6CCS3PRJ Final Year
Data Visualization Application for Punjab Police

Final Project Report

Author: Junaid Sadiq Masood
Supervisor: Maxime Crochemere
Student ID: 1236003

April 27, 2015

Abstract

Improving public service delivery is one of the major challenges faced by governments in developing countries. However the spread of computers and technology has given us the tools to redevelop systems and tackle these problems in new and innovative ways.

This project studies the problems faced by the Punjab Police in managing and understanding their crime data, and aims to provide an application to help them collect and visualize crime reports. It will include a database component, a charting component and a mapping component to help store data and create graphics from it.

The project will provide insight into both the workings of the Punjab Police and how to create applications for the public sector.

Originality Avowal

I verify that I am the sole author of this report, except where explicitly stated to the contrary. I grant the right to King's College London to make paper and electronic copies of the submitted work for purposes of marking, plagiarism detection and archival, and to upload a copy of the work to Turnitin or another trusted plagiarism detection service. I confirm this report does not exceed 25,000 words.

Junaid Sadiq Masood

August 13, 2014

Acknowledgements

I would like to acknowledge my project supervisor Professor Maxime Crochemere, for his invaluable support and guidance throughout this project.

Contents

Section 1 - Introduction:

1.1 - What is the project about?

Computers have revolutionized the way we collect and interpret information. They have allowed us to gather large amounts of data efficiently and have also allowed us to present and subsequently understand this information in new ways. Data visualization techniques now help governments, corporations and other organizations to make more informed judgments and have helped streamline the decision making process.

Unfortunately many third world countries still lag behind when it comes to utilizing computers, particularly in the public sector. In the case of this project Pakistan's police stations have no proper system to help them maintain crime data electronically [1]. In the rare case when they do manage to store the information on a computer they do not have the sufficient knowledge or tools to help them understand the data trends and therefore draw useful conclusions from the aforementioned data. Provided adequate software, the police can better understand the crimes that are occurring in the city and thus can take informed steps to address the problems. For example the police would be able to identify crime hot-spots or streets that burglars or thieves specifically target with the aid of a good mapping tool.

This project aims to rectify this problem. The idea is to develop a piece of software which will allow the police to enter in crime report data into a database and then be given the tools to analyze and understand that information to help reduce illegal activities. A desktop application will be programmed that will not only collect and store crime reports electronically but crucially will contain data manipulation and visualization tools that will allow the police to better understand the data they collect. Data manipulation is a key component of this project and the aim is to give the user as much flexibility and options as possible to choose different frames of reference when analyzing the data. An important consideration that must be undertaken here is to make sure that whilst our data visualization process is comprehensive it should not be overcomplicated. This means that the visualization tools must be brief and to the point in order to make sure that there is no extraneous material on screen – every item should serve a communicative purpose [2].

1.2 - Aims and Objectives:

The overall aim of this project is to produce an interactive Java application which would allow police officers, particularly division heads and the city chief, to better understand crime patterns across the city of Lahore. It would also allow them to compare the crimes

rates in different localities and judge the performance of police stations. Keeping these factors in mind the software application must be able to do the following:

- Provide an efficient method to store crime reports as collected by the Lahore Police Force
- Allow for the numerical data of the stored crime reports to be reproduced as charts
- Include a geo-mapping component to tag and subsequently analyze crime incidents geographically
- Provide some form of access control so as to prevent access to confidential crime data
- Be adaptable so that the system could be implemented for other cities as well
- Be scalable so that additional criterion and information may be added if needed in the future

1.3 - Motivation:

One of the more curious sights one can expect to see in the police stations of Lahore is a map of the city in the corner with a large number of pins stuck on it. Upon further inspection

it is discovered that this arrangement is what the police force use to analyze and understand crime hotspots and risky locations. In a world where technology is progressing by leaps and bounds such a primitive arrangement is indicative of the problems public service institutions face in Pakistan. As has been mentioned already the police force of Pakistan still does not have a reliable centralized system to hold computerized records. Whilst effort are underway to remedy this [3] and implement an information management system, progress has been slow and most crime reports are still stored as hard copies. Policing in third world countries is already a tough job. Problems such as corruption, nepotism and lawlessness mean that the police are unable to serve the public to their full potential. Furthermore these problems are compounded by the frequent terrorist attacks that take place in Pakistan. According to one government report Pakistan is the most terror hit nation in the world [4] and it is the security personnel who are often the targets. Terrorists have targeted everyone from the lowest of police recruits to the police chiefs in charge of the whole force, and these attacks have been carried out on police transport vehicles, offices and even a police training academy [5]. These incidents just serve to highlight what a difficult job the police force of Lahore is faced with every day. In the current situation the police force of my hometown requires all the assistance it can get and this was the prime motivating factor behind this project. A data visualization application

combined with a computerized information management system would greatly improve the effectiveness and efficiency of the police force, and would be a welcome upgrade to the police force's current system.

1.4 Technical and Software Details

The project will be coded in the programming language Java, making use of its versatility, scalability and its vast library support. Java is a tried and tested approach for creating desktop applications, and it would allow easy shifting of platforms when needed as the code runs on the Java Virtual Machine. Whereas initially the application will be designed to run on a Windows platform, it can be easily run on other operating systems that support Java such as Linux or Mac OSX. Once the back end has been developed we can also use our application as a blueprint for an Android or a Java based web application. Thus the application could be easily modified and updated to suit the evolving needs of the Lahore police force.

Section 2 - Background and Literature Review:

2.1 - Data Visualization and the Public Sector?

2.1.1 - Introduction and overview

With the advent of the computer and the information age it has heralded, the value of data has skyrocketed and it has emerged as one of the most valuable commodities. This fact is perhaps best illustrated by the value of companies such as social networking sites which hold huge amounts of data: Facebook is now worth about $200 billion. By comparison United Airlines, “a company that holds assets such as airplanes and has licenses to lucrative things like airport facilities and transoceanic routes between the U.S. and Asia, among other places, is worth only $34 billion” [6]. Companies now make large amounts of money off the information they gather and data analytics is a hugely lucrative market. The reasons for this become clear when the we consider how useful information from analyzing and understanding data is to making important decisions.

Data visualization is a general term that describes any effort to help people understand the significance of data by placing it in a visual context. Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognized easier with data

visualization software [7]. Thus good data visualization tools can play a crucial role in the decision making process. It also improves efficiency as the data is presented in a way that can be easily analyzed, thereby reducing the time and effort involved. Whereas in the past data visualization was mostly the forte of private businesses, governments and international organizations have now recognized the value of data representation software and how it can improve public service delivery.

One example is of the United Nations working in collaboration with Microsoft in order to create real time visualizations of the Millennium Development Goals (MDGs) available at : http://www.mdgleaders.org/. Understanding the importance of data tools in assisting development the UN Secretary General Ban Ki-moon named an Independent Expert Advisory Group on the Data Revolution for Sustainable Development to provide him with inputs to shape “an ambitious and achievable vision” for a future development agenda beyond 2015 to succeed the United Nations MDGs [8]. Microsoft is also developing data tools for use by governments across the world.

The potential gains from good data visualization software are amplified when we consider developing countries. These nations are faced with a whole deluge of problems that are not

present in the more advanced countries of the world, therefore the room for improvement in public service delivery is much greater.

2.1.2 What are good data visualization practices?

Good data visualization can help represent both quantitative and qualitative data in a more informative and intuitive way. Research has shown that the human brain picks up information from pictures, graphs and charts much quicker than from plain text formats. At the same time however it is easy to get data visualization wrong and end up with poorly designed visuals which defeats the whole purpose of the exercise.

The most important decision a designer must make is to select the correct visual aid or graphic to display the data. He or she should keep in mind the type of data the graphic is supposed to represent and the goal of the visual aid whilst making the selection. The following graphic (Figure 1) not only represents the decision making process, but also exhibits the traits of a good visual aid.

