SENG202 Project - Phase Three

Insight

Group 10

George Carr-Smith: 69583529
Daniel Pallesen: 65881229
Bede Skinner-Vennell: 33944229
Sofonias Tekele Tesfaye: 51152535
Jonathan Tomlinson: 48760961

Table of Contents

Change log

Phase 2

Section number & title	Description of change	Author
1 Business Context	Updated the business context to better reflect our current application. Changed to make it more concise.	George Carr-Smith
1.2 System Context	Replaced the system context diagram	George Carr-Smith
2. Stakeholders and requirements	Deleted the “functional requirements” section.	Jonathan Tomlinson
2.1 Stakeholders	Changed the priority of people moving house (S_05) to high. Merged S_06 (policymakers), S_08 (police forces), and S_09 (government statistics departments) into one stakeholder S_06 (governments). Added mention of users that were considered but not prioritised to the table description. Swapped S_05 (people moving house) and S_03 (development team) to better group stakeholders together. Removed S_11 (Businesses).	Jonathan Tomlinson
2.2 Requirements (Quality requirements)	Re-wrote all quality requirements. Merged some that were duplicates.	Jonathan Tomlinson
2.2 Requirements (UML use case diagram)	Remade UML use case diagram to better reflect our current application. Added use case 8,9, 10 and 11.	Daniel Pallesen
2.2 Textual Description of Use Cases	Added textual use cases for adding and deleting records	George Carr-Smith
2.2 Textual Description of Use Cases	Updated textual use cases to better reflect our applications uses	All
2.2 Requirements (Key Drivers)	Updated key drivers table to reflect new quality requirements. Added discussion of results.	Jonathan Tomlinson

2.2 Requirements (Textual use cases)	Updated textual use cases to better reflect our current program. Separated alternative and exceptional flow to make it clearer which was which. Added use case 8,9 and 10.	Daniel Pallesen
6. UML Class Diagram	Completely replaced class diagram and discussion.	Bede Skinner-Vennell
7. Testing Procedures	Added testing procedures	All
8. Current Product Version	Added Current Product Version	All
9. Risk assessment	Added R_16 about the coronavirus. Replaced R_3, R_4, and R_9, about time constraints, user requests, and user feedback respectively with poor scope, unproductive meetings, and bugs in dependencies respectively.	Jonathan Tomlinson
9. Risk assessment	Removed risks R_05, R_08, and R_12 about the software running slowly, features not being included, and errors in the analysis part of the software respectively. Replaced R_05 about the software running slowly with a risk of poor task allocation.	Jonathan Tomlinson

Phase 3

Section number & title	Description of change	Author
3. Acceptance Tests	Deleted some tests that weren't useful, and added some more based around our stakeholders needs. Rewrote the description	Bede Skinner-Vennell
2.2 Requirements	Re-did the key drivers table again, so that key drivers do not have a 1:1 mapping to quality requirements. Also added a paragraph and replaced another in the discussion.	Jonathan Tomlinson
2.2 Requirements	Updated the use case to match the current product version. Added use case UC_14 (analysis popup).	Jonathan Tomlinson
2.2 Requirements	Quality requirements: replaced QR_4 (GUI showing main features) with a measurable usability requirement for novice users. Replaced QR_13 (advanced keyboard shortcuts) with a similar requirement for	Jonathan Tomlinson

	experienced users. Removed QR_9 (respond to intuitive actions) and added discussion of this to the quality testing section. Removed QR_12 (ask for confirmation before important actions) and ensured it was mentioned in the use cases. Re-wrote QR_5 (reuse of code) in an attempt to be more specific.	
4. GUI Prototypes	Completely rewrote GUI prototypes discussion to demonstrate how the app has changed and the reasons behind the changes.	Daniel Pallesen
2.2 Requirements (UML Use Case Diagram)	Added UC_14 to use case diagram	Daniel Pallesen
7. Testing Procedures	Added manual tests for new functionality, rewrote a large part of the discussion to be more up to date and reflect newer testing procedures	Bede Skinner-Vennell
6. UML Class Diagram	Updated class diagram and discussion to reflect updated code.	Bede Skinner-Vennell
8. Current Product Version	Updated current product version to reflect new features - Added mapping to notable features - Removed heatmap and loading bar from features not in this release	Bede Skinner-Vennell
1. Business and System Context	Updated the business case to reflect which “additional features” have already been implemented, and which features are only possible after phase three ends (as they are no longer planned for the initial release).	Jonathan Tomlinson
7.1 Functional Testing	Added and ran a test for UC_14 (analysis pie chart popup)	Jonathan Tomlinson
1.2 System Context	Updated the system context diagram and system context description.	George Carr-Smith
11. Lessons Learned	Added Lessons Learned section	All

1. Business and System Context

1.1 Relevant Business Information

Many enterprises have been making investments in software and hardware infrastructure that supports and facilitates data collection, integration, analysis and interpretation. (Eldridge, 2006). As a result, the demand for software tools that aid in data visualisation and analytics has been growing over the past 25 years. Figure 1 shows that currently the size of the market for data visualization tools is estimated to be 6 billion USD with an estimated growth of 4.8 billion USD from 2022 to 2027. Dictated by the appealing market potential, many software development companies have been delivering large and complicated commercial software products to large enterprises.

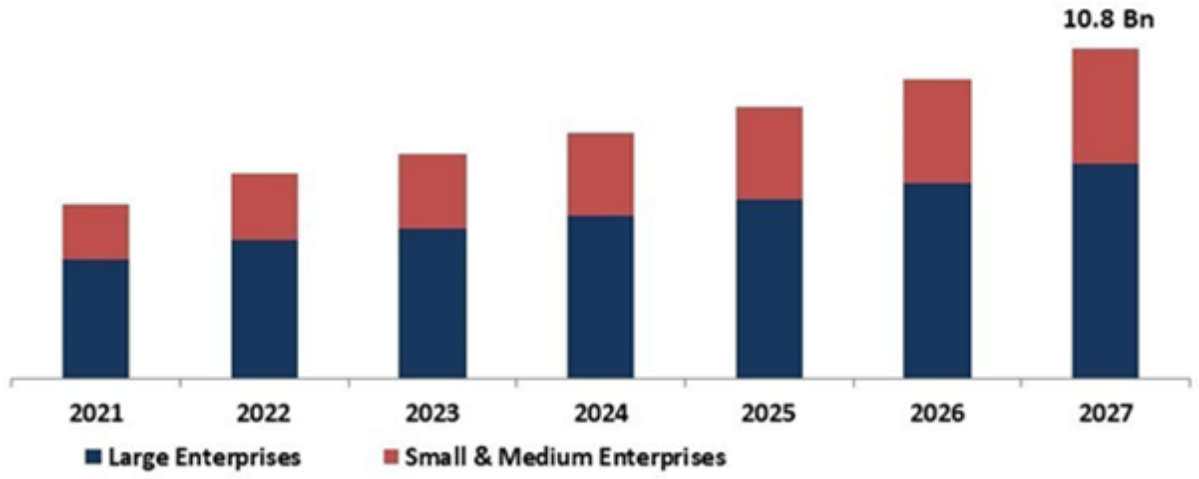


Figure 1 - Data visualization Tools Global Market size in USD, By Enterprise Size (Marqual IT Solutions Pvt. Ltd (KBV Research), 2021)

However, in line with its lesser market potential, the available market for data visualisation tools for small enterprises and individuals has not been the primary focus of the software industry. Hence many small organizations and individuals find it very hard to find data visualisation tools that are easy to use, customized to their needs, and not exorbitantly priced.

This is where our project **Insight** comes in. Insight's first edition will be a free, single-user desktop application. Insight offers small scale community-based organizations, advocacy groups, activists, students, those moving house, and other interested individuals basic crime data analysis and visualisation capabilities. Insight's main purpose is crime data visualisation with a focus on being simple and easy to use. The growing demand for data visualisation tools, along with this simplicity of use, is what is expected to drive demand for Insight.

Business Case Outline

The target markets for Insight are organisations and individuals who are unable to access expensive commercial crime data visualisation software and do not need the extra complexity that such software can bring.

The groups and individuals we have decided to target with Insight are:

- **Local activists** and **advocacy groups,** who will use Insight to visualise crime data in their local community. They will be able to produce maps, statistics, tables, and graphs which they can then use to help lobby their local policymakers. Insight will help activists push for changes in their communities and help them solve local problems.
- **Students** engaged in university or school research may use Insight to produce statistics or visuals for their projects.
- **People moving houses,** who may use Insight to research the prevalence and type of crime in areas they are considering moving to.

These groups will be able to import, filter and manipulate crime data in the application. This data will be stored so the user can access it even after restarting the program. The user can graph the data, view the data on a map, analyse the data points and export the data to a readable file. If Insight's development is to be continued past the initial release (phase three of the project), possible future features include the ability to import data other than crime records, and storing data over a network.

Insight will only target those living in English-speaking countries and support the English language, as the team is unable to translate to other languages and this would increase the workload significantly. Also, Insight will initially only support one data format, and as such dates will be displayed in the American month/day/year format. As Insight is only targeted at those in the USA, this will not cause a usability issue.

Product Features

Minimum Viable Product (MVP) Features

The features offered by Insight are designed to cater to these target markets. At its core, Insight is a GUI-based crime data visualisation tool. Users will be able to import existing crime data from csv files and view it in a table. The user will also be able to modify existing data by creating, editing or deleting records. The user will be able to filter the data to view subsets of the data based on filters like date ranges, crime types, locations, or other parts of the record. The user will also be able to sort by any of the fields. The user will be able to search records by their case number.

The user will be able to perform basic analyses of the data, such as calculating the distance or time between two crimes. They will also be able to view tables of the locations with the highest crime rates, lowest crime rates, most common types of crimes and least common types of crimes. In addition to this there will be an integrated web search feature which will limit users' searches to news or government sites.

The data will be saved persistently so the user can access it after reopening the application. The data can be saved into different lists and the user will be able to navigate between these lists by selecting them in the file viewer. The user will also be able to export the data in the table to csv files.

Insight's visualisation features will include mapping and graphing capabilities. Users will be able to apply the same filters to mapped data as to tabulated data. Insight will be able to

create graphs about the crime data, such as the number, type, or location of crimes over time.

Insight will initially only support crime datasets from the United States of America (USA), because the sample data is from the USA, and it is the largest English-speaking country.

Additional Features

Insight has several features on top of the MVP, which were initially planned for:

- A heat map to show crime density
- Loading bars when performing long operations like importing or exporting data
- Maps integrated into the add and edit crime record windows to let the user precisely edit the location of a crime record

Past the initial release at phase three of the project, Insight could include:

- Persistent file storage will be saved over the network so the user can access the data from any computer.
- Calculated data will also be able to be stored alongside the crime records.
- Support for New Zealand crime datasets.

Further Business Information

Competitive Advantages

The first release of Insight will offer the following competitive advantages:

- Free availability: This is one of the key market driving factors. Insight will be distributed as free software, which will appeal to its target market because they are unlikely to be able to afford commercial data visualisation software, making Insight an attractive, streamlined alternative.
- Designed for a specific group of target audience: As opposed to many data visualisation tools available on the market which target a much wider user base, Insight will be designed for a specific target market. Insight's features and design will be tailored to this target market.
- The simplicity of use: The use of data visualization software products is often complicated and requires training. However, Insight will be very simple and intuitive to use and will not require training; documentation will be provided.
- Portability: Insight will be able to work on Microsoft Windows, Linux, and OSX computers (desktops and laptops). This is necessary to be somewhat comparable to web-based alternatives.

Limitations

- Confined to crime data: The product's first release is only confined to work on crime data. This may disadvantage the market value of the product as the target market may need to visualize additional data on top of crime data to fit their purposes. For instance, activists may want to see crime data with other statistics such as population or income status. As outlined in the relevant business information section this limitation is going to be resolved in the future releases of the product.

- Limited to a single crime reporting code: The product can only handle a single crime data reporting code. This limits its use to cities, towns and places that use the same crime reporting codes. This limitation is also set to be resolved in future product releases.

Similar Extant Products

The New Zealand Police offer a “Crime Snapshot” visualiser on their website (New Zealand Police, 2021), which displays aggregate crime data by area within New Zealand, along with graphs of the types of crimes, crime type by year, and crime type by month. It also shows a table of the number of each crime type for the selected date range compared to the previous period. Compared to this, insight will offer the ability to view each crime as a data point on the map when that information is available, rather than just aggregate data by area. The New Zealand Police tool has limited filtering options, only allowing the user to select a date range, location, and all/violent/property crimes; Insight will offer greater filtering capabilities like an expanded range of crime types.

The LexisNexis Community Crime Map (LexisNexis, 2021) is more comparable to Insight. It maps individual crimes and provides similar graphs to the New Zealand Police tool, but does not allow the user to view aggregate data by area within a city, which Insight will be able to do, eg. by ZIP code in the USA. LexisNexis offers similar filtering capabilities to what Insight will offer - crimes can be filtered by date range, all crime types, and whether the offender was known to the victim.

One disadvantage of Insight compared to these products is that it is a standalone desktop application that requires the user to manually download and import a dataset in a supported format, while the two tools mentioned above are web-based, work on desktop and mobile devices, and automatically load data without the user’s input. However, Insight will be able to offer some of the capabilities of both apps (individual data points and data by area) in a desktop application, which may be faster to load than a web application as the dataset is already on the user’s computer. Furthermore, Insight’s GUI will be designed to be more user-friendly than these two apps.

It is worth developing the product as it is expected to be low-cost to develop and have a small but robust user base. After the first release, the file parser will be updated to read a wider array of crime data formats. The added flexibility in the parser will allow the software to be able to handle crime data from different major cities around the world. The project will be an open source software funded by user donations and having a global customer base will allow it to generate good annual returns. Given the small size of the project, the development costs will be low and will not require significant investment.

1.2 System Context

As shown in figure 2 below, the system context diagram shows the applications' use by users and its external connections to Google and the Chicago Data Portal. The arrows from the users to the application shows what actions the users can do. The application uses Google to get its map and search data and the application has support for crime data from the Chicago Data Portal.

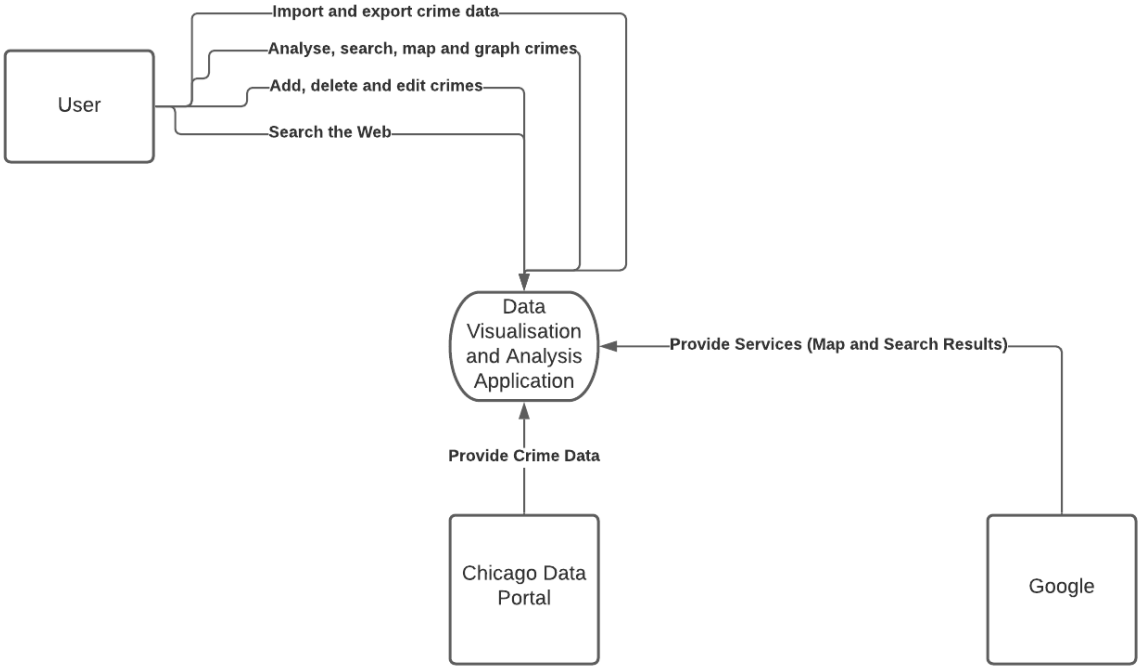


Figure 2 - System context diagram

2. Stakeholders and Requirements

2.1 Stakeholders

Table 1 shows the stakeholders for Insight. Stakeholders were brainstormed as falling into three categories: end users of Insight, internal stakeholders that relate to development, and other external stakeholders relevant during development and deployment. Stakeholders are prioritised based on how much the actions of the team should depend on them. For example, Google is critical to Insight's functionality, but will not be impacted by the development team whatsoever, and thus has a low priority. The concerns of stakeholders are their aims.

