A CRIME INVESTIGATION TOOL BASED ON PATTERN MATCHING IN CRIME EVENT STREAMS

by

Osamai Osbert

Reg. No: 2010/HD18/1259U

BITC (KYU), DCS (KYU),ITIL

Department of Networks

School of Computing and Informatics Technology, Makerere University

E-mail: omeja4ever@gmail.com,omeja4ever@yahoo.com

Tel: +(256)712 738530

A Project Report Submitted to the School of Graduate Studies in Partial Fulfillment for the Award of a Master of Science in Data Communication and Software Engineering Degree of Makerere University.

OPTION: Mobile Computing

November, 2013

Declaration

I Osamai Osbert do declare to the best of my knowledge that the information presented in this report is original and my own work and effort and has never been submitted to any college, institution or University for any form of award whatsoever.

Signed................................ Date..................................

Osamai Osbert

Approval

This project report has been submitted with my approval as supervisor.

Signed................................. Date...................................

Dr. Benjamin Kanagwa, PhD

Department of Networks,

School of Computing and Information Technology,

Makerere University.

Dedication

This work is dedicated to the men and women of the Uganda Police who tirelessly serve and protect our nation.

Acknowledgement

I would like to thank and appreciate all those who contributed time and effort towards this report. Special thanks go to my supervisor Dr. Benjamin Kanagwa for every support accorded to me in completing this project.

I am greatly indebted to my family (My wife Lucy, daughter Alba, sisters Kevine and Suzan and brothers Euphrasius and Godfrey) for all the support and encouragement they offered throughout the course. I also express my gratitude to my workmates at Tropical Bank Uganda especially Mr. Nicholas Ssesinde for the morale and encouragement they provided me during this course.

May the Almighty God reward them abundantly.

LIST OF ACRONYMS

EJB	Enterprise Java Beans
SOA	Service Oriented Architecture
CEP	Complex Event Processing
XHTML	Extensible HyperText Markup Language
JPA	Java Persistence API
EIS	Extended Information Server
PDF	Portable Document Format
IEEE	Institute of Electrical and Electronic Engineers
EL	Expression Language
HTML	HyperText Markup Language
SNA	Social Network Analysis
GIS	Geographic Information System
SPP	Spatial Point Patterns
AMIT	Active Middleware Technology
GPL	General Public License
SQL	Structured Query Language
ECA	Event Condition Action
POJO	Plain Old Java Object
EPL	Event Processing Language
SDLC	System Development Life Cycle
UML	Unified Modeling Language
JSF	Java Server Faces
Java EE	Java Enterprise Edition
CEL	Cayuga Event Language
ESBs	Event Service Buses
ICT	Information and Communication Technology
CCTV	Closed-Circuit Television
GPL	General Public License
API	Application Programming Interface

Abstract

Crime investigation world over is considered a difficult and laborious process that heavily relies on efficient crime analysis to ensure accurate and timely conclusions. This is not helped by the fact that law enforcement agencies deploy manual investigation processes and computerized systems that cannot quickly identify complex crime patterns.In Uganda, the situation is no different with these agencies facing massive delays in crime solving and increasing numbers of sophisticated crimes, most of them of the organized category. Every year tens of thousands of cases are carried forward to the following year, uncompleted.Majority of these crimes evolve in a long period of time making them even more difficult to predict. Criminals have become very intelligent due to the advancement of technology and therefore they conduct crimes in an untraceable manner. Intelligence and law enforcement agencies are often faced with the dilemma of having too much data, which in effect makes too little value and the lack of sophisticated network analysis tools and techniques to utilize the data effectively and efficiently. In this project the phenomenon of Complex Event Processing (CEP) is used to detect patterns in crime related events using a CEP engine for high throughput and performance. CEP analyses low level events to produce a single complex event and has been successfully used in Stock Trading, Network Analysis and other areas. CEP uses a collection tools and techniques()to detect pre-defined patterns/rules in data streams. The patterns are created as queries using query languages and are stored instead of storing data as is the case with databases. Because of this design,the tool will enable quick processing and analysis of crime data and also provide a basis for unearthing emerging patterns.

Contents

1 CHAPTER ONE: Introduction

Crime investigation world over is considered a difficult and laborious process that heavily relies on efficient crime analysis to ensure accurate and timely conclusions. This is not helped by the fact that law enforcement agencies deploy manual investigation processes and computerized systems that cannot quickly identify complex crime patterns.

In Uganda, the situation is no different with these agencies facing massive delays in crime solving and increasing numbers of sophisticated crimes, most of them of the organized category. Every year tens of thousands of cases are carried forward to the following year, uncompleted. As the usual circle of crime would dictate, fresh cases are reported every day, and, gradually, older cases left uncompleted lose the urgency they initially generated and, inadvertently, they die a natural death [17].

To avert this problem there is need to deploy sophisticated tools, technologies and resources that can enable crime investigators quickly reach reasonable conclusions by identifying patterns of behavior in criminals. In so doing, not only will crimes be investigated faster but some future crimes may be prevented based on recurring patterns identified. However, intelligence and law enforcement agencies are often faced with the dilemma of having too much data, which in effect makes too little value. On one hand, they have large volumes of raw data collected from multiple sources: phone records, bank accounts and transactions, vehicle sales and registration records, and surveillance reports. On the other hand, they lack sophisticated network analysis tools and techniques to utilize the data effectively and efficiently [27].

In this project the phenomenon of Complex Event Processing (CEP) is used to detect patterns in crime related events using the Esper engine [5] which is CEP engine for high throughput and performance.Esper is an open-source CEP engine developed by EsperTEch Inc. and volunteers under the GNU General Public License(GPL v2).Esper takes advantage of its two flavours(Esper for Java and NEsper for .NET) to provide APIs that enable processing of events using the Event Processing Language.

Esper was selected for this project because it is freely available as an open-source project,has been proven to scale well with high throughput,has low latency given that dat is not saved first then queried, combines stream processing and Complex Event Processing on one platform,the processing language is similar to SQL and easy to use,supports multiple formats for incoming events(XML,CSV,HashMaps,HTTP,Sockets and more) and has been proven [5] to simplify the complexity of pattern detection in the areas of Stock Trading,Network Analysis and others.

1.1 Statement of the problem

Due to the manual nature of crime investigations in Uganda, case backlog remains a critical issue leading to delayed justice, unpunished criminals and of course a loss of confidence in the agencies and government by the citizenry.

Criminals have become very intelligent due to the advancement of technology and therefore they conduct crimes in an untraceable manner. Majority of those crimes evolve in a long period of time making them even more difficult to predict. Therefore the rate of organized crime is on the rise most of which are orchestrated in vast geographical areas using these complex techniques.

Therefore, manual techniques of analyzing such data with a vast variation have resulted in lower productivity and ineffective utilization of manpower[12].There is need to develop a tool to quicken the investigation process by accurately guiding investigators in evidence analysis and also use recurrent patterns to prevent future crimes.

1.2 Objectives

1.2.1 General Objective

To develop a crime investigation tool that helps investigators by detecting crime patterns in event streams to provide investigative leads.

1.2.2 Specific Objectives

Specifically, the objectives of the study are:

(i) Identify existing crime patterns associated with crimes related to causing financial loss.

(ii) Generate events from existing data source into a CEP engine for processing.

(iii) Setup and detect crime patterns in events using a CEP engine.

(iv) Display to the user/client events matching the pattern and the rate(success percentage).

1.3 Scope

The tool was developed for the Criminal Investigation Department of the Uganda Police to enable investigation officers easily detect crime patterns by quickly processing evidence data.For demonstration purposes,the project focused on crimes related to corruption in government ministries due to the national significance of such cases and the readily available structured data.

1.4 Significance of the study

To Enhance the crime investigation process by using CEP techniques and tools to detect predefined patterns in good time and perform efficient correlations of events. The goals of the study are:

(i) Reduce case backlog through quick and multiple processing of case files to restore public confidence in the Police.

(ii) Reduce human intervention by officers in terms of cross referencing and analysis of crime data.

(iii) Provide a foundation for prediction of future crimes based on current crime trend analysis.

(iv) Reduce government expenditure and reliance on manual crime analysis methods.

2 CHAPTER TWO: Literature Review

This section provides a general literature review of major data mining techniques used in existing crime investigation systems,discuses some studies and systems at the Uganda Police related to this project and explains the concept of Complex Event Processing which is the preferred technology for this project.

2.1 Current Technologies in Crime Investigation Systems

Existing crime investigation systems tend to vary in terms of their overall capabilities and technical operation. In one study [23]the existence of prominent criminal investigation software like HOLMES2, BRAINS and Analysts' Notebook which are used by criminal analysts in the United Kingdom and Holland was acknowledged. This category can be classified as early generation systems that mainly focused on analyzing evidence separately without linking multiple sets of evidence in order to solve crimes. They were also not designed to communicate with existing case management systems to facilitate data exchanges.These systems make use of relational database queries using SQL which is slow in terms of processing.

Second generation systems around the world rely heavily on data mining techniques to query large datasets for meaningful patterns in order to help investigators solve crimes. These methods however fall short in terms of supporting decision making largely due to poor processing speeds and querying algorithms.These systems mainly produce graphical representations of links between criminals and other crime entities.

Link Analysis is a technique used in data mining. These tools have for long been used by law enforcement agencies to identify, analyze and visualize relationships between crime entities. In a study [19] , it is revealed that through association paths linking suspects and victims in crime, link analysis discovers information about motives and hence provides investigative leads.Link analysis requires an extensive amount of data preparation and is highly labour intensive.The performance of this technique detoriates with increasing data amounts and is therefore suitable for smaller observation sets.

In order to correlate entities , investigators must manually search for associations by examining a large number of documents that may range from structured database records of crime incidents to unstructured report narratives. Link Analysis is similar to the breadth-first search alogrithm in which a search tree rooted at one of the known enitities. The process involves examining one or more documents and consumes a considerable amount of time.

Another problem with link analysis is high branching factors as a result of very many asociations between entities which increases the complexity of the search alogrithm. Also paths found during analysis may not be useful as they may contain unimportant links.Link Anlaysis heavily relies on domain knowledge and experience making it very difficult to automate the process. Investigators need to determine whether an association between two crime entities is important for uncovering investigative leads. As a result this is a highly costly technique though effective in a way.

Another technique used in crime pattern detection involves several data mining steps like hotspot detection, crime clock, crime comparison and crime pattern visualization. Numerous algorithms are used to relate multiple crime scenes, represent a number of crime scenes on a daily basis, compare different crimes to estimate growth rates and visualize the changes in crime occurrence frequencies [12].

A study on crime network analysis [27] suggests that law enforcement agencies need to deploy reliable data and sophisticated tools as critical tools in the discovering useful patterns in data. The study introduces a data mining technique called Social Network Analysis (SNA) which is used to discover hidden patterns in large volumes of crime related data. An approach of SNA referred to as block modeling is used in criminal networks to reveal associations between subgroups based on a link density measure. Discovery of new structural patterns during this process can enable prevention of crimes and also modify conventional view of certain crimes by investigators.This approach is expected to provide more advanced analytical functionality to assist crime investigation.Sophisticated structural analysis tools are needed to go from merely drawing networks to mining large volumes of data to discover useful knowledge about the structure and organization of criminal networks [27].

2.2 Related Work in Uganda

In Uganda, some studies have been conducted in the area of crime investigation. One such study [22] discusses a model for forensic investigations that performs detection of incidents through system monitoring and performs data analysis to unearth the crime scene, suspect and how the crime was perpetrated. The study proposed a new model based on five iterative phases that were meant to strengthen the crime detection and analysis process. The model suggested depicts the forensic process as iterative as opposed to linear,differentiates the investigations at the primary (suspect) and secondary (victim) crime scenes, introduces a new phase (Traceback) that would reflect the process of arriving at the perpetrators scene, re-defines the phases in the physical and digital crime scene investigation phases in the previous models, re-defines the Deployment phase in the previous model to include the physical and digital crime investigations,reserves only one reconstruction (at the end) but provides for investigative hypotheses during the entire process and is suitable for cyber-crime investigations.

Another study [15] on crime prevention suggested a combined application of data mining techniques alongside GIS (Geographical Information Systems) to discover crime data in disorganized settings like Uganda. Spatial Point Patterns (SPP) based on coordinates of events such as locations of crime incidences and the time of occurrence are used.All or a sample of point pattern may be plotted on the map. The aim of SPP analysis is to detect whether the point pattern is distributed at random, clustered or regular. SPP is typically interpreted as analysis of clustering. A dot map is commonly used to represent SPP. The tool effectively used for analysis of clustering effects is the K function. This method assesses clustering of crime incidences in detection of hot spots wheretime and space relationship analysis is required, the methods used are Knoxs method,Mantels Method and K-nearest neighbour method.

