

ABSTRACT

The purpose of this research is to come up with a system that will assist the Ministry of Public Service, Labour and Social Welfare (MPSLSW) to develop efficiencies in delivering child welfare, preventive protection, and response to violence, abuse and exploitation against children. It provides a platform for ensuring systematic handling of child protection cases amongst multiple stakeholders in Zimbabwe. The research was carried out so as to understand the problems that are being encountered when using the current system. The managers were failing to effectively supervise district level staffs who are supposed to follow through all cases, resulting in some cases not reaching their conclusion. Such a scenario was curtailing a well-intended purpose of protecting children and responding to protection violations.

In order to address the challenges in the current system, the researcher sought to come up with a system that digitize case management data, captures referral information, produces alerts and reports and store data and information with easy retrieval of same and production of statistical reports. The researcher gathered data on how the system should like by using different research techniques which include observations, questionnaires as well as interviews. This helped the researcher come up with solutions to the problems identified. The researcher identified a gap necessitating development of a computerised system solve the problems being faced by the ministry.

The system is now able to manage all information and data requirements of Case Management with features if report production, alert messages email notifications and easy case number linked information retrieval. The system developed in response was done using Microsoft Visual Studio 2012 which is from Microsoft, for the development of both console and graphic user interface (GUI) programs with windows forms applications. It is compatible to a wide range of programming languages, which allow the code editor and debugger to support (to varying degrees) nearly any programming languages, provided a language specific service exists. The developer used C# for coding the system since ease-to-development; the rich class library makes many functions easy to implemented and also good support for distributed system. As for the databases the developers used Microsoft SQL server 2008 R2 database a relational database management system from Microsoft.

DECLARATION FORM

I, **GARNET S. CHIWESHE**, declare that I am the sole author of this thesis. All sources I have used or quoted have been indicated and acknowledged as complete references. I authorise Midlands State University to lend this dissertation to other institutions or individuals for purposes of scholarly research only.

Signature _____ Date _____

APPROVAL FORM

The dissertation/thesis entitled “**Child Protection Case Management System**“ by **Garnet S Chiweshe** meets the regulations governing the award of the degree of Bachelor of Information Systems Honours Degree of Midlands State University, and is approved for its contribution to knowledge and literal presentation.

Supervisor _____

Date _____

ACKNOWLEDGEMENTS

The researcher acknowledges the support rendered by the various people during the research period at the Ministry of Public Service, Labour and Social Welfare. Upmost sincere gratitude goes to my supervisor Mr SS Furusa for patience, guidance and unwavering support in carrying out this project, without your guidance this research would not have being successful. May the almighty bless you, not forgetting all my lecturers' guys I really want to thank you for the support and encouragement.

Moreover, I want to express my sincere appreciation to my work supervisor Mr. V Ngulube for being tolerant of my torrential visits to his office. I am infinitely grateful for his mentorship, invaluable suggestions, in calculative encouragement and inspiration. I also want to thank my work colleagues Mr. L Svosve and Mr. T Mhishi for their support and understanding despite my pestering with requests for review of the system and technical input to make the system more responsive to user needs. Thirdly, I am forever indebted to Mr. Muwoni. In spite of stressful times you believed in me and stood by me through thick and thin.

To my dearest sisters Gloria and Gay I am still overwhelmed by your utmost generosity, counsel and compassion. Without your support, help, love, dedication, generosity and inspiration I would not have reached this far. To all my friends, I want to say thank you for showing me the value in friendship, I really appreciate your consistency, encouragements and solid support.

Heartfelt thanks goes to my husband Nyasha, L.M and my lovely daughter Vinette who are always there for me even in trying times. Lastly, I would like to thank God Almighty for his profound love; for he has brought me this far.

DEDICATION

I would like to dedicate this project to God and my family. To my daughter Vinette and my loving husband Nyasha, L.M who instilled in me the work ethic, perseverance and curiosity for knowledge that has made this project possible. I also dedicate this to my caring sisters Gay and Gloria who have made this journey a sensational one for me, I love u all! Not forgetting my late parents all this is for you Daddy and Mum, may your soul rest in eternal peace.

TABLE OF CONTENTS

Abstract	i
Declaration	ii
Approval	iii
Acknowledgements	iv
Dedication	v
Contents	.vi
List of Figures	x
List of Tables	xii
List of Acronyms	xiv
List of Appendices	xvi
Contents	

CHAPTER 1: INTRODUCTION

1.1: Introduction.....	1
1.2: Background of the study	1
1.2.1: Background of the organisation	3
1.2.2: Organisational structure:	4
1.2.3: Vision:	5
1.2.4: Mission Statement:.....	5
1.3: Problem defination:.....	5
1.4 Aim	6
1.5 Objectives	7
1.6 Development Instruments	7
1.7 Justification	8
1.8 Conclusion	8

CHAPTER 2: PLANNING PHASE

2.1 Introduction.....	9
2.2 Reasons for building the system.....	9
2.3 Business Values	10
2.3.1 Customer value.....	10
2.3.2 Employee value	11
2.3.3 Managerial value	11
2.4. Technical analysis.....	11
2.4.1 Technical feasibility	12
2.4.2 Economic feasibility.....	14
2.4.2.1 Cost Benefit Analysis.....	18
2.4.3 Social Feasibility.....	21
2.4.4 Operational feasibility.....	21
2.5 Work Plan.....	22
2.5.1 Gantt chart.....	22
2.6 Technical Risks	24
2.6.1 Economic Risks.....	24
2.6.2 Personnel /Social Risks.....	24
2.6.3 Management Risks	24
2.6.4 Contractual and legal risks	25
2.6.5 Other Risks.....	25
2.7 Conclusion.....	25

CHAPTER 3: ANALYSIS PHASE

3.1 Introduction.....	26
3.2 Information Gathering Methods	26
3.2.1 Interviews.....	27
3.2.2 Focus Group Discussion	28
3.2.3 Questionnaire	29
3.2.4 Field Observation	32
3.3 Analysis of the Current System	33
3.4 Data Analysis	34
3.4.1 Context Diagram	35
3.4.2 Data flow Diagram (DFD)	35
3.5 Weaknesses of the current system	35
3.6 Evaluation of system development alternatives.....	38
3.6.1 Improvement of the current system.....	38
3.6.2 Outsource (Off-The-Shelf Package).....	39
3.6.3 In-house Development	40
3.6.4 Recommendation Approach.....	40
3.7 Requirements Analysis	43
3.7.1 Functional Requirements.....	43
3.7.2 Non-Functional Requirements	43
3.8 Conclusion.....	45

CHAPTER 4: DESIGN PHASE

4.1 Introduction.....	46
4.2 System Design	46
4.2.1 Description of Proposed System.....	47
4.2.2 Dataflow Diagram.....	49
4.3 Architectural Design	50
4.4 Physical Design.....	52
4.5 Database Design.....	54
4.6 Entity Relationship.....	62
4.7 Functional Design	64
4.8 Conclusion.....	71

CHAPTER 5: IMPLEMENTATION PHASE

5.1 Introduction.....	72
5.2 Coding.....	72
5.2.1 Pseudo Code.....	72
5.3 Testing.....	76
5.4 Installation	87
5.5 Maintenance	93
5.5.1 Corrective maintenance.....	93
5.5.2 Perfective maintenance	93
5.5.3 Adaptive Maintenance.....	94
5.6 System security	94
5.7 System backup	96
5.8 Recommendations.....	97
5.9 Conclusion	97

LIST OF FIGURES

Fig 1.1 Organisational structure	4
Fig 2.1 Risk handling techniques	23
Fig 3.1 Context diagram	35
Fig 3.2 Data flow diagram	36
Fig 4.1 Context diagram of the proposed system	48
Fig 4.2 Data flow diagram	49
Fig 4.3 Server building design	51
Fig 4.4 Proposed system physical design	53
Fig 4.5 Database structural design	54
Fig 4.6 Entity relationship diagram of the proposed system	62
Fig 4.7 System Layout	63
Fig 4.8 User group activity	64
Fig 4.9 Add user group activity	65
Fig 4.10 Login activity	65
Fig 4.11 View case activity	66
Fig 4.12 Case intake activity	66
Fig 4.13 Case Assessment	67
Fig 4.14 Care plan	67
Fig 4.15 Case review	68
Fig 4.16 Case referral	68
Fig 4.17 Case allocation	69
Fig 4.18 Case closure	69
Fig 5.0 Testing Modules	77
Fig 5.3 Direct Implementation	90
Fig 5.4 Pilot Implementation	91

Fig 5.4 Parallel Implementation	92
Fig 5.5 Maintenance Process	93

LIST OF TABLES

Table 2.1 Software and hardware	12
Table 2.2 Server Requirements	13
Table 2.3 Network Equipments	13
Table 2.4 Software Requirements	14
Table 2.5 Development Costs	15
Table 2.6 Operational Costs	16
Table 2.7 Tangible benefits	17
Table 2.8 Intangible benefits	17
Table 2.9 Cost benefit analysis	18
Table 2.10 Work plan	21
Table 2.11 Gantt chart	22
Table 3.1 Targeted population	31
Table 3.2 Cost summary of alternatives	42
Table 4.1 User field	56
Table 4.2 Case type	56
Table 4.3 Case intake	57
Table 4.4 Case assessment	58
Table 4.5 Case plan	59
Table 4.6 Case allocation	59
Table 4.7 Case review field	60

Table 4.8 Case closure field	61
Table 4.9 Case referral field	61
Table 5.1 Validation test case s	81
Table 5.2 Verification test cases	84
Table 5.3 Training schedule	88

LIST OF ACRONYMS

MPSLSW	Ministry of Public Service Labour and Social Welfare
UNICEF.....	United Nations Children’s Fund
WEI	Word Education Incorporated
DCWPS	Department of Child Welfare Probation Services
NCMS	National Case Management System
MIS.....	Management Information System
DSS	Department of Social Services
SQL	Sequential Query Language
CMS.....	Case Management System
CPC	Child Protection Committee
CCW	Community Case Workers
HSCT	Harmonized Social Cash Transfer
BEAM	Basic Education Assistant Module
CPCMS.....	Child Protection Case Management System
DFD	Data Flow Diagram
ERD.....	Entity Relationship Diagram
LAN	Local Area Network
WAN.....	Wide Area Network
ISP.....	Internet Service Provider
DBA	Database Administrator

ROIReturn on Investment
NPVNet Present Value
ORGANOGRAM.....Organizational Structure
CBACost Benefit Analysis
RAMRandom Access Memory

LIST OF APPENDICES

Appendix A: User manual.....	100
Appendix B: Interview questions.....	116
Appendix C: Questionnaires.....	119
Appendix D: Observation score sheet.....	122
Appendix E: Code snippet.....	124

CHAPTER 1: INTRODUCTION

1.1: INTRODUCTION

This chapter will introduce the study concept, giving background information to the system that the student has developed. The student is employed in the Ministry of Public Service, Labour and Social Welfare (MoPSLSW) which has the statutory mandate for child care and protection. A number of problems were identified in one of the Ministry's flagship child protection programmes, the National Case Management System. This gave rise to need to develop an Information Management System for the case management system. This chapter details the current system structure, proposed solutions and objectives. Child Protection Case Management system will help digitize and automate information related to children entering the system for assistance from case intake all the way up to case closed.

1.2: BACKGROUND OF THE STUDY

The complex social, economic and political challenges that have confronted Zimbabwe for the last decade have resulted in increased numbers of vulnerable households and families. Zimbabwe is home to some 3 million households, 1.1 million of which are categorized as not poor, 1.4 million moderately poor and 0.5 million extremely poor, Muwoni et al (2014). In statistical terms the extremely poor households live below the food poverty line.

The HIV/AIDS pandemic has interacted with and aggravated these socio economic challenges. As a result of this pandemic, an estimated 1.6 million children, including 1.0 million who have lost one or both parents, have been made vulnerable by the HIV and AIDS epidemic. These children live below the poverty datum line and one in three of these children suffer from chronic malnutrition. Additionally, many of these children are unable enroll and stay in school because of their inability to afford the tuition fees, the uniform and other associated costs. Further, violence and abuse of children is on the increase with 60% of reported rape survivors being children and the majority of them are girls. One in eight girls is reportedly being sexually harassed at school and 22% of children are reportedly being abused by care givers.

In response to the increasing child protection needs of children in Zimbabwe, the Ministry of Public Service, Labour and Social Welfare (MPSLSW) with financial and technical support from World Education Inc (WEI) and UNICEF developed a Child Protection Case Management

System to better provide wraparound protection services to children and families. The system was developed in 2014 based on existing approaches to child protection by the Ministry's Department of Child Welfare and Probation Services (DCWPS). These approaches are anchored on established paper based systems that have been in use since the early 80s when use of computer systems had not yet been fully espoused.

Case management is an approach to care that is, by definition, a collaborative process that maximizes the effective use of resources through identification, communication and coordination of care. It is a process that rests on the assessment of an individual in the context of their family and larger environment, plans care based on a comprehensive assessment and then implements the care through the coordination of services offered by a range of service providers at governmental and non-governmental levels. The NCMS leverages on the prevalent culture of care and support inherent in Zimbabwean communities to identify hard to reach orphans and vulnerable children and provide them with critical interventions at their point of need.

From a regulatory perspective, the NCMS is the Government of Zimbabwe's framework for the definition and co-ordination of all child care and protection services and service providers. It defines the core values, principles, and beliefs that underpin and guide all initiatives undertaken by stakeholders in fulfilling the country's commitment to protecting and safeguarding children; as guaranteed by Zimbabwean legislation.

While the intend on coordination and provision of comprehensive services is clearly defined and understood, the major challenge that remains is to make such coordination and service provision more efficient. In an increasingly computerized era where technology based information systems that save time, human and financial resources, it is important that such lifesaving programme set ups like the NCMS be supported by simple but effective management information systems.

The need for development of an MIS for the NCMS is identified in the official framework document. The document proposes that children should ideally have unique identifiers that make it easier to make service referrals. Clinical supervision for professional social workers and community workforce is equally identified as important and for this to happen, there is a clear need

for a computer based system that is easy to view and use, making it more feasible for managers to supervise lower level staff who have constant direct interaction with children.

It is against this background that this study has been conceptualized to help provide a solution to the challenge of complex paper systems that make it difficult to implement the NCMS effectively.

1.2.1: BACKGROUND OF ORGANISATION

The Ministry of Public Service, Labour and Social Welfare (MoPSLSW) is the arm of government with the statutory responsibility for the protection, care and support of vulnerable children and other groups in Zimbabwe. The Department of Social Service (DSS) and the Department of Child Welfare and Probation Services (DCWPS) in the Ministry employ Social Workers and other professionals to implement policies and programmes that are designed to protect and support vulnerable groups such as:

- i. Children in need of foster, and institutional care or adoption
- ii. Children in need of education support
- iii. Child victims of abuse
- iv. Children in conflict with the law
- v. People living with disabilities
- vi. Refugees
- vii. Windows and widowers of national heroes
- viii. Victims of national disasters such as drought and other persons considered to be vulnerable in terms of the social Welfare Assistance Act and other pieces of legislation.

1.2.2: ORGANISATIONAL STRUCTURE:

The Ministry is made up of many different departments and the hierarchy is as follows.

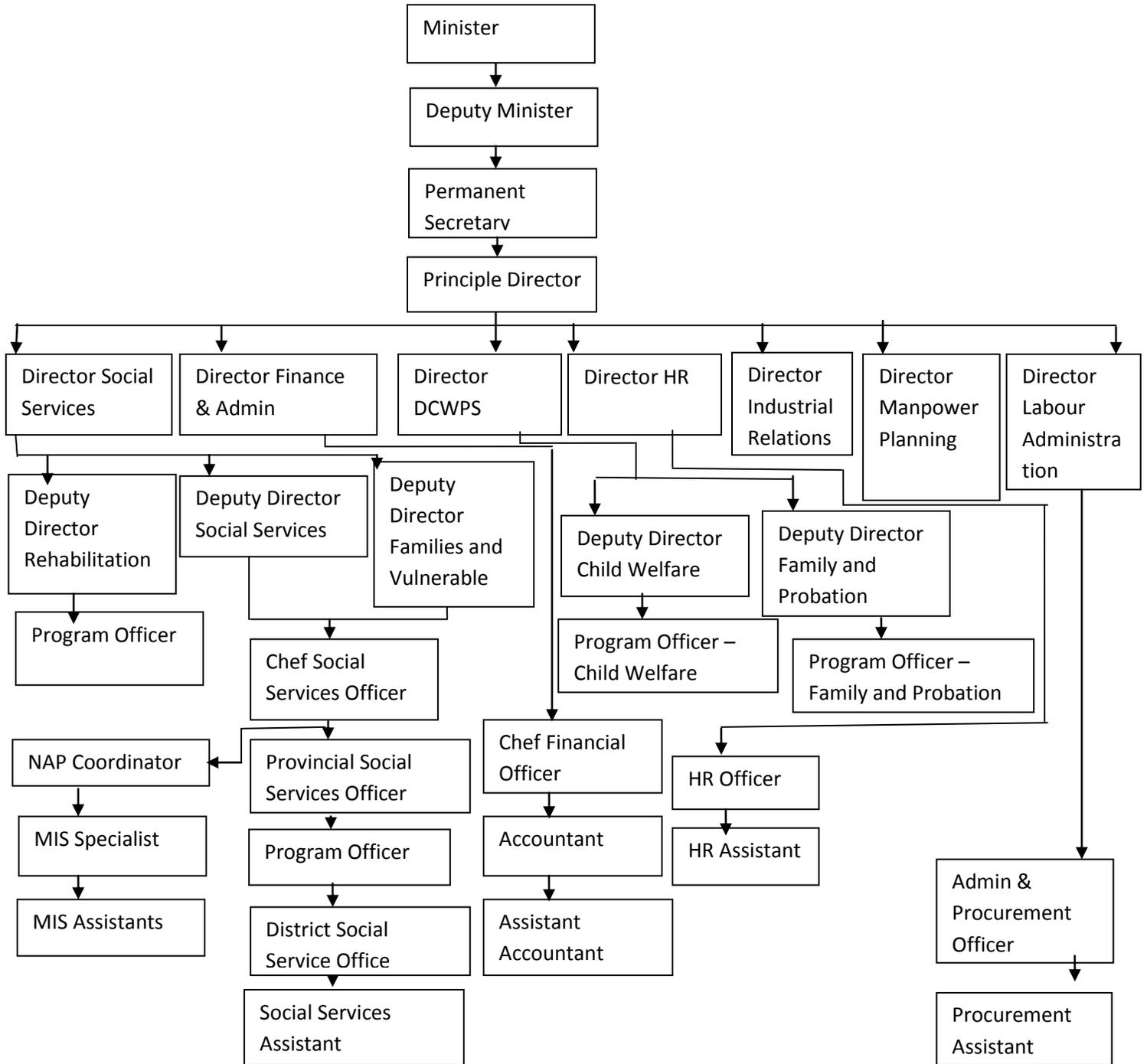


Fig 1.1 Organizational Structure

1.2.3: VISION:

A high performing Public Service that delivers quality services and a labour market which guarantees labour rights, productivity and access to decent jobs for all as well as a social protection system that promotes a decent standard of living by 2020.

1.2.4: MISSION STATEMENT:

To promote an efficient, effective and accountable Public Service, fair labour practices; enhanced labour productivity and social protection.

1.3: PROBLEM DEFINITION:

The establishment of a National Child Case Management Framework has been affected by a lack of a well coordinated, accessible Management Information System. Some of the major challenges as a result of this are:

1. Case Referrals are much more difficult to manage and follow up as one has to go through different papers and pages to find referral information. Some children referred for service access are failing to get the services or the officers failing to recall where the child was referred, particularly if they forget to record in the manual forms.
2. It is Difficult to keep statistics after certain period. Data collected over time is difficult to track in manual storage systems. There is also no proper system to enable managers to effectively supervise district level staff who are supposed to follow through all cases.
3. Some cases currently go without follow up or just end up being “forgotten due to lack of a system that not only track but also sends alerts to managers so that follow ups can be made. It has since been established that lack of a clear case tracking system to facilitate follow ups and holding courts accountable was the main reason why some cases literally “died a natural death”..
 - Some cases end up not reaching their conclusion hence curtailing well intended purpose of protecting children and responding to protection violations. As an example, only “24 per cent of alleged perpetrators of cases of child abuse reported through Childline in 2009 were arrested. A court hearing date was

obtained in 23.5 per cent of the cases, but only 8 percent of the cases went to trial” (Childline Zimbabwe and CCORE (2010)).

4. Efficiency on the part of the Officers has been a significant challenge as they would spend hours on end writing reports and compiling statistical reports for submission to Head Office. This affected the quality of time spent on actual service provision to the child
5. Data Safety – It is very difficult to maintain security on sensitive data since all records are kept in a manual format. There is always an inherent danger of records disappearing or destruction of files due to accidents and disasters. .

1.4: AIM:

The main purpose of the Child Protection Case Management System is to digitize and automate all the phases of the Case Management System cycle.