Some potential users of Insight that are not prioritised are frequent exercisers like cyclists or joggers, because although they want to know about crime in their area and figure out which places are safer than others, Insight does not offer a way to constantly monitor crimes in a very local area; it is more focused on longer time periods being very deliberately analysed. If these users were to be prioritised, some way to automatically download data from police departments would be considered.

ID	Stakeholder & Description	Concerns	Priority
S_01	**Activists** Activists have an interest in Insight because they will be able to use it in support of their activism. They will be able to do this by using Insight to learn more about crime and other issues in their community so that they can better understand the problems their community may be facing. Activists will be able to use the software to produce graphs, statistics, and tables that can influence policymakers in their communities. Activists will use Insight because it will be free and easy to use compared to expensive and complicated commercial applications. This is because activists do not generally have large budgets with which to lobby their communities and representatives. A subset of activists are nonprofits and advocacy groups, who may use Insight in their advocacy or research, but are likely to have access to commercial software.	● Want to be able to present data in graphs, tables, or statistics that are compelling and can influence others. ● Want a low-cost and easy to use tool. ● Software should handle data from their area of interest (local community, state, country, or other administrative subdivision).	High
S_02	**University & College Students**	● Want software that is free and	High

	Students have an interest in Insight because it may be useful for them in their projects where they need to analyse crime data in certain areas. They will be able to use Insight to produce graphs, statistics, and tables for their project. However, in this area, Insight is likely to face competition from other software like ArcGIS that can do similar things and has more analysis capabilities, like producing a heatmap. Students may have access to commercial software through their institution. The main advantage of Insight will be that it is free and easy to use.	easy to use. ● Want software that can export graphs or images. ● Want software that can perform analysis on the data to yield useful statistics, such as the most common type of crime in an area.	
S_03	**People Moving Location (Movers)** Those individuals and groups that are considering moving location may use Insight to inform their decision on where to move to, as they may not want to move to a location with more crime overall, or more violent crime, compared to other locations. Because movers are more "casual" users compared to activists and students, they are only a medium priority.	● Want an easy tool that can work in many locations in their country. ● Want analytics tools to look at types of crimes and different areas within a city.	High
S_04	**Course Staff** The course staff have an interest in how the project progresses because they are administering the course and grading the deliverables produced by the development team. As the development team, we must pay attention to what the course staff do and say because that determines how the project will run and what tasks need to be completed by what date.	● Want development to run smoothly. ● Want deliverables by project due dates. ● Software must be workable and meet the main requirements set out in the brief.	High
S_05	**Development Team** As the development team, we have an inherent interest in the success of the project and Insight because we are in charge of producing it and are also graded on our work. This means it is in our best interests to produce high quality, robust software and documentation that meets the requirements set out by the course staff and ourselves.	● Want good grades. ● Want to finish on time. ● Want to produce at least the minimum viable product. ● Want development to run well.	High
S_06	**Governments** Governments have a very low interest in Insight because it is a student project and not something they would use to analyse crime data. Governments may be impacted by the users of Insight, if they use it for	● Government policymakers want to be reelected, and so they may be inclined to listen to their constituents such as through	Low

	activism, which will bring more attention to the issue of crime in their country. This impact would be more pronounced at a local level, if there is an impact at all. Governments and their police forces also provide the crime data that Insight analyses. The type of crime data and what information it includes directly impact the features that Insight can provide to its users; however, the development team will not be able to change this and thus it is given a low priority.	submissions. ● In general, governments are not concerned about Insight.	
S_07	**Residents** Individual residents may use Insight to map crime in their local area for their own purposes, like deciding how they travel around and when. However, because Insight is desktop-only and requires a dataset to be downloaded, it is unlikely to be used “casually” in the same way that one checks the weather, and so residents are not considered to be the target market for Insight.	● Want to move around safely in their area and be informed of local dangers. ● As they may be of any age or demographic, residents will generally want an easy to use tool that is fast and can be checked quickly.	Low
S_08	**Google or Other API Provider(s)** Google will affect Insight because they will provide the mapping API (Google Maps) that it will use, thus making the functionality of Insight contingent upon the Google Maps API being available. Any outages or disruptions to this service will essentially stop Insight from functioning properly. Additionally, any changes to how the API is accessed (eg. it becomes paid rather than free) will necessitate a major change to Insight, potentially forcing us to use a different mapping API. However, as we are a group of university students, the actions of Google are well outside of our control.	● Are not concerned about Insight. ● May have terms of service or other agreements that preclude the use of their API.	Low

Table 1 - Stakeholder requirements

2.2 Requirements

UML Use Case Diagram

Figure 3 shows the use case diagram for Insight. Because Insight is a single-user application, the only actors are the user, the mapping and searching API provider (in this case Google) and the database.

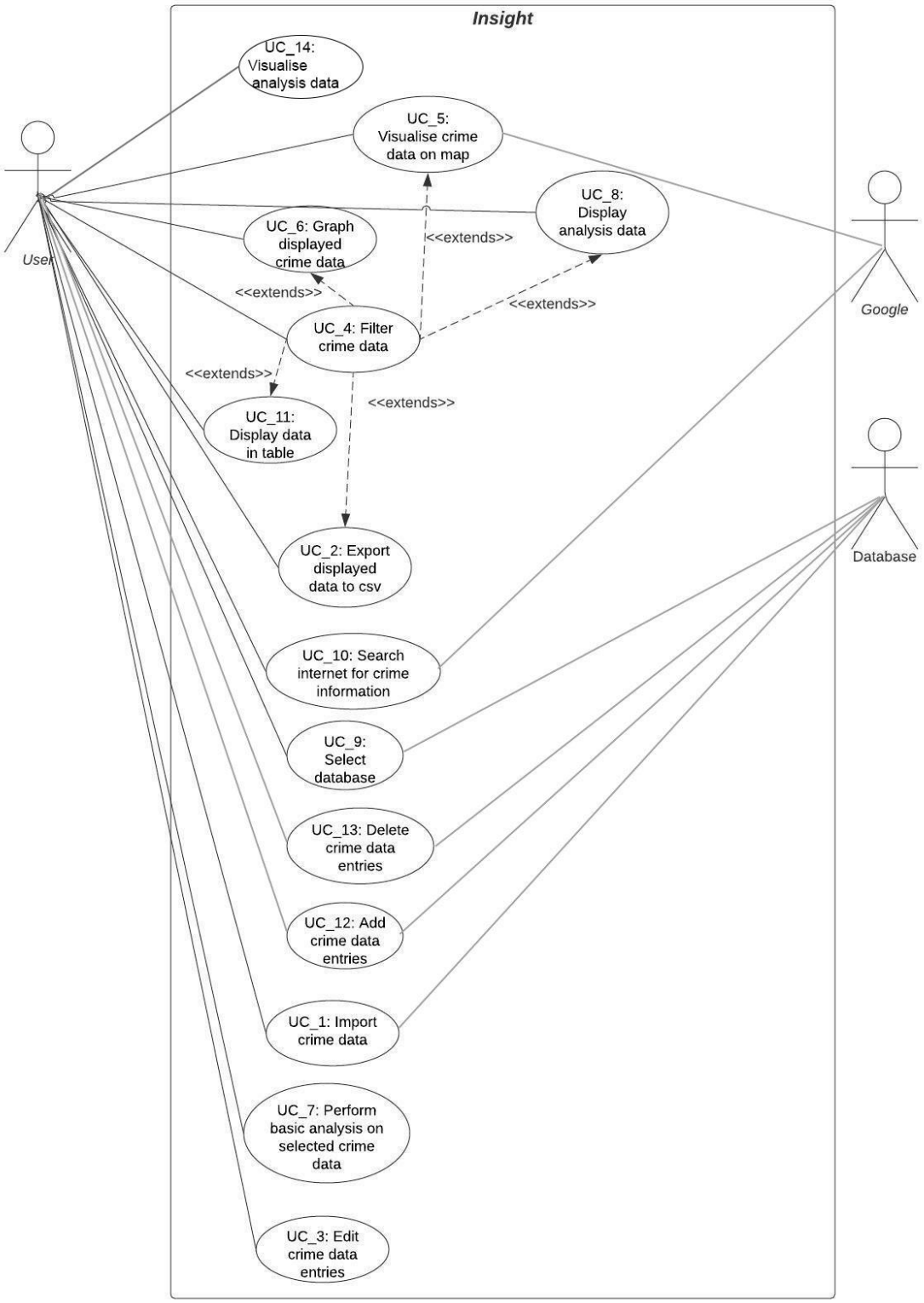


Figure 3 - UML Use Case Diagram

Textual Description of Use Cases

Table 2 below shows the textual descriptions of the use cases for Insight. These use cases reflect the core functionality of the app and how they interact with the GUI.

ID	UC_1
Name	Import crime data from a .csv file.
Actor(s)	User, Database
Precondition(s)	None.
Basic Flow	1. The user clicks the “Import CSV” button in the GUI under the “File” heading on the top bar. 2. Open file explorer for the user to select a .csv file to import. 3. Check that the file is a .csv file and contains crime data in a supported format. 4. The user is asked if they want to add the data to the current database or a new one. The user clicks “New Database”. 5. The new crime data is saved in a new database. 6. Add pins to the map for each data entry with location data.
Alternative Flow	At step 4 of the basic flow, the user clicks “Current Database”. They are then prompted if they would like to append to or replace the current data. If the user chooses to append, then the data should be displayed in the table alongside any existing data; if the user chooses to replace, then the data in the table should be replaced by the newly imported data.
Exceptional Flow	At step 2 of the basic flow, the program finds that the file type is incorrect, or the data in the .csv is invalid and displays an error pop-up to the user. The data loaded into the program and table is unchanged.
Post-Condition(s)	The data in the table in the main window shows the newly imported data.

ID	UC_2
Name	Export displayed data to .csv file.
Actor(s)	User.
Precondition(s)	There is data loaded into the app.
Basic Flow	1. User clicks the “Export CSV” button in the GUI, under the “File” heading from the top bar. 2. File Explorer opens for the user to select the destination folder and filename.

	3. A CSV file containing the data currently displayed in the program is saved at the specified location.
Alternative Flow	At step 2 the user selects a destination folder and filename that already exists, the program asks them if they would like to replace it.
Exceptional Flow	At step 2 the user closes the file explorer instead of selecting a file, the program cancels the export operation and no file is created.
Post-Condition(s)	If the file was created, a success message is displayed.

ID	UC_3
Name	Edit crime data entries.
Actor(s)	User.
Precondition(s)	● The program has a dataset loaded and is in either the table or map view.
Basic Flow	1. The user either double-clicks on a data record in the table, or clicks on a data record in the table to select it, and then either clicks the “Edit” button, or right-clicks on the table and select “Edit” from the menu. 2. The user inputs the changes they wish to make to the selected data record. 3. The user clicks the “Save changes” button. 4. The changes made are applied to the data record and a success message is shown.
Alternative Flow	At step 3 of the flow, the user clicks on another data record to select it, and they are prompted whether they want to save their changes (if any).
Exceptional Flow	At step 1, the user selects multiple records or no record and then clicks on the edit button. An error message stating that they must have one record selected is displayed. At step 3, the user enters invalid information and clicks the “Save changes” button. An error message is displayed telling the user which field is invalid and what the required format is. All invalid fields are outlined in red.
Post-Condition(s)	If one record was selected and the changes made to the record are valid, then the changes are saved in the database, the record is visually updated in the table, and a success message is displayed.

ID	UC_4

Name	Filter crime data.
Actor(s)	User.
Precondition(s)	● A dataset is loaded into the app
Basic Flow	1. The user selects a filter option from the side panel. a. Case Number: the user enters a case number. b. Date range: the user selects two different dates, where one is after the other. c. Crime type: the user checks a box for a crime type to toggle it in the current view. d. Location: the user selects a location through a drop down list and a text field for Ward, Beat. Latitude and Longitude. The user enters a radius to search around the given lat and long using a slider. e. Other metric: the user will select if the crime was domestic or if an arrest occured from a combobox. 2. The user clicks the “apply filters” button at the bottom of the side panel. 3. Only data matching the current filters is displayed.
Alternative Flow	At step 1.d the radius is set to 0 m, the filter will ignore the lat, long and radius field.
Exceptional Flow	At step 1.a, the user selects a second date before the first date and is notified that this is not valid with a message. The invalid date is not saved.
Post-Condition(s)	● Only data matching the current filters is displayed in the table, on the map, in any graphs, and in the analysis data.

ID	UC_5
Name	Visualise crime data on map.
Actor(s)	User, Google.
Precondition(s)	● There is a database loaded in the program.
Basic Flow	1. The user clicks on the “Map” tab. 2. The user clicks on, or is already in, the “Markers” mode. 3. The crime data is displayed on a map as individual markers. 4. The user clicks on an individual crime marker. 5. A popup appears above the record showing basic information about the crime (case number, date, description and location).

Alternative Flow	At step 2 of the basic flow, the user clicks on, or is already in, the “Heatmap” mode. The crime data is shown as a meatmap and the user cannot see or click on individual markers.
Exceptional Flow	At step 1 if the program can’t connect to the mapping API, the program displays an error message.
Post-Condition(s)	● Crime data is displayed on a map as data points.

ID	UC_6
Name	Graph displayed crime data
Actor(s)	User.
Precondition(s)	● There is a database loaded.
Basic Flow	1. The user clicks the Graph Data button in the GUI. 2. The user selects the type of graph they want from a drop down menu. 3. The user selects, from a combo box, what specific series they would like to graph (if applicable). 4. The user clicks “Generate Graph”. 5. A graph of the data is displayed in the main pane.
Alternative Flow	None.
Exceptional Flow	At step 2 of the basic flow, If there are no crime records loaded an error popup is displayed.
Post-Condition(s)	● The main window displays a graph of the data which has labelled x and y axes and a title.

ID	UC_7
Name	Perform basic analysis (comparison) on two crime records.
Actor(s)	User.
Precondition(s)	● There is a crime dataset loaded.
Basic Flow	1. The user selects two crime data records in the table. 2. The user either clicks on the “Compare Two Records” button, or right clicks on the table and clicks “Compare Two Records”. 3. A pop up window appears, showing the time and location difference between the two crimes, as well as their locations on a map.