2.3 Existing Systems at Uganda Police

Currently there is no crime investigation system being used at the Uganda Police but there are ongoing efforts to deliver a case management system to be used by all police stations in the country.Most crime related systems in uganda centre on monitoring crime and criminals rather than investigations and they include;online tracking systems for monitoring cyber criminals,mobile phone tracking system to track stolen or lost phones,camera tracking software to monitor identified criminal vehicles via CCTV cameras.

2.4 The Crime Investigation Process

An investigation is an examination, a study, a survey and a research of facts and/or circumstances,situations, incidents and scenarios, either related or not, for the purpose of rendering a conclusion of proof. An investigation, therefore, is based upon a complete and whole evaluation and not conjecture, speculation or supposition.Crime detection and investigation is both an art and a science; a collaboration of common sense, judgment, intellect, experience and an innate intuitiveness along with a grasp of relative technical knowledge. The criminal investigator must continually apply those skills, acquired through study and experience, to the examination and observation of the criminal and his behavior, as well as his social and physical environment [4].

There are several basic types of investigations that law enforcement personnel may undertake in the routine discharge of their duties: Investigations of incidents, which are violations of laws and/or ordinances that include; criminal acts (robbery, assaults, larceny, burglary, murder, illegal weapons, etc) and traffic accident investigations (serious injuries, likely to die, property damage). Personnel investigations into the background, character and suitability of persons in an effort to determine their eligibility for positions of public trust. Investigations of illegal conditions or circumstances, which if left unchecked would cause an increase in traditional crimes. These conditions may include the following: narcotics sales, illegal weapons trafficking, vice type crimes (prostitution, gambling), street gang activity, organized crime, terrorist front activities, fraud and con games, identity theft and computer crimes. Although many of these conditions would dictate self-initiated investigations based upon intelligence rather than reacting to a citizen crime complaint, there are however, times that investigations will in fact result from such individual crime complaints [4].

The official purpose of criminal investigation in most countries is to retrieve information that can be used as evidence in court.The obtained evidence then becomes the basis for judges' and juries' decisions concerning the guilt of prosecuted defendants and the sentences imposed on those found guilty. From the above description it is evident that investigative activities cannot be fully understood if viewed detached from its context, but should be seen as intertwined with other components of the criminal justice system. Therefore, it is helpful to consider how investigators' work relates to the prosecution process. In a prosecutor's application for a summons, a claim is to be made concerning the criminal behavior of a defendant in the past. A prerequisite for issuing a summons is, first and foremost, that the identity of the defendant is clear. Furthermore, the criminal act must be specified with regard to the time and place of the offense. Finally, the circumstances surrounding the offense should be detailed and proven to fulfill the legal requisites for the specified crime classification. The investigative work carried out by the police authority serves to provide the prosecutor with all the above information [2].

2.5 Complex Event Processing

A Complex event is an event that abstracts or aggregates simple (or member) events [11]. Simple and complex events are normally represented in linear ordered sequences called Event Streams. These streams are usually bound by time intervals and may contain different types of events.

Complex Event Processing (CEP) is defined as the process of detecting complex events using continuously incoming events on a lower abstraction level [3]. This study justifies the need for CEP given the fact that single events on their own may not be sufficient in determining certain patterns.

CEP is a foundational technology for detecting and managing the events that happen in event driven enterprises. It is a collection of methods, tools and techniques applied in processing events as they happen. In order to achieve a lot from CEP, happenings of events in enterprises need to be well understood. This can be achieved by organizing events into structures or hierarchies, identifying relationships among events (causal, time, aggregation) and organizing events in different views from different personnel. In CEP, higher- level knowledge is derived from lower-level events which are a combination of various occurrences. CEP can be viewed in two types, the first one involving specification of complex events as patterns and detecting them effectively, whereas the other type involves detecting new patterns as complex events. In the first case, event query languages offer convenient means to specify complex events and detect them efficiently. In the second case, machine learning and data mining methods are applied to event streams.

Detection of complex events is, of course, not an end in itself; an event-driven information system should react automatically and adequately to detected events. Typical reactions include notifications (e.g., to another system or a human user), simple actions (e.g., buy stocks, activate fire extinguishing installation), or interaction with business processes (e.g., initiation of a new process, cancellation or modification of a running process).

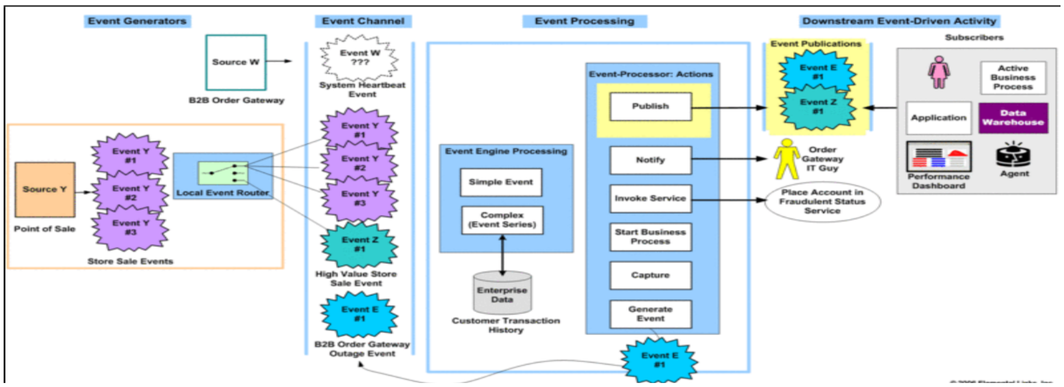


Figure 1: Event Driven Architecture [6]

A study about the history of CEP [9] traces its roots to university and company research groups in the late 90's which were involved in the areas of active databases, event driven simulation, networking and event processing in middleware. This explains partly why the CEP query languages are based on SQL syntax having been influenced by research on active databases. CEP products at that point did not generate much interest until the late 2000's when CEP was deployed as add-ons on SOA architectures and ESBs.

Databases are distinct from event queries used in CEP because of the latter's ability to continuously detect events as they happen rather than just acting on stored datasets. Event processing languages need to enable the possibility of joining several individual events together, so that their combined occurrences over time yield a complex event and complex events must contain the element of time, to track times when events occur.One study [8] introduced the need for revision of events in cases of erroneous data. In practice, there are a number of reasons requiring revisions in event stream processing. For example, an event was reported by mistake, but did not happen in reality (and the mistake was realized later); an event happened, but it was not reported (due to failure of either a sensor, or failure of the event transmission system); or an event was triggered and later revoked due to the transaction failure. Also very often streaming data sources contend with noise (e.g., financial data feeds, Web streaming data, updates etc.) resulting in erroneous inputs and, therefore, erroneous complex event results.

Through Complex Event Processing (CEP), companies and organizations can manage processes in close to real-time. It is however noted that due to the complexity of generic event processing frameworks offered by the industry, the configuration and setup of CEP applications are left to external experts who are more knowledgeable in complex event logic. A CEP application retrieves events for all noteworthy incidents in the business environment. In various parts of the application, event-pattern rules are applied on the incoming event stream to detect relevant patterns, e.g., an uptrend in application errors or execution delays. In response to such patterns, the CEP engine proactively intervenes in the business environment, e.g., by temporarily allocating additional resources, throttling uncritical business tasks or notifying system administrators [9].

Pattern matching is a key feature of all CEP technologies which involves finding subsets of data matching a given pattern and also relationships between those subsets. A study about CEP under uncertainty [25] explains the role pattern matching plays in allowing users to look beyond individual events and find specific collection sets.

Solution templates are proposed [14]to perform data mining procedures on historical data to identify event patterns besides real-time monitoring. The central concept of any template's event processing infrastructure is the event processing map, a predefined orchestration of event adapters and event services. Event adapters may be considered the actual interface to the underlying source system: Depending on their implementation, event adapters translate real-world actions (such as a user actually placing a bet in an online gambling platform) into event representations of a certain event type, and vice versa. Event services receive events from event adapters or other event services, process them based on implementation of specific logic, and respond back to the map.Detecting events is based on some considerations like, some events sharing time elements, the order of events, time bounds within events and detection of events of long time lags. To gain an insight into processes there is need to include the following components in CEP application, facilities (graphical or textual) for precise description of complex patterns of events, scalable performance, modular rules engines to detect complex patterns of events, facilities for defining and composing event pattern triggered rules for pattern abstraction.

2.6 Alogrithms and BNF for Esper Engine

The engine used for this project is the Esper engine [5] which makes use of indexes, a data structure that improves the speed of data retrieval operations. For sorted access it may prefer a binary tree index while a hash-based index is great for key lookups.For efficient matching of incoming events to statements the engine uses inverted indexes.Multi-version concurrency control is a concept used for variables and also for filters to allow concurrency and reduce locking.The match-recognize pattern matching functionality is built using nondeterministic finite automata (NFA).Query planning based on the analysis of expressions used in the where-clause is another technique used by the engine. The execution strategy may choose nested-loops versus merge joins [5].

The Esper Engine uses the EPL(Event Processing Language) to define statements stored in the engine. EPL queries are created and stored in the engine, and publish results to listeners as events are received by the engine or timer events occur that match the criteria specified in the query. Events can also be obtained from running EPL queries via the safeIterator and iterator methods that provide a pull-data API.The select clause in an EPL query specifies the event properties or events to retrieve. The from clause in an EPL query specifies the event stream definitions and stream names to use. The where clause in an EPL query specifies search conditions that specify which event or event combination to search for [5].

EPL queries can be simple queries or more complex queries.A simple select contains only a select clause and a single stream definition. Complex EPL queries can be built to feature a more elaborate select list utilizing expressions, may join multiple streams, may contain a where clause with search conditions and so on [5].Below is the syntax/BNF for the EPL language used by the Esper engine.

2.7 Event Processing Engines

An analysis of Event Processing Languages [3] revealed the need to shift from using general-purpose languages like C, Java, C++ e.t.c for CEP applications due to low-level complexities. Using such languages along with complexities like data structures and algorithms can only complicate the development process. The study provided a detailed analysis of existing CEP programming

High Level BNF

```
[annotations]
[expression_declarations]
[context    context_name]
[insert into   insert_into_def]
select  select_list
from              stream_def [as name]  [,
   stream_def [as name]]  [,...]
[where  search_conditions]
[group by  grouping_expression_list]
[having grouping_search_conditions]
[output output_specification]
[order by  order_by_expression_list]
[limit  num_rows]
```

Figure 2: BNF for Esper Engine(Source: Esper Tutorial)

Languages and platforms based on their expressivity and integration capabilities. Expressivity is measured by one of the following abilities, filtering streams by event type, processing a subset of events (windows), data extraction and aggregation of data over events, performing conjunctions and disjunctions, show temporal relations between events, showing causality of events, negation and counting of events, event instance selection and consumption to prevent reuse of events in pattern detection and integration of event data and non-event data (data from outside). Languages are also grouped into the categories of data stream query languages, composition-operator-based languages, production rule languages and logical formulas.

STREAM (Stanford Stream data Manager) is a language whose focus was to develop methods to manage and query data in data streams and was a result of a research project at Stanford University. The project also produced a CEP engine called STREAM and an Event Query Language called Continuous Query Language to query events. STREAM was a basis for other data stream languages like Esper and its querying syntax resembles SQL very strongly.

Borealis is a CEP engine developed at Brandeis University and MIT that uses a ”boxes and arrows” approach. Queries are described graphically with queries as boxes and streams as arrows connecting boxes. The approach was first used in an earlier engine, Aurora. The main difference between Borealis and stream languages is the focus on query evaluation that Borealis offers resulting in less abstract queries than STREAM.

