



JARAMOGI OGINGA ODINGA UNIVERSITY OF SCIENCE AND
TECHNOLOGY

BSc. COMPUTER SECURITY AND FORENSICS

**AUTOMATED POLICE RECORD MANAGEMENT
SYSTEM (PRMS)**

Evans Chumba

1132/0882/2013

Ephantus Githua

1132/0890/2013

A project submitted to School of Informatics and Innovation Systems for the
degree in Computer Security and Forensics at Jaramogi Oginga Odinga
University of science and Technology
(JOOUST)

2016

DECLARATION

We declare that this project is our original work and has been submitted to Jaramogi Oginga Odinga University of science and technology for examination or award of a degree.

Signed..... Date.....
Evans Chumba

Signed..... Date.....
Ephantus Githua

This project has been submitted for Examination with my authority as the University supervisor

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

Dr. Solomon Ogara
Senior Lecturer, School of Informatics and Innovative systems
Jaramogi Oginga Odinga University of science and Technology

DEDICATION

EVANS CHUMBA

I dedicate the entire work carried here in to my Dad Mr. Gilbert Bartenge, my Mum Mrs. Nancy Mwei for their inspiration all through my course. I would like to appreciate all my lecturers who instilled in me lifelong values and the desire for education. Finally I would like to appreciate all of you who have contributed to my success and have not been mentioned above, remember that all your efforts are highly appreciated and you will never be forgotten for your stake in my life, God bless you all.

EPHANTUS GITHUA

I dedicate this work to my mother Mrs. Lucy Wangui Gichohi and my father, the late Mr. Samson Gichohi, who their love for education has seen me come this far, my Siblings, Emily Gacheru and Jane Wanjiku and other family members who ever gave me moral support.

It's hard to forget all the Staff from the school of informatics and innovative systems in Jaramogi University for their tireless efforts in making our course a success.

ACKNOWLEDGEMENT

We would like to acknowledge God Almighty for the strength, sound mind and provision during the period of study.

Special thanks to our supervisor Dr. Ogara and Moderator Professor Rodrigues for their immense support, guidance and patience, and without their constructive criticism and advice this work would not have been complete.

We also register our gratitude to our parents and siblings for their continuous encouragement during this journey, my university Administration (Jaramogi University) for providing an enabling environment for us to complete the course and access the labs during weekends and at night.

Also special regards to Inspector Boniface Gacheru (Administration Police Officer) and Constable Judy Cherop (Kenya police officer) for their help in understanding various police formulations and categories.

Thank you to all my friends who contributed to the completion of this academic document both directly and indirectly. They provided me with logistical and moral support that gave us every reason to work harder and ensure that this study becomes a success.

To all of you, God bless and increase you immensely.

ABBREVIATIONS & ACRONYMS

API: application programming interface
DFD: data flow diagrams
MRO: Materials Repair and Operational
NPS: National Police Commission
PDA: personal device assistant
PRMS: police record management system
SDLC: system development lifecycle

ABSTRACT

This project explored the relevance of management systems to the contemporary society where computers have made tasks easier, more efficient and organized. Police record management system is a planned system of the storage processing and dissemination of data in the form of information needed to carry out various functions which come to simplify the tiresome manual way of reporting and writing the details on papers. There is need for an automated record management system that can be used to store reported criminal activities in a centralized database which should replace the manual system that is currently being used. The manual system has few challenges that include the ever increasing paper load, difficulty in enforcing access control as well as cases of missing files and information.

Our objective in regard to the project is to come up with an application that will be used to automate the manual management of Kenyan criminal records and hence the title “police management system (PRMS)”. Being a web based application; there is an assumption of users being connected to the internet.

The scope of the project cuts across administrators and 2 types of clients; police officers and other clients: including private investigators, lawyers and other personalities seem fit by the Administrator to handle an account.

The research methodology used in this project was System development lifecycle, using waterfall method simply because the project was small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project

The design was majorly aided by the dataflow context diagrams which are often used as a preliminary step to create an overview of the system. Use-case diagrams and activity diagrams also helped in coming up with the database class diagram which we used to develop the database.

TABLE OF CONTENTS

DECLARATION.....	i
DEDICATION	ii
ACKNOWLEDGEMENT	iii
ABBREVIATIONS & ACRONYMS	iv
ABSTRACT.....	v
TABLE OF CONTENTS.....	vi
LIST OF FIGURES	xi
LIST OF TABLES	xi
ABSTRACT.....	xiii
CHAPTER 1.....	1
1.0 INTRODUCTION	1
1.1 Background of the study.....	1
1.2 Problem statement.....	1
1.3 Project objectives	2
1.3.1 Main objectives.....	2
1.3.2 Specific objectives	2
1.4 Justification and significance of the system	2
1.5 Assumptions.....	3
1.6 Scope of the study	3
CHAPTER 2.....	4
2.0 LITERATURE REVIEW.....	4

CHAPTER 3.....	5
3.0 RESEARCH METHODOLOGY	5
3.1 Introduction.....	5
3.2 System development methodology	5
3.1.1 System Planning	6
3.1.2 System Analysis.....	6
3.1.3 System Design	6
3.1.4 System Coding and Implementation.....	6
3.1.5 System Testing	7
3.1.6 Deployment of system	7
3.3 Reason for using Waterfall Model	7
CHAPTER 4.....	8
4.0 DESIGN AND CODING.....	8
4.1 Introduction.....	8
4.2. Functional Requirements.....	8
4.2.2 Register a new user	8
4.2.3 Ban user.....	8
4.2.4 Add/edit/delete police Category and police formulation	8
4.2.5 Add Station.....	9
4.2.6 Add criminal.....	9
4.2.7 Report Cases.....	9
4.2 Non-functional Requirements	9

4.2.1 Reliability requirements	9
4.2.2 Usability requirements	9
4.2.3 Implementation requirements	9
4.2.4 Security Requirements	10
4.3 System Analysis	10
4.3.1 Context Dataflow Diagram	10
4.4 Use case diagram.....	11
4.4.1 Admin Use case Diagram.....	12
4.4.2 Users' use case diagram	13
4.5 Activity diagrams	13
4.5.1 Admin activity diagram.....	14
4.5.2 Police data flow diagram.....	15
4.5.3 Client Data Flow Diagram	16
4.6 System Class diagram.....	17
CHAPTER 5.....	19
5.0 TESTING AND RESULTS	19
5.1 Introduction.....	19
Requirements Analysis and Test Cases	22
5.2 Functional Requirements	22
5.2.1 User Login.....	22
5.2.2 Register a new police/Agent.....	23
CHAPTER 6.....	24

6.0 IMPLEMENTATION AND DEPLOYMENT	24
6.1 Implementation	24
Introduction	24
6.2 User Interfaces	24
6.2.1 Users login window	24
6.2.2 Admin Dashboard	25
6.2.3 Police Dashboard	26
6.2.4 Client Dashboard	27
6.2.5 Register New Admin, Police and Client Window	27
6.3 Deployment.....	28
6.3.1 Phased Approach	28
CHAPTER 7.....	29
7.0 DISCUSSION.....	29
7.1 SECURITY ASPECTS IN RELEVANCE TO PROJECT	29
7.2 Future Improvements	29
CHAPTER 8.....	30
8.0 CONCLUSIONS AND RECOMMENDATIONS	30
8.1 Discussion.....	30
8.2 Limitations	30
8.2.1 Information Source	30
8.2.2 Financial Incapacitation	30
8.2.3 Learning and implementing a new language.....	30

8.3 Recommendations	31
8.3.1 Provision of project finances to the students.....	31
8.3.2 Compelling some institutions to pave way for the students to develop.....	31
8.4 Conclusion	31
REFERENCES.....	32
APPENDICES.....	33
APPENDIX A.....	33
Work Plan	33
APPENDIX B.....	34
Police Record Management System Budget.....	34
APPENDIX C.....	35
Work Breakdown Structure (As A Project).....	35
APPENDIX D.....	36
System Specification Requirement.....	36
APPENDIX E.....	37
Constant helpers codes	37
Authentication Source Codes.....	37
Dashboard source codes.....	38
Database_handlers source codes	41

LIST OF FIGURES

Figure 1.1: system development life cycle	5
Figure 3.2: Gantt chart diagram showing our work plan in SDLC methodology	7
Figure 4.1: Context dataflow diagram	11
Figure 4.3 Clients use-case diagram	12
Figure 4.2: Admin use-case diagram	12
Figure 4.4 Admin Activity diagram.....	14
Figure 4.5 police data flow diagram.....	15
Figure 4.6 client data flow diagram	16
Figure 4.7 class diagram	18
Table 5.1: user's System details	19
Table 5.2: police station table details	19
Table 5.3: police station table details	20
Table 5. 4: police station table details	20
Table 5.5: police category table details.	21
Table 5.6: criminal table details.....	21
Table 5.7: county table details.....	21
Table 5.8: county table details.....	22
Figure 5.1: Wrong credentials window	22
Figure 5.2: Checking Empty Fields.....	23
Figure 6.1: Login window.....	24
Figure 6.2: Administrator Operation Window Appearance.....	25
Figure 3.3: Police Operation Window Appearance.	26
Figure 6.4: police officer Operation Window Appearance	27
Figure 6.5: Registering new system user.....	27

LIST OF TABLES

Table 5.1: user's System details	19
Table 5.2: police station table details	19

Table 5.3: police station table details 20
Table 5. 4: police station table details 20
Table 5.5: police category table details. 21
Table 5.6: criminal table details..... 21
Table 5.7: county table details 21
Table 5.8: county table details 22

ABSTRACT

This project explored the relevance of management systems to the contemporary society where computers have made tasks easier, more efficient and organized. Police record management system is a planned system of the storage processing and dissemination of data in the form of information needed to carry out various functions which come to simplify the tiresome manual way of reporting and writing the details on papers. There is need for an automated record management system that can be used to store reported criminal activities in a centralized database which should replace the manual system that is currently being used. The manual system has few challenges that include the ever increasing paper load, difficulty in enforcing access control as well as cases of missing files and information.

Our objective in regard to the project is to come up with an application that will be used to automate the manual management of Kenyan criminal records and hence the title “police management system (PRMS)”. Being a web based application; there is an assumption of users being connected to the internet.

The scope of the project cuts across administrators and 2 types of clients; police officers and other clients; including private investigators, lawyers and other personalities seem fit by the Administrator to handle an account.

It targets all the police stations and posts in all the 47 counties of the country and covering all the police stations and posts there is in every ward.

The research methodology used in this project was System development lifecycle, using waterfall method simply because the project was small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project

The design was majorly aided by the dataflow context diagrams which are often used as a preliminary step to create an overview of the system. Use-case diagrams and activity diagrams also helped in coming up with the database class diagram which we used to develop the database.

We faced some challenges in developing the systems in terms of having to learn new programming language on a short time frame and also experienced financial constraints on printing and buying of some devices that would have helped us come up with better system like fingerprint scanner.

In conclusion we would like to say that taking this project all through has been a wonderful experience for us. Without the practical knowledge that we acquired throughout the course, this project would not have materialized.

CHAPTER 1

1.0 INTRODUCTION

The use of computer today is in peak point and each and every field uses computer system for the fast processing and security of the data. Management systems are one of the fields where computers have made tasks easier, more efficient and organized. Police record management system is a planned system of the storage processing and dissemination of data in the form of information needed to carry out various functions. Simply this system ensure that the criminal activities reported by the police and any other clients authorized to use the system are feed into the system in real time while in the same time, maintaining the Confidentiality and Availability of data in the system.

1.1 Background of the study

In the technological advancement bid, computers have greatly contributed to the simplification of the way things are done and in return have contributed to the automation of the various activities that take place in and around our day to day activities. According to the way technology has been advancing and evolving, the use of computer and internet technology is slowly replacing some aspects of tedious paper work which can be implemented by using an online automated system. Now that many systems have been automated in Kenya, the security sector should not be left behind. Many systems so far have been developed in relation to security issues. Closed Circuit Television (CCTV) cameras have been put in place to monitor everything, but no system has been made to automate the record of criminal activities reported by the police and their agents which is high concern in the Kenyan government. The overall objective of this proposal was to develop an application that is sufficient to automate the police information of criminal case reported.

1.2 Problem statement

Kenya as a nation needs a platform as far as security sector is concerned. This covers automation of police record management system in Kenya police, and all other security organs country wide. Considering the Kenya government, we need automated record management system that can be used to store reported criminal activities in a centralized place. The manual system that is currently used is getting challenged by few problems it has. The current system poses problem that include the ever increasing paper load, difficulty in enforcing access control as well as cases of missing files and information.

Currently all operations for managing day to day activities in Kenya police department are not computerized. These results in the following problems;

1. Increased in criminal activities due to delay in response time from the police department. The common Kenyan (Mwananchi) currently lacks platform to report criminal activities encountered on different parts of the country.
2. Missing files containing criminal reports. This result in criminal going unpunished when hearing a case in the court of law. Missing files means missing crime report collected at the crime scene.
3. Lack of effective coordination between the different national security organs.

Based on the above drawbacks the current system has been proposed.

1.3 Project objectives

1.3.1 Main objectives

To develop an automated police record management system.