Alternative Flow	None.
Exceptional Flow	At step 2 of the basic flow if there are more or less than two records selected, the program will display a message prompting the user to select only two records and will not proceed with the analysis.
Post-Condition(s)	• The result of the analysis is displayed. The user must close the popup to continue using the rest of the app.

ID	UC_8
Name	Display analysis data.
Actor(s)	User.
Precondition(s)	• There is a database loaded into the app.
Basic Flow	1. The user clicks on the “Analysis” tab. 2. Analysis data such as most and least common crimes is displayed.
Alternative Flow	None.
Exceptional Flow	None.
Post-Condition(s)	• The main window displays multiple tables each with different crime statistics.

ID	UC_9
Name	Select Database
Actor(s)	User, Database
Precondition(s)	None.
Basic Flow	1. The user clicks the “Select Database” button in the GUI under the “File” heading on the top bar. 2. Open file explorer for the user to select a .db file to import. 3. Check that the .db file is a valid database and contains crime data in the correct format. 4. The new database data is displayed in the table and any data manipulation will be saved to the new database.
Alternative Flow	None.

Exceptional Flow	At step 4 if the selected database file is not valid, the program will display an error popup and cancel the select database operation.
Post-Condition(s)	● Crime data is now read from the new database, and the data is displayed in the table, map, and analysis tabs.

ID	UC_10
Name	Search internet for crime information
Actor(s)	User, Google
Precondition(s)	None.
Basic Flow	1. The user clicks on the “Web Browser” tab. 2. The user selects if they want government websites or news results. 3. The user types their query into the text field and clicks search. 4. The program displays results for that query based on which option was selected
Alternative Flow	None.
Exceptional Flow	At step 3 if the program cannot connect to Google, an error message is displayed.
Post-Condition(s)	The browser window displays search results to the user based on their query.

ID	UC_11
Name	Display data in table
Actor(s)	User
Precondition(s)	● There is a database loaded
Basic Flow	1. The user clicks on the “Table” tab. 2. Data in the current database is shown in a table.
Alternative Flow	1. User has applied a filter, then only the data which corresponds to the filter. 2. There is no data in the table so an empty table is displayed, showing a message explaining why the table may be empty and how to fix it.
Exceptional Flow	None.
Post-Condition(s)	All data that matches the user’s filter(s) in the loaded

	database is shown in the table.

ID	UC_12
Name	Add crime data entries.
Actor(s)	User.
Precondition(s)	None.
Basic Flow	1. User clicks “Add Record” button 2. The user inputs the data for the record they would like to add 3. The user clicks the “Save record” button. 4. The record is added to the database and is visible in the table. A success message is displayed.
Alternative Flow	At step 3 of the flow, the user clicks on another data record to select it, and they are prompted whether they want to save their changes (if any).
Exceptional Flow	At step 3, the user enters invalid information and clicks the “Add Record” button. An error message is displayed telling the user which field is invlaid and what the required format is. All invalid fields are outlined in red.
Post-Condition(s)	If one record was added and it is valid, then the changes are saved in the database and a success message is displayed.

ID	UC_13
Name	Delete crime data entries.
Actor(s)	User.
Precondition(s)	None.
Basic Flow	1. User selects records in the Table tab. 2. User either: a. clicks the “Delete Selected Row(s)” button; b. presses the DELETE or BACKSPACE keys on their keyboard with the table element being in focus; or c. right-clicks on the table and click the “Delete” option. 3. Regardless of the option chosen above, a popup is shown to the user, showing them the number of rows to be deleted and asking them for confirmation.. 4. The user clicks “Yes”. 5. The rows are removed from the table and the

	database. A success message is displayed.
Alternative Flow	At step 4 of the basic flow, the user clicks “No” or closes the popup. No rows are deleted or changed.
Exceptional Flow	At step 2 of the flow, if no entries are selected, an error message will be displayed and nothing will be removed.
Post-Condition(s)	The row is removed from the database. A success message is displayed.

ID	UC_14
Name	Visualise analysis data.
Actor(s)	User.
Precondition(s)	There is a valid dataset loaded into the app, and the user is in the “Analysis” tab.
Basic Flow	1. The user clicks on one of the “Visualise” buttons above a particular analysis table. 2. A popup window with a pie chart of the data in the table is displayed. The user must close the popup to use the rest of the app.
Alternative Flow	None.
Exceptional Flow	None.
Post-Condition(s)	No changes are made to the data or current app instance.

Table 2 - Textual Use Case Descriptions

Quality Requirements

Table 3 describes the expected requirements for the software from the customers and developers. The description column describes the requirement. The stakeholder column lists the stakeholders that this quality requirement is relevant to. The priority column describes the priority in which we implement the requirements.

The quality requirements correspond to the areas of importance identified in our key drivers analysis, specifically usability (QR_2, QR_4, QR_11), security (QR_10), speed (QR_6, QR_7, QR_8), maintainability (QR_3, QR_5), and API terms (QR_9).

ID	Description	Stakeholder	Priority
QR_1	The product should be completed within the time period shown in the project plan.	Development team (S_05) Course staff (S_04)	High
QR_2	The product should function correctly, as described in the use cases, under blue sky scenarios where the user only performs valid actions.	Development team (S_05) Course staff (S_04) Activists (S_01) Students (S_02) Movers (S_03) S_07 (Residents)	High
QR_3	The product should have well documented code that allows for maintenance. This means that all classes and methods have javadoc (except simple getters and setters), and methods that do not have clearly understandable code should have inline comments to explain what the code is doing.	Development team (S_05) Course staff (S_04)	High
QR_4	Given a valid .csv file, .db file, and the user manual, an average user that has not used Insight before and does not extensively use computers should be able to complete all use cases (main flow) without assistance from another person and within three minutes each (processing/loading times notwithstanding).	Development team (S_05) Activists (S_01) Students (S_02) Movers (S_03) S_07 (Residents)	High
QR_5	The product’s code should allow for the reuse of code wherever possible, through following object oriented principles, or using static methods / variables. This means that	Development team (S_05) Course staff (S_04)	High

	code should not be duplicated where possible. Specifically, all reasonably actionable warnings in IntelliJ should be acted upon and removed. Examples of warnings that aren't reasonably actionable are: ● Raw use of parameterized class ● Unchecked call to <method> as a member of raw type		
QR_6	The product should respond in less than five seconds for use cases involving operations on a dataset of 200,000 or fewer records: ● Importing or exporting data (UC_1, UC_2); ● Sorting data; ● Applying filters to the data, including refreshing the table, (UC_4); and ● Loading data on the map (UC_5).	Development team (S_05) Activists (S_01) Students (S_02) Movers (S_03) Residents (S_07)	High
QR_7	The following use cases should have a response time of less than one second: ● Adding a record to the table (UC_12); ● Editing a record in the table (UC_3); ● Deleting record(s) from the table UC_13); ● Creating a graph of the data (UC_6); and ● Loading the "Analysis" tab and creating the data in it (UC_8).	Development team (S_05) Activists (S_01) Students (S_02) Movers (S_03) Residents (S_07)	High
QR_8	If an operation takes longer than one second, a loading / progress bar should be shown to the user. Thus, the following use cases should display a progress bar: ● Importing or exporting data (UC_1, UC_2); ● Filtering data (UC_4); ● Sorting data in the table; and ● Adding data to the map (UC_5).	Development team (S_05) Activists (S_01) Students (S_02) Movers (S_03) Residents (S_07)	Medium
QR_9	The product should operate inline with the terms of use of the Google Maps API.	Development team (S_05) Google (S_10)	Medium
QR_10	The product should not allow for the user to directly send SQL queries to the database; rather, queries should be generated automatically by the product when the user performs certain actions, like applying filters, or adding / editing / deleting data.	Development team (S_05) Activists (S_01) Students (S_02) Movers (S_03) Residents (S_07)	Medium

QR_11	Given a valid .csv file, .db file, and the user manual, an experienced user (who has used Insight for at least six hours) should be able to complete all use cases (main flow) without assistance within one minute each (processing/loading times notwithstanding).	Development team (S_05) Activists (S_01) Students (S_02) Residents (S_07)	Medium

Table 3 - Quality Requirements

Key Drivers

Table 4 shows the key quality drivers for Insight. Each stakeholder is weighted by its priority and a weighting of 1, 0.5 and 0.1 is given for high, medium and low priorities respectively. Google (S_08) was included and given a weight of 0.2 because although they are not concerned about Insight as a project and are thus a low priority stakeholder, Insight does use the Google Maps API, and is thus bound by its terms of service, which Google does care about. S_06 (Governments) was excluded from this table because they have no interest in the quality requirements of our project. The totals were calculated using a weighted sum, where the totals for each key driver were calculated using a sum of each stakeholder's weight multiplied by their score for that key driver.

Activists, and to a lesser extent students and residents, were given scores reflecting that they are likely to use the software regularly (not as a one-off) and would probably process large data sets. Movers were given scores reflecting that they want a simpler to use product that they won't necessarily use regularly. Course staff and developers were given scores reflecting the importance they place on the code of the project, along with the quality at runtime, hence why they are not concerned about cost. The sole concern of Google is that we adhere to their terms of service - otherwise, our product is inconsequential to them.

The specific key drivers were roughly defined as follows:

- **Usability:** how easy it is for a user to complete a desired use case in the app. This includes things like the visibility of buttons, clarity of feedback (e.g. validation feedback, loading bars) and error messages, confirmation messages before important actions, and the use of intuitive controls like using the delete key to delete a record. Higher scores reflect a user's desire for the app to be usable - some users may be more tech-savvy than others.
- **Advanced functionality:** how customisable is the app; can it produce very specific results like statistics or graphs? How many options does the user have? For example: the number of different statistics available, or the range of graph customisation options. Higher scores represent a desire for more features and customisation.
- **Security:** is the app vulnerable to or protected from a potential attack? Higher scores reflect more security.
- **Cost:** is there a cost to obtain Insight, and if so, how much? Users prefer lower costs and higher scores reflect this.

- **Speed:** how long does the app take to process data, particularly large datasets? Higher scores represent a faster desired speed.
- **Maintainability:** how easy is it for the developers to make changes or fixes to the app in the future? Higher scores show a greater interest in continued support, or ease of development.
- **API Licence:** does the app comply with all relevant terms of use, chiefly the Google Maps API terms? Higher scores reflect a greater importance.

The consequence of the key driver analysis is that we as the development team need to prioritise the usability and speed of our product over advanced functionality, and that Insight should be free or very low cost - our target market is not wealth and it is also highly unlikely that they would spend much (if any) of their money on a student project. Advanced functionality is still important for our target market however; it is just less important than the basic ease of use of the app. Furthermore, the security and maintainability of the app is not a significant priority for our users, however Insight should still be maintainable for the sake of easier development and performance, and it should be secure where possible (though Insight only connects to the internet to access Google Maps, and is designed to work on open source data, so presents a low security risk in the first place).

ID	Stakeholders	Weight	**Key Drivers**							Total
			Usability	Advanced Functionality	Security	Cost	Speed	Maintainability	API Licence	
S_01	Activists	1	25	25	10	10	20	10	5	100
S_02	University & College Students	1	20	20	5	25	25	5	0	100
S_03	Movers	1	35	15	5	20	20	5	0	100
S_04	Course Staff	1	40	25	10	0	20	5	0	100
S_05	Development Team	1	30	35	5	0	20	10	10	100
S_07	Residents	0.1	30	20	10	10	20	10	0	100
S_08	Google	0.2	0	0	0	0	0	0	100	100

	Total	**153**	**122**	**36**	**56**	**107**	**36**	**35**	

Table 4 - Key Drivers

3. Acceptance Tests

Table 5 below contains tests describing the desired behaviour of the system. They each have an ID number, a description, acceptance criteria (the desired behaviour), a criticality score, and the number of the use case they correspond to. The criticality ratings are either high, medium, or low. Tests with a high criticality are testing features that are essential to our stakeholders, whereas tests with a low criticality rating, such as AT_10 or AT_11 are not. This may be because these features likely won't be used often, or, in the case of AT_10 and AT_11, because there are readily available alternatives that may also meet our stakeholders needs.

ID	Description	Acceptance Criteria	Criticality	Use Case
AT_01	The user should be able to graph data.	● Given there is data loaded into the system ● When the user selects graph options ● And the user tries to create a graph ● Then the user should see a graph in the graph tab matching the options they selected in the graphing sidebar tab	High	UC_6
AT_02	The user should be able to import a valid CSV file.	● Given a CSV file in the correct format for the program ● When the user tries to load the CSV into the system ● Then the user should see the data appear in the table view	High	UC_1, UC_11
AT_03	The user should be able to import an existing database file.	● Given a valid database file in the correct format for the program ● When the user tries to load the database file into the system ● Then the user should see the data appear in the table view	Medium	UC_9, UC_11
AT_04	The user should be able to export data to a CSV file.	● Given there is data loaded into the system ● When the user clicks export data ● And the user selects a valid file location to save the new file ● Then the user should see a new CSV file appear in the selected location that contains the data from the program	High	UC_2

AT_05	The user should be able to add a new crime record.	● Given the user has entered valid information for a crime record ● When the user tries to save the record ● Then the user should see the new record in the table view	Low	UC_12, UC_11
AT_06	The user should be able to edit a crime record.	● Given the user has made valid edits to the record ● When the user tries to save the record ● Then the user should see the updated record in the table view	Medium	UC_3, UC_11
AT_07	The user should be able to delete records from the database.	● Given there is data in the table view ● When the user tries to delete a record or records ● Then the user should see the record or records disappear from the table view	Medium	UC_13, UC_11
AT_08	The user should be able to filter data.	● Given there is data loaded into the system ● When the user tries to filter data ● Then the user should see the filtered data in the table view	High	UC_4, UC_11
AT_09	The user should be able to clear any applied filter conditions.	● Given there is a filter applied to the data ● When the user tries to clear the filter ● Then the user should see all data in the database in the table view	High	UC_4, UC_11
AT_10	The user should be able to search for news articles on the internet.	● Given the user has an internet connection ● When the user selects the news results browser option ● And the user tries to search a query ● Then the user should see news articles related to their search query	Low	UC_10
AT_11	The user should be able to search for government websites on the internet.	● Given the user has an internet connection ● When the user selects the government websites browser option ● And the user tries to search a query ● Then the user should see government websites related to their search query	Low	UC_10
AT_12	The user should be able to see	● Given there is data loaded into the system	High	UC_5

	the data as markers on a map.	● When the user selects the markers map option ● Then the user should see the data displayed as markers on a map		
AT_13	The user should be able to see the data as a heatmap.	● Given there is data loaded into the system ● When the user selects the heatmap map option ● Then the user should see the data as a heatmap	High	UC_5
AT_14	The user should be able to compare two crime records and see the results.	● Given the user has selected two data records ● When the user tries to compare the records ● Then the user should see the results of the comparison (distance and time between the crimes) in a popup	Medium	UC_7
AT_15	The user should be able to see the most and least common crime types.	● Given there is data loaded into the system ● When the user selects the Analysis tab ● Then the user should see the most and least common crime types in the loaded data	Medium	UC_8
AT_16	The user should be able to see the blocks with the highest and lowest crime rates.	● Given there is data loaded into the system ● When the user selects the Analysis tab ● Then the user should see the blocks with the the highest and lowest crime rates in the loaded data	Medium	UC_8

Table 5 - Acceptance Tests

4. GUI Prototypes

Figures 5 and 7 below show how Insight's GUI has changed since the initial prototype. The filter side panel has had significantly more parameters added, this was done to allow the users to more accurately and easily select the exact data that they want. The arrest and domestic check boxes were also changed to combo boxes as this allowed for the user to ignore these fields in the filter.