Active Middleware Technology (AMiT) enables IBM middleware to become event-based. This technology is implemented in several products, most notably extending Web-Sphere Broker with CEP capabilities. As WebSphere is a commercial product, it is not freely available (requires registration .Basic events are declared with their attributes in event tags. Lifespans are windows defined by two events, an initiator and a terminator event. Lifespan types are therefore declared by referencing start and end event types. Whenever an event matching the initiator specification is detected, a new lifespan of this type is opened, and when an event matching the terminator specification is detected, the lifespan is closed. Complex events are called situations. A situation consists of at least one data attribute (it has to carry at least one kind of information), exactly one operator, and a lifespan type. Situations are only tried to be detected in lifespans of its type. A lifespan may be referenced by multiple situations

RuleCore is a CEP engine developed by Analog Software, building on research at the University of Skyde. As the name suggests, rules are the central concept of ruleCore. The ruleCore engine processes events using ECA (Event-Condition-Action) rules that consist of three parts: for every event (basic or complex), check a condition; if it is true, execute the action. ruleCore has two implementations; an open source variant called ruleCore, released under the terms of the GPL; as well as a commercial version called ruleCore CEP Server. RuleCore uses so-called detector trees for event detection. Leaf detector nodes (detector nodes without children) detect single events (they pick up events of their type). They are inactive until an event of their type is delivered to the rule (usually by entering the system, although exceptions are mentioned in the next paragraph), after which point they are always active. To detect complex events, a detector tree is built: the leaves detect simple events, and inner nodes detect complex events depending on whether its children detected events.

SASE+ is a CEP system developed at the University of Massachusetts, Amherst. It is an extension of the older SASE system. The system is designed for event streams with many events per time unit and also queries using large time windows, creating new issues regarding efficient query execution. The project's purpose is to devise techniques for high-performance querying of event streams, using a declarative, composition-operator-based language. Although SASE+ is an agile language and concentrates only on pattern matching on streaming data, the pattern matching properties of

SASE+ can be used in more general contexts.

Esper is an open-source CEP engine, developed by EsperTech Inc. and volunteers, released under the GNU General Public License (GPL v2). As stated on the official web site[5], it is designed for CEP and Event Stream Processing (ESP).There are two implementations of Esper, Esper for Java and NEsper for .NET. Both supply an API to access the engine features, such as deploying queries, sending events into the engine and retrieving events out of the engine, in their respective language. Events are objects in their respective language; for Esper, events can be instances of java.util.Map,org.w3c.dom.Node (Java representations of XML documents), or other Java objects. Regardless of the implementation language, queries are stated in a SQL-like language called Event Processing Language (EPL).

Cayuga is a research CEP engine developed at Cornell University. It sets itself apart from other engines in that it deliberately sacrifices expressivity for performance, targeting applications running large numbers of queries. It is free software, available under the terms of the BSD license. Cayuga uses an Event Query Language called Cayuga Event Language (CEL). While its syntax resembles SQL, like many data stream languages, it also offers patterns, although using a different approach compared to Esper, inspired by regular expressions.

Drools, also known as JBoss rules, is a production-quality business rule management system, including a production rule engine. It is free software, released under the Apache License. The Drools engine is implemented in Java, as is JBoss, and is also controlled using that language. Initialization of the engine and deployment of rules is implemented in Java. Also, as unusual for production rule engines, rules never fire by themselves, but are issued to do so by the Java program that controls the engine. In addition, Drools can be extended by defining so-called Domain Specific Languages. These are languages that may have a different syntax than the standard Drools syntax to write queries in. Rules in Domain Specific Languages are then translated into the Drools language when inserted into the engine.

XChangeEQ is a research Event Query Language. It is developed at the University of Munich and designed for automated reasoning on the Semantic Web. XChangeEQ introduces a new style of event querying. It separates event query features into four so-called dimensions: Data extraction, event composition, temporal relationships, and event accumulation. Most operators belong to exactly one of these dimensions. This was done to define clear semantics. As XChangeEQ is designed for use on the Web, it works best at processing tree-structured events, such as XML messages. Queries are generally structured like the XML representations of the events queried. For querying simple events, it embeds the Xcerpt language. Xcerpt queries apply patterns to XML documents, similar to templates. Change is a reactive programming language. Using Event-Condition-Action rules, it allows Web sites to react to changes at other Web sites, for example by updating its own data.

TelegraphCQ, from the University of California at Berkeley, is designed to provide event processing capabilities alongside relational database management capabilities by utilizing the PostgreSQL open source code base. The existing architecture of PostgreSQL is modified to allow for continuous queries over streaming data. Several components of the PostgreSQL engine underwent very little modification, while others were significantly changed. The most significant component of the TelegraphCQ system is the ”wrapper,” which allows for data to be pushed or pulled into the Telegraph processing engine, and custom wrappers allow for data to be obtained from any data source .

BEA Systems in 2007 introduced of their WebLogic Real Time and WebLogic Event Server systems. More specifically, their Event Server technology is a focus on event-driven service oriented architecture which provides a response to events in real-time. As part of the package, they provide a complete event processing and event-driven service-oriented architecture infrastructure that supports high-volume, real-time, complex, event-driven applications. This is one of the few commercial offerings of a complete, integrated solution for event processing and service-oriented architectures.

Truviso is a commercial event processing engine that is based on the research toward the Telegraph CQ project at UC Berkeley. The”claim to fame” for Truviso is that it supports a fully functional SQL, and integrates PostgreSQL relational database alongside a stream processing engine. The integration of PostgreSQL leads to other aspects of the Truviso system. The queries are simply standard SQL with extensions that add functionality for time windows and event processing. Carried over from PostgreSQL are user-defined functions, as well as JDBC and ODBC interfaces. In addition, the use of an integrated relational database allows for easy caching, persistence, and archival of data streams, as well as queries that include not only real-time data, but also the historical data [7].

3 CHAPTER THREE: Methodology

In this chapter, the methods used to acomplish the project are discussed. Section 3.1 explains how requirements analysis was performed to identify the key stakeholders and determine the functional and non functional requirements for the investigation tool. The tool's design is discussed in section 3.2 along with UML representations of the data and process flows. Section 3.3 discusses the implementation methods used while section 3.4 presents testing and evaluation procedures used to assess the system functionality.

3.1 Requirements Gathering

3.1.1 Interviews

Interviews were conducted with investigative officers at Police headquarters, Jinja Road Police station and the Special Investigation Unit, Kireka. This was carried out to ascertain the nature of financial crime cases, the investigation processes involved and the criteria used to zero down on a suspect. Telephone interviews were also conducted in situations where physical access was challenging.A total of nine officers were interviewed who included the Director of Research at Police Headquarters(Deputy IGP),IT Manager,IT officers,CID(Criminal Investigations Department) Officers (2) (Police Headquaters),CID officer(Special Unit of Investigations,Kireka),CID officer(Kibuli) and two(2) CID officers at Jinja Road Eastern Region CID offices.

The questions asked were related to the following:-

(i) Type of evidence gathered and used in such investigations.

(ii) The patterns which are common in most cases and which the system will base upon.

(iii) Any existing systems used for crime investigation and formats of existing crime records.

A detailed list of interview questions asked and a request for permission form are attached in Appendix B and Appendix A respectively.

3.1.2 Existing Literature

Existing literature on Complex Event Processing was read to determine the best design approach for the project and to discover the benefits of using the CEP model.Implementations of the Esper engine in particular were studied to provide a benchmark for the project implementation.

The Annual Police report was also consulted to acquire an in depth understanding of the impact of crimes to society and government. Case statements some handwritten were also analyzed and these provided guidance on data to capture and track in the system.

3.1.3 Functional Requirements

(i) The tool should provide user-friendly interfaces both for data capture and information display to the user.

(ii) The tool should generate reports on PDF saved to the local disk and a web page view.

(iii) The tool should enable extraction of crime data from an existing database related to the case file number supplied by the user.

(iv) The tool should be able to process incoming crime data and display information based on user input values.

Refer to Appendix C for detailed functional requirements.

3.1.4 Use Case

This was designed based on requirements gathered during the interview sessions held with police investigators and administrators. The figure below shows user and system interactions.

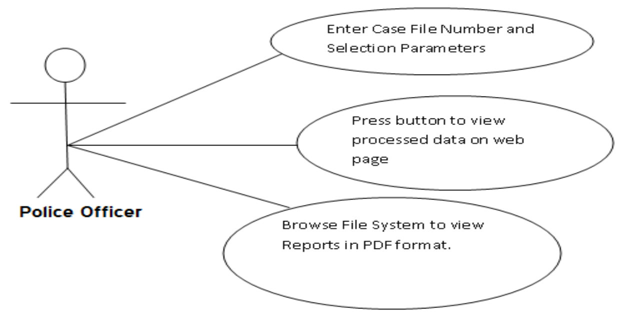


Figure 3: Police Officer Use Case

3.1.5 Non-Functional Requirements

(i) The tool should be easy to use by Police Officers with little or no experience in computer technology and basic applications.

(ii) The tool should simplify the investigation process by identifying records matching the defined pattern /rules.

For more details refer to Appendix C.

3.2 System Design

This section describes the system environment, interactions, requirements, architecture and input/output formats as well as processing logic. The system was modeled using UML (Unified Modeling Language) tools based on the object oriented design approach.

3.2.1 Design Overview

The design includes an existing System/Database that receives instructions from the Application GUI to extract data and sends the data to a CEP engine running which processes the data displays information back to the user.

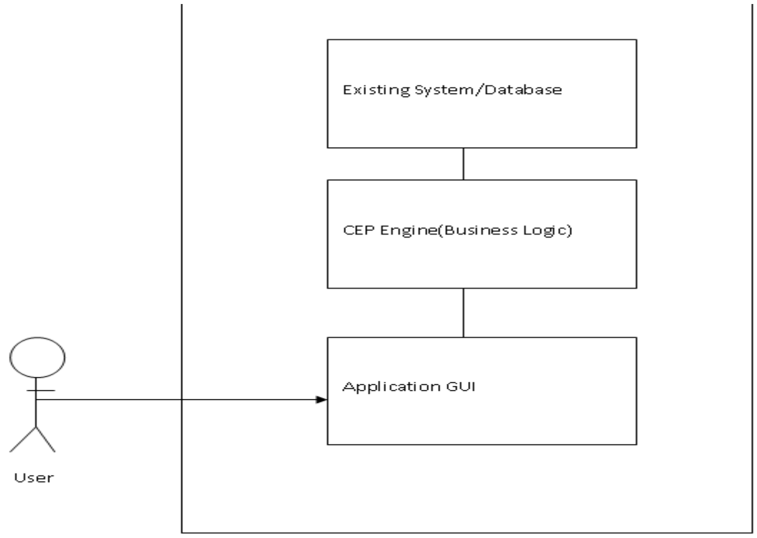


Figure 4: Design Overview

3.2.2 Logical Design

The main entities in the logical design are listed below and their associations are shown in the figure.

(i) Investigator/Police Officer : The main actor.

(ii) Application: The tool the investigator interacts with.

(iii) CEP engine : Initialized by the application.

(iv) Information: Processed data returned by the engine listeners.

(v) Input Data: Data sent to the CEP Engine from the database.