1.5: OBJECTIVES:

According to Mochal (2004) objectives are the end products that are to be brought or generated by a particular project. As a fact, every kind of a system has to meet certain targeted goals hence for this project; the researcher came up with the objectives below:

1. To develop a system that captures referral information, providing details of where the child has been referred to and what services the child is expected to receive.
2. To develop a system that stores all relevant information about children and their families in a way that is easy to retrieve and enhance user access to better managed data for timely case response and statistics.
3. To come up with a system that produces alerts to notify of pending or overdue cases for follow up actions
4. Develop reporting modules that produce standard reports that users can periodically generate on their own without the support of IT personnel.
5. Design user interfaces with different levels user rights and that restrict unauthorised access by allocating privileges to relevant users for different type of information.

1.6 DEVELOPMENT INSTRUMENTS:

- **C#:** it's a pure object oriented language.

C# is more type safe

It builds more dynamic and flexible relationships between classes

It's easy to use and to develop

- **SQL:** a scalable relational database.
- **Visual studio:** Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, and Windows Presentation Foundation.

1.7 JUSTIFICATION:

According to Yin (1994) Justification refers to a stage in where the sense of the study is clarified and at the same time evaluation of the project based on the goals is done. MoPSLSW handles large amounts of data and its systems would be benefit from computerization. There is need for quick interaction with customers

The challenges being faced by MoPSLSW can be addressed through utilization of a case management system. The system should be able to handle all the problems that emanate from the usage of the current system and solves these problems. The proposed system will enable to establish a standardised 'wrap-around' response service system that protects children from abuse, violence, exploitation and neglect within a coordinated continuum of care.

A case Management System will enable the Ministry to manage each case referred to its 65 districts countrywide. NCMS MIS will provides an Effective, Centralized repository of electronic documents to streamline the, Capturing, Classification, Retention, Disposition and management of critical documents. It will also assist managers to better supervise lower level staff and quality control work at all stages and levels.

1.8 CONCLUSION:

The Chapter has demonstrated that the current case management system in the MoPSLSW can benefit from enhanced functionality through improved system design. The chapter provided an overview of the MoPSLSW and how the case management approach evolved, including the design of the MIS for the approach. Areas that need enhancement in the current system design have been identified including a clear statement of objectives. During the planning phase, a feasibility study will be carried out to facilitate production of a work plan and schedule.

CHAPTER 2: PLANNING PHASE

2.1 Introduction

This focuses on the project development life cycle. The chapter discusses the following four important aspects in relation to the development of the proposed project:

- Outline of the gains of implementing the proposed system through the business value. The business value focuses on comparison between the benefits and the costs likely to be incurred in implementing the system by the organization.
- The feasibility study that illustrates the feasibility of designing and implementing the project given time and resource constraints.
- The risk analysis whereby the possible risks regarding the project are analyzed and appropriate measures considered to mitigate potential risks.
- The work plan which is a schedule that shows how the project development activities will be done.

2.2 Reasons for building the system

The system is being built mainly because of the challenges noted in the current manual system including:

- The difficulties in managing case referrals where officers spent considerable time going through volumes of files, papers and forms.
- Loss of data due to ineffective manual filing systems leading to managers failing to properly supervise case movements.
- Lack of case follow ups due to non-existence of alerts / reminder protocols.
- The fact that users were finding it difficult to compile manual reports and statistical returns for case reviews and reporting to head office.

Resultantly, it was noted that a new system that would address these and other associated challenges was needed. The system would ideally:

- Provide a clear case tracking system and facilitate follow ups.
- Reduce human interaction that is usually prone to errors.
- Provide efficiency when dealing with cases.
- Help to address response to protection violations like violence, abuse and exploitation and preventive welfare, providing a platform for ensuring systematic handling of child protection cases amongst multiple stakeholders in Zimbabwe.

2.3 Business Values

According to Williams et al, (1999) business values are “values that establish the reputation, goodwill and all intangible assets of an organization in the future”. Indeed a computerized system will bring positive change to the MoPSLSW because timely reports and case alerts generated by the system will help in the decision making of the ministry hence attainment of protection goals.

Additionally an electronic centralized database will help to link various districts which include Goromonzi, Zvimba, Bindura, Marondera and Bulawayo together as well as reduce paper work and data redundancy. To add more it will save management time and make work easy in so far as coordination of the districts is concerned. Business values include the following:

- The ministry personnel need reduction in work load
- Increased of efficiency and time saving to the current system paper system freeing up time towards attending to clients who report cases.
- Improved Ministry standards and high professionalism as well as reputational preservation.

2.3.1 Customer value

The clients will view the ministry as efficient and effective in their day to day activities. It will guarantee the clients of a better service and create and maintain a good clientele relation.

2.3.2 Employee value

As defined by the Oxford dictionary, an employee is a worker who is given work to do in return for payment in form of either wages or salaries. Employees of the Ministry can receive the below benefits.

- The business process is simplified by the proposed system as they will use minimum effort due to the “easy to operate” system being proposed.
- Extra morale is raised within the employee environment because of the system which is simple and friendly to use.
- Too much paper documentation is reduced by the new system and this is a benefit to the employees.

2.3.3 Managerial value

Wrapp (2012) characterized management as “those individuals who lead and control the employees and the organization at large in so far as the achievement of organizational goal is concerned”. The proposed project will have the following Managerial benefits:

- Increase in productivity and efficiency by employees
- Accuracy and quickness in the work done at the Ministry
- Easy practice of appraisal of employees.

2.4 Feasibility analysis

The stage of feasibility study, the scholar is much concerned with study of finishing the project positively with the help of technical, operational, social and economic factors Billy (2008). The research is going to offer recommendation to Child Protection Case Management system in advising whether they must continue with the project or not. Feasibility study is characterized into social, operational, economic and technical which means that the advantages must not exceed total cost for the system to be measured possible.

2.4.1 Technical feasibility

Resource and time limitations can constrain the feasibility of system building. (Bentley and Whitten, 2007), technical feasibility considers whether “up-to-date technology” can be operated with modern computers and whether current software and hardware can satisfy identified user needs. Staff willingness to undergo training willingness to adapt to change is important for success to be registered.

Technical feasibility this is when the important functions of the system over hardware, software and technical skills is evaluated. The Child Protection Case Management System can be built if the available resources are considered. One employee is the technical expertise identified to support development of the system. An in house Systems Engineer will be involved in the coding and construction of CMS. The researcher opted for C# and SQL for developing of the system. Almost all the technical requirements needed are in place and the researcher is convinced that technical aspects can be met. The hardware and software needed to develop the system are given below.

Hardware requirements

Component	Minimum required	recommended	Available
Computer Pentium 4 256 MB RAM,HDD 500GB	4	5	0
HP printer or better	1	2	0
Network cable	2m	6m	0
Network points plug inns	3	4	0

Table 2.1 hardware requirements

ITEM	Requirements
Server E5-2660v3 – 2.6 GHz/10-core /20Mb/95W 32GB 1866Mhz 14900R V2 (RDIMM) Smart Array P440ar/ 2Gb FBWC	HP Proliant ML10 Server- Intel Pentium dual core
Software	Windows 8 C# Sql Database Visual Studio

Table 2.2 Server requirements

ITEM	SPECIFICATION
Router	CISCO 1900
Switch	10 ports
Internet and Ethernet Cables	RJ 45 connectors
Uninterrupted power supply	Power Backup 220Va

Table 2.3 Network Equipment

Software requirements

Component	Quantity Required	Quantity Available
Visual studio and C# for designing and coding	1	0
MySQL for storage	1	0
Microsoft windows8	1	0
Microsoft office word for typing the document	1	0
Eset antivirus	1	0

Table 2.4 software requirements

2.4.2 Economic feasibility

Cost-benefit analysis can be used to define economic feasibility as a measure of the financial impact of the project. The Cost Benefit Analysis will be employed to assess the economic assessment of the proposed system. Feasibility has often been taken as the bottom line in many projects. Economic feasibility contents with the issues of whether the system can actually be built depending on finance availability (Brien, 1996). Powertel and Bijoe Cabling have most of the required software and hardware hence development will be easier. There are two types of costs which are operational and developmental to be considered in this project.

2.4.2.1 Developmental costs

These are usually incurred at the outset during system development. They can be estimated from the beginning can be refined at towards the end of each phase in the project. The case management development costs are:

Item	Quantity	Amounts
Microsoft Windows 8	1	100
HP desktop PC (Intel Dual Core 2.5GHz Processor 2GB RAM, 500GB HDD DVDRW)	4	800
Microsoft Office 2010	1	250
Eset antivirus	2	70
HP Proliant ML10 Server- Intel Pentium dual core	1	5000
Microsoft Windows Server Standard 2008 R2 Software	1	100
Powertel and Bijoe		1000
Printer	1	250
Total	10	7620

Table 2.5 development costs

2.4.2.2 Operational costs

Operational costs are incurred throughout the life cycle of a system. (Butt, 1994). These can include costs for maintaining the system throughout. They encompass both fixed and variable costs. Fixed costs are relatively at a fixed rate and are incurred at regular intervals. Variable costs on the other hand depend on usage. Some of the supplies that constitute variable costs include compact disks, printing paper, magnetic tapes and variable overhead costs. For the case management system, the following constitute operational costs,

Item description	Cost (\$)
Computer consumables	300.00
Hardware Maintenance	200.00
Stationary and consumables	300.00
Labour	500.00
Other	300.00
Total Cost	1 600.00

Table 2.6 Operational costs

2.4.2.2.1 Benefits

Benefits refer to the gains which are reaped when the system has been installed. These benefits include tangible and intangible benefits. The table represents benefits calculated as savings to be accrued through use of MIS for case management:

2.4.2.2.2 Tangible benefits

Benefits	Cost USD (\$)
Improve staff efficiency	3000
Reduced errors and omissions	2000
Reduced stationary expenses	500
Increase in business (New product)	2000
Case inquiry time saving	2500
Total Tangible benefits	10 000

Table 2.7 Tangible benefits

2.4.2.2.3 Intangible benefits

Item	Associated value(\$)
Improved clients loyalty	1700
Goodwill	1300
Total benefits	3000

Table 2.8 Intangible benefits

2.4.2.2.4 Cost Benefit Analysis

According to Boardman (2006) cost benefit analysis “is a systematic procedure for calculating the organizational costs and gains for a particular scheme”. This is important to determine whether it is economical to proceed with a project or not. It involves the comparison of the overall expected costs against total expected benefits.

The cost and benefits analysis include three tools which are return on investment, net present value and payback period analysis. However the decision whether to quit or continue with a project is made when the benefits outweigh the costs.

Costs	First year	Second year	Third year
Cost	Amount(\$)	Amount(\$)	Amount(\$)
Development cost	6620		
Operational costs	1600	1600	1600
Total costs	8220	1600	1600
Benefits			
Tangible	10000	10000	10000
Intangible	3000	3000	3000
Total benefits	13000	13000	13000
Net benefits/costs	4780	11400	11400

Table 2.9 Cost benefit analysis

The above calculations show that the project has potential to produce positive benefits. A conclusion can be drawn that the project can be economically feasible.

2.4.2.2.5 Payback Period

According to Wood (2001), the time it takes in years for revenue from investment in a scheme to provide a return matched to the initial investment is the payback period. “Payback period is the easiest way of measuring how early or how late a particular investment is going to return the capital funds used and it is one of the simplest investment appraisal techniques”. For a system like the Case Management MIS the payback period is considered mostly in terms of time savings, efficiency and effectiveness return on investment over time.

According to <http://accountingexplained.com/managerial/capital-budgeting/payback-period> there are several advantages and disadvantages of payback period.

The advantages include:

- That it is simple and easy to understand and compute
- It is used universally and is easy to understand.
- It lends more weight on liquidity for decision making on investment proposals
- It addresses risk
- Can provide an added advantage of capital expenditure calculation

The disadvantages include:

- Time value of money may not be recognized.
- It ignores profitability and gives high emphasis on liquidity.
- Cash flow occurring after payback period may not be considered

The Payback period for the proposed system is:

The Net Benefits of the first year which is \$8220 is far much greater than the initial investment of \$4780 which clearly shows that the initial investment is returned in the very first year, thus the payback period is 1 year.

payback period is therefore = **1 year**

Return on investment

A return on investment according to Alvarez (2007) 'is a financial measure to see the gains which can be brought by an investment in a particular gauged period of time'.

Return on investment was used in the calculation of the rate at which the project brings back its initial capital. It is calculated as follows.

$$\begin{aligned}\text{Return on investment} &= \frac{(\text{total benefits} - \text{total costs}) * 100}{\text{total benefits}} \\ &= \frac{(39000 - 11420) * 100}{39000} \\ &= 71\%\end{aligned}$$

Net Present Value (NPV) Analysis

It shows how viable the project is in a particular period of time .thus it really shows the time value of money.

$$\text{Net Present Value} = \frac{\text{total benefits} - \text{total costs}}{(1 + r)^n}$$

r = interest rate (15 percent)

n= number of years

$$\begin{aligned}\text{NPV after 3 years} &= \frac{39000 - 11420}{(1 + 0.15)^3} \\ &= \underline{\underline{\$1.52}}\end{aligned}$$

The net present value of the project is \$1.52 and it is positive thus it shows that the project is viable enough to be undertaken. This NPV considers the gains brought by the project in its entire life span of usage.

2.4.3 Social Feasibility

Schawable (2008) characterized social feasibility as a measure of how the stakeholders combine well with the desired system within an organization.

The proposed system can change the attitudes of the employees since the system provides fast efficient and easy way of tracing the cases. The proposed system can boost the morale of the Case Management Officer (CMO) workers since it is not tiresome and confusing as compared to the current system.

Towards the Child Protection Committee (CPC), Community Case Workers (CCWs) and client side, the proposed system could enable them to believe in the Ministry and to be supportive since the process of placing a child in a place of safety will be fast and also efficient to the process of connecting a child to a specific service provider to address an identified need.

2.4.4 Operational feasibility

As indicated by Kendal (1999) operational feasibility is the eagerness and capacity of administration and workers to work with and utilize the proposed project under development. The principle objective of the study was to distinguish how potential end clients will consider on the system and how the proposed framework will solve their problems.

2.5 Develop a Work Plan

Phase	Start	End	Duration in weeks
PROPOSAL	13/07/2015	27/07/2015	2
CHAPTER 1	27/07/2015	03/08/2015	1
CHAPTER 2	04/08/2015	11/08/2015	1
CHAPTER 3	12/08/2015	01/09/2015	3
CHAPTER 4	02/09/2015	09/09/2015	1
CHAPTER 5	10/09/2015	17/09/2015	1

Table 2.10 Work Plan



The table above indicates the time required to complete the project. The start time and completion date for each phase is indicated. The proposed project will run for 9 weeks starting on the 13th of July 2015 and ending on the 17th of September 2015.

2.5.1 Gantt chart

As postulated by Bentley and Whitten (2007) a Gantt chart is a pictorial representation of a schedule which shows the time frames in which certain tasks or activities must be accomplished.

ACTIVITY	WK One	WK two	WK Three	WK four	WK five	WK Six	WK seven	WK eight	WK nine
Project proposal	■	■							
Planning phase			■						
Analysis phase				■	■	■			
Design phase							■		
Implimentation phase								■	
Maintenance phase									■
Documentation	■	■	■	■	■	■	■	■	■

Table 2.11 Gantt chart

2.6 Risk Management

Risks are simply potential problems. Risk management involves assessment of risks that have the potential to affect a project, monitoring the risks by managers who also have to address problems as they arise. (Sommerville, 2011) Below is a diagrammatic representation of risk handling techniques.

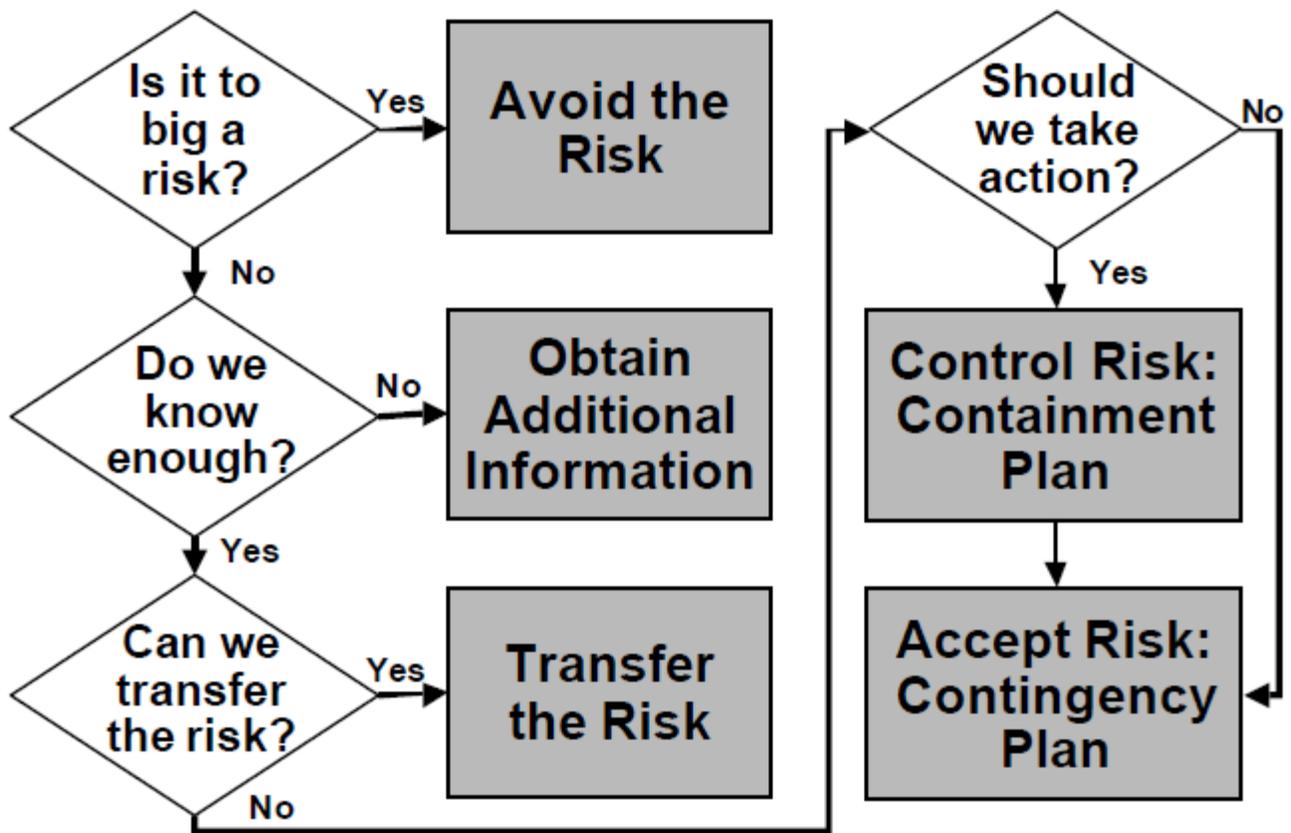


Fig 2.1: Risk handling techniques

Source: William (1999)

The risk management phase involves identifying and evaluating what might go wrong and trying to foster solutions before the project is derailed by unmitigated risk.

2.6.1 Technical Risks

Technical risks result from excessive constraints caused by limited experience and over reliance on third parties outside of the primary agency. They typically relate to languages, magnitude of the project, its functionality and choice of standards and processes. Different interfaces used by users before could pose a challenge when a developer comes up with a different interface that does not appeal or seem familiar to users. Users of the child protection case management system MIS may be unable to effectively maintain the system or restore it in the event of a crash it being a new system, it will need to be backed up and fully supported. Ongoing support will be provided to ensure that there constant back up until users again familiarity and are able to continue operating with minimal technical support.

2.6.2 Economic Risks

Costs of hardware and software, which can escalate has the potential to cause project abandonment if they are considered unattainable. The case management MIS has been budgeted to ensure adequate financing. A contingency budget will be considered to mitigate the risk. Therefore a reserve account should be maintained to cater for these risks if they are to occur so that no losses will be occurred.

2.6.3 Personnel /Social Risks

These have been noted too often involve challenges with staff capacity, ethical considerations, and team sprit within staff. No such risks are anticipated in the case management system. Training and capacity development will be considered in the event that personnel/social risks are identified.

2.6.4 Management Risks

According to Westfall (2001) these involve poor planning, management without requisite competencies, limited authority and communication challenges. Planning tools and information and simplified information briefs may be developed and used for management briefings to assist the in planning and decision making.

2.6.5 Contractual and legal risks

Government regulations, changes in the market, changing needs and operational health and safety issues may pose contractual legal risks. These are not anticipated in the case management MIS since this is a government project.

2.6.6 Other Risks

These include equipment and supplies availability, inadequate tools, inadequate facilities, unavailability of computer resources and slow response times. The economy is still struggling which is negatively affecting the availability of electricity to ensure full effort devotion to the project. Poor electricity supply can lead to challenges in implementation in the proposed 5 districts.

2.6 Conclusion

All project activities have been taken into consideration and system development is now set to commence. The reasoning being is:

- The users together with the Ministry senior management have agreed
- Task scheduling and establishment is complete
- Consideration of risk has been considered
- Overall project feasibility has been considered

The project is therefore viable for the develop to proceed.

CHAPTER 3: ANALYSIS PHASE

3.1 Introduction

According to Whitten (2003), the process of identifying and verifying statistics and demonstrating relevant information through statistical transformation is taken as the analysis phase. It is the stage where the project lifecycle begins. The stage is critical in the system development lifecycle. It also entails a complete study of current system in terms of how an organization conducts business.