The main objective of the project is to come up with an application that will be used to automate the manual management of Kenyan police records, the application is web based thus will be access anywhere provided you are connected to the internet.

1.3.2 Specific objectives

1. To develop a system that will store and manage crime and criminal records in a centralized location. This will be achieved since the system uses one database with relation of tables.
2. To develop a system that will ensure that access to the police sensitive information is managed. This means that only authorized users (those in possession of login credentials provided to them by them the system admin) can gain access to the system.
3. To develop system that ensures that criminal activities are reported in real times. This is achieved since the system will enables the police and clients to reported cases anywhere in real time.

1.4 Justification and significance of the system

Upon the completion of design of the system, the various users of the system will benefit from the system in the following ways:

1. Improvement of security since the system unsure that criminal activities are reported in real time.
2. The system will eliminate corruption that has been experience in the Kenyan court when finding files of criminal records in police departments to be used in court of law as evidence.
3. The system will reduce the incidences of missing files since the system replaces the current manual system that is much of paper.

1.5 Assumptions

1. The system has the assumption that the system users are connected to internet
2. All system users have the knowledge of using smart phone, tablets, PDA and computer.
3. There is network connection wherever the system users are.

1.6 Scope of the study

Generally, the idea of police record management should be of national concern, since the Kenyan police sensitive information pertaining criminal's records should be kept in a centralized and secure database. The system is meant to capture all the details required for a criminal reported, criminal activities, and retrieve them when required. The users of the system will be police, police undercover agents, and administrator from the different security organs who has all the privileges to manipulate the data in the system. It should also target all the police stations and posts in all the 47 counties of the country and covering all the police officers who are approximated to be 80,000-100,000 police officers. It should also be scalable to accommodate around 10,000 private investigators in the country.

CHAPTER 2

2.0 LITERATURE REVIEW

Computer security is the field that tries to keep information confidential and with high level of integrity while at the same time make it available to individuals who access the data sensitive information. Part of this include authentication and validation (making sure you are who you claim to be), encryption (making sure data gets where you want to go, without others being able to understand it) and physical security.

In the modern high-tech world, information is so crucial that every aspect of our lives is based on it. Values placed on information in the present generation cannot be underestimated. Almost all information is electronically stored and a large percentage of it is stored in a computer system.

According to the report by the national police service commission (Serianu, 2014) a number of the security lapses have been experienced during the period under review can be attributed to the lack of effective coordination between the different national security organs. Through interviews with various officers within the National Police Service (NPS), the Commission established that there are serious lapses of coordination within the service. Lack of operational congruence between the different national security organs has affected information sharing and the carrying out of joint policing operations.

The Kenya Police Records Management System is a system that will provide for the storage, retrieval, Retention, manipulation, archiving, and viewing of information, records, documents, or files pertaining To law enforcement operations.

The system covers the entire life span of records development—from the initial record generation to Record completion. Moreover, the system allows single entry of data, while supporting multiple Reporting mechanisms. The records include: incident and accident reports, arrests, citations, warrants, Case management and field contacts. There is no existing record management in Kenya but research has been done Raptor Technology limited showing the need of having a record management system in the police sector. The most unique feature they came up with was the Volume, rate, and trends of crime indicators in their proposed system. Each statistic meant to provide a different perspective of the crime experience. (Raptor, 2016)

No country in east and central Africa using an Automated Police record system due to the huge financial requirements and lack of enough experts to maintain and handle the systems.

CHAPTER 3

3.0 RESEARCH METHODOLOGY

3.1 Introduction

The project methodology used in the development of the system is the System Development Life Cycle (SDLC) specifically following the waterfall method. The SDLC is the process of understanding how information system can valid to the user needs, then designing the system, building it and delivering it to the potential users. This methodology is composed of some phases. The structured design methodology will be waterfall development.

3.2 System development methodology

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of model is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model the testing starts only after the development is complete

In the course of developing the system we chose to use the Waterfall Development Model as illustrated in the figure, below.

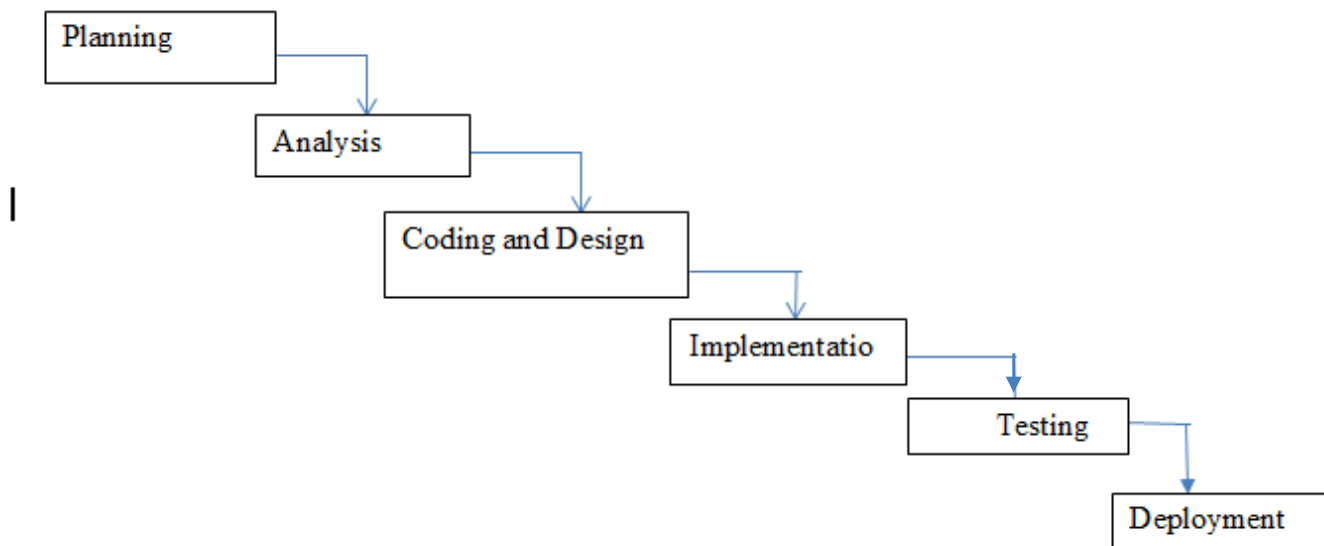


Figure 1.1: system development life cycle

3.1.1 System Planning

In planning phase which was first step was to identify needs for police record managements system. This was first phase of the system and it entailed determining the necessary information that was required for the system to be fully operational as well as function in the expected manner. The requirements that we captured were subjected to thorough scrutiny to determine the level of essence of it as well as eliminate the unwanted requirements. We did this based on the fact that though the requirements might have been raised by the target users found through our research, they might not be realistic or might not be so much important.

3.1.2 System Analysis

In this phase, we analyzed and considered the current systems and investigated any problems associated with it. Other sources of information about system and the new requirements were also investigated at this time. The important information from the planning phase was highly used in this phase, and the valid information gathered was analyzed for the design stage.

3.1.3 System Design

After the requirements, having already been captured and analyzed, the design of the information flow was done to show the logical structure of the system. It is in this phase that the use case, dataflow diagrams and entity relationship diagrams were drawn to show flow of information and the activity diagrams were developed to show the connection that will exist between one party to another. Details are discussed in our next chapter 4.

3.1.4 System Coding and Implementation

After the design of the interfaces as well as the indication of the information flow through the Sequence diagrams and the flowcharts algorithms, the next step was to develop code abstracting the functionality of the system to realize the already fore-mentioned set objectives. The code was in relation to the information flow and therefore the flowcharts developed earlier were now applied here. The requirements documentation was referred throughout the rest of the system development process to ensure the developing project aligns with the needs and requirements or scope. A proper execution of the previous stages ensured an easier realization of this phase in the course of my development. Upon completion of the coding, the various components of the system were then integrated in to one system in order to function collectively as a single component. This is also covered in detail in chapter 4

3.1.5 System Testing

Last phase is system testing done when development is complete and the system is ready for deployment. The testing phase come next to determine if the earlier intended objective to have been realized by then. Testing was done based on whether completeness will have been realized or functional testing that determined whether the software is doing what it is expected correctly and in the right way. User testing was then carried out by a group of Informatics students to ascertain that the users will be contented with what will have been achieved then and this has been discussed in detail in chapter 5.

3.1.6 Deployment of system

Ones the system has undergone testing and passed all the validations and verifications, then the system would be termed to be acceptable and therefore it will be handed over to the owners for maintenance and operation as well.

3.3 Reason for using Waterfall Model

- ❖ It is easy to understand and follow as it provides a clear guideline.
- ❖ Clearly define stages and processes and results are well documented.
- ❖ Simple and easy to understand and use.
- ❖ Works well for smaller projects where requirement are very well understood.
- ❖ In this model phases are processed and completed one at a time. Phases do not overlap.



Figure3.2: Gantt chart diagram showing our work plan in SDLC methodology

CHAPTER 4

4.0 DESIGN AND CODING

4.1 Introduction

Upon the completion of the Police Record Management System, there are a number of things that will be expected of it not only by the prospected users but also for the administrator of the system. These will therefore form the requirements of the Police Record Management System and will be broadly classified in to the functional requirements and the Non-functional requirements.

4.2. Functional Requirements

4.2.1 User Login

This feature is used for the 2 types of users (police and Client) to login in to the system. Where client in this case can be a lawyer, undercover police agents or private investigators. They are required to key in the Email Address and the password before they are granted permission to enter the system. The username and the password will be verified and the invalid username and password are not allowed to enter the system. All users are registered by the administrator. The system must only allow the user with valid username and password to enter the system.

4.2.2 Register a new user

This feature is used by the administrator to register new system users to ensure that all details are available and valid. System ensures that the information supplied by the user (Admin) is of the correct form through the input type of the registration forms.

4.2.3 Ban user

This feature allows the administrator to user from using the system. This functionality is only meant for system administrator and other system user can access this functionality. System searches the record from the database and displays the entire details to the user. System must be able to show out the correct information about particular user.

4.2.4 Add/edit/delete police Category and police formulation

This feature will allow the admin user to add, edit and delete the records of other police category and police formulation in the database. Admin selects the particular police category or police formulation from the records to view the details to be edited. The System responds by displaying the entire details of that particular category or formulation. The Admin then edits the records as required and updates. The admin is also able to delete police category and police formulation from the police category and police formulation system database tables. System

must be able to fetch the correct details from the database and display them to the user. System must inform the system admin of the update success once the record has successfully updated.

4.2.5 Add Station

This feature allows the administrator to add stations in the system. Once the police station details has been added to the system database, the admin is also able to edit the details and delete the details from the system database. System searches the record from the database and displays the entire details to the system user. System must be able to show out the correct information about particular police station.

4.2.6 Add criminal

This feature allows all the system users to add criminal's records into the system. Once criminal's records have been added to the system, the system admin has the privilege to edit, upload the criminal's picture and deleting the criminal's information in the system.

4.2.7 Report Cases

This feature is used by all system users (admin, police, and client). This feature allows the system users to feed reported case incidences into the system. Once the case information has been stored in the system's database, the system admin has the privilege of retrieving the case information, edit and can also delete the case information from the system database.

4.2 Non-functional Requirements

4.2.1 Reliability requirements

The system must perform accurately towards the administrator request. For example, when the administrator saves the edited details of the user, after he reviews the details later, they must be changed according to the latest details that was updated. Moreover, the agent is not allowed to view the details that the administrator has. Besides that, the login form will have validity check to ensure that only the authorized users gain access to the system.

4.2.2 Usability requirements

This system must be designed with user-friendly and easy to use by all the users so that they can perform their tasks easily.

4.2.3 Implementation requirements

In implementing the system, it uses pup as the main programming language. This forms the front-end and the middleware.

At the back-end, MYSQL will be synchronized and be used to maintain the information in the database. This is formed by the databases and other data stores.

4.2.4 Security Requirements

This system authentication protocol and access control must be highly secured in the login part. Where username identities and passwords are encrypted and hashed using MD5 function. This is because some privileges are only meant for the System Administrator only. Meaning that if the security is compromised, the whole system is compromised. Our Database is also hashed using both MD5 and SHA-1 encryption algorithms functions.

4.3 System Analysis

In the system analysis we chose to use the Object Oriented System Analysis. This is due the fact that our system employed the object-oriented approach and due to the use of a php programming, the use of the object-oriented System Analysis was and appeared to be more appropriate than the other approaches that exist towards the same. (Lerdorf R.& MacIntyre, P, 2006) The diagrams that illustrate the systems flow of information are shown below:

4.3.1 Context Dataflow Diagram

A data flow diagram is a graphical representation of the flow of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing.

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

The figure below shows the flow of information in the Police record management system which entail three users of the system and these are the administrator, the police and the client.