The filter and graph sidebar tabs were replaced with an accordion layout. This was done as it is more consistent with the toggle columns menu added shown in figure 7. It also eliminates any confusion with the other tabs on the for the main window of the program.

The map remained mostly consistent with the original prototype except for the addition of a heatmap and a redesign of the popup for a selected record. The heatmap was added to give the users more choice in how they visualised the data. The record popup was changed to make the data more readable and user friendly.

All of the navigation bar items except for "File" were removed. This was done as we opted to move the all functionality contained in these to a bar below the table and the right-click menu with the exception of sorting which was available already by clicking on the table headers, and any separate sorting GUI would have been more confusing for the user (shown in figure 7). This was done in order to make this functionality more visible and intuitive to the user.

The graphing sidebar was simplified from the original GUI prototype. This was done as our users will not need to be able to graph any two variables against each other as in the original prototype (e.g. graphing Ward against Beat does not make sense). By having a selection of graphs ready to be generated our users can more quickly and easily create what they're looking for. By doing this the program also removes some of the user error that could come from less experienced users being given too much choice.

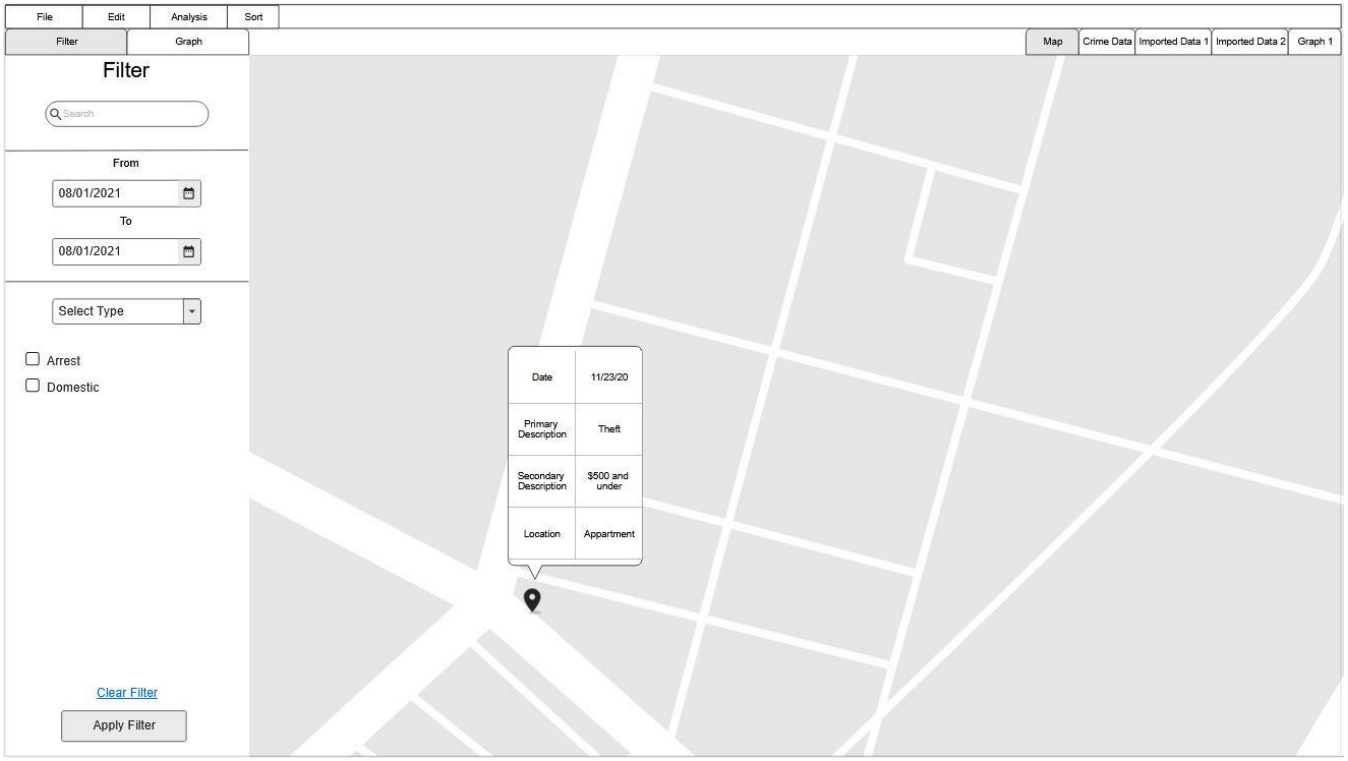


Figure 4: Original GUI prototype map screen and filter panel

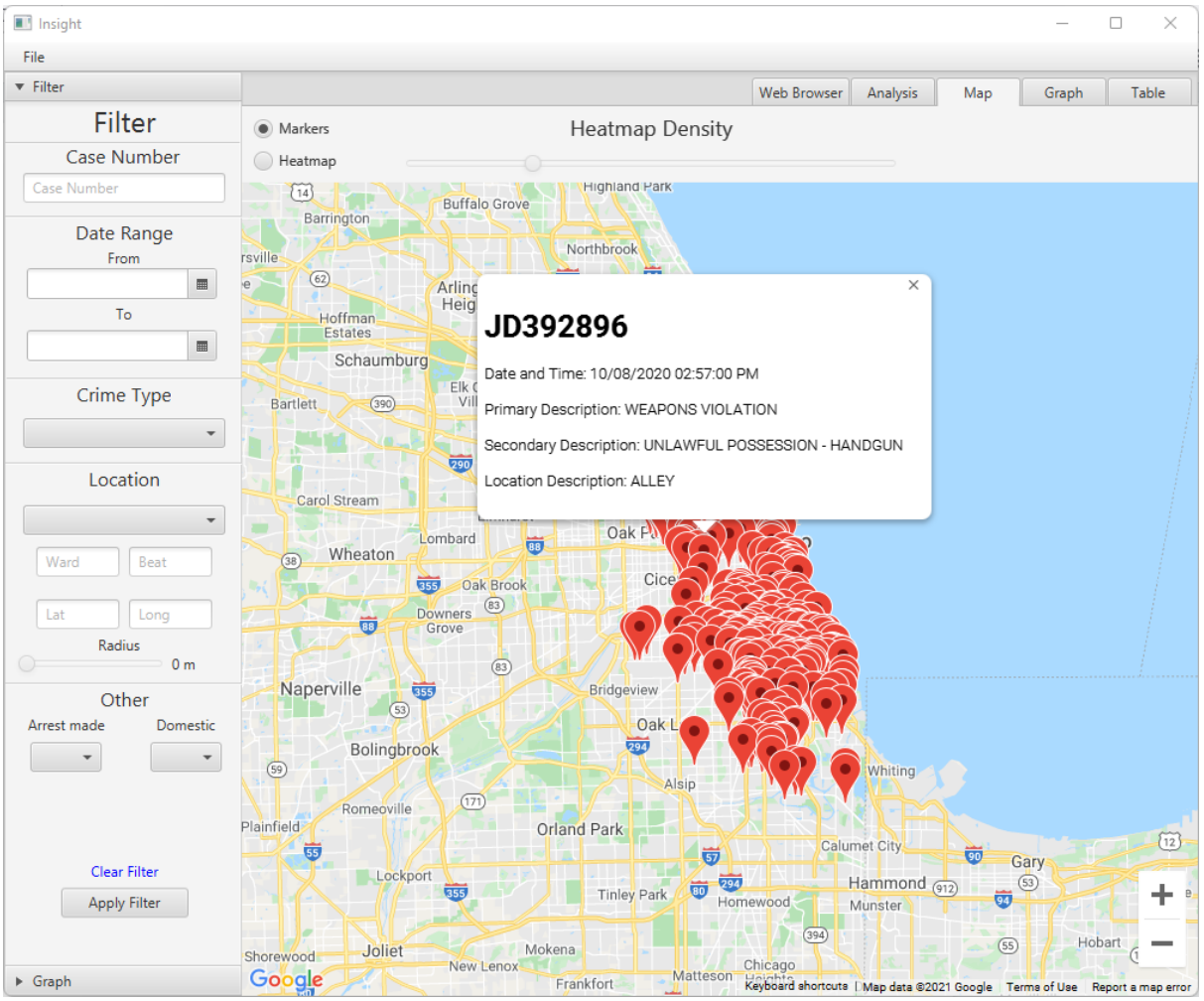


Figure 5: Current map screen and filter panel

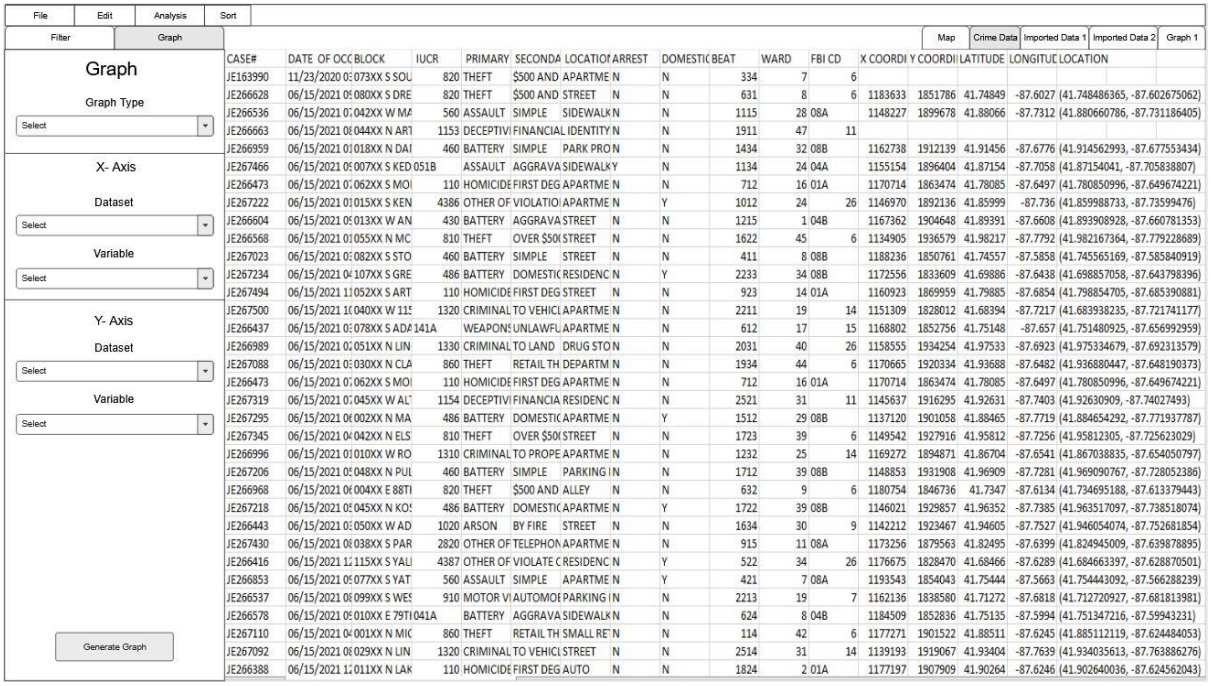


Figure 6: Original GUI prototype table screen and graph panel

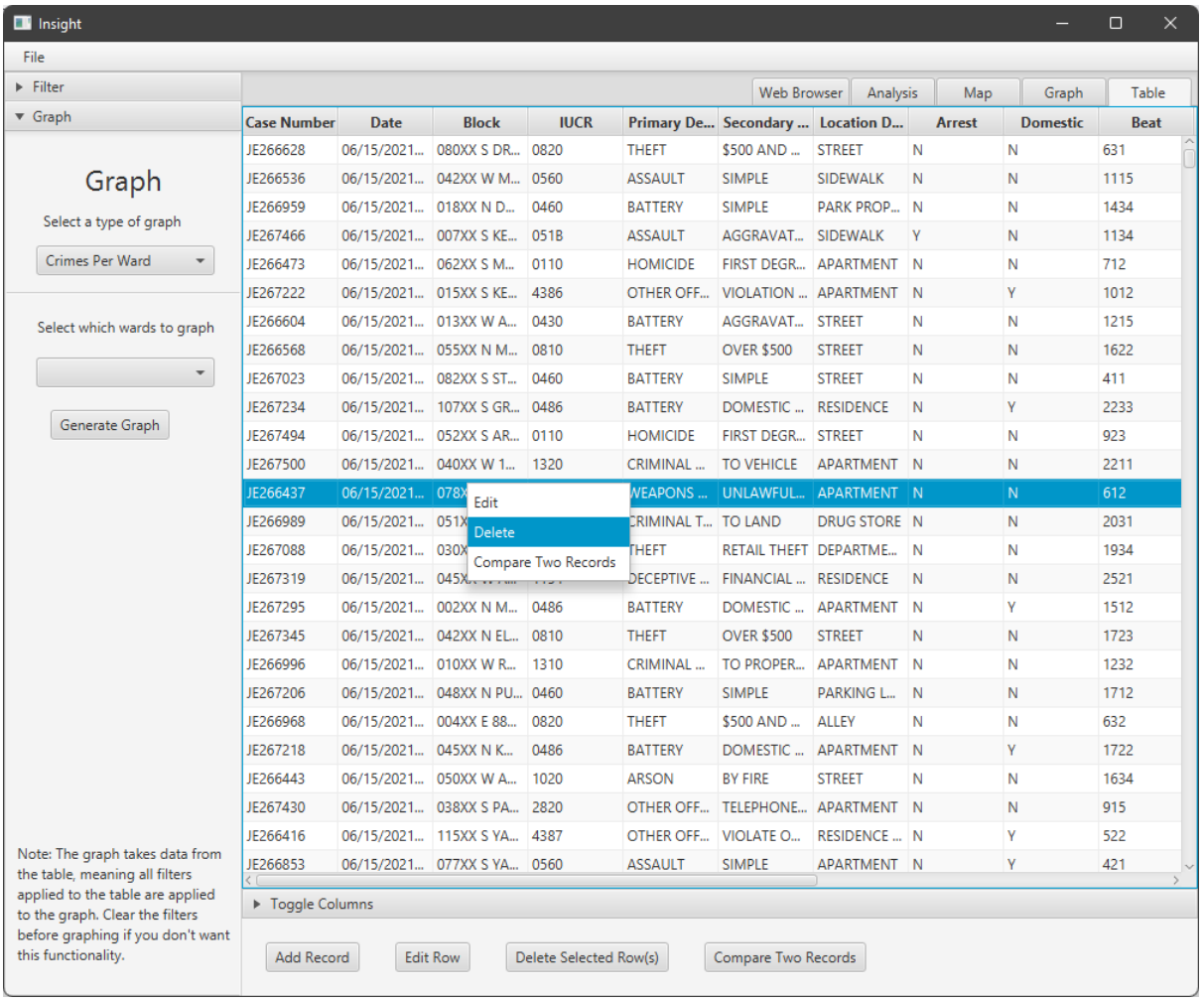


Figure 7: Current table screen and graph panel

5. Deployment Model

Insight's first release is a desktop application. As shown in figure 8 below, all artifacts except Google Maps server-side services live on the user node. To run the product a user will need access to any node with 256 MB of RAM, 500 MB of free secondary storage and installed hardware for internet connectivity. The Java runtime environment also needs to be installed on the user node.