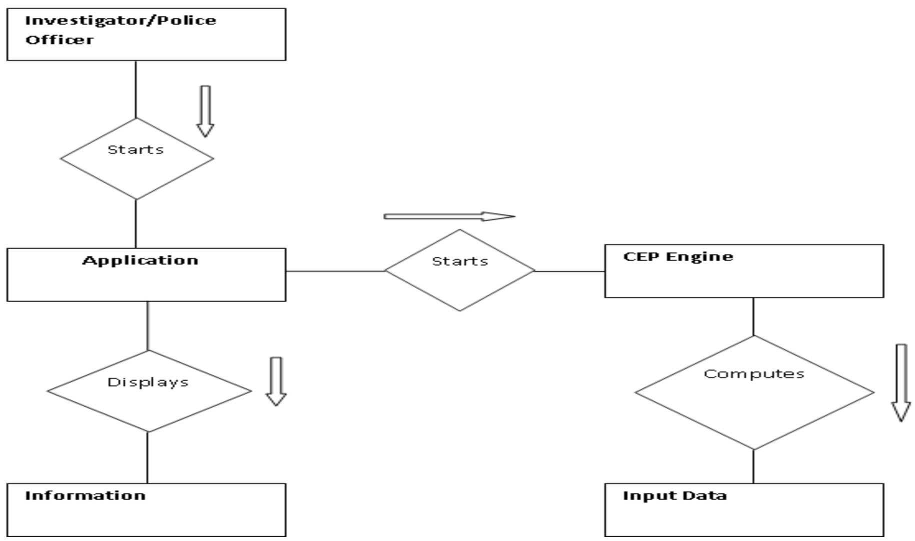


Figure 5: Logical Design

3.2.3 Application Architectural Overview

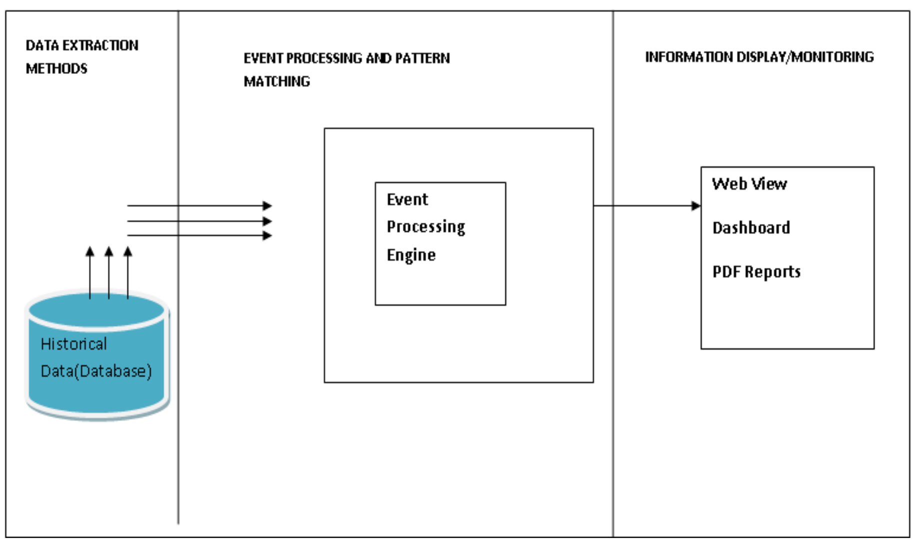


Figure 6: Application Architecture

3.2.4 Data Flow Diagram

The process was designed to start from the web client (user) making a request for records from an existing database. The returned data is sent to the processing engine and checked against the defined pattern. Based on the computation a response is displayed to the client in form of matching events and match percentages.

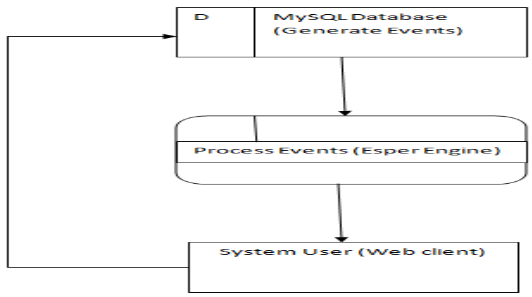


Figure 7: Data flow Diagram

3.3 Implementation

The tool was developed as a web-based application to enable access by users throughout the Police network and to provide capabilities for access by mobile devices as well. Other reasons for choosing a web application include; OS platform independence, ease of maintenance and limited upgrades from the client side. It was built on the Java EE platform using the glassfish application server and the JSF 2.1 framework [16].

3.3.1 Tools and Platforms

The investigation tool was developed using a range of tools and platforms reflecting the MVC(Model View Controller) paradigm to achieve efficiency and easy management. Some of tools used are;

(i) Netbeans IDE 7.3.1/JSF 2.2 Framework/Java

(ii) MySQL WorkBench 5.2

(iii) PrimeFaces 3.5 library

(iv) Esper Engine 4.8.0 library

(v) GlassFish Server 3

This was achieved using the JPA API [6] and its query language to return data matching a given select statement which included a WHERE clause containing the Case File Number. The Case File Number is received from the Facelet(JSF) page (XHTML) initiated by the user and using the Expression Language[6] is linked to a Managed Bean [6] containing the Entity Manager instance.

The Managed Bean then interacts with an Entity Class which queries the database and returns values to the bean . These values are written as CSV format files to the application class path ready to be sent to the engine for processing.The figure below shows a sample JPA query to extract data.

```
// Named Query
@NamedQuery(name = "Phonelogs.findByCasecode",
query = "SELECT pd FROM Phonelogs pd WHERE
pd.caseCode.caseCode =:caseCode")
//
@PersistenceUnit
   EntityManagerFactory emf;
//
List<Phonelogs> itemList = emf.createEntityManager().

createNamedQuery("Phonelogs.findByCasecode").setPar
ameter("caseCode", caseNo).getResultList();
```

Figure 8: Sample JPA Query

3.3.2 Detecting Pattern in Events

Using the CSVAdapter class from the Esper [5] library, CSV files are read from the class path sent to the engine for processing .The fields in the CSV files are mapped to POJO classes for each of the entities being analyzed using Hash Map configurations. The processing is based upon SQL-like statements defines using the Event Processing Language (EPL) [3] as shown in sample below.

```
//
String expression2 = "select "
        + "bdate,transfertype,"
        + "debitacctno,debitamount,"
        + "creditbank,creditacct,bankstaffid,status,limits from "
        + " Boubean.win:length(1)"
        + " where (debitamount " + limsymb + " limits) or (status " + saccstat + " " + "'" + accstat + "'" + ") "
        + "";

//
```

Figure 9: Sample EPL Statement

Listeners are then defined to store records matching the defined pattern or to hold returned results from the EPL queries. These are then written to text files in a local location on the workstation.

3.3.3 Reports and Web View

Events processed by the engine are read from the text files periodically and displayed to a client XHTML web page using the Expression Language (EL) [16] which interacts with the JSF [16] managed bean to return computed results. The EL allows page authors to use simple expressions to dynamically access data from JavaBeans components.

The PrimeFaces library [18] was used to provide a look and feel to the interface. PrimeFaces is a lightweight library that hides complexities from user while providing extended capabilities to the default XHTML controls as well as quick software development. Records matching the pattern are displayed in tabular format along with the percentage matching rate computed. Another output in PDF format is also created to the local disk.The snippet below shows how data was returned to the JSF page for the user to study.

```
<h:outputText value="Percentage Pattern Match :" />
<h:outputText value="#{mainBean.totp} " style="color: #00f" id="tot2"/>
<!--<p:poll interval="25" update="tot2" />-->
<h:outputText value=" (" />
<h:outputText value="of" />
<h:outputText value=") " />
<h:outputText value="#{mainBean.totn}" style="color: #00f" />
<h:outputText value="----->(" />
<h:outputText value="#{mainBean.totperc}" style="color: #00f" />
<h:outputText value="%" />
<h:outputText value=") " />

<br></br>

<br></br>

<p:dataTable var="rep8" value="#{mainBean.pats77}"  >

    <p:column>
        <f:facet name="header">
            <h:outputText value="Filename" />
        </f:facet>
        <h:outputText value="#{rep8.filename}" />
    </p:column>
```

Figure 10: Sample Facelet code

3.3.4 Prototyping

A prototype was developed for the crime investigation tool based on the specification requirements document and the main interface is shown below.

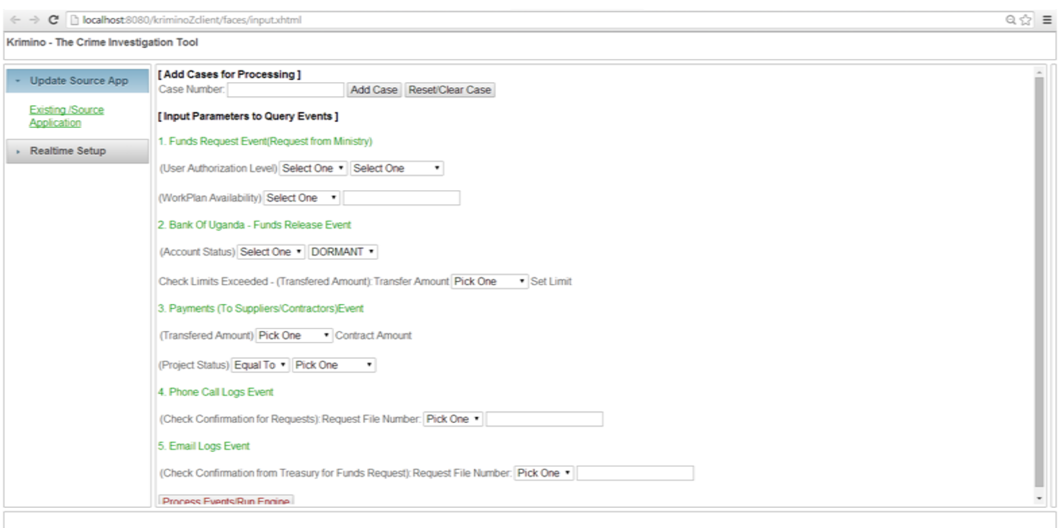


Figure 11: Krimino Tool Prototype

The above interface contains controls for the user to customize the pattern and launch the processing and display processes.

3.4 System Testing and Evaluation

The tool was tested by two crime investigators and below are guidelines and parameters used to perform the tests.

Use Case ID	Case Description	Test Case
UC-1	Use Case File Generation from Database	TC-1
UC-2	Use Case Record processing by Esper Engine	TC-2
UC-3	Use Case Display of results to web page	TC-3
UC-4	Use Case Display of results to PDF file	TC-4
UC-5	Use Case Launch Application	TC-5
UC-6	Use Case Exit Application	TC-6

Figure 12: Test Cases

More details can be found in Appendix D and the test results are discussed in the next chapter.

4 CHAPTER FOUR: Presentation of Tool and Results

In this chapter we discuss the crime pattern that was built in the tool for purposes of proof of concept, the operation and functionality of the tool and we present the test results based on the test cases in the previous chapter and in Appendix D.

4.1 Crime Pattern and Parameters

The pattern is made of five major events (Funds Request, Funds Release, Payments, Phone Logs, Email Logs) which together represent the general pattern used.Any violations detected in any of the five events leads to a degree of pattern match.

4.1.1 Funds Request By Ministry

Before any amount of money is disbursed to any government entity there must be a budget designed at the Ministry of Finance (government) stipulating funds available for different activities. These funds are availed on a periodic basis to the entity most commonly quarterly.

Likewise the government entity requesting for funds must provide a detailed budget to the Ministry of Finance if it is another ministry or through the Ministry of local government for Local Governments. This work plan provides a form of accountability to government for the released amounts.Requests for release of funds must duly be authorized by a designated official permitted to act on behalf of the entity.

Therefore for this event the parameters used are;

(i) Identification of authorization level of the requesting officer (Must be the Permanent Sectary or authorized replacement)

(ii) Checking the availability of a workplan for a given request.

4.1.2 Funds Release By Central Bank (BOU)

This is the process of releasing funds to requesting entity on a periodic basis as per the budget arrangements. It involves transferring funds from the government account at the Central Bank (Bank of Uganda) to the respective accounts via electronic payment systems (EFTs).

The parameters used here are;

(i) Checking that the debited account is not in a dormant status(Dormant Accounts not to be debited).

(ii) Checking that the disbursed amount does not exceed the set limit for daily releases to ministries.

4.1.3 Emails and Phone Logs Events

These are closely linked to the Funds Release event given that before money is released a confirmation email must be sent from the treasury endorsing the request and a phone call must be made from the Central Bank to the Permanent Secretary confirming the authorizer.

Therefore for these two events,the parameters are ;

(i) Availability of a confirmation email record from the treasury.

(ii) Availability of confirmation phone call record from Bank of Uganda to the Permanent Secretary.

4.1.4 Payments Event

This is a record of how the funds were spent and may include payment details, account statements from the entity and suppliers accounts and also account opening details.A report of activities completed periodically must be availed to get value for money. Here the work done is verified against the budget or work plan submitted at the point of request for funds.

The parameters include;

(i) Checking if amount paid exceeds agreed amount in contract.

(ii) Checking that work being paid for was completed and not pending.

4.2 Tool Functionality/Operation

4.2.1 Client Request

In the single mode one case file number is added for processing while in the multi-mode more than one case is selected or added for processing.Selections are made at the different event sections defining conditions for returned records and the request is submitted.