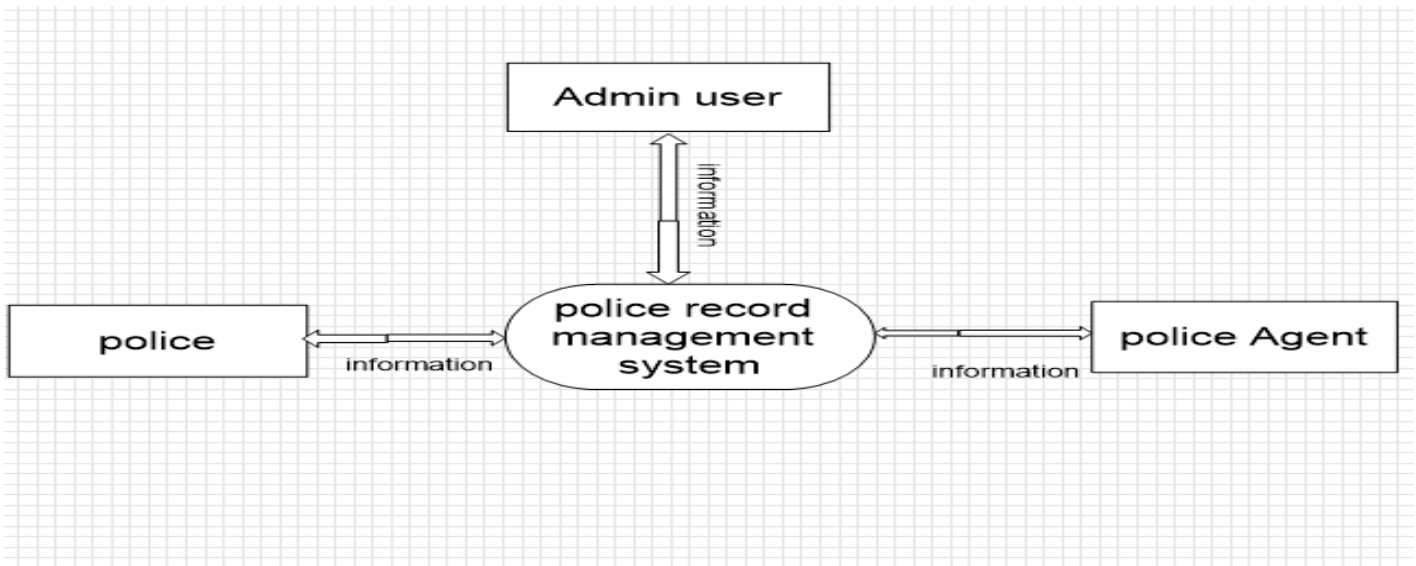


Figure 4.1: Context dataflow diagram

TIP: Client can be a private investigator or an undercover agent.

The Police and client can interact with the system when reporting cases, viewing most wanted criminals, adding criminals, viewing closed and open cases. The system admin users adds criminals, view and add cases, add case type and more importantly add system user.

4.4 Use case diagram

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified. The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose.

4.4.1 Admin Use case Diagram

The below use case diagram is for the police and clients. It shows the privileges accorded to each of them.

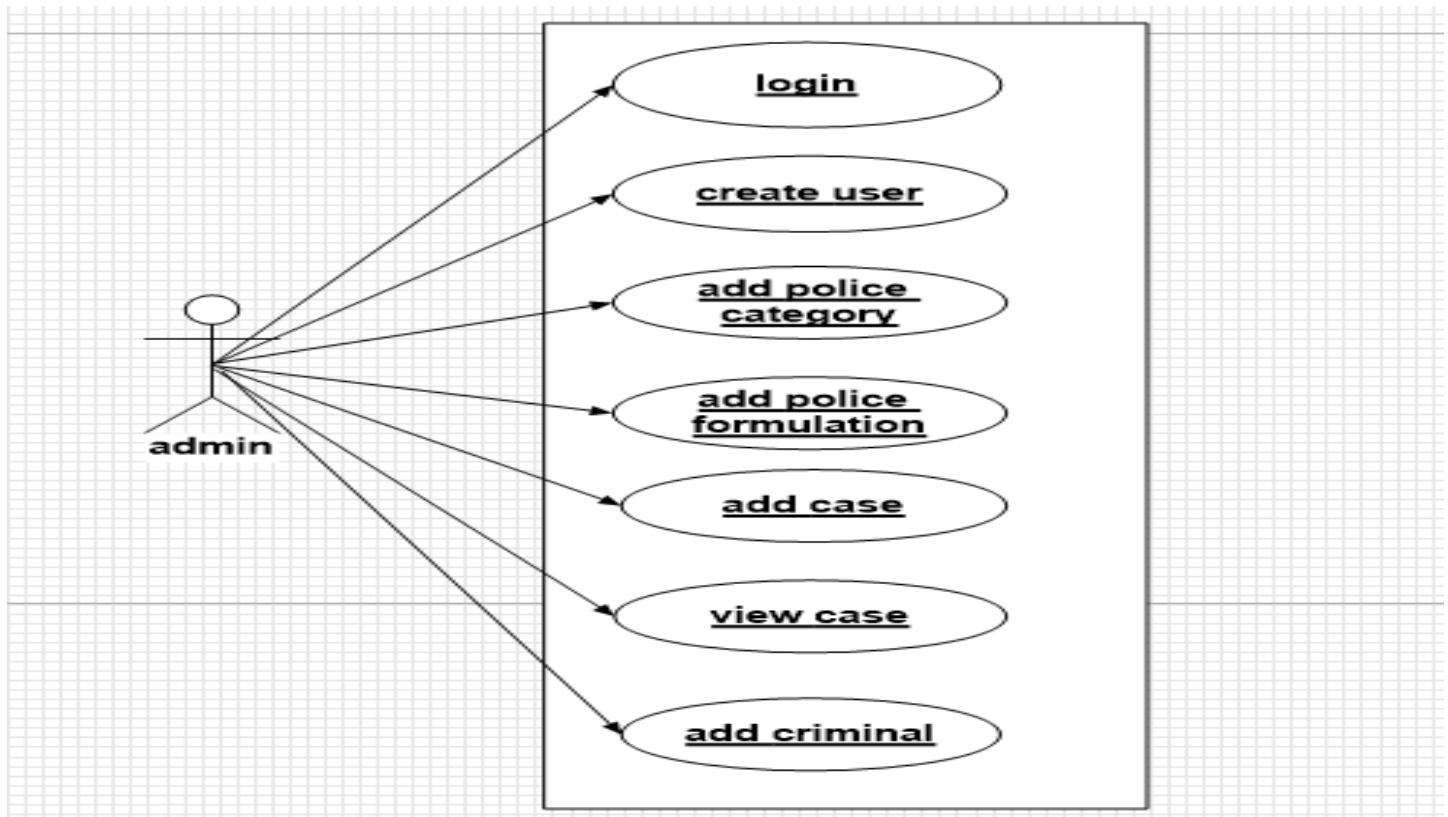
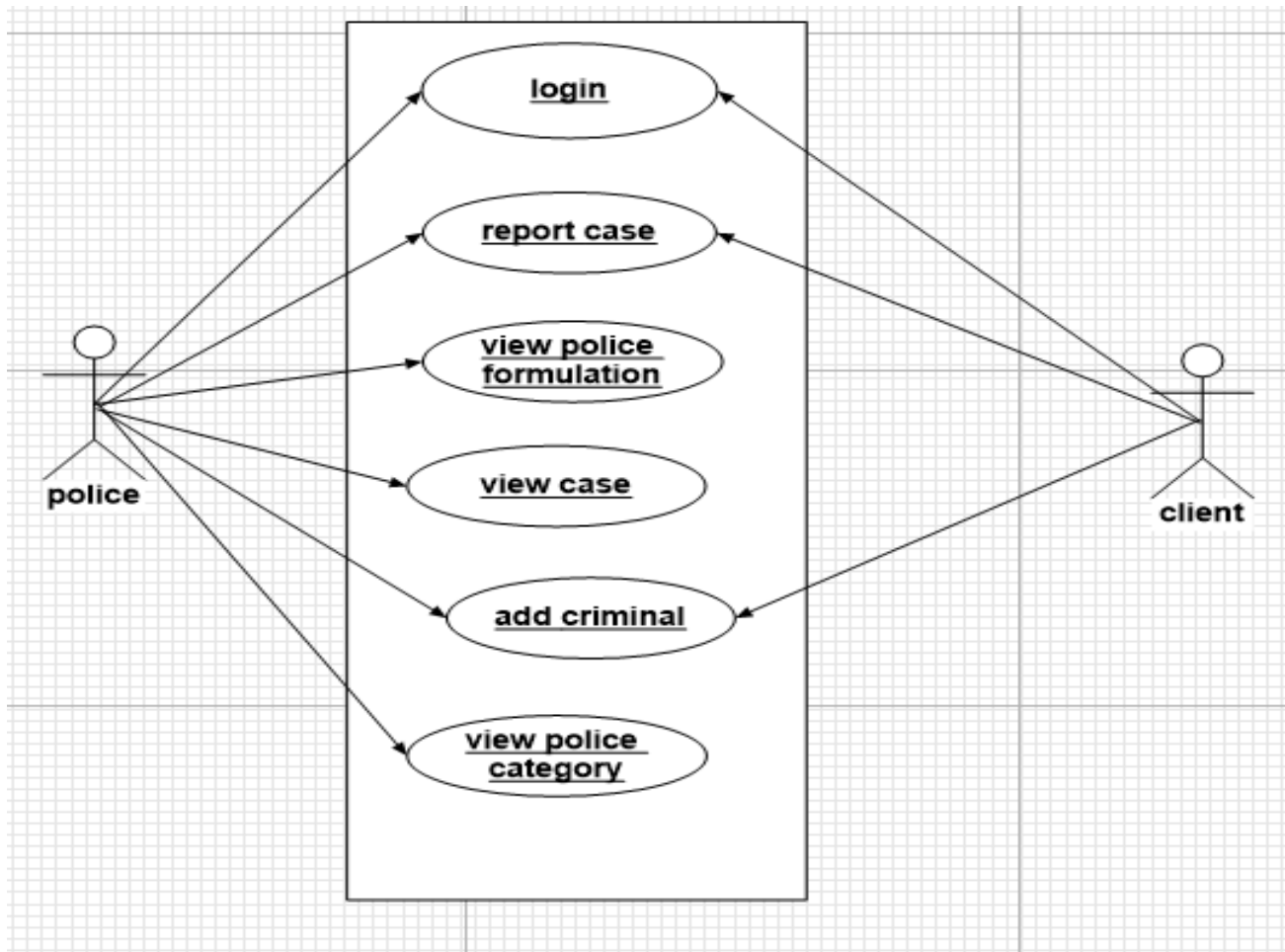


Figure 4.3 Clients use-case diagram

Figure 4.2: Admin use-case diagram

The below diagram explains the functions the administrator is required by the system to perform. The system should allow the administrator to login, register new users of the system who are the students and all employees, add case type, ban users, report case, open and close cases, He/ She should have the privilege to update their details, view all the registered members. Finally should also allow him/her to remove any member who exits to be part of the fraternity.

4.4.2 Users' use case diagram



4.5 Activity diagrams

Activity diagrams capture the dynamic behavior of the system to show message flow from one activity to another. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deal with all types of flow control by using different elements like fork, join etc. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in an activity diagram is the message part. It does not show any message flow from one activity to another.

4.5.1 Admin activity diagram

The below diagram shows all the activities/functions of the administrator though they are combined in one diagram. According to the diagram, administrator should first login to the system. If the username and password are correct, the system allows him to proceed. Then as shown by the diagram, he can create user, add police category, add police formulation, add case type, add reported case add criminal, open and close a case.