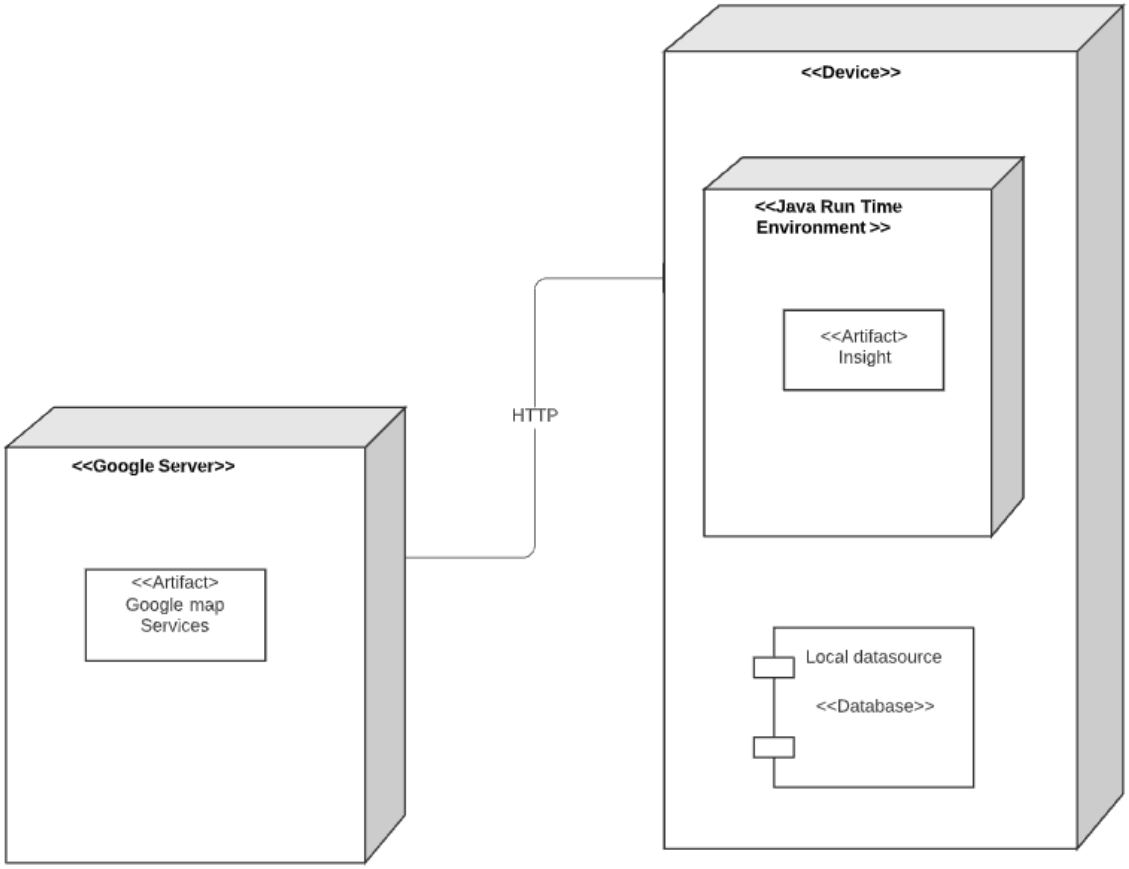


Figure 8 - Deployment High-Level UML Diagram

6. UML Class Diagram

There is a full-size image version of the UML diagram included in the submission.

The UML class diagram below (figure 9) shows the current layout of the system. You can see that there are 3 separate blocks in the class diagram. The first consists of the Main and MainGUI classes. This is used to start the program by loading the main fxml file, which in turn, loads the relevant controllers and displays the main window.

The second block consists of the BrowserTabController class. Because it does not interact with any other Java classes, only fxml files, it does not have any associations in the class diagram. Its fxml file is loaded by the main fxml when the browser tab is opened, which is what starts the BrowserTabController instance, allowing the browser tab to operate.

The third block consists of the rest of the program. When the program starts, the main fxml file creates an instance of the MainController, which then loads all the data required for the program. The MainController uses most of the backend classes to update the view as required by the user's inputs.

For the most part, the system follows the MVC design pattern, with the exception of some of the controller classes talking to each other, due to the way the program view is implemented. The main window has tabs, each of which have their own controller to update the information displayed in them, however, there is also a side menu on the window where there are options that apply to the whole program, such as filtering data. These options fall under the scope of the MainController, however, the MainController needs to pass the requests on to more relevant controllers, such as the TableTabController, to update the data in that tab.

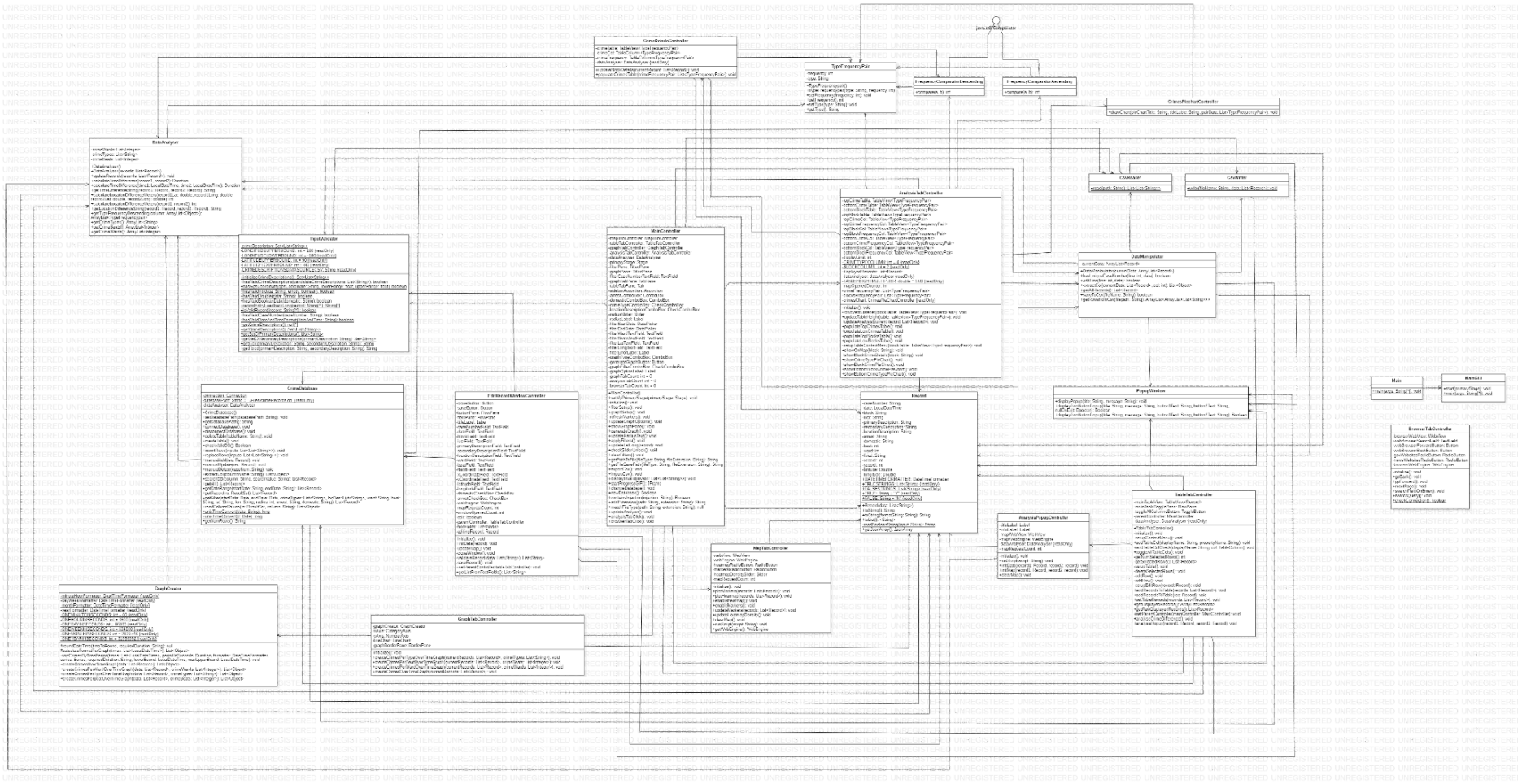


Figure 9 - UML Class Diagram

7. Testing Procedures

7.1 Functional Testing

Item	Description
Test (or ID)	Import CSV to new database
Description	Select File -> Import CSV, then select the full list of records to be imported into a new database.
Tester	Daniel
Result	The program made a new database file with the new records and displayed it in the app.
Pass/fail	Pass
Criticality	High
Use case	UC_1
Test (or ID)	Import CSV and append to existing data
Description	Select File -> import CSV -> Existing Database -> Append, to tell the program to import a new csv and append it to the end of the current data.
Tester	Daniel
Result	The selected csv is appended to the end of the current data.
Pass/fail	Pass
Criticality	High
Use case	UC_1
Test (or ID)	Import CSV and replace existing data
Description	Select File -> import CSV -> Existing Database -> Replace, to tell the program to replace all the current data with the new csv
Tester	Daniel
Result	The original data is replaced with the data from the selected csv
Pass/fail	Pass
Criticality	High
Use case	UC_1
Test (or ID)	Import invald CSV to new database
Description	Select File -> import CSV -> New Database, selecting an invalid file instead of a valid crime record csv
Tester	Daniel
Result	An error popup is displayed saying that an unknown error occurred when importing CSV if the file is not a CSV. If it is a CSV file but an invalid one a popup will show up saying that the CSV is in an invalid format.

Pass/fail	Pass
Criticality	High
Use case	UC_1
Test (or ID)	Import non csv file
Description	User selects ImportCSV and selects a file that does not have the .csv extension
Tester	George
Result	An error pop up is displayed and nothing is imported
Pass/fail	Pass
Criticality	Low
Use case	UC_1
Test (or ID)	Export table data to CSV
Description	Select File -> export CSV and then select a location and name for the new CSV file.
Tester	Daniel
Result	A new file is created with the specified name at the specified location containing all the currently displayed records.
Pass/fail	Pass
Criticality	High
Use case	UC_2
Test (or ID)	Select a new database
Description	Select File -> Select Database then select a database file
Tester	Daniel
Result	The contents of the selected database are displayed and any changes will be saved to that database
Pass/fail	Pass
Criticality	Medium
Use case	UC_9
Test (or ID)	Selecting an invalid database file
Description	Select File -> Select Database then select a file that isn’t a valid database
Tester	Daniel
Result	An error popup is displayed if the selected file is not a .db file or if it’s not valid for the program
Pass/fail	Pass
Criticality	Low
Use case	UC_9
Test (or ID)	Filter by case number

Description	Enter a case number into the case number text field and click apply filter
Tester	Daniel
Result	Only records matching or similar the input case number are displayed. If the input case number is not complete then records with similar case numbers will be displayed.
Pass/fail	Pass
Criticality	High
Use case	UC_4
Test (or ID)	Filter records after a specific date
Description	Select a date under the first date picker and click apply filter
Tester	Daniel
Result	Only records occurring on or after the selected date are displayed
Pass/fail	Pass
Criticality	High
Use case	UC_4
Test (or ID)	Filter records before a specific date
Description	Select a date under the second date picker and click apply filter
Tester	Daniel
Result	Only crimes that occured before or on the selected date are displayed
Pass/fail	Pass
Criticality	High
Use case	UC_4
Test (or ID)	Filter records between a specific date range
Description	Select a date in both data pickers and click apply filter
Tester	Daniel
Result	Only crimes that occured on or between the selected dates are displayed
Pass/fail	Pass
Criticality	High
Use case	UC_4
Test (or ID)	Filter records by crime types
Description	The user should be able to filter crimes to specific crime types by selecting the desired crime types from the crime type combo box and clicking apply filter
Tester	Daniel
Result	Records with a primary description matching any of the selected crime types are displayed.
Pass/fail	Pass

Criticality	High
Use case	UC_4
Test (or ID)	Filter records by crime location description
Description	The user should be able to filter crimes to specific location descriptions by selecting the desired location descriptions from the location description combo box and clicking apply filter
Tester	Bede
Result	Records with a location description matching any of the selected location descriptions are displayed.
Pass/fail	Pass
Criticality	High
Use case	UC_4
Test (or ID)	Filter records by ward
Description	The user should be able to filter crimes to specific wards by entering the desired ward in the ward text field and clicking apply filter
Tester	Bede
Result	Records with a ward matching the specified ward are displayed
Pass/fail	Pass
Criticality	High
Use case	UC_4
Test (or ID)	Filter records by beat
Description	The user should be able to filter crimes to specific beats by entering the desired beat in the beat text field and clicking apply filter
Tester	Bede
Result	Records with a ward matching the specified beat are displayed
Pass/fail	Pass
Criticality	High
Use case	UC_4
Test (or ID)	Filter records by radius from latitude and longitude
Description	The user should be able to filter records based on their distance from a specific location by entering the desired radius, latitude and longitude
Tester	Bede
Result	Records with a location value within the radius from the specified location are displayed
Pass/fail	Pass
Criticality	High
Use case	UC_4
Test (or ID)	Filter records by arrest made

Description	The user should be able to filter crimes to whether an arrest was made by selecting the desired option in the arrest made drop down and clicking apply filter
Tester	Bede
Result	Records with an arrest matching the specified arrest value are displayed
Pass/fail	Pass
Criticality	High
Use case	UC_4
Test (or ID)	Filter records by domestic
Description	The user should be able to filter crimes to whether the record was domestic by selecting the desired option in the domestic drop down and clicking apply filter
Tester	Bede
Result	Records with an arrest matching the specified arrest value are displayed
Pass/fail	Pass
Criticality	High
Use case	UC_4
Test (or ID)	Clear filter
Description	The user should be able to clear all filters by clicking the clear filter button
Tester	Bede
Result	All records stored in the current database are displayed
Pass/fail	Pass
Criticality	High
Use case	UC_4
Test (or ID)	Generate graphs with no data loaded
Description	The user shouldn’t be able to graph data when there is no data loaded into the system.
Tester	Bede
Result	When the user attempts to select a graph type, a popup box appears telling them to clear the filter or import some data, and the graph type is deselected
Pass/fail	Pass
Criticality	Low
Use case	UC_6
Test (or ID)	Generate graphs with one record loaded
Description	The user should be able to graph each type of graph with one record loaded in to the system
Tester	Bede
Result	All graphs display correctly, with the correct times

Pass/fail	Pass
Criticality	Low
Use case	UC_6
Test (or ID)	Generate graphs with full 200k record set loaded
Description	The user should be able to graph each type of graph with a large record set loaded in to the system
Tester	Bede
Result	The user is able to correctly generate graphs using all 200k records
Pass/fail	Pass
Criticality	High
Use case	UC_6
Test (or ID)	Generate graphs with more than 5 options selected
Description	The user should get a popup telling them to deselect options if they attempt to graph with more than 5 options selected
Tester	Bede
Result	The user gets a popup telling them that at most 5 options are allowed
Pass/fail	Pass
Criticality	Low
Use case	UC_6
Test (or ID)	Browser back button
Description	The browser goes back to the page the user was last on, if any, when the back button is pressed
Tester	Bede
Result	The browser goes back to the page the user was last on, if there is a page, otherwise stays on the same page
Pass/fail	Pass
Criticality	Medium
Use case	UC_10
Test (or ID)	Browser back button
Description	The browser goes forward to the next page in the users history, if any, when the forward button is pressed
Tester	Bede
Result	The browser goes forward to the next page in the users history, if there is a page, otherwise stays on the same page
Pass/fail	Pass
Criticality	Medium
Use case	UC_10
Test (or ID)	Browser reset button