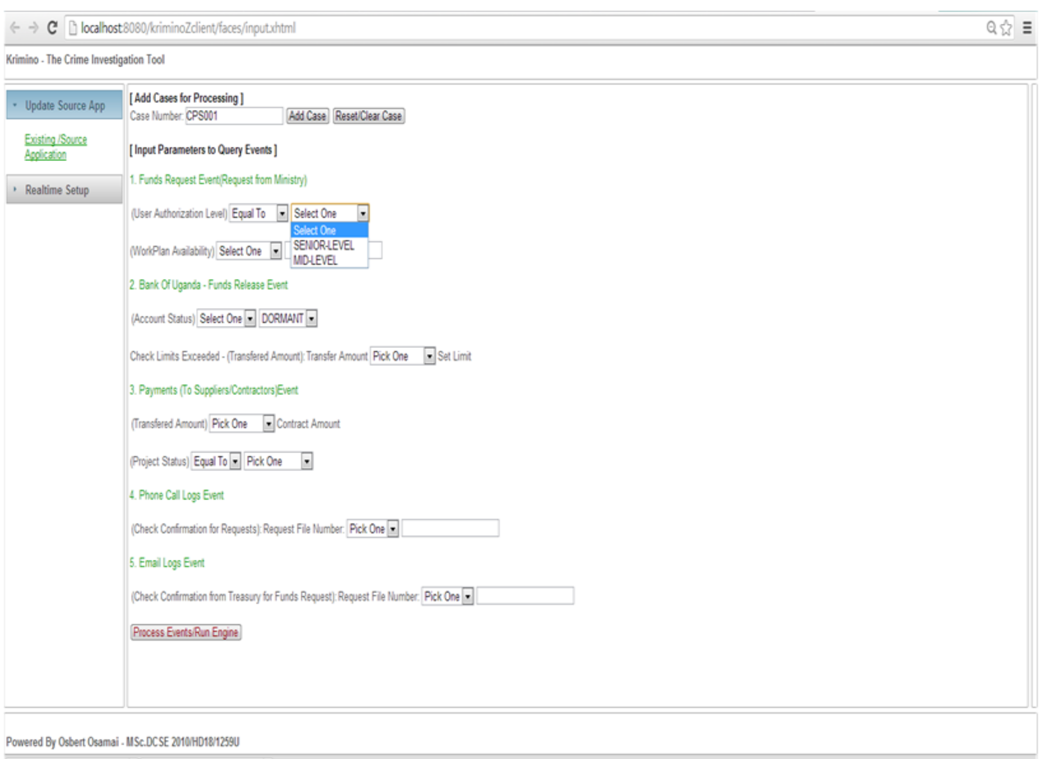


Figure 13: Case Input and Selection screen

4.2.2 System Response

The output is form of a single web view and PDF format files depending on the input mode (single or multi-mode).For the multi-mode a PDF file is created for the same number of cases file numbers processed and a single web view for one of the cases as below;

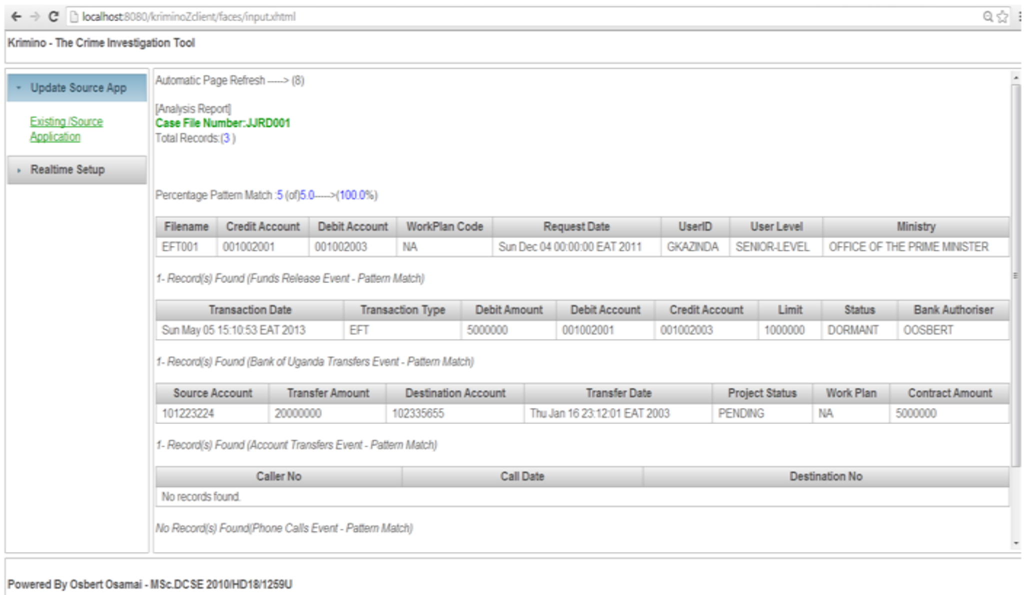


Figure 14: Web View Display

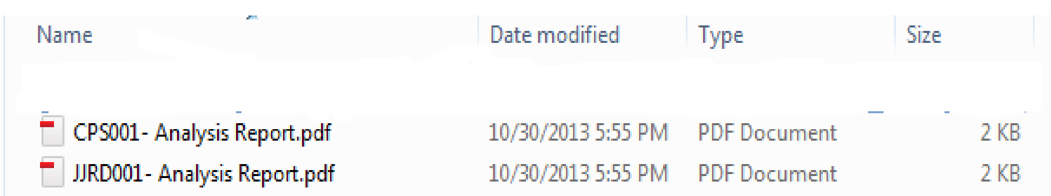
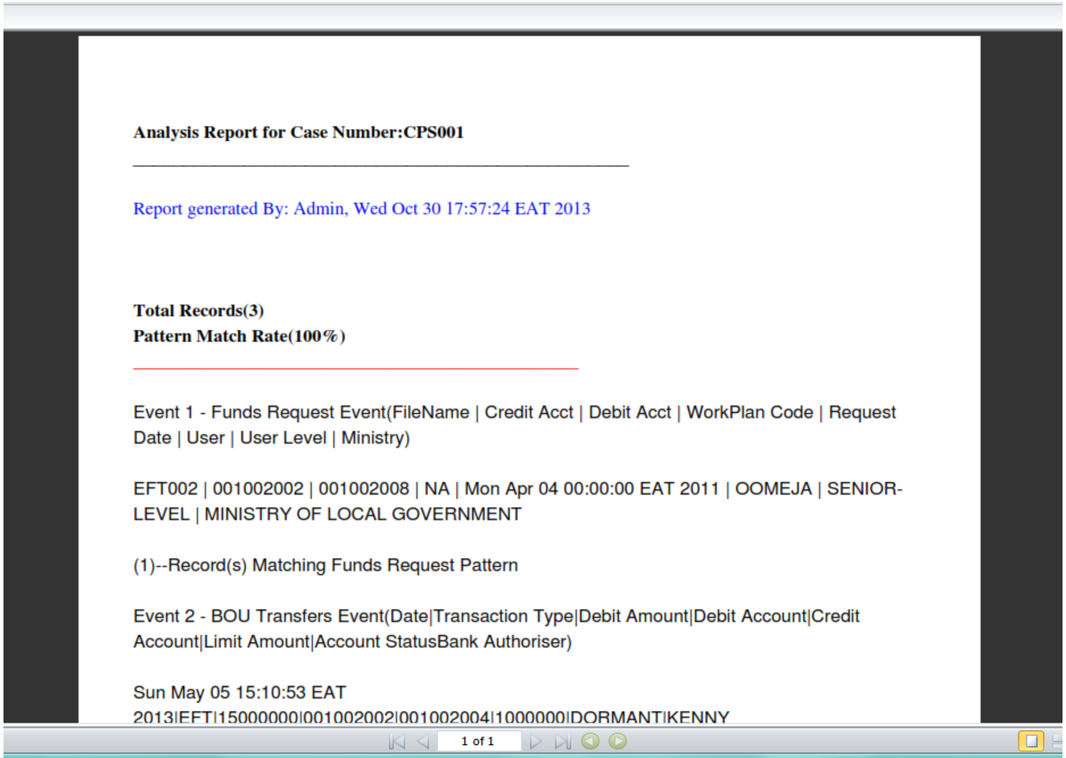


Figure 15: Generated PDF files for multiple Case Files


Analysis Report for Case Number:CPS001

Report generated By: Admin, Wed Oct 30 17:57:24 EAT 2013

Total Records(3)
Pattern Match Rate(100%)

Event 1 - Funds Request Event(FileName | Credit Acct | Debit Acct | WorkPlan Code | Request Date | User | User Level | Ministry)

EFT002 | 001002002 | 001002008 | NA | Mon Apr 04 00:00:00 EAT 2011 | OOMEJA | SENIOR-LEVEL | MINISTRY OF LOCAL GOVERNMENT

(1)--Record(s) Matching Funds Request Pattern

Event 2 - BOU Transfers Event(Date|Transaction Type|Debit Amount|Debit Account|Credit Account|Limit Amount|Account StatusBank Authoriser)

Sun May 05 15:10:53 EAT 2013|EFT|15000000|001002002|001002004|1000000|DORMANT|KENNY

1 of 1

Figure 16: PDF File Display

4.2.3 Interpretation of Test Results

This section discusses the test results from the tests that were carried out following the guidelines defined in Chapter three and Appendix D. The tests were carried out by the developer/analyst along with two(2) Police officers from the Crime Investigation Department. Two types of tests were conducted in the categories of functional tests and Human Computer Interaction tests with 6 and 5 test cases respectively.

4.2.4 Functional Tests

These were carried out to determine if the tool was developed in accordance with Software Requirements Specification document in Appendix C. The test cases used are as shown in the figure below.

Use Case ID	Case Description	Test Case
UC-1	Use Case File Generation from Database	TC-1
UC-2	Use Case Record processing by Esper Engine	TC-2
UC-3	Use Case Display of results to web page	TC-3
UC-4	Use Case Display of results to PDF file	TC-4
UC-5	Use Case Launch Application	TC-5
UC-6	Use Case Exit Application	TC-6

Figure 17: Functional Test Cases

The following is a graphical representation of the functional tests results for the tests carried out by the two officers and the developer/analyst.

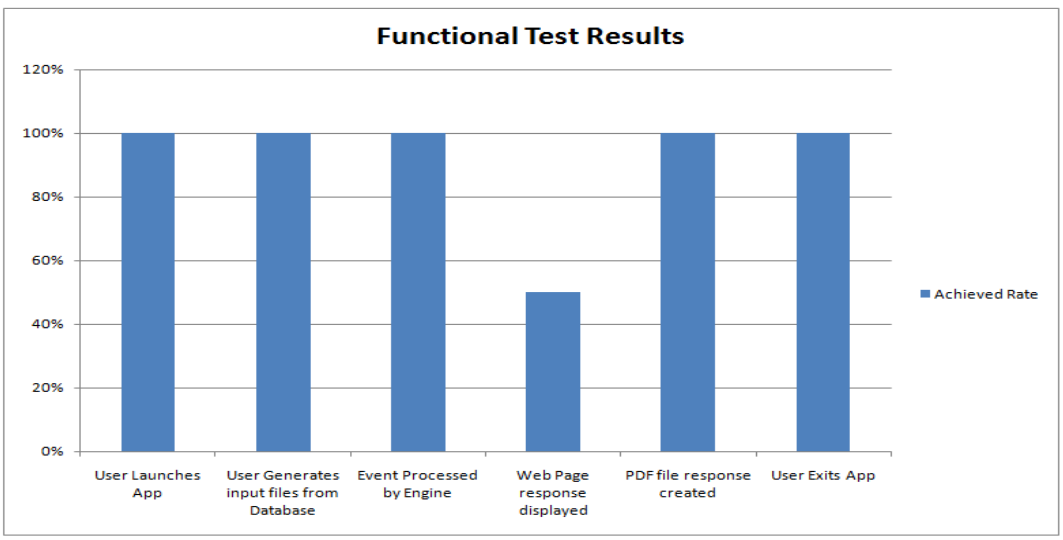


Figure 18: Functional Test Results

The tests were generally succesfull save for the web page display for the real-time mode of the tool because of a bug with the listener class of the esper library.This meant that some results were duplicated before the next batch could be displayed.However the web display for the non real-time mode was succesfull.

4.2.5 Human-Computer Interaction Tests

These were carried to ascertain the level of computer-literacy as well as the user-friendliness of the tool. The test cases include; Computer-literacy levels,exposure to web applications,reading abilities, navigation around the application and ability to interprete responses or output reports. The following chart shows the results for the five(5) test cases.