3.2 Information Gathering Methods

According to Wyse (2012) the main purpose of the phase is to compile requirements. It is about understanding and recording the needs of users. According to Collins (2003), the chosen methodology has a strong bearing on the choice of research method and influences decisions on data collection and analysis including rationale, location, approach and timing. To assess effectiveness and success of Child Protection Case Management System, data collection was done in 5 Districts. The data was collected from senior staff of the DCWPS, District Staff and staff from major supporting stakeholders. The following was used to collect data:

- Interviews
- Focus Group Discussion
- Observations
- Questionnaires

3.2.1 Interviews

Bentley and Whitten (2007) define interviews as a truth discovering system where analysts gather data from people through a face to face situation. There is guarantee of more accurate information from the respondent with regards to screening questions such as gender, age, or race.

It has been suggested that most case studies are mostly about human affairs and give well-informed respondents an opportunity to provide important insights into the situation (Yin, 1994). The interview approach is the preferred way of efficiently collecting quality information (Marshall and Rossman 1995). Key informants are important contributors to any system review and development process as they provide a more technical perspective on how the system operates and what the user needs are. Qualitative in-depth key informant interviews were held with the Deputy Director for Child Welfare at head office, the Provincial Child Welfare Officer and District Child Welfare Officer.

Structured interview –This is also known as a formal interview. The questions are asked in a set/standardized order and the interviewer will not deviate from the interview schedule.

Unstructured interviews – These are much more casual and unrehearsed. They are sometimes called informal interviews. To guide the process, unstructured interview guide was developed and used. The guide is attached as Appendix B.

3.2.1.1 Advantages

- The main advantage was that it allowed the researcher to probe and gain a deeper understanding, not only from implementers but also from the people who designed the programme like the Deputy Director who is also responsible for charting the policy direction.
- Responses were provided timely
- High rates of investigations were completed
- Gestures and postures were observed and interpreted

3.2.1.2 Disadvantages

- The main disadvantage was that the knowledge provided particularly by higher level officers like the provincial head and the Deputy Director was sometimes not the reality on the ground.
- It was also time consuming.
- In some cases respondents were not very free to contribute responses to questions asked by an officer from Head Office.

3.2.1.3 Findings from Interviews

From the discussions, it emerged that there are different roles at each level of national, province and district in the case management system. The national office mostly provides an oversight policy directional role, technical and administrative supervision to district office. The provincial level role is mainly that of support and supervision and quality control to the district office which the system implementation hub is. The district level is responsible for the implementation of the system. They receive clients and assist them all the way up to case closure.

3.2.2 Focus Group Discussion

The researcher took advantage of a workshop for all districts conducted at Head Office to invite the district head, the Child Welfare Officer and the Case Management Officer from 2 of the districts covered in the study. A focus group discussion was held with them, providing rich data on the realities of the case management system and how implementation is currently taking place. The needs of the teams and the gaps in the current system were noted. It was interesting to note the various and sometimes divergent views that were coming from the two districts.

3.2.2.1 Merits Discussion Groups

- Instantaneous and first-hand information was acquired.

3.2.2.2 Demerits Discussion Groups

- It was cheaper to carry out and arrange
- Focus group discussions were time consuming especially when the arguments arose from the participants

3.2.2.3 Findings from focus group discussion

During data collection the interviewer noted that all five districts have executed the care plan, but the child is linked to a service provider without enough resources in some instances. It also emerged that sometimes referrals made not carefully monitored or followed up to check if the various services have actually been accessed by the child or not.

3.2.3 Questionnaire

The researcher chose the use of questionnaires as the principal methods for data collection. According to Pressman (1994) questionnaires are sets of written or printed questions asked by the researcher with different choices of answers, devised for the purpose of a statistical study.

Open ended questionnaires: An open ended questionnaire was used by the analyst as a guide in the gathering of the necessary information that will be provided in Appendix B. The questionnaires were emailed to all the five districts giving the researcher an added advantage of collecting data from more people in the full sample.

Closed questionnaires: The interviewer avoided closed ended questionnaires as they would limit responses typical of quantitative data collection yet there was a deliberate aim to collect information of a qualitative nature

The reason why the questionnaire was posted to many people in each of the five districts is because the researcher wanted to control of non-response bias- especially which is especially true for those questionnaires distributed by post. Non-response is often crucial in a questionnaire survey because your research design will be based on the assumption that you are going to generalize from the sample to the population.

3.2.3.1 Merits of questionnaires

- Questionnaires were relatively easy to analyze
- Questionnaires were cheap whereby large sample of the given people in a short period of time were contacted.
- It promoted free participation, precise and honest because of unidentified input hence effective information was obtained.

3.2.3.2 Merits of questionnaires

- There is lack of validation.
- It was difficult to tell how truthful a respondent was being
- Questionnaire lacked honesty of additional explanation on apparently vague responses and questions.
- Respondent may read differently into each question and x therefore reply based on their own interpretation of the question.

3.2.3.1 Findings from Questionnaire

Responses from the questionnaires were varied and gave a glimpse of how the current approach is not standard as various views and way of doing business were shared. Most of these are reflected in the analysis section of this report.

3.2.3.1.1 Target Population

To evaluate the effectiveness and success of the Child Protection Case Management System, a total of five districts offices of the DCWPS were selected. All the Districts were chosen as these are the ones where the system is operating. From the organograms it was determined that each office has three officers who deal directly with Case Management. These three most senior officers were targeted hence there were a total of 16 respondents in total. The districts offices chosen were as follows.

DCWPS District Office	Target Population	Number of Respondents
Rusape	3	2
Bindura	3	3
Marondera	3	3
Bulawayo	3	3
Zvimba	3	2
Total	18	16

Table 3.1 Sample Selection for Evaluating Effectiveness of Child Protection Case Management Information System

3.2.3.1.2 Response Rate

For DCWPS officers, an electronically distributed questionnaire was considered suitable. This is because it allowed an effective reach to the District Officers without necessarily visiting their offices to deliver the questionnaires. A high response rate 89% was delivered; all the participants received and fill in their responses from questionnaire except for the 2 officers who were off sick.

$$\frac{16}{18} \times \frac{100}{1}$$

=89%

3.2.4 Field Observation

The visual of mechanical or human study of an activity in order to gain or gather information or to learn about people's trends, character and behavior in their regular setting is referred to as observation. As the consultants of the company do their day-to-day duties the analyst observed. The observation card sheet will be shown in (Appendix D). The processing cycle, reports, records, consultations and examining each form were observed by the analyst. Person working with the system were also taken into account. When checking for validity of data gathered using questionnaires and interviews the researcher used a fact finding method called observations.

There are basically two types of observation which are participant and non-participant observation. In participant observation, the researcher is actively involved and intervenes in the research environment, embedding him/herself with the processes and interactions under study. This allows high external validity as the environment remains naturalistic. Non-participant observations involves the researcher observing behaviors and interactions as they naturally occur. There is no researcher intervention and there is no attempt to manipulate variables.

3.2.4.1 Advantages

- The data was collected where and when an event or activity was occurring.
- There was less interference with the people's work schedule
- Observation was not rely on participants willingness to give information

3.2.4.2 Disadvantages

- Participants usually perform better when they know they are being observed.
- Prone to observer bias.
- Observation was not added up understanding of why people act e the way they do.

3.2.4.3 Findings from field Observation

The main finding from field observation was that one cannot explore the causes of certain behaviors and sometimes people may adjust their behaviors if they become aware that they are under observation.

3.3 Analysis of the Current System

The MoPSLSW is currently using steps documented in the Child Protection Case Management framework document as the business process flow to address preventive welfare, violence and abuse to children. The analyst has discovered that there is lack of coordination, accessible Management Information System across the department. This has resulted in inadequacy and an incapacitate response, tracking and management of cases. Due to mismanagement and lack of adequate information, some cases end up not reaching their conclusion hence curtailing well intended purpose of protecting children and responding to protection violations. Also there is lack of clear case tracking system to facilitate follow ups. Case Referrals becomes more difficult to manage and follow up as one has to go through different papers and pages to find referral information because case files are stored in filing cabinets which unauthorized personnel have easy access to and lead to manipulation of confidential information .Moreover, Officers will take long time to find a document because one has to look for the name of the file in which it was filed than go to the cabinet if it's not then the officer will have to check with the officers. Furthermore, there is no proper system to enable managers to effectively supervise district level staff who are supposed to follow through all cases. This has caused extremely high duplication of data because there is a lack of validation methodologies in a manual record and filing system.

3.3.1 Inputs to the current System are:

- Client particulars
- Referral details
- Case details

3.3.2 Processes

- Assessing a case
- Case planning
- Implementation of case
- Review care plan
- Case closure

3.3.3 Outputs

- Clients files
- Assessment outcome
- Care plan files
- Reasons for closure

3.4 Data Analysis

According to Presto (2002) defines data analysis as a process of modeling, transforming, cleaning and scrutinize data. The main aim of this stage of analysis is to demonstrate the processes and data flow in the current system. It additionally provides a visual picture of what is really happening within the Ministry with the current system. The methods include investigating and displaying of procedures included in the current system think of essential information structures that will fit current system into the proposed system. The system analyst will make use of dataflow diagram and context diagram to demonstrate the outputs, processes and inputs of the current system operations

3.4.4.1 Context Diagram

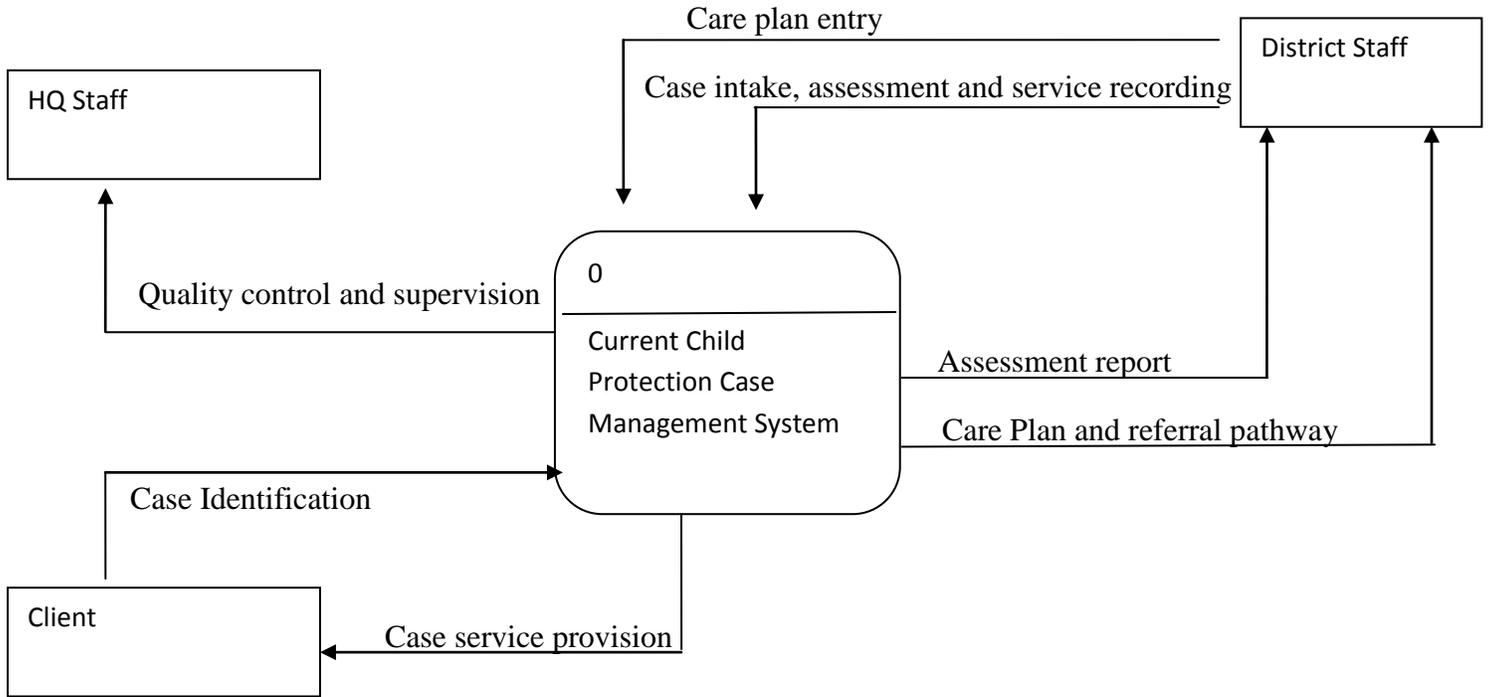
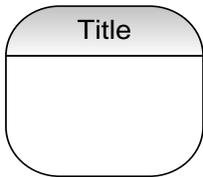


Fig 3.1 Context Diagram

Context Diagram Key



Entity



System Processing



Flow of information

Data Flow Diagram

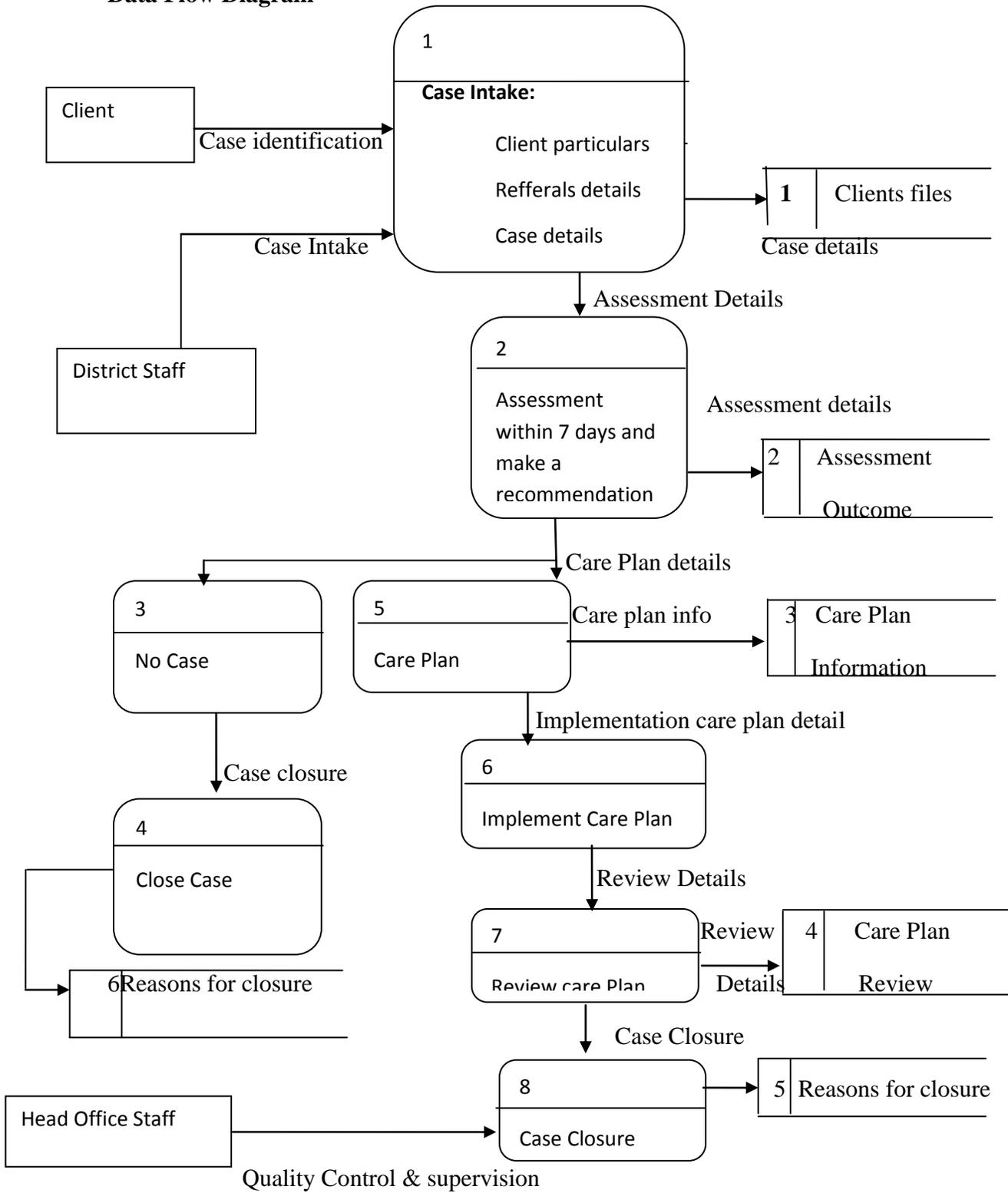
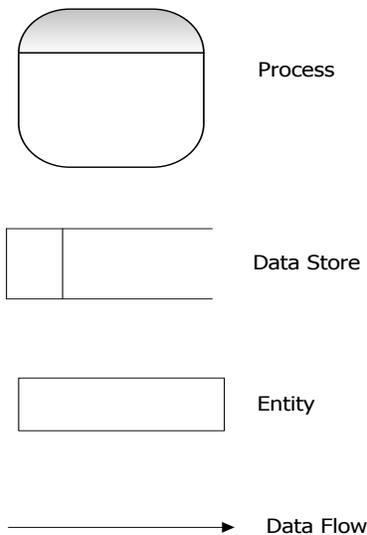


Fig 3.2 Data flow diagram of the current system

Data Flow Diagram key



3.5 Weakness of current system

The current system is linked to many different weaknesses for example there is lack of security where by clients documents are stored in filing cabinets which unauthorized personnel have easy access to and could lead to manipulation of confidential information.

- There is lack of adequate information and mismanagement some cases end up not reaching proper closure
- The researcher discovered that there is also lack of a case tracking system to facilitate follow ups
- The current system is too slow in case referrals of client which results in more difficult to manage and follow up cases.
- It is time consuming when an officer is looking for a file of a client it takes long time to find because one has to go the cabinet with heap of files.
- Case Referrals are much more difficult to manage and follow up as one has to go through different papers and pages to find referral information

3.6 Evaluation of system development alternatives

The alternative of system development is evaluated to observe whether the system is possible to proceed with the development or to abort since it may cause the company to incur extra cost. To resolve the problem that has been identified on the above, another possibility has been assessed namely:

Improving the Current System: by identifying Child Protection Case Management system loopholes and then try to rectify them

Off-The-Shelf Package (Outsource) : This is whereby purchasing an already developed software package and will be customised and implemented following suppliers' stipulations.

In-house Software Package Development: This involves undertaking our own design, analysis of the system and then package development that will satisfy the desires of the Ministry.

3.6.1 Improvement of the Current System

It involves detecting system processes and its weaknesses and improving them. Improving the system currently used doesn't necessarily mean the operation will change rather there will be improvements and changes of needs that will satisfy the customer.

3.6.1.1 Advantages

- It is cheaper as there will be no need of additional software, hardware and training since the system users will be already aware of the system prevailing.
- It also saves time in terms of implementation since the system is not developed from beginning.

3.6.1.2 Disadvantages

- Problem Legacy – prevailing glitches may sneak into the system undetected then it might produce unwanted output in the future
- Complicated Glitches – A number of the difficulties may be tough to fix for instance the system performance ,data amount that can be hold the system

3.6.1.3 Evaluation:

Due to problem inheritance it might be considered not viable for the development of the system since the system may cause problems in the future thereby it might worsen the situation by making the system more costly than the current system which was used before.

3.6.2 Outsource (Off-The-Shelf Package)

According to Griffith (2005) outsourcing is the vital utilization of outside expertise or assets to perform duties generally being done by inside staff and resources Off-The-Shelf Package stated above by the researcher comprises of software that are already programmed and existing for sale to either a company or to a general client from a vendor that is this type of programs are common to all customers for the application to be produced it doesn't take into consideration the requirements and specifications of the organizational so on average the package will be not tailor made.

Below there are clarified advantages and disadvantages of outsourcing system that is:

3.6.2.1 Advantages

- No Development Cost – since the system already exists there is no need of development once the system is bought its immediately installed.
- Relatively Low Cost – this is because costs are spread among various customers so there is no need to charge high prices.
- User Friendly – some off the shelf package can be ease of use that is the user must choose carefully.
- Less time Implementation – as the system is available and designed already, outsourcing software does not need more time to be implemented.
- Usually outsourcing from outsiders is characterized by quick and better results since the outside providers have some equipment and idea as well as expertise that are not possessed internally.

3.6.2.2 Disadvantages

- Lack of management authority over the outsourced property
- Require the vendor in case of future upgrades might not address critical user needs thereby making it dependant on the vendor.
- Vendors may not be flexible enough to meet fixed contracts depending on their own objectives.
- There is uncertainty of data integrity and confidentiality when contracting outsources.
- It might be difficult to support system in the event that the vendor relocate
- It might be expensive to the Ministry since licence fees are paid regularly

3.6.3.3 Evaluation:

In conclusion it's unworthy to buy an Off the Shelf system because its drawbacks have offset their benefit that is the alternative should be rejected because it might cause the firm to earn loss because of increased costs of Outsourcing.