Description	The browser should reset to the home page when the reset button is pressed
Tester	Bede
Result	The back and forward buttons disappear, the search bar is emptied, and the browser goes to a blank screen.
Pass/fail	Pass
Criticality	Low
Use case	UC_10
Test (or ID)	Browser search button
Description	The browser should search the query in the search bar when the search button is pressed, or when the user presses the enter key while in the search bar, unless the search bar is empty
Tester	Bede
Result	If the search bar is empty, the browser stays as is. If the search bar has a query in it, the browser searches the query.
Pass/fail	Pass
Criticality	High
Use case	UC_10
Test (or ID)	Browser search options
Description	The browser should search for .gov websites only if the “government websites” option is checked, or for news articles only if the “news results” option is checked
Tester	Bede
Result	The browser searches only for .gov pages or news articles, depending on which option is checked
Pass/fail	Pass
Criticality	Medium
Use case	UC_10
Test (or ID)	Map updates with filter
Description	The markers on the map should change when the filter on the data changes
Tester	Bede
Result	When the filter changes, the map markers are cleared then replaced with new ones
Pass/fail	Pass
Criticality	High
Use case	UC_5
Test (or ID)	Analysis tables work with less than 10 different crime types or blocks
Description	The analysis tables should display all the crime types or blocks they have data for if the total number is less than 10
Tester	Bede

Result	The analysis tables display up to 10 rows and function as expected with less than 10 crime types or blocks
Pass/fail	Pass
Criticality	Medium
Use case	UC_8
Test (or ID)	Analysis tables work with more than 10 different crime types or blocks
Description	The analysis tables should display the top or bottom 10 crime types or blocks if the total number is more than 10
Tester	Bede
Result	The analysis tables display the top or bottom 10 rows with more than 10 different crime types or blocks
Pass/fail	Pass
Criticality	Medium
Use case	UC_8
Test (or ID)	Analysis tables are empty when there is no data
Description	The analysis tables should be empty when the persistent data storage is empty
Tester	George
Result	All analysis tables say “No content in table”
Pass/fail	Pass
Criticality	Medium
Use case	UC_8
Test (or ID)	Analysis tables sort by count then name
Description	If two or more of the crime types or blocks have the same count, the analysis tables should order them depending on the alphabetical order of their name
Tester	Bede
Result	The analysis table correctly sorts crime types or blocks by name if they have the same count
Pass/fail	Pass
Criticality	Low
Use case	UC_8
Test (or ID)	Analysis tables update with the filter
Description	When the user filters the current data, the analysis tables should be updated to show the change in data set
Tester	Bede
Result	The tables update when the user filters the current data
Pass/fail	Pass
Criticality	High

Use case	UC_8
Test (or ID)	User selects whole table (full crime data file used)
Description	User selects every record in the table using Ctrl + A.
Tester	George
Result	Program freezes
Pass/fail	Fail
Criticality	Medium
Use case	UC_11
Test (or ID)	Analysis with no records
Description	An analysis button pressed with no records selected
Tester	George
Result	The exceptional flow for UC_7 is taken. An Error message displayed asking the user to select two records.
Pass/fail	Pass
Criticality	Medium
Use case	UC_7
Test (or ID)	Analysis with more or less than two records
Description	An analysis button pressed with one or more than two records selected
Tester	George
Result	The exceptional flow for UC_7 is taken. An Error message displayed asking the user to select two records.
Pass/fail	Pass
Criticality	Medium
Use case	UC_7
Test (or ID)	Analysis with two records
Description	An analysis button pressed with two records selected
Tester	George
Result	A pop up is displayed to the user with the analysis data
Pass/fail	Pass
Criticality	High
Use case	UC_7
Test (or ID)	Add empty record
Description	Add record button pressed, record is added while all of the fields are empty.
Tester	George
Result	Records are not added and the required fields are highlighted in red.
Pass/fail	Pass

Criticality	High
Use case	UC_12
Test (or ID)	Add a valid record
Description	User clicks “Add Record” and enters the valid details for a record
Tester	George
Result	The record is added to the table
Pass/fail	Pass
Criticality	High
Use case	UC_12
Test (or ID)	Add an invalid record
Description	User clicks “Add Record” and enters invalid data into any of the fields
Tester	George
Result	Records are not added and the incorrect fields are highlighted in red.
Pass/fail	Pass
Criticality	High
Use case	UC_12
Test (or ID)	Add record whose fields to not match
Description	User clicks “Add Record” and enters a crime description, FBICD and IUCR that do not match.
Tester	George
Result	A pop up error message tells the user that the crime description, FBICD and IUCR must match. The record is not added.
Pass/fail	Pass
Criticality	Medium
Use case	UC_12
Test (or ID)	Add record with incorrect latitude or longitude
Description	User clicks “Add Record” and enters valid data except for the longitude or latitude where they enter a value greater than 90 or less than -90
Tester	George
Result	A pop up error message tells the longitude or latitude is not within the accepted range.
Pass/fail	Pass
Criticality	Medium
Use case	UC_12
Test (or ID)	User edits record and submits empty data
Description	Edit record button pressed, all the fields in the edit record window are made empty.

Tester	George
Result	Records are not added and the required fields are highlighted in red.
Pass/fail	Pass
Criticality	High
Use case	UC_3
Test (or ID)	Edits a record correctly
Description	User presses “Edit Record” and enters the valid information into the fields then presses “Save Record”.
Tester	George
Result	The record is changed in the table to reflect the users input
Pass/fail	Pass
Criticality	High
Use case	UC_3
Test (or ID)	Edits a record with invalid information
Description	User presses “Edit Record” and enters the valid information into the fields then presses “Save Record”.
Tester	George
Result	Record is not changed and the incorrect fields are highlighted in red.
Pass/fail	Pass
Criticality	High
Use case	UC_3
Test (or ID)	Edit a record with fields that do not match
Description	User clicks “Edit Record”, enters a crime description, FBICD and IUCR that do not match then clicks “Save record”
Tester	George
Result	A pop up error message tells the user that the crime description, FBICD and IUCR must match. The record is not added.
Pass/fail	Pass
Criticality	Medium
Use case	UC_3
Test (or ID)	Delete records
Description	User selects records and clicks “Delete Selected Row(s)”
Tester	George
Result	The rows are deleted from the table
Pass/fail	Pass
Criticality	High
Use case	UC_13

Test (or ID)	Edits a record with incorrect latitude or longitude
Description	User clicks “Edit Record” and enters valid data except for the longitude or latitude where they enter a value greater than 90 or less than -90
Tester	George
Result	A pop up error message tells the longitude or latitude is not within the accepted range.
Pass/fail	Pass
Criticality	Medium
Use case	UC_3
Test (or ID)	User adds a record with the existing case number
Description	User clicks Add Record and enters a record with existing case number
Tester	Sofonias
Result	A pop up error message tells the user that that case number is already used in the database. Case numbers must be unique.
Pass/fail	Pass
Criticality	High
Use case	UC_12
Test (or ID)	Table updates when filter is applied
Description	User inputs their parameters into the filter sidebar and clicks apply
Tester	Daniel
Result	Only records that match the filters specified by the user are displayed
Pass/fail	Pass
Criticality	High
Use case	UC_11
Test (or ID)	Marker shows up in edit/add window when valid coordinates entered
Description	When the user has valid coordinates entered in the text boxes in the edit or add window, the crime should be shown on the map as a marker
Tester	Bede
Result	The marker isn’t shown when the text boxes are filled incorrectly but it is shown when they are
Pass/fail	Pass
Criticality	Medium
Use case	UC_3, UC_12, UC_5
Test (or ID)	Marker in add/edit window has an infowindow when the record has a case number
Description	When adding/editing a record, if there is a case number in the case number field, the marker on the map should have an infowindow that displays the case number and some more information if those fields have data too

Tester	Bede
Result	The marker has an infowindow as expected
Pass/fail	Pass
Criticality	Low
Use case	UC_3, UC_12, UC_5
Test (or ID)	Heatmap is shown when the user selects the heatmap radio button
Description	When the user has data in the table, when they go into the map tab and select the heatmap radio button, the crimes should display as a heatmap
Tester	Bede
Result	The crimes are displayed as a heatmap as expected
Pass/fail	Pass
Criticality	High
Use case	UC_5
Test (or ID)	Heatmap density changes when the user moves the heatmap density slider
Description	When the heatmap is shown, when the user moves the heatmap density slider, the density of the heatmap should change accordingly
Tester	Bede
Result	The heapmap density increases and decreases as expected
Pass/fail	Pass
Criticality	Medium
Use case	UC_5
Test (or ID)	Crimes are shown as markers when the user selects the marker radio button
Description	When the user has data in the table, when they go into the map tab and select the marker radio button, the crimes should display as markers
Tester	Bede
Result	The crimes are displayed as a markers as expected
Pass/fail	Pass
Criticality	High
Use case	UC_5
Test (or ID)	When records are added or edited, the table is updated in place
Description	When the user clicks save record when adding or editing a record, the table should be updated in place, rather than refreshing the whole thing
Tester	Bede
Result	The table is updated in place and the new/updated record is shown correctly
Pass/fail	Pass
Criticality	Medium
Use case	UC_3, UC_12

Test (or ID)	Deleting using delete and backspace keys
Description	The Delete and Backspace keys should function the same as the delete button in the program.
Tester	Bede
Result	The user is prompted to confirm the deletion of the selected records, as expected
Pass/fail	Pass
Criticality	Low
Use case	UC_13
Test (or ID)	Map functionality of comparing two records
Description	When the user clicks the compare two records button, the two selected records should be displayed as markers on a map in the popup window, with a line between them
Tester	Bede
Result	The two records are shown as markers with a line between them, as expected
Pass/fail	Pass
Criticality	High
Use case	UC_7
Test (or ID)	Mapping blocks in analysis tab
Description	When the user right clicks on a block and selects the view on map option, the crimes that occurred in that block should be displayed on the map in the popup window
Tester	Bede
Result	The crimes are displayed on the map as expected. Some are in weird places due to incorrect lat/long data
Pass/fail	Pass
Criticality	Medium
Use case	UC_5, UC_8
Test (or ID)	Pie charts in analysis tab
Description	When the user clicks the visualise button above a table in the analysis tab, the data displayed in the table should be shown in a pie chart form
Tester	Bede
Result	The data is shown as a pie chart, as expected
Pass/fail	Pass
Criticality	Low - already shown in the table
Use case	UC_6, UC_8
Test (or ID)	Top crimes in block in analysis tab
Description	When the user right clicks on a block and selects the show details option, the

	top crimes in that block should be shown in a table in the popup
Tester	Bede
Result	The top crimes in the block are correctly shown in a table in the popup
Pass/fail	Pass
Criticality	Medium
Use case	UC_8
Test (or ID)	Display analysis pie chart
Description	When the user is in the analysis tab and clicks on the “Visualise” button, a popup window is shown with a pie chart of the rows in that analysis table.
Tester	Jonathan
Result	A popup window is shown with a pie chart of the rows in that analysis table. The window must be closed to use the rest of the application.
Pass/fail	Pass
Criticality	Medium
Use case	UC_14

7.2 Quality Testing

Test (or ID)	Importing data
Description	Importing data should take less than 5 seconds for it to appear in the table
Tester	Bede
Result	Testing with the 200k record file on the lab computers, the data takes approximately 2 seconds to appear in the table
Pass/fail	Pass
Criticality	Medium
Use case	UC_1
Test (or ID)	Refreshing table view
Description	Refreshing the table view should take less than 5 seconds
Tester	Bede
Result	Testing with the 200k record file on the lab computers, the table takes approximately 1 second to refresh
Pass/fail	Pass
Criticality	Medium
Use case	UC_1
Test (or ID)	Sorting table view

Description	Sorting the table view should take less than 5 seconds
Tester	Daniel
Result	Testing with the 200k record file on the lab computers, the data takes approximately 3 seconds to be sorted in the table
Pass/fail	Pass
Criticality	Medium
Use case	UC_11
Test (or ID)	Map displays markers without graphical errors
Description	The map shouldn't have any graphical errors when displaying markers
Tester	Daniel
Result	The app limits the maximum number of markers to prevent too many pins being shown and graphical errors occuring.
Pass/fail	Pass
Criticality	High
Use case	UC_6
Test (or ID)	Clicking on record updates lat and long filter
Description	Clicking on a record should cause its latitude and longitude to be put into the lat and long field for the filter without a noticeable delay
Tester	Daniel
Result	The data appears to be instantly put into the text fields
Pass/fail	Pass
Criticality	Low
Use case	UC_4

7.3 Discussion

7.3.1 JUnit Tests

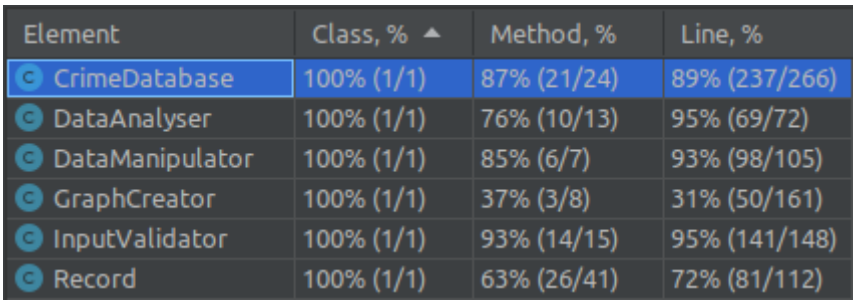
Element	Class, %	Method, %	Line, %
data	100% (6/6)	74% (80/108)	78% (676/864)
frequencyComparator	100% (3/3)	87% (7/8)	81% (18/22)
gui	0% (0/13)	0% (0/130)	0% (0/1205)
importExport	100% (2/2)	100% (2/2)	100% (28/28)

Figure 10 - Overall test coverage

Our JUnit tests covered all of the classes in the data, frequencyComparator, and importExport packages, with coverage around 80% and higher in each of them. The

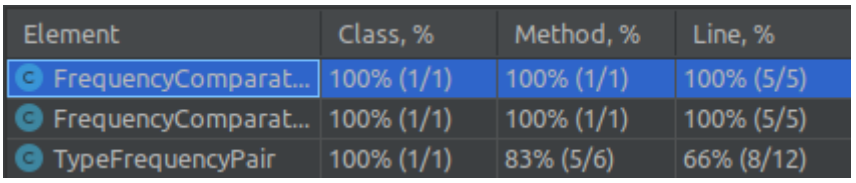
importExport class had 100% line coverage, while the frequencyComparator and data packages only had around 80% coverage.

Our JUnit tests were designed to cover methods of backend classes that were not getters or setters, and to cover different branches of these methods as well. We did this by testing with both valid and invalid inputs (usually records) to check that the correct response was received. Therefore, because the Record class and TypeFrequencyPair class both largely consist of getters and setters, line coverage is only around 80% in the frequencyComparator and data packages.



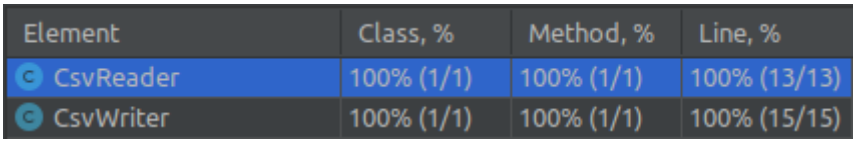
Element	Class, %	Method, %	Line, %
CrimeDatabase	100% (1/1)	87% (21/24)	89% (237/266)
DataAnalyser	100% (1/1)	76% (10/13)	95% (69/72)
DataManipulator	100% (1/1)	85% (6/7)	93% (98/105)
GraphCreator	100% (1/1)	37% (3/8)	31% (50/161)
InputValidator	100% (1/1)	93% (14/15)	95% (141/148)
Record	100% (1/1)	63% (26/41)	72% (81/112)

Figure 11 - Data package test coverage



Element	Class, %	Method, %	Line, %
FrequencyComparat...	100% (1/1)	100% (1/1)	100% (5/5)
FrequencyComparat...	100% (1/1)	100% (1/1)	100% (5/5)
TypeFrequencyPair	100% (1/1)	83% (5/6)	66% (8/12)

Figure 12 - FrequencyComparator package test coverage



Element	Class, %	Method, %	Line, %
CsvReader	100% (1/1)	100% (1/1)	100% (13/13)
CsvWriter	100% (1/1)	100% (1/1)	100% (15/15)

Figure 13 - ImportExport package test coverage

The GraphCreator class in the data package only has 31% line coverage due to it using JavaFX objects to create graphs, and therefore, methods using those are far easier to test manually by creating graphs and checking the output compared to our manual calculations of the results.

