**A GRAPH BASED CLUSTERING APPROACH FOR RELATION ORIGIN FROM CRIME DATA**

**A PROJECT REPORT**

###### ***Submitted by***

**MADHUMITHA.D**

**(3501910536)**

**NITHUL K ANIL**

**(3501873501)**

**RAJKISHORE KUMAR**

**(3501910549)**

**GULSHAN KUMAR**

**(3501910522)**

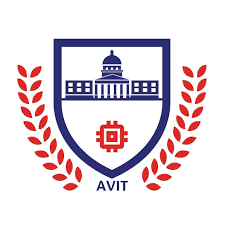
***In partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

in

**COMPUTER SCIENCE AND ENGINEERING**



|  |  |  |
| --- | --- | --- |
| V | **VINAYAKA MISSION’S RESEARCH FOUNDATION**  **AARUPADAI VEEDU INSTITUTE OF TECHNOLOGY**  ***DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING*** |  |

**VINAYAKA MISSION’S RESEARCH FOUNDATION**

**AARUPADAI VEEDU INSTITUTE OF TECHNOLOGY**

**BONAFIDE CERTIFICATE**

Certified that this project report titled “ **A GRAPH BASED CLUSTERING APPROACH OF RELATION ORIGIN FROM CRIME DATA ”** is the bonafide work of **MADHUMITHA.D (3501910536), RAJKISHORE KUMAR (3501910544),NITHUL K ANIL (3501873501) and GULSHAN KUMAR ( 3501910522)** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr. J. JAGADEESAN** **Dr. J. JAGADEESAN**

**HEAD OF THE DEPARTMENT PROJECT GUIDE**

Department of Department of

Computer Science and Engineering Computer Science and

AVIT. Engineering,AVIT.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

Submitted for the University Project Examination held on \_\_\_\_\_\_\_\_\_\_\_\_

**ABSTRACT**

Today mobile and informationtechnology have become an integral part of our lives. A new area where mobile integrated with technology is useful for crime reporting since readily accessible information is not available at any point in investigation, which is a key drawback for communication in police department. Thus Using cloud, we will try to make all the information related to the criminals, Theft case, Missing Case available on the Android Application to the police during their investigation which would speed-up the entire process of tracking down the criminals. A mobile application is also made available to the common people in order to track down the Criminals and Possible of theft. Using this application the user can file the case through their Mobile Application. This will make the communication bridge between the People and the police. This application will be useful for the remote access for the police of criminal data which will be helpful for the investigations carried by police department. Also, it will be useful to store the large amount of data and it will provide a easy way to get details of the crime happened. The Information will be stored on cloud to gain remote access. Because the Crime Rate has been rapidly increased So it is always safety to maintain a cloud server.

**ACKNOWLEDGEMENT**

We would like to express our sincere thanks and gratitude to our Chancellor **Dr. A. S. GANESAN** and our Director **Dr. ANURADHA GANESAN** for giving us the opportunity to pursue our Bachelor’s degree in CSE from this prestigious institution.

My sincere thanks and profound sense of gratitude to Principal **Dr. G. SELVAKUMAR** and Vice Principal Prof. **Dr. S. P. SANGEETHA** and Prof. **Dr. L. PRABHU** for all efforts and administration in educating me in her premiere institution.

I take this opportunity to thank **Dr. J. JAGADEESAN**, Head of Department of Computer Science and Engineering for his kind co-operation in completing this project.

I take the privilege to extend my hearty thanks to my Project coordinator **Dr.S.SELVAKUMAR**,Assistant Professor Gr. II and Project guide **Dr.J.JAGADEESAN** with profound indebtedness for providing the facilities available to us whenever needed and also his constant encouragement throughout this work, which has made the project a success.

Finally with great enthusiasm I express my thanks to all department Faculty members and technical staff member for providing necessary information and their interest in my part of fruitful completion.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **TABLE OF CONTENTS** |  |
| **CHAPTER NO.** | **TITLE** | | **PAGE NO.** |
|  | **ABSTRACT** | | **iii** |
| **1.** | **INTRODUCTION** | | **6-7** |
|  | 1.1 | STATEMENT OF THE PROBLEM | 6 |
|  | 1.2 | ADVANTAGES | 7 |
| **2** | **LITERATURE SURVEY** | | **7-12** |
| **3** | **SYSTEM ANALYSIS** | | **13-14** |
|  | 3.1 | EXISTING SYSTEM | 13 |
|  | 3.2 | PROPOSED SYSTEM | 14 |
| **4** | **SYSTEM SPECIFICATION** | | **15-20** |
|  | 4.1 | SOFTWARE REQUIREMENT | 15 |
|  | 4.2 | HARDWARE REQUIREMENT | 17 |
|  | 4.3 | USED TECHNOLOGY | 18 |
| **5** | **SYSTEM DESIGN** | | **20-31** |

5.1 ARCHITECTURE OF THE SYSTEM 20

5.2 USE CASE DIAGRAM 21

5.3 ACTIVITY DIAGRAM 22

5.4 SEQUENCE DIAGRAM 23

5.5 COLLABORATION DIAGRAM 24

5.6 CLASS DIAGRAM 25

5.7 DATAFLOW DIAGRAM 26

5.8 ER DIAGRAM 27

**6. SYSTEM IMPLEMENTATION 32-39**

6.1 MODULE 32

**7. SYSTEM TESTING 39-41**

7.1 TESTING OBJECTIVES 39

7.2 TYPES OF TESTING 40

7.3 TESTING STRATEGIES 41

**8. FEASIBILITY STUDY 41**

**10. REFERNCES 42**

**11. APPENDIX 43-82**

**CHAPTER 1**

**INTRODUCTION**

Today’s era is seeking a great scope in mobile technology. This technology can be used in many other fields and application such as Gaming, Maps, E -mail, Messaging, Photography and so on. One such area is crime area detection and storing criminal data record. A recent mobile application named Mobile Vic PD, released by the Victoria police in Canada for fighting crime. The mobile application can be used to report minor crimes, offer anonymous tips to police, stay updated on crimes in progress, receive missing child reports or check on stolen property. As the criminal data is not available remotely there is a communication gap between the police officials investigating any case. The disadvantage of this application was that it was prone to fake reporting of crime and there was no other way to verify that the incident was true. This caused chaos among the general public. This paper focuses on overcoming this disadvantage by providing a method for verifying the incident.

This application will be useful for the remote access of criminal data which will be helpful for the investigations carried by police department. Also, it will provide the general users with the facilities like reporting any incidents which would lead to traffic jam. Moreover, it will also provide an alternate safe path on user’s demand before entering the crime area. The database for this project will be stored on cloud to gain remote access. For avoiding any false incidence to be notified to other user, the information provided will be first verified by the police officials. After approval of the information it will be broadcasted to other users using the application.

**OBJECTIVE**

* The objective of this project is to develop an application which is capable of maintaining crimes. This reporting application can be used for maintaining a crime or accident in any time and place.
* To make portable application to create the communication bridge between police and user.

**CHAPTER 2**

**LITERATURE SURVEY**

**[1] Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in *Proc. CVPR*, 2008.**

The aim of this paper is to address recognition of natural human actions in diverse and realistic video settings. This challenging but important subject has mostly been ignored in the past due to several problems one of which is the lack of realistic and annotated video datasets.

First contribution is to address this limitation and to investigate the use of movie scripts for automatic annotation of human actions in videos. We evaluate alternative methods for action retrieval from scripts and show benefits of a text-based classifier. . We present a new method for video classification that builds upon and extends several recent ideas including local space-time features, space-time pyramids and multichannel nonlinear SVMs. Their experimental results demonstrate that post-processing techniques can significantly improve the foreground segmentation masks produced by a BGS algorithm. They provided recommendations for achieving robust foreground segmentation based on the lessons learned performing this comparative study.

**DRAWBACKS**:

* This paper is based on the human activities. Finding the suspicious activities of human is difficult for large areas.
* The main disadvantage of the SVM algorithm is that it has several key parameters that need to be set correctly to achieve the best classification results for any given problem. Parameters that may result in an excellent classification accuracy for problem 1, may result in a poor classification accuracy for problem 2. The user may, therefore, have to experiment with a number of different parameter settings in order to achieve a satisfactory result. This takes more time for analysis.

**[2] J. Nielbes, H. Wang, and L. Fei-Fei, “Unsupervised learning of human action categories using spatial-temporal words,” *IJCV*, vol. 79, pp. 299–318, 2008**

We propose a Spatio-Temporal Manifold (STM) model to analyze non-linear mul-tivariate time series with latent spatial structure and apply it to recognize actions in the joint-trajectories space. Based on STM, a novel alignment algorithm Dynamic Manifold Warping (DMW) and a robust motion similarity metric are proposed for human action sequences, both in 2D and3D. DMW extends previous works on spatio-temporal alignment by incorporating manifold learning.