3.6.3 In-house Development:

According to Lewis (2006) In house development is a software development strategy where by software development is done internally within the Ministry; it implies that the very full time software development team of a particular company would carry out system development. Documentation will be done on each and every aspect undertaken on the development process to convey reports. In-house development is ideal for MoPSLSW since the Ministry has its own IT team in the department which is well technically to do software development.

Developing a new system is associated with some merits and demerits as follows

3.6.3.1 Merits

- It is less expensive to maintain simply because that the system was developed within the Ministry.
- High Satisfaction of Users because users requirements and needs are taken into consideration by the developer thereby increasing the probability of system success.
- Easy System Support – As the documentation of the system is done within the Ministry during the development of the system it will be easy for the IT team to support and maintain or upgrade the system since there is somewhere to reference when a problem arise.
- It provides leadership and control over system development
- It's cheaper to train users
- Guaranteed system possession by the time the system is initially developed because requirements match with specification of the system
- In-house development promotes full time participation of the employees in the system hence the Ministry will add expertise and more skills to the staff

3.6.3.2 Demerits

- More time is required in staff development
- A tailor made system represents a danger of lose of funds if the system neglects to meet expressed goals.

3.6.3.3 Evaluation Approach

In conclusion the challenges that are faced by the Ministry currently can be determined and it is worth developing a new system because the benefits are outweighing the demerits. In house system is better than buying an Off Shelf Package because the in house system meets user requirements and needs rather than an Off Shelf system that will be containing some modules which are irrelevant to the user and at the same time when coming with system the user requirements are not taken into consideration that is why the system fail to succeed since the clients might reject it. Lastly there will be no dependency syndrome; there is no need to consult the vendor just because the package will be internally develop

Alternative	Cost
In house development	\$6 620
Outsourcing	\$7 000
Improvement	\$6 700

Table 3.2 Cost summary of alternatives

3.6.4 Recommended Alternative

In house development is less expensive to implement as compared to other alternatives. Furthermore Users of the system will assume ownership of the system since their requirements are incorporated during system development. Above all the system is developed basing on user requirements and specification internally whereas outsourcing there is a great danger of losing data integrity and confidentiality. In house system will be perfect choice for MoPSLSW considering the requirements demanded by Child Protection Case Management System whereby each case has its own specific requirement. The system has room for expansion in respect with changing technology.

3.7 Requirements Analysis

According to Setende (2005), requirement analysis is an obligation and optimum conditions required to achieve the development of desired system and these requirements take into account various needs of system users and developers. There are two types of requirements which are functional and non functional requirements and are explained below:

3.7.1 Functional Requirements

According to Bentley and Whitten (2007) these requirements are a breakdown of a system services which must be provided by a system. Functional allows the system to improved performance capacities. The functional requirements comprise of data management, processing, technical details and calculations as well as other specific functionality that define what the system must achieve.

- An automated centralized electronic database is ideal rather storage of files in form of hard copies in cabinets. The database will ensure data integrity, data redundancy, data security; data portability, data administration and updates to any given set of data are immediately received by every end user. Furthermore data kept in the same location is easier to be changed, re-organized, mirrored or analyzed. Any staff at different locations within the Ministry can access data.
- Case Management System should allow input of data, must be able to compute and manipulate the data
- Should be display a list of pending cases available
- Use of query processing modules to enable cases.

3.7.2 Non-Functional Requirements

As explained by Case (2006) Non functional requirements are the system constrains that are met when the system is being developed. It defines how a system is supposed to be. Furthermore according to Cashman and Rose (2009) it characterized as requirements not directly inclined to system. Lastly but not least Rosa (2000) described these requirements constrains as software global attributes, faulty tolerance as well as security.

Reliability: The CMS system should have the capacity to be trusted by users in saving its purpose

Usability: -The user interfaces of a CMS system should be simply and easy to use thus our system should be user friendly. The system interfaces should be able to communicate to users with understandable language through notification and alert messages. Case management system there should be able to show the alert messages of pending cases.

Error Handling: It's an obligation that due to system validation errors for data entry should be handled.

Access control: system should perform identification, authentication, access approval and accountability of entities through login credentials including passwords.

Maintainability: It's an obligation that the system should not be developed with complicated tools in order to allow easy maintainability .The development tools shouldn't be too expensive to buy, scarce or difficult to find online help. The system should be flexible to add or remove an extra task.

Response time: Is the time elapse in between user sent an enquiry and the response to the enquiry .The CMS should offer quick retrieval of cases captured in the system as well as accessibility of data whenever it is required.

Operability: The system must be compatible with other systems. For instant to develop a module with common identifier for children on case management who are also getting education assistance through BEAM and who are in households receiving Harmonized Social Cash Transfer (HSCT) as well.

Security: The system should request strong password for a person to capture, process and review cases that is the password must be a minimum of 8 characters mixture of digits and characters. The password must not be easy to guess.

Backups: Utilization of storage backups inside the Child Protection Case Management system is encouraged for MoPSLSW will assure that valuable data is safe and can be still recovered in case of accidents which could occur including theft, fire or virus.

3.8 Conclusion

In conclusion the analysis phase brought to light the challenges the current system is facing. It has brought to light the possible solutions that are to be implemented in the design stage .Given that all the alternatives that have been analyzed and evaluated it is feasible to develop the proposed system considering all the task undertaken in this stage. Both the functional and non functional requirements have been brought to light and can be vital in the designing phase.

CHAPTER FOUR: DESIGN

4.1 INTRODUCTION

This phase of system development focuses on convert the logical system model into the physical model. The designing of the system is a methodical advance which considers the simplicity or difficulty of various system problems and then presentation of how the system model should work physically (Dubbey, 2006). This chapter will provide a detailed overview of the physical and database design, logical and program design, interface and architectural design. This includes use of diagrams in DFD format, ERDs, Tables, Databases and interface strategies.

4.2 System Design

According to Doug (1994), system design is looks at the process involving the definition of key elements making up the architectural design of the system. This looks at modules, components and data related to the system. Design of a coherent and efficient system is positioned to satisfy the needs of an organization of a business entity. The system should be operative, reliable and maintainable

Reliability: It is important to ensure reliability. The system being proposed has been designed in such a way that reliability will be guaranteed as much as possible. The Child Protection Case Management System will remove the possibility of data loss when in use and will mitigate the challenges faced with the current system.

Maintainability: The system will be compatible enough to update, modify or even transformed to meet the demands of the Ministry. Regular system updates will be undertaken in response to issues identified during periodic checks.

Operability: The proposed system will have simple design that is user friendly. Easy interface will be provided which will save costs incurred when cases are processed.

Efficiency: A well-designed system should be in a position to execute in a certain time frame. The system will be able to produce reports in the anticipated time. Moreover, it should do work at the right time and provide quick access to the users so that they will not waste time trying to get a certain job done.

Effectiveness: The proposed system should be able generate vital objectives.

4.2.1 Description of Proposed System

The system is designed for the child protection and probation welfare department to increase efficiency when dealing with the cases. After the user being created in the system by system administrator and given all necessary credentials, it depends with the level and responsibility of the user. The user will then login with his /her username and password. The child protection case management system will allow users to process the caring of a child in the system within 24hours. When the case is recorded, the officer will then assess a case by collecting more information within 7 days. After the assessment of the case the action will be taken to make sure the child's needs are met and they are protected. Then execution of the care plan. A case can be referred to another strategic organization which has capacity to handle certain aspects of the case. After that the system will allow users to check that the child's care plan is on track and continues to meet his or her needs. Then the system will permit the closure of a case upon reasons that the situation is resolved or child dies

4.2.1 Context Diagram

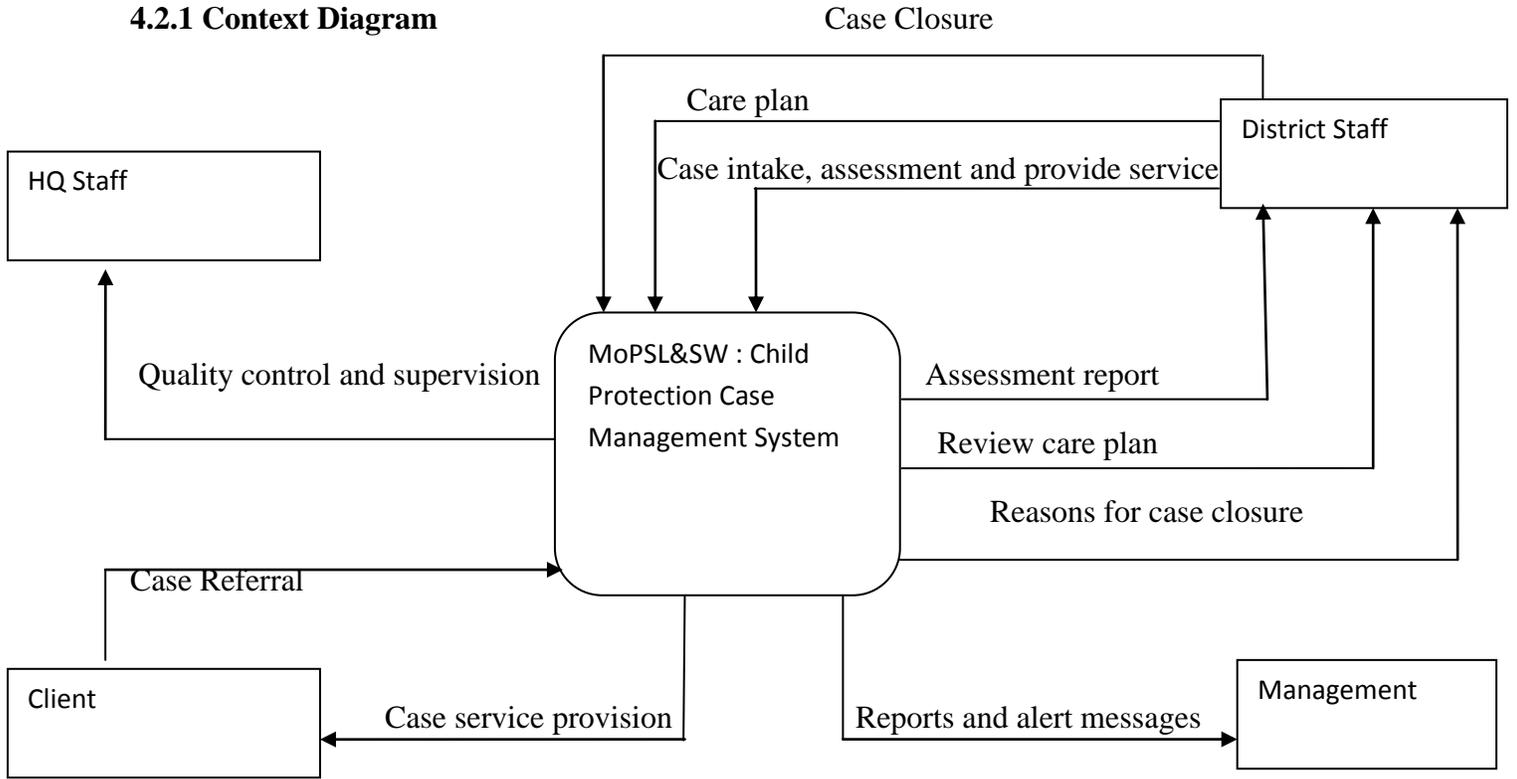
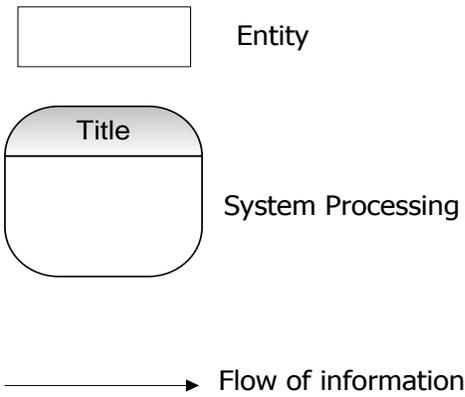


Fig 4.1: Context diagram for the proposed system

Key



4.2.2 Dataflow Diagram

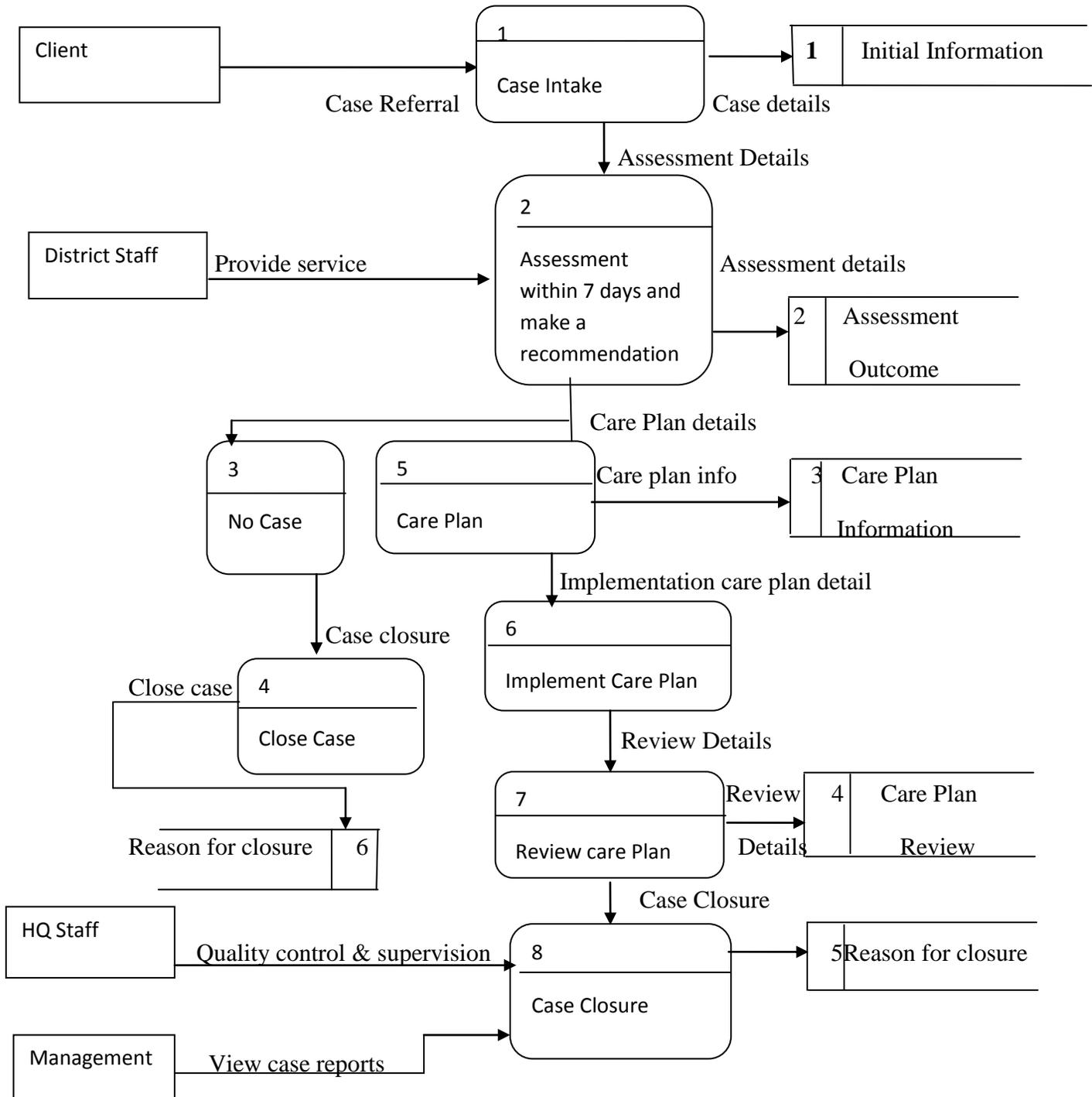
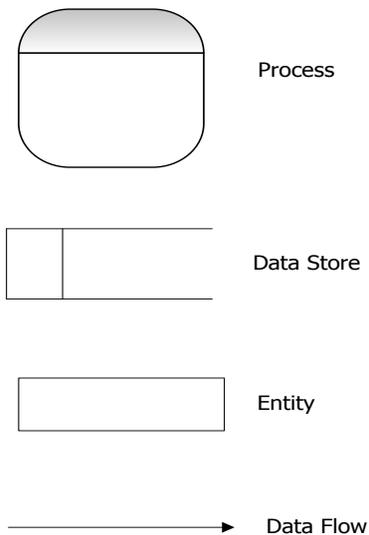


Fig 4.2 Data Flow Diagram

Key



4.3 Architectural Design

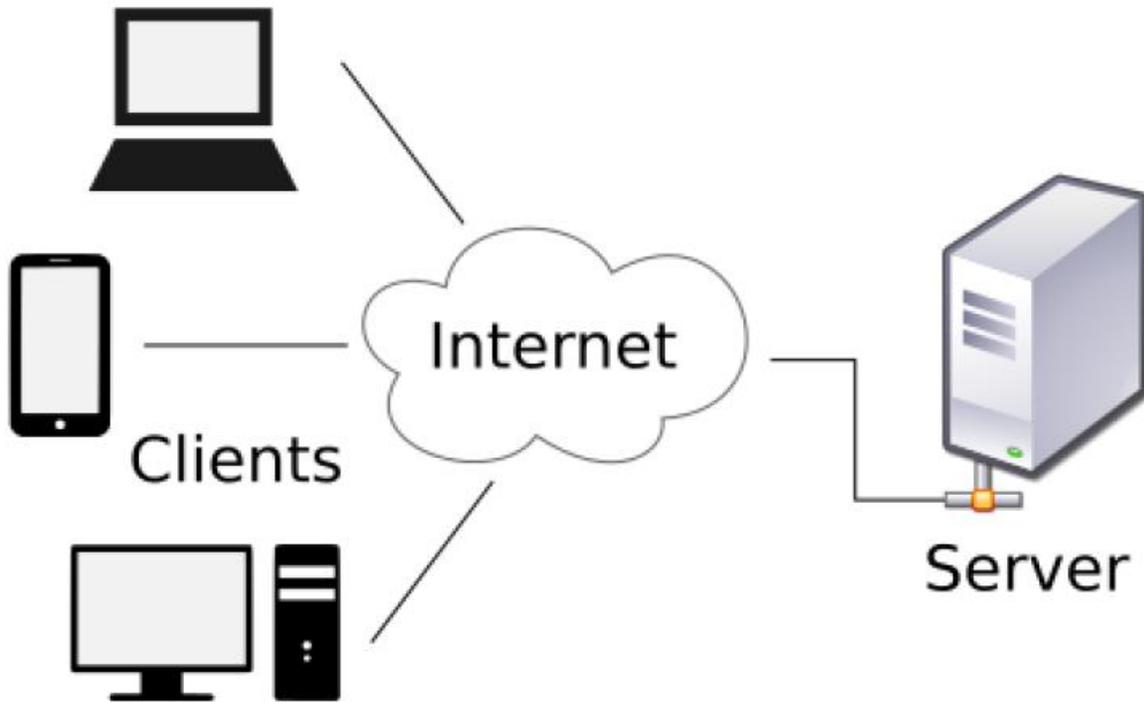
Bonley (2005), explains architectural design as the stage concerned with hardware set up for the anticipated system. This involves identifying the subsystem, building it up and a mechanism for communication and control. The stage basically seeks to ensure that the hardware and software issues that may affect production are well taken care of. Hardware components are also measured with fairly low costs. Local area network (LAN) will be used to develop the proposed system. For the sake of power cuts limitless power supplies are going to be available and also surge protectors are going to be in place to safeguard against electricity faults. The architectural design will include the following components:

4.3.1 Ethernet cables

These cables they make available of connections in the LAN and WAN networks.

4.3.2 Server

It's a special computer used to store all the necessary information for MoPSL&SW of child protection .The server will be operating in a networked ministry whereby all the clients send requests via a server using network. Apache will be used and the diagram below outlines the client server relationship.



Source:

<http://netacad.com>

Fig 4.3 Client Server

4.4 Physical Design

According to Waxon(2008) defined physical design as the outlay of the hardware components in the system, that is how the devices relate to each other, their interconnectedness and interaction within the proposed system. In fact it is how the devices communicate and usually user computers should rely to a server machine as well as the database servers for retrieval of data from stored files. Moreover, there is an emphasis on the relationship between all hardware components.