JUnit tests were not able to cover controllers in the gui package; these were instead covered by our manual testing. These controller classes make heavy use of the classes (and static methods) contained in the backend package, which were tested, wherever possible; the controllers deal with the GUI side of the program per the MVC architecture that Insight was designed with.

7.3.2 Cucumber Acceptance Tests

We also tested that our program was functioning correctly using Cucumber acceptance tests. These consisted of 12 tests, with 4 targeting the DataAnalysis, and 8 targeting the CrimeDatabase class.

The purpose of these tests was to ensure that the program performs the correct operations and returns what the stakeholders want to see.

The database acceptance tests tested filtering, editing, adding and deleting records in the database. The analysis acceptance tests tested comparing records, as well as getting crime types and blocks in descending frequency.

7.3.3 Manual Functional & Quality Tests

The manual functional tests that we conducted were based on our use cases, covering the main flow (“blue sky” scenario), alternative flows, and any exceptional flows. This included conducting boundary value analysis to uncover bugs. Our tests were mainly focused on ensuring that the product worked under the “blue sky” scenario (QR_2) and any common exceptional flows - e.g. misformatted records with invalid fields, because from our key driver analysis we saw that our potential users care most about the app working in these basic scenarios, rather than being able to handle a great deal of unlikely inputs or be completely free of random, uncommon bugs.

Element	Class, %	Method, %	Line, %
AnalysisPopupController	100% (1/1)	100% (6/6)	90% (29/32)
AnalysisTabController	100% (1/1)	95% (19/20)	86% (149/172)
BrowserTabController	100% (1/1)	85% (6/7)	88% (32/36)
CrimeDetailsController	100% (1/1)	100% (2/2)	100% (12/12)
CrimesPieChartController	100% (1/1)	100% (1/1)	100% (23/23)
EditRecordWindowController	100% (1/1)	100% (12/12)	92% (142/153)
GraphTabController	100% (1/1)	100% (5/5)	100% (74/74)
Main	100% (1/1)	100% (1/1)	100% (2/2)
MainController	100% (1/1)	100% (31/31)	87% (402/458)
MainGUI	100% (1/1)	100% (2/2)	100% (13/13)
MapTabController	100% (1/1)	92% (12/13)	93% (46/49)
PopupWindow	100% (1/1)	100% (5/5)	100% (50/50)
TableTabController	100% (1/1)	92% (23/25)	79% (127/159)

Figure 14 - gui package manual test coverage

Our manual tests covered all aspects of the gui, as shown in figure 14 above. Overall, our manual tests gave a line coverage of 89% in the gui package, with the 11% not covered consisting mostly of exception catch blocks, which we couldn’t force to occur while testing.

During manual testing the program failed one test case. This was the program freezing when the user presses Ctrl + A with the 200k record set loaded. We couldn’t find a fix for this that didn’t involve disabling all selection, which is needed for the function of our program, or completely redesigning the table to use pages, which was impractical due to time constraints. Furthermore, this error is unlikely to occur due to selecting all records in the table not being useful to the function of the program. Therefore, we have decided to leave this error in the program.

We also had several friends and family members use the program, giving them simple tasks to complete without any instruction on the functionality of the program. Examples of such tasks are:

- Create a graph of all crimes between the 12th of October 2020 and the 12th of December 2020
- Map all records within 5 kilometers of (41.65, -87.65)
- Search for news articles about “crime in Chicago”
- Import this csv file into the program

Some of the tasks were harder than others, so there was some delay while they figured out how to use the program, however, for the most part, they were all able to complete their tasks in a quick, efficient manner. We also received numerous compliments about how easy the program was to use. These users weren’t aware of some of the usability features we had implemented into the app (pressing DELETE/BACKSPACE to delete records, double clicking to edit), and so didn’t use them. Originally, we thought that these features were part of the basic usability of the app, but upon conducting quality tests with inexperienced users, we found that users didn’t try to see what happened if they did these things that we assumed were intuitive. Instead, these features are likely to provide more experienced users with the ability to use the app faster, and as such we mentioned these additional features in the user manual for this purpose.

Our manual quality tests covered basic usability of the program, such as loading speed and responsiveness. When tested in the target environment of the UC COSC lab computers, the program was able to exceed all loading time and responsiveness targets.

8. Current Product Version

Use Cases and Features in this Release

The main features included in this release of Insight are as follows:

- **Importing and exporting of crime data (UC_1, UC_2):** The user can import data from a .csv file provided it is in the accepted format. Invalid rows in a file will not be imported, but valid ones will. When exporting, the currently loaded data in the app will be saved to a .csv file in the same format that the program recognises, which can then be used again.
- **Storing of imported data in a database, with support for multiple databases (UC_9):** When the user imports data, it is stored in a database file. This allows for SQL queries to be run on the data, which meant that we were able to implement efficient filtering, adding, editing, and deleting data features. The user can switch between different database files in the app.
- **Viewing of data in a table, with support for simplified and extended viewing and filtering (UC_11, UC_4):** The main display of the app shows all loaded records in a table. The user can choose which columns are displayed using checkboxes located below the table. The user can also sort the loaded data by clicking on the column headers. The user can also implement filters for the data, searching by case number, date range, ward, beat, crime type, location, or the radius from a set of specified coordinates.
- **Support for adding, editing, and deleting crime records (UC_3, UC_12, UC_13):** The user can use the buttons located below the table to add new records, edit an existing record, or delete existing record(s). Adding and editing records requires that new / edited records are in a valid format, and provides detailed validation feedback to the user - this includes error messages that specify the required format as well as highlights on the specific fields that are invalid.
- **Four different types of graphs about the displayed data (UC_6):** The user is able to view four graphs, which are line graphs over time: the total number of crimes, crimes per ward, crimes per beat, and crime types over time. The user can select as many or few beats/wards/types of crime as they like, and the graphs are quickly displayed along with a legend for the user.
- **Data mapping functionality (UC_5):** The user can switch to the mapping tab of insight, where all of their currently loaded data will be displayed as data points on the map, and additional information about each data point will be displayed when they are clicked.
- **Data analysis features (UC_7, UC_8):** The user can, with two records selected, view the time difference or spatial distance between the two records. Additionally, they can view tables showing the most & least frequent crime types, as well as the blocks with the highest and lowest rate of crimes.
- **Internet search features (UC_10):** Insight allows users to search the web for additional crime data - displaying results from government or news websites - right from within the app. The browser has back, forward, and home buttons to allow for easier navigation.

This release of Insight includes all of the major planned features defined in the business case as part of the MVP, in at least basic forms.

Notable Features

Insight includes several notable features that the development team are proud of:

- **Record validation and feedback:** When the user is adding or editing records, the values in all fields of the record are checked for validity and the user is given field-by-field feedback (as a larger, red outline) along with a helpful error message if any fields are invalid. Key to this validation is that it ensures that the crime type matches the FBICD and IUCR codes, and that the primary and secondary descriptions match.
- **Mapping popups:** When the user clicks on a marker on the map tab, a message box pops up above the marks showing it's case number in bold, with the date and time, description and location description below that.
- **Radius search:** When filtering data the user can select records in a set radius around a specific record. This is done by clicking on the record in the table to auto fill its latitude and longitude then setting a radius between 0 and 10000m with the slider. This uses the same distance calculating function as the analyse distance option to find all the records within the given radius.
- **Import feedback:** When the user tries to import a csv file that has invalid data in its rows, a feedback is given saying how many rows were in valid.
- **Fast filter speed.** Because Insight is built to use a database, filters can be implemented using SQL queries, which gives very fast response times for even complex queries that the user may create.
- **Designing the program around the database:** We implemented the database at the start of development which kept the complexity down as we did not need to go back and change the implementation to include the database. We have a very large data access layer as we built it up as we progressed through development.
- **GUI layout:** Insight has a simple, modern, and intuitive gui. The development team has received positive feedback from numerous people about the way it looks. The user is not able to accrue a stash of popup windows, as each must be closed to continue using the application, thus reducing screen clutter.
- **Auto-suggestions for crime types:** When adding or editing crime records, if the user starts typing in the primary or secondary description fields, Insight will automatically bring up a list of crime types that match the string the user has currently placed in the text field - like a form of predictive text. If the user selects an option from the list, it automatically fills out the FBICD and IUCR fields so that the user doesn't have to remember the codes or look them up in a table. This makes adding or editing records much easier for the user.
- **Mapping crimes:** The user is able to map crimes in the form of both markers and a heatmap. These modes are easy to switch between using the radio buttons in the map tab, and the density of the heatmap can easily be adjusted using the slider.

Features not in this Release

Originally, Insight was envisaged as being able to work with other types of data (non-crime), e.g. wealth and income data, employment rates, etc. However, in implementing the basic functionality of the app, the development team realised that this was not going to be possible to include within the timeframe we have for the app, and is also of questionable value in the app in the first place. The amount of different types of data and the geographic regions which they represent make it difficult to include them in such an app - the data would be in so many different formats and represent many complicated geographic regions that representing it in an app alongside crime data is very difficult. Instead, Insight can be used to support this analysis of how crime relates to other societal statistics by providing analysis and visualisation features for crime data.

9. Risk assessment

Table 7 shows the project's risk assessment. This risk assessment was carried out by brainstorming potential risks in four main areas: risks arising from the team itself; business risks involving users or testing; technical risks to do with the software and systems used to create Insight; and external risks that come from actors outside of the team (and which are generally outside of the team's control). The impact (column **I**) and likelihood (column **L**) of each risk arising go from minimal/low (1) to severe/high (3). Risks are sorted in descending order based on their impact multiplied by their likelihood, giving the overall risk (column **OR**). Severe risks are those that are very difficult or impossible to prevent or manage should they occur, moderate risks can be prevented or managed with some effort, and minimal risks carry little consequence. Highly likely risks are nearly guaranteed to occur at some point during the project, somewhat likely risks may occur, and low likelihood risks are very unlikely to occur. The responsibility for each risk generally means who should try to resolve it, which usually falls with the team, even if the risk is the result of an individual.

ID	Description	I	L	OR	Responsibility	Consequences	Prevention
R_01	Team member(s) injured or sick.	2	3	6	Whole team.	That team member may be unable to attend meetings or classes and may have a reduced or zero work output depending on the severity of their ailment.	This is hard to prevent beyond observing good hygiene practices and staying safe. To mitigate the impacts, we can take notes during lectures/labs to let the affected person know what they missed, have them attend meetings via Zoom, only assign them tasks that they will be able to complete while sick or increase the workload of the rest of the team.
R_02	Team member's code is unreadable by others.	2	3	6	Team member who wrote the code.	The impact depends on how vital the code is to the software, but generally, this means that other team members will not be able to improve, maintain, fix, or otherwise usefully edit or contribute to the unreadable code, which will slow down the team's progress on that part of the software.	The team should draft and agree on a set of coding standards that should be followed, and then follow them. Team members who are having difficulty with coding should ask for help so that they do not end up creating "spaghetti code". When team members are finished with a piece of code, it should be reviewed by another team member. A formal system for tracking this through Git should be established so that the team knows what code has and has not been reviewed.

R_03	The project is poorly scoped.	2	2	4	Whole team.	Team members could spend time researching and implementing extra features and then find that there is not enough time to do so, resulting in less work being done towards the MVP and an overall worse quality product that may lack even more basic features. This in turn could render the software unusable if critical features are left out, or low quality if simple improvements cannot be made.	The product should be scoped realistically, by first starting with the basic requirements list and an MVP, before any additional features are planned. While it may be useful to research additional features prior to implementing them, this should be done reasonably, ie. Nobody should be researching cloud storage before a graphing functionality is implemented into the product.
R_04	Team meetings are not productive.	2	2	4	Whole team.	If team meetings are unproductive, it is likely that team members will not have a clear idea of what they should be doing or the current state of the project, meaning that work is much slower and less focussed. There may be problems integrating the work of different team members or multiple team members may do overlapping work.	Team meetings should be planned out in advance with a clear agenda, including a recap of what has been accomplished since the last meeting and the setting out of tasks to complete before the following meeting. These tasks should be followed up on at the next meeting so that any problems that arise can be dealt with quickly.
R_05	Poor, unfair, or unclear task allocation.	2	3	6	Whole team.	If team members are not clear on what they need to do, then they will have lower	The allocation of tasks should be fair, so that all team members have enough work to do before the next meeting and review / allocation of tasks. Furthermore,

						productivity because their tasks are unclear or they do not have enough tasks assigned before the next meeting. This will slow the project down overall.	tasks should be SMART - specific, measurable, achievable, relevant, and time-bound. This way, team members will know what they need to do and when it needs to be completed, and tasks are assigned to fairly distribute the workload of the project.
R_06	Team conflict or disagreement.	2	2	4	Whole team.	The team's work will be slowed, depending on what the conflict is about. Team members involved in the conflict may not deliver work on time or to an acceptable standard.	The team should ensure that tasks are delegated fairly and with clear expectations so that each team member contributes their fair share to the project and there is less likely to be conflict. Additionally, team members should regularly and constructively review the work of others to make sure it is of good quality and provide feedback or comments.
R_07	The team is unfamiliar with APIs or libraries; lack of coding knowledge.	2	2	4	Whole team.	If team members are not able to work with external APIs and libraries, or lack some coding knowledge, they will not be able to make as much progress on their tasks and the project may experience a slowdown.	If possible, the necessary APIs, libraries, and coding knowledge should be ascertained before work starts on those sections of the project, so that the team can become more familiar with those areas at a comfortable pace. Furthermore, team members should be confident to get help, be it online, from another team member, or a tutor.
R_08	APIs, libraries, or other dependencies have bugs.	1	3		Whole team.	As these bugs likely will not be immediately apparent, they may at worst force the team to use a different dependency for the project, potentially undoing a lot of work. In other cases, the bug may be relatively minor and be able to be worked around.	Any dependencies that are used in the project should ideally be widely-used and popular, as these are less likely to have bugs and more likely to have bugs be reported and workarounds devised than less popular dependencies.

