondition: A file containing two or more crime records exists

Basic flow: File is imported > calculations between each set of two subsequent points are made > calculation results are shown in expanded data viewer

Alternative flows: File is imported > records are missing location or time information > the calculations for these records return a ‘null’ value

Postcondition: Results of calculations can be viewed in expanded data viewer

UC10: Display Warnings:

Actors: End-users: Residents

Goal: Display warnings about areas with high crime rates

Precondition: Data file has been loaded with a large number of entries

Basic flow: User navigates to ‘warnings’ tab > A textual summary of areas with high crime rates is displayed

Alternative flows: -

Postcondition: Warnings have been displayed on screen

Handle Data

UC11: Import Data File:

Actors: All end-users, as well as raw data suppliers

Goal: To successfully load a pre-existing file of crime data to the program

Precondition: Application is open, no data has been loaded

Basic flow: User selects File > Import Data > Selects File > File shows up in raw viewer

Alternative flows:

1. File > Import Data > Selects File > File is corrupt > Error message displayed
2. File > Import Data > Selects inappropriate file type > Error message displayed
3. File > Import Data > Selects appropriate file type > File is missing some information (specific fields on specific records) > Program fills in missing fields with null > Data is displayed in viewer, popup notification is shown stating that some data fields were empty

Postcondition: Loaded data shows in the data viewer

UC12: Add New Record:

Actors: End-users: residents

Goal: To add a new crime record to an existing data file

Precondition: An appropriate data file has been loaded into the application

Basic flow: User selects a “create record” button > A screen with blank fields is shown > user enters data > submits data > unique ID is generated > data is written to file and shows on screen

Alternative flows: User selects “create record” > A screen with blank fields is shown > user enters some data, but leaves some fields blank > submits data > is prompted to fill in blank data > blank data is set to null > unique ID generated > data written to file and shows on screen

Postcondition: The loaded file has been appended with the new record and it shows in the raw data viewer

UC13: Edit a Record:

Actors: End-users: residents

Goal: To alter the contents of an existing record

Precondition: An existing data file has been loaded into the application

Basic flow: User selects a record to view as expanded record > selects “Edit” button on this frame > Data fields become editable > User edits one or more fields > selects “save changes” > data is saved to file > record returns to view mode

Alternative flows:

1. User selects a record to view as expanded record > selects “Edit” button on this frame > Data fields become editable > User does not change anything > selects “save changes” > Pop-up asks user to reconsider empty data fields> record returns to view mode
2. User selects a record to view as expanded record > selects “Edit” button on this frame > Data fields become editable > User changes data > selects “cancel” > User prompted to save changes > user clicks “save changes” > data is saved to file > record returns to view mode
3. User selects a record to view as expanded record > selects “Edit” button on this frame > Certain data fields become editable > User changes data > selects “cancel” > User prompted to save changes > user clicks “Do not save” > record returns to view mode
4. User selects a record to view as expanded record > selects “Edit” button on this frame > Data fields become editable > User does not change anything > selects “cancel” > record returns to view mode

Postcondition: The edited record has been updated and saved to reflect the user’s changes

UC14: Export Data:

Actors: End-users: residents, police departments

Goal: Save data in a shareable file for other users to load into their own application

Precondition: Data exists within the application

Basic flow: User selects file > export data > file browser appears to ask user where to save the data file > a copy of the data file is made to this location

Alternative flows: -

Postcondition: A copy of the user’s current data file has been saved to a specified location

UC15: Handle Missing Data:

Actors: End-users: residents

Goal: Handle data with empty fields

Precondition: A data file has been selected to load which contains one or more crime records with missing data in certain fields

Basic flow: The system locates the records with missing data > fills these empty fields with a ‘null’ variable > alerts the user via pop-up box that missing fields were found. Methods handle data accordingly

Alternative flows: -

Postcondition: Data is loaded into program, missing fields of incomplete data have been filled with ‘null’.

UC16: Store Data Persistently:

Actors: End-users: residents

Goal: Data is saved to file as it is updated

Precondition: Data file is loaded into application

Basic flow: User edits a record > selects an interaction to save their changes > the changes made to the record are saved to the file in which the data is stored. If user drifts from window when changes are made, prompt window asks if they want to save unsaved changes

Alternative flows: User creates a record > record is appended to the original data file

Postcondition: The data file has been altered to reflect the user's changes.

UC17: Separating Data into Lists:

Actors: End users: residents, government officials

Goal: To separate data into lists for later reference or analysis.

Precondition: There is a file that exists and is imported in the program.

Basic flow: User creates list > user selects imported file > user selects 'add to list' > duplicate of specified data is created and stored in list specified by the user.

Alternative flows: -

Postcondition: Specified data exists in main database and list

2.3. Sequence Diagrams for Major Use Cases

UC3: Filter Data

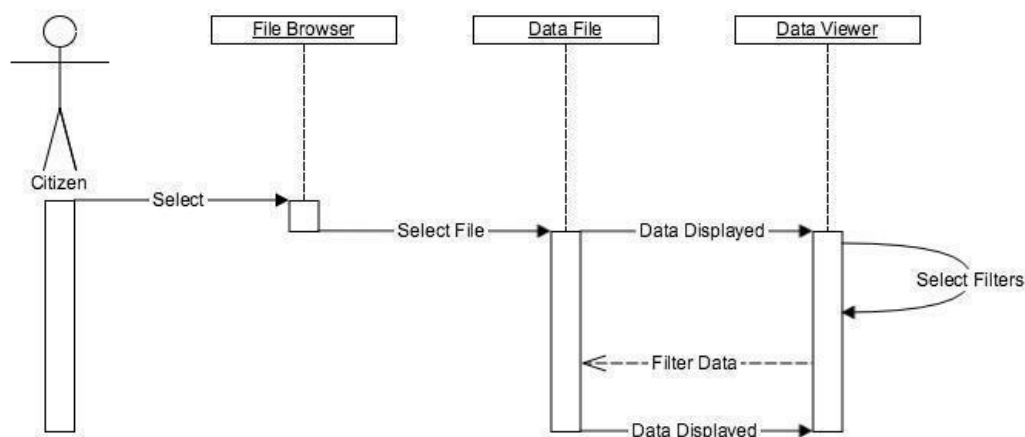


Figure 3: Use case UC3, displaying the sequence diagram for Filter Data

UC6: Draw Graph

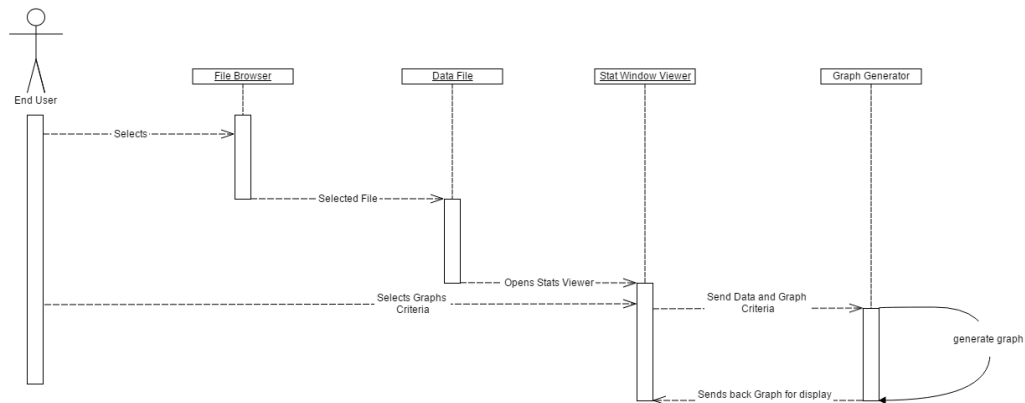


Figure 4: Use case UC6, displaying the sequence diagram for Creating a New Record

UC7: View Crime Data on Map

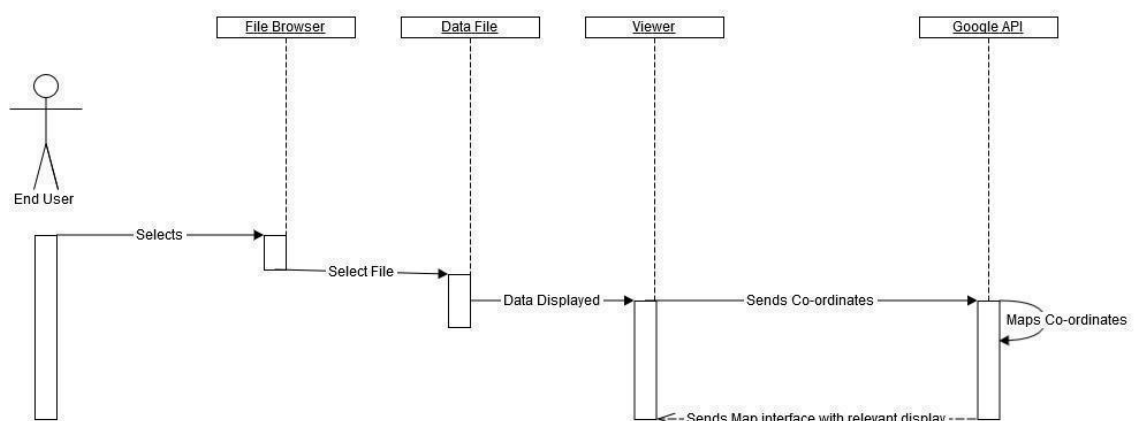


Figure 5: Use case UC7, displaying the sequence diagram for View Crime Data on Map