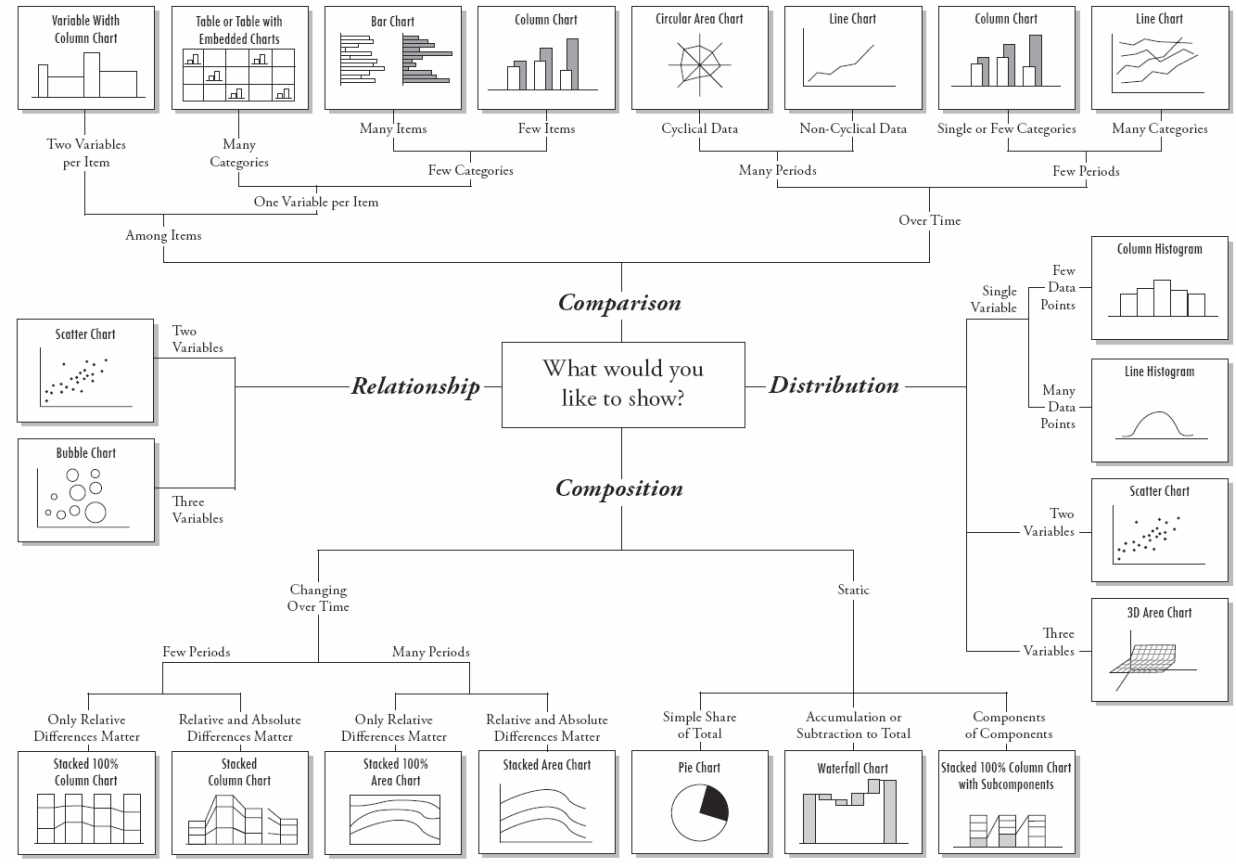


Figure 1 [10]

One of the most common mistakes that designers commit whilst designing visualisation tools is to try and display too much information. It is important to keep the visualisation concise and to the point; there should be no extraneous information on the screen [2]. Any item which does not give the user any relevant information is just taking occupying space and should be avoided.

It is also important to remember that whilst a good presentation is important, it must not take precedence over the message the graphic intends to convey. It is a case of substance

over style in this regard and the design should be minimal and subtle. It is important to avoid very bright colours, sharp contrasts, gradient backgrounds[9], three dimensional charts and the other forms of styling that might distract the viewer or make the graphic difficult to read. This also includes avoiding variety for the sake of variety; it is better to sacrifice diversity in terms of options for clarity.

It is critical to use balanced and unbiased visual aids when displaying information and to avoid distorted graphs i.e. those which misrepresent the data. Graphs may be misleading through being excessively complex or poorly constructed. Even when constructed to accurately display the characteristics of their data, graphs can be manipulated to display different conclusions [11]. For instance scaling the axis to exaggerate a change in numbers or displaying incomplete data are some common examples of dishonest visual aids. Maintaining the integrity of your graphs is an extremely important consideration during the design phase and strongly impacts the credibility of the final solution.

Dashboards are a popular method of visualizing large amounts of information and are widely used in the analysis of public sector data. Whilst designing dashboards it is important to ensure that there is no clutter or unnecessary information; all the data should fit on one screen. Dashboards are usually glanced at and therefore they should clearly show

any updates or changes that have been witnessed in our dataset [10]. Finally we should keep in mind that most of the time the user would not read each piece of information on the screen so the design should ensure that important facts and figures are highlighted on the screen and placed in prominent positions.

In order for data visualization to be effective it must put forth the information as cleanly as possible and allow the users to draw conclusions from the data. All graphics and visual aids should be analyzed carefully to ensure they serve their intended purpose and effectively convey the information they are designed to.

2.1.3 Data Visualization in combating crime

The police and other security agencies all over the globe collect a large amount and variety of data which can be used in a variety of ways. The police can use this information to map crimes and to identify locations, times and types of crimes that occur most frequently. The data can also assist the criminal justice system at large and most importantly help prevent future crimes. It can help assist courts hand out sentences and the police plan how to take on gangs or repeat offenders.

However one of the major disadvantages of the collected data is that it usually exists in textual or tabular form. The datasets are very large and complex, and given their wide scope and variety of values making good use of the information is a significant challenge. A simple and clear visualisation of this data allows the different branches of criminal justice and lawmakers to gain a better understanding of the facts on the ground and thus respond to the challenges with greater background knowledge. Crime detection and prevention can be combined with other relevant factors and data streams in order to get a clearer picture.

There are many different types of data visualization tools being used by police forces across the world, and perhaps the most common technique is crime mapping. Crime mapping involves the analysis of geographical co-ordinates of various crimes using a Geographical Information System (GIS). The co-ordinates are used to draw crime locations on a map and these in turn can be used to identify crime hot-spots and other geographical patterns. These maps can also be integrated with other datasets such as socio-economic indicators, weather, traffic and geospatial maps [12] in order to give a comprehensive overview of the causes and links behind criminal activities.

Traditional charts such as histograms and pie-charts, whilst they may be a bit old fashioned, still have a role to play as well. They allow for a numerical analysis of the data

and make comparisons between different police divisions much simpler. This also makes it easier for a police department to allocate resources and determine the effectiveness of measures being taken by individual police stations. The greater the range of data manipulation that is provided by the software the more useful conclusions can be drawn from it. Comparisons can be drawn between crime rates, frequency, timings, victim demographics and so on.

Keeping in mind all the possible benefits it is not surprising to learn that governments and police forces across the globe are investing into data manipulation and analysis tools in order to aid their day to day activities. If a usable system is implemented properly it allows for greater risk identification and therefore allows for a stronger defense to be built. Crime prevention and planning not only helps save the governments millions of dollars in the long run, but more importantly secures the lives of citizens and result in a safer society.

2.1.4 - Efforts to modernise police systems in developing countries

Given all the benefits of bringing police systems up to date, it is not surprising to learn that such efforts are underway in various third world countries. It is important to note here that policing in third world countries is a hugely challenging task as the state's writ is much

weaker than in developed countries. The police forces are also poorly equipped and underfunded, therefore introducing new technology and techniques is a tougher task.

To counter these difficulties governments often charge specific departments with the sole job of overhauling and modernizing the police force. India, for example, has set up a dedicated division of the Ministry of Home Affairs called aptly the "Police Modernization Division". This team studies the existing law and order system, and the challenges facing it, and also evaluates the effectiveness of existing programs and projects.

With these goals in sight the Indian government approved 12,000 crore rupees (or approximately £1.28 billion) [14] in 2013 to modernize the police infrastructure across the states, with twenty percent additional funds being given to large cities such as Mumbai, Delhi and six others. The funds will be used by these cities to procure modern equipment like GPS/GIS for "Dial 100" systems and patrol cars, CCTV systems, vehicle scanners, vehicle number plate identification systems, cyber patrol and communication monitoring systems and integrated GIS-based automated vehicle tracking and management systems [12].

Moreover the Indian government has tried to engage students and active citizens by implementing an open data interface of government statistics and data at

https://data.gov.in/. This data is freely available to be used by any budding entrepreneur in order to help improve public service delivery and tackle governance problems. Given the large number of IT professionals and graduates in India it is no surprise that this approach has paid dividends and a large number of applications and systems are now available to the Indian government, including software that help aid and assist the police in managing and analyzing their data.

Non-governmental organizations (NGOs) can also play an important role in supporting police data collection and analysis as is the case in South Africa. The Institute for Security Studies is an African institute that aims to "advance human security in Africa through evidence-based policy advice, technical support and capacity building" [15]. They have collected crime data from across the country and have developed interactive mapping tools that can assist both policy makers and law enforcement officials.