R_09	Internet or power outage.	3	1	3	Whole team.	If the internet is unavailable, team members will be essentially unable to share their work or see the work of others. If there is no power, no work can be done at all.	To lessen the severity of an internet outage, team members should if possible maintain an offline backup of the Google Drive folder, regularly commit, push and pull to the Git repository, and generally share their work and download the work of others quickly. Furthermore, team members should each have at least one task assigned to them that they will be able to work on by themselves on the Trello board, even if multiple people are assigned to one task.
R_10	There is a miscommunication with course staff.	3	1	3	Whole team and/or staff member (it depends).	Any miscommunication will likely result in the team not completing work in the manner that the course staff are expecting, or the team missing important announcements, deadlines, or activities that they were unaware of.	Any uncertainties should be followed up with course staff promptly so that the team is clear about what they are expected to do.
R_11	Team member leaves.	3	1	3	Whole team.	Approximately 20% of the team's work output will disappear, and the remaining team members will have to increase their workload to compensate.	Team members can be asked to give a heads-up if they may leave the team or course for any reason so that the rest of the team can prepare. In addition, team members should be encouraged to stay with the team by maintaining a positive work environment that encourages team members to stay and makes them feel valued.
R_12	API service experiences an outage.	2	1	2	API provider.	While some development of the project may be able to progress, this risk would have the most consequences post-deployment, as the software would no longer	The software should include useful error messages if some API is not able to be reached via the internet. Alternatively, the software should include the necessary API files to work offline, eg. downloading maps of certain cities to store offline. If possible, the software could be designed to work without certain features that require external APIs.

						function without important APIs like those used for mapping, eg. Google Maps.	
R_13	Team member does not complete a task(s) on time.	1	2	2	Team member who did not complete the task.	The consequences depend on the importance of the assigned task and the time at which the team noticed it was not completed. The general consequences are that other team members have to complete the task instead, and the team risks either missing deadlines or handing in subpar work.	Team members should have due dates on their tasks, at which point they can be reviewed by another team member. This means that the team will know if a task has not been completed. Tasks should also be clearly assigned on the Trello board so that team members understand what they have to do and when to have it done. If possible, tasks should be assigned to the team members who prefer them, so that there is a higher likelihood that they will complete the task. A yellow-card policy should be developed and employed to discourage not completing work when it is due.
R_14	There are lockdowns or other restrictions due to an outbreak of the coronavirus.	2	3		Whole team.	The consequences will likely be that in-person work will no longer be possible because team members will be in different bubbles. This can make work more difficult, reduce meeting productivity, and reduce the effectiveness of pair programming.	The team should be familiar with using online tools and have practiced using them before such an outbreak occurs. For example, everyone should check that they can attend a meeting on Zoom. Furthermore, the team should regularly use online organising tools like Trello and Slack so that the transition to remote work goes smoothly and does not require the team's method of work to change too drastically.

Table 7 - Risk Assessment

10. Project plan

Table 8 shows the project plan. We plan to create the MVP and complete Insight's core features before we start work on adding additional features like progress bars, heatmaps, or creating more graph types.

Date formats are day/month, eg. 2/8 means August 2nd.

Milestone	Date	Description	Related Tasks
M_0.1	Monday 2nd August.	Set 1 of design document tasks drafted.	Drafted by Monday 1st of August: ● Business context by George and Sofonias ● Stakeholders by Jonathan ● Acceptance tests by Bede ● GUI prototypes by Daniel ● A risk assessment by Jonathan
M_0.2	11:59 PM Sunday 8th August.	Set 2 of design document tasks drafted, and tasks from M_0.1 reviewed.	Drafted by Thursday 5th of August: ● Requirements by Sofonias and Bede ● Use cases by Daniel ● Deployment model by Sofonias ● UML class diagram by George ● Project plan by All Reviewed by all members by Sunday 8th of August: ● Business context ● Stakeholders ● Acceptance tests ● GUI prototypes The design document should be frozen to all major changes from 11:59 PM on Sunday 8th August, pending the final review.
M_0.3	3:00 PM Friday 6th August	Checklist finished and signed off	● Set up all required communication channels by Bede ● Set up git, maven, JUnit and CLI by Bede and George
M_0.4	3:00 PM Monday 9th August	Final document review performed.	The document is checked for (by all members): ● Grammar and spelling; ● Appropriately formal language; ● Images and tables are numbered and have descriptions; ● Other software, data, and research is cited and referenced; and ● References are included in the end section of the document.
M_1	5:00 PM Monday	Design document &	Checklist completed by 8/8 11:59 PM. Business and system context fully drafted by 2/8.

	9th August	project checklist completed for the first submission deadline.	Stakeholders identified and drafted by 2/8. Requirements drafted by 7/8. Use cases drafted by 7/8. Acceptance tests drafted by 2/8. GUI prototypes drafted by 2/8. Deployment model finalised by 8/8 11:59 PM. UML class diagram finalised by 7/8. A risk assessment performed and drafted by 2/8. Project plan completed by 8/8 11:59 PM. The final draft of the design document completed by 8/8 11:59 PM. Final review of design document conducted by 9/8 2:00 PM. Final, final review of design document conducted during team meeting on Monday 9th August 2:00-3:00 PM. The document can then be submitted on Monday 9th August between 2:00 and 5:00 PM.
M_1.1	Monday 23rd of August	The program is able to import given crime data and display it in a table	● The given crime data csv file is able to be imported into the program and viewed in a tabular format ● JUnit tests written for this ● Verified that data imports correctly
M_1.2	Friday 3rd of September	The program can perform basic data analysis on given crime data	● Program can calculate time between two selected crimes ● Program can calculate distance between two selected crimes ● JUnit tests written for these analyses
M_1.3	Monday 6th of September	The program can modify and export data	● Data can be updated using the GUI ● Data can be added/deleted using the GUI ● Data can be exported to a new csv file using the GUI ● JUnit tests written for this
M_1.4	Monday 13th of September	Crime locations can be visualised on a map	● Able to view crimes of map using google api ● Able to filter which crimes are displayed ● Option to view extended data for each crime on map
M_1.5	Friday 17th of September	Complete GUI for all use cases and functional requirements	● Able to do everything listed in functional requirements through the user interface

M_1.6	Monday 20th of September	Verify acceptance tests	• Run acceptance tests
M_1.7	Monday 20th of September	Verify quality requirements	• Test quality requirements
M_2	Monday 27th of September	Implementation and testing completed	The program can import crime data The program can export crime data The program can display crime data in a table The program can view crime data on a map The program can display crime data in a graph The program can filter data based on user selection The program allows the user to edit data JUnit tests for all classes are implemented
M_2.1	Friday 8th of October	Have final draft of project	All code conforms to code standards All code is commented and has a javadoc written
M_2.2	Sunday 10th of October	Final code check completed	Everyone reads over the code for the application to check it is commented correctly and conforms to code standards
M_3	Monday 11th of September	Project and Demo handed in	Plan out how the demo will work Hand in final deliverable October 10th Implement extra functionality

Table 8 - Project Timeline

Figure 15 shows the dates of assignments for group members, these dates will have to be worked around to ensure deadlines are still met.

13 AUG, FRI	3 – 4pm	COSC265 Part 1 of Assignment 1 Due - Bede, Jonathan, George, Daniel, Sofonias
14 AUG, SAT	All day	Bede - Away at tournament
15 AUG, SUN	All day	Bede - Away at tournament
22 AUG, SUN	11:59am	COSC264 Assignment Due - Bede, Jonathan, Daniel, Sofonias
27 AUG, FRI	3 – 4pm	COSC265 Assignment 1 Due - Bede, Jonathan, George, Daniel, Sofonias
13 SEPT, MON	7 – 9pm	ENCE260 Lab Test - Jonathan, Daniel, Sofonias
20 SEPT, MON	7 – 8:30pm	COSC264 Mid Term Test - Bede, Jonathan, Daniel, Sofonias
21 SEPT, TUE	3 – 4pm	COSC367 Mid Semester Test - George
22 SEPT, WED	5pm	COSC368 Assignment Part 1 Due - Bede, George
7 OCT, THU	3 – 4pm	COSC265 Lab Test - Bede, Jonathan, George, Daniel, Sofonias
15 OCT, FRI	7 – 8pm	COSC264 Lab Test - Bede, Jonathan, Daniel, Sofonias
20 OCT, WED	5pm	COSC368 Assignment Part 2 Due - Bede, George

Figure 15 - Assessment Calendar

11. Lessons Learnt

Sofonias Tekele Tesfaye:

- Role of teamwork in software development: One of the biggest lessons that I have learned in SENG202 is the role that teamwork has in a software development environment. Similar to the SENG202 project, esources and time are among the top constraints of many software development projects. In such a constrained environment it is extremely difficult to come up with the desired outcomes without having the knowledge, structure, resources and platform that will allow you to successfully implement effective teamwork. Hence I have learned that personally and collectively a good amount of emphasis should be put towards creating an environment where teams can thrive.

- Documentation and related software development practices: SENG202 project has given me a glimpse of what software projects in the real world might look like. One of the insights I have gained is the amount of people and code a software development project may involve throughout its life cycle. With such a large amount of code and people it would literally be impossible to build software without a clear method for proper documentation and other related software development practices. Hence I have learned once again that a good amount of effort should be put towards observing the standards of documentation practices in a software development environment.

- Manual and Auto Testing: Software code grows and evolves with time. Developers may add a code every now and then and the newly added code may end up having an undesired impact on existing behaviour of a code. This is where testing comes. Throughout this project I have been able to appreciate the test driven software development practice and also the existing platforms for conducting manual and auto testing on a software code.

Bede Skinner-Vennell

- Working in a team: I think the most important lesson I've learnt during SENG202 is how to work effectively in a team. There are numerous aspects to this that I've never had to do to this level before. The main challenge was learning how to communicate effectively. Our whole team works on different schedules, so effective communication about when we're going to complete our tasks was essential. The time pressure of this assignment made this even more important, as everything had to be completed on time in order for other team members to be able to do their work. Overall, I think that I've learnt a lot about working in a team and I'm looking forward to taking my experiences from this course and applying them to other courses in the future.

- Public speaking/presentation skills: During the weekly stand ups I got the chance to present my work to groups of people, something that I've struggled with in the past. Through feedback from the other students in the course, I was able to pinpoint some things to work on and actively worked towards improving these aspects of my

presentation, such as maintaining eye contact and talking to the listeners rather than the projector screen.

- Using version control with multiple people: I've never had to use version control with multiple people working on the project at the same time. During this project, I learnt how to use branches to have multiple people working on different features at the same time without breaking things that other people are working on. I think that this skill is going to be useful in future projects, and at my summer internship.

Daniel Pallesen:

- Effectively working in a team: One of the main challenges in the project has been working effectively in a team of five people. I have learnt how important it is to have good communication, this is very important in order to ensure all team members have enough work to do that they are significantly contributing but not too much that they are overwhelmed. The use of tools such as Slack, Trello and Zoom have all been very helpful in facilitating this.

- Presentation preparation and delivery: The weekly stand-ups significantly improved my ability to present my work to an audience. I learnt how to effectively prepare a presentation while avoiding just reading off a script. The frequency of the standups also gave an opportunity to practice using appropriate body language before the lockdown.

- Building maintainable software: Due to the length and size of the SENG202 project it gave good insight into how important keeping software maintainable is. As a team we aimed to build a maintainable program but there were still a few times where it was necessary to refactor code to make it more modular. The importance of good commit messages was also highlighted as there were a few times we had to look through commit history to find where a bug was introduced.

Jonathan Tomlinson

- Communication, the importance of: Even in a small group project such as this, the importance of each team member being on the same page and having a clear idea of what they needed to do was very important. In the earlier SENG201 project, it was somewhat possible for each person to just "do work" on the project without necessarily having a clearly defined task, however, due to the larger scale and open ended nature of this project, that does not work. I learned that it is vital for each individual team member to know what their tasks are, but also for the rest of the team to know what everyone else is doing - that way the whole team is on the same page and progress can be made faster, particularly when working on features that depend on each other. I found that having a shorter, second meeting each week where the team updated their tasks was very useful for this, and helped to boost productivity.

- Testing & Documentation: This project made me appreciate the importance of both testing and documentation. With a group of five people each working on rather in-depth features - like databases, maps, filters etc. - it's not always possible to keep up with and understand every change that someone else has made to the code, and I found that methods without javadoc or inline comments were often very hard to

understand, which meant that I couldn't design tests for them, or easily fix bugs in them either. Methods with good javadoc were much better to work with. Also, having a broad set of unit tests with good coverage (especially hooked up so that they automatically ran on each commit) made it much easier to develop code, as one could implement a feature or change, and then check that it hadn't broken the rest of the app, which is very important when working with other people - nobody wants to pull broken spaghetti code.

- The challenges of working remotely: Although I do sometimes feel nervous speaking in front of other people, I much prefer doing it in person than talking to a screen online. I found delivering standups online much harder, as I couldn't easily judge people's reactions to what I was saying, or use gestures as well as I could in person. Additionally, we found as a group that doing things online could lead to unclear communication - several times we agreed during a video call when to hold our next meeting, only for someone to either forget or not remember it being scheduled, which didn't happen with our in-person meetings. I think it is therefore useful to ensure things are *extra* clear when working online - it's easier to get agreement about a meeting time when you're all sitting at the same table, but it's much harder to tell that everyone is in agreement on a video call.

George Carr-Smith

- It is important to research how to properly implement java libraries before starting to use them. When creating the database class I did not thoroughly research how to implement the SQLite JDBC Driver java library. When I wrote the methods in the class I did not use try-with-resources statements which automatically close resources and I did not close the prepared statements and resultsets. This resulted in the program throwing Database Locked errors. Our group spent hours refactoring the code to close the database resources after use so we would no longer get the error. I was unable to refactor the database class to use try-with-resources statements as the class had grown too large by then. I have learned that I should have spent more time at the beginning of the development cycle making sure that I had the correct implementation in order to save time from refactoring later in the cycle.

- Communication in the team: Communicating through the many communication channels has allowed our team to stay organised throughout the project. Using trello to communicate tasks helped the team in remembering and prioritising tasks. Slack allowed us to ask each other questions to clarify parts of the code and allowed us to communicate any bugs that we found to the team. Communication through the biweekly meetings allowed us to show our progress and allocate tasks. Without effective channels of communication the productivity and organization in the team would have suffered.

- Using feedback is important to growing as a software engineer. Throughout the project I have used feedback to improve my skills and become a better software engineer. The most common feedback that I got and improved on was my presentation skills. Feedback from my presentations told me that I was not making enough eye contact and I was reading off a script too much. I improved my

presentation skills by actively trying to make more eye contact and memorising my notes instead of reading them.

References

Batt, S., Grealis, T., & Harmon, O. (2020). Learning Tableau: A data visualisation tool. *The Journal of Economic Education*, *51*(3-4), 317-328. https://doi.org/10.1080/00220485.2020.1804503

Eldridge, J. (2006, 3). Data visualisation tools—a perspective from the pharmaceutical industry. *World Patent Information*, *28*(1), 43-49. https://doi.org/10.1016/j.wpi.2005.10.007

LexisNexis. (2021, 7 31). *Community Crime Map*. LexisNexis. https://communitycrimemap.com/

Marqual IT Solutions Pvt. Ltd (KBV Research). (2021). *Global Data Visualization Tools Market By Tool Type, By Business Function Type, By Deployment Type, By Enterprise Size, By End User, By Regional Outlook, Industry Analysis Report and Forecast, 2021 - 2027*. Marqual IT Solutions Pvt. Ltd (KBV Research). https://www.researchandmarkets.com/reports/5345549/global-data-visualization-tools-market-by-tool

New Zealand Police. (2021, 7 31). *Crime Snapshot*. New Zealand Police. https://www.police.govt.nz/crime-snapshot

Appendix

Table 9 shows which team member worked on which section of the design document. Note that all team members contributed to reviewing, proofreading, and providing feedback on the work of others.

Section	Worked on by
Business and System Context	George and Sofonias
Stakeholders and Requirements	Jonathan (Stakeholders), George and Sofonias (Requirements)
Textual Use Cases	Daniel and Jonathan

Quality Requirements and Key Drivers	Sofonias Jonathan
Acceptance Tests	Bede
GUI Prototypes	Daniel
Deployment Model	Sofonias
UML Class Diagram	Bede
Testing Procedures	Everyone
Current Product Version	Jonathan
Risk Assessment	Jonathan
Project Plan	Everyone
Lessons Learnt	Everyone

Table 9 - Project Work Distribution