4.4.1 Application server

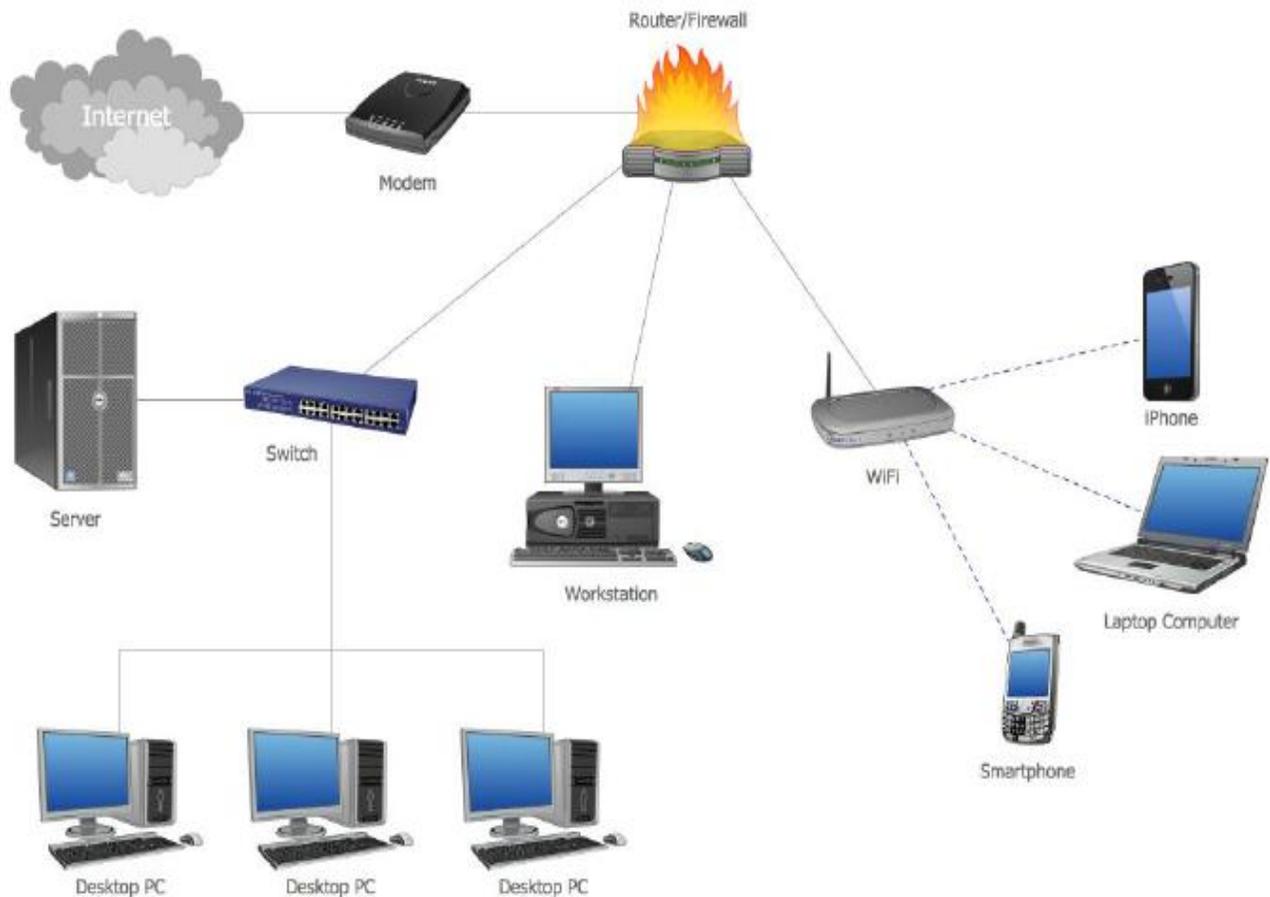
The application server will host all the relevant information for MoPSL&SW and will store the databases as well. This will maintain case details and security measures.

4.4.2 Router

The router is a device designed to forward data packets along networks. The router will be used by MoPSL&SW to manage information sharing with and between districts. This router is the one responsible for communicating with the internet service provider (ISP) –Powertel

4.4.3 Firewall

Is a network security system that is found either in hardware or software form whose main purpose is to control network traffic, both incoming and outgoing based on a set of rules. There is always a threat of security breaches, including hacking and firewall secures the network from such threats. This helps to keep the network secure from threats such as hacking. Execution of incoming or outgoing packets is determined by the firewall.



Source:

<http://netaccad.com>

Fig 4.4 Proposed system physical design

4.5 Database Design

Whitten (2003) defines a database as an organized system with the capability to store information in a retrievable manner. Data is stored and organized in the form of tables and schemas within the database. There are three layers – namely conceptual, internal and external view layers in a database.

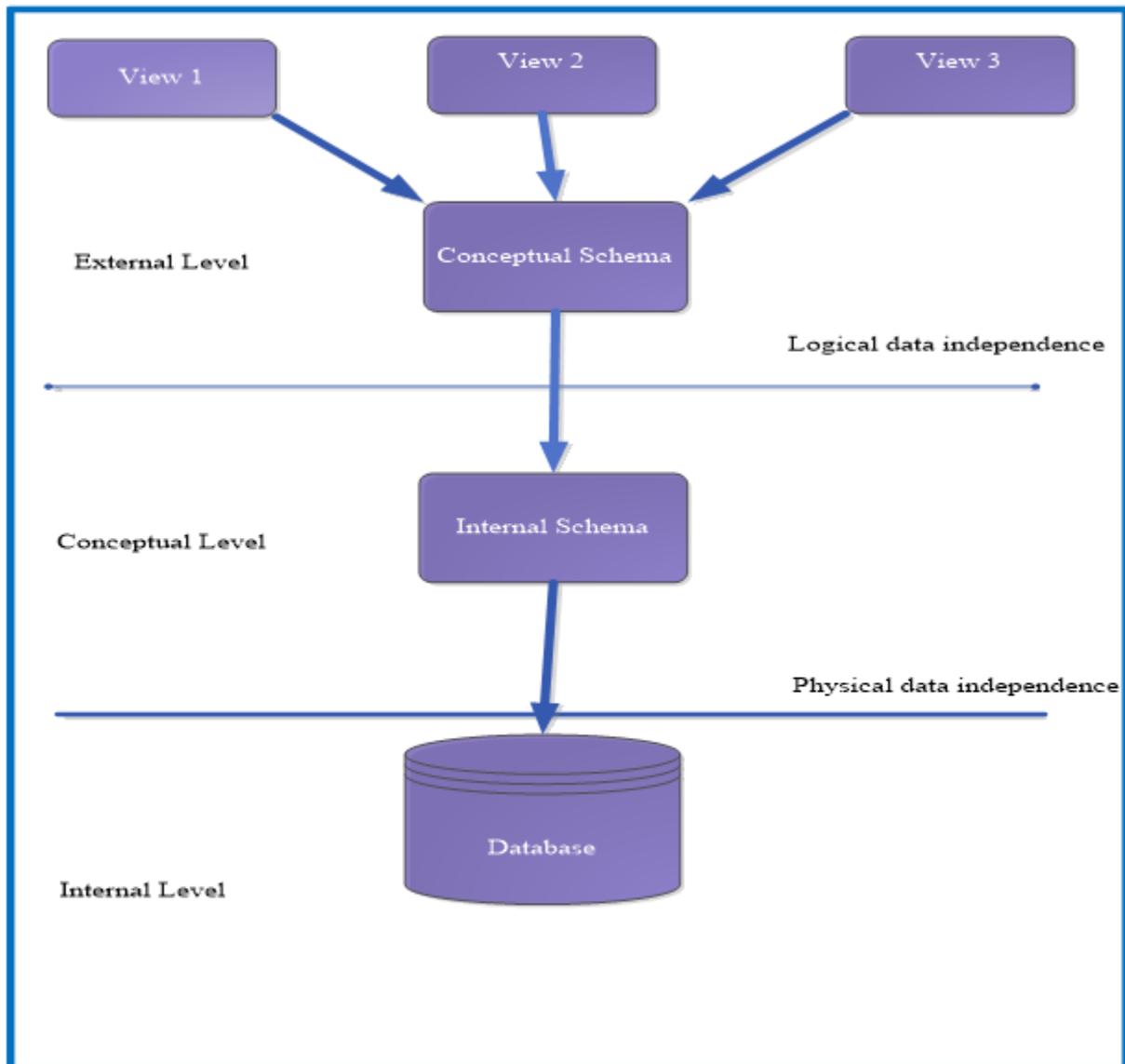


Fig 4.5 Database Structural Design

4.5.1 External Level

The external layer provides a user's view and the particular part of the database relevant to a specific user. Same data can be presented differently at this level.

4.5.2 Conceptual Level

It also called community view. This level describes what data is stored in the database and how it is interrelated. There is no specification of how data is physically stored at this level. It represents all entities, their attributes and the relationship. The database administrator of child protection case management system is the only one allowed modifying or structuring this level.

4.5.3 Internal Level

This level looks at the physical layout of the database on a computer. Encryption and data indexing is also done on this level.

4.5.4 Physical Design Level

This is the physical representation of the database and has an internal schema that describes how the data is stored in the database (Wesley, 2005).

4.5.5 Database Tables

4.5.5.1 User Field

Field Name	Data Type	Explanation
FirstName	nvarchar(50)	Username
Surname	nvarchar(50)	User surname
Password	Nvarchar	User password
UserId	Nvarchar	User National id
Email	Nvarchar	User email address
PhoneNo	int(20)	User phone
Address	nvarchar(100)	User address

Table 4.1: User Field

4.5.5.2 Case Type Table

Field Name	Data Type	Explanation
Id	Int	Case id
Case Type	nvarchar(50)	Type of cases eg rape

Table 4.2: Case Type

4.5.5.3 CaseIntake

Title	Type of data	Explanation
Province	Int	Province where the case occurred
District	Int	District where the case occurred
Ward	Int	Ward where the case occurred
Name	nvarchar(50)	Name of a client
Surname	nvarchar(50)	Surname of a client
Gender	Int	Is a client female or male
BirthRecord	nvarchar(50)	Client birth details
DOB	Int	Date of birth of a client
Address	nvarchar(50)	Physical address of a client
DateReffered	Date	Date of case reffered
EmergencyAction	Nvarchar	Emergency action eg medical emergency
Referee	nvarchar(50)	A person who brought the case
Relationship	Int	The relationship between a client and referee
RefAge	Int	Age of referee
RefAddress	nvarchar(200)	Physical address of referee
RefPhone	nvarchar(50)	Contact details of referee
CaseNumber	nvarchar(20)	Unique identify
ReasonOfRef	nvarchar(20)	Reason of a case
IsDeleted	Int	Interger

Table 4.3: CaseIntake

4.5.5.4 CaseAssessment

Title	Type of data	Explanation
ID	Int	Unique id of a client
CaseNumber	nvarchar(20)	Unique id of a case
Family	Nvarchar	Client family details
Survival	Nvarchar	Client survival details
GeneralHealth	Nvarchar	Client health status
Development	Nvarchar	Is the client is developing appropriately for his or her age.
Behavior	Nvarchar	Clients behavior eg a child who begins to be fearful.
SocialHistory	Nvarchar	Client's environment in which lives .
Educational	Nvarchar	How does the client perform in school.
ChildComments	Nvarchar	Client's comments
Conclusion	Nvarchar	Conclusion of a case
Dateassessed	Datetime	Date
Lastassessedby	Nvarchar	Assessed by officer

Table 4.4: CaseAssessment

4.5.5.5 Case Plan

Title	Type of data	Explanation
Identity	Int	Unique id
CaseNumber	nvarchar(20)	Unique identify
CasePlan	Nvarchar	Action taken
DateOfCompletion	Date	Date
CaseExplanation	Nvarchar	Details of a case
IsDeleted	Int	Interger

Table 4.5: Case Plan

4.5.5.6 Case Allocation

Field Name	Data Type	Explanation
ID	Int	Unique id
CaseNumber	nvarchar(50)	Unique identify
AllocatedTo	nvarchar(50)	Case allocated to eg institution
AllocatedBy	nvarchar(50)	Allocated by eg Case Management Officer
DateAllocated	Datetime	Date
Priority	nvarchar(50)	Case priority eg high ,low
IsDeleted	Int	Interger

Table 4.6: Case Allocation

4.5.5.7 Case Review Table

Field Name	Data Type	Explanation
ID	int	Unique id
CaseNumber	nvarchar(20)	Unique identify
InitialCategory	nvarchar	Case category
AssessedCategory	nvarchar	Assessment of case category
ReviewofCase	Nvarchar	Case review
ReviewofCarePlan	Varchar	Case review plan
ReviewofChildNeeds	nvarchar	Child needs review
Recommendation	nvarchar	Recommendation
DateofNextCR	date	Date of care review
CRApprovedBy	nvarchar	Approved by eg DSSO
DateByME	date	Date

Table 4.7: Case review field

4.5.5.8 Case Closure Table

Field Name	Data Type	Explanation
ID	Int	Unique id
CaseNumber	nvarchar(20)	Unique identify
DateCaseClosed	Date	Date
Assessment	Nvarchar	Details of assessment
Comment	Nvarchar	Case comments
Reason	Nvarchar	Case reason closure
ChildNotificationDate	Date	Date
CareNotificationDate TypeofInfo	Date nvarchar	Date
ApprovedBy	Nvarchar	Approved by eg DSSO
ChildAware	Nvarchar	Child aware eg yes or no

Table 4.8: Case closure field

4.5.5.9 Case Referral Table

Field Name	Data Type	Explanation
ID	Int	Unique id
CaseNumber	Nvarchar	Unique identify
Problem	Nvarchar	Problems eg lack of resources
ReasonOfRef	Nvarchar	Reasons of case ref eg lack of resources
ReferredBy	Int	Referred by eg DSSO
DateReferred	Date	Date
ReferredTo	Int	Referred to e
IsDeleted	Int	Interger

Table 4.9: Case referral field

4.6 Entity Relationship

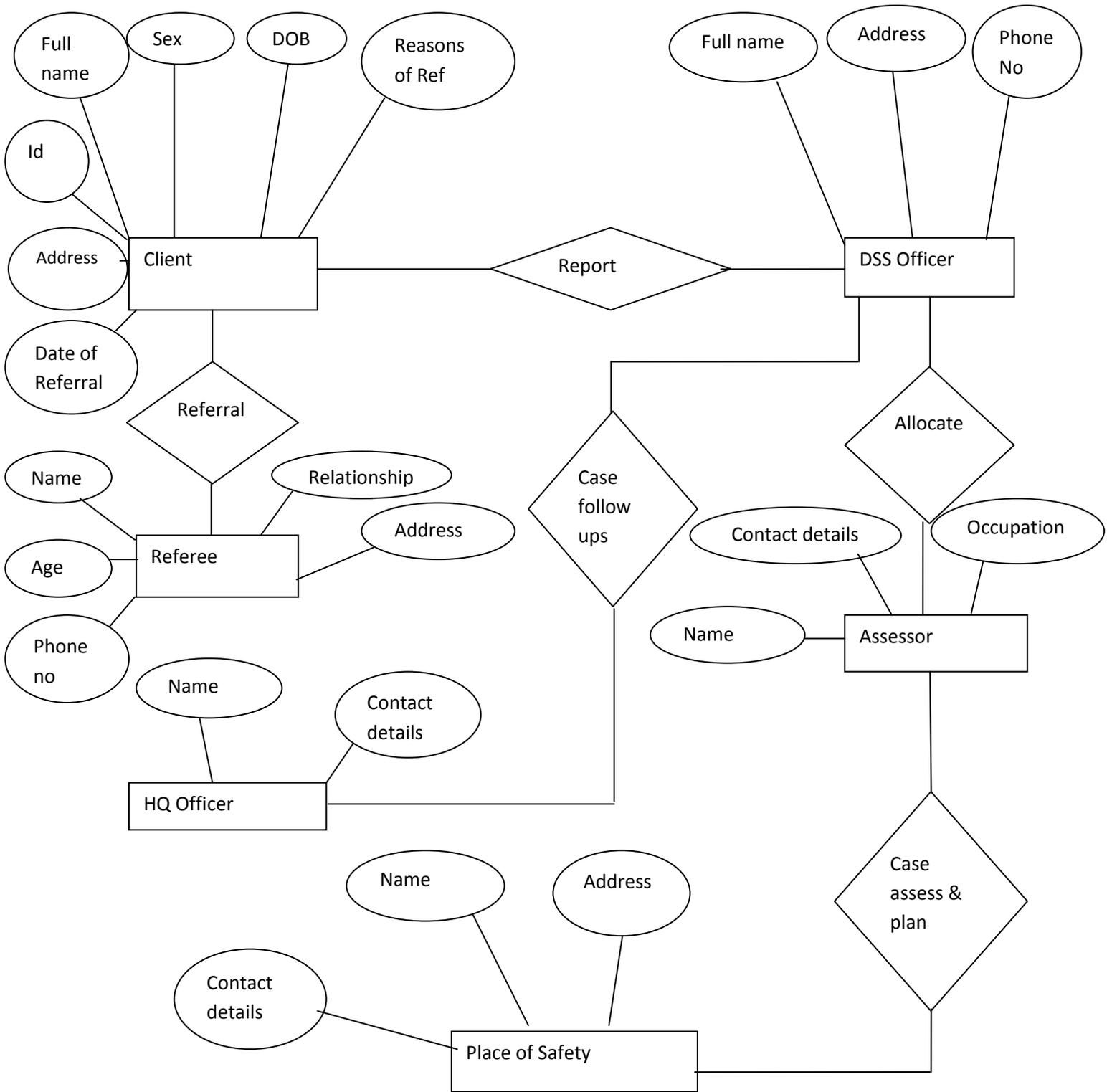


Fig 4.6.1 Entity Relationship

4.7 Interface Design

According to Morgan (2002) defines an interface as a pictorial presentation of controls and icons that are used by the users to communicate and interact with the system. Different interfaces will be shown to users. The diagram below provides a sketch of interlinkages in the system modules. It shows layouts and activities at any given phase.

4.7.1 System Layout

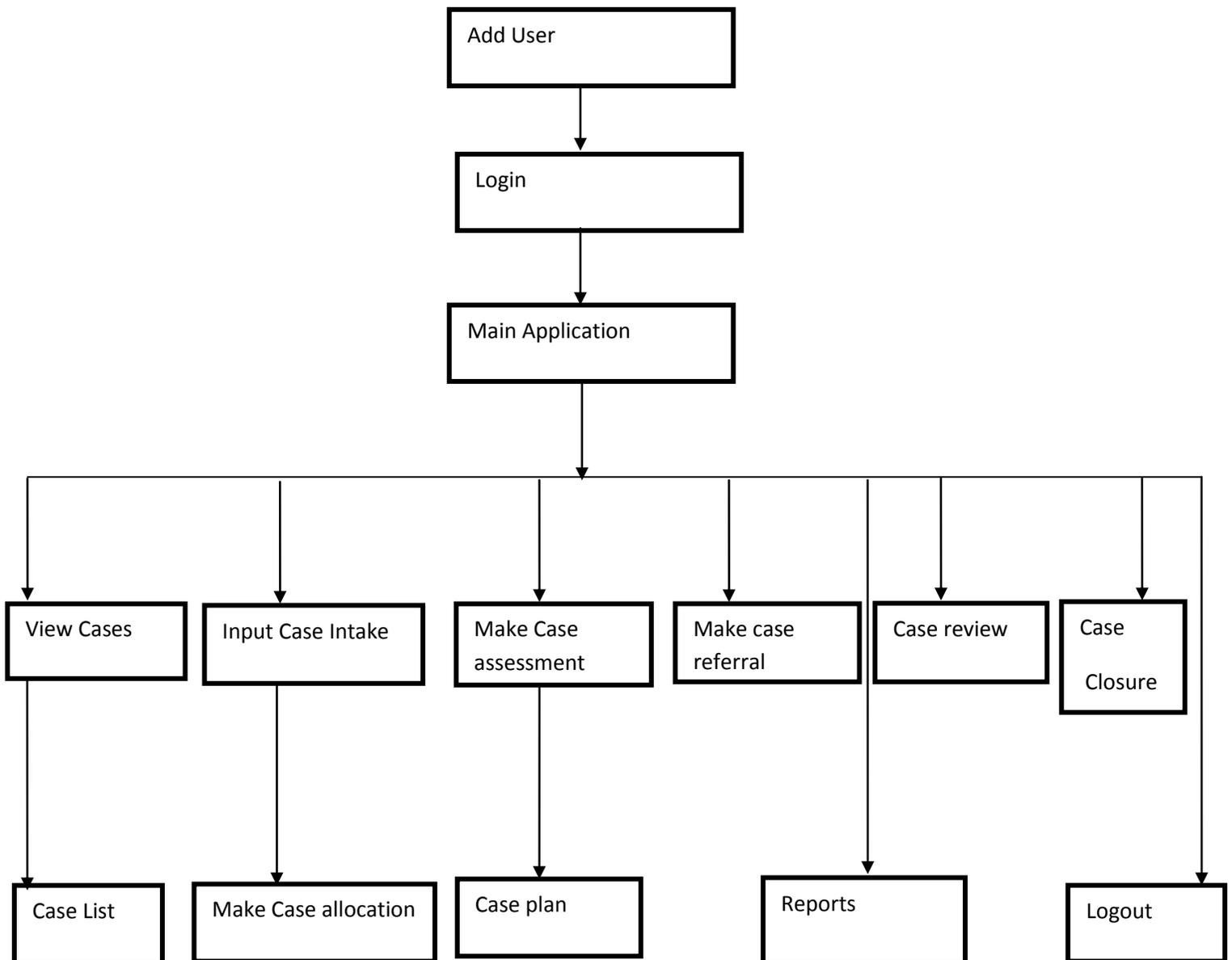
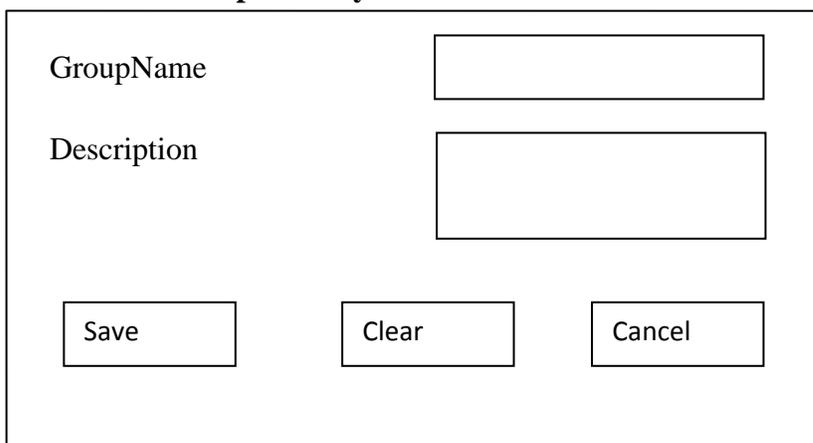


Figure 4.7 System Layout

4.7 Functional Design

Functional design phase displays the outline of the input and output activities that will be operated by users operating a window applications. Each activity operation will be displayed differently as data captured. Activities at input level will allow users to capture appropriate data. Output activities will permit the users to view their processed information for example a list of pending cases or a list of referral cases made before. The forms help the users to come up with their reports. The diagram below shows an outline of input and output designs.