This model can be applied in application domains like, for example, remote monitoring, control and surveillance. Classic approaches to real-time systems do not provide the flexibility and fault-tolerance required in new emerging environments that need to combine a high degree of dynamism with temporal predictability. Their approach addresses these new challenges by combining concepts from the service oriented paradigm and distributed real-time systems.

We evaluate and compare the approach to state-of-the-art methods on motion capture data and realistic videos. Experimental results demonstrate the effectiveness of our approach, which yields visually appealing alignment results, produces higher action recognition accuracy, and can recognize actions from arbitrary views with partial occlusion.

**DRAWBACKS:**

* Since it is a time based variant it is not stable in result therefore prediction is difficult.
* Capturing the image on large area is difficult and hard to analyze.
* Since the captured image are in arbitrary position it is difficult to process.

[3] P. **Matikainen, M. Hebert, and R. Sukthankar, “Representing pairwise**

**spatial and temporal relations for action recognition,” in *Proc. ECCV*,**

**2010, vol. 6311, pp. 508–521**.

This paper presents a model for quality-of-service (QoS)-aware service composition in distributed systems with real-time and fault-tolerance requirements. Human action recognition in videos is a challenging problem with wide applications. State-of-the-art approaches often adopt the popular bag-of-features representation based on isolated local patches or temporal patch trajectories, where motion patterns like object relationships are mostly discarded. This paper proposes a simple representation specifically aimed at the modeling of such motion relationships.

We adopt global and local reference points to characterize motion information, so that the final representation can be robust to camera movement. Our approach operates on top of visual code words derived from local patch trajectories, and therefore does not require accurate foreground-background separation, which is typically a necessary step to model object relationships.

**DRAWBACK**:

* Since it is based on local path trajectories which is to be collected from each individual which is not possible for the large set of data.
* object relationship model is difficult to handle since object nature vary from place to place and by time.

**[4] P. Natarajan and R. Nevatia, “View and scale invariant action recognition using multiview shape-flow models,” in *Proc. CVPR*, 2008.**

Actions in real world applications typically take place in cluttered environments with large variations in the orientation and scale of the actor. We present an approach to simultaneously track and recognize known actions that are robust to such variations, starting from a person detection in the standing pose. In our approach we first render synthetic poses from multiple viewpoints using Mocap data for known actions and represent them in a Conditional Random Field(CRF) whose observation potentials are computed using shape similarity and the transition potentials are computed using optical flow. We enhance these basic potentials with terms to represent spatial and temporal constraints and call our enhanced model the Shape, Flow, Duration-Conditional Random Field(SFD-CRF). We find the best sequence of actions using Viterbi search in the SFD-CRF.

We demonstrate our approach on videos from multiple viewpoints and in the presence of background clutter. Then, based on the shadow model, an MRF model is constructed for shadow removal.

**DRAWBACKS:**

* CRF is highly computationally complex at the training stage of the algorithm. It makes it very difficult to re-train the model when newer data becomes available.
* Second, they constructed an MRF model to represent the dependencies between the label of a pixel and the shadow models of its neighbors, so that price increases to the public sector associated with private service provision.

[5] **P. Yan, S. M. Khan, and M. Shah, “Learning 4d action feature models for arbitary view action recognition,” in *Proc. CVPR*, 2008.**

Our approach represents videos in each view using both the corresponding view-specific dictionary and the common dictionary. More importantly, it encourages the set of videos taken from different views of the same action to have similar sparse representations.

In this way, we can align view-specific features in the sparse feature spaces spanned by the view specific dictionary set and transfer the view-shared features in the sparse feature space spanned by the common dictionary. Meanwhile, the incoherence between the common dictionary and the view-specific dictionary set enables us to exploit the discrimination information encoded in view specific features and view-shared features separately. We demonstrate a video surveillance system- comprising passive and active pan/tilt/zoom (PTZ) cameras-that intelligently responds to scene complexity, automatically capturing higher resolution video when there are fewer people in the scene and capturing lower resolution video as the number of pedestrians present in the scene increases. To this end, we have developed behavior based-controllers for passive and active cameras, enabling these cameras to carry out multiple observation tasks simultaneously.

**DRAWBACK:**

* In this we uses the active camera for the monitoring so that the images can be analyzed frame by frame thus it comes under the processing of image which is possible when the image is accurate.
* The probability of capturing the suspicious activity is difficult and also it is inaccurate to process therefore the result is not apt.

**CHAPTER 3**

**SYSTEM ANALYSIS**

The purpose of the System Analysis is to produce the brief analysis task and also to establish complete information about the concept, behavior and other constraints such as performance measure and system optimization. The goal of System Analysis is to completely specify the technical details for the main concept in a concise and unambiguous manner.

**EXISTING SYSTEM:**

* In Existing, we make cluster analysis by on criminal dataset of India. The cluster input is used to create custom India map with the cluster zones of states.
* The custom maps displays an overall crime profiles of states which helps police and law enforcement department to take additional preventive measures to combat against the crime and plan advanced investigation strategies.
* This Details are visible to Police officials. Also People can know only the crime zone and not the criminals.
* They can't file the cases against the crime.
* The Storage Maintenance plays a tough role in this system.

**DISADVANTAGES**:

* There is information lack about the criminal.
* Zoning and mapping is difficult to handle.
* Less interaction between Police and Public.
* Criminal Activities Cannot be Filed in online.

**PROPOSED SYSTEM:**

* The purpose of this paper is to develop an android application for crime area detection and store criminal records.
* It provides an application for the user that would provide an details about the criminals .
* It allows user to report incidents, theft, missing and get it verified by the police oﬃcials.
* It will consist of an application for police oﬃcial which can perform database operations on criminal record and allows eﬃcient retrieval of required information from the centralized database present on Cloud.
* The application targets general public and police oﬃcials for managing the incidents and crime without consuming much time.
* Fine interaction between People and Police is maintained.
* Also the user can rate the Police Officials.

**ADVANTAGES:**

* Information about the criminal can be viewed by Police as well as the Public.
* More interaction between Police and Public.
* Criminal Activities Can be Filed in online.
* Low Risk of Data Loss since the data gets stored in Cloud.
* Cases can be shared between the officials easily.

**CHAPTER 4**

**SYSTEM REQUIREMENTS SPECIFICATION**

The purpose of the System Requirement Specification is to produce the specification of the analysis task and also to establish complete information about the requirement, behavior and other constraints such as functional performance and so on. The goal of Software Requirement Specification is to completely specify the technical requirements for the software product in a concise and unambiguous manner.

**4.1 SOFTWARE REQUIREMENTS:**

* Operating system :    Windows XP
* Technology Used :    Android
* IDE :    Eclipse 3.4 (min)
* Emulators :    AVD
* Plug-in :    ADT plug-in
* Tools used :    Android SDK.

**4.2 HARDWARE REQUIREMENTS:**

* Processor : Pentium P4
* Motherboard : Genuine Intel
* RAM : Min 1 GB
* Hard Disk : 80 GB

**4.3   TECHNOLOGY USED:**

This project can be implemented only in JAVA because Android supports only JAVA for user applications.

**JAVA**

Java is Platform Independent. Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. It implements a strong security model, which prevents compiled Java programs from illicitly accessing resources on the system where they execute or on the network. Popular World-Wide Web browsers, as well as some World-Wide Web servers and other systems implement Java interpreters. These are used to display interactive user interfaces, and to script behaviour on these systems.

**ANDROID**

Android is a complete set of software for mobile devices such as tablet computers, smartphones, electronic book readers, notebooks, set-top boxes etc. It contains an Linux-based OS, middleware and key mobile applications. It can be thought of as a mobile operating system. But it is not limited to mobile only. It is currently used in mobiles, tablets, televisions etc.

**MYSQL SERVER:**

* Microsoft SQL Server is an application used to create computer databases for the Microsoft Windows family of server operating systems. Microsoft SQL Server provides an environment used to generate databases that can be accessed from workstations, the Internet, or other media such as a personal digital assistant (PDA).
* MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for " Linux, Apache, MySQL, Perl/PHP/Python".
* MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, Mac OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

**XML**

* XML is a mark-up language for documents containing structured information.
* Structured information contains both content (words, pictures, etc.) and some indication of what role that content plays (for example, content in a section heading has a different meaning from content in a footnote, which means something different than content in a figure caption or content in a database table, etc.).
* Almost all documents have some structure.
* A mark-up language is a mechanism to identify structures in a document. The XML specification defines a standard way to add mark-up to documents.

**ECLIPSE**

Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages: Ada, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Perl, PHP, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang.

It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others. The initial codebase originated from IBM VisualAge.

The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Eclipse uses plug-ins to provide all the functionality within and on top of the runtime system. Its runtime system is based on Equinox, an implementation of the OSGi core framework specification.