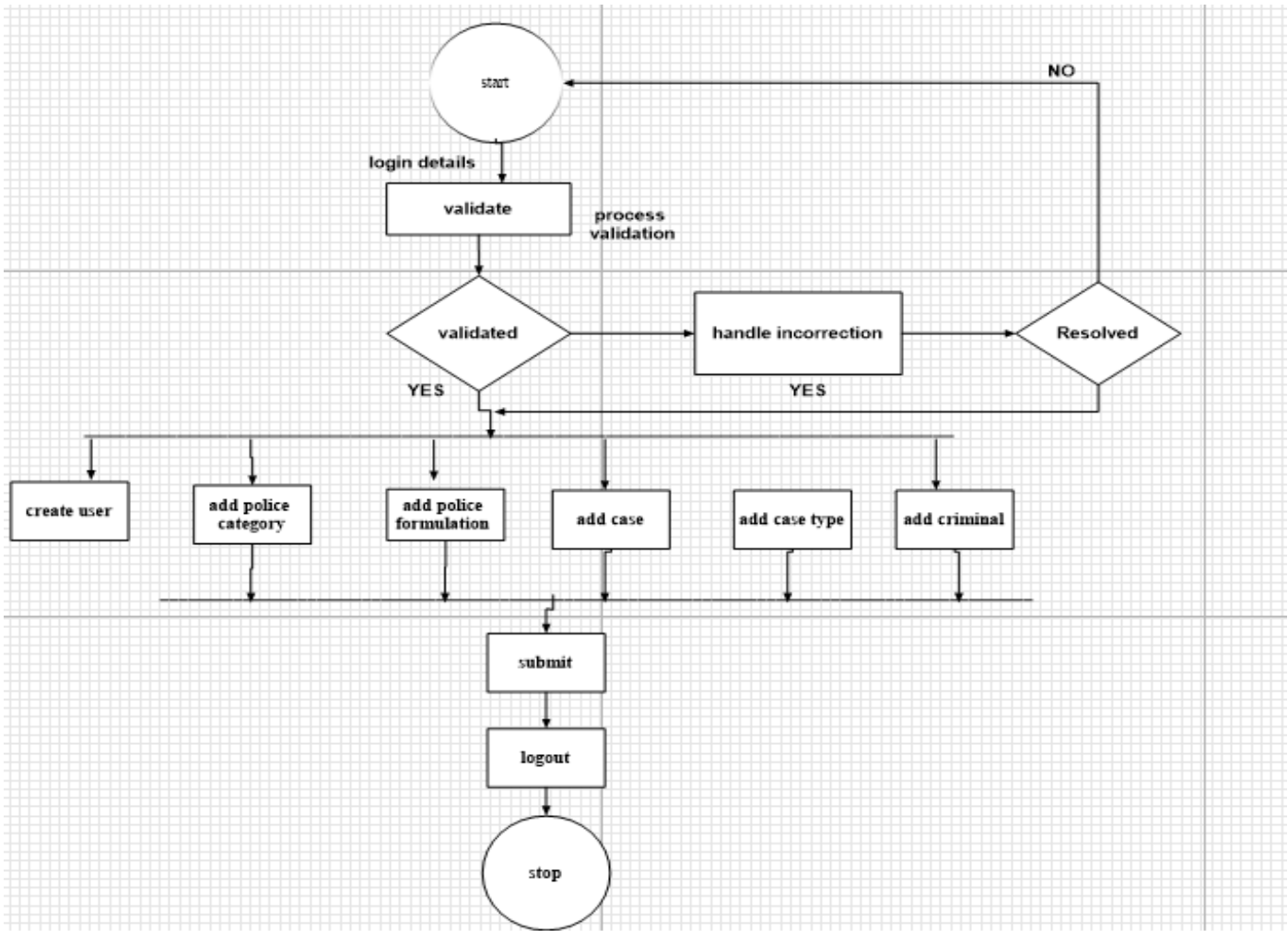


Figure 4.4 Admin Activity diagram

4.5.2 Police data flow diagram

The diagram below shows the flow of activities when the police login in to the system. If he is registered, then the system allows him to view the reported cases details, view police category, view police formulation, report cases and add criminal to the system.

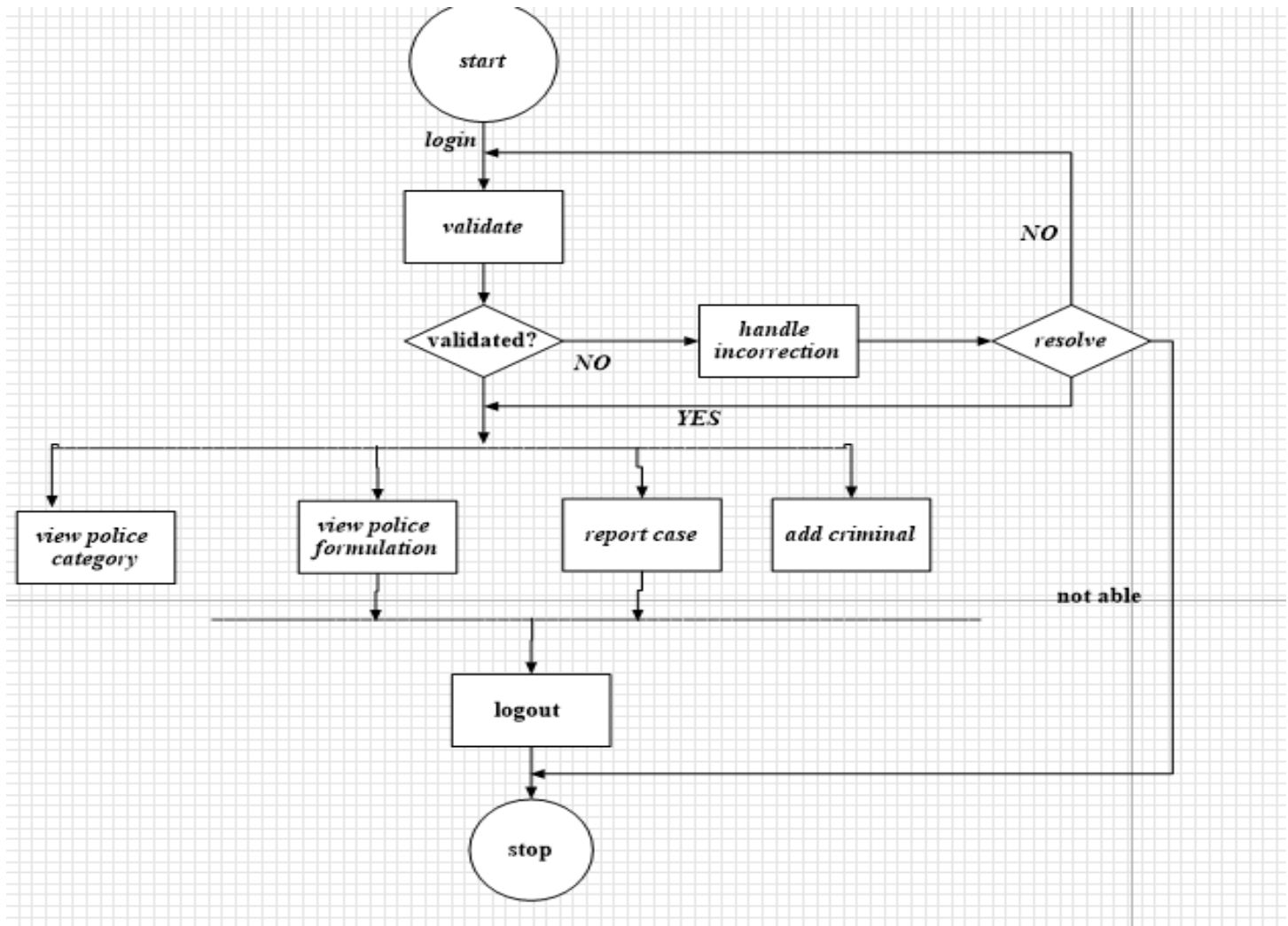


Figure 4.5 police data flow diagram

4.5.3 Client Data Flow Diagram

The diagram below shows the flow of activities when the client login in to the system. If he is registered, then the system allows him to view the reported cases details, add criminal and report case.

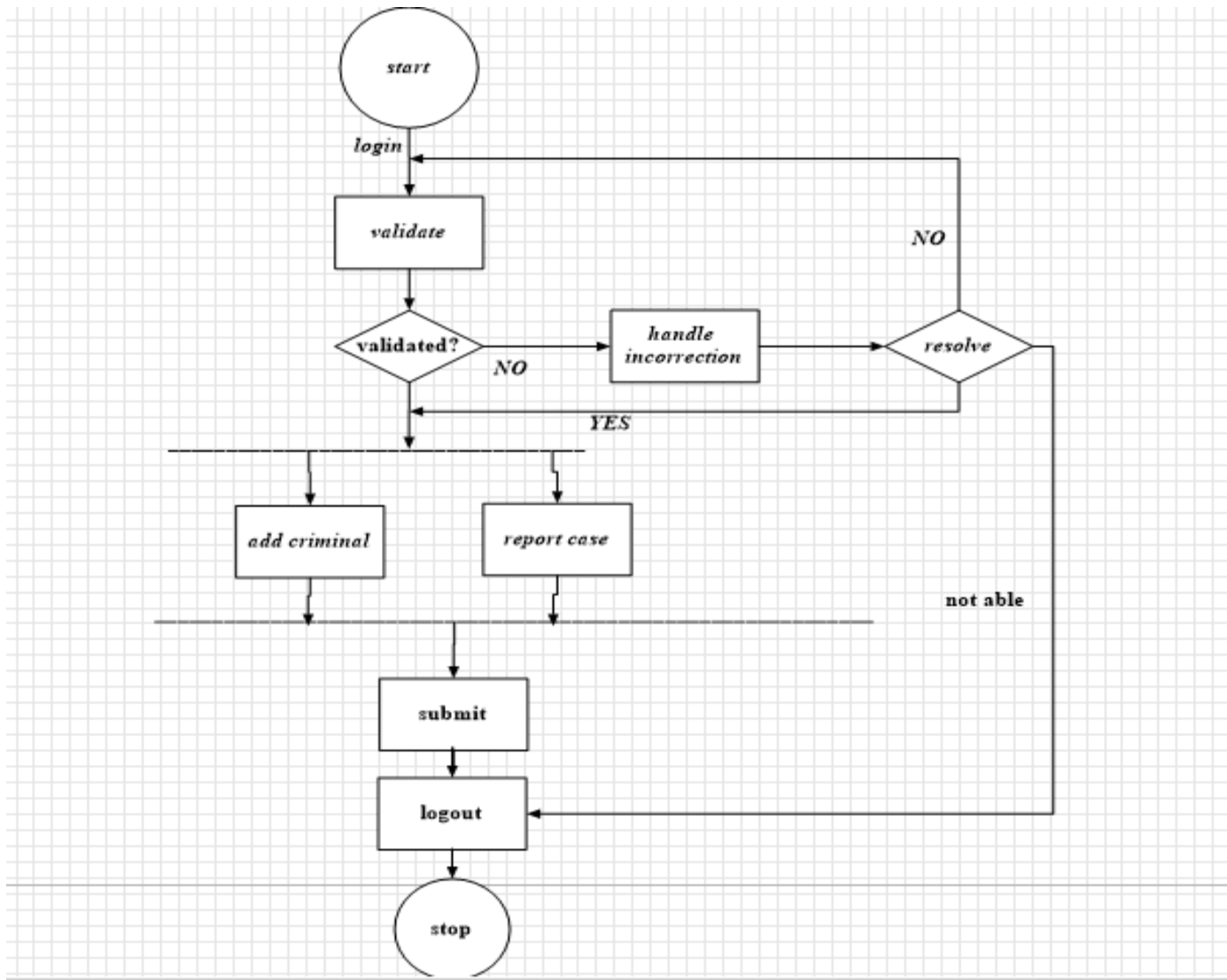


Figure 4.6 client data flow diagram

4.6 System Class diagram

The class diagram models the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. Class diagram is basically a graphical representation of the static view of the system and represents different tables of the systems database of the application. So a collection of class diagrams represent the whole system.

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only class diagrams which can be mapped directly with object oriented languages. The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction.

The class diagrams like activity diagram, sequence diagram can only give the sequence flow of the application but class diagram is a bit different. So it is the most popular class diagram in the coder community. So the purpose of the class diagram can be summarized as analysis and design of the static view of an application, describe responsibilities of a system, base for component and deployment diagrams.

Below is our systems Class Diagram.

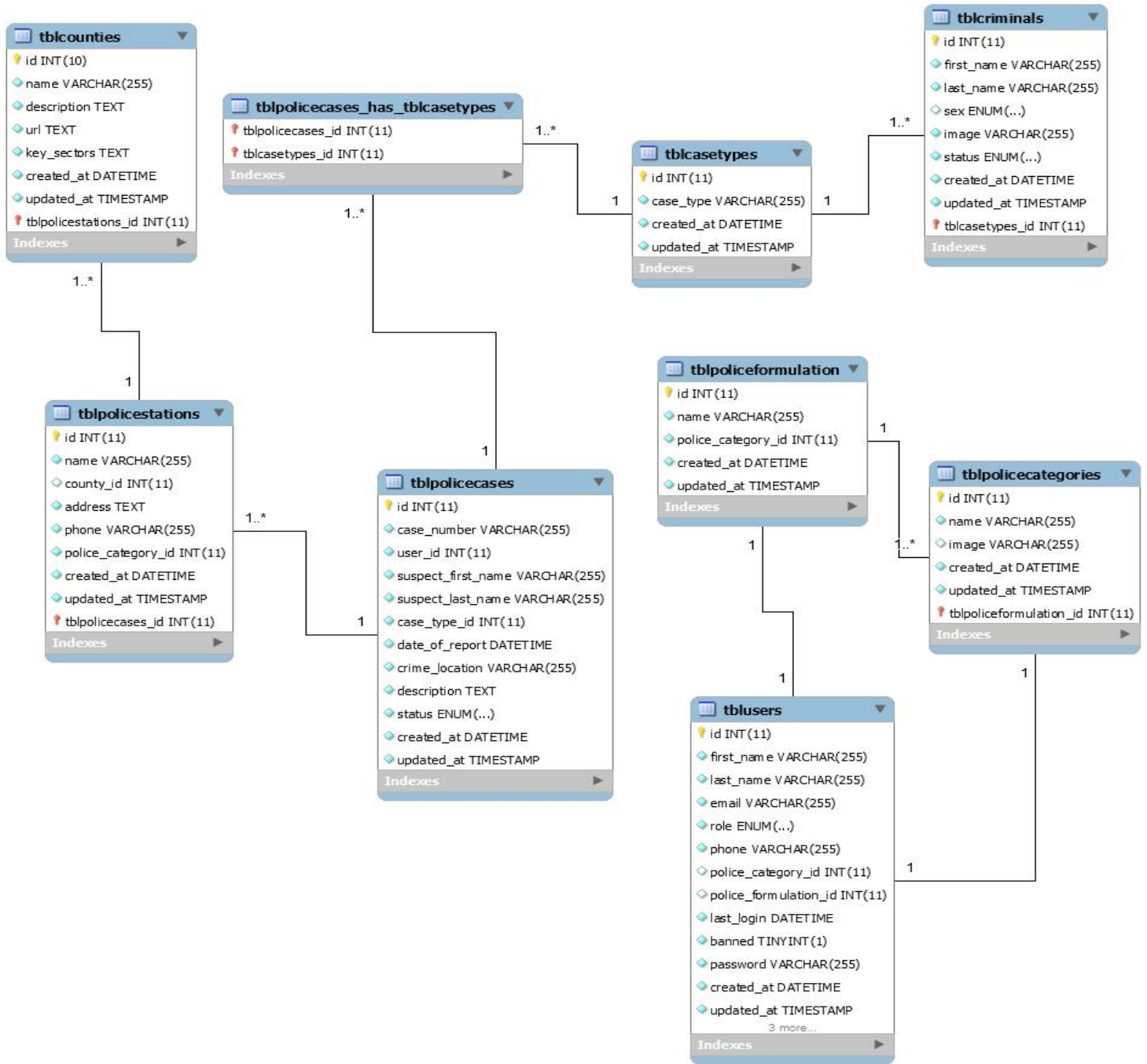


Figure 4.7 class diagram

CHAPTER 5

5.0 TESTING AND RESULTS

5.1 Introduction

Having designed and coded the system according to the user's needs. Dummy data was fed into the system so as to test its functionalities and any possible errors.