These examples just serve to highlight some of the work that is being carried out in order to improve the policing and justice systems in states that face similar challenges to Pakistan. Both countries share a common colonial heritage (ruled by the British until World War II) with Pakistan and as a result have comparable policing and governance models. Therefore

models that have been successful in these countries stand a much better chance of working in Pakistan and provide a good model for upgrading the system.

2.2 - Police System Review

2.2.1 - History and structure of the Punjab Police

Pakistan is a country spread over 796,096 square kilometers and has a population in excess of a 170 million according to official figures. Unofficial figures suggest that this number may be closer to 200 million. Punjab is the largest province of the country in terms of population, holding more than fifty percent of the total inhabitants. Given the large population and area it covers, maintaining law and order in the Punjab is a challenging task.

The history of policing in the sub-continent dates back to the ‘War of Independence’ of 1857, as Asad Jamal notes in the Human Rights Initiative’s report [13] on the matter: "[T]he Police Commission of 1860 recommended the abolition of the Military Arm of the Police; the appointment of an Inspector General of Police in the Province; and the placement of police in a district under the District Superintendent with general control wielded by the District Magistrate. Based on the recommendations of the Commission the Government of India submitted a bill that was passed into the Police Act of 1861.” After Pakistan’s independence

in 1947, “[t]he police in the various provinces and regions … [were] established as separate establishments without any nationwide integration of these policing bodies. However, the federal government … established a series of specialised investigating agencies at the national level …”"The legal framework of the police underwent a major change as a consequence of the Devolution of Power Plan. In line with the devolution of Power Plan the office of the District Magistrate was abolished in 2001 and a system of Public Safety Commissions was introduced. These changes were incorporated into a new Police Law which was promulgated in 2002.” Since then the Police Order of 2002 has served as the basic guiding document for all police officers, laying down the fundamental rules and regulations of joining and being a part of the police force.

The ‘police’, itself, serves as an umbrella term for its multiple constituent organizations. The Federal Police alone operates vis-à-vis fifteen different organizations, ranging from the Federal Investigation Agency to the Anti-Narcotics Force. Furthermore, the federal police force and the provincial police forces operate independently of one another. A full diagram listing the various law enforcement agencies that work in tandem has been included in the Appendix 3.

The Punjab Police is headed by the Inspector General of the Police (IGP) Punjab who oversees the workings of the law enforcement agencies throughout the province. The province is further divided into ten regions, each centered around a sizable city and having a Regional Police Officer (RPO) in charge. Punjab is further divided into thirty six divisions, each having a District Police Officer (DPO) who is in charge of policing the district. Each district in turn has a number of police stations with an Inspector (IP) in charge who oversees the day to day workings of the station and its staff. A flow diagram representing this set up has been added to the appendix. The staff of a police station consists of Assistant Sub-Inspectors (ASIs), Head Constables and Constables. This hierarchy is illustrated in the following diagram from Asad Jamal's report [13].

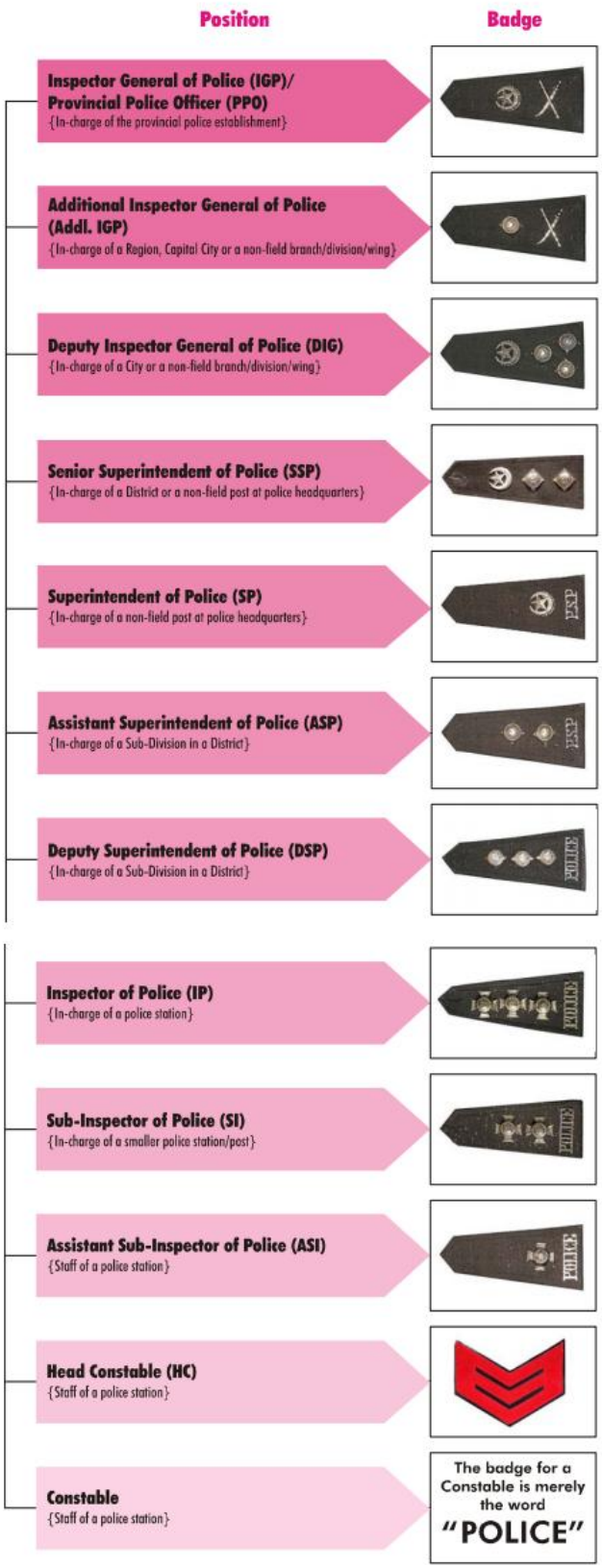

Position
Badge
Inspector General of Police (IGP)/ Provincial Police Officer (PPO)
{In-charge of the provincial police establishment}
Additional Inspector General of Police (Addl. IGP)
{In-charge of a Region, Capital City or a non-field branch/division/wing}
Deputy Inspector General of Police (DIG)
{In-charge of a City or a non-field branch/division/wing}
Senior Superintendent of Police (SSP)
{In-charge of a District or a non-field post at police headquarters}
Superintendent of Police (SP)
{In-charge of a non-field post at police headquarters}
Assistant Superintendent of Police (ASP)
{In-charge of a Sub-Division in a District}
Deputy Superintendent of Police (DSP)
{In-charge of a Sub-Division in a District}
Inspector of Police (IP)
{In-charge of a police station}
Sub-Inspector of Police (SI)
{In-charge of a smaller police station/post}
Assistant Sub-Inspector of Police (ASI)
{Staff of a police station}
Head Constable (HC)
{Staff of a police station}
Constable
{Staff of a police station}
The badge for a Constable is merely the word "POLICE"

Figure 2

2.2.2 - An overview of the FIR method and how it is stored

In Pakistan the police system is based on a system of First Information Reports (also referred to colloquially as FIRs). These are written documents prepared by the police when they receive information or a complaint regarding any criminal activity. It is a report of information that reaches the police first in point of time and that is why it is called the First Information Report. [16]

In essence it is a complaint lodged with the police by either the victim of the crime or by someone on his/her behalf. Any person may get an FIR registered, either orally, in written form or even over the telephone. One does not need to be the victim as representatives, witnesses and in some cases concerned citizens can also get FIRs registered. By law the police are required to register an FIR when they are approached and can not refuse a complainant. The police officer is also bound to fill in the FIR in keeping with the words of the complainant; he or she must be satisfied that the information has been recorded as per the details described. Once contented with the report the person giving the information signs the FIR thereby formalizing the complaint on his or her behalf. Usually an FIR contains

information regarding the offence committed, the complainant's details and some data regarding the police station and/or reporting officer.

In addition to the responsibilities of the reporting police officer, the person who is getting the FIR registered is also bound by certain requirements. He may not provide any false, misleading or vague statements to the police. Frivolous reporting is not allowed; the complaint must be regarding an actual offence under the Pakistan Penal Code. Moreover the complainant is encouraged to get the FIR registered without delay, as the more time he takes the more difficult it becomes for the police to investigate the case.