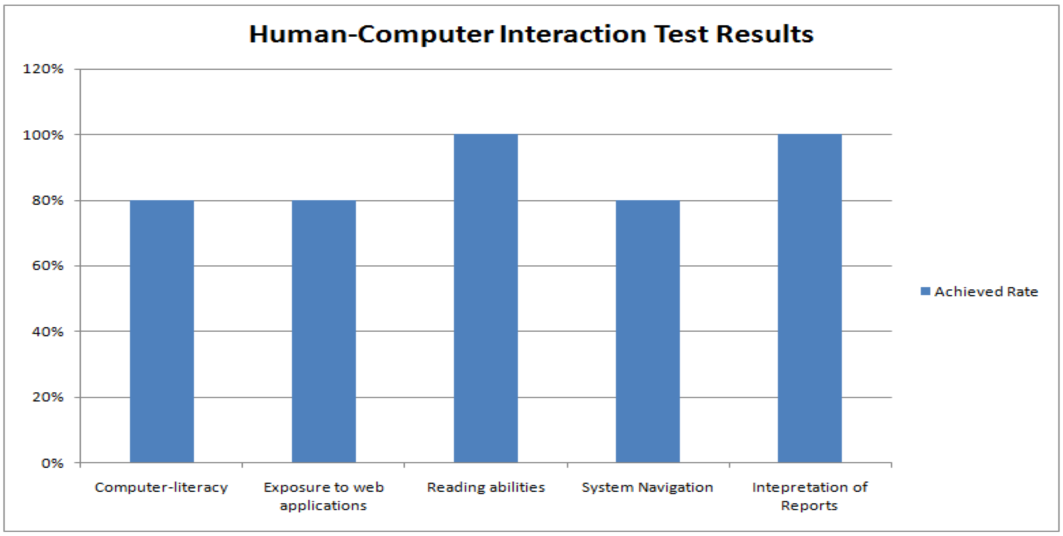


Figure 19: Human-Computer Interaction Test Results

From the above results, it was noted that much as all the officers could read and write and also easily interprete the system reports, they were not so comfortable with web applications and navigating the web pages.All in all the tests presented a high degree of success given the officers' level of computer-literacy.

5 CHAPTER FIVE: Conclusion

Crime investigations are an integral part of any law enforcement agency and if mismanaged can effect negatively on both government and the populace.By deploying techniques such as Complex Event Processing, Crime Investigation Systems can perform better and faster in analyzing data and discovering hidden patterns.Such tools will enable speedy investigations,overcome the labour shortage issue at the force and also improve public faith in the agencies and government at large.

Based on the tests carried out during the project the crime investigation tool built was able to process records and accurately report events matching preset rules.The project was successful although the input data formats were of a fixed type (CSV) and some few challenges.It is expected that with further developments and consultations the tool will be improved to cover all crimes and allow different file input formats as well as real-time monitoring.

5.1 Challenges

The following challenges were faced during developement;

(i) Enabling the realtime mode given the complex architecture.

(ii) Representing crime data in a structured format to feed the engine.

(iii) Learning the Esper engine API which was time-consuming.

(iv) Interviewing some top Police officials who claimed to be busy.

5.2 Contribution

(i) To computing, using CEP techniques and the esper engine to solve a problem in crime investigations which proved more effective than data mining techniques.

(ii) To citizens, enabling quick crime anlaysis leading to a good turn around time as well as managing small work force using technological initiatives.

(iii) To government and the Police,public confidence will be restored.

5.3 Recommendations

The following recommendations were made ;

(i) Develop a well organized database to supply the tool engine with data and deploy it on a country-wide Police network.

(ii) Agree and create case file numbers unique to every Police Station in the country.

(iii) Agree and arrange channels linking the system with other stakeholders' systems to provide straight-through communication e.g to the judiciary.

(iv) The Uganda Police needs to train investigators in computer-enabled investigation and equip them with neccesary ICT skills.

5.4 Future Work

To improve the performance and usefullness of the tool, the following are suggested ;

(i) Extending the tool to detect new emerging patterns that can help in crime prevention based on available records.

(ii) Include a criminal monitoring module for continuous surveillance of criminals/areas based on known patterns.

(iii) Extend the client module to support mobiles to enable remote requests.

References

[1] D. Anicic, S. Rudolph, P. Fodor, and N. Stojanovic. Retractable complex event processing and stream reasoning. In *Proceedings of the 5th international conference on Rule-based reasoning, programming, and applications*, RuleML'2011, pages 122–137, Berlin, Heidelberg, 2011. Springer-Verlag.

[2] K. Ask. *Criminal investigation : motivation, emotion and cognition in the processing of evidence*. PhD thesis, Goteborg University, Sweden, 2006.

[3] H.-L. Bui. Survey and Comparison of Event Query Languages Using Practical Examples . Master's thesis, Institut fur Informatik, Munich,Germany, 2009.

[4] I. Charles M. Alifano, Worldwide Law Enforcement Consulting Group. Fundamentals of criminal investigation. `Available:http://www.worldwidelawenforcement.com/docs/FUNDAMENTALSOFCRIMINALINVESTIGAT%IONS.pdf`. Accessed: 2012-06-25.

[5] Codehaus. Esper reference tutorial. `Available:http://esper.codehaus.org/tutorials/tutorial/tutorial.html`. Accessed: 2012-02-26.

[6] P. Dekkers. Cordy's - Simplifying Business. Master's thesis, Radboud Universiteit Nijmegen, The Netherlands, 2007.

[7] N. Dindar, B. Güç, P. Lau, A. Ozal, M. Soner, and N. Tatbul. Dejavu: declarative pattern matching over live and archived streams of events. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, SIGMOD '09, pages 1023–1026, New York, NY, USA, 2009. ACM.

[8] A. A. Kulkarni. Arcade - abstraction and realization of complex event scenarios using dynamic rule creation. In *Proceedings of the 5th ACM international conference on Distributed event-based system*, DEBS '11, pages 23–28, New York, NY, USA, 2011. ACM.

[9] D. Luckham. A short history of complex event processing part3 [online]. `Available:http://www.complexevents.com/2008/12/18/a-short-history-of-co%`

mplex-event-processing-part-3-the-formative-years/. Accessed: 2011-05-09.

[10] D. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems.* Addison-Wesley, Reading, MA, USA, 2002.

[11] D. Luckham. Complex event processing in financial services. In *Journal of Financial Services Technology, The, Vol. 2, No. 1*, pages 13–19, Sydney, Australia, 2008. Financial Standard.

[12] S. M.A.P. Chamikara, Y.P.R. D. Yapa and J. Gunathilake. Sl-securenet: Intelligent policing using data mining techniques. In *International Journal of Soft Computing and Engineering (IJSCE)*, 2231-2307, Volume-2, Issue-1, 2012.

[13] S. Myles. Criminal Investigation Case Management [Online] . Available:http://www.emich.edu/cerns/downloads/papers/PoliceStaff/Unsorted/CriminalInv%estigation/CaseManagement.pdf. Accessed: 2011-09-30.

[14] H. Obweger, J. Schiefer, M. Suntinger, F. Breier, and R. Thullner. Complex event processing off the shelf - rapid development of event-driven applications with solution templates. In *Control Automation (MED), 2011 19th Mediterranean Conference on*, pages 631–638, June 2011.

[15] D. O. P. Okwangale Fredrick R. Survey of data mining methods for crime analysis and visualization [online]. Available:http://cit.mak.ac.ug/iccir/downloads/SREC_06/Okwangale%20Fredrick%20R_06.pdf%. Accessed: 2011-11-22.

[16] Oracle. The java ee 6tutorial [online]. Available:http://docs.oracle.com/javaee/6/tutorial/doc/. Accessed: 2012-03-09.

[17] U. Police. Annual crime and road safety report 2010 [Online] . Available:http://www.upf.go.ug/attachments/article/5/Annual_Crime_Report_2010.pdf. Accessed: 2011-03-20.

[18] PrimeTek. Prime faces users guide 3.5 [online]. Available:https://primefaces.googlecode.com/files/indexed_primefaces_users_guide_3_5.p%df. Accessed: 2012-03-019.

[19] J. Schroeder, J. J. Xu, H. Chen, and M. Chau. Automated criminal link analysis based on domain knowledge. *JASIST*, 58(6):842–855, 2007.

[20] N. P. Schultz-Møller, M. Migliavacca, and P. Pietzuch. Distributed complex event processing with query rewriting. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, DEBS '09, pages 4:1–4:12, New York, NY, USA, 2009. ACM.

[21] F. Technologies. Accelerating complex event processing with memory-centric database (mcdb)[online]. `Available:http://www.fedcentric.com/collateral/MCDB_Solutions.pdf`. Accessed: 2011-05-09.

[22] F. Tushabe. Computer forensics for cyber based crimes. Master's thesis, Makerere University, Kampala,Uganda, 2004.

[23] S. W. van den Braak, H. van Oostendorp, H. Prakken, and G. Vreeswijk. Representing narrative and testimonial knowledge in sense-making software for crime analysis. In *JURIX*, pages 160–169, 2008.

[24] K. H. Vellani. Crime analysis for problem solving security professionals in 25 small step [online]. `Available:http://www.popcenter.org/library/reading/pdfs/crimeanalysis25steps.pdf`. Accessed: 2011-08-20.

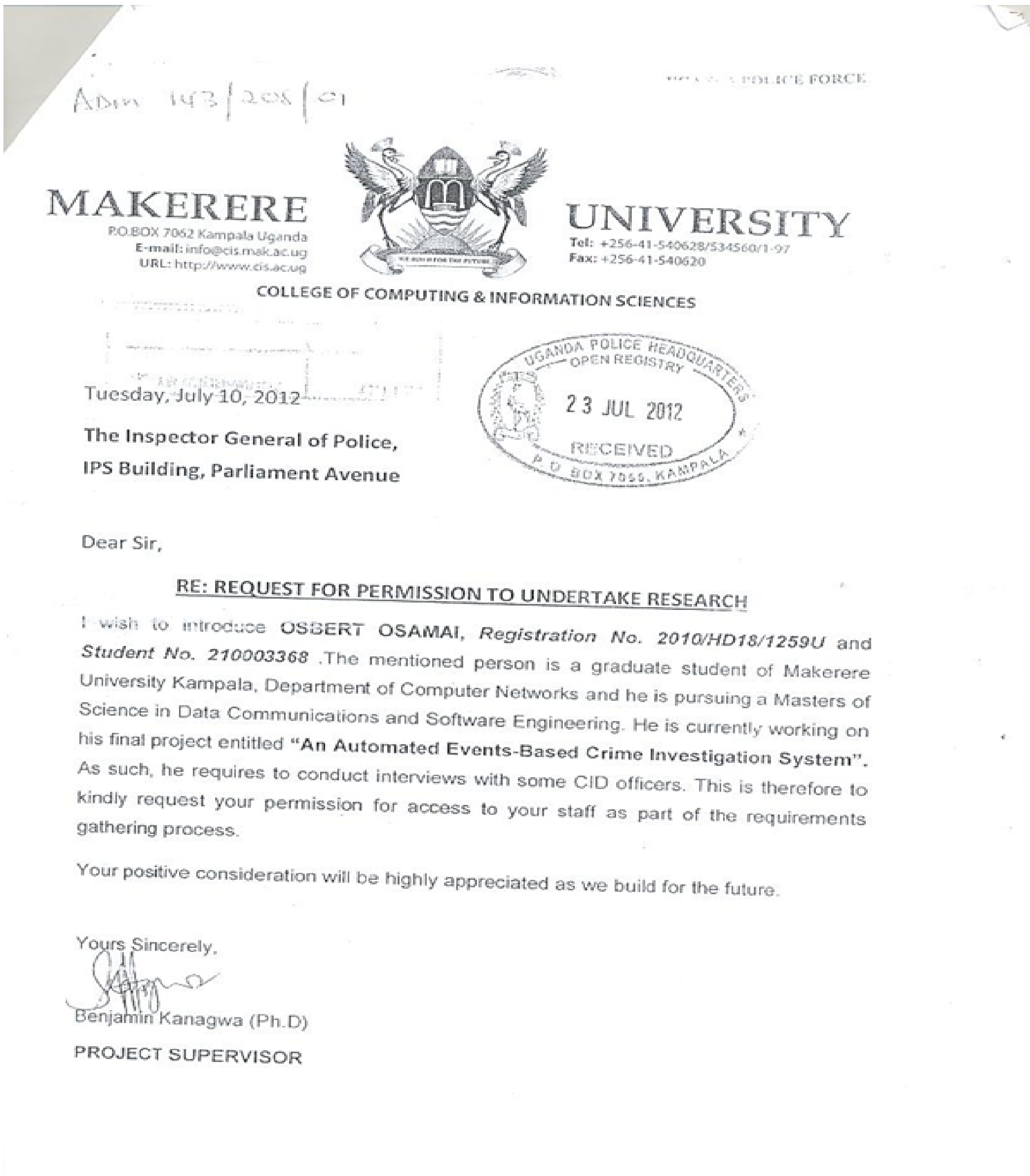
[25] S. Wasserkrug, A. Gal, O. Etzion, and Y. Turchin. Complex event processing over uncertain data. In *Proceedings of the second international conference on Distributed event-based systems*, DEBS '08, pages 253–264, New York, NY, USA, 2008. ACM.