UC11: Import Data File

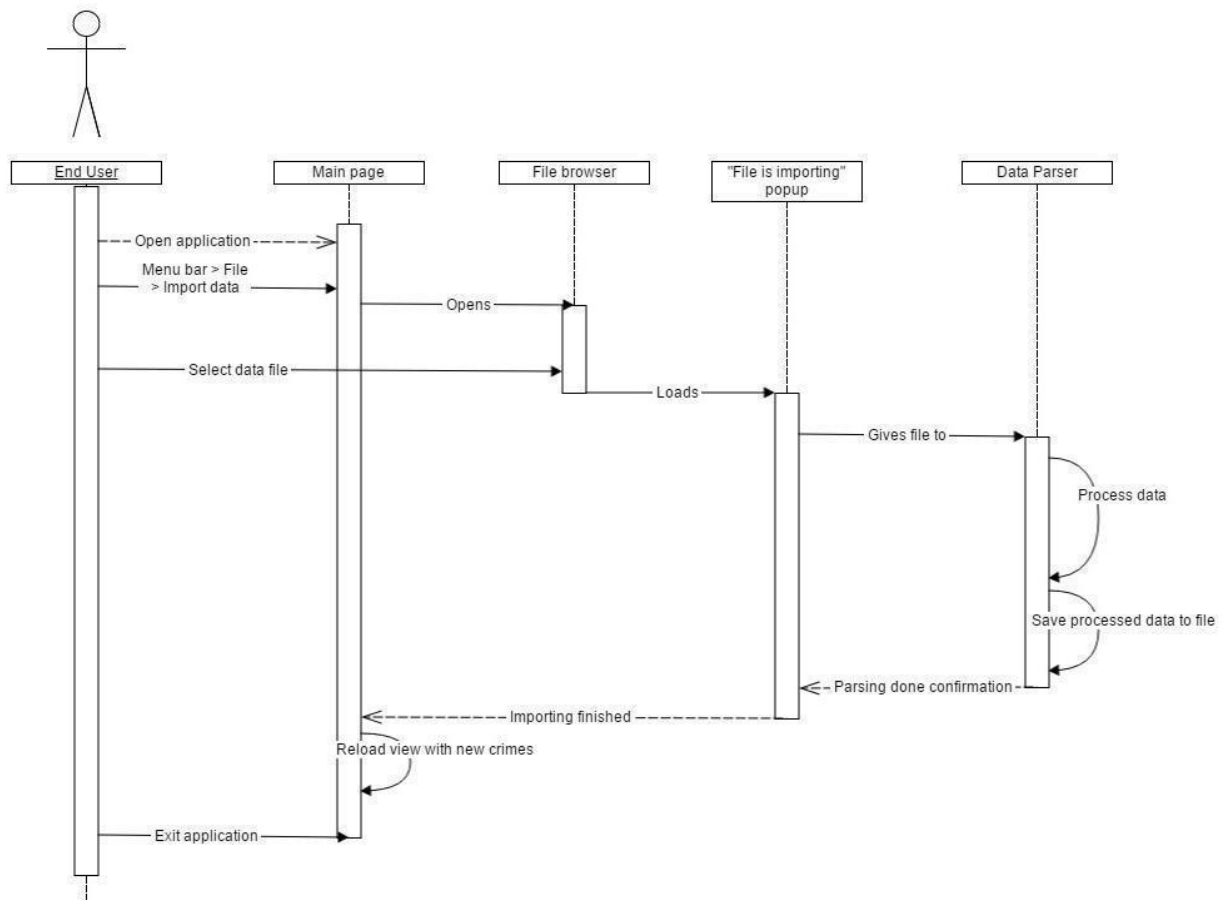


Figure 6: Use case UC11, displaying the sequence diagram for Import Data File

UC12: Creating a New Record

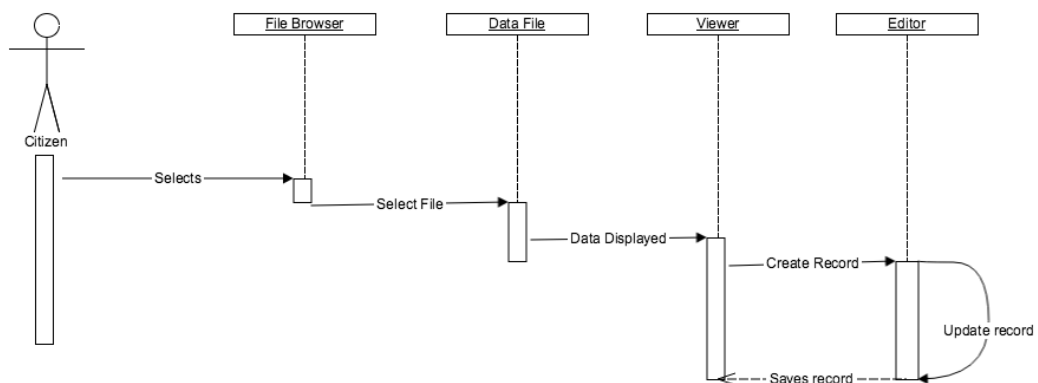


Figure 7: Use case UC12, displaying the sequence diagram for Creating a New Record

2.4. Functional Requirements

ID	Description	Stakeholder	Priority	Use Case ID
FR1	User is able to import existing data	SH3,SH4,SH5	10	UC11
FR2	System has a main tab (GUI)	SH3	10	UC1
FR3	Program displays data in a textual format (raw data viewer)	SH3	10	UC1
FR4	Data can be filtered by date	SH3,SH4	9	UC3
FR5	Data can be filtered by crime type	SH3, SH4	9	UC3
FR6	Time difference of two subsequent crimes can be calculated	SH3	8	UC9
FR7	Physical distance between two subsequent crimes can be calculated	SH3	8	UC9
FR8	Data can be filtered by location, arrest status, domestic violence status	SH3,SH4	8	UC3
FR9	Single data records can be viewed in-depth	SH3	9	UC2
FR10	User can store data in multiple lists	SH3, SH4, SH5	9	UC17
FR11	Data is permanently stored in a text file	SH3, SH4, SH5	7	UC16
FR12	Data is searchable by crime record ID	SH3, SH4	6	UC4
FR13	User can add a new crime record	SH3, SH4	8	UC12
FR14	Data is permanently stored in a database	SH3, SH4	5	UC16
FR15	Users can alter existing records, but the record ID and IUCR non-editable	SH3, SH4	7	UC13
FR16	User can view basic crime statistics	SH3, SH4, SH5	7	UC8
FR17	Users are presented with	SH3, SH4, SH5	6	UC10

	warnings			
FR18	User can graph crimes according to specified criteria	SH3, SH4, SH5	9	UC6
FR19	User can view crimes plotted on a map	SH3, SH4, SH5	10	UC7

Table 4: Displays the functional requirements with the stakeholder who would be affected, the priority of the requirement, and what use case it is associated with

2.5. Quality Requirements

2.5.1. Commercial

ID	Description	Stakeholder	Priority	Use Case ID
CQR1	Program functions the same on all main platforms and operating systems	SH1 ,SH3, SH4, SH5, SH8	8	-
CQR2	GUI is simple to navigate, not cluttered or confusing	SH1 ,SH3, SH4, SH5	7	-
CQR3	Program is written in grammatically correct English, makes sense to read	SH1 ,SH3, SH4, SH5, SH8	7	-

Table 5: Displays the commercial requirements with the stakeholder who would be affected, the priority of the requirement, and what use case it is associated with

2.5.2. Technical

ID	Description	Stakeholder	Priority	Use Case ID
TQR1	Loading animation shown while data is importing	SH3, SH4, SH5	4	UC11
TQR2	Switching between tabs appears instant - happens in less than 0.1 seconds	SH1 ,SH3, SH4, SH5, SH8	8	UC5
TQR3	Time taken to generate graphs, reports, maps does not exceed 10	SH1 ,SH3, SH4, SH5, SH8	9	UC6, UC7, UC8

	seconds			
TQR4	The program displays error messages instead of crashing	SH1 ,SH3, SH4, SH5	8	UC1, UC3.UC11
TQR5	Data files containing empty fields fill in 'null' for missing items	SH3, SH4, SH5	6	UC15
TQR6	Records with 'null' location variables are not included on maps	SH3, SH4	6	UC7
TQR7	Users are notified via popup box when loaded data has been corrupted	SH1 ,SH3, SH4, SH5, SH8	6	UC11

Table 6: Displays the technical requirements with the stakeholder who would be affected, the priority of the requirement, and what use case it is associated with

3. Domain Model

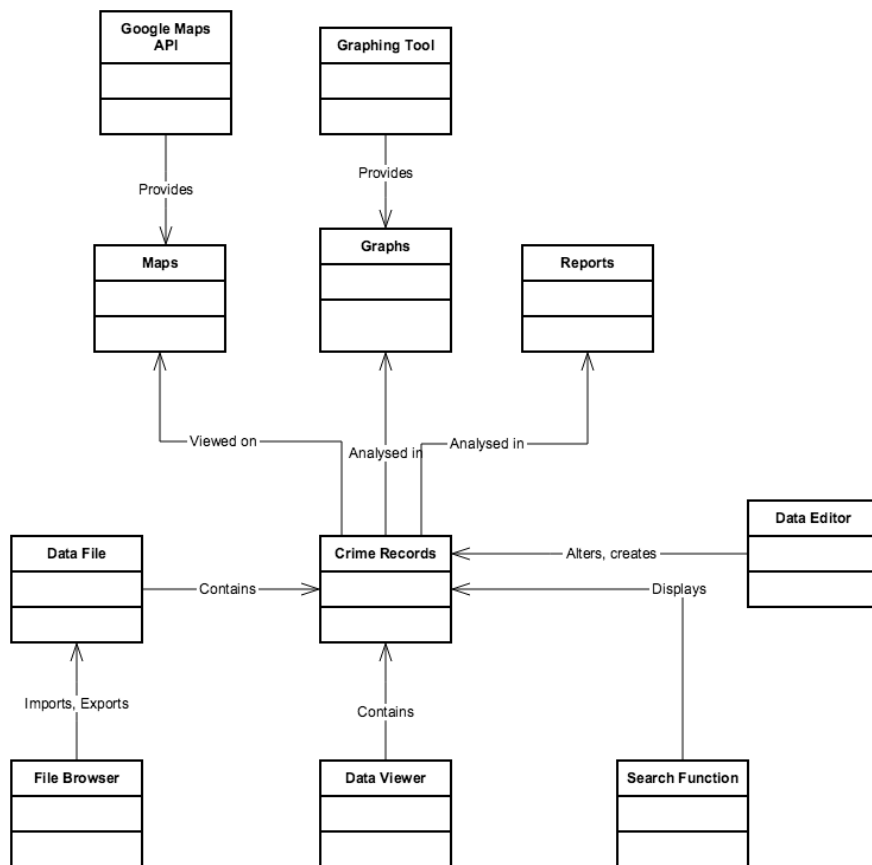


Figure 8: The domain model of SANTA

4. Acceptance Tests (textual)

Table 7 displays the tests SANTA needs to pass for it to be of acceptable functionality (as in, it runs how it should in reasonable time). The table elaborates on the importance of each acceptance test. It also references it to its use case, responsibility (Stake Holder) and feature package associated with it.