Once the procedure is completed a copy of the FIR is also provided to the complainant. Any delay or hurdle is viewed as an obstruction in the criminal justice system and may result in disciplinary proceedings against the reporting officer. Once the FIR is registered at the relevant police station the investigation procedure begins. Therefore one can not overstate the importance of the registering of an FIR as it is the first step in the criminal justice system [16]. It also supplies the police with the initial information about any criminal activity that takes place in their jurisdiction and is the platform by which citizens can interact with the law enforcement agencies.

The FIR system in Pakistan is paper based and the FIRs and other crime related data are stored as hard copies. Crime report data is entered into registers which are used to store all the information regarding crimes in the jurisdiction of any given police stations. Whilst there have been efforts to computerise this system as previously discussed, these have only been implemented in a few police stations nationwide and are still in the trial phase. Furthermore all the other information regarding the operations and administrations of the police force is also stored in a similar manner.

This system of keeping records manually is beset with a variety of problems and leads to an inefficient and unresponsive police force. One of the main issues faced by any system like this is the difficulty faced in locating and retrieving records. Not only is it time and resource intensive as compared to an electronic system, the data can only be accessed locally meaning there is no central database that can be accessed to view all the collected records. It also means that sharing of data between different branches is a tiresome task. This leads to another problem as well, as each branch stores separate records it often results in the double entry of data. Apart from data duplication there are no checks on the input data, and given the lack of literacy in the country, there are often plenty of common errors and

mistakes in the record entries. Finally security is always an issue when storing the data in hard copy format; not only is it hard to implement any form of access control it also runs the risk of getting damaged physically, stolen or lost.

Section 3 - Design:

3.1 - Requirements

3.1.1 - Software Requirements

The initial version of this application is designed to be used by the police force of the city of Lahore in order to better understand the data they store regarding crimes. Keeping the client's requirements in mind the following features must be included in the application.

Enter crime report data: The user should be able to add in new crime reports to the database. An entry screen should be designed with all the relevant fields and should include a text field where a descriptive entry may be made.

Edit crime report data: The user should be able to edit crime reports that have previously been added to the database.

Delete crime report data: The user should be able to delete erroneous crime reports that have been entered into the database.

Create new graph (by crime type/police station/time of day): The user should be able to create a variety of charts and graphs (such as a histogram, bar charts, pie chart, line graph, etc) using variables and perimeters of his choosing. He or she should also

be able to isolate certain data fields so that a trends and inferences may be understood between different columns.

Export graph: A user should be able to export a graph to PNG or JPEG formats so that it may be used outside of our application.

View Map: The user should be able to view the map of the entire city and all the crimes that have been tagged onto the map. The user should be able to zoom in and out of the map.

Tag Crime on Map: The user should be able to tag a crime onto the map using it's FIR number so as to identify the crime.

Filter crimes (by type/police station/time of day): The user should be able to choose which crimes to display on the map i.e. be able to select the type of crimes that is displayed or crimes committed between 9pm and midnight.

The following functionalities whilst not necessary are desirable in our application.

Search crime report data: The user should be able to search for individual crime reports using their FIR no. and be able to view the crime report on the screen.

Login Screen and Access Control: The application is protected by a login screen and access is only given to select personnel. Alternatively it could be installed only on selected computers which have already been secured.

Administrator Privileges: The administrator would have privileges to add a new user, delete users and approve password changes.

3.1.2 - General Requirements

In addition to the specific software requirements the following general stipulations should also be considering in designing the application.

Usability: It is important to keep in mind that computer literacy in Pakistan is one of the lowest in the world [19]. Therefore user-friendliness and simplicity is of prime importance as the average user would be unfamiliar with most desktop conventions. The application should be intuitive and the layout should be clear and concise. Moreover a user guide could be created in both English and Urdu in addition to a set of tutorial videos that would provide a run-through of the application's functions.

Security: Crime data is sensitive and very valuable, therefore it must be held securely and should prevent unauthorized access. The application must also maintain data integrity and cleanly transmit the data from the front end to the back end.

Reliability: Once implemented the application will be used in an important phase of the criminal justice process and given the environment it will be used in, any complications and errors that would arise could have serious consequences. Reliability of this software is key to its acceptance and widespread use, otherwise its potential benefits would go to waste. Consequently a well thought out and uncomplicated design is the need of the hour, along with the use of reliable libraries. Thorough testing and debugging would also be integral to its success.

Availability: The term availability refers to the ratio of time a system or component is functional to the total time it is required or expected to function [20]. As has already been discussed with regards to the environment the application will be used in, availability should be maintained so as to not interrupt the smooth working of the police force. In particular the application's maps interface should be up to date and should use a reliable provider which does not break down too often.

Adaptability: The system should be designed in such a way so that it would be easily adaptable for much heavier loads and different locations. Whilst the initial version will target the city of Lahore, the software should be designed in such a way that it can easily be extended for use in other cities and, if it performs well enough, the whole country.

Maintainability: Keeping in mind the evolving needs of combating crime our system will need to be constantly improved and updated. Our system should be laid out in such a way that it is easy to isolate specific functionalities and work on them. The classes and functions should be clearly demarcated and it should also allow for whole classes and functions to be easily replaced and improved to meet new requirements.

Scalability: It should be easy to extend the applications functionality and add in features to improve both the data manipulation and data visualisation operations. Additional chart types, map features, data criterion and additional database functionality should not be too time or resource-intensive to implement.

3.1.3 - System Requirements:

The data visualisation system does not require very sophisticated or advanced hardware to run, any Windows 7 PC can run the application. Therefore its requirements are the same as for the Windows 7 operating system [28].

1 gigahertz (GHz) or faster 32-bit (x86) or 64-bit (x64) processor

1 gigabyte (GB) RAM (32-bit) or 2 GB RAM (64-bit)

16 GB available hard disk space (32-bit) or 20 GB (64-bit)

DirectX 9 graphics device with WDDM 1.0 or higher driver

An internet connection would also be required in order to use the mapping capability of the application though this requirement may be removed in a later version by storing map tiles offline.

3.2 - Software Architecture

The relationship between the three main components of our data visualisation software are represented in the following software architecture diagram.