The Modeling project contains all the official projects of the Eclipse Foundation focusing on model-based development technologies. They are all compatible with the Eclipse Modeling Framework created by IBM. Those projects are separated in several categories: Model Transformation, Model Development Tools, Concrete Syntax Development, Abstract Syntax Development, Technology and Research, and Amalgam. Eclipse supports development for [Tomcat](http://en.wikipedia.org/wiki/Apache_Tomcat), Glassfish and many other servers and is often capable of installing the required server (for development) directly from the IDE. It supports remote debugging, allowing the user to watch variables and step through the code of an application that is running on the attached server.

**ARCHITECTURE OF ECLIPSE**

In addition to allowing the Eclipse Platform to be extended using other programming languages such as C and Python, the plug-in framework allows the Eclipse Platform to work with typesetting languages like LaTeX, networking applications such as telnet and database management systems. The plug-in architecture supports writing any desired extension to the environment, such as for configuration management. Java and CVS support is provided in the Eclipse SDK, with support for other version control systems provided by third-party plug-ins.

With the exception of a small run-time kernel, everything in Eclipse is a plug-in. This means that every plug-in developed integrates with Eclipse in exactly the same way as other plug-ins; in this respect, all features are "created equal".[citation needed] Eclipse provides plug-ins for a wide variety of features, some of which are through third parties using both free and commercial models. Examples of plug-ins include for UML, for Sequence and other UML diagrams, a plug-in for DB Explorer, and many others.

Model Transformation projects uses [EMF](http://en.wikipedia.org/wiki/Eclipse_Modeling_Framework) based models as an input and produce either a model or text as an output. Model to model transformation projects includes [ATL](http://en.wikipedia.org/wiki/ATLAS_Transformation_Language), an open source transformation language and toolkit used to transform a given model or to generate a new model from a given [EMF](http://en.wikipedia.org/wiki/Eclipse_Modeling_Framework) model. Model to text transformation projects contains [Acceleo](http://en.wikipedia.org/wiki/Acceleo), an implementation of [MOFM2T](http://en.wikipedia.org/wiki/MOFM2T), a standard model to text language from the [OMG](http://en.wikipedia.org/wiki/Object_Management_Group). [Acceleo](http://en.wikipedia.org/wiki/Acceleo) is an open source code generator that can generate any textual language (Java, PHP, Python, etc.) from [EMF](http://en.wikipedia.org/wiki/Eclipse_Modeling_Framework) based models defined with any metamodel ([UML](http://en.wikipedia.org/wiki/Unified_Modeling_Language), [SysML](http://en.wikipedia.org/wiki/Systems_Modeling_Language), etc.).

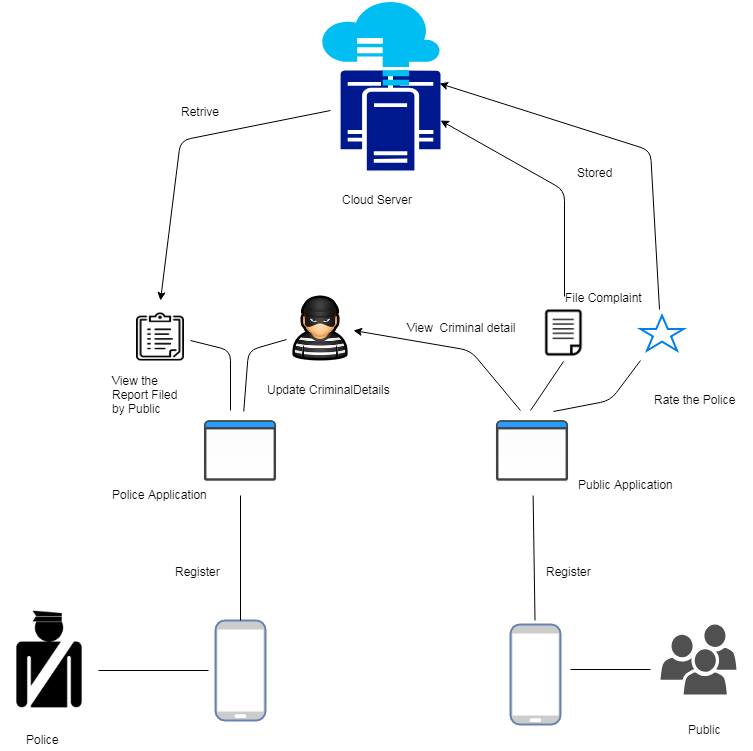
The Eclipse SDK includes the Eclipse Java development tools (JDT), offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis.

The IDE also makes use of a workspace, in this case a set of metadata over a flat file space allowing external file modifications as long as the corresponding workspace "resource" is refreshed afterwards.