For each of the following priority scores, the remark has been given out of ten after conferring with team members as to their opinions on each requirement, and which are most important to work with the most functionality and extent, while which others may be cut away if time restraints become a pressing factor.

The responsibility for each test is determined by which users are likely to use the functions.

The number given in the 'Feature Package' column is the expected iteration/development phase that the Use Case/Requirement is likely to be implemented for, and therefore which iterations the Acceptance test may be run. For some of these acceptance tests, we anticipate changing the way the program accesses basic functions over the course of development, and so they may not be run in the same manner. These are therefore iteration dependant and are represented as such in the feature package column.

ID	AT1
Description	User selects a file to import in the file browser, and then clicks import to import the records.
Acceptance Criteria	Complete information about the crime records may be viewed in the data viewer, with the information/fields showing exactly as recorded in the data file.
Priority	10
Use Case/ Requirement	UC11, FR1
Responsibility	SH3, SH4, SH5
Feature Package	1-4
ID	AT2
Description	User opens up SANTA and imports a data file containing crime records as per UC11.
Acceptance Criteria	Data contained in the imported file is viewable, ordered by case number, in the main window, with the data exactly as it appears in the data file.
Priority	10
Use Case/ Requirement	UC1, FR3
Responsibility	SH3, SH4, SH5
Feature Package	1
ID	AT3 6
Description	User clicks on the changes filter mode to "domestic only". User changes date range to cover only the most recent week in the series of data imported. This date range can be determined by going to the file viewer and sorting through in order of Case Numbers.
Acceptance Criteria	Only entries with domestic incidents show in the data viewer. Only entries from the specified week show in the data viewer.
Priority	6
Use Case/ Requirement	UC3
Responsibility	SH3, SH4, SH5
Feature Package	1-4
ID	AT4
Description	User navigates to the second record, when files are sorted by Case Number in the file viewer, and takes note of the last two columns in the data viewer.

Acceptance Criteria	There is a non-editable field in the viewer displaying the difference in time and physical location between the previous crime and current crime, and the next crime and current crime. User checks that this time and location difference is correct by comparing with records both immediately above and below.
Priority	6
Use Case/ Requirement	UC9, FR6
Responsibility	SH3, SH4
Feature Package	2
ID	AT5
Description	User clicks on a crime record summary in the new data viewer.
Acceptance Criteria	A side panel containing all of the information as contained in the data file that the record was imported from appears over the main GUI.
Priority	8
Use Case/ Requirement	UC2, FR9
Responsibility	SH3, SH4, SH5
Feature Package	2-4
ID	AT6
Description	Click filter by type, and change type to assault.
Acceptance Criteria	Data viewer should only show crimes that are of type assault. User is to navigate the viewer and determine whether there are any files which fall outside of the bounds of the search parameters.
Priority	6
Use Case/ Requirement	UC3, FR8
Responsibility	SH3, SH4
Feature Package	2-4
ID	AT7
Description	User imports data into the SANTA program as per UC11. The user then closes the program.
Acceptance Criteria	The User reopens SANTA. The data which was imported is still viewable in the In-Depth Data Viewer.
Priority	7
Use Case/ Requirement	UC2, FR11
Responsibility	SH3, SH4
Feature Package	2-4
ID	AT8
Description	The user inputs the crime ID of the second crime in the data file into the editable search field and clicks the search button.
Acceptance Criteria	SANTA opens the edit panel for the user to either edit the crime record or simply view the record. The user checks that all entries are correct, as they should be, by comparing these results to the raw data from the file.

Priority	5
Use Case/ Requirement	UC4, FR12
Responsibility	SH3, SH4, SH5
Feature Package	2-4
ID	AT9
Description	The user filters the results by type, setting the filter flag to domestic. The user then clicks add data selection to custom list. The user enters the name "List of Domestic Incidents".
Acceptance Criteria	The new list has been created and appears on the left side of the data viewer under "Custom Lists". The new list has the correct name as entered by the user. The user checks that only the data with the domestic flag have been added to the "List of Domestic Incidents".
Priority	4
Use Case/ Requirement	UC17, FR10
Responsibility	SH3, SH4, SH5
Feature Package	2-4
ID	AT10
Description	User selects a record in the data viewer and clicks edit. User clicks the type of crime and changes to/from domestic. User tries to change Crime ID and IUCR. These fields should not be editable. Then clicks submit changes.
Acceptance Criteria	The user opens the same file they edited and checks that the fields have been changed.
Priority	7
Use Case/ Requirement	UC13, FR15
Responsibility	SH3, SH4
Feature Package	3-4
ID	AT11
Description	User clicks file then create record. User then enters data in the appropriate fields and clicks submit.
Acceptance Criteria	1. User checks to see that the record was created, by navigating to the creation ID in the data viewer. 2. User clicks on the record and opens up the record. User checks that the data is the same as what they entered
Priority	9
Use Case/ Requirement	UC12, FR13
Responsibility	SH3, SH4
Feature Package	3-4
ID	AT12
Description	User edits a record as per UC13, and then closes the program. User creates a record as per UC12, and then closes the program.
Acceptance	1. User reopens the program and then navigates to the record that was

Criteria	edited, using data viewer. User clicks on the file. User checks that the changes they made are still there. 2. User reopens the program and then navigates to the record that was created, using the data viewer. User clicks on the file. User checks that the record they created has the correct values in the appropriate fields.
Priority	7
Use Case/ Requirement	UC16, FR11, FR14
Responsibility	SH3, SH4
Feature Package	3-4
ID	AT13
Description	User clicks on the analysis tab at the top of the GUI, and then selects all data, and then clicks generate basic report.
Acceptance Criteria	A report is generated based on all the data currently in the system. This report shows the top three most common crimes in the data set in descending order.
Priority	6
Use Case/ Requirement	UC8, FR16
Responsibility	SH3, SH4, SH5
Feature Package	3-4
ID	AT14
Description	User clicks on the warning at the top of the GUI.
Acceptance Criteria	Listed are the top five most dangerous areas of the most recent week in the data range.
Priority	2
Use Case/ Requirement	UC10, FR17
Responsibility	UC10, FR17
Feature Package	3-4
ID	AT15
Description	User selects "domestic" as a filter, clicks Analyse Data and navigates to "Trend Overview".
Acceptance Criteria	A line graph is generated which shows the change in amount of domestic incidents over the date range of the data set. The data is represented as a line graph, with the corresponding dates being the x-axis, and the number of crimes committed on the y-axis
Priority	7
Use Case/ Requirement	UC6, FR19
Responsibility	SH3, SH4, SH5
Feature Package	4
ID	AT16
Description	User navigates to the Map tab of the GUI.
Acceptance	All crimes contained within the currently selected list (main list by default)

Criteria	are represented on the map. Those crime records with incomplete or missing location fields are not represented on the map.
Priority	7
Use Case/ Requirement	UC7, FR19
Responsibility	SH3, SH4
Feature Package	4
ID	AT17
Description	User clicks on a different tab to the currently active in the GUI.
Acceptance Criteria	It takes less than a second to switch tabs under heavy load (Populating list in data viewer after importing data, or other tabs run during lab sessions when there is more strain on the lab computers. This tab change does not include the time to load the data into the tab
Priority	8
Use Case/ Requirement	TQR2
Responsibility	SH3, SH4, SH5
Feature Package	All
ID	AT18
Description	User creates a data record as per UC12, but leaves the location field blank or incomplete (x val missing, longitude missing etc), and then adds it to an empty list. User clicks on Maps tab to show map as per AT16.
Acceptance Criteria	Newly created crime record does not show on the map. A warning is given when creating the record that the mandatory data is incomplete
Priority	4
Use Case/ Requirement	TQR7
Responsibility	SH3, SH4, SH5
Feature Package	4

Table 7: Displays the acceptance tests, using a description, its priority, as well as the use case and feature package it associates with

5. GUI Prototypes

When the SANTA application is launched, the raw data view will be shown as below in **Figure 11**. Navigation through the database will be achieved by using the navigation bar at the top of each page shown in **Figure 10**. After clicking on the navigation bar button “File”, there will be a drop down menu. Clicking on “Import Data” will allow the user to import data from a file. Clicking on “Input Form” will open the crime record input form which allows the user to enter records manually, as shown in **Figure 14**. Clicking on “Export” will let the user write data to a file. Clicking “Close” on the File dropdown menu will shut the program down. The “Data” button will open the raw data viewer

depicted in **Figure 11** and display the currently selected data. The “Stats” button will display the currently selected data in the default graph view, shown in **Figure 13**. The “Map” button will display the currently selected data on a google map view, shown in **Figure 14**. The drop down box to the right of “Currently Selected Data” is used so that the user can select different data sets.

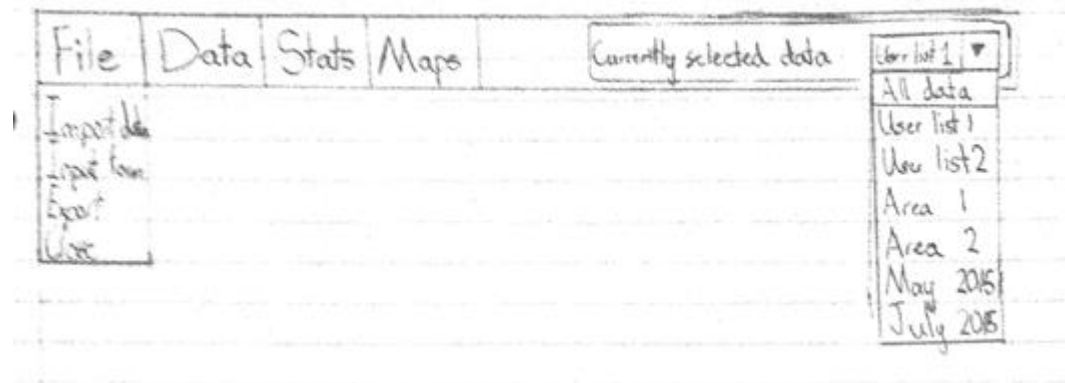


Figure 10: Navigation bar

The raw data viewer displays a summary of the currently selected data as shown in **Figure 11**. Underneath the navigation bar there is a number of text field and drop down boxes that can be used to filter and search the through the data. Clicking on a summarized record will open a window in front of the data viewer which will display more specific information relating to that crime as shown in **Figure 12**.