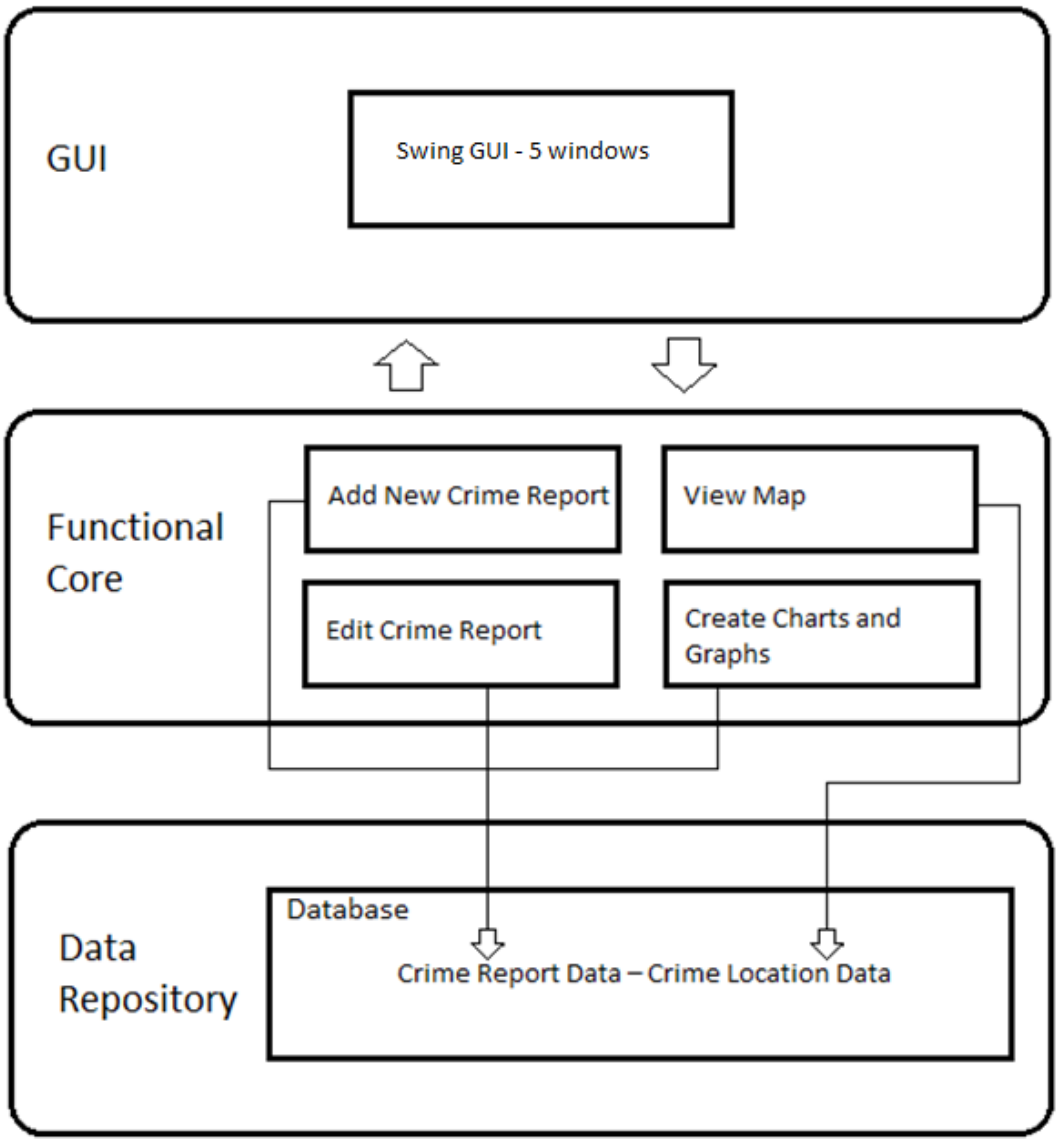


Figure 3

3.3 - User Interface

The user interface shall be designed on the Netbeans IDE using the Swing GUI widget toolkit. The interface will consist of five main windows for our GUI which the user will be able to navigate through. These will include a home screen, a window where crime report data may be input to, edited or deleted from the database, a window which will include the charting and graphing tools, and finally a separate frame with the embedded map and related features. These have been illustrated as follows.

Home Screen:

This is the default window that appears once the application is run. It will allow the user to navigate to all other windows. It will also be the window that the user would return to after pressing the back button on each subsequent window. Additionally it could incorporate some form of access control before opening for the first time so as to prevent unauthorised access to sensitive police crime data.

Input Data Screen:

This window must contain the relevant input fields so that the user may enter crime report data into the database. It must contain both text fields for descriptive forms of data, spinners for numerical forms of data and drop down menus for options. It should

also include checks on the length of the input data for text-fields to ensure the user does not try to enter in a greater number of characters than the database field limit. Finally there should be a check to ensure that none of the required fields is left empty. Ideally when activated this piece of code would highlight the required fields so that the user would know where he or she needs to enter in the data. This screen should also contain an option to clear the values in all fields and a "back" button to return to the home screen without saving the data.

View Data Screen:

This window should allow the user to view the database entries in a tabular form. It should display all fields except those that contain the geo-position co-ordinates. The table should be interactive and should allow the user to select single or multiple entries. These selected entries can then be deleted or a single entry may be updated. The table should dynamically update the values as they are changed.

View Map Screen:

This is perhaps the most important window of the whole application; it should display a scalable map of the city of Lahore along with the crime mapping tools. The crime geo-position co-ordinates should be used to draw markers on the maps to represent the

locations of the crimes. This will allow the user to view crime scatter and hotspots. In addition a data manipulation tool should also be added so as to allow the user to choose which type of crimes to view on the map, based on different factors such as date, category, time, etc. Checkboxes can be used to filter the data according to various types and these in turn would activate input fields which the user could use to customise the range of values used to query the database.

Screenshots of the final graphical user interface have been added to the appendix.

3.4 - Database Design

After looking over the crime reporting system of Pakistan in the Background and Literature Review sections, the following database model for the application has been designed. It provides a basic layout of the sort of crime data that will be stored in our data repository and the data types that will be required. It consists of two tables, one to hold the FIR data and the other to hold the crime location data for the mapping interface.

Crime reports table:

Entry ID (Integer - 10 - Primary Key)

FIR Number (Integer - 10)

Date (Date)

Time (Time)

Complainant First Name (Variable Character Field - 100)

Complainant Last Name (Variable Character Field - 150)

Complainant Address (Variable Character Field - 200)

Offence under Penal Code of Pakistan (Variable Character Field -)

Crime Type (Variable Character Field - 50)

Police Station ID (Integer - 10)

Police Station (Variable Character Field - 100)

Location (Variable Character Field - 150)

Weapons (Variable Character Field - 200)

Stolen Goods (Variable Character Field - 200)

Criminal Vehicle Details (Variable Character Field - 200)

Details (Mediumtext)

Crime Locations table:

Entry ID (Integer - Primary Key - 10)

FIR No. (Integer - 10)

Latitude (Double - 22)

Longitude (Double - 22)

Key:

Field Name (Datatype - Length - Additional Information)

The following crime types will be used for this application as listed down by the Pakistan Bureau of Statistics [17] and the Punjab Police website [18] - Murder, Attempted Murder,

Injury, Kidnapping/Abduction, Kidnapping/Ransom, Robbery, Burglary, Motor Vehicle Theft , Motor Vehicle Snatching, Rape and Gang-rape. Moreover a list of all 89 police stations of the Lahore area has been obtained with each one getting assigned a number from 1 to 89. These steps will aid in the design of the data manipulation tools.

The database will be developed as an embedded database using the Java Derby framework and a JBDC connection will be used to link our database with the Java application.

3.5 - JAVA Backend

The main JAVA code can be divided into three parts: a database interface which would allow our Java application to interact with the database, an implementation of a charting library which would allow for graphics to be created and thirdly an embedment of the mapping component which would manage the geographical data. The implementation of these three parts can be carried out separately and in this order so as to develop the application in an efficient manner.

Given the widespread use of databases with Java, there is a lot of documentation and guides available on which to base the data management segment. The Java Database Connectivity (JBDC) library has been developed by Oracle Software which allows the Java client to access and interact with the database tables. Once the database schema has been

finalised, the main steps required would be to set up our database and connect it with our Java application. This may be done by creating an entity class for each of our tables by implementing the Java Persistence API (JPA). Once our database is up and running we can add Read, Update and Delete functionality to our application using already existing methods from our entity class. Not only does this class contain general data access queries but it also allows us to define custom queries ourselves which will be useful in filtering the dataset for the mapping and charting components.

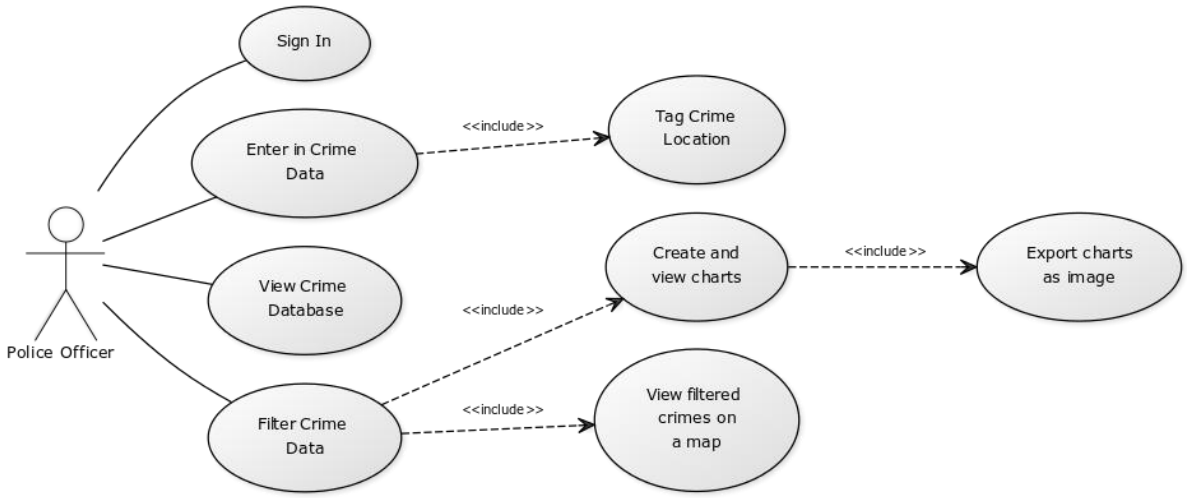
The graphs and charts section has two aspects to it – data manipulation and data visualisation. Data manipulation would allow the user to choose a variety of different parameters and variables with which to construct his or her chart. These can be specified using the input perimeters defined in the user interface which in turn can be used to define the database queries. This would allow the user to select different types of charts along with defining parameters for the range. Once the query has been executed we should load the results in a dataset and then use this to build our chart. There are various options available with this functionality but the most widely used library for Swing is JFreeChart. The API is well documented, thoroughly tested and has proven to be reliable over the years

and it is for this reason that it is the ideal choice to implement the required charting functions.