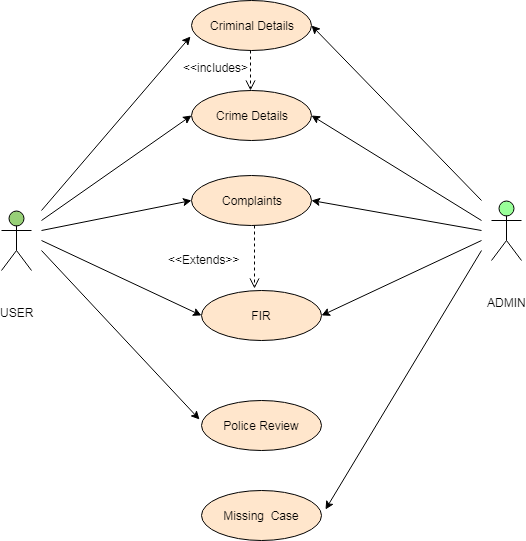
**CHAPTER 5**

**SYSTEM DESIGN SPECIFICATION**

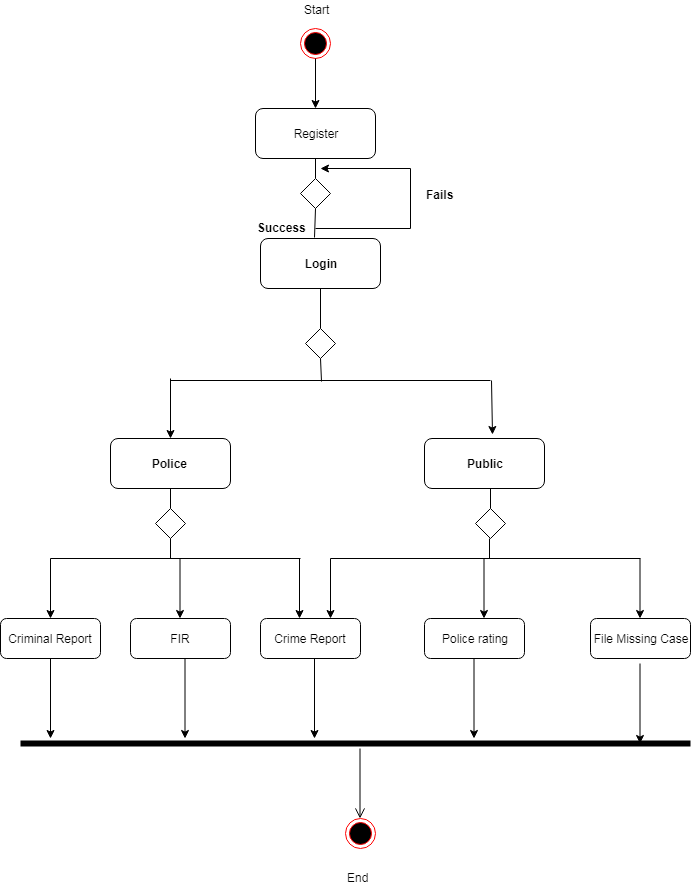
**5.1ARCHITECTURE DIAGRAM:**



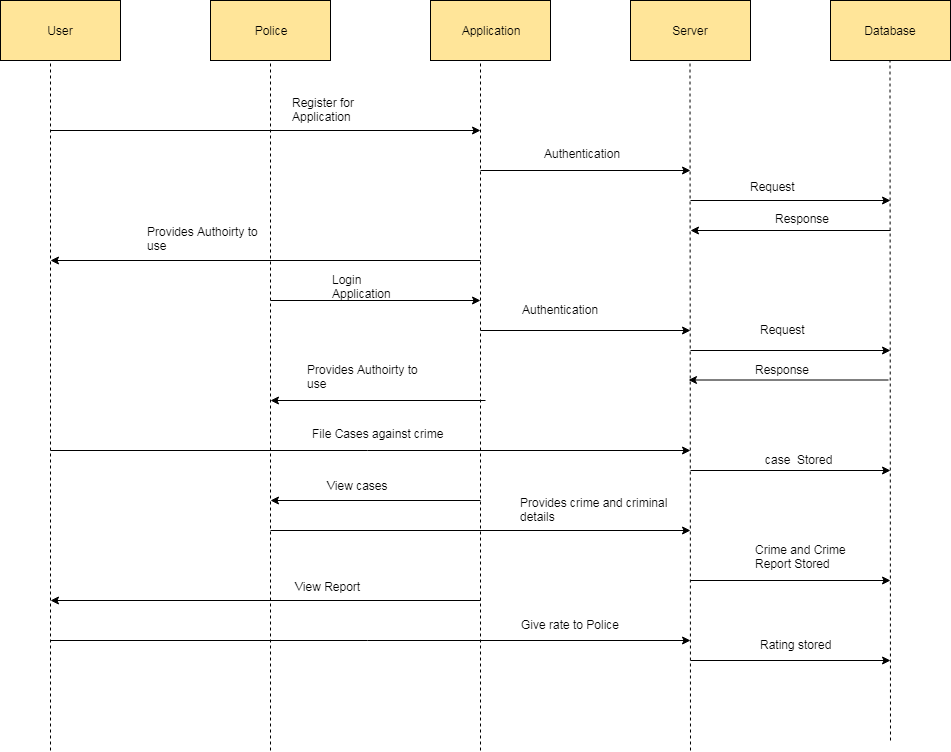
**5.2USECASE DIAGRAM:**

****

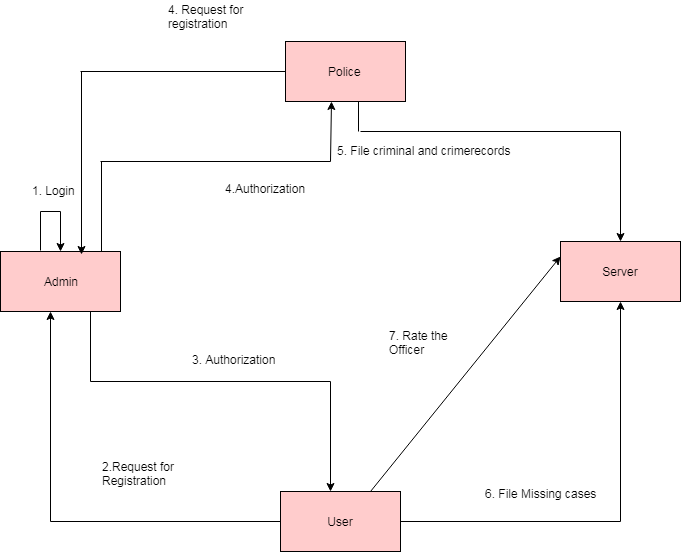
**5.3ACTIVITY DIAGRAM:**

****

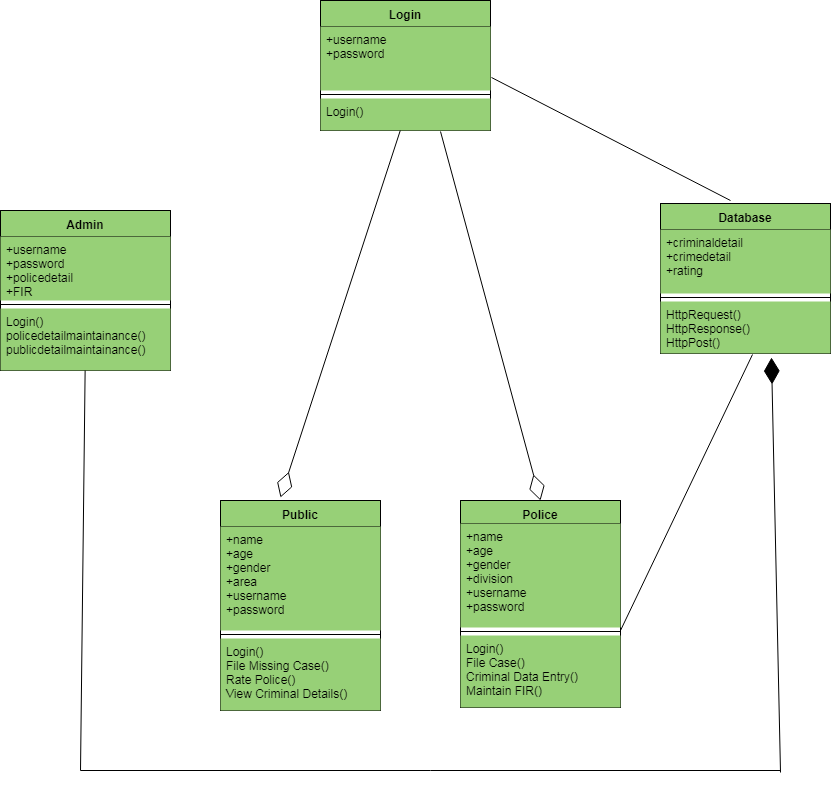
**5.4 SEQUENCE DIAGRAM:**

****

**5.5 COLLABORATION DIAGRAM:**

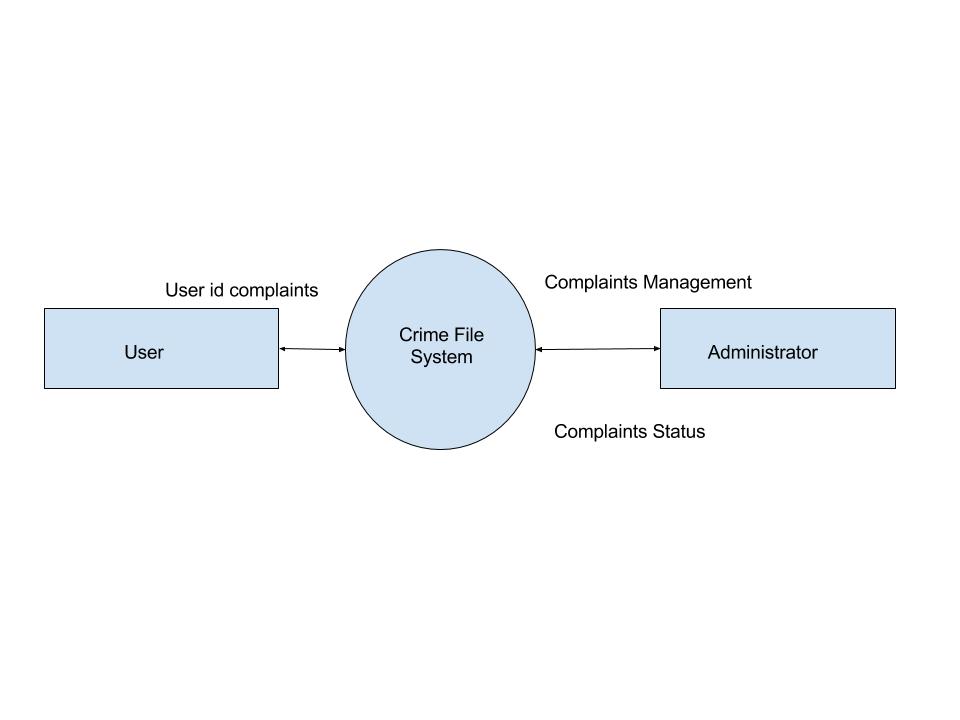
****

**5.6CLASS DIAGRAM:**

****

**5.7 DATAFLOW DIAGRAM:**

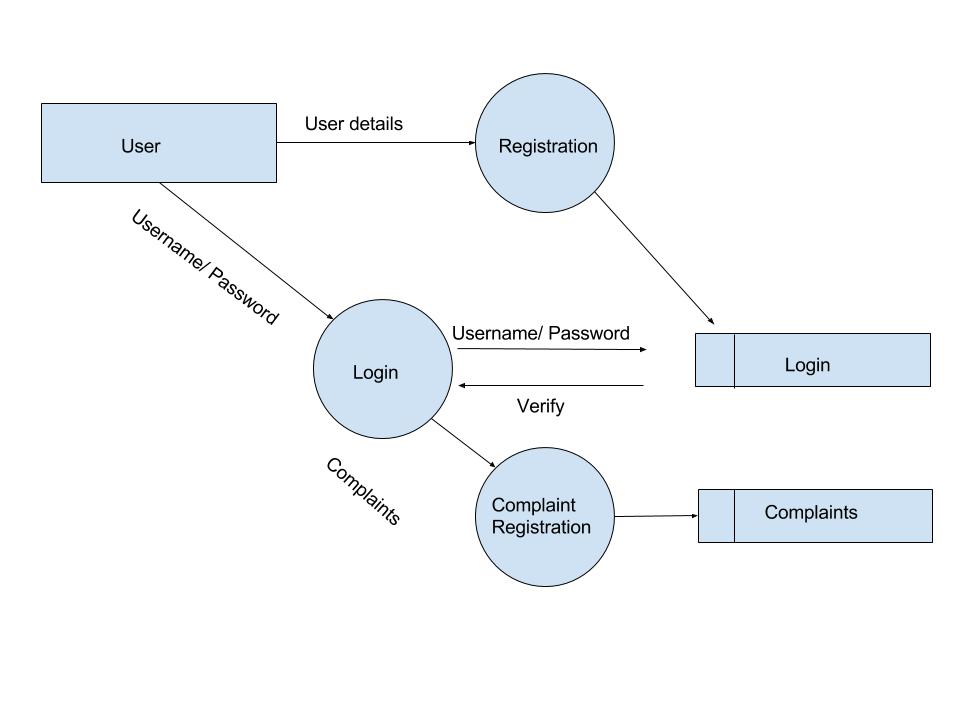
**LEVEL 0:**



Crime Details

Police

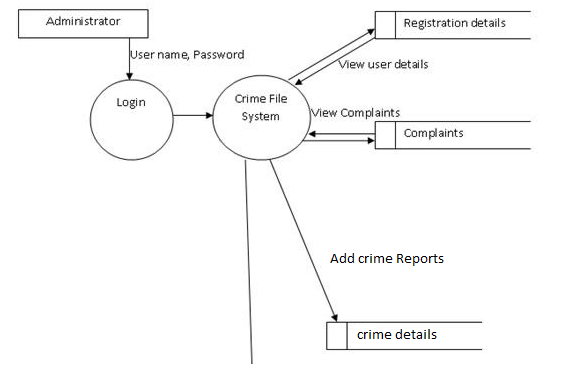
**LEVEL 1:**



Crime record

View Police Details

**LEVEL 2:**

****

Police Registration

**LEVEL 3:**

Police

Username

Password

Username Password Verify

Login

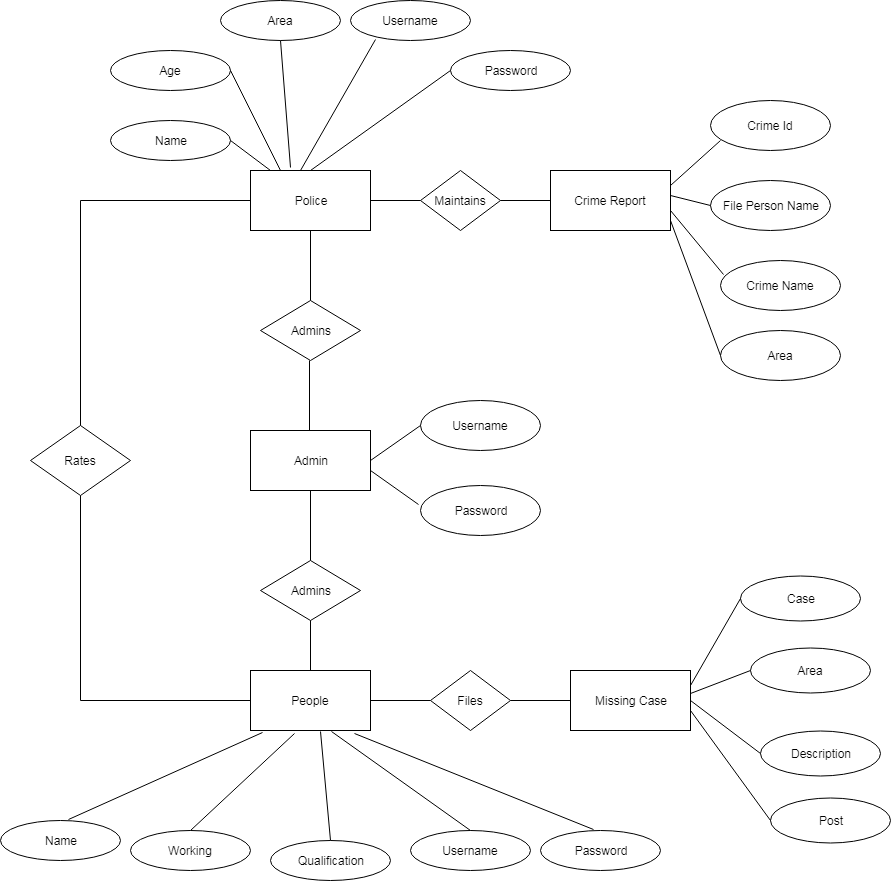
criminal

record

Crime Details

Crime Report