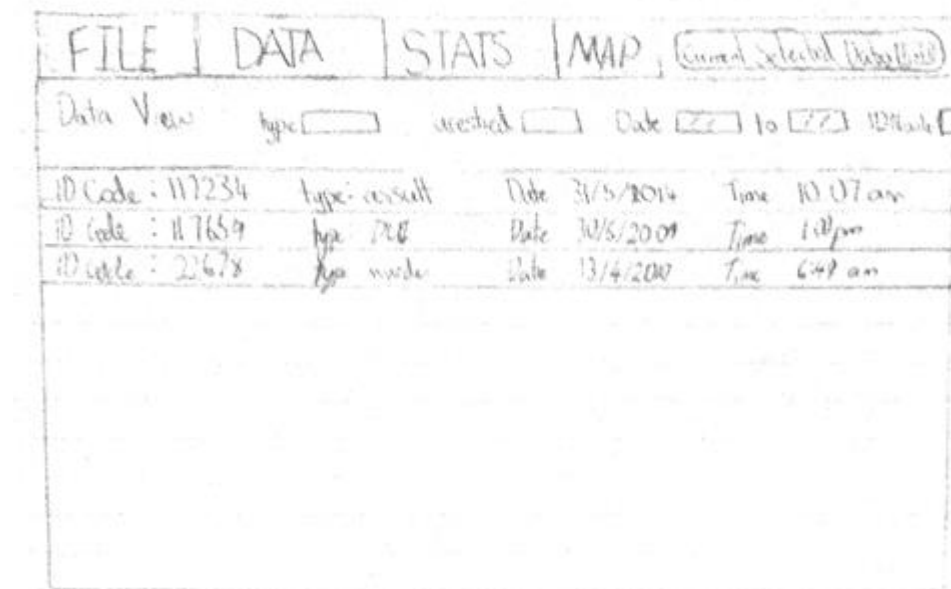


Figure 11: Raw data view of summarized records

The crime information window in **Figure 12** has a small menu bar at the bottom that allows the user to switch to the next/previous record, edit the current record, or show the crime in map view.

FILE D	117234 Incident Report		⊗	ated Data [Edit]
Data View	ID Code: 117234	Date/Time: 31/5/2015:20:35	<input checked="" type="checkbox"/> ID Number <input type="checkbox"/>	
ID Code: 117234	Type: assault	arrested: Yes	me 20:35	
Location Information				
Ward: christchurch		Beat: Caemure		
place: Park		Block: J12		
Longitude: 16298		Latatude: 117234		
Time from previous crime: 20min		Distance from previous crime 120m		
<div> ◀ Edit Record To Map ▶ </div>				

Figure 12: Data view with pop up window showing particular information of crime

The Stats view can be used to generate graphs and reports of the currently selected data as shown in **Figure 12**. The bar underneath the navigation bar has a number of text field and drop down box that can be used to select the criteria of the statics. The user will then choose between a report or graph view by ticking one of the box next to graph and report. The report or graph will then be generated when the user clicks the GO button.

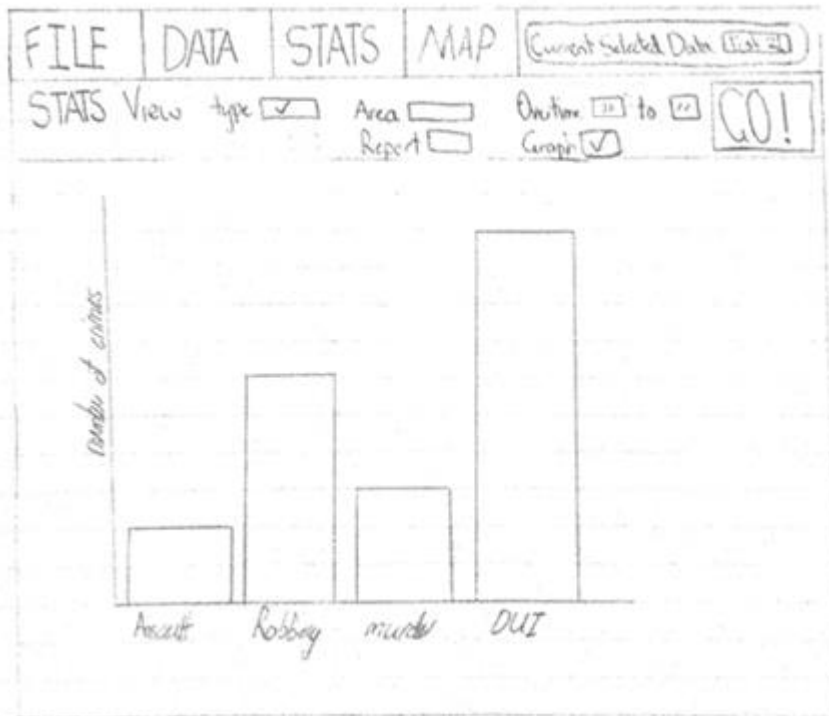


Figure 13: Statistic viewer showing graph generator

Map view. When the map viewer is first opened, all of the currently selected data will be displayed on the map. Using the filtering bar on the left of the screen the user will be allow to show only specific crimes which meet set criteria, which will be selected and shown when the Go button is pushed (clicking on a crime tag on the map will show the particular information relating to that crime).

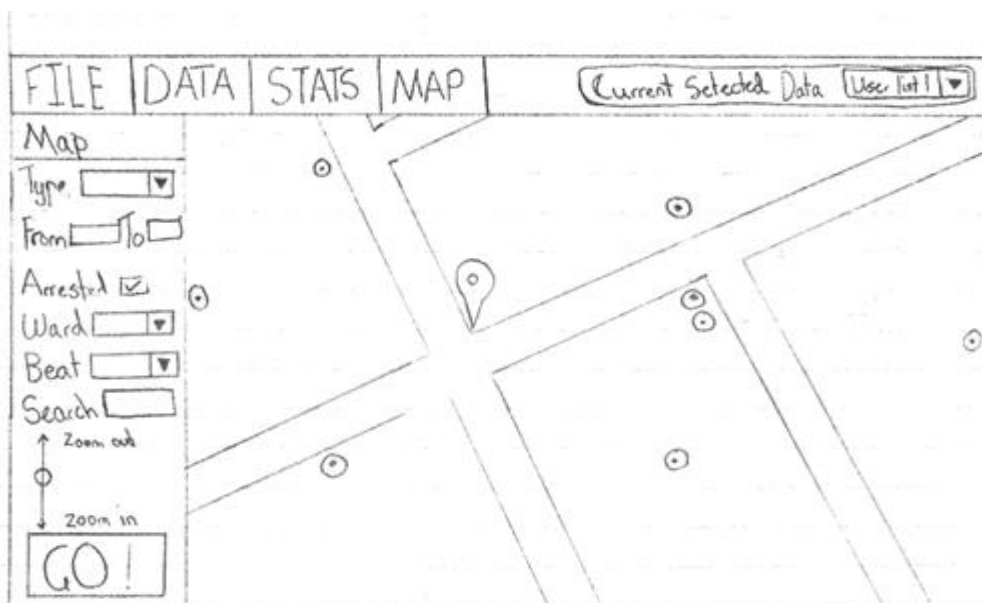


Figure 14: Map view of crimes

User entry form. The user entry form will consist of a number of text fields which users will fill out with the information of a single crime. Set text fields will display errors if the submission button is hit without them being correctly filled out. To edit existing crime records, the same form will be used but the already existing data will fill the corresponding text fields.

Figure 15: User data entry form and editing form

6. Risk Assessment

A table representing the potential implications the project and the developing team may encounter. Each proposed implication describes the estimated likelihood of an issue arising, the consequence and impact if one were to happen, and preventative measures to reduce the chance of it happening.

ID	RA1
Description	Not being able to finish a deliverable before the due date
Impact	Medium
Likelihood	Medium
Responsibility	Everyone in the group
Consequences	Grade penalties
Prevention	Completing work before deadlines and working as a team. Using a time

	buffer
ID	RA2
Description	All of the project files are deleted and cannot be recovered
Impact	High
Likelihood	Low
Responsibility	Everyone in the group
Consequences	Have to revert to a backup, or worse, restart the project
Prevention	Take backups regularly
ID	RA3
Description	A team member gets sick and is unable to work
Impact	Medium
Likelihood	High
Responsibility	Individual members
Consequences	The other members would have to pick up the workload temporarily
Prevention	Vaccinations, healthy diet and exercise
ID	RA4
Description	A team member drops out of the course
Impact	High
Likelihood	Low
Responsibility	Individual members
Consequences	The other members would have to pick up the workload permanently
Prevention	Make sure that everyone is happy about the project and help them if they are having difficulty
ID	RA5
Description	Project doesn't work on the Linux mint OS installed on the lab computers during a deliverable
Impact	High
Likelihood	Low
Responsibility	Vincent
Consequences	Could result in a poor grade if we miss a deadline. We would have to find what is causing the problem, and fix it. Test the project on multiple OS's.
Prevention	Use our 2 day buffer to resolve any issues that arise from running on multiple OS's. Have a test run on lab computers before deliverables.
ID	RA6
Description	The stakeholders are not happy with the final product.
Impact	Low
Likelihood	Low
Responsibility	Everyone in the group
Consequences	Brings coding group into disrepute. Poor grades. May lose time having to alter what our stakeholders were dissatisfied with

Prevention	Acceptance testing. Using iterative approach with many prototypes to catch problems early. Communicating with stakeholders early.
ID	RA7
Description	Everyone cannot understand the code of a group member
Impact	Low
Likelihood	Medium
Responsibility	Each individual member
Consequences	Will slow down development, and/or result in more errors
Prevention	Use Javadoc comments and also inline comments. Also use good variable names
ID	RA8
Description	The group has difficulty using Java FX in the GUI
Impact	Medium
Likelihood	Medium
Responsibility	All team members
Consequences	Would take time to gain sufficient skill to achieve what is needed for SANTA with FX
Prevention	Watch YouTube videos and read internet tutorials. Ask lecturers for help
ID	RA9
Description	Team conflicts
Impact	Medium
Likelihood	Medium
Responsibility	The members involved
Consequences	Unhappy team members, poor/uncompleted work
Prevention	Be considerate of members, and make an effort on all work assigned
ID	RA10
Description	Our Google Drive account is terminated. (e.g. Copyright material is detected)
Impact	High
Likelihood	Low
Responsibility	The individual organisations
Consequences	Could lose files and be forced to move onto a competitors service.
Prevention	Keep local copies of work
ID	RA11
Description	After a Push and merge, the program fails its unit tests and it cannot be debugged.
Impact	Medium
Likelihood	Medium
Responsibility	The person who coded changes

Consequences	All the time and effort put into the changes was wasted. Additional time will be required to get the program working again
Prevention	Use the backups to revert to a recent working version. Use comments to find where the issue arose.
ID	RA12
Description	A group member works on SANTA while on a trip and has no internet connection
Impact	Low
Likelihood	Medium
Responsibility	The member on the trip
Consequences	The new work won't be available for the rest of the team until an internet connection is established. Meaning that if lots of changes are made by the member on the trip, and the rest of the group also make changes, it could be difficult to merge successfully.
Prevention	Try not to make many complex changes to the project (which you think could result in merging problems) while away from the internet.
ID	RA13
Description	Team member doesn't contribute to the level expected
Impact	Medium
Likelihood	Low
Responsibility	Each individual
Consequences	We could fall behind on our pre assigned deadlines. Other members would have to pick up the workload. Could lead to tension.
Prevention	Talk to the group member in question before it becomes a problem. Seek lecturer's advice for disciplinary action if necessary.