And this is how the data was stored in the system is shown in the following tables:

User's Table Details

Table 5.1: user's System details

FIELD NAME	DATA TYPE	DESCRIPTION
id	integer	user id
first_name	varchar	User first name
last_name	varchar	User last name
email	varchar	User email address
role	varchar	User role in the system(admin, police, client)
phone	integer	User phone number
police_category_id	integer	Random police category id
police_formulation_id	integer	Random police formulation id
last_logged_in	datetime	Time the user last logged in
banned	varchar	Banned from using the system

This user's table details above shows the data that is to be stored in the users table. It shows the field name, the data type and the description of the data.

Police station table details

Table 5.2: police station table details

FIELD NAME	DATA TYPE	DESCRIPTION
id	integer	police station id
name	varchar	Police station name
County_id	integer	Random county id
address	varchar	Address of the police station
phone	varchar	Police station phone number
Police_category_id	integer	Random police category id
Created_at	date time	Data insertion date time
Update_at	timestamp	Data updated time

This police station table details above shows the data that is to be stored in the police station table. It shows the field name, the data type and the description of the data.

Police formulation table details

Table 5.3: police station table details

FIELD NAME	DATA TYPE	DESCRIPTION
id	interger	police formulation id
name	varchar	Police formulation name
Police_category_id	interger	Random police category id
Created_at	datetime	Data insertion datetime
Updated_at	timestamp	Data updated time

This police formulation table details shows the data that is to be stored in the police formulation table. It shows the field name, the data type and the description of the data

Police cases table details

Table5. 4: police station table details

FIELD NAME	DATA TYPE	DESCRIPTION
id	interger	Case id
Case_number	interger	Random case number
Suspect_first_name	varchar	First name of the suspect
Suspect_last_name	varchar	Second name of the suspect
Case_type_id	interger	Random Case type id
Date_of_report	datetime	Case reporting date
Crime_location	varchar	Location of crime scene
Description	text	Case description
Status	varchar	Case status
Created_at	datetime	Data insertion datetime
Updated_at	timestamp	Data updated time

The police cases table details above shows the data that is to be stored in the police cases table. It shows the field name, the data type and the description of the data.

Police category table details

Table 5.5: police category table details.

FIELD NAME	DATA TYPE	DESCRIPTION
id	integer	Police category id
name	varchar	Name of the police category
description	text	Police category description
created_at	datetime	Data insertion datetime
Updated_at	timestamp	Data updated time

The police category table details above shows the data that is to be stored in the police category table. It shows the field name, the data type and the description of the data.

Criminals table details

Table 5.6: criminal table details

FIELD NAME	DATA TYPE	DESCRIPTION
id	integer	Criminal id
first_name	varchar	Criminal first name
last_name	varchar	Criminal last name
sex	Enum('male', 'female')	Criminal gender
image	string	Image of the criminal
status	Enum('wanted', 'most wanted', 'regular')	Status of the criminal
created_at	datetime	Data insertion datetime
Updated_at	timestamp	Data updated time

The criminal table details above shows the data that is to be stored in the criminal table. It shows the field name, the data type and the description of the data.

County table details

Table 5.7: county table details

Field Name	Data Type	Description
id	integer	County id
name	varchar	County name
created_at	datetime	Data insertion datetime
Updated_at	timestamp	Data updated time

The county table details above shows the data that is to be stored in the county table. It shows the field name, the data type and the description of the data.

Case types table details

Table 5.8: county table details

Field Name	Data Type	Description
id	integer	Case type id
case_type	varchar	Type of case reported
created_at	datetime	Data insertion datetime
updated_at	timestamp	Data updated time

The case types table details above shows the data that is to be stored in the case types table. It shows the field name, the data type and the description of the data.

These therefore formed the requirements of the Police record management system and were broadly classified into the functional requirements and the Non-functional requirements.

Requirements Analysis and Test Cases

5.2 Functional Requirements

5.2.1 User Login

This feature is used by the User (Administrator/Police officer/ other clients) to login in to the system. They are required to key in the email address and the password before they are granted permission to enter the system. It is verified and the invalid email address and password are not allowed to enter the system. If the username and the password do not match, some message will pop up to notify the user that the information supplied was invalid. If valid, user successfully login to the system.

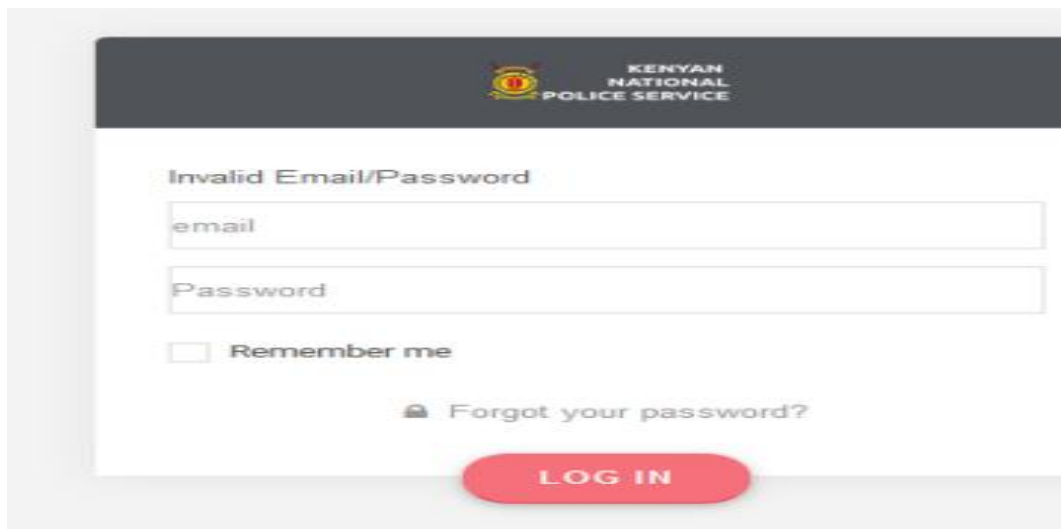


Figure 5.1: Wrong credentials window

Functional Requirements

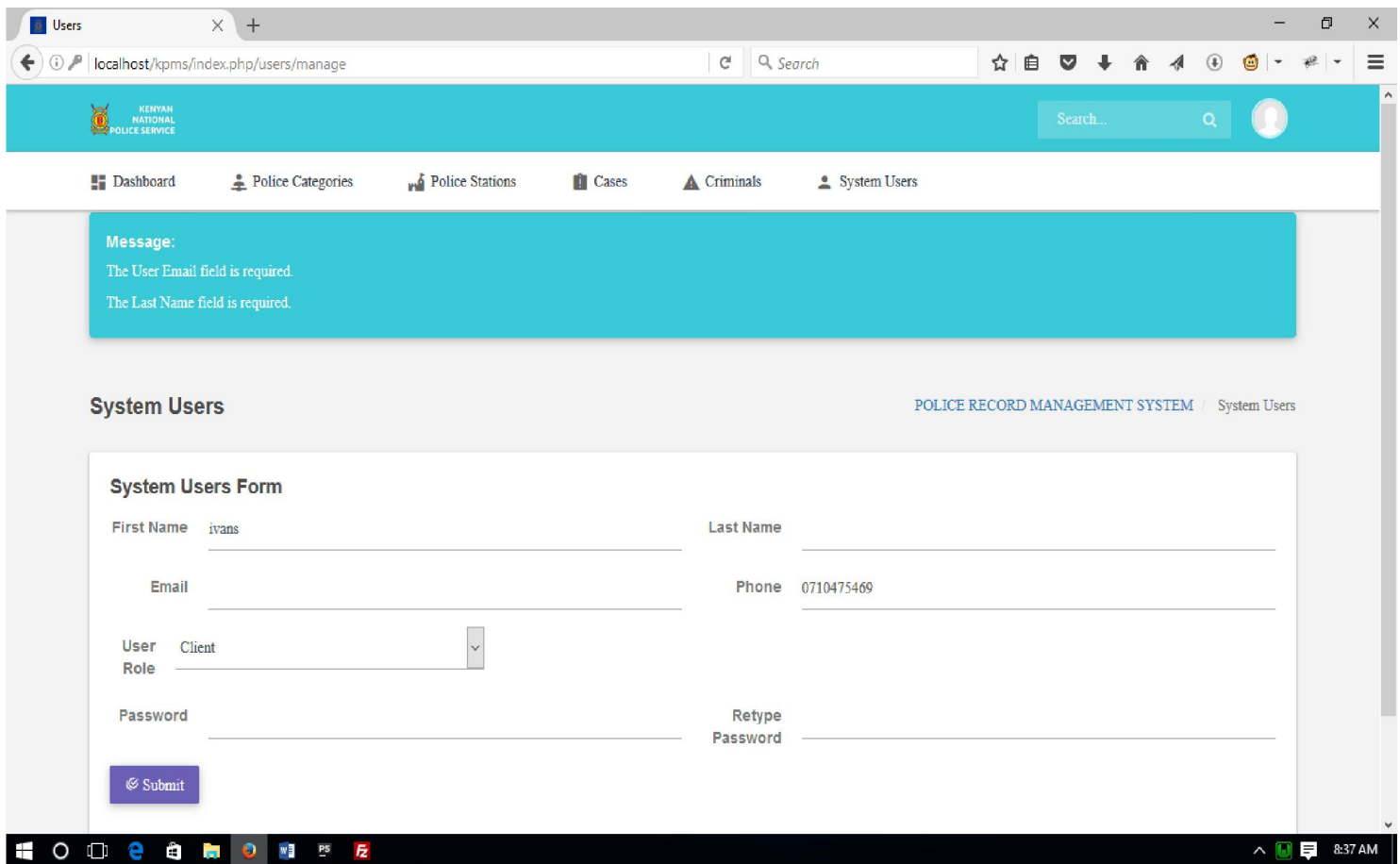
Username is provided when they register with the system. The system must only allow the user with valid email address and password to enter the system. The system must be able to perform authorization process which decides what the user's level can access to. The user must be able to logout after they are finished using the system.

5.2.2 Register a new police/Agent

Description of Feature

This feature will be used by the administrator to register a new users of the system.

The admin tabs Add New User, System prompts add police or agent. Admin enters the details in the form provided. System ensures that the information supplied by the user (Admin) is of the correct form through the input type of the form. After the admin tabs the submit button a message will be provided informing to the user action successful meaning data enter into the database successful.



The screenshot displays a web browser window at the URL `localhost/kpms/index.php/users/manage`. The page header includes the Kenyan National Police Service logo and a search bar. The navigation menu contains: Dashboard, Police Categories, Police Stations, Cases, Criminals, and System Users. A prominent teal message box states: "Message: The User Email field is required. The Last Name field is required." Below this is the "System Users" section, titled "POLICE RECORD MANAGEMENT SYSTEM / System Users". The "System Users Form" contains the following fields: First Name (ivans), Last Name, Email, Phone (0710475469), User Role (Client), Password, and Retype Password. A "Submit" button is located at the bottom left of the form.