**5.8 ER DIAGRAM:**

****

**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

**6.1 MODULE**

Software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements. Modularity is the single attribute of software that allows a program to be intellectually manageable.

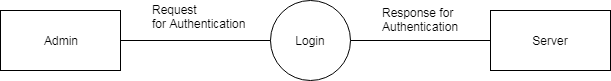
There are 4 modules:

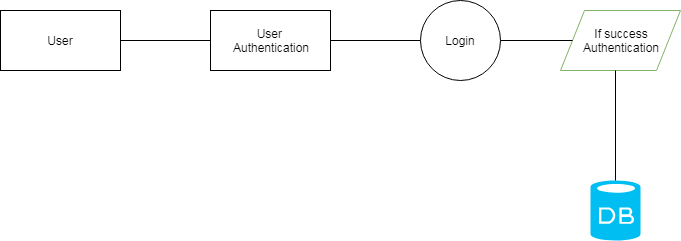
* Authentication Module
* Admin Module
* Police Module
* User Module

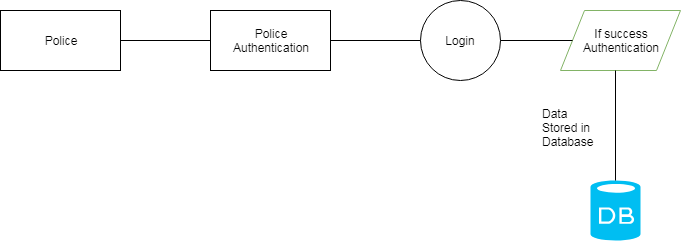
**AUTHENTICATION MODULE:**

In this module all the three participant should gets Authentication to the application. The Login process is then verified by the Server. If the login success then they gets authentication to use the application.

Separate Login for Admin, User and Police has been provided.

****

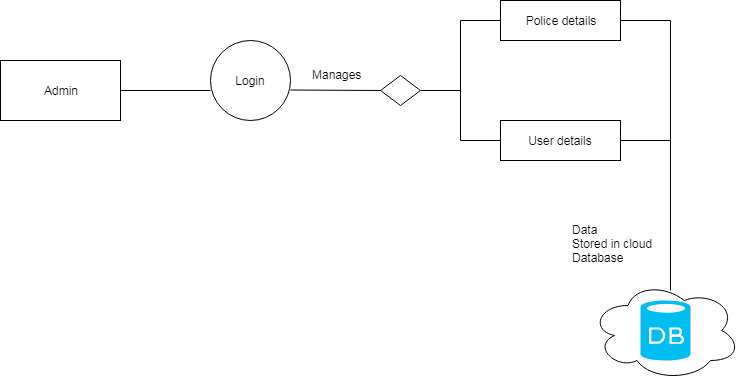
****

****

**ADMIN MODULE:**

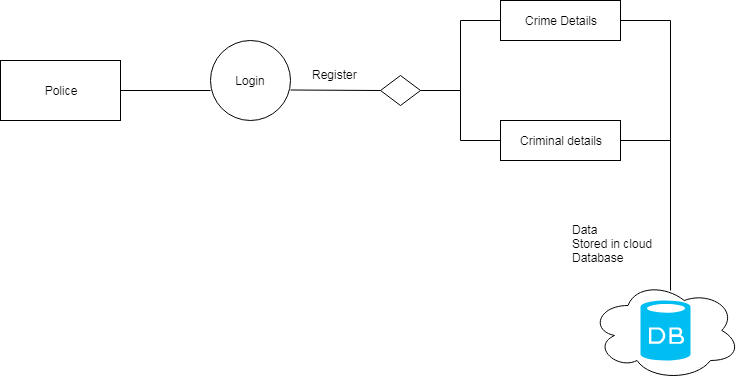
In this module the admin manages the details of user and the police. They can manage their details as well as provide service to them. After the validation process the admin can do this.