Finally the third part of main Java application will be to integrate maps into our application so that our crime reports may be geo-tagged and viewed on the map of the city of Lahore. There are multiple ways to go about doing this but the methodology to achieve this has been narrowed down to two main options. One method is to use the Google Maps Java API but the problem with it is that it is designed for browsers only and written in JavaScript. It is possible to work around this by embedding a browser in our Java Swing application and using it to access Google Maps via an HTML page. The benefit of this method is that it would allow the use of the inbuilt JavaScript functions and methods that have predefined by Google. An alternative method would be to use a Java wrapper for OpenStreetMap defined by Java.net in conjunction with NASA. Whilst OpenStreetMap does not have the same detail and options as Google's interface it is an open-source package which means that it can be used by anyone free of charge. Google's API on the other hand has certain permission issues, especially for desktop applications and getting a license could be tricky.

3.6 - Use Case Scenario

A use case diagram was designed to help depict the required functionality of our application.



3.7 - Class Diagrams

The Class diagrams for our project have been added to the Appendix 5.

Section 4 - Implementation

4.1 - Implementation overview

The project consists of eleven classes that have already been presented in the class diagrams in section.

The PoliceDataApp class is the main project class that is created automatically by the NetBeans IDE. The only function it performs it to create a new instance of the MainWindow class and makes it visible.

The MainWindow class is the default class that is loaded when the project is run. It performs navigation functions only and contains code that creates a new instance of each of the remaining classes once the user makes a selection. It then disposes of the MainWindow class.

The CrimeReports class provides an interface between the database and our front end. It implements the Java Persistence API (JPA) and implements the Serializable interface. Object serialization is mechanism provided by Java where "an object can be represented as a sequence of bytes that includes the object's data as well as information about the object's

type and the types of data stored in the object" [21]. It defines how the Java variables and functions will be stored in memory. The class defines variables for all the database fields and uses get/set methods to update this data. It also contains Java Programming Query Language (JPQL) queries which can be used to extract crime reports data according to specific criteria as well. Some custom queries have also been implemented to allow for counting entries for the charting data set. All database functions are run through this class and the Persistence.xml file.

The AddReport class is used to input crime reports' data into our database. It uses an entity manager to create a Crime Report entity object which has variables assigned on the basis of the database fields. Each input field value is then assigned to the appropriate variable of the Crime Report object and then the whole entity is passed into the database as one complete entry once the user clicks the "submit" button.

The AddLocation class is used to input the crimes' geo-position into our database. The constructor receives the database field identity from the AddReport class and uses this to load the saved entity object. Two new fields ,the latitude and the longitude, which were initialized to null values are now updated. The map is displayed using a JXMapKit which is explained in detail a few paragraphs later. In order to get the mouse input we add a mouse

listener which listens for mouse click events. Once the mouse is clicked over the map we obtain the x and y co-ordinates of the point which are then converted to a geo-position using an inbuilt method. The following snippet of code illustrates this procedure:

```
jXMapKit1.getMainMap().addMouseListener(new MouseInputAdapter() {

    @Override
    public void mouseClicked(MouseEvent e) {

    //Get the poisition of the mouse
    Point2D mouse_over = e.getPoint();

    //Convert the point to a GeoPosition
    geoPosition =
jXMapKit1.getMainMap().convertPointToGeoPosition(mouse_over);

    point_lat = (geoPosition.getLatitude());
    point_long = (geoPosition.getLongitude());

    jXMapKit1.setAddressLocation(new GeoPosition(point_lat, point_long));
    jXMapKit1.setAddressLocationShown(true);

    }
}
```

Once the points have been selected and assigned to the entity object, these are persisted to the database.

The ViewData class uses a JTable and a list structure to display the values from the database. The entity manager library is used to select all values from the database which are stored in a list data structure. Our JTable is then bound to this list to represent any update or change in the values. The JTable is designed in such a way so as to only show the values which were input in the AddReport class and the others are hidden. An integer array is used to store the selected rows from the table and these can then be edited (single row only) or deleted.

The UpdateData class uses the same interface as the AddReport class except that it takes in the value of a database field identity number and uses this to obtain the field values. These are then set into the input fields so that the user may view the data before editing it. The user then has the option to either overwrite the saved exit without saving. The UpdateLocation class follows a similar pattern, the only difference being it deals with map co-ordinates instead of text based and numerical data.

The CreateCharts class implements the JFreeChart library and uses the JPA API to build the chart dataset from the database. It calls on custom JPA queries that have written to count the number of crimes committed based on specific values. Specific parameters may be passed into the queries in order to limit the range of data that is used to build our chart, for

example only using a specific year or time frame. The response to our query is returned in the form of a single result and not a list as it usually the case with database queries, which is then added to our data set as a new value i.e. a separate bar or pie slice. Once all the data set values have been specified we use the inbuilt JChartFactory method to build a chart using our data collection. This is then displayed in a Java panel, which was initially blank, in the same frame. Once the chart has been created it is possible to export the chart as a JPEG image in the current directory courtesy the exportChart() method.

The ViewMap class implements the JXMapKit wrapper and uses this to display crime locations on the map. The JXMapKit is a wrapper of the JXMapViewer library and provides a Swing graphical user interface component with integrated zoom and mini-map. It can be configured to use a variety of tile providers to build up the map but has been currently set up with OpenStreetMaps. It also allows for the use of inbuilt JXMapViewer functions such as those used to convert mouse co-ordinates to geo-positions etc.

The JTextFieldLimit class implements a Document Listener which would limit the length of our text field input. This is to ensure that we do not try to persist any strings into our database that are longer than the field's column width. The method takes an integer value as a parameter which determines the maximum number of characters the text field can

hold. Once this limit is reached the text field the excess characters are offset and do not get added to our string.

4.2 - Changes from Design

During the implementation stage there were some changes and optimizations made to our original design.

Database tables: In the preliminary design our MySQL database was set up with two tables, one containing the crime reports data and the other containing the geo-position co-ordinates of each crime. These would be linked by a foreign key such as the FIR number by which an entry into one table could be linked to an entry in the other. However upon further consideration it was decided to merge both tables and contain all the data in one final table in order to improve efficiency and simplify the design. This approach would also remove the need for choosing and implementing a foreign key which could prove tricky.

As a result the Crime Reports table was extended and latitude and longitude fields were added to it. These are initiated to a null value whenever a new crime report entry is added to the database. The entry is accessed again whenever a crime's location is being tagged and these values are then updated with the new co-ordinates as specified by the user using

the mouse listener described in Section 4.1. If any database entry contains null co-ordinates the user is prompted to enter in this data whenever he tries to access the details.

Map Tile Provider: This was not as much a change as a choice that was already explained in Section 3.5. The final decision to use the JXMapKit map interface powered by OpenStreetMaps was influenced by the licensing and copyright issues involved with the alternatives. It is also easily integrated in Java directly and does not need an HTML page to access the mapping API. There is a slight compromise as far as the detail of the map and added features is concerned (Google's API is much more extensive) but there are no permission issues which means the application can easily be adapted for other areas as well.

4.3 - Libraries used

EclipseLink: Used to provide the Java Persistence functions and methods in our application. It allows us to define persistence entity objects for each table in our database and use these to add, remove and update data. In addition to this EclipseLink allows us to use JPQL queries to interrogate the data we have stored.

SwingX: This is a library which contains extensions to the Swing GUI toolkit, including new

and enhanced components that provide functionality commonly required by rich client applications [22]. It contains input components and validation that was implemented in our AddReport class.

MySQL JBDC: Used to provide database connectivity from our Java Application to the MySQL database that is being used to store data.

JFreeChart: It is the most popular Java charting library. It allows the user to create a variety of charts into swing applications and is free to use. The charting component of the application is based on the JFreeChart framework and uses it to create bar and pie charts.

JCommon: This class is used by the JFreeChart and contains miscellaneous classes that support configuration and dependency management code, a general logging framework, text utilities, user interface classes for displaying information about applications and custom layout managers [23].