Figure 5.2: Checking Empty Fields

CHAPTER 6

6.0 IMPLEMENTATION AND DEPLOYMENT

6.1 Implementation

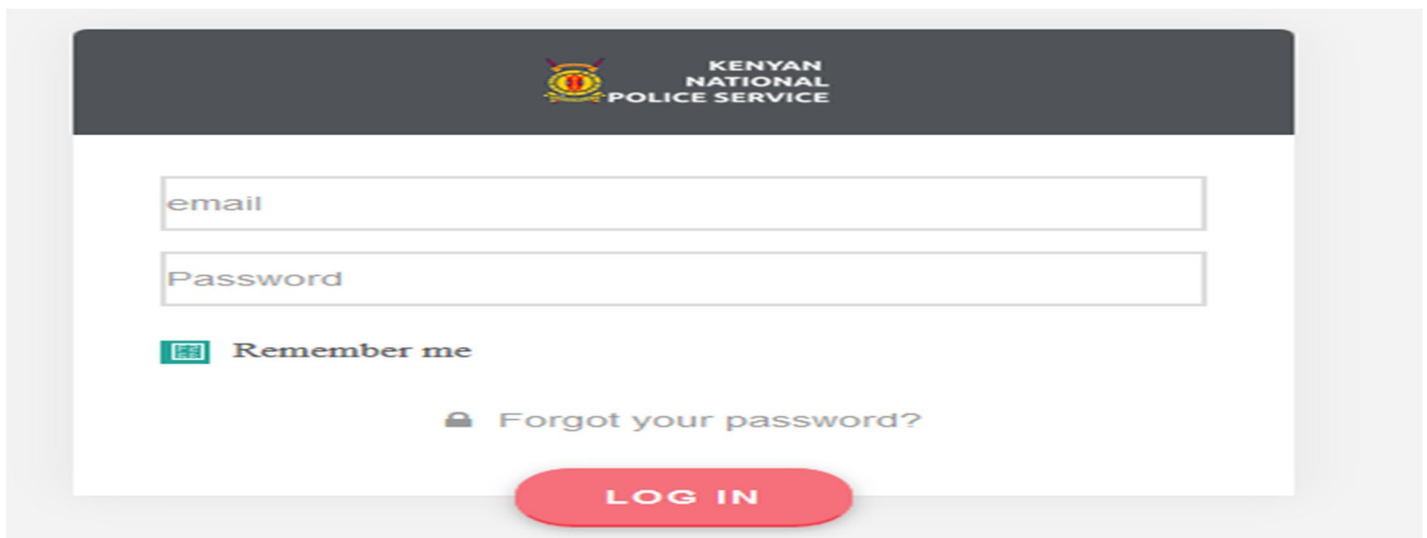
Introduction

In this section and with the requirements well defined, we embarked on the implementation of the findings so far. It is in this stage that we developed the programs that will help meet the expectations of the system. Moreover we developed the interfaces that will interact with the various users of the system and they are as shown below.

6.2 User Interfaces

This system is used by both the Administrator, police and the client so we designed it simply without so much use of colors. The interface is user friendly and easy to use. This can be proving when the user knows what the button's function is when he or she looks at the button. This is due to the fact that the button's text is clear and easy to understand. For example when the user wants to log into the system, he or she uses "Log In" as button's text and not anything else.

6.2.1 Users login window

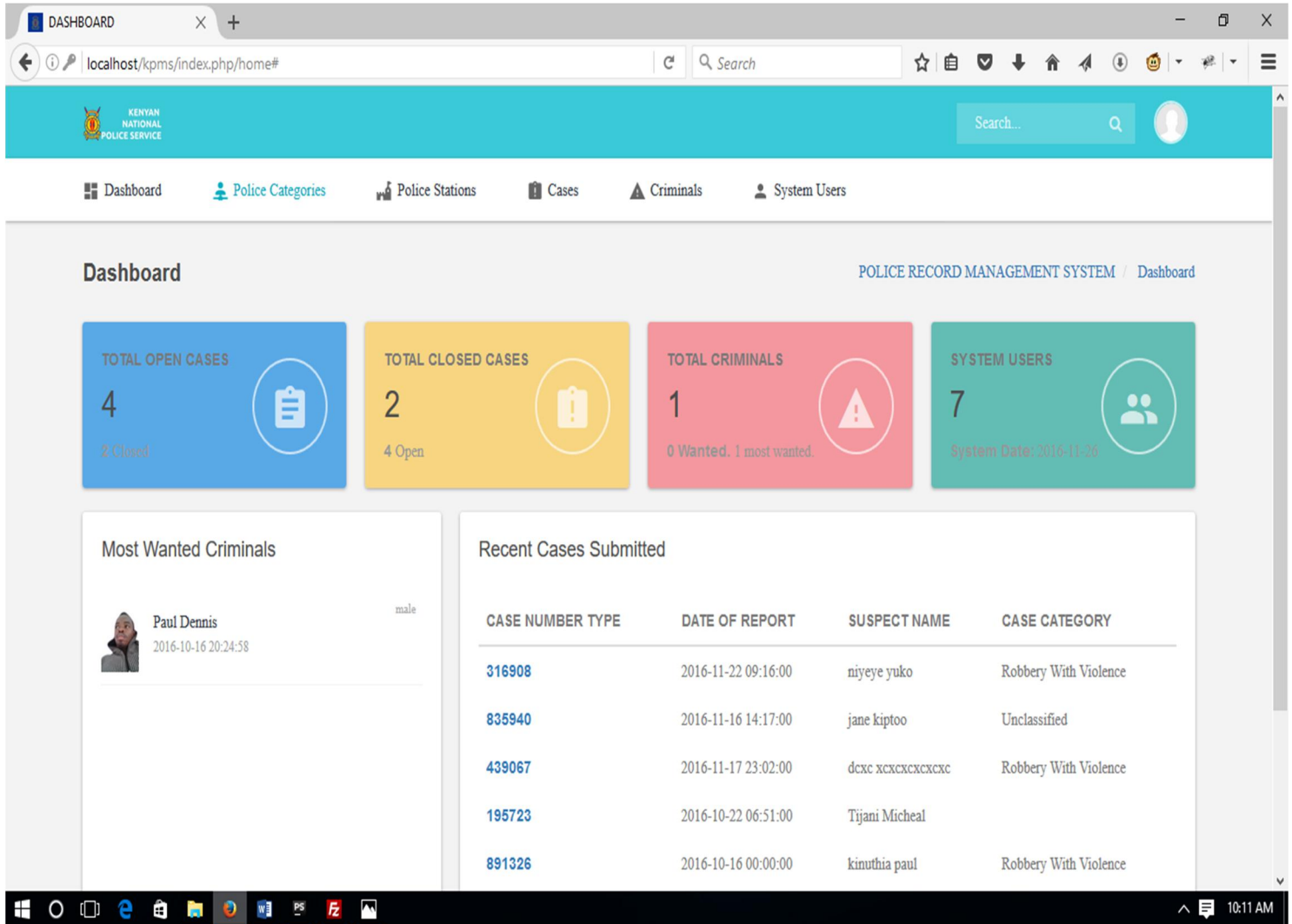


The image shows a login window for the Kenyan National Police Service. At the top, there is a dark grey header containing the service's logo and the text "KENYAN NATIONAL POLICE SERVICE". Below the header, there are two white input fields: the first is labeled "email" and the second is labeled "Password". Underneath the password field is a checkbox labeled "Remember me". Below the checkbox is a link that says "Forgot your password?". At the bottom of the form is a large, rounded red button with the text "LOG IN" in white capital letters.

Figure 6.1: Login window

6.2.2 Admin Dashboard

This is the page for the administrator side of the System and it is in this page that the administrator can choose to view existing and add users, add police category and formulation, report cases, add criminal details and police stations.



Dashboard POLICE RECORD MANAGEMENT SYSTEM / Dashboard


TOTAL OPEN CASES
4
2 Closed

TOTAL CLOSED CASES
2
4 Open

TOTAL CRIMINALS
1
0 Wanted, 1 most wanted.

SYSTEM USERS
7
System Date: 2016-11-26

Most Wanted Criminals

 **Paul Dennis** male
2016-10-16 20:24:58

Recent Cases Submitted

CASE NUMBER	TYPE	DATE OF REPORT	SUSPECT NAME	CASE CATEGORY
316908		2016-11-22 09:16:00	niyeye yuko	Robbery With Violence
835940		2016-11-16 14:17:00	jane kiptoo	Unclassified
439067		2016-11-17 23:02:00	dcxc xcxcxcxcxcxc	Robbery With Violence
195723		2016-10-22 06:51:00	Tijani Micheal	
891326		2016-10-16 00:00:00	kinuthia paul	Robbery With Violence

Figure 6.2: Administrator Operation Window Appearance.

6.2.3 Police Dashboard

This is the page the police officer with login details provided by the administrator. Once logged in, the police office is redirected to this page. It is in this window that the police can access features that enable him to view police category and formulation, view police stations, report cases and add criminals information to the system.

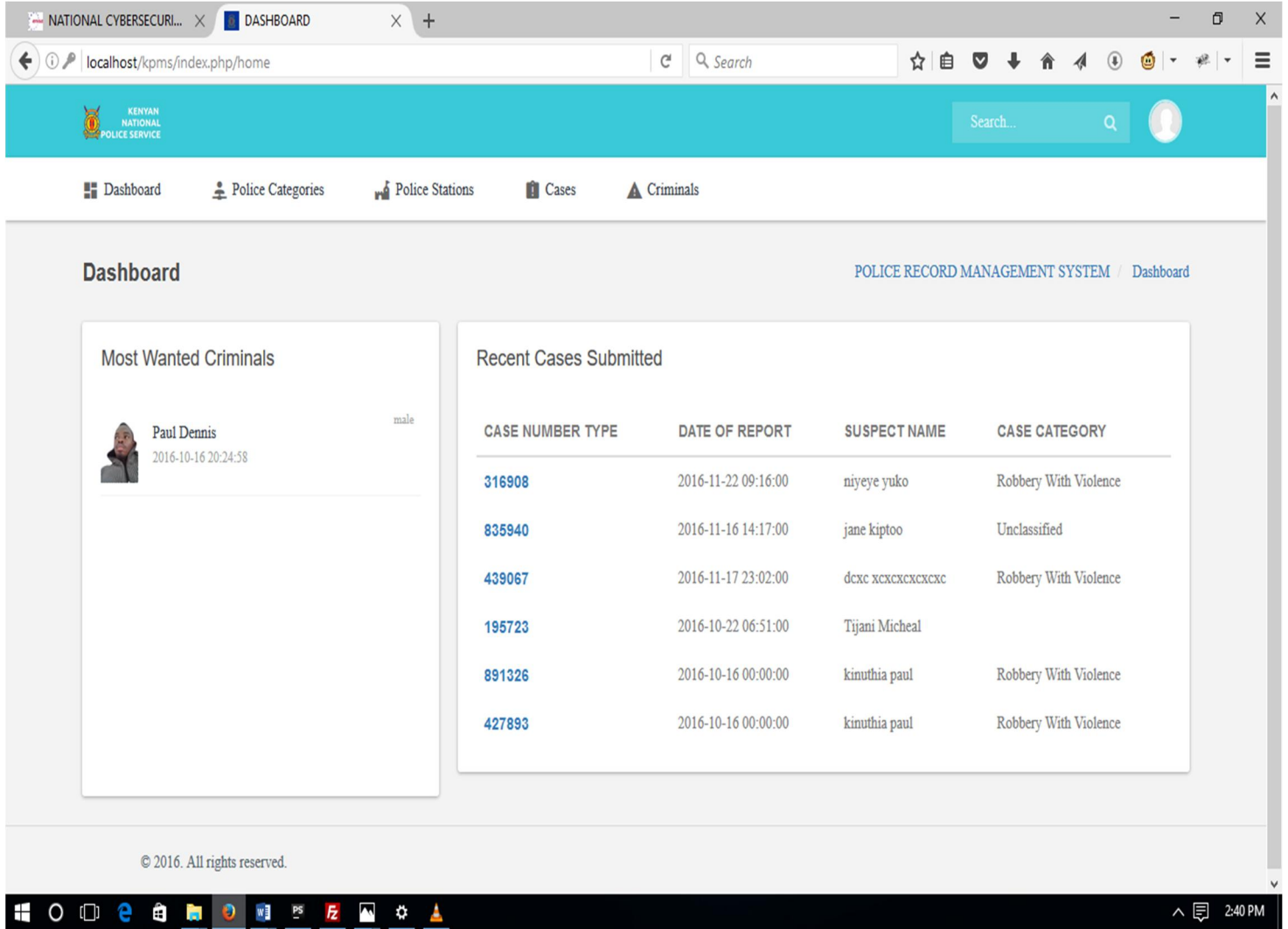


Figure 3.3: Police Operation Window Appearance.