4.7.1 Add User Group Activity



GroupName	<input type="text"/>	
Description	<input type="text"/>	
<input type="button" value="Save"/>	<input type="button" value="Clear"/>	<input type="button" value="Cancel"/>

Fig 4.8 Add user group activity

4.7.2 Add User Activity

FirstName	<input type="text"/>
Surname	<input type="text"/>
ID	<input type="text"/>
Email	<input type="text"/>
Phone Number	<input type="text"/>
Address	<input type="text"/>
Status	<input type="text"/>
Username	<input type="text"/>
Password	<input type="text"/>
Confirm password	<input type="text"/>

Fig 4.9 Add user activity

4.7.3 Login Activity

Enter UserName	<input type="text"/>
Enter Password	<input type="text"/>
<input type="button" value="Login"/>	<input type="button" value="Exit"/>

Fig 4.10 Login activity

4.7.4 View Case Activity

Case Number	<input type="text"/>
Name	<input type="text"/>
<input type="button" value="Load"/>	<input type="button" value="Exit"/>

Fig 4.11 view case activity

4.7.5 Input case intake activity

Province	<input type="text"/>	District	<input type="text"/>	Ward	<input type="text"/>
Child					
Name of Child	<input type="text"/>	Surname	<input type="text"/>	Sex	<input type="text"/>
DOB	<input type="text"/>	Reasons of	<input type="text"/>	Emergency	<input type="text"/>
Address	<input type="text"/>	Referral	<input type="text"/>	Action	<input type="text"/>
		Date of Ref	<input type="text"/>		
Referee					
Name	<input type="text"/>	Relationship	<input type="text"/>		
Address	<input type="text"/>	Phone Number	<input type="text"/>		
Decision	<input type="text"/>	Further Action	<input type="text"/>		
Detailed Assessment	<input type="text"/>				
<input type="button" value="Save"/>	<input type="button" value="Clear"/>	<input type="button" value="Cancel"/>			

Fig 4.12 case intake activity

4.7.6 Make case assessment

Province	<input type="text"/>	District	<input type="text"/>	CaseNumber	<input type="text"/>
Child					
Name	<input type="text"/>	DOB	<input type="text"/>	Date	<input type="text"/>
Reason	<input type="text"/>	Date of contacted	<input type="text"/>		
Assessor					
Name	<input type="text"/>	Occupation	<input type="text"/>	Contact Details	<input type="text"/>
Visits details					
Date	<input type="text"/>	Details	<input type="text"/>	Development	<input type="text"/>
			Add	Behavior	<input type="text"/>
Case Number	Effective Date	Description		Social History	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>		Educational	<input type="text"/>
				Other Organisations	<input type="text"/>
				Child's Comments	<input type="text"/>
Survival	<input type="text"/>			Conclusion	<input type="text"/>
General health	<input type="text"/>				

Fig 4.13 Case assessment

4.7.7 Make care plan

Primary care	<input type="text"/>	Explain case plan	<input type="text"/>
Secondary care	<input type="text"/>	Date of completion	<input type="text"/>
<input type="button" value="save"/>	<input type="button" value="Clear"/>	<input type="button" value="Exit"/>	

Fig 4.14 Care plan

4.7.8 Case Review

Case Number	<input type="text"/>	Name	<input type="text"/>	<input type="button" value="Load"/>
Child <input type="text"/>				
Assessed Abuse	<input type="text"/>	<input type="button" value="Review case plan"/>		
Review Child's needs	<input type="text"/>			
Recommendations	<input type="text"/>			
Next Review Date	<input type="text"/>			
<input type="button" value="Save"/>	<input type="button" value="Clear"/>	<input type="button" value="Cancel"/>		

Fig 4.15 Case review

4.7.9 Case Referral

Search Criteria <input type="text"/>				
Case Number	<input type="text"/>	Name	<input type="text"/>	<input type="button" value="Load"/>
Child <input type="text"/>				
<input type="button" value="Load Care Plan"/>				
Details of Problem	<input type="button" value="Toggle All"/>	<input type="text"/>		
Reason of Referral	<input type="text"/>			
Refer to	<input type="text"/>			
<input type="button" value="Process"/>	<input type="button" value="Clear"/>	<input type="button" value="Cancel"/>		

Fig 4.16 Case referral

4.7.10 Case Allocation

Select Search Criteria				
<input type="radio"/> Case Number	<input type="text"/>	<input type="radio"/> Name	<input type="text"/>	<input type="button" value="Load"/>
Right Click on Child				
<input type="button" value="Allocate"/>	<input type="button" value="Close case"/>	<input type="button" value="Delete"/>	<input type="button" value="Update"/>	

Fig 4.17 Case allocation

4.7.11 Case Closure

Search Criteria				
<input type="radio"/> Case Number	<input type="text"/>	<input type="radio"/> Name	<input type="text"/>	<input type="button" value="Load"/>
Child				
Assessment of current status	<input type="text"/>	Comment on Outstanding needs	<input type="text"/>	
Reason of closure	<input type="text"/>			
Date of Notification Child	<input type="text"/>	Date	<input type="text"/>	
Caregiver	<input type="text"/>	Child aware?	<input type="text"/>	
Type of info	<input type="text"/>	Approver	<input type="text"/>	
<input type="button" value="Save"/>	<input type="button" value="Clear"/>	<input type="button" value="Cancel"/>		

Fig 4.18 Case closure

4.8 Output Design

4.8.1 Produce Pending Cases Report

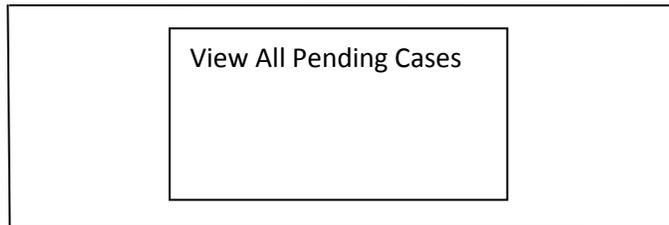


Fig 4.8.1 Produce pending cases report

4.8.2 Make Juvenile Delinquency Case Report

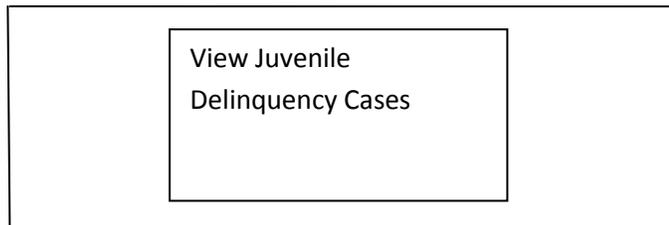


Fig 4.8.2 Make Juvenile delinquency case report

4.8.3 Extract Case Referrals Report

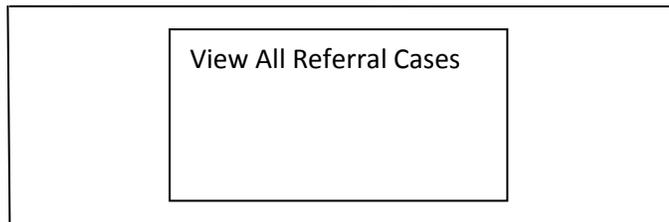


Fig 4.8.3 Extract view case referrals report

4.8.4 Generate Case Summary Report

District	Reason of Ref	Boys	Girls
Marondera	Neglect	0	4
Bindura	Abuse	1	3
Total		1	7

Fig 4.8.4 Generate view case summary report

4.9 Conclusion

This phase facilitate the system developer to design the framework of the anticipated system, how the proposed system interface, database tables as well as connectivity and interconnectedness of entities and processes should look like the anticipated system. Furthermore, the design phase will also facilitate developers to design the sequence flow of activities given the inputs, processes and outputs involved in the system. Implementation phase is the next phase whereby the actual implementation of the system is done.

CHAPTER FIVE: IMPLEMENTATION

5.1 Introduction

The implementation phase is the one at which the project starts to take shape. The system under development will be deployed in parallel to the current manual system under use in the Ministry of Public Service, Labour and Social Welfare. Any challenges in the new system will not necessarily affect workflow as the old system will still be operational. As this is the execution phase, it is important to ensure that implementation efficiency is maintained.

5.2 Coding

According to Synergy (2001) coding involves converting the programme logic what can be executed by the computer by way of instructions. This is the stage where what was indicated in the analysis phase is now brought to functionality through system programming. Different modules are interlinked together to form one integrated system. C# was used to generate codes of the system and sql was used in the creation of databases that helps the full functioning of the system.

5.2.1 Pseudo Code

An algorithm written in a naturistic way explaining the logic of system operational ability is known as a pseudo code (Synergy 2001). For the case Management MIS, the system pseudo code is as below:

5.2.1.1

Start (after login)

Enter username and password

Evaluate user details

If correct then

Else

Access denied

Try again and Re-enter details

End if

Go to **Child Protection Case Management** system main menu

End

5.2.1.2 Case Intake

Start

Select province, district and ward

Capture case details

Save

If case details are complete and record doesn't exist

Proceed

Else

Complete records

End

5.2.1.3 Case Allocation

Start

Search case

Allocate to officer

If case allocated

Prompt reallocation

Else

Proceed

End

5.2.1.4 Case Assessment

Start

Search case

Enter assessment details

If assessment complete

Proceed

Else

Reassess

End

5.2.1.5 Case Plan

Start

Search case

Select case plan

If case is complicated

Add case stakeholder

Else

Proceed

End

5.2.1.6 Case Referral

Start

Search case

Select the case referral details

Save

End

5.2.1.7 Case Review

Start

Search case

Review case plan

If case needs are not met

Meet the needs

Else

Case closed

End

5.2.1.8 Case Closure

Start

Search case

If all case are met

Close case

Else

Close not closed

End

5.2.1.9 Pseudo code for reports on case

Start

Select report

View

Closed

End

5.3 Testing

Post code development, it is important to test to ensure that all requirements are being met and addressing user specifications. This is important particularly testing the unit, testing for integration, acceptability and entire testing of the system.

The stages of testing process

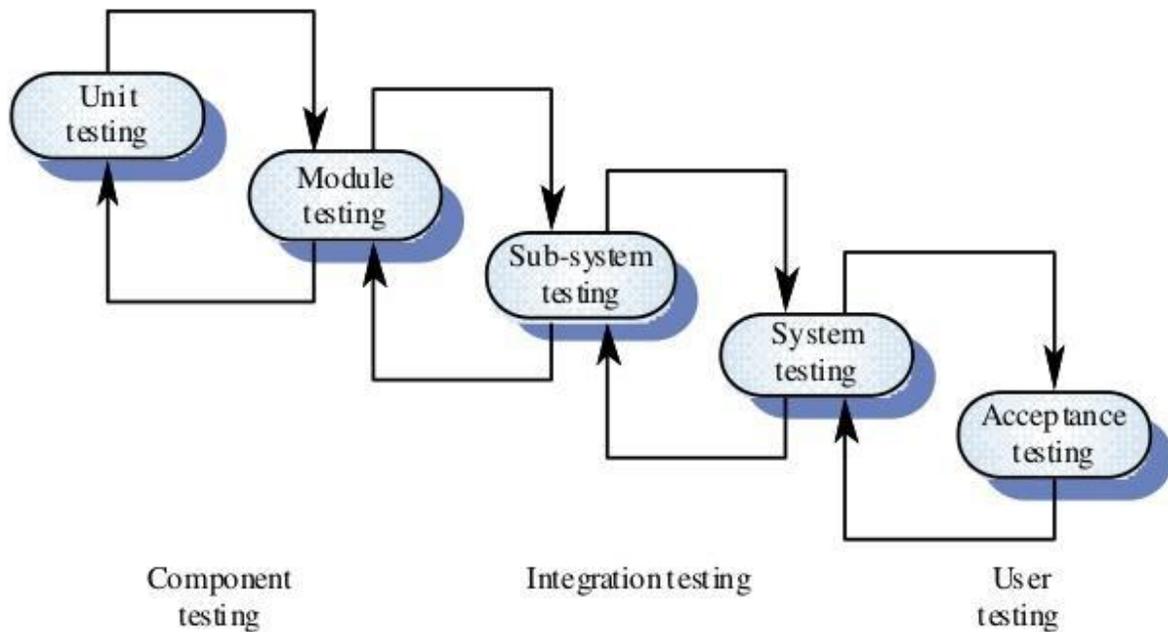


Fig 5.0 Testing Modules

Source:

<http://www.slideshare.net/raheelaslam9237/software-engineering-process>

5.3.1 Unit Testing

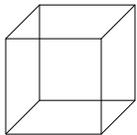
The breaking down of small system units for individual and independent analysis during the software development process is known as unit testing. The process involves scrutinizing each unit to determine whether it executes each assigned task. There are two instances for doing this:

Fig 5.3.1.1 performing a black box test



This is defined by Godfrey (2001) as an overall system functionality examination that is performed without necessarily going into its internal set up. This can be performed at any stage of the development of the system. It focuses on the inputs and outputs of the unit only. This was fully performed for the case management system under development.

Fig 5.3.1.2 performing a white Box Test



The white box test is a method of internal system checking without necessarily testing functionality. Unlike the black box, the white box focuses on how the unit works from its internal structure, noting specific component errors that may be difficult to identify when treating the whole unit as a black box.

5.3.2 Testing the sub-system

This treats the whole unit as whole and performs an overall test of sub-systems. Any issues of non-compatibility from programme to programme are identified and for the case management system, the test proved that that the system was functioning well.

5.3.3 Module Testing

Dependent components are combined and tested together under this type of testing. This was done for the case management system. It was also possible to test a single component independently. Objectives of the system were well produced leading to the conclusion that the system had proper functionality.

5.3.4 System Testing

This involved the testing of the entire system looking at possible scenarios. All the different modules highlighted were treated as one system which was then tested for any errors that could affect overall system performance.

5.3.5 Defect Testing

Defect testing is an error or a bug in the application which is created. This is more like virus or bacteria creation in medicine to troubleshoot where there could be problems with the intention of creating the right solution to the problem before full and final deployment of the system. This was done and all potential errors were addressed.

5.3.6 Acceptance Testing

A sample data test was done before full system deployment. It was noted the system was meeting user requirements and sampled users were very much in agreement with the design. At least 5 users were involved in the pre-test and this good representation of all the users in the pilot districts. An acceptance test generally tests if the system is proficient enough to meet the requirements and expectations of users.

5.3.6.1 Alpha Testing

Alpha acceptance testing was done in the office of the Director of Child Welfare and Protection Services with the participation of at least 5 officers who are the end users. This was done to utilize alpha to point to any problems in the system and pick any errors for correction.

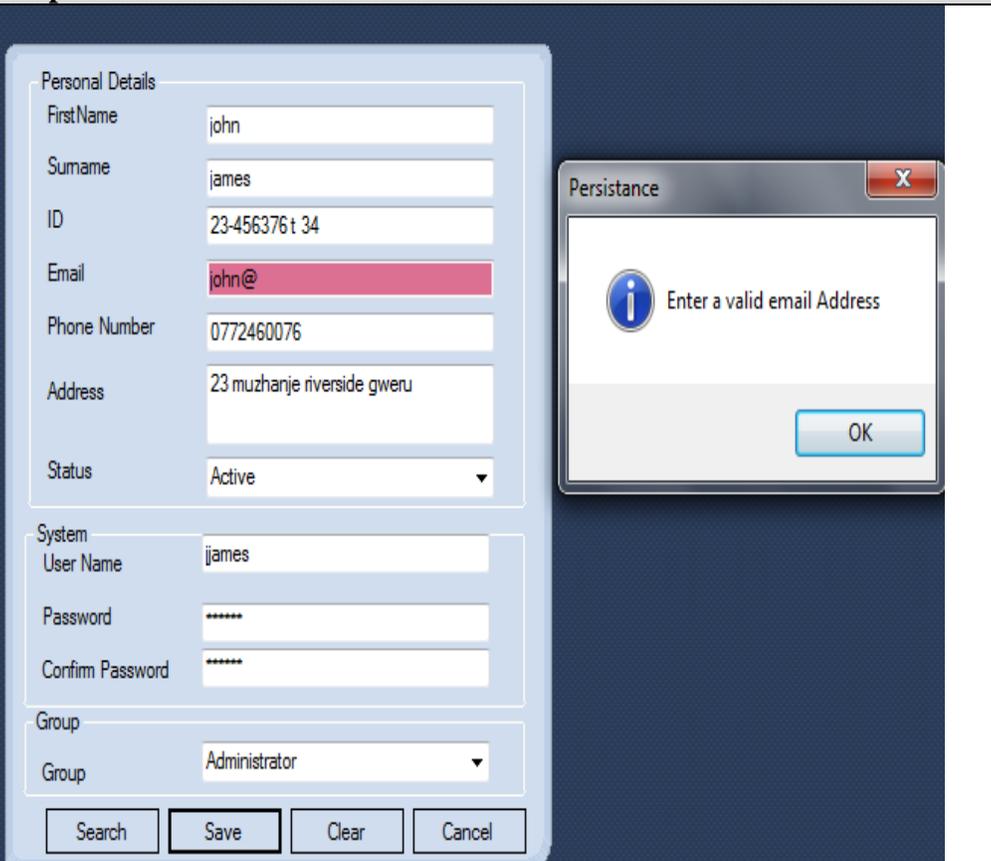
5.3.6.2 Beta Testing

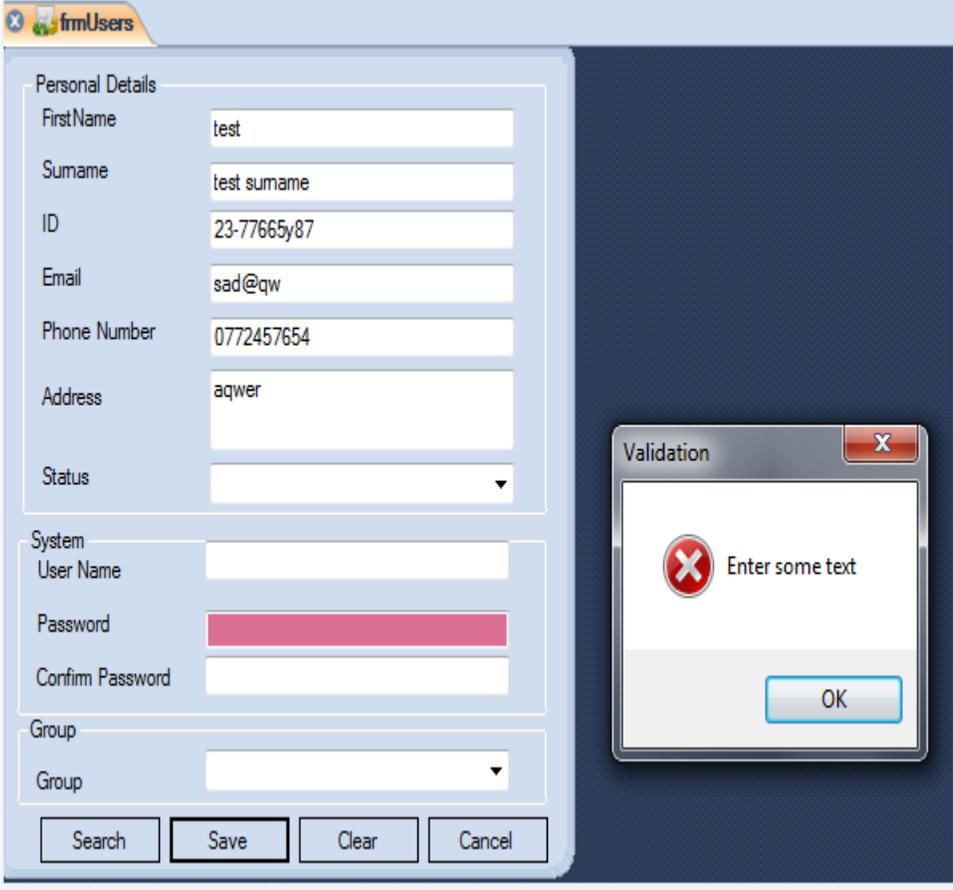
Case management Officers were asked to enter data from their case files to identify any weakness in system comprehensiveness. Some errors were noted and process was repeated two more times until the developer was satisfied that the system now had very minimal errors and the Ministry management was also now confident that the system would perform efficiently.