4.4 - Problems faced during Implementation

Every project faces hurdles during the production phase and this one was no different. Some of the issues faced by the developer were as follows:

Query Builder: One of the most important feature of the crime-mapping interface was the capability of the user to filter the query data and view crime locations in the city according to a specific criteria. The interface was designed in such a way so as to allow this criteria to be selected on the basis of checkboxes which in turn enabled input components. A JPQL query would then be chosen which would include the selected criterion based on the checkbox selections, and parameter values obtained from the input components. However this model was problematic as it involved writing a huge number of if/switch statements in order to match every possible checkbox combination. As the number of checkboxes increases the number of possibilities also increases exponentially so this model is also neither scalable nor easily adaptable. In fact given four checkboxes the total number of possibilities is:

$$4C3 + 4C2 + 4C1 + 2 = 16 \text{ total states}$$

In order to work around this a query string builder was designed and implemented and thereby bypass the need for any conditional statements based on the checkbox values. In the updated implementation each checkbox's value is stored in an array; 0 for not selected and 1 for selected. This array (called is_checked) is then iterated through and all the selected values' indexes (where the indexed value is equal to one) are added to an Array List (called is_checked_chosen). This Array List, containing the array's index values, is then

iterated through and a switch statement is called upon each time to choose a query object, a query value, a perimeter object and finally a perimeter value based upon each index value stored in the array list. The selected values and objects are then appended to a query string in order to build our final query.

The pseudocode is described below:

Hold checkbox values in array
Check for positive checkbox indexes <---- Hold in ArrayList
for (Iterate through ArrayList)
    Check Index Value
    Choose query object value
    Choose query parameter value
    Concatenate strings to build final query
Execute query
Display resultant co-ordinate points on map

Input checks: Any reliable application must check the input values which the user is trying to enter into the database and make sure the data is valid before it is persisted into the database. The initial approach taken to implement data validation was to add key listeners and prevent further data from being entered into the text field once a specific limit had been reached but this method proved to be unreliable and too cumbersome. The solution to this

was to implement a document listener as explained in a tutorial on Java2s.com [24]. The document listener is set down in a separate class called JTextFieldLimit and contains a method which takes the maximum input length in as a parameter. This method can then be called on every text field to specify the input range.

Adding Crime Locations: Initially it was planned that the frame in which the user entered in the crime report details would also contain a map component so that the user may be able to enter the location in simultaneously. However this meant that the screen became too cluttered and user-friendliness was compromised. In order to rectify this a new frame was added which would only contain a map wrapper and a mouse listener to input location data. However a problem arose here concerning how to link the location data to the specific crime report it pertains to. The solution was to build new constructors for the AddLocation class which would take the database field ID as a parameter and use this to update the latitude and longitude columns of the entry. Whenever the Add Location frame is initiated the database ID from the Add Crime Report field is fed into the constructor and this ensures that the correct entry is updated each time. This functionality was also extended to the

UpdateData and UpdateLocation class which require data to be pulled from the SQL tables as well.

Section 5 - Testing

Testing is a crucial stage in any project's development and allows for any errors or problems in our projects to be identified and remedied. It also helps us optimize our code and plan improvements for future updates and editions. Unfortunately due to a shortage of time only a limited amount of testing could take place.

5.1 - Unit Testing:

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed [25]. Whilst extensive unit testing was not carried out, the database functionality was tested using JUnits.

In order to do this we used the DBUnit library and created an in-memory database. We then populate our database with preset values using XML datasets. Finally we check to ensure that the data has been successfully passed into our database by creating queries and comparing the data with our preset values. As both values match then we know that data is being successfully entered and pulled from our database.

Due to a shortage of time unit tests could not be set up for the other components but this will be remedied for future editions.

5.2 - Real World Testing:

In order to get a better idea of the application's usability and performance in the real world the application was run through in front of some civil servants and senior police officials, and then they were given a chance to use the application themselves.

They were given a short tutorial on how to use the application and were then asked to enter in one crime report and then view it on the map. Despite being given clear instructions some users still struggled to run the application properly and needed further guidance. In addition to this the chart interface was criticized to be too complex and was updated to make it a bit more intuitive.

On a more positive note the input checks and error checking components worked well and prevented incorrect data from being entered into our application. The mapping component's functionality was also praised for being versatile and usable.

Section 6 - Ethical issues and considerations

There are a lot of ethical issues and considerations that have to be kept in mind whilst designing, programming and implementing any piece of software. These are detailed on the British Computing Society (BCS) website [26]. Some of the concerns that the developer had to face are discussed here.

Use of police data: One of the main dilemmas faced by the developer was how to test the application for the large amounts of data the police would enter into the database. Would the software be able to take the load and how would performance be effected? One way of testing how the system would cope would be to use actual police data and add it to our database. The developer had access to a significant amount of police data with crime details and locations, however due to legal and ethical concerns this database could not be used. Police data is sensitive and confidential in nature and it is not allowed to be used by non-police personnel. Even if they were approached through the right channels it was unlikely that the police would give permission to use this data. As a compromise the development team decided to approach the police and inquire whether they held any dummy or outdated data that could be used in the testing phase but unfortunately there was no such details available. In the end it was decided that the data shall be input

manually in order to mimic the working conditions under which the system is expected to operate.

Data Security: As has already been mentioned multiple times in this report police data is highly sensitive and needs to be protected. Nobody should be able to access and view this data without prior authorization and authentication. Data security should be maintained so as to ensure its confidentiality, integrity and availability as has been covered in Section 3.1.2.

Plagiarism and Licensing: As with any project originality and copyright issues had to be kept in mind. Efforts have been taken to give references and citations wherever external information has been used in order to give credit where it is due. The code has also been commented on wherever the ideas from tutorials have been implemented so as to clear up any misconceptions.

Similarly care has also been taken to ensure that all the libraries and other JARs used in our project's functionality are either open source or a license has been obtained in order to ensure no copyright infringement takes place.

Integrity: This application will be used in a high pressure environment and will assist decision-making that will have significant effects on the populace of a major city. Taking this into account it must be ensured that our application maintains the integrity of the data entered into it whilst making visualizations. Integrity here refers to the truthfulness of the data; the graphics and charts that we produce should not give misleading information or lead to untruthful conclusions. It was important to ensure when selecting our charts and designing them that they display the data as it is rather than what the user might want to see.

Section 7 - Evaluation

7.1 - Software Evaluation

7.1.1 - Functional requirements:

Requirement	Status
Enter crime report data	Implemented
Edit crime report data	Implemented
Delete crime report data	Implemented
Create new graph (by crime type/police station/time of day)	Partially Implemented
Export graph	Implemented

View Map	Implemented
Tag Crime on Map	Implemented
Filter crimes (by type/police station/time of day)	Implemented
Search crime report data	Not Implemented
Login Screen and Access Control	Not Implemented
Administrator Privileges	Not Implemented

As is evident from the table most of the basic requirements have been met but there is still significant room for improvement left over. The optional functionalities that were desirable have not been included in the first version but hopefully will be implemented later on.

With regards to the necessary requirements most of these have been implemented. The application is able to access an embedded database and store crime reports in it. These can

be accessed later as well and viewed, deleted and edited. The database functionality performs all of these tasks up to the standard required and ensures the data storage component has been implemented up to the required standard.

The crime mapping component also covers most of the essentials and meets the basic needs of the user. The crimes can be geo-tagged on the map and more importantly these can then be filtered so as to analyse the distribution and patterns. Crime hotspots can be identified

7.1.2 - Non-functional requirements:

Usability: Efforts have been made to make the user experience as straightforward and simple as possible. In addition to this a user guide has also been prepared so as to give the user a run through of the application. The application was also tested with lay users so as to test the functionality and ease of use. Given the intuitive and uncomplicated layout the application performs well in this regard and scores well on usability.

Security: Security is always a concern with any application, particularly more so with one that deals with sensitive or important data. Unfortunately the software does not do well in this category as few security concerns have been addressed. Although the application's

data is held locally which means server security and data transmission is not an issue, authentication and authorization needs are not addressed. The software is only as secure as the system it is held on.

Reliability: The application was tested in a variety of situations in order to judge reliability. In addition to this the libraries used in the project are widely used and well reputed and hence are unlikely to fail. Whilst there were no serious issues during testing more extensive trials are required to determine the reliability under more heavy loads.

Availability: The application does not perform so well in this regard as it does not work without internet connectivity, however this can be remedied in a future version by storing the map tiles offline. Apart from this the application is functional under most conditions so availability is not a serious issue as long as there is a sound internet connection.

Adaptability: The application can easily be adapted for different locations by changing the map co-ordinates and updating the AddLocation class if needed. The crime reports procedure is uniform across Pakistan meaning the rest of the application can work across the board. This means our software works well with regards to adaptability and can easily be modified to meet different needs.

Maintainability: The application is divided neatly into different modules each one encapsulating a different functionality. This means it is easy to isolate a specific task's code and to update it. This allows our application to be easily maintained and improved when necessary.