[26] C. Westphal. Anatomy of a financial crime [online]. `Available:https://support.visualanalytics.com/technicalarticles/whitepaper/pdf/anatomy%_financial_crime.pdf`. Accessed: 2012-01-20.

[27] J. Xu and H. Chen. Criminal network analysis and visualization. *Commun. ACM*, 48(6):100–107, June 2005.

6 Appendices

6.1 Appendix A : Request for Permission and Approval Letters



ADM 143/208/01

POLICE FORCE

MAKERERE UNIVERSITY

P.O.BOX 7062 Kampala Uganda
E-mail: info@cis.mak.ac.ug
URL: http://www.cis.ac.ug

Tel: +256-41-540628/534560/1-97
Fax: +256-41-540620

COLLEGE OF COMPUTING & INFORMATION SCIENCES

UGANDA POLICE HEADQUARTERS
OPEN REGISTRY
23 JUL 2012
RECEIVED
P.O. BOX 7055, KAMPALA

Tuesday, July 10, 2012

The Inspector General of Police,
IPS Building, Parliament Avenue

Dear Sir,

RE: REQUEST FOR PERMISSION TO UNDERTAKE RESEARCH

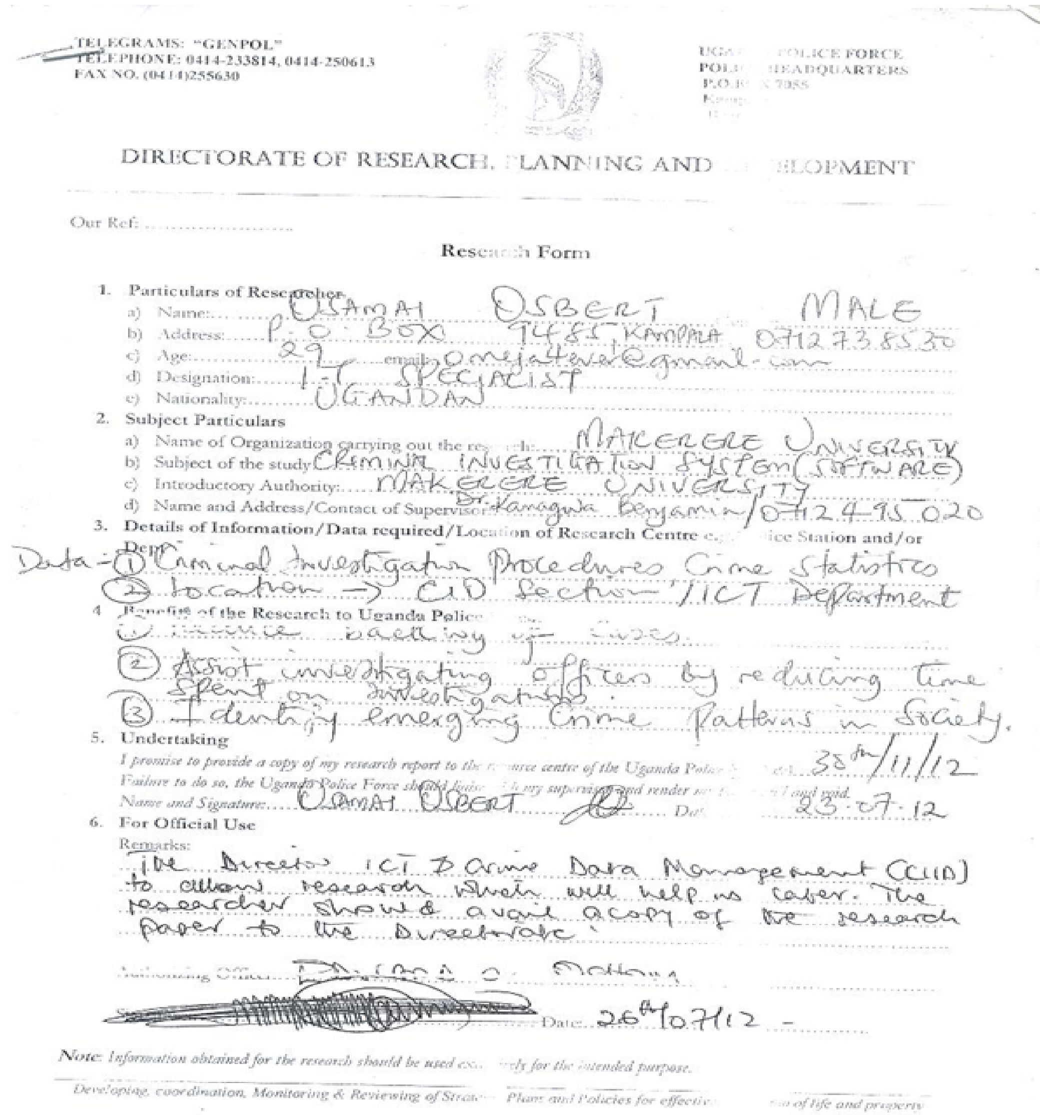
I wish to introduce **OSBERT OSAMAI**, ***Registration No. 2010/HD18/1259U*** and ***Student No. 210003368*** .The mentioned person is a graduate student of Makerere University Kampala, Department of Computer Networks and he is pursuing a Masters of Science in Data Communications and Software Engineering. He is currently working on his final project entitled **"An Automated Events-Based Crime Investigation System".** As such, he requires to conduct interviews with some CID officers. This is therefore to kindly request your permission for access to your staff as part of the requirements gathering process.

Your positive consideration will be highly appreciated as we build for the future.

Yours Sincerely,

Benjamin Kanagwa (Ph.D)
PROJECT SUPERVISOR

Figure 20: Request for Permission Letter


TELEGRAMS: "GENPOL"
TELEPHONE: 0414-233814, 0414-250613
FAX NO. (0414)255630

UGA[illegible] POLICE FORCE
POLI[illegible] HEADQUARTERS
P.O.[illegible] 7055

DIRECTORATE OF RESEARCH, [illegible]LANNING AND [illegible]ELOPMENT

Our Ref: ....

Resea[illegible]h Form

1. **Particulars of Researcher**
   a) Name: OSAMAI OSBERT MALE
   b) Address: P.O. BOX 9485 KAMPALA 0712738530
   c) Age: 29 email: omeja4ever@gmail.com
   d) Designation: I.T SPECIALIST
   e) Nationality: UGANDAN
2. **Subject Particulars**
   a) Name of Organization carrying out the res[illegible]ch: MAKERERE UNIVERSITY
   b) Subject of the study: CRIMINAL INVESTIGATION SYSTEM (SOFTWARE)
   c) Introductory Authority: MAKERERE UNIVERSITY
   d) Name and Address/Contact of Supervisor: Dr. Kanagwa Benjamin / 0712495020
3. **Details of Information/Data required/Loca[illegible]on of Research Centre e.[illegible] [illegible]ice Station and/or Dept**
   Data - (1) Criminal Investigation Procedures Crime Statistics
   (2) Location → CID Section / ICT Department
4. **Benefits of the Research to Uganda Police**
   (1) [illegible] backlog of cases.
   (2) Assist investigating officers by reducing time spent on investigations.
   (3) Identify emerging Crime Patterns in Society.
5. **Undertaking**
   I promise to provide a copy of my research report to the r[illegible]urce centre of the Uganda Police [illegible] 30th/11/12
   Failure to do so, the Uganda Police Force should liaise [illegible] my supervisor and render [illegible] and void.
   Name and Signature: OSAMAI OSBERT ...... *Date* 23.07.12
6. **For Official Use**
   Remarks:
   The Director ICT & Crime Data Management (CIID) to allow research which will help us later. The researcher should avail a copy of the research paper to the Directorate.

[illegible] Officer: [illegible] O. Mattany

Date: 26th/07/12

Note: Information obtained for the research should be used ex[illegible]ly for the intended purpose.

Developing, coordination, Monitoring & Reviewing of Stra[illegible] Plans and Policies for effectiv[illegible] [illegible]n of life and property

Figure 21: Approval Letter from Uganda Police

6.2 Appendix B : Interview Guidelines and Questions

This sections describes the guidelines used to carry out interviews and design questionnaires and some of the questions asked as well as responses from the interviewees.

6.2.1 Guidelines

The guidelines followed were related to the main goals of the project, determining whom to interview and also establishing relevant subgroups depending on their kind of work. The main goals of the interviews were :-

(i) Understanding the current crime investigation process from the point of the case being reported to time of sending the investigation report to the Director of Public Prosecutions (DPP).

(ii) Identifying crime entities involved and their associations.

(iii) Discovering the kind of data gathered during investigations and their corresponding sources and formats.

(iv) Studying known crime patterns related to financial crimes investigations.

(v) Identifying any existing crime systems if any and the kind of data stored.

(vi) Identifying expected outputs in terms of reports from the system.

There was also the need to determine whom to interview.To do this management was asked to provide details of officers responsible for criminal investigations and also guide about the departments specializing on financial crimes. In doing this, the roles of the officers were identified and a clear picture of their operations was understood. This also helped in gathering their thoughts about the current system and how it could be improved.

Another factor was detrmining the number of people to interview.It was important to determine the number of people to interview by clearly understanding the different roles played in the investigation process and also discovering the sections or departments that will be covered by the project.

6.2.2 Interview Questions

These were the interview questions asked during sessions with the director of investigations, investigative police officers and Information Technology experts at the force.

(i) Please describe the current investigation process for the crime of ”Causing financial loss to government” ?

(ii) In such a case who are the key players involved and what kind of records do you get from them for crime analysis?

(iii) Are the above records stored in an electronic database in house?

(iv) How is the above data used to unearth criminal activities and what known pattern in used as reference?

(v) Are the investigative officers skilled in ICT procedures and to what level?

(vi) What challenges are faced in the current system and how do they impact government and the citizens?

(vii) Which officers are knowledgeable about financial crime investigations and under which departments are the attached?

(viii) Are all Police Stations code-named uniquely and What is the reference number format?

(ix) What kind of reports are generated after the investigations are completed and in what format?

(x) Which departments will use the new system?

(xi) Who has got access to investigation data?

6.3 Appendix C : SRS Document

SRS Document

Software Requirements Specification

Version 1.0

December, 2013

Krimino Crime Investigation Tool

Osbert Osamai

2010/HD18/1259U

Submitted in partial fulfillment of the award of Master of Science in Data Communications

and Software

Engineering

1 Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the ”Krimino” Crime Investigation Tool.It will explain the purpose and features of the tool, the interfaces of the tool, what the tool will do, the constraints under which it must operate and how the tool will interface with external systems. This document is intended for both the stakeholders and the developers of the tool.

1.2 Scope of Project

This software application/tool will be a web-based application using Complex Event Processing techniques to aid the crime investigation process especially during data analysis and will be used by police investigators. It will focus on crimes related to causing financial to government in ministries.

1.3 Glossary

Complex Event Processing : It is a foundational technology for detecting and managing the events that happen in event-driven enterprises. It is a collection of methods, tools and techniques applied in processing events as they happen.

User : Anyone using the application (mainly police officer)

Esper Engine : The component of the application responsible for detecting events matching pattern.

2 Overall Description

2.1 Assumptions

The assumptions are that the tool will read data events from an existing MySQL Database and that the input data format will be Comma Seperated Values (CSV).

2.2 Constraints

(i) The tool will read incoming events in CSV formats only for now.

(ii) The tool will be developed using the Java programming language.

(iii) The tool will use the Esper engine library.

(iv) The tool will focus on financial crimes with structured data available.

2.3 Dependencies

The tool requires deployment on glassfish server 3 before it can be launched.