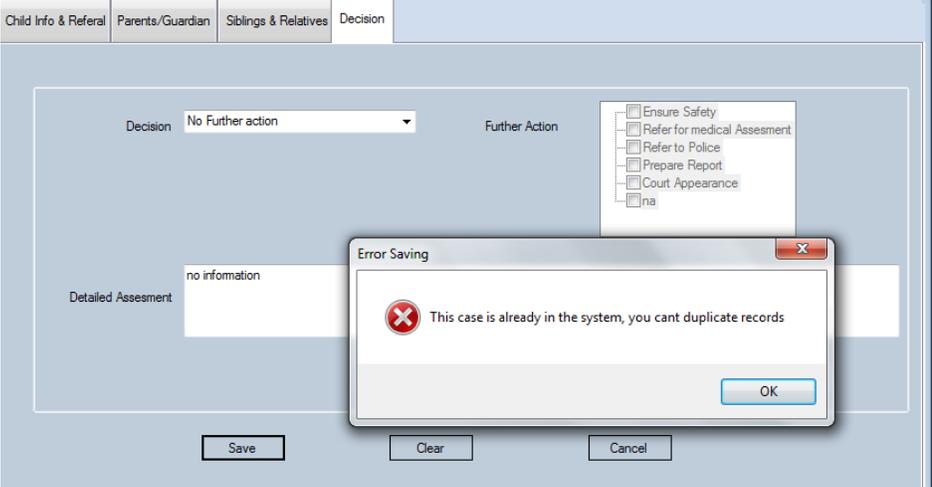
5.3.7 Validation

Validation was done with the participation of key system users in the Ministry. Validation is defined as the process of evaluating the system to determine if the right system has been developed. The exercise involved an analysis of whether specific system objectives were achieved. This was done through checking whether the right fields were accepting or rejecting correct or wrong data entry respectively. For instance numeric values were entered in the alpha text field and the system reported an error. The opposite was also done to ensure that alpha text fields did not accept numeric values. The tests led to the conclusion that the system is a true reflection of the national case management system. The following tests were performed to validate:

Validation Test Cases

Case	Description	Output
1	Invalid email address	 The screenshot shows a user registration form with the following fields: Personal Details (First Name: john, Surname: james, ID: 23-456376 t 34, Email: john@, Phone Number: 0772460076, Address: 23 muzhanje riverside gweru, Status: Active), System (User Name: james, Password: *****, Confirm Password: *****), and Group (Administrator). The Email field is highlighted in red, indicating an error. An error message dialog box titled "Persistence" is displayed, containing the text "Enter a valid email Address" and an "OK" button.

2	Missing text	 <p>The screenshot shows a web form titled 'frmUsers' with several input fields. The 'Personal Details' section includes fields for First Name (test), Surname (test surname), ID (23-77665y87), Email (sad@qw), Phone Number (0772457654), and Address (aqwer). The 'System' section includes User Name, Password, and Confirm Password fields. The 'Group' section includes a dropdown menu. A 'Validation' dialog box is open, displaying a red 'X' icon and the message 'Enter some text' with an 'OK' button.</p>
---	--------------	---

3	Case is already exist	 <p>The screenshot shows a web form with tabs for 'Child Info & Referral', 'Parents/Guardian', 'Siblings & Relatives', and 'Decision'. The 'Decision' tab is active, showing a dropdown menu for 'Decision' set to 'No Further action' and a 'Further Action' section with a list of checkboxes: 'Ensure Safety', 'Refer for medical Assessment', 'Refer to Police', 'Prepare Report', 'Court Appearance', and 'na'. An 'Error Saving' dialog box is open, displaying a red 'X' icon and the message 'This case is already in the system, you cant duplicate records' with an 'OK' button.</p>
---	-----------------------	--

4	The date allocated shouldn't be greater than today's date	
---	---	--

Table 5.1 Validation test cases

5.3.8 Recovery Testing

A recovery test was carried out with dummy data from one of the districts. This is a test that is done to determine if a designed application is able to survive crashes, and other issues like hardware failures. It also helped to design proper data recovery systems.

5.3.9 Regression Testing

Although the old system was manual, a regression test was still done to determine if the old system could still work alongside the new electronic system. Regression tests also help to detect if the any change in one part of the system causes changes in other parts. The test was done and system compatibility was found to be sound.

5.3.10 Security Testing

Security testing was undertaken to restrict unauthorised access to application, to assess the risks of anyone entering the system without privilege authority. Security is determined in consideration of physical and software measures. Passwords, security user levels and burglar bars are some of the examples of security measures.

5.3.11 Performance Testing

System performance was tested to for the aim of determining the efficacy of the system.

5.3.12 Stress Testing

Stress testing also known as torture testing. This test was done to determine the robustness of child protection case management application. The reason being that there is high demand on the application operating system.

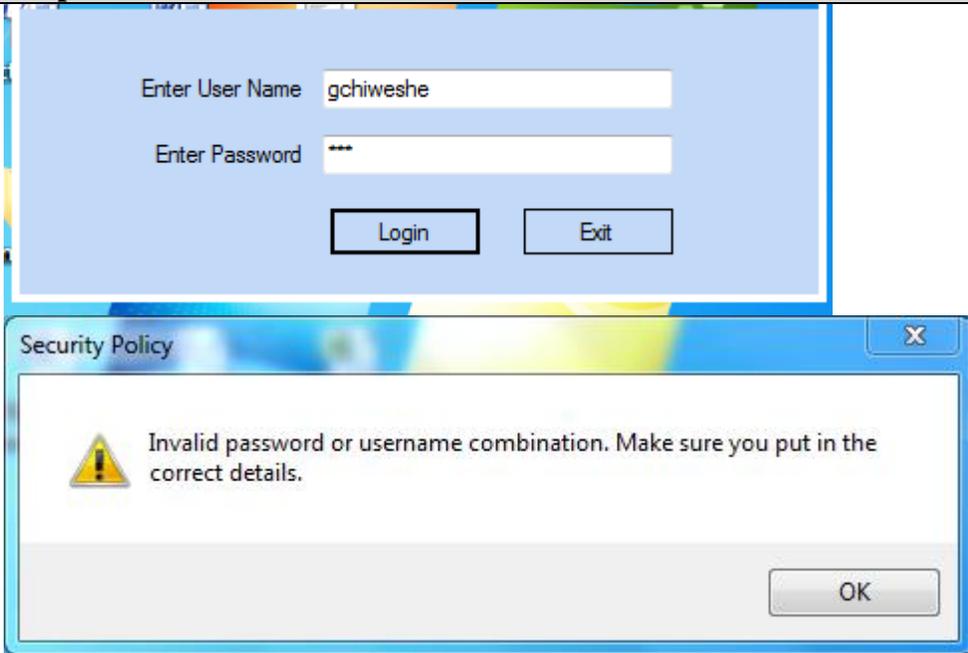
5.3.13 Usability Testing

This involved hardening of the system applications in the software to outfit any desired changes.

5.3.14 Verification

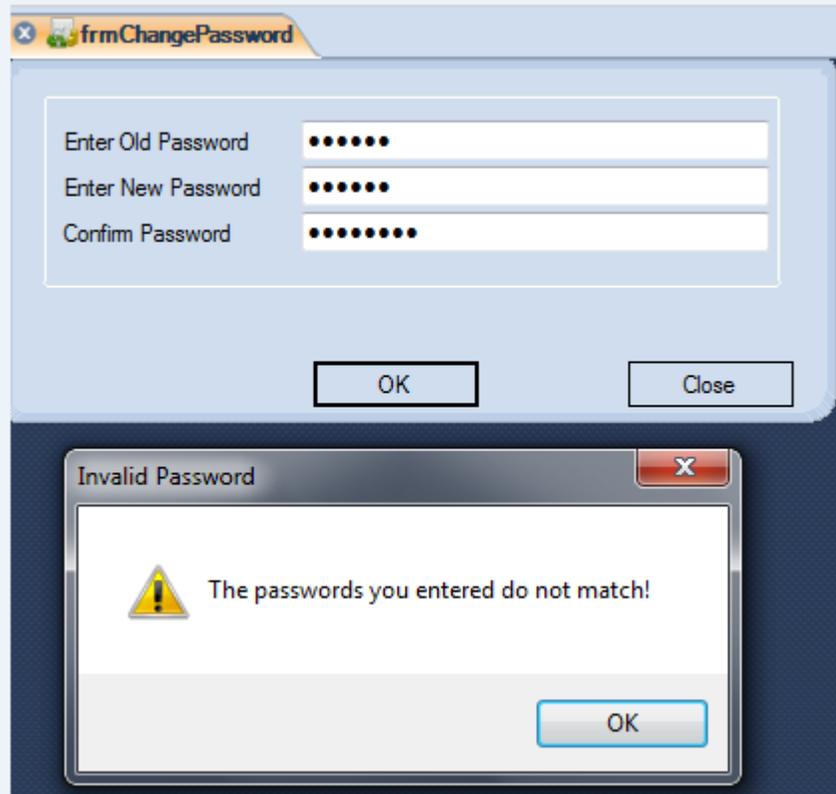
This is a system check on conformity of the system to specifications aligned during design. The analyst verified system data with what the case management officers are used to.

Verification Test Cases

Case	Description	Output
1	Invalid login	 The screenshot shows a login interface with two input fields: 'Enter User Name' containing 'gchiweshe' and 'Enter Password' containing three asterisks. Below the fields are 'Login' and 'Exit' buttons. An error dialog box titled 'Security Policy' is overlaid, featuring a yellow warning triangle icon and the text: 'Invalid password or username combination. Make sure you put in the correct details.' with an 'OK' button at the bottom right.

2

Change password



3	Assign user permissions	
4	Password Expired	

Table 5.2 Verification test cases

5.4 Installation

The system was installed in the district offices of the MoPSLSW using the parallel conversion method. This allowed for simultaneous operation of the new and old system. This will continue for a period of six month, after which a review will be undertaken to determine if the old system could be phased out. This was to ensure data preservation. The factors documented below were considered.

5.4.1 Steps for Application Software Installation

- Install microsoft.net 4.5 framework
- Install crystal reports
- Install system to path. (It is recommended that the system be installed to the program files folder.)
- Verify that the system is installed properly .i.e. all folders are installed.

		module		CMO
Day 4:22/10/2015				
8am - 12:00pm 2pm – 4:00pm	Mandel Training centre	<ul style="list-style-type: none"> • Case review module • Case closure module 	MIS Assistant	Program Officer-HQ PSSO DSSO
Day 5 :23/10/2015				
8am – 12:30pm 2pm – 3pm 3pm – 4pm	Mandel Training centre	<ul style="list-style-type: none"> • Reports Module(Pending cases, Juvenile delinquency, Case referral, Child discharge and Case summary) • Recap • Question and Answer session 	System Administrator	PSSO DSSO DSSA CMO Program Officer-HQ

Table 5.2 Training schedule

5.4.2 System Level

All DSSOs and case management officers who use the system were trained.

5.4.3 Module Level

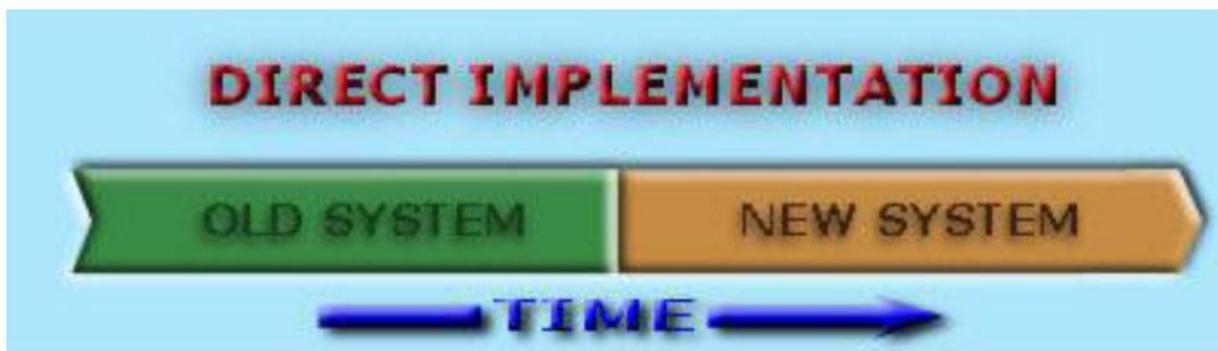
A module by module approach was also used so that Officers would develop stronger familiarity with all the modules in the system. Each module was fully explained before moving to the next.

5.4.4 Conversion

Conversion, According to Cotterell and Hughes (2004) is the stage at which the new system is used to completely replace the old system. This will only be done after regression testing stage is completed. The following will be considered when the stage is reached:

5.4.4.1 Direct Conversion

Direct Conversion denotes the time when the old system is fully replaced by the new system. This is a changeover that is usually done during the time when there is less demand on the use of the system eg during a public holiday to ensure that work is not adversely affected.



Source:

<http://web1.muirfield-h.schools.nsw.edu.au/mahara/view/view.php?id=7048>

Fig 5.3 Direct implementation

5.4.4.2 Pilot Conversion

Pilot conversion relates to gradual and phased implementation of the new system on a small scale basis until full functionality is reached. This will be done during regression testing over the next 6 months until the new system is fully deployed.



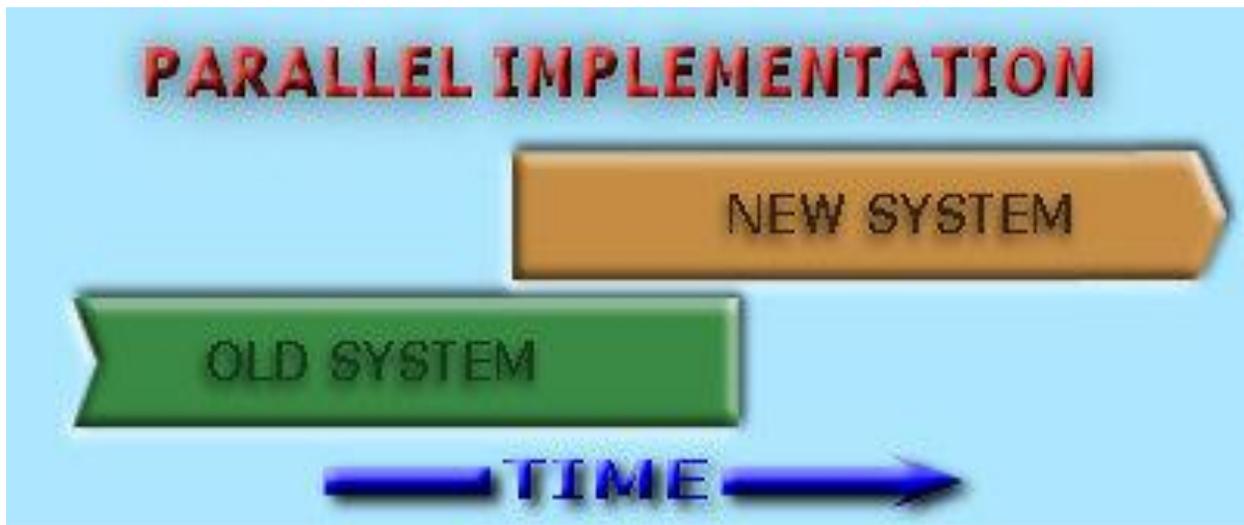
Source:

<http://web1.muirfield-h.schools.nsw.edu.au/mahara/view/view.php?id=7048>

Fig 5.4 Pilot Implementation

5.4.4.3 Parallel Conversion

This is the conversion model selected for the case management system where the new system will run parallel to the old system. Once the new system starts to run efficiently, the old system will be completely phased out.



Source:

<http://web1.muirfield-h.schools.nsw.edu.au/mahara/view/view.php?id=7048>

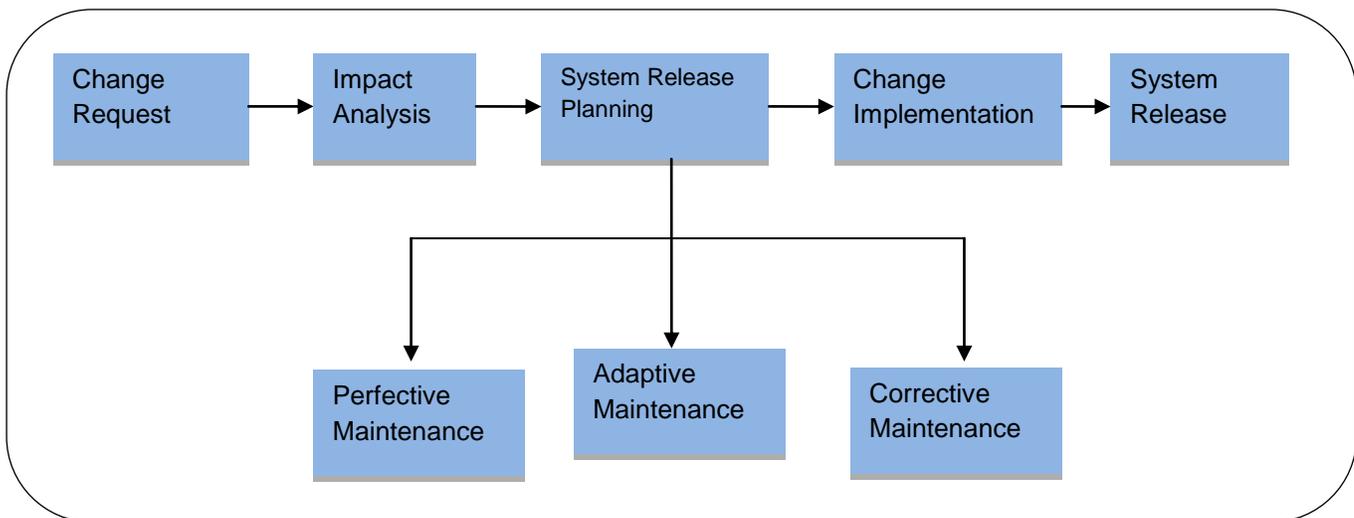
Fig 5.5 Parallel conversion

5.4.4.4 Justification for use of parallel system

- Allows continuity as any problems in the new system will be covered up for through resorting to the old system
- It allows enough transition time for users, allowing them to get used to the new system without slowing down work while getting used to the new system.
- It makes a review of the new system easier as the two systems can be compared side by side
- Reduces the loss of data during implementation

5.5 Maintenance

Ongoing system maintenance will be undertaken to ensure that the system maintains a high level of functionality in line with the ever changing business environment. The system should meet its objectives and system maintenance is a key activity to achieve this. The developer will together with the users update the system regularly to ensure that it response to the dynamism of the ever changing external environment. Below is a diagrammatic representation of system maintenance process.



Source: Erdil (2003)

Fig 5.5 Maintenance Process

5.5.1 Corrective maintenance

This will be an ongoing maintenance that takes cognizance of the previous corrections at design and implementation stages. Errors established by the District Officers and how these were rectified will be taken as the starting point to implement any new changes.

5.5.2 Perfective maintenance

According to Kendal (2005) this type of maintenance is undertaken to modify software products post-delivery and initial application to adapt it and boost performance. It more focuses on improving the child protection application to be assured of its functionality.

5.5.3 Adaptive Maintenance

This will be done throughout the implementation of the system to allow constant updating of the system to meet the requirements of the case management programme and the officers who operate the system. Since the developer is employed by the Ministry, she will periodically update and upgrade the system to meet the programme requirements.

5.6 System Security

According to Ashland (2009) the below are the reasons for protecting a computer facility physically.

- Avoiding physical theft of computer hardware.
- Avoidance of loose of data or damage of valuable data.
- Prevention of disruption of facilities

The protection of the system physical is a measure which is imminent to make sure that physical disruption of the inputs processing and the output of the computer system is protected, and moreover to make sure that risks are minimised in the operational environment of the system. Some of the physical risks which affect the computer environment include affection by natural disasters and fire, theft and viruses as well as Trojans only to mention but a few. Ashland (2009) went on to explain some of the measures which can be practised to minimise the common risk as follows.

a) Physical security

Physical security can be done by practising the following physical procedures:

- **Physical Locks:** Some security company should be employed to guard the server room at all time of the day. Need for burglar bars to enclose the laboratories and all areas with valuable computer asserts
- **Login Book:** There is need for a login book for people who get in the server room to keep records and accountability for personnel who get in the server room.
- **Fire Control Equipment**

There is need for a fire control equipment to make sure there is no fire risks such equipment includes fire detector and alarm to make people aware of the fire. Moreover fire extinguishers must be planted at strategic places to guard against fire cases

b) Software based security

There is need of software based security mechanisms to compliment physical security of a computer system.

- **The use of username and passwords:** To access the database there must be authenticity to a few number of people obliged to do that, however when one enters the system there must be access levels put in place to limit user access. All users of Child Protection Case Management System are going to be assigned user profiles that assign them usernames and passwords.
- **Audit Trials:** Audit trials must also be put in place to track about people who manipulate or alter the data; this can be done using the audit stamps which can be attached to the login operations.