Once the admin logins to the application they can maintain the details of the police as well as the public

****

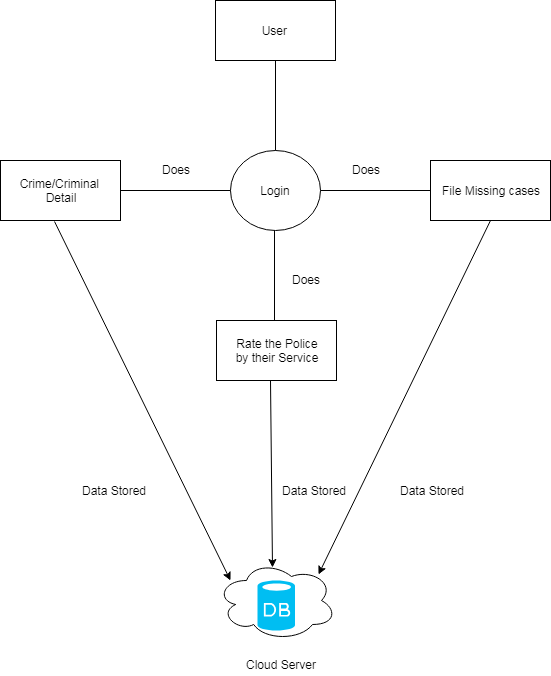
**POLICE MODULE:**

* This module mainly concentrate on police. When the police gets login, they can have the option for entering the details of criminal and crime details.
* They can also read the complaints filed by the public. They separate the crime by the area and store the detail in the cloud server.

****

**USER MODULE:**

* This module mainly concentrate on Public. When the Public gets login, they can have the option for entering the details of missing case.
* They can also view the crime and criminal information nearby them
* They also can rate the police in their area since they cannot go to the station directly also they cannot file the case against police.

****

**CHAPTER 7**

**SYSTEM TESTING**

**7.1 TESTING OBJECTIVES:**

Testing is a set of activities that can be planned in advance and conducted systematically. For this reason a template for software testing, a set of steps into which can place specific test case design techniques and testing methods should be defined for software process. Testing often accounts for more effort than any other software engineering activity. If it is conducted haphazardly, time is wasted, unnecessary effort is expanded, and even worse, errors sneak through undetected. It would therefore seem reasonable to establish a systematic strategy for testing software

**7.2 TYPES OF TESTING:**

**Unit Testing:**

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect.

Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use. In the company as well as seeker registration form, the zero length username and password are given and checked. Also the duplicate username is given and checked.

In the job and question entry, the button will send data to the server only if the client side validations are made. The dates are entered in wrong manner and checked. Wrong email-id and web site URL (Universal Resource Locator) is given and checked.

**Integration Testing:**

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions.

Thus the system testing is a confirmation that all is correct and an opportunity to show the user that the system works.

**Validation Testing:**

The final step involves Validation testing, which determines whether the software function as the user expected. The end-user rather than the system developer conduct this test most software developers as a process called “Alpha and Beta were testing” to uncover that only the end user seems able to find.

The compilation of the entire project is based on the full satisfaction of the end users. In the project, validation testing is made in various forms. In question entry form, the correct answer only will be accepted in the answer box. The answers other than the four given choices will not be accepted.

**7.3 TESTING STRATEGIES:**

A number of software testing strategies have been proposed in the literature. All provide the software developer with a template for testing and all have the following generic characteristics:

* Testing begins at the component level and works “outward” toward the integration of the entire computer-based system.
* Different testing techniques are appropriate at different points in time.
* The developer of the s/w conducts testing and for large projects, independent test group.

**7.3.1 White Box Testing**

It is just the vice versa of the Black Box testing. They do not watch the internal variables during testing. This gives clear idea about what is going on during execution of the system. The point at which the bug occurs were all clear and were removed.

**7.3.2 Black Box Testing**

In this testing we give input to the system and test the output. Here i do not go for watching the internal file in the system and what are the changes made on them for the required output.

**7.3.3 Interface Testing**

The Interface Testing is performed to verify the interfaces between sub modules while performing integration of sub modules aiding master module recursively.

**7.3.4 Module Testing**

Module Testing is a process of testing the system, module by module. It includes the various inputs given, outputs produced and their correctness. By testing in this method we would be very clear of all the bugs that have occurred.

**CHAPTER 8**

**FEASIBILITY STUDY**

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study  activity is to determine whether it would be financially and technically feasible to develop the product. The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system as well as various constraints on the behavior of the system.

**Technical Feasibility**

This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but might include:

* + The facility to produce outputs in a given time.
  + Response time under certain conditions.
  + Ability to process a certain volume of transaction at a particular speed.
  + Facility to communicate data to distant locations.

In examining technical feasibility, configuration of the system is given more importance than the actual make of hardware. The configuration should give the complete picture about the system’s requirements: How many workstations are required, how these units are interconnected so that they could operate and communicate smoothly? And what speeds of input and output should be achieved at particular quality of printing.

**Economic Feasibility**

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as Cost / Benefit analysis, the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system. Otherwise, further justification or  alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an outgoing effort that improves in accuracy at each phase of the system life cycle.

**Operational Feasibility**

This is mainly related to human organizational and political aspects. The points to be considered are:

* + What changes will be brought with the system?
  + What organizational structure are disturbed?
  + What new skills will be required? Do the existing staff members have

these skills? If not, can they be trained in due course of time?

This feasibility study is carried out by a small group of people who are familiar with information system technique and are skilled in system analysis and design process. Proposed projects are beneficial only if they can be turned into information system that will meet the operating requirements of the organization. This test of feasibility asks if the system will work when it is developed and installed.

**CHAPTER 9**

**CONCLUSION**

We have overcome the problem of communication gap between the police during their investigation. We also provide solution to bridge the communication gap between police and general user. Also, the criminal information will be readily accessible to the police officials as it is stored on the cloud. The problem of reporting fake crimes will be overcome as this application will need the verification of police to report incidents reported by user to broadcast it to other users using the same application. This work makes user friendly so that the user can file the complaints at the home itself.

In future, some other security algorithms can be used to provide better security measures for the criminal database.

**CHAPTER 10**

**REFERENCES**

1. **William Akotam Agangiba,Millicent Akotam Agangiba**, Mobile solution for Metropolitan Crime Detection and Reporting, Journal of Emerging Trends in Computing and Information sciences, Vol.4, No. 12, 2013, 2079-8407.
2. **VicPD**, Report Crime, Tack Crime, Fight Crime, From your pocket available  at: https://www.vicpd.ca/mobile [Accessed: 29/10/2013].M.
3. **Manav Singhal, Anupam Shukla**,”Implementation of location based services in Android using GPS and Web Services”,(IJCSI) International Journal of Computer Science Issues, Vol. 9, Issue 1, No. 2, January 2012, 1694-0814.
4. **Mayur Dhande, Amruta Barawkar, Raman Dhoot**,”Android Bachaosos Application”, (IJCTA) International Journal of Computer Technology and Application, Vol. 5 (3), 826-828.
5. **Pragya Gupta, Sudha Gupta**,”Mobile Cloud Computing: The Future of Cloud”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 1, Issue 3, September 2012.
6. **Surbhi Aggarwal, Neha Goyal, Kirti Aggarwal**, “A review of comparative study of MD5 and SHA security Algorithm”, International Journals of Computer Application (0975-8887), Vol. 104-No. 14, October-2014

**APPENDIX**

**CODING AND SCREENSHOTS**

**DATABASE CODE;**

package com.uniq.dao;

import java.io.IOException;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.Date;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import com.google.android.gcm.server.Message;

import com.google.android.gcm.server.MulticastResult;

import com.google.android.gcm.server.Result;

import com.google.android.gcm.server.Sender;

import com.uniq.connection.ConnectionManager;

public class UpdateDB {

static Connection conn = ConnectionManager.getConnection();

public void registration(String email, String password, String phone,

String region) {

try {

Statement st = conn.createStatement();

st.execute("INSERT INTO usertable(password, email, phone) VALUES('"

+ password + "', '" + email + "', '" + phone + "')");

st.execute("INSERT INTO user\_region(email, region) VALUES('"

+ email + "', '" + region + "')");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static void UserRegistration(String name, String email, String mobile,

String psw, String address) {

// TODO Auto-generated method stub

try {

Statement st = conn.createStatement();

st.execute("INSERT INTO newuser(name,email,mobile,password,address) values('"

+ name

+ "','"

+ email

+ "','"

+ mobile

+ "','"

+ psw

+ "','" + address + "')");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static boolean UserLogin(String username, String password) {

// TODO Auto-generated method stub

boolean success = false;

try {

Statement st = conn.createStatement();

ResultSet rs = st

.executeQuery("SELECT name FROM newuser WHERE password = '"

+ password + "'");

while (rs.next()) {

if (rs.getString("name").equals(username)

) {

success = true;

} else {

success = false;

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return success;

}

public boolean login(String email, String password, String region) {

boolean success = false;

try {

Statement st = conn.createStatement();

st.execute("UPDATE user\_region SET region='" + region

+ "' WHERE email='" + email + "'");

ResultSet rs = st

.executeQuery("SELECT password FROM usertable WHERE email = '"

+ email + "'");

while (rs.next()) {

if (rs.getString("password").equals(password)) {

success = true;

} else {

success = false;

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return success;

}

public String getRegions() {

JSONArray arr = new JSONArray();

try {

Statement st = conn.createStatement();

ResultSet rs = st.executeQuery("SELECT \* from city\_regions");

while (rs.next()) {

JSONObject obj = new JSONObject();

obj.put("region", rs.getString("regions"));

arr.put(obj);