Table 8: Displays possible implications that may potentially happen during the development of SANTA. It outlines who is responsible for these implications, what may happen if an issue does arise, and preventative measures.

7. Project Plan

The project plan assists the developers to complete the relevant sections for the given deadlines. It also helps keep the project on schedule and is a useful control tool for planning. The project plan as displayed in **Table 9** shows the relevant milestones of when set objectives need to be completed and important dates for the project SANTA and its developers.

Week	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
1	13-Jul	14-Jul	15-Jul	16-Jul	17-Jul	18-Jul	19-Jul
		Lecture 1		Lab 1	Lab 2		

2	20-Jul	21-Jul	22-Jul	23-Jul	24-Jul	25-Jul	26-Jul
		Lecture 2		Lab 3	Lab 4		
			Team Meeting				
3	27-Jul	28-Jul	29-Jul	30-Jul	31-Jul	1-Aug	2-Aug
		Lecture 3		Lab 5	Lab 6		
		Complete report draft, submit for feedback	Team Meeting				Have report completed , ready for editing
4	3-Aug	4-Aug	5-Aug	6-Aug	7-Aug	8-Aug	9-Aug
		Lecture 4		Lab 7	Lab 8		
		Deliverable One Due	Team Meeting Practice for Presentation of Deliverable One, work on peer review		Present Deliverable One		Set up IDEs for development, have peer review completed
5	10-Aug	11-Aug	12-Aug	13-Aug	14-Aug	15-Aug	16-Aug
		Lecture 5		Lab 9	Lab 10		
	Work on updating report sections	Complete updated report	Team Meeting Begin coding basic functionality	MS: Base code including importing implemented	Code GUI and database	Code GUI and filtering	MS: GUI completed . Stop Coding: Begin packaging deliverable two
6	17-Aug	18-Aug	19-Aug	20-Aug	21-Aug	22-Aug	23-Aug
		Lecture 6		Lab 11	Lab 12		
	Discuss presentation	Deliverable Two Due	Team Meeting Practice for presentation	COSC265 Assignment Due	Present Deliverable Two	Code updated filtering	MS: Data view has advanced filtering
Break	24-Aug	25-Aug	26-Aug	27-Aug	28-Aug	29-Aug	30-Aug
						MS: Data view has details available	

						on click	
	Emily away	Emily away	Emily, Dominic away	Emily, Dominic away	Emily, Dominic away	Emily, Dominic away	Emily away
Break	31-Aug	1-Sep	2-Sep	3-Sep	4-Sep	5-Sep	6-Sep
	Emily away	Emily away	Emily away	Emily away	Emily away	Emily away	Emily away
7	7-Sep	8-Sep	9-Sep	10-Sep	11-Sep	12-Sep	13-Sep
		Lecture 7		Lab 13	Lab 14		
	ENCE260 Lab Test	COSC264 Assignment Due	Team Meeting		COSC264 Midterm		
8	14-Sep	15-Sep	16-Sep	17-Sep	18-Sep	19-Sep	20-Sep
		Lecture 8		Lab 15	Lab 16		
	MS: Custom data addition implemented		Team Meeting MS: Data editing implemented			MS: 'Dangerous Areas' viewer implemented	Stop coding: Begin packaging deliverable three
9	21-Sep	22-Sep	23-Sep	24-Sep	25-Sep	26-Sep	27-Sep
		Lecture 9		Lab 17	Lab 18		
		Deliverable Three Due	Team Meeting Practice for presentation Complete peer review	Present Deliverable Three		MS: Data is persistently stored in database	Have peer review completed
10	28-Sep	29-Sep	30-Sep	1-Oct	2-Oct	3-Oct	4-Oct
		Lecture 10		Lab 19	Lab 20		
		MS: Graphing implemented	Team Meeting	COSC264 Lab Test	MS: Mapping implemented	MS: Filtering crimes on map	Stop coding: Begin packaging deliverable four
11	5-Oct	6-Oct	7-Oct	8-Oct	9-Oct	10-Oct	11-Oct
		Lecture 11		Lab 21	Lab 22		
		Deliverable Four Due	Team Meeting Practice for presentation	Demos	Present Deliverable Three at 3pm ENCE260 Assignment		

					t Due		
12	12-Oct	13-Oct	14-Oct	15-Oct	16-Oct	17-Oct	18-Oct
		Lecture 12		Lab 23	Lab 24		

Table 9: Displays the project plan with important milestones

8. Data Model

The type of files SANTA will be able to upload will be csv files. Within this file, there will be various keys of information identifying the crime. These will be parsed and then passed into the local crime database, which stores the crime information into a table. These keys include:

- Case/ID number
- Date of the crime and the time stamp
- Primary and secondary description
- Where it occurred, including the block, beat, ward, x and y coordinates longitude and latitude
- FBI CD
- IUCR
- Whether an arrest was made or not
- Whether it is domestic or not

Features that may be used within the application will access information from the crime database via imports and queries to analyse as required. A current data model of the predicted final system is displayed in **Figure 11** (this may be subject to change for future prototypes). The below diagram shows how the data will potentially be stored in the database generated by the application. Each user list will have a unique id within the single crime table in the database, that is populated by the crime records. Each combination of user list ID and case ID must be unique, but not the individual values for list ID and case ID.

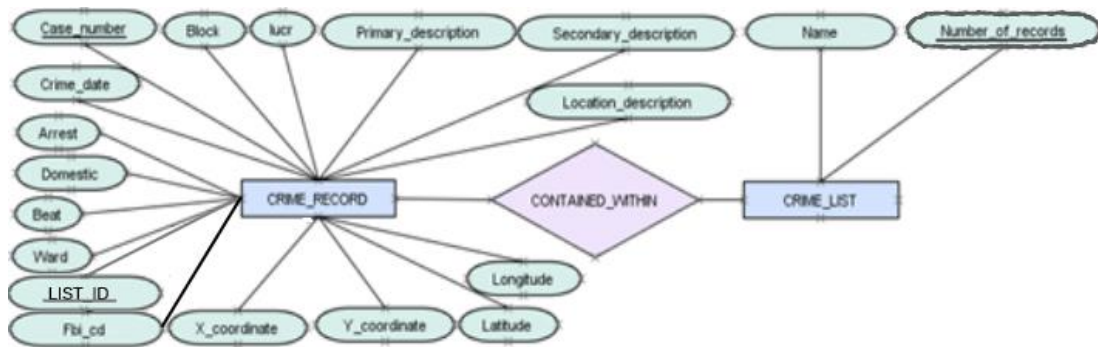


Figure 15: Current data model of the proposed system

9. Deployment Model

The deployment model displayed in **Figure 12** shows the way that the program is deployed on the physical architecture available. The OS node shows the currently supported (tested) operating systems. The application requires java runtime environment version 1.8.0.25 to execute the application. Included libraries ensure that the JRE is the only requirement for the end user to run the application locally. SANTA only accepts csv files as input.

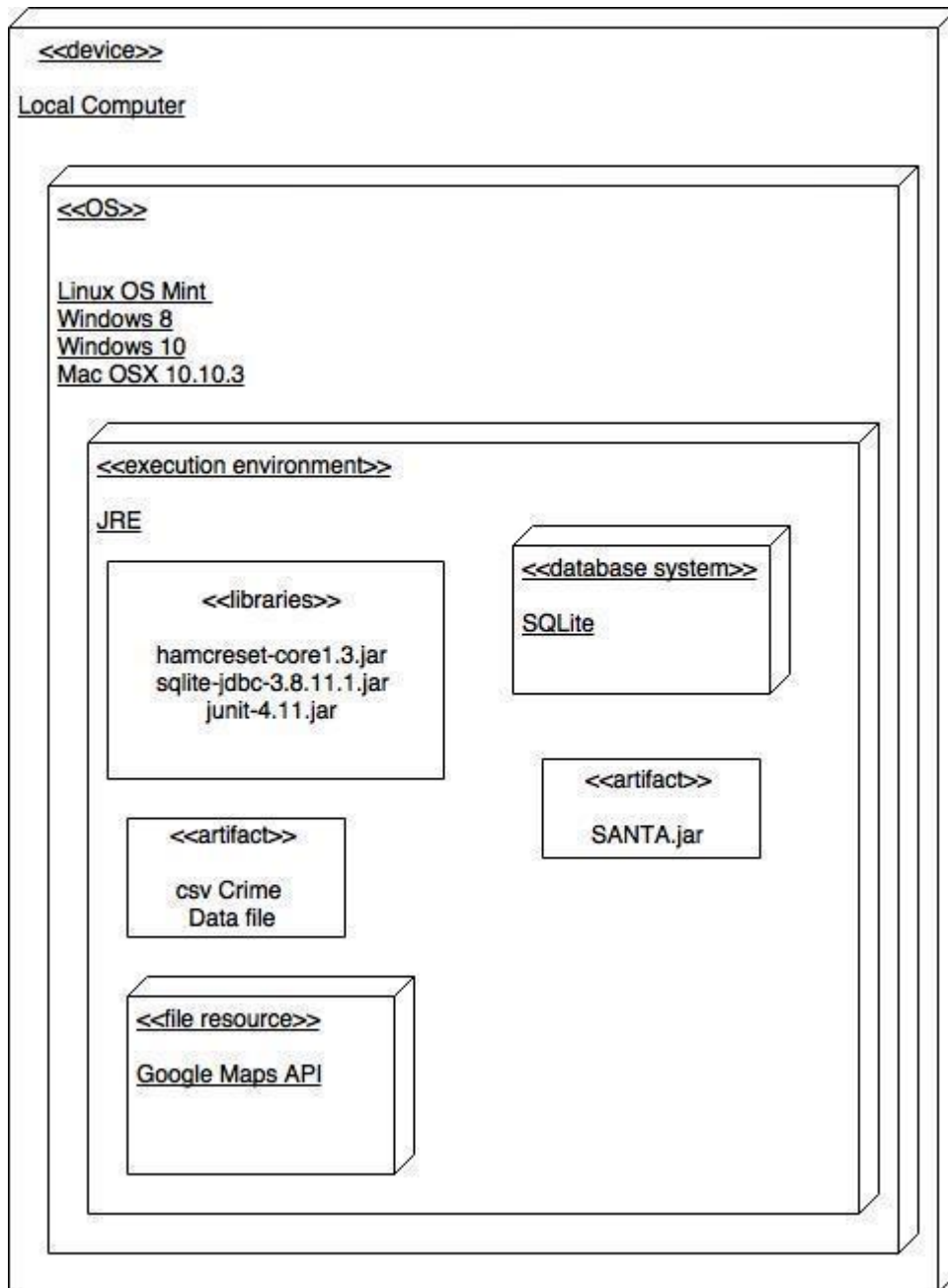


Figure 16: The deployment model for projected development

10. UML Class Diagram

The UML class diagram in **Figure 13** shows the relationships between the classes of SANTA, as well as listing their operations and attributes.