Scalability: As was discussed in the previous section the design separates the functionalities and this helps the scalability of the application. As there is a separation of roles it is easy to extend particular classes and add more code to extend our software.

7.2 - Project Evaluation:

The project benefitted from being planned in advance and being divided up into separate sections each with a different set of disciplines. As a result a majority of the project was complete in time and most of the basic functions were implemented. However extensions and further improvements could not take place which will hopefully be addressed in future versions.

The background research and literature review sections provided a strong understanding of the topic area and the context in which the application would be used. It helped to provide a

solid foundation to the project and general guidelines on how to design and implement the application.

The design phase involved understanding the needs of the user and designing the software features accordingly. It also meant selecting appropriate libraries and trying to add in as much functionality as possible. This phase involved a few lapses but was carried out smoothly on the whole. The implementation phase was faced with a few hiccups, but in the end most of the requirements were met and added functionality will be included in future updates.

The testing phase was carried out on a modest level due to time constraints and it is hopeful more extensive trials will be held on the software for future versions. The evaluation period allowed for some reflection on the final product and an analysis of both the achievements and weaknesses.

In conclusion the project was a fair attempt at a fairly difficult problem. Public service, particularly in third world countries, is a tough field to operate in and requires a noteworthy amount of effort to bring about any improvement. Whilst the project did not meet all of its

aims it did provide a good starting point for improving police service delivery in Pakistan and has the potential to make a significant difference provided extra time and effort.

A Ganntt chart of the project's deliverables, prepared in October 2014, has been added to the appendix.

Section 8 - Conclusions and Future Work

This project was motivated after a visit to a local police station in my hometown Lahore where I observed the difficulties the police faced in storing and understanding their records. Given the lack of resources and technical knowledge the police force are handicapped by the aim was to design a simple and undemanding application which would provide as much functionality as possible. The project was able to meet most of the initial aims and requirements set out for it and has provided a good platform to further explore and improve police service delivery in Lahore.

As with every project there is room for further improvement and extensions to be put in place. One of the major issues faced by the develop this time around was not being able to get in touch with either the Central Police Office in Lahore or the Punjab Information Technology Board (PITB) [27] despite repeated attempts. Although the proper channels were explored the red tape and bureaucracy involved meant that none of the officials who manage police work or the government's computerization efforts could be contacted. These office-bearers can provide valuable guidance regarding the procedures and methodologies of the public sector, and give an inside understanding of efforts to modernize it. As an

example this project could be linked with the police's central criminal database which would allow crime data to be matched to specific criminal individuals and organizations.

Moreover the project would benefit from having a larger team working on it. Creating a comprehensive and effective software for public institutions is an ambitious and vast operation. Whilst this project has strived to include as much functionality as possible, given the limited time frame and resources there is still room for many new features and improvements. These include upgrading the mapping interface to show greater details about criminal activities, creating a more dynamic and interactive user interface and adding more charts and visuals to analyse the data set.

Finally on a person level the project gave me a good understanding into some of the problems the public sector faces and how systems can be modernized in order to improve public service delivery. The value of data can not be overstated in the modern era, and decisions making is now driven by how well we understand the information available to us. With the advent of greater connectivity, smart phones and the mobile internet we now have more tools than ever to upgrade and update the systems used to run our civic institutions, and by consequence improve the quality of life in our countries.

Section 9 - Bibliography

1. WebDesk (2011) *Pakistan ranks low 105 on WEF's Information Technology report*, Karachi: Express Tribune.
2. Ilinksy, N.Steel, J. (2011). *Designing Data Visualisations*. United States of America: O'Reilly Media. 45.
3. Punjab Police. (2010). *Police Record and Office Management Information System (PROMIS).* Available: http://punjabpolice.gov.pk/promis. Last accessed 16th Apr 2014.
4. Khan, I. (2014). Pakistan most terror-hit nation. Available: http://www.dawn.com/news/1088864/pakistan-most-terror-hit-nation. Last accessed 16th April 2015.
5. Reuters. (2009). Operation ends; police overpower terrorists in Manawan. Available: http://www.dawn.com/news/957792/operation-ends-police-overpower-terrorists-in-manawan. Last accessed 16th April 2015.
6. Baldwin, H. (2015). Drilling Into The Value Of Data. Available: http://www.forbes.com/sites/howardbaldwin/2015/03/23/drilling-into-the-value-of-data/. Last accessed 16th April 2015.
7. Rouse, M. (2012). What is Data Visualization?. Available: http://searchbusinessanalytics.techtarget.com/definition/data-visualization. Last accessed 16th April 2015.
8. United Nations. (2014). Data revolution advisory group named by UN Secretary-General. Available: http://www.un.org/apps/news/story.asp?NewsID=48594#.VS-kQiGqqko. Last accessed 16th April 2015.
9. MicroStrategy. (2013). Best Practices in Data Visualisation. Available: https://www.microstrategy.com/Strategy/media/downloads/training-events/microstrategy-world/2014-barcelona/MSTRWorldEU2014_T1_S3_Best_Practices_in_Data_Visualizations.pdf?ext=.pdf. Last accessed 16th April 2015.
10. Abela, A (2013). Advanced Presentations by Design: Creating Communication that Drives Action. 2nd ed. United States of America: Pfeiffer. 99.

11. Kirk, R (2007). Statistics: An Introduction. Lahore: Cengage. 52.
12. Snelling, P. (2013). How data visualization helps crack down on crime. Available: http://www.sas.com/tr_tr/insights/articles/risk-fraud/data-visualization-crime.html. Last accessed 16th April 2015.
13. Jamal, A (2010). *Police Organisations in Pakistan*. New Delhi: Commonwealth Human Rights Initiative. 1212.
14. Time of India. (2013). India approves Rs 12k crore plan for police modernisation. Available: http://geospatialworld.net/News/View.aspx?id=26417_Article. Last accessed 16th April 2015.
15. Institute for Security Studies. (2013). About. Available: http://www.issafrica.org/crimehub/about. Last accessed 16th April 2015.
16. Commonwealth Human Rights Initiative. (2012). First Information Report - A Guide for Citizens. Available: http://www.humanrightsinitiative.org/publications/police/fir.pdf. Last accessed 16th April 2015.
17. Government of Pakistan. (2012). *Crime Statistics 2003-2012.*Available: http://www.pbs.gov.pk/content/crimes-reported-type-2003-2012. Last accessed 4th Dec 2014.
18. Punjab Police. (2014). *Crime Statistics.* Available: http://punjabpolice.gov.pk/crimesstatistics. Last accessed 4th Dec 2014.
19. 4 Children Community Initiative. (2012). Computer Literacy - Providing Computer Access & promoting E- learning. Available: http://www.4cci.org.pk/FocusAreaComputerLiteracy.php. Last accessed 16th April 2015.
20. Rouse, M. (2014). Reliability, Availability and Serviceability (RAS). Available: http://whatis.techtarget.com/definition/Reliability-Availability-and-Serviceability-RAS. Last accessed 16th April 2015.
21. Tutorials Point. (2015). Java - Serialization. Available: http://www.tutorialspoint.com/java/java_serialization.htm. Last accessed 16th April 2015.
22. Java.net. (2014). SwingX. Available: https://swingx.java.net/. Last accessed 16th April 2015.

23. JFree. (2014). JCommon. Available: http://www.jfree.org/jcommon/. Last accessed 16th April 2015.
24. Java2S. (2015). Limit JTextField input to a maximum length. Available: http://www.java2s.com/Code/Java/Swing-JFC/LimitJTextFieldinputtoamaximumlength.htm. Last accessed 16th April 2015.
25. Software Testing Fundamentals. (2015). Unit Testing. Available: http://softwaretestingfundamentals.com/unit-testing/. Last accessed 16th April 2015.
26. British Computing Society. (2015). BCS Code of Conduct. Available: http://www.bcs.org/category/6030. Last accessed 16th April 2015.
27. Punjab IT Board. (2015). About US. Available: http://www.pitb.gov.pk/aboutus. Last accessed 16th April 2015.
28. Microsoft. (2014). *Windows 7 system requirements.* Available: http://windows.microsoft.com/en-GB/windows7/products/system-requirements. Last accessed 4th Dec 2014.
29. FindPk. (2009). Travel & Tourism in Pakistan. Available: http://www.findpk.com/yp/html/Punjab-Pakistan-Map.png. Last accessed 16th April 2015.
30. Punjab Police. (2014). Field Organizational Chart of Punjab Police. Available: http://www.punjabpolice.gov.pk/orgchartfield. Last accessed 16th April 2015.