}

} catch (SQLException | JSONException e) {

e.printStackTrace();

}

return arr.toString();

}

public void updateGcm(String gcm\_id) {

}

public String getGcm(String email) {

ArrayList<String> list = new ArrayList<String>();

try {

Statement st = conn.createStatement();

ResultSet rt = st

.executeQuery("SELECT \* FROM usertable WHERE email='"

+ email + "'");

if (rt != null) {

while (rt.next()) {

list.add(rt.getString("gcm\_id"));

}

System.out.println("Location...." + list.toString());

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return list.toString();

}

public void createCrime(String email, String subject, String category,

String vehicle\_number, String description, String crime\_date,

String region) {

// TODO Auto-generated method stub

try {

Statement st = conn.createStatement();

st.execute("INSERT INTO crime(email, subject, category, vehicle\_number, description, crime\_id, crime\_date, region) "

+ "VALUES('"

+ email

+ "', '"

+ subject

+ "', '"

+ category

+ "', '"

+ vehicle\_number

+ "', '"

+ description

+ "', '"

+ new java.sql.Timestamp(new Date().getTime())+ "', '" + crime\_date + "', '" + region + "')");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

// sendPostAlert(email);

}

public String getCrimes(String email) {

JSONArray arr = new JSONArray();

try {

Statement st = conn.createStatement();

ResultSet rs = st.executeQuery("SELECT \* FROM crime");

while (rs.next()) {

JSONObject obj = new JSONObject();

obj.put("email", rs.getString("email"));

obj.put("category", rs.getString("category"));

obj.put("vehicle\_number", rs.getString("vehicle\_number"));

obj.put("description", rs.getString("description"));

obj.put("crime\_id", rs.getString("crime\_id"));

obj.put("status", rs.getString("status"));

obj.put("crime\_date", rs.getString("crime\_date"));

obj.put("region", rs.getString("region"));

obj.put("subject", rs.getString("subject"));

arr.put(obj);

}

} catch (SQLException | JSONException e) {

e.printStackTrace();

}

return arr.toString();

}

public String getComments(String crime\_id) {

// TODO Auto-generated method stub

JSONArray arr = new JSONArray();

try {

Statement st = conn.createStatement();

ResultSet rs = st

.executeQuery("SELECT \* FROM comments WHERE crime\_id='"

+ crime\_id + "'");

while (rs.next()) {

JSONObject obj = new JSONObject();

obj.put("comments", rs.getString("comments"));

obj.put("date\_time", rs.getString("date\_time"));

arr.put(obj);

}

} catch (SQLException | JSONException e) {

e.printStackTrace();

}

return arr.toString();

}

public void postComment(String crimeId, String comment, String date\_time,

String email) {

try {

Statement st = conn.createStatement();

st.execute("INSERT INTO comments(crime\_id, comments, date\_time, email) VALUES('"

+ crimeId

+ "', '"

+ comment

+ "', '"

+ date\_time

+ "', '"

+ email + "')");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

sendCommentAlert(email, comment);

}

public void updateGcmId(String email, String gcm\_id) {

try {

Statement st = conn.createStatement();

st.execute("UPDATE usertable SET gcm\_id='" + gcm\_id

+ "' WHERE email='" + email + "'");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public void sendPostAlert(String email) {

ArrayList<String> list\_gcm\_id = new ArrayList<String>();

try {

Statement st = conn.createStatement();

ResultSet rs = st.executeQuery("SELECT \* FROM usertable");

while (rs.next()) {

if (!rs.getString("email").equals(email)) {

list\_gcm\_id.add(rs.getString("gcm\_id"));

}

}

} catch (SQLException e) {

e.printStackTrace();

}

if (list\_gcm\_id.size() == 1) {

Sender sender = new Sender(

"AIzaSyB36xhJBNHfDKwoq8\_qAVkJ8Wg9uLoBE2c");

// AIzaSyD4NbJbVTwW0jqDj5UWak\_ZMIdGQe6n15s

Message message = new Message.Builder().collapseKey("1")

.timeToLive(3).delayWhileIdle(true)

.addData("subject", "Theft Alert !!").build();

try {

Result result = sender.send(message, list\_gcm\_id.get(0), 3);

System.out.println(result.toString());

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

} else if (list\_gcm\_id.size() != 0 && list\_gcm\_id.size() > 1) {

Sender sender = new Sender(

"AIzaSyD4NbJbVTwW0jqDj5UWak\_ZMIdGQe6n15s");

Message message = new Message.Builder().collapseKey("1")

.timeToLive(3).delayWhileIdle(true)

.addData("subject", "Theft Alert!!").build();

try {

MulticastResult multicastResult = sender.sendNoRetry(message,

list\_gcm\_id);

System.out.println(multicastResult.toString());

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

public String assignmentdetail1(String send)

{

boolean success=false;

// System.out.println(dept+year+dates);

JSONArray arr = new JSONArray();

Statement st1;

try

{

st1 = conn.createStatement();

// BETWEEN 10 AND 20a

ResultSet rs = st1.executeQuery("select \* from casedetails where caseid='"+send+"'");

System.out.println(rs);

System.out.println(rs.getFetchSize());

while (rs.next())

{

System.out.println(rs.getString("name"));

System.out.println(rs.getString("age"));

System.out.println(rs.getString("casetype"));

System.out.println(rs.getString("casetype"));

success = true;

JSONObject jsonObject = new JSONObject();

try

{

jsonObject.put("name", rs.getString("name"));

jsonObject.put("contect", rs.getString("age"));

jsonObject.put("vehicle", rs.getString("casetype"));

jsonObject.put("location", rs.getString("location"));

jsonObject.put("state", rs.getString("status"));

arr.put(jsonObject);

}

catch (JSONException e1)

{

System.out.println("Error1: " +e1);

}

}

}

catch (SQLException e)

{

System.out.println("Error: " +e);

}

return arr.toString();

}

public void sendCommentAlert(String email, String comment) {

ArrayList<String> list\_gcm\_id = new ArrayList<String>();

try {

Statement st = conn.createStatement();

ResultSet rs = st.executeQuery("SELECT \* FROM usertable");

while (rs.next()) {

if (!rs.getString("email").equals(email)) {

list\_gcm\_id.add(rs.getString("gcm\_id"));

}

}

} catch (SQLException e) {

e.printStackTrace();

}

if (list\_gcm\_id.size() == 1) {

Sender sender = new Sender(

"AIzaSyB36xhJBNHfDKwoq8\_qAVkJ8Wg9uLoBE2c");

Message message = new Message.Builder().collapseKey("1")

.timeToLive(3).delayWhileIdle(true)

.addData("subject", "Comment: " + comment).build();

try {

Result result = sender.send(message, list\_gcm\_id.get(0), 3);

System.out.println(result.toString());

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

} else if (list\_gcm\_id.size() != 0 && list\_gcm\_id.size() > 1) {

Sender sender = new Sender(

"AIzaSyB36xhJBNHfDKwoq8\_qAVkJ8Wg9uLoBE2c");

Message message = new Message.Builder().collapseKey("1")

.timeToLive(3).delayWhileIdle(true)

.addData("subject", "Theft Alert!!").build();

try {

MulticastResult multicastResult = sender.sendNoRetry(message,

list\_gcm\_id);

System.out.println(multicastResult.toString());

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

public void Registration(String name, String psw, String qulaification,

String category, String gender, String dob, String nationality,

String state, String qualificationcategory, String nri1) {

try {

Statement st = conn.createStatement();

st.execute("INSERT INTO registration values ('" + name + "','"

+ psw + "','" + qulaification

+ "','" + category

+ "','" + gender

+ "','" + dob

+ "','" + nationality

+ "','" + state

+ "','" + qualificationcategory

+ "','" + nri1

+ "')");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public boolean login(String name, String psw) {

// TODO Auto-generated method stub

boolean success = false;

try {

Statement st = conn.createStatement();

ResultSet rs = st

.executeQuery("SELECT Name FROM registration WHERE password = '"

+ psw + "'");

while (rs.next()) {

if (rs.getString("Name").equals(name)) {

success = true;

} else {

success = false;

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return success;

}

public String activities()

{

// TODO Auto-generated method stub

String jsonString = null;

try {

JSONArray jsonArray = new JSONArray();

Statement st = conn.createStatement();

String query = "SELECT \* FROM wanteddetails";

System.out.println(query);

ResultSet rs = st.executeQuery(query);

//System.out.println("query executed");

//System.out.println(rs.getFetchSize());

while (rs.next())