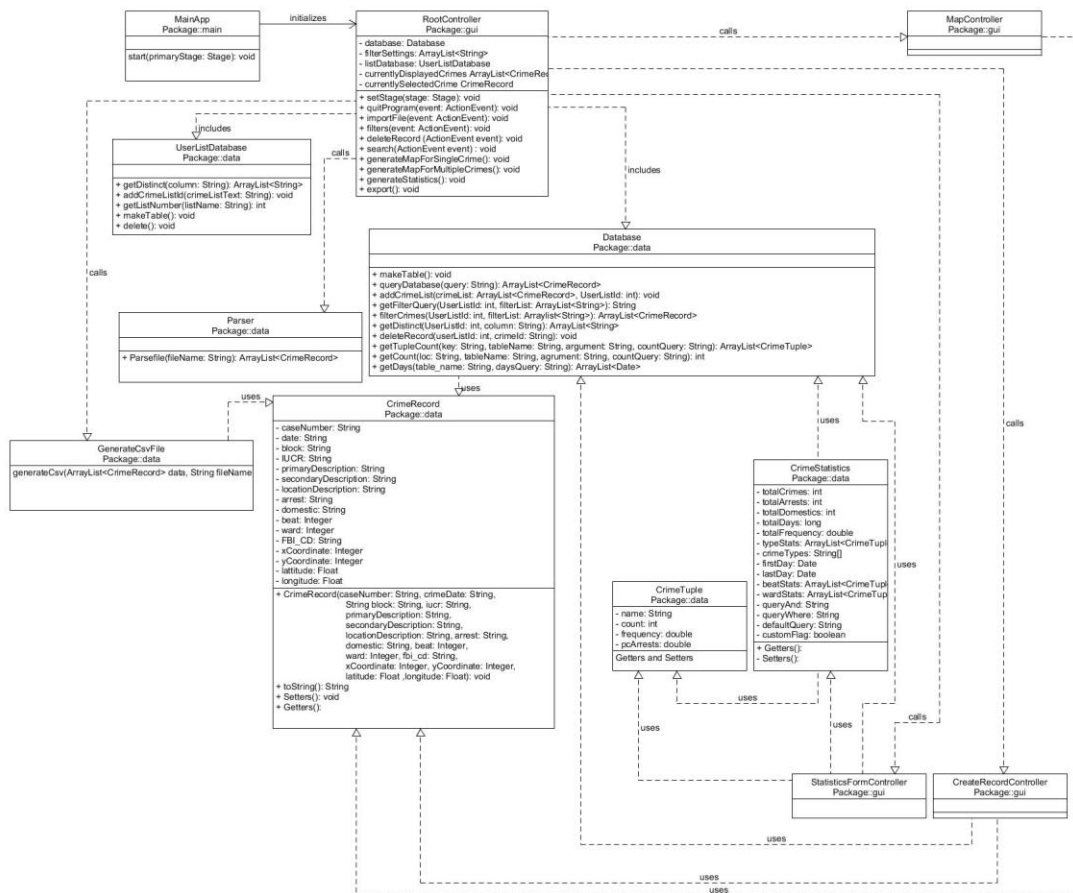


Figure 17: This is the class diagram for the current deliverable. For the future deliverables this diagram will be expanded.

11. Prototype Description – Final Version

For the final release of SANTA, the main focus was to ensure our application was fully functional with all bugs and inefficiencies dealt to. This section will highlight each main component of the SANTA application, with the corresponding features entailed.

Main GUI Window

This is the main window of the application. It has a top panel, with drop down bars and buttons, and main tabs; “Record Viewer” and “Map View”. When the application is launched, the “Record Viewer” tab will be open by default.

In the top left panel, there are three drop down menus; “File”, “View” and “Help”. These are described in further detail below.

Featured down the left side in the “Record Viewer” tab are the filtering features. You can select a date range on a calendar, type in a key word and click on “Search”, or tick types of interest that will filter the data open in the centre viewer accordingly after you click the button “Filter”. The types that are displayed down the left side are determined by the crime records in the viewer (meaning it won’t show types that aren’t present in the data set).

In the centre on the “Record Viewer” tab is the raw data viewer, which shows the data list of interest in grid view. Each column represents the data as labelled accordingly. The user is able to scroll up, down, left and right for viewing, and able to adjust the dimensions of the viewer. When a user clicks on a crime record in the data viewer, more advanced details appear in the empty fields in the “Viewing Record” panel on the right hand side. “Viewing Record” has three buttons called “Edit Record”, “Show Record on Maps” and “Delete Record”. These three buttons are described more below.

On the right in the top panel are 3 buttons and a drop down menu; “Analyse Data”, “Add” and “Delete”. The drop down bar contains all of the custom data sets a user has made. The drop down bar will display the first custom set in alphabetical order by default which are described in more detail below. “Add” and “Delete” are located on either side of the drop down bar. “Add” will prompt the user the name the data set, then store the current data in the viewer as a custom list. “Delete” will warn the user if they want to delete the current custom set selected, and will delete if the user consents. “Analyse Data” is further described below.

The “Map View” tab will display Google Maps when selected, with the corresponding points plotted that were currently in the raw data viewer. There are more details about this tab below.

“File” Dropdown Menu

In this drop down menu, there are four features; “Import”, “Export”, “Create Record” and “Exit”.

File->Import launches the file browser so the user can browse for/select a .csv file. Once the file is selected, the application will parse the file and import into the SANTA database for use by the application.

File->Export launched a window prompting the user to name the file they wish to export. It will convert the current data set into a csv file, and save the file to

File->Create Record launches a window with editable fields to enable the user to create their own crime record. There are more details of this feature described below.

File->Exit closes the application, but windows that have been opened by the application (e.g. “Analyse Data” window) need to be closed separately.

“View” Dropdown Menu

In this drop down menu, there are two features; “Show Filter Panel” and “Show Viewing Record Panel”. When the program is opened, by default, these have a tick next to them, indicating that the filter panel and the viewing record panel are visible in the main window. When these are clicked, the tick will disappear, and the corresponding panel will be minimized. The raw data viewer will adjust its size to accommodate the space left by the minimized panel. Clicking one of these boxes when it is unticked will make the tick reappear, and the corresponding panel will be maximized. The raw data viewer will adjust its dimensions accordingly.

“Help” Dropdown Menu

In this drop down menu, there is one feature; “User Manual”. Clicking on this will launch a new window with a manual describing what SANTA is, what it is intended to be used for, the features it has, and how to use them.

“Map View” Tab and “Show Record on Map” button

When this tab is selected in the raw data viewer, it will launch a new window that plots the locations of the data displayed in the raw data viewer on Google Maps. There is an allocated limit to the number of points that can be plotted due to performance issues. If there are too many records to be plotted, there is a pop up warning notifying the user that there are too many records to plot.

When the “Show Record on Map” button in the “Viewing Record” panel is pushed, it will launch a new window and show where that one crime was committed in Google Maps.

If the user clicks on a point in the maps, a box will appear giving details of the corresponding crime committed in that particular place.

“Analyse Data” button

When this button is clicked in the raw data viewer, it will launch a new window with four tabs; “Tutorial”, “Overview”, “Type Breakdown” and “Trend Overview”.

The window opens in the “Tutorial” tab by default. This tab gives the user an overview of what this feature does, and descriptions on what each tab contains.

The “Overview” tab will list the main statistics of the selected data set, such as the number of crimes in the data set, the total number of arrests made from that data set etc

The “Type Breakdown” tab down the left hand side shows two boxes. One box will have each type of crime detailed by the total incidents, average offense per day (over the date range of the data set) and the percentage of those offenders arrested. The other box displays the top 3 most frequent crime types, along with the information given to the above box. On the right hand side, will be a pie graph which shows the composition of each crime type. There is also a bar graph with the crime type on the y-axis, and the number of crimes committed on the x-axis.

The “Trend Overview” tab displays a trend graph showing the number of crimes committed over a period of time. The x-axis is the date range of the data set and the y-axis is the number of crimes committed.

Create Record

When this button is clicked in the raw data viewer, it will launch a new window with empty fields that await an input from the user. Each field represents a piece of information that makes up the data record (e.g. date of crime, crime type, primary description etc). The Crime ID and IUCR number is generated by the application, and is not editable. After all of the mandatory fields have been completed, the user can submit the crime record, and it will be stored in the SANTA database. Some of the fields do not have to be filled in for the crime record to be saved.

“Edit Record” button

When this button is clicked on the “Viewing Record” panel, it will make the fields of the crime record editable. The user can change the fields, and click “Save Changes”, which will commit the changes to the crime record. If the user tries to navigate away from the “Edit Record” mode, the application will prompt the user that there are unsaved changes.

“Delete Record” button

When this button is click on the “Viewing Record” panel, it will delete the selected record from the database.

12. Testing Procedures

This section discusses the overall test results of the SANTA application, and covers the details of unit and acceptance tests performed.