2.4 System Environment

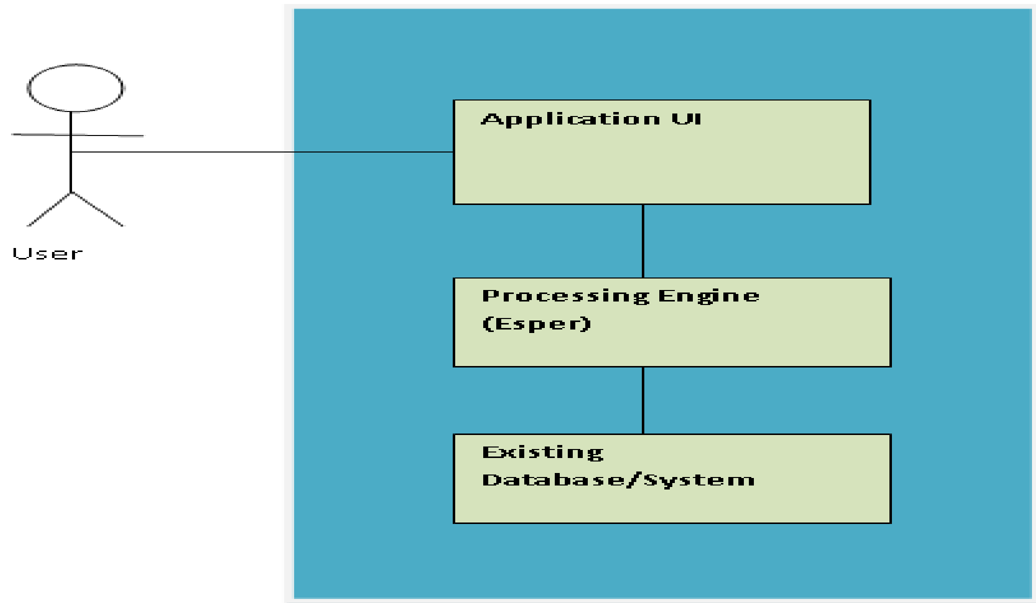


Figure 22: The system environment of the Krimino Crime Investigation Tool

The tool supports one or more current users who access the it through a web client page by interacting with the existing database to generate CSV format input files for processing by the engine and eventual display to the client page.

2.5 Functional Requirements

This section outlines the use cases for each of the Police investigative officer.

2.5.1 Police Officer Use Case

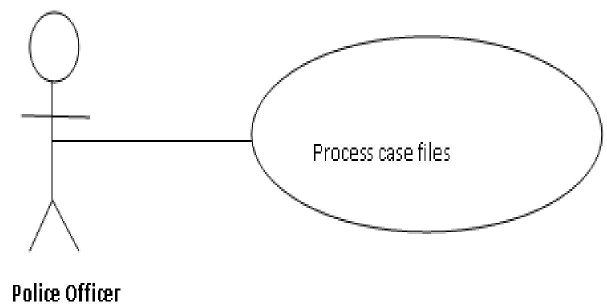


Figure 23: Police officer simple Use Case

2.5.1.1 Brief Description

The Police officer uses the application to process files related to a given case file number.

2.5.1.2 Step-By-Step Description

(i) The Officer launches the application via a web browser.

(ii) The Officer processes case files.

2.5.2 Complete Police Officer Use Case

The Officer has the following complete use cases. Use Case: Launch App

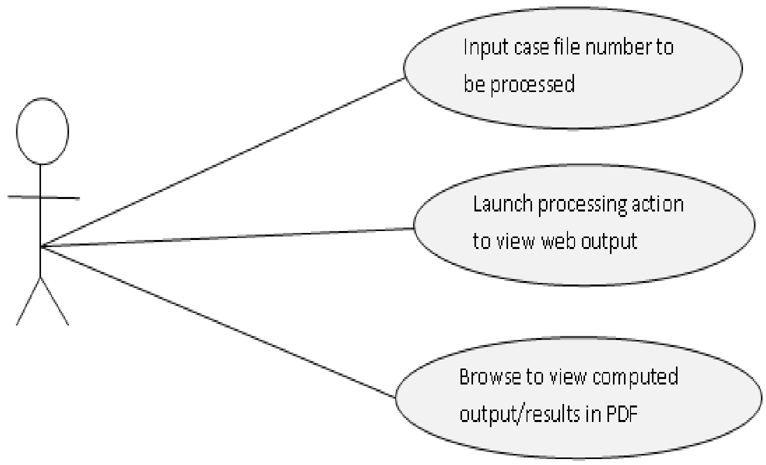


Figure 24: The complete Use Case model for the Crime Investigation Tool

2.5.2.1 Brief Description

On launching the application the user inputs the case file number to be processed after which he/she define the parameters for the pattern to be check against. The use then clicks the process button to initiate the generation of input files from the existing database and display results processed by the engine.

2.5.2.2 Initial Step-By-Step Description

(i) The Officer launches the App.

(ii) The application displays the web user interface for entry of case number and parameters.

(iii) The Officer enters the case file number and pattern parameters.

(iv) CSV format files are generated for the selected case file number and pushed to the engine for processing.

(v) The officer is provided with an XHTML web page with records matching the pattern and the rate of matching.

2.6 User Characteristics

The officer is expected to have basic computer skills to enable comfortable navigation and use of the tool.

2.7 Non-Functional Requirements

The crime investigation tool will have the following non-functional requirements:

(i) The tool should be easy to use for a Police Officer with elementary computer technology skills.

(ii) The output should be easily readable and understood.

(iii) The tool should be capable of simplifying the investigation process.

3 Requirements Specification

3.1 External Interface Requirements

The application requires a connection to an existing MySQL database using a connector and also connection to a Complex Event Processing engine called Esper for Java developments.

3.2 Functional Requirements

3.2.1 User Interface Requirements

(i) The tool should provide menus and buttons that are self-explanatory and easy to use.

(ii) The web page display should provide clearly worded titles and accurate data.

3.2.2 Operational Requirements

(i) The user should enter case file numbers that exist in the database.

(ii) The application should be able to generate CSV files from the database and process them according to the rules set.

(iii) The results should be display both on screen and in a PDF format file in a local location.

3.3 Detailed Non-Functional Requirements

3.3.1 Logical Structure of the Data

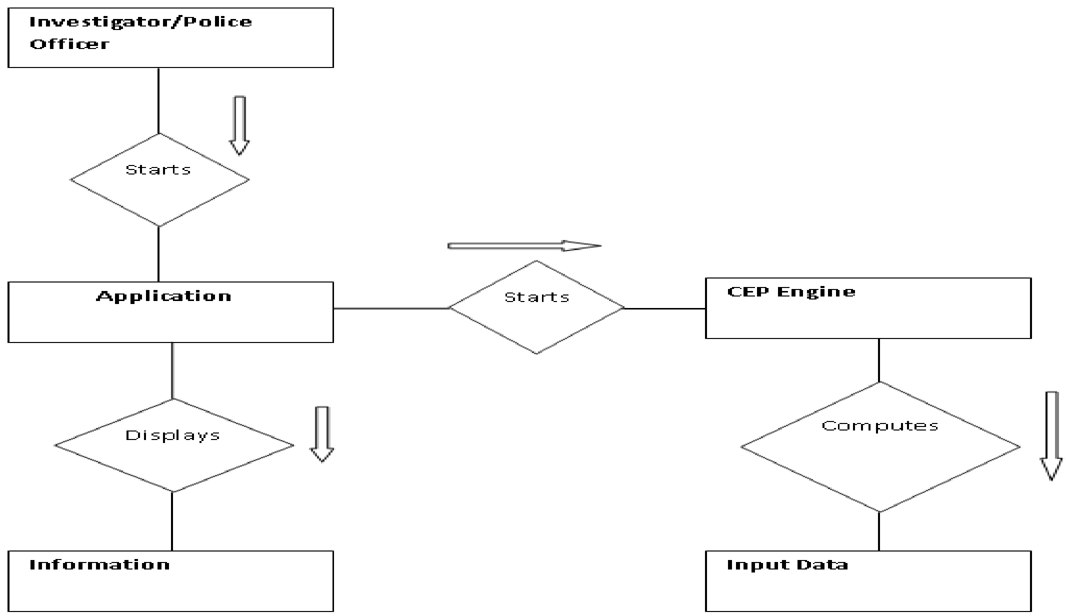


Figure 25: The logical design of the tool

6.4 Appendix D : Test Plan

Test Plan

Functional Test Plan for Krimino Crime InvestigationTool

Version 1.0

Prepared by Osbert Osamai 2010/HD18/1295U

Data Communications and Software Engineering

1 Introduction

This Test Plan provides the strategy for testing the tool. This section of the Test Plan document describes the following:

(i) Purpose

(ii) Objectives

1.1 Document Purpose

This Test Plan reviews

(i) Existing project information

(ii) Application Requirements and critical transactions to be tested

(iii) Testing types and strategies to be implemented

(iv) A proposed testing schedule

1.2 Objectives

This document provides the platform for testing the Krimino Crime Investigation tool. The objective of testing is to ensure that the project meets the Software Requirement Specifications version 1.0 created for the same . This document will provide the guidelines for the testing team to test the functionalities embedded in the project and to help for evaluating the Software Quality of the project.

2 Project Scope

This section of the Test Plan document describes the scope of the tool.

2.1 In Scope

The tool will be tested for its functionality on a any computer running a web browser with glassfish server installed, RAM of not less than 1GB and processor speed of not less than 2GHz. Testing will include business/functional requirements testing .The tests will run for not more than one week and will include several functions like:

(i) Selecting and setting parameters for the pattern

(ii) Generating files from database

(iii) Processing the files based on rules set

(iv) Displaying response to screen

(v) Navigations about the system

2.2 Out of Scope

User acceptance testing will not constitute the scope of this test.

3 Assumptions

All modules of the system will be working and investigative officers will be available for the tests.

4 Dependencies

The testing process depends on the completion of the software requirements specification document and the implementation of the application prototype.

5 Risks

The risks include the following:

(i) Failure to complete implementation of the application prototype by the testing period.

(ii) Failure to identify appropriate Test Cases which will result in misappropriated testing effort.

6 Project Resources

The roles, responsibilities and persons responsible for testing the investigation tool are described in the following table:

Role	Responsib ility	Person Responsible
Testers	Execute Test • Cases • Find, report and track defects based on domain knowledge Analyze • results.	Officers, Osbert Osamai
Developers	• Deliver complete builds of the application • Install the application on web server • Eliminate agreed upon defects	Osbert Osamai

Figure 26: Project roles and responsibilities

7 Test Strategies/Techniques

This section of the document describes the test design and test data for the Crime Investigation tool.

7.1 Test Design

Considering the scope of the project and the time limitations, we will be performing following tests:

7.1.1 Business/Functional Requirements : This test verifies whether specific requirements that were spelled out in the software requirements specification are met. For performing the above mentioned tests, we will create test cases as shown in the following table:

Use Case ID	Case Description	Test Case
UC-1	Use Case File Generation from Database	TC-1
UC-2	Use Case Record processing by Esper Engine	TC-2
UC-3	Use Case Display of results to web page	TC-3
UC-4	Use Case Display of results to PDF file	TC-4
UC-5	Use Case Launch Application	TC-5
UC-6	Use Case Exit Application	TC-6

Figure 27: Use Cases and Test Cases

The following table describes the description of each test case presented in previous table, and the results expected and actual achieved results

Test Case	Use Case	Description	Expected Result	Actual Result
TC-1	UC-1	Officer inputs case file number, selects parameters and clicks the process button.	CSV format files should be generated in the installation path of the application.	
TC-2	UC-2	Officer inputs case file number, selects parameters and clicks the process button.	Text files should be generated in a location indicating processed data that match pattern.	
TC-3	UC-3	Officer inputs case file number, selects parameters and clicks the process button.	All computed results should be displayed to the web page.	
TC-4	UC-4	Officer inputs case file number, selects parameters and clicks the process button.	Computed results should be written to a PDF file in a local location.	
TC-5	UC-5	Officer starts the application by opening shortcut icon on the desktop	The app should launch and display the input page.	

Figure 28: Description, expected results and actual results of each Test Case

8 Defect Responsibility/Resolution

Possible defects identified through manual testing will be reviewed to verify that the observed behavior constitutes a defect. Defects found will be resolved or be maintained as known bugs.

9 Exit Criteria

Testing can proceed to the next stage of the process when a sufficient proportion of the current stage has been completed .All exit criteria should be satisfied by the end of the project.

10 Goals and Deliverables

10 Goals and deliverables of the test plan of the tool are summarized below:

10.1 Goals:

(i) To accomplish all tasks described in this test plan.

(ii) To install a measurable, improvable, repeatable, and manageable test process.

(iii) To verify the functionality and content of the current version of the application.

(iv) To reduce the frequency of errors associated with manual testing.

(v) To find and track as many of defects present as possible as well as along the user path defined in this plan.

10.2 Deliverables

(i) Test result document/summary report.

(ii) Defects report.