{

JSONObject obj = new JSONObject();

try {

String clgname=rs.getString("Name");

obj.put("collegename", clgname);

obj.put("cutoffmark", rs.getString("Fathername"));

obj.put("totalseats", rs.getString("sex"));

obj.put("cse", rs.getString("Dob"));

obj.put("it", rs.getString("Age"));

obj.put("eee", rs.getString("crime id"));

obj.put("ece", rs.getString("crime type"));

obj.put("othersdept", rs.getString("crime level"));

obj.put("bc", rs.getString("personal Address"));

obj.put("oc", rs.getString("city"));

obj.put("mbc", rs.getString("state"));

obj.put("otherscaste", rs.getString("pincode"));

// obj.put("vacancyseats", rs.getString("vacancyseats"));

// obj.put("fillingseats", rs.getString("fillingseats"));

jsonArray.put(obj);

}

catch (JSONException e1) {

System.out.println("Error1: " +e1);

e1.printStackTrace();

}

jsonString = jsonArray.toString();

//System.out.println(jsonString);

}

}

catch (SQLException e)

{

e.printStackTrace();

System.out.println("Error: " +e);

}

return jsonString;

}

public String activities1()

{

// TODO Auto-generated method stub

String jsonString = null;

try {

JSONArray jsonArray = new JSONArray();

Statement st = conn.createStatement();

String query = "SELECT \* FROM missingdetails";

System.out.println(query);

ResultSet rs = st.executeQuery(query);

while (rs.next())

{

JSONObject obj = new JSONObject();

try {

String clgname=rs.getString("Name");

obj.put("collegename", clgname);

obj.put("cutoffmark", rs.getString("Fathername"));

obj.put("totalseats", rs.getString("sex"));

obj.put("cse", rs.getString("Dob"));

obj.put("it", rs.getString("Age"));

obj.put("eee", rs.getString("Identification Details"));

obj.put("ece", rs.getString("Contact Number"));

obj.put("othersdept", rs.getString("contact Email"));

obj.put("bc", rs.getString("personal Address"));

obj.put("oc", rs.getString("city"));

obj.put("mbc", rs.getString("state"));

obj.put("otherscaste", rs.getString("pincode"));

jsonArray.put(obj);

}

catch (JSONException e1) {

System.out.println("Error1: " +e1);

e1.printStackTrace();

}

jsonString = jsonArray.toString();

//System.out.println(jsonString);

}

}

catch (SQLException e)

{

e.printStackTrace();

System.out.println("Error: " +e);

}

return jsonString;

}

public void createpost(String name, String gender, String age, String caseid, String casetype, String location, String title, String descrption, String status) {

try {

Statement st = conn.createStatement();

st.execute("INSERT INTO casedetails values ('" + name + "','"

+ gender+ "','" + age

+ "','" + caseid

+ "','" +casetype

+ "','" +location

+ "','" + title

+ "','" + descrption

+ "','" + status

+ "')");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

**REGISTRATION CODE:**

package com.uniq.theftintimation;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import org.apache.http.NameValuePair;

import org.apache.http.message.BasicNameValuePair;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import android.app.Activity;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Spinner;

import android.widget.Toast;

import com.uniq.theftintimation.RequestMethod.RequestListener;

public class RegistrationActivity extends Activity {

private EditText edit\_reg\_email, edit\_reg\_password, edit\_phone;

private Button btn\_submit;

private Spinner reg\_spinner\_region;

private RequestMethod requestMethod;

private ProgressDialog pd;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.registration\_page);

requestMethod = new RequestMethod(this);

edit\_reg\_email = (EditText) findViewById(R.id.edit\_reg\_email);

edit\_reg\_password = (EditText) findViewById(R.id.edit\_reg\_password);

edit\_phone = (EditText) findViewById(R.id.edit\_phone);

reg\_spinner\_region = (Spinner) findViewById(R.id.reg\_spinner\_region);

loadRegions();

btn\_submit = (Button) findViewById(R.id.btn\_submit);

btn\_submit.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

if (TextUtils.isEmpty(edit\_reg\_email.getText().toString())) {

Toast.makeText(RegistrationActivity.this, "mail cannot be empty",

Toast.LENGTH\_SHORT).show();

}

else if(!isemail(edit\_reg\_email.getText().toString())){

Toast.makeText(getApplicationContext(), "invalid mail",Toast.LENGTH\_SHORT).show();

}

else if (TextUtils.isEmpty(edit\_reg\_password.getText().toString())) {

Toast.makeText(RegistrationActivity.this, "password cannot be empty",

Toast.LENGTH\_SHORT).show();

}

else if(!isvalidpassword(edit\_reg\_password.getText().toString())){

Toast.makeText(getApplicationContext(), "invalid password\n enter minimum 6 letter",Toast.LENGTH\_SHORT).show();

}

else if (TextUtils.isEmpty(edit\_phone.getText().toString())) {

Toast.makeText(RegistrationActivity.this, "phone cannot be empty",

Toast.LENGTH\_SHORT).show();

}

else if(!isvalidmobileno(edit\_phone.getText().toString())){

Toast.makeText(getApplicationContext(), "invalid phone",Toast.LENGTH\_SHORT).show();

}

else if (TextUtils.isEmpty(reg\_spinner\_region.getSelectedItem().toString())) {

Toast.makeText(RegistrationActivity.this, "region cannot be empty",

Toast.LENGTH\_SHORT).show();

}

else

{

pd = ProgressDialog.show(RegistrationActivity.this, "",

"Registering you...", false, false);

requestMethod.setRequestListener(new RequestListener() {

@Override

public void onRequestError(String error\_response) {

// TODO Auto-generated method stub

if (pd != null && pd.isShowing()) {

pd.dismiss();

}

Toast.makeText(RegistrationActivity.this,

error\_response, Toast.LENGTH\_SHORT).show();

}

@Override

public void onRequestSuccess(String success\_response) {

// TODO Auto-generated method stub

if (pd != null && pd.isShowing()) {

pd.dismiss();

}

if (success\_response.equals("yes")) {

Toast.makeText(RegistrationActivity.this,

"You have been registered successfully",

Toast.LENGTH\_SHORT).show();

startActivity(new Intent(RegistrationActivity.this,

LoginActivity.class));

finish();

}

}

});

List<NameValuePair> nameValuePair = new ArrayList<NameValuePair>();

nameValuePair

.add(new BasicNameValuePair("flag", "registration"));

nameValuePair.add(new BasicNameValuePair("email",

edit\_reg\_email.getText().toString()));

nameValuePair.add(new BasicNameValuePair("password",

edit\_reg\_password.getText().toString()));

nameValuePair.add(new BasicNameValuePair("phone", edit\_phone

.getText().toString()));

nameValuePair.add(new BasicNameValuePair("region",

reg\_spinner\_region.getSelectedItem().toString()));

requestMethod.postMethod(nameValuePair);

Intent i = new Intent(RegistrationActivity.this,

LoginActivity.class);

startActivity(i);

}

}

private boolean isemail(String string) {

// TODO Auto-generated method stub

String EMAIL\_PATTERN = "^[\_A-Za-z0-9-\\+]+(\\.[\_A-Za-z0-9-]+)\*@"

+ "[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$";

Pattern pattern = Pattern.compile(EMAIL\_PATTERN);

Matcher matcher = pattern.matcher(string);

return matcher.matches();

}

private boolean isvalidmobileno(String string) {

// TODO Auto-generated method stub

if (string != null && string.length()==10) {

return true;

}

return false;

}

private boolean isvalidpassword(String string) {

// TODO Auto-generated method stub

if (string != null && string.length() > 6) {

return true;

}

return false;

}

});

}

private void loadRegions() {

pd = ProgressDialog.show(RegistrationActivity.this, "",

"Getting regions...", false, false);

requestMethod.setRequestListener(new RequestListener() {

@Override

public void onRequestError(String error\_response) {

// TODO Auto-generated method stub

if (pd != null && pd.isShowing()) {

pd.dismiss();

}

Toast.makeText(RegistrationActivity.this, error\_response,

Toast.LENGTH\_SHORT).show();

}

@Override

public void onRequestSuccess(String success\_response) {

// TODO Auto-generated method stub

if (pd != null && pd.isShowing()) {

pd.dismiss();

}

ArrayList<String> region\_list = new ArrayList<String>();

try {

JSONArray j\_arr = new JSONArray(success\_response);

for (int j = 0; j < j\_arr.length(); j++) {

JSONObject j\_obj = j\_arr.getJSONObject(j);

region\_list.add(j\_obj.getString("region"));

}

ArrayAdapter<String> adapter = new ArrayAdapter<String>(

RegistrationActivity.this,

android.R.layout.simple\_spinner\_dropdown\_item,

region\_list);

reg\_spinner\_region.setAdapter(adapter);

} catch (JSONException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

});

List<NameValuePair> nameValuePair = new ArrayList<NameValuePair>();

nameValuePair.add(new BasicNameValuePair("flag", "getRegions"));

requestMethod.postMethod(nameValuePair);

}

}