SUT (System Under Test)	CrimeRecord Class
Test Type	Unit
Test	Calling the getters and setters for field of the crime record
Test Description	Tests to see if the if the getters and setters return the value as expected and required
Result	The test returns the correct values for each method call of the Crimerecord class with developer-generated tests
Pass/Fail	Pass
Tester	Vincent Rielly
Date	19th September 2015
UC/ requirement	FR3, FR9, CQR1, CQR3

Table 11: Shows the unit test details and results for the CrimeRecord class of SANTA

SUT (System Under Test)	CrimeStatistics Class
Test Type	Unit
Test	Returning statistics about entire database
Test Description	Tests to see if the CrimeStatistics class returns accurate information about the entire database with measurable return values: Total Crimes, total arrests, total domestic incidents, first day in range, last day in range.
Result	The test returns the correct values for each method call of the created CrimeStatistics object.
Pass/Fail	Pass
Tester	Dominic Jarvis
Date	18th September 2015
UC/ requirement	FR16, FR18, CQR1, CQR2, CQR3, TQR2

Table 12: Shows the unit test details and results for the CrimeStatistics class of SANTA

SUT (System Under Test)	UserListDatabase Class
Test Type	Unit
Test	Adding lists to the User List database
Test Description	Adding different lists of data sets and ensuring the database can

	distinguish each list from one another
Result	The database has 2 default lists, but they are not customizable - i.e no crime records can be added to them as of yet. There is no implementation to make a custom list, but the database can accommodate lists.
Pass/Fail	Pass
Tester	Sam Cassidy
Date	18th September 2015
UC/ requirement	FR10, CQR1, CQR3

Table 13: Shows the unit test details and results for the UserListDatabase class of SANTA

SUT (System Under Test)	Database Class
Test Type	Unit
Test	Adding/deleting/updating crime records
Test Description	Tests the data base by adding and deleting crime records from it. It also tests to see if editing/updating changes the crime record in the database
Result	The data base adds, removes and updates crime records successfully
Pass/Fail	Pass
Tester	Sam Cassidy
Date	20th August 2015
UC/ requirement	UC11, UC12, UC13, UC16, FR13, FR14, FR15, CQR1, CQR3

Table 14: Shows the unit test details and results for the Database class of SANTA

SUT (System Under Test)	Database Class
Test Type	Unit
Test	Filtering Methods
Test Description	Performing queries on the database so it returns crime records of interest. This included searching for type, crimes within a date range, and selected key words
Result	The database return the crime records filtered from the data set
Pass/Fail	Passed
Tester	Sam Cassidy
Date	20th August
UC/ requirement	UC3, UC4, FR3, FR4, FR8, FR14, CQR1, CQR3

Table 15: Shows the unit test details and results for the Database class of SANTA

SUT (System Under Test)	Parser Class
Test Type	Unit
Test	Parsing data files
Test Description	Parsing csv files of different sizes (0.98MB, 1.97MB, 51.8MB). This was tested on Linux Mint, Windows 8, Windows 10, MAC OSX 10.10.3
Result	It parses each file within reasonable time. As the size of the csv file

	<p>increases, so does the time it takes to parse (51.8MB takes roughly 4.5sec).</p> <p>There is a memory limit of 100MB as to what the application will allow to be imported at any one time. If a data file over 100MB is selected for import, the application will not import it.</p> <p>This is for performance</p>
Pass/Fail	Pass
Tester	Vincent Reilly (using JUnit), Sarah Jackson
Date	16/09/15
UC/ requirement	UC11, FR1, CQR1, CQR3

Table 16: Shows the unit test details and results for the Parser class of SANTA

SUT (System Under Test)	SANTA main interface
Test Type	Acceptance
Test	Ensuring it performs its function
Test Description	<ol style="list-style-type: none"> 1. That every button operates as expected (e.g clicking “Analyse Data” takes you to the “Analyse Data” window). The launch time should appear instant. The features to be tested are in the main panel on the top right are 3 buttons; “Add”, “Delete” and “Analyse Data”. There is also a drop down bar that should display saved lists of crime records. “Edit Record” on the Viewing Record pane should make the crime record fields editable (apart from the ID number and IUCR). If another record is clicked while in “Edit Record” mode, it should default the Viewing Window pane and not remain in “Edit Record” mode, and prompt user of unsaved changes. “Show Record in Maps” on the Viewing Record pane should launch map window and indicate where that crime was committed on Google Maps. There should be a “Delete Record” button on the Viewing Record pane that enables the user to delete a crime record permanently from the app 2. The drop down file->import should launch the file browser to enable importing a crime record. File->export should prompt user to name crime list, and save as csv in .jar directory. file->quit should close the program. When the program is reopened, the data should still be present. Exported csv file should be present in directory 3. In the main viewer, you should be able to scroll through the crime records shown and observe the corresponding information in the indicated columns. 4. When you select a filtering method, it should show in the main viewer the crimes filtered accordingly (type, date range, domestic, arrest). 5. When you click on a crime record in the viewer, it should fill in the fields of the Viewing Record pane on the right hand side with more details on that particular crime record. It should show the time and

	<p>distance in between that crime, and the last committed crime</p> <ol style="list-style-type: none"> There should be a button called “Add” which should save selected crime records, and prompt the user to name this list. After is has been saved, it should appear in the drop down bar on the top right. “Delete” should prompt the user to delete currently selected custom list It should look aesthetic and uncluttered
Result	<ol style="list-style-type: none"> The “Analyse Data” buttons launches its corresponding window. The window pops up in near instant time. The drop down bar is present, and displays the default custom list, as well as customs lists added using the “Add” button. “Delete” will delete the selected custom data set. Edit Crime Record works as expected, and enables editing of the fields. It does not enable editing of the ID tag or IUCR. If you click elsewhere, it will prompt user that there are unsaved changes. “Show in Maps” works in near-instant time. “Delete Record” will delete currently selected record from the database The file drop down bar works, and performs functions accordingly (imports and exports csv files and quits program when prompted). The “Create Record” option launches the create record window. When the program has been closed and re-opened, data in the database is still present Scrolling through the records was possible, as is changing the dimensions of the columns It filtered the records appropriately and accordingly It fills out the appropriate fields with the corresponding information of the selected crime record. Will show the distance and time from the last crime committed You can add your own customs lists using the “Add” button. You are prompted to name your list. When saved, it appears in the drop down bar. The “Delete” button will delete the selected custom list. First the user is prompted to ensure they did not accidentally click the button All of the buttons and viewer are appropriately coordinated and proportioned
Pass/Fail	Pass
Tester	Sarah Jackson
Date	17th September 2015
UC/ requirement	UC1, UC2, UC3, UC4, UC11, UC12, UC13, FR1, FR2, FR3, FR4, FR5, FR6, FR7, FR8, FR9, FR12, FR13, FR16, CQR1, CQR2, CQR3, TQR3

Table 17: Shows the acceptance test details and results for the GUI main interface of SANTA

SUT (System Under Test)	SANTA Analyse Data window
Test Type	Acceptance
Test	Ensuring it performs its function
Test Description	<ol style="list-style-type: none"> When the button “Analyse Data” is clicked, it launches a new window with 4 tabs; “Tutorial”, “Overview”, “Type Breakdown” and “Trend Overview”.

	<ol style="list-style-type: none"> 2. "Tutorial" tab should be closable, and give the user information as to what each tab will contain 3. "Overview" tab should show the data overview of the selected set, such as total crimes, arrests etc 4. "Type Breakdown" should have dialog boxes showing the basic statistics of each type of crime, and the information of the top 3 crimes. It should display a pie graph showing the crime composition of each type of crime, and a bar graph showing the number of crimes committed for each type 5. "Trend Overview" should show a trend graph of how many crimes are committed over the date range of the data set 6. Performs in reasonable time in relation to the size of the data set (does not exceed more than 20seconds for the largest data file)
Result	<ol style="list-style-type: none"> 1. It launches near instantly. Calculates the statistics of all of the crime records in the database and selected lists of crime records in with the corresponding tabs visual 2. Tab is closable and informative of each tab 3. Shows the overview of each data set in a well-organized and easy-to-read manner 4. Shows a visually attractive pie graph with all of the corresponding crime types and their composition. Appropriately labelled and easy to read and analyse. Likewise with the bar graph. Shows the crime type breakdown well with the main statistics of each type 5. Shows a visually attractive trend graph that displays the corresponding data well 6. Performs in reasonable time
Pass/Fail	Pass
Tester	Sarah Jackson
Date	17th September 2015
UC/ requirement	UC5, UC6, UC8, FR18, CQR1. CQR2, CQR3, TQR2

Table 18: Shows the acceptance test details and results for the GUI "Analyse Data" window of SANTA

SUT (System Under Test)	SANTA Map view tab
Test Type	Acceptance
Test	Ensuring it performs its function
Test Description	<ol style="list-style-type: none"> 1. When the tab "Generate Map" is clicked, it generates the location of crime record(s) on Google Maps. The tab should launch in near instant time, and the map should load within reasonable time in relation the number of crimes being plotted (5 seconds of less) 2. The map should occupy the majority of the window 3. Should be able to interact with the map by navigating with the mouse (dragging) and able to zoom in and out 4. When a location of a crime is clicked in the map, it should pop up with a box with a brief overview of that crime 5. Should ignore crime records that do not have a given location

Result	<ol style="list-style-type: none"> 1. It launched the window quickly and loaded the map to indicate the locations of each crime record in the viewer within 5 seconds. If there are more than 500 records, it will not plot (a limit was put on it). This is due to performance issues 2. Occupied the majority of the space in the window 3. Navigation and zoom fully functional 4. Pops up with a box when you click on a crime data point with relevant data 5. Unable to be tested currently as "Create Record" requires all fields to have an input (custom records cannot be made)
Pass/Fail	Pass
Tester	Sarah Jackson
Date	17th September 2015
UC/ requirement	UC6, UC7, FR19, CQR1, CQR2, CQR3, TQR2, TQR3,

Table 19: Shows the acceptance test details and results for the GUI "Generate Map" window of SANTA