6.2.4 Client Dashboard

The window below is for the 1st client, a Police Officer. Where they will be interacting

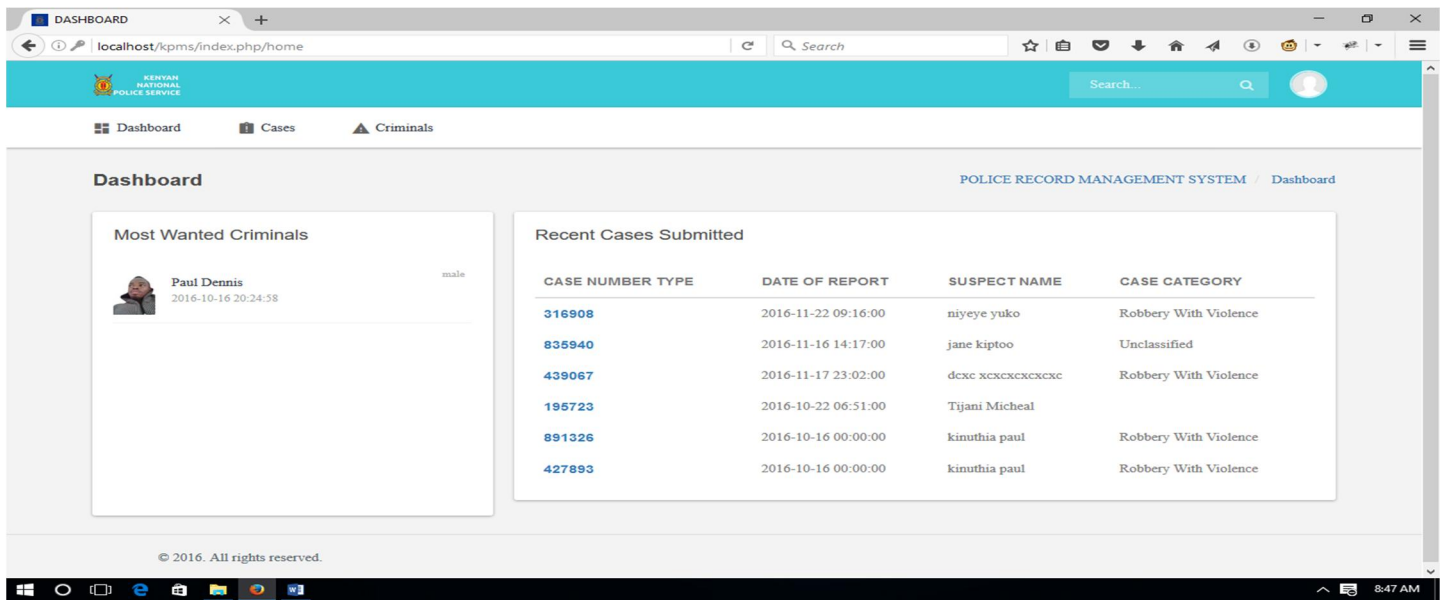


Figure 6.4: police officer Operation Window Appearance

6.2.5 Register New Admin, Police and Client Window

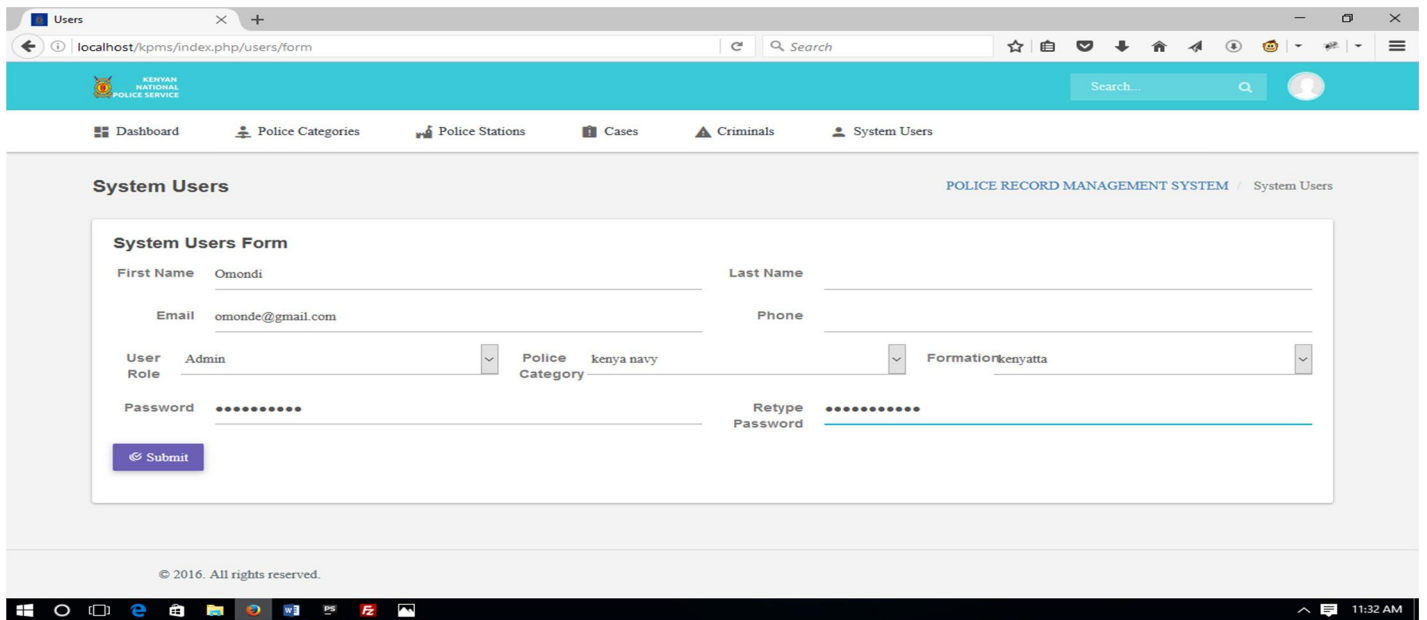


Figure 6.5: Registering new system user

6.3 Deployment

Generally there are four approaches for the implementation of the system in an organization. These are: Direct cutover, phased approach, pilot approach and the parallel approach. We greatly analyzed the four approaches to the system implementation and chose the phased operation:

6.3.1 Phased Approach

Phased operation works in different stages. It normally entails the implementation of the new system in modules. It is also a combination of the cutover and the parallel approach. We intend to implement it this way because, due to the fact that the system is new and stress as to the number of the users is not clear at this moment, it would be therefore essential to take it and implement it module by module till the last module of the system proves to be effective and well operational as required. Risk of errors or failures in this system may also have prompted us to use this method. Risks will not affect the entire system but to the single module or the several modules implemented so far. The third reason for its use is the cost involved in its implementation may be relatively lower compared to other approaches such as the direct approach which entails the overall implementation of the system at once.

Phased operation works in different phases or stages and it's also a combination of direct cutover and parallel similar to pilot operation. But in this approach the entire system is provided to some users instead of all system users. (Morgan S, 2009)

In phase operation the risk of errors or failures is limited to the implemented module only and also phased operation is less expensive than the full parallel operation. But in some cases, phased operation can cost more than a pilot approach where the system involves a large number of separate phases.

With the time that has elapsed so far, we have been able to make a great progress on the project schedule and have managed to achieve the following:

We were able to identify the project that we are currently undertaking in the course of the first week as shown below in the schedule and this has formed the basis for our entire work. This was followed by the proposal writing as well as the editing and rectification of the areas that required correction as was directed by the supervisor. The completion of the proposal writing was then followed by the presentation and defending of the idea behind my written proposal.

Then came the project design of our system and here we tried to design a clear flow of information in the system and this further helped us gain a clear picture of the various entities and the relationship between them as well as the exchange of the information amongst them. This formed the last task before the elapse of time.

CHAPTER 7

7.0 DISCUSSION

7.1 SECURITY ASPECTS IN RELEVANCE TO PROJECT

For a user to be issued with an account he has to be properly authenticated by the system administrator to make sure that he is who he is saying by having proper documentation.

To enhance proper access control, the user is supposed to use a username and password where username could be the users police identification number.

Both the user_Id and the password have been encrypted using SHA-1 and MD5 functions to maintain database integrity of the data..

In the database the passwords are further hashed using SHA-64 algorithm to make sure its as confidential as possible

Event logs of all clients are easily retrievable from the system for accountability of users

7.2 Future Improvements

- Fingerprint (Biometrics) integration module: for live scanning of suspects fingerprints
- The system to provide crime statistics for decision making since the law enforcement community has an ever-increasing need for timely and accurate data for a variety of purposes such as planning, budget formulation, resource allocation, assessment of police performance and the evaluation.
- GPRS tracker of the IP address used to report crimes for non- repudiation purposes.
- Progress report on the cases and information records reported or fed into the system.

CHAPTER 8

8.0 CONCLUSIONS AND RECOMMENDATIONS

8.1 Discussion

Police record management system a software that has been developed to enable police officers and clients report and keep criminal records effectively and efficiently in a centralized database. Research on the current system of handling criminal records was carried out and proven to be outdated at its worst. Also in order for police industry to be strategically aligned with vision 2030 they need automation fast! (PRMS) project was developed in such a way that it will minimize most of the limitations of the current system if not doing away with them. There was also thorough testing to ensure no problem will arise during operations.

8.2 Limitations

The development of the police record management system has not been as easy as it may appear as severally we've had to face a number of challenges and these are listed below:

8.2.1 Information Source

Information required to develop the application was not readily available on the Internet and if it did, this was not direct despite the high internet connection. This did not only apply for the source codes only but also did apply for the Literature review still. The codes available on the internet required a bit of customization and thorough editing in order to realize the required functionality. This was not easy as for the information such as the codes to be edited they required to first be understood before procession to edit the contents from their current form. We also sought help from 2 police officers, Inspector Boniface Gacheru (Administration Police Officer) and Constable Judy Cherop (Kenya police officer) for their help in understanding various police formulations and categories

8.2.2 Financial Incapacitation

It was also challenging for us to buy devices such as the fingerprint scanner which we really wanted to incorporate with the system, hence we had to work with what we had. This was the most challenging part.

8.2.3 Learning and implementing a new language

Considering that php holds a large number of similarities to those of Java, implementing and learning it has not been easy though after regular interaction with the language it has enabled us gain a greater insight on what it entails. Moreover, the implementation of the unique part of our system has not been easy as it has had to involve proper understanding of the language as well as the twist of the existing php source codes.

8.3 Recommendations

To reverse the above problems and realize improvements in succeeding developments, I would like to make the following recommendations.

8.3.1 Provision of project finances to the students

I would recommend that some special finances be provided to act as the support for the students who face difficulties in the development and research process.

8.3.2 Compelling some institutions to pave way for the students to develop

Some institutions have been a major obstacle in the development of the projects and the higher level institutions should compel them to release as well as loosen the restrictions they have over their intellectual property such API (Application Programming Interface).

8.4 Conclusion

In conclusion we would like to say that taking this project all through has been a wonderful experience for us. Without the practical knowledge that we acquired throughout the course, this project would not have materialized. This is a very important part of our course and has helped us familiarize with the market expectations of the course at large. The project also helped us realize that having a framework from which we were working from, simplified most details to make the project a success.

REFERENCES

- Morgan, S, 2009. Analyzing Information Systems Research,
- Griffiths, A. (2010). CodeIgniter 1.8 Professional Development. Packt Publishing Ltd.
- Groenewald, H., & Peake, G. (2004). Police Reform through Community-Based Policing. International Peace Academy and Safe World.
- (Anonymous, 2016) https://en.wikipedia.org/wiki/Administration_Police (accessed on Nov 2016)
- (Anonymous, 2016) https://en.wikipedia.org/wiki/Kenya_Police (accessed on Nov 2016)
- Lerdorf R, Tatroe, K., & MacIntyre, P. (2006). Programming Php. "O'Reilly Media, Inc."
- Nijssen, G. M., & Halpin, T. A. (1989). Conceptual Schema and Relational Database Design: a fact oriented approach. Prentice-Hall, Inc.
- Raptor Technology LTD (2016) Police Record Management system draft proposal (issue 1)
- Serianu (2014) Kenya cyber security report 2014 issue 1
- Tapani, Z. (2010). Study and Integrate Bootstrap 3 for OpixManager.

APPENDICES

APPENDIX A

Work Plan

Start Date & Finish Date	Mon 10/18/16	Wed 10/19/16	Mon 10/24/16	Fri 10/28/16	Sat 11/19/16	Thu 11/24/16	Mon 11/28/16	Fri 12/2/16	Mon 12/5/16
	Task Name Tue 10/18/16	Sun 10/23/16	Thu 10/28/16	Fri 11/18/16	Wed 11/23/16	Sun 11/28/16	Thu 12/1/16	Sat 12/3/16	Tue 12/6/16
proposal									
planning									
design									
coding									
implementation									
integration									
deployment									
maintenance									
closing									

APPENDIX B

Police Record Management System Budget.