5.7 : System Backup

The system application must have its own independent backup for when there is need to reinstall the system. The system can thus be reinstalled in cases that the system is not function due to any reason that may cause system function not to be as expected.

- **Standby Sever:** The system will be backed up on a daily basis and update the standby server. A network will be used in the process of updating the standby server. In Case of the system crushing the system can be reinstalled from the stand-by server.
- **Backup Tapes:** At the end of every working day the system will be backed on backup tapes that will be stored **Off-site**.
- **Print Outs:** The system will produce reports and these will be used in the reconciliation of the ministry and queries from clients respectively for better planning and control of the production manager

5.8 Recommendations

The system should have system backup in the case of anything that can happen whilst the system is being used. If data is to be lost or corrupted if the system have backup, the data can be recovered easily by reinstalling the system or prints unlike when the old system did not have system security. Moreover, from what has been gathered throughout the Life cycle of the project the system developer also recommends that the system be installed and begin to be used with immediate effect, I also strongly recommended that users should stick to the basics learnt during the training, and if any problems arise they should refer to the User Manual or seek assistance from the Systems Administrator. Users should responsibly use the system that is they should not disclose their password to any unauthorized personnel or let anyone temper around with the system. The system developer also recommends that the firm should be consistent in maintaining the system. This should not be compromised, as it is an integral process as far as the performance of the system is concerned. Maintenance should be included in the core activities of the ministry and this should be done over a stipulated period of time.

5.9 Conclusion

After all the stages were carefully executed, the developer comes to the conclusion that the system was well designed. All approaches were done to specifications and the fact that users of the system agreed that the system fully meets their expectations bear testimony to this. The user manual clearly documents all stages and is attached as one of the annexes.

References

Bentley and Whitten (2007), Operations research, McGraw, New York

Child line Zimbabwe, CCOPE (2010).

Collins (2003) **Practical Guide for Undergraduate and Postgraduate Students**, 2nd Ed, p 36

DCWPS (2014) Journal

Esterbrook and Nuseibeh(2000) ,IT solution series: Humanizing information technology :Advice from experts, IGI Global

Godfrey, A (2001), system design, Pearson education, United Kingdom

Government of Zimbabwe (2010), Millennium Development Goals Report

Griffith (2005) Systems Analysis and Design for beginners, 2nd Ed

Kendal, E (2002), System analysis and design, Prentice Hall, India

Lee, A.S., 'A Scientific Methodology for MIS Case Studies', *MIS Quarterly*, Vol 13, No. 1, Mar 1989, pp 33-50.

Lewis(2006) Fundamentals of Project Management 3rd Ed, Worksmart Publishing company

Marshall & Rossman, Methods for Business Research, (New York, HarperCollins, 1995), p 38.

Mochal, A.(2004). **The survey handbook**. Thousand Oaks, Sage

Muwoni, L, et al (2014), **assessing the benefits of unified transfers to multiple categories of individuals through targeting the household**: Zimbabwe's Harmonized Social Cash Transfers, Harare, UNICEF internal publications

Nuseibeh and Easterbrook (2000), Principles of Database, prentice Hall, United Kingdom

Pressman, R. (1994) Software Engineering, Prentice Hall, USA

Presto(2002) Management Information Systems.

Setende(2005) Software Engineering ,Prentice Hall ,USA

Synergy(2001), Business Management, McGraw Hill, India

Whitten (2003), System Analysis, Prentice Hall, USA

Wyse (2014) Management information crisis. Harvard Business Review,p115

Victim Friendly Unit Police Reports (2008 – 2010)

Yin (1994), Case Study Research, Design and Methods, Beverley Hills, CA,Sage

Zimbabwe National HIV and AIDS Estimates, 2009

<http://www.lifecyclestep.com/open/410.0ANALYSISPHASE.htm> Accessed 18 August 2015

Appendix A

User Manual

Introduction

In order for the users to be educated on how to use the system, a user manual has been developed so that they can refer when they are using the system.

In order for the system to run the following software should be installed

- Crystal report
- Microsoft framework 4.5
- Visual studio 2012

Then the executable file of the system can be installed and run.

System Home Page

After a successful login, the system home page appears as below. The system has buttons that lead may want to do for example to input child protection cases.

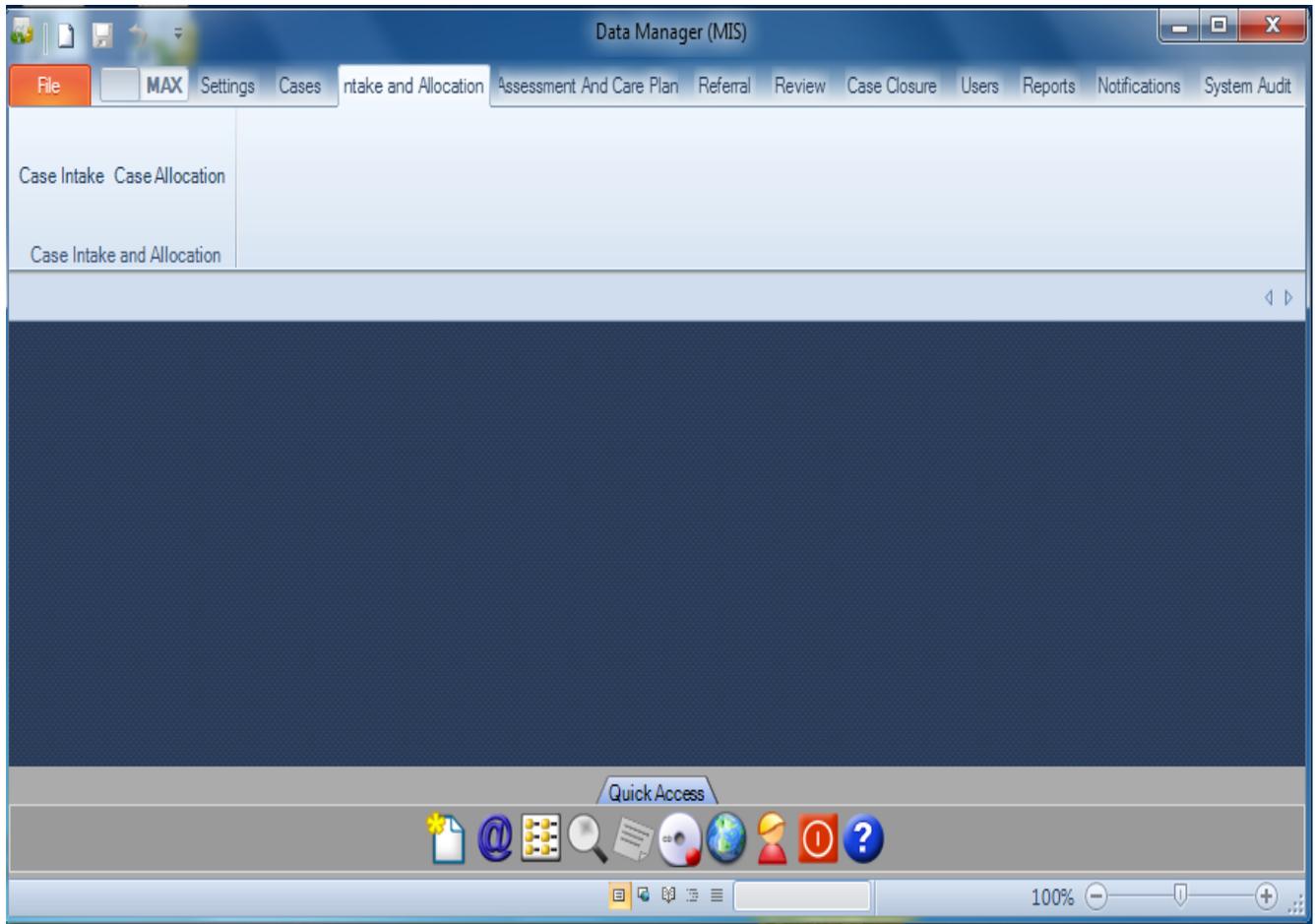


Fig A1: System home page

User group form

This screen is used by MIS Specialist or MIS Officer to enter user functional post and its description. The functional posts that are added are the ones for the users that will be using the system.

You have to Click the save button to save the changes and cancel to exit the form.

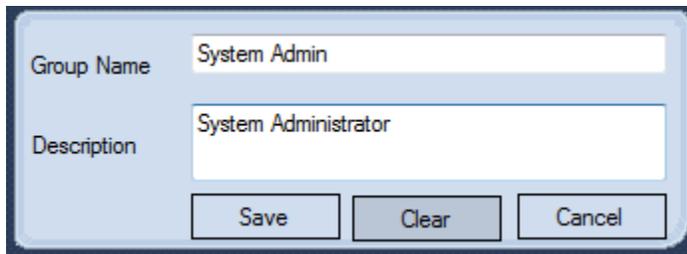
A screenshot of a user group form. The form has a light blue background and a dark blue border. It contains two text input fields: 'Group Name' with the value 'System Admin' and 'Description' with the value 'System Administrator'. Below the fields are three buttons: 'Save', 'Clear', and 'Cancel'.

Fig A2: User group form

Add user form

The fig below shows a platform to add a new user to the system.

Register all required user details, user should choose *active* status option.

Create a username and password. You should remember these credentials as you will use them each time you want to access the system

The user should input a strong password with a minimum of six characters a mixture of numbers and alpha numeric values.

Then choose the group you are working in for example Accountants or Manager

Click the save button for changes to be effective in the system

Click cancel to exit the form.

Personal Details	
First Name	Gamet
Surname	Chiweshe
ID	63-14363763 L17
Email	gchiweshe@zimnapove.co.zw
Phone Number	0776 536 636
Address	compensation hse
Status	Active

System	
User Name	gchiweshe
Password	*****
Confirm Password	*****

Group	
Group	Manager

Search Save Clear Cancel

Fig A3: Add user into the system

User Rights Form

User rights are specific access and ability permissions that can be assigned to customizable user groups.

Groups can then be assigned to (or removed from) users through the special user rights.

MIS specialist (administrator) is responsible for allocating user rights depending on the specific group category. After allocating rights, the administrator clicks the apply button and save.

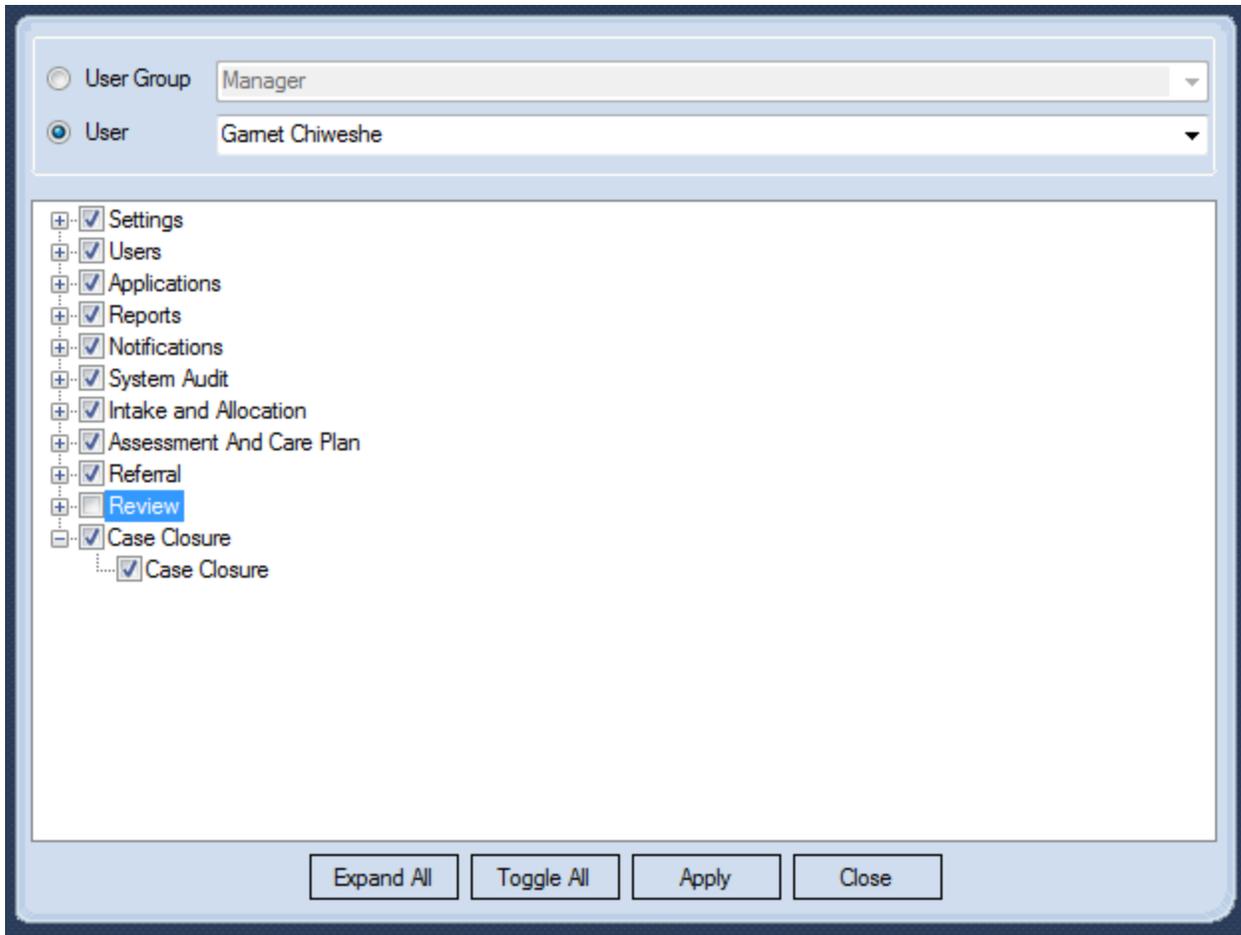


Fig A4: User right form

Login Form

This form shows how to login as a client or administrator. One has to select where appropriate.

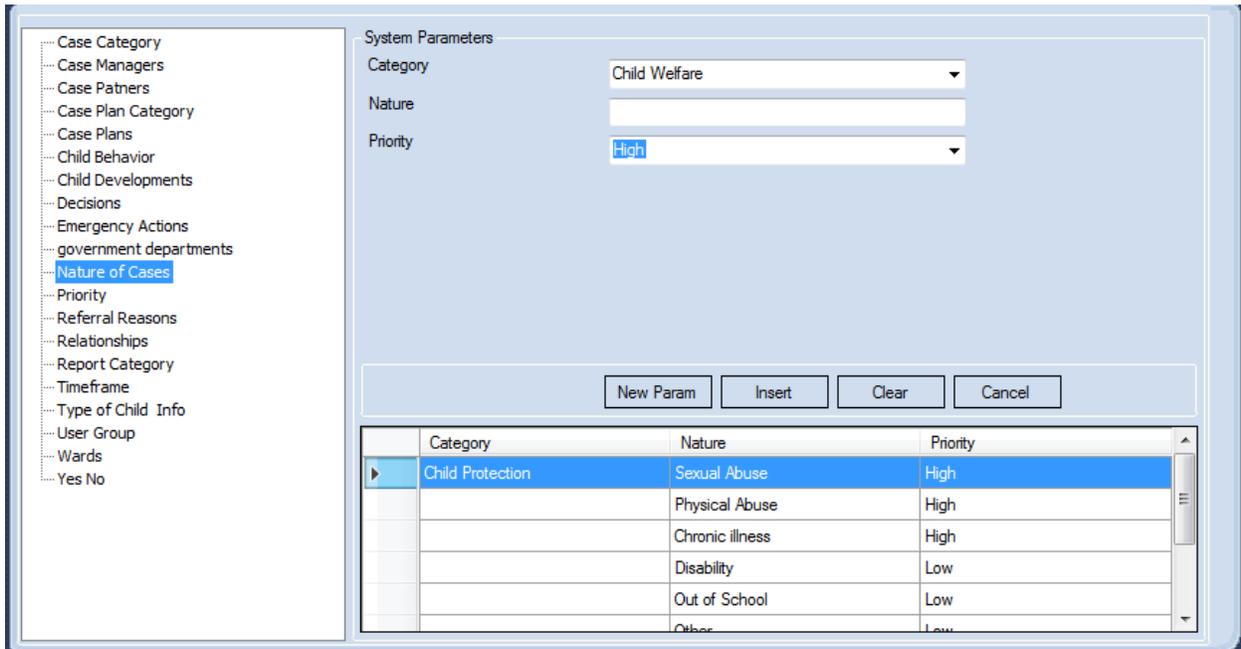
Only registered individuals with valid credentials are able to login.



Fig A5: Login form

System Parameter Form

This screen is used to capture static data, mostly for fields that remain constant over time. The example below shows case categorization and ranking in terms of priority.



Category	Nature	Priority
Child Protection	Sexual Abuse	High
	Physical Abuse	High
	Chronic illness	High
	Disability	Low
	Out of School	Low
	Other	Low

Fig A6: System Parameter Form

Case Intake Form

This screen is displayed in order establishes basic information about the child and the reason why the child needs assistance

The intake must be completed within 48 hours in order to determine what happens next.

After Inputting all the required information, a decision must be undertaken either:

- Emergency action is needed to protect the child
- There is no need for emergency action, but a more detailed assessment is required to determine a child's needs.
- No further action is needed

Then click save

The case number will automatically generate using district code, current year and case number for example ED/15/12

Child Info & Referral	Parents/Guardian	Siblings & Relatives	Decision
Province	Mash East	District	Marondera
		Ward	1
<p>Child</p> <p>Name of Child: Liberty Surname: Svosve Sex: Male</p> <p>DOB: <input type="radio"/> Documented <input checked="" type="radio"/> Estimated Reasons Of Referral: Sexual Abuse</p> <p>DOB: 11/ 9/2006</p> <p>Address: compensation house</p> <p>Date of Ref: 1/ 1/2007</p> <p>Emergency Action: <input type="checkbox"/> Ensure Safety, <input checked="" type="checkbox"/> Refer for medical Ass, <input type="checkbox"/> Refer to Police, <input type="checkbox"/> Prepare Report, <input type="checkbox"/> Court Appearance, <input type="checkbox"/> na</p>			
<p>Referee</p> <p>Name: Ula Relationship: Mother Age: 25</p> <p>Address: 21 Ruwa Phone Number: 0776 536 782 Case#:</p>			

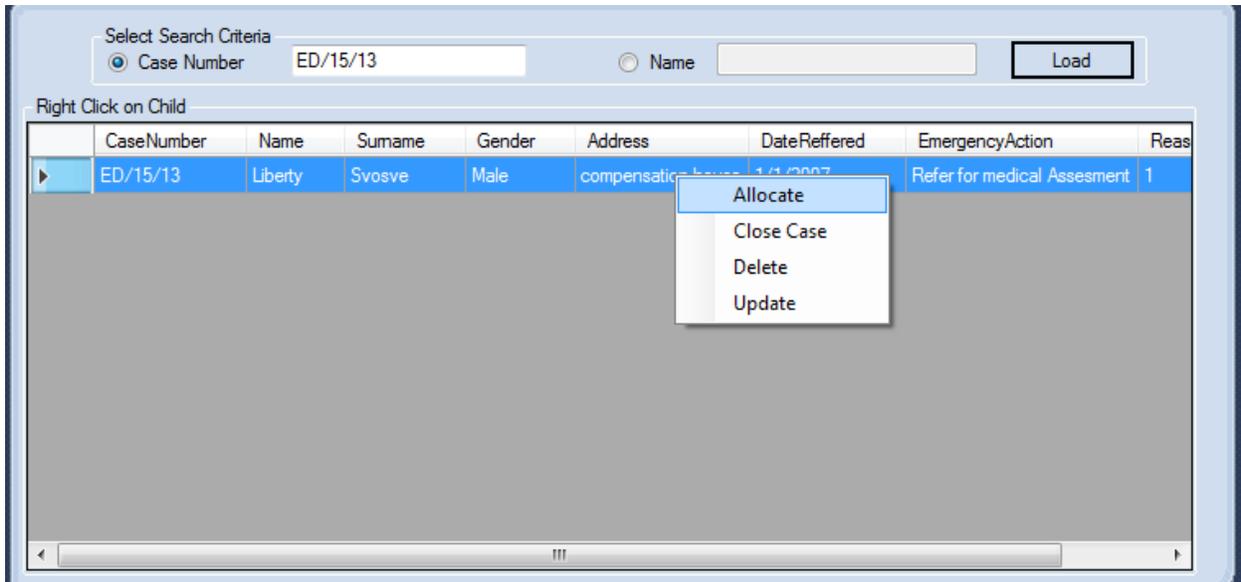
Quick Access

Child Info & Referral	Parents/Guardian	Siblings & Relatives	Decision
Decision	Emergency action and detailed assessm	Further Action	<input type="checkbox"/> Ensure Safety <input checked="" type="checkbox"/> Refer for medical Assesment <input type="checkbox"/> Refer to Police <input type="checkbox"/> Prepare Report <input type="checkbox"/> Court Appearance <input type="checkbox"/> na
Detailed Assesment	<p>Medical checkup</p>		
<input type="button" value="Save"/> <input type="button" value="Clear"/> <input type="button" value="Cancel"/>			

Fig A7: Case intake form