SUT (System Under Test)	SANTA Create Record window
Test Type	Acceptance
Test	Ensuring it performs its function
Test Description	<ol style="list-style-type: none"> 1. When the button "Create Record" is clicked, it should launch a window in near-instant time that enables a user to fill in fields to add a new crime record 2. Each label and field should be appropriately placed, informative, intuitive and aesthetic 3. When the "Save" button is selected, it will store the crime record in the database 4. An ID number and IUCR number for the new crime record should be generated, not inputted 5. Entering some fields should remain optional, and should still generate when submitted (e.g. location is left as null) 6. Will prompt user if mandatory fields have been left blank, and informs them what fields they left out
Result	<ol style="list-style-type: none"> 1. It launched the "Create Record" window in near-instant time 2. Each label has been strategically and logically placed. Each label is informative and intuitive 3. It stores the crime record when you select "Save Changes" 4. ID number generates and checks for duplicates to ensure it is unique. IUCR number generates upon primary and secondary description being selected 5. Will submit when optional fields are left blank 6. Prompts user and informs them of mandatory fields they left blank
Pass/Fail	Pass
Tester	Sarah Jackson
Date	20th September 2015

UC/ requirement	UC12, FR13, CQR1, CQR2, CQR3, TQR2
-----------------	------------------------------------

Table 20: Shows the acceptance test details and results for the GUI “Create Record” window of SANTA

SUT (System Under Test)	SANTA
Test Type	Acceptance
Test	Operation on multiple OS
Test Description	The SANTA application should launch and be fully functional on Windows 8, Windows 10, Linux Mint and OSX
Result	Successfully works on the listed OS’s mentioned
Pass/Fail	Pass
Tester	Vincent Reilly, Sarah Jackson, Emily Price, Dominic Jarvis
Date	20th September 2015
UC/ requirement	UC12, FR13, CQR1, TQR2, CQR2, CQR3

Table 21: Shows the acceptance test details and results for the SANTA application

SUT (System Under Test)	SANTA
Test Type	Acceptance
Test	Warning dialog boxes
Test Description	Shows appropriate warning prompts for different methods <ol style="list-style-type: none"> 1. If a record is not selected and you click “Show Record on Map” it will show a warning window saying no record is selected 2. If you try to save a crime record in “Create Record” with mandatory fields left empty, it will prompt the user saying fields are missing, and list the fields that are required 3. If you are in the “Edit Record” pane, and try to navigate away without saving, the user will receive an error prompt saying there are unsaved changes 4. Will prompt the user to rename the list if they try to submit a custom list where the name is already taken
Result	Showed appropriate warning boxes and prompts for all of the cases above
Pass/Fail	Pass
Tester	Sarah Jackson
Date	5th October 2015
UC/ requirement	UC3, UC4, UC6, UC10, UC12, UC17, FR17, CQR1, CQR2, CQR3, TQR4, TQR7

Table 22: Shows the acceptance test details and results for the SANTA application

SUT (System Under Test)	SANTA
Test Type	Acceptance
Test	Loading Screen
Test Description	1. When SANTA is launched, there is loading screen with the

	name of the application, the logo, and a loading animation with an appropriate label, showing that the program is loading 2. When a user imports a data file, a loading message with a loading animation appears in the center of the main panel where the user can see 3. The loading animations and messages appear in strategic locations, and do not make the application look clutter. 4. They are aesthetic, and the user is able to easily identify and understand what they mean
Result	1. Successfully shows all of the above criteria 2. Successfully shows a loading animation and message when data is imported 3. Do not make the application appear cluttered. 4. Are very intuitive and obvious when active
Pass/Fail	Pass
Tester	Sarah Jackson
Date	5th October 2015
UC/ requirement	UC3, UC4, UC6, UC10, UC12, UC17, FR17, CQR1, CQR2, CQR3, TQR4, TQR7

Table 23: Shows the acceptance test details and results for the SANTA application

Discussion:

The majority of the acceptance tests were covered; especially the ones that had the highest priority. The acceptance tests that were performed on SANTA covered a diverse range, testing for functionality, efficiency, performance, complexity and aesthetics on a variety of operating systems.

The parser was tested with 3 different csv files of varying sizes. It parsed all of the files very quickly (4.5s for the 54MB file). An issue with the parser is that if a file of a large size is passed to the parser, it will try to parse it and will not stop until it has completed. This is inefficient as large files could take a significant amount of time and processing power, and could lead to system failure. Although the likelihood of an enormous file is not significantly high, this problem is highly critical. A data cap was implemented so that a user cannot upload a file that is more the 100MB

The main interface failed the acceptance test because users are unable to create their own custom lists that will add to the drop down bar. It does however come with 2 default lists that are stored in the User List Database. The “Delete Button” also has not been added yet into the GUI due to time constraints. The code for both adding custom lists and deleting crime records is written and implemented, it just needs to be integrated. This will happen for the next deliverable. Everything else is fully functional and efficient. This failure has a medium criticality because it requires features that needs to be implemented and fully functional, but it does not affect the overall base functionality of the application.

Acceptance tests that were not covered are AT9. AT9 is where warnings appear at the top of the GUI, warning the user of the 5 most dangerous areas from the most recent week in the data range. This method was not implemented as in the “Analyse Data” window, there is a breakdown of the most frequent crimes with sufficient information about location. Therefore we deemed this feature unneeded.

The test coverage for acceptance tests was relatively extensive. Most acceptance tests were covered, and those that weren’t have been justified above.

The requirements that the acceptance tests did not make were:

- TQR4 (an error message is displayed rather than the program crashing)
- TQR8 (the user will receive a warning if a file is corrupted)

TQR4 can be justified with time constraint, and that other features took a higher precedence to become functional. There is a lot of error handling currently in the program with warning boxes, but in very rare cases, the crashes are inevitable and unpredicted, such as the Google Map. Sometimes when you zoom in and out too fast on the displayed map, the program will crash. This is due to the technology we are using to compliment the application, and some of this error handling is beyond our control.

TQR8 was classed as redundant and will not be implemented (due to complexity it will impose and low priority rating).

Test coverage for the unit tests are as follows. Each percentage represents the line coverage for the corresponding class:

- CrimeRecord - 100%
- CrimeStatistics - 88%
- CrimeTuple - 72%
- Database - 78%
- Parser - 80%
- UserListDatabase - 78%
- IUCR Parser - 75%
- AsyncTask - 0%
- GenerateCsvFile - 0%

CrimeTuple does not have extensive unit tests. This class consists mainly of getters and setters. As these methods are rather straight forward, they were not tested extensively. Our goal was to have at least 70% code coverage for our “model” classes. Although CrimeRecord consisted mainly of getters and setters, because it is the main object the application orientates around, we deemed it necessary and essential that each method was fully tested.

The GUI classes have 0% unit tests, because all of the GUI classes are not “model” classes. The GUI was thoroughly tested with acceptance tests, and the most important of these mentioned in the above tables.

GenerateCsvFile has 0% because it just appends data into a list. AsyncTask has 0% because it is an abstract class, and therefore cannot be tested.

13. Future Advice for SENG Students

- Get organized! Get together with your team and start planning ahead. Make sure you discuss tasks, evenly distribute the load, and set deadlines for when they should be completed by. Always stay one step ahead, so if it doesn't go to plan, you have plenty of time to readjust.
- Communicate. This will be the first course that you really have to work as a team. Although this can lead to a number of problems, there are various advantages and you can achieve so much more if you work effectively as a team. Talk to one another and collaborate, or else you will find when you synchronize your work together, there will be inconsistencies. You can not achieve this project on your own. Depend on one another's strengths and weaknesses, and ask for help the moment you need it. You all need to work together and support one another if you want to achieve highly. If someone isn't pulling their weight, go and talk to the lecturer.
- Never leave anything last minute. In your project plan, insert dates for code freezes days before the final deadline, to give a buffer zone for bugs that may arise. Ensure you have git, maven and jenkins running right from the beginning. There are tutorial sessions to help you do this. Make sure you start fixing a failed build right away rather than leaving it last minute, because your program will get more and more complex as time goes on, and therefore much harder to fix.

14. References

Alexander, W. (2011, April, 4th). *Crime Spy UK* [Mobile application software]. Retrieved July 26th, 2015 from <http://data.gov.uk/apps/crime-spy-uk>

Brocka, B. (2012, September 18th). *What is the threshold where actions are perceived as "instant"?* Retrieved August 3rd 2015, from <http://cogsci.stackexchange.com/questions/1664/what-is-the-threshold-where-actions-are-perceived-as-instant>

City of Chicago. (N/A). *Crimes 2001-present*. Retrieved July 23rd 2015, from <https://data.cityofchicago.org/view/5cd6-ry5g>

NZ Stat. (updated 2015, July, 31st). *Crime and Justice*. Retrieved July 23rd 2015, from http://www.stats.govt.nz/browse_for_stats/people_and_communities/crime_and_justice.aspx

Wikipedia. (2015, January 27th). *Data Corruption*. Retrieved August 1st 2015, from https://en.wikipedia.org/wiki/Data_corruption

14. Appendix

This appendix outlines what each individual of SENG202 group 5 mainly contributed to this report. Everyone participated in editing the other sections, ensuring there were no inconsistencies throughout the report.

Section	Subsection (if applicable)	Team Member(s) Involved
1. System Context and Relevant Business Information	1.1 System Context	Sarah Jackson
	1.2 Business Vision and Rationale	Sarah Jackson
2. Requirements	2.1 Stakeholders and Concerns	Sam Cassidy
	2.2 User Stories and Cases	Emily Price
	2.3 Sequence Diagrams for Major Use cases	Dominic Jarvis (sequence for UC3) Sam Cassidy (sequence diagram for UC6) Sarah Jackson (sequence diagram for UC7) Vincent Reilly (sequence diagram UC11) Emily Price (sequence diagram UC12)
	2.4 Functional Requirements	

	2.5 Quality Requirements	
3. Domain Model		Emily Price
4. Acceptance Tests		Dominic Jarvis
5. GUI Prototypes		Sam Cassidy
6. Risk Assessment		Vincent Reilly
7. Project Plan		Sam Cassidy Emily Price
8. Data Model		Sarah Jackson Dominic Jarvis
9. Deployment Model		Sarah Jackson Dominic Jarvis
10. UML Class Diagram		Vincent Reilly
11. Prototype Description - Beta		Sarah Jackson
12. Testing		Sarah Jackson
13. Future Advice for SENG students		Sam Cassidy Sarah Jackson
13. References		Sarah Jackson
14. Appendix		Sarah Jackson
Title Page		Emily Price and Sarah Jackson
Contents Page		Sarah Jackson
Report Formatting		Sarah Jackson
Change Log		Sarah Jackson

Table 24: Shows what sections team member(s) mainly contributed to in this report