ITEMS	AMOUNT IN KSH.
Wamp Server v3.0.6	Open source
MS office 2016	18,000
PHP Storm v.6.0.2	5,000
SnagIt Editor v12	2,000
Monitor	5,000
Dual Core C.P.U	15,000
Coding Net Labor	100,000
Stationary	1,000
TOTAL	145,000

APPENDIX C

Work Breakdown Structure (As A Project)

Level 1	Level 2	Level 3
1. Automated Police Record Management System	1.1 Initiation	1.1.1 Evaluation & Recommendations 1.1.2 Develop Project Charter 1.1.3 Deliverable: Submit Project Charter 1.1.4 Project Supervisor Reviews Project Charter 1.1.5 Project Charter Signed/Approved
	1.2 Planning	1.2.1 Create Preliminary Scope Statement 1.2.2 Determine Project Team 1.2.3 Project Team Kickoff Meeting 1.2.4 Develop Project Plan 1.2.5 Submit Project Plan 1.2.6 Milestone: Project Plan Approval
	1.3 Execution	1.3.1 Project Kickoff Meeting 1.3.2 Verify & Validate User Requirements 1.3.3 Design System 1.3.4 Procure Hardware/Software 1.3.5 Install Development System 1.3.6 Testing Phase 1.3.8 Install Live System 1.3.8 User Training 1.3.9 Go Live
	1.4 Control	1.4.1 Project Management 1.4.2 Project Status Meetings 1.4.3 Risk Management 1.4.4 Update Project Management Plan
	1.5 Closeout	1.5.1 Document Lessons Learned 1.5.3 Update Files/Records 1.5.4 Gain Formal Acceptance 1.5.5 Archive Files/Documents

APPENDIX D

System Specification Requirement

Operating systems

Windows 8 and higher versions

Software specifications

PHP v5.6

MySQL Server v5.8

Sublime text v6.2

Mozilla Firefox v361.0

Hardware specifications

Intel 3.0 GHz or higher version

2GB RAM

20GB HDD Space

APPENDIX E

Constant helpers codes

```
<?php
// Used all over the application
define('APP_VERSION','1.0.0');
define('APP_NAME','POLICE RECORD MANAGEMENT SYSTEM');

//Define Database Tables
define('TABLE POLICE CATEGORIES','tblpolicecategories');
define('TABLE USERS','tblusers');
define('TABLE POLICE STATIONS','tblpolicestations');
define('TABLE COUNTIES','tblcounties');
define('TABLE POLICE FORMULATIONS','tblpoliceformulation');
define('TABLE POLICE CASES','tblpolicecases');
define('TABLE CASE TYPES','tblcasetypes');
define('TABLE CRIMINALS','tblcriminals');

// Other constants used in the application.
define('ERROR', 'error');
define('SUCCESS', 'success');
define('ACTION SUCCESFUL', 'Action Succesful. ');
define('ACTION UNSUCCESSFUL', 'Action Unsuccessful. ');
define('ADMIN', 'admin');
define('POLICE', 'police');
define('CLIENT', 'client');
```

Authentication Source Codes

```
<?php

Class Auth
{
    static function is_loggedin()
    {
        $CI = & get_instance();
        if ($CI->session->has_userdata('user_id') != '')
            return TRUE;
        else
            return FALSE;
    }

    static function Role($role_id = 1*1)
    {
        $roles = [
            0 => 'Client1',
            1 => 'Police',
            2 => 'Client2',
        ];
        return $roles[$role_id];
    }

    static function Logout($type = 'user')
    {
        $CI = & get_instance();
```

```

unset($_SESSION['user_id']);
unset($_SESSION['user_role']);
unset($_SESSION['previous_url']);
$CI->session->sess_destroy();
redirect(site_url('authentication/landing'));
}

static function Login($data = array()){
    $CI = & get_instance();
    $query = $CI->db->get_where(TABLE_USERS, $data);
    $row = $query->row();
    if($query->num_rows() > 0){
        $data = ['last_login'=>date('Y-m-d H:i:s')];
        DB::update(TABLE_USERS, ['id'=>$row->id],$data);
        if($row->banned == 1) {
            $CI->session->set_flashdata(ERROR, 'You have been banned. Please contact the admin.');
```

```

            return FALSE;
        }
    }

    $userdata = [
        'user_id'=>$row->id,
        'user_role'=>$row->role,
        'name'=>$row->first_name.' '.$row->last_name,
        'user_role'=>$row->role,
    ];
    $CI->session->set_userdata($userdata);
    return TRUE;
} else{
    $CI->session->set_flashdata(ERROR, 'Invalid Username or Password');
```

```

    return FALSE;
}
}

static function Bounce($uri_string = '')
{
    $CI = & get_instance();
    $CI->session->set_userdata('previous_url', $uri_string);
    redirect(site_url('authentication/login'));
}
}
}

```

Dashboard source codes

```

<link rel="stylesheet" type="text/css"
href="<?=base_url('assets/plugins/fancybox/source/jquery.fancybox.css') ?>">

<div class="container">

    <!-- Page-Title -->
    <div class="row">
        <div class="col-sm-12">
            <div class="page-title-box">
                <div class="btn-group pull-right">
                    <ol class="breadcrumb hide-phone p-0 m-0">
                        <li>
                            <a href="#"><?=APP_NAME ?></a>
                        </li>
                        <li class="active">

```

```

        <a href="#">Dashboard</a>
    </li>
</ol>
</div>
<h4 class="page-title">Dashboard</h4>
</div>
</div>
<!-- end page title end breadcrumb -->

<?php if(user_role() == ADMIN): ?>
<div class="row">

    <div class="col-lg-3 col-md-6">
        <div class="card-box widget-box-two widget-two-primary">
            <i class="mdi mdi mdi-clipboard-text widget-two-icon"></i>
            <div class="widget-two-content">
                <p class="m-0 text-uppercase font-600 font-secondary text-overflow"
title="Statistics">
                    Total Open Cases
                </p>
                <h2><span data-
plugin="counterup"><?=number_format(DB::num_rows(TABLE_POLICE_CASES, ['status'=>'open']))
?></span></h2>
                <p class="text-muted m-0"><b><?=number_format(DB::num_rows(TABLE_POLICE_CASES,
['status'=>'closed'])) ?></b> Closed</p>
            </div>
        </div>
    </div><!-- end col -->

    <div class="col-lg-3 col-md-6">
        <div class="card-box widget-box-two widget-two-warning">
            <i class="mdi mdi mdi-clipboard-alert widget-two-icon"></i>
            <div class="widget-two-content">
                <p class="m-0 text-uppercase font-600 font-secondary text-overflow" title="User
This Month">
                    Total Closed Cases
                </p>
                <h2><span data-
plugin="counterup"><?=number_format(DB::num_rows(TABLE_POLICE_CASES, ['status'=>'closed'])) ?>
</span> </h2>
                <p class="text-muted m-0"><b><?=number_format(DB::num_rows(TABLE_POLICE_CASES,
['status'=>'open'])) ?></b> Open</p>
            </div>
        </div>
    </div><!-- end col -->

    <div class="col-lg-3 col-md-6">
        <div class="card-box widget-box-two widget-two-danger">
            <i class="mdi mdi mdi-alert widget-two-icon"></i>
            <div class="widget-two-content">
                <p class="m-0 text-uppercase font-600 font-secondary text-overflow"
title="Statistics">
                    Total Criminals
                </p>
                <h2><span data-
plugin="counterup"><?=number_format(DB::num_rows(TABLE_CRIMINALS)) ?></span> </h2>
                <p class="text-muted m-0"><b><?=number_format(DB::num_rows(TABLE_CRIMINALS,
['status'=>'wanted'])) ?> Wanted.</b> <?=number_format(DB::num_rows(TABLE_CRIMINALS,
['status'=>'most_wanted'])) ?> most wanted.</p>
            </div>
        </div>
    </div><!-- end col -->

    <div class="col-lg-3 col-md-6">

```

```

        <div class="card-box widget-box-two widget-two-success">
            <i class="mdi mdi-account-multiple widget-two-icon"></i>
            <div class="widget-two-content">
                <p class="m-0 text-uppercase font-600 font-secondary text-overflow" title="User
Today">
                    System Users
                </p>
                <h2><span data-plugin="counterup"><?=number_format(DB::num_rows(TABLE_USERS))
?></span> </h2>
                <p class="text-muted m-0"><b>System Date:</b> <?=date('Y-m-d') ?></p>
            </div>
        </div>
    </div><!-- end col -->

</div>
<!-- end row -->
<?php endif; ?>

<div class="row">
    <div class="col-md-4">
        <div class="card-box">
            <h4 class="header-title m-t-0 m-b-30">Most Wanted Criminals</h4>

            <div class="inbox-widget slimscroll-alt" style="min-height: 302px;">
                <?php $most_wanted = DB::get(TABLE_CRIMINALS, 'id', 'DESC', 10, 0,
['status'=>'most_wanted']) ?>
                <?php foreach($most_wanted as $row): ?>
                    <a href="#">
                        <div class="inbox-item">
                            <?php
                                if(empty($row->image)){
                                    if($row->sex == 'male'){
                                        $image = 'male.png';
                                    }else{
                                        $image = 'female.png';
                                    }
                                }else{
                                    $image = $row->image;
                                }
                            ?>
                            <div class="inbox-item-img">
                                <a class="fancybox" rel="group"
href="<?=base_url('assets/criminals/'.$image) ?>">
                                    first_name ?>" class="img img-responsive">
                                </a>
                            </div>
                            <p class="inbox-item-author"><?=$row->first_name. ' '. $row->last_name
?></p>

                            <p class="inbox-item-text"><?=$row->created_at ?></p>
                            <p class="inbox-item-date"><?=$row->sex ?></p>
                        </div>
                    </a>
                <?php endforeach; ?>
            </div>
        </div> <!-- end card -->
    </div>
    <!-- end col -->

    <div class="col-md-8">
        <div class="card-box">
            <h4 class="header-title m-t-0 m-b-30">Recent Cases Submitted</h4>

```

```

        <div class="table-responsive">
            <table class="table table table-hover m-0">
                <thead>
                    <tr>
                        <th>Case Number Type</th>
                        <th>Date of Report</th>
                        <th>Suspect Name</th>
                        <th>Case Category</th>
                    </tr>
                </thead>
                <tbody>
                    <?php $recent_cases = DB::get(TABLE_POLICE_CASES, 'id', 'DESC', 10, 0); ?>
                    <?php foreach($recent_cases as $row): ?>
                        <tr>
                            <th><strong><a href="#" title="" ><?=$row->case_number
?></a></strong></th>
                            <td><?=$row->date_of_report ?></td>
                            <td><?=$row->suspect_first_name.' '.$row->suspect_last_name ?></td>
                            <td><?=DB::get_cell(TABLE_CASE_TYPES, ['id'=>$row->case_type_id,
'case_type') ?></td>
                        </tr>
                    <?php endforeach; ?>
                </tbody>
            </table>

        </div> <!-- table-responsive -->
    </div> <!-- end card -->
</div>
<!-- end col -->

</div>
<!-- end row -->

<?php $this->load->view('includes/footer') ?>

</div>

```

Database_handlers source codes

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class General_model extends CI_Model
{
    function __construct()
    {
        parent::__construct();
    }

    public function get_submissions($where = [], $limit = 0, $offset = 0)
    {
        $query = $this->db->join(TABLE_USERS, TABLE_USERS.'.id = '.TABLE_USER_SUBMISSIONS.'.user_id',
'both');

        if(count($where) < 1)

```

```

        $query = $this->db->where($where);

        if($limit != 0)
            $query = $this->db->limit($limit, $offset);

        $query = $this->db->get(TABLE_USER_SUBMISSIONS);
        return $query->result();
    }

    public function api_register($data = [])
    {
        $message = 'Nothing took place';

        $user['username'] = (array_key_exists('username', $data) ? $data['username'] :
'');
        $user['email'] = (array_key_exists('email', $data) ? $data['email'] :
'');
        $user['password'] = (array_key_exists('password', $data) ? $data['password'] :
'');

        if(empty($user['username']) || empty($user['email']) || empty($user['password'])) {
            $message = ['status'=>2, 'message'=>'Some fields are empty'];
        } else if(valid_email($user['email']) == FALSE) {
            $message = ['status'=>3, 'message'=>'Email is invalid.'];
        } else if(DB::it_exists(TABLE_USERS, 'email', $user['email'])) {
            $message = ['status'=>4, 'message'=>'Email already exists.'];
        } else if(DB::it_exists(TABLE_USERS, 'username', $user['username'])) {
            $message = ['status'=>5, 'message'=>'Username already exists.'];
        } else {
            $data = [
                'username'=>$user['username'],
                'email'=>$user['email'],
                'password'=>hash('sha256', $user['password']),
                'created_at'=>date('Y-m-d H:i:s')
            ];

            if(DB::save(TABLE_USERS, $data)) {
                $message = [
                    'status'=>1,
                    'message'=>'Data Saved succefully.',
                    'username'=>$user['username'],
                    'email'=>$user['email'],
                    'password'=>'*****',
                ];
            } else {
                $message = [
                    'status'=>0,
                    'message'=>'Data passed validation. but an unexpected error ocured. data not
saved.',
                    'username'=>$user['username'],
                    'email'=>$user['email'],
                    'password'=>$user['password'],
                ];
            }
        }

        return $message;
    }

    public function api_login($email = '', $password = '')
    {
        $message = 0;
        if(Auth::IsLoggedIn()){
            $message = ['status'=>2, 'message'=>'User already logged in.'];
        }
    }

```



```

}else{
    if(empty($email) || empty($password)){
        $message = ['status'=>3, 'message'=>'Email or Password is empty'];
    }else if(valid_email($email) == FALSE){
        $message = ['status'=>4, 'message'=>'Email is invalid'];
    }else{
        $login = [
            'email'=>$email,
            'password'=> hash('sha256', $password)
        ];

        if(Auth::login($login)) {
            $message = [
                'status'=>1,
                'message'=>'Login Successful.',
                'username'=>$this->session->userdata('user_username'),
                'email'=>$email,
                'password'=>$login['password']
            ];
        } else {
            $message = [
                'status'=>0,
                'message'=>'Invalid Username or password',
                'email'=>$email,
                'password'=>$password
            ];
        }
    }
}
return $message;
}

function get_api_ads()
{
    $today = date('Y-m-d');
    $query = $this->db->where('status', 'active');
    $query = $this->db->where('startdate <=', $today);
    $query = $this->db->where('enddate >=', $today);
    $query = $this->db->get_where(TABLE_ADS);
    return $query->result();
}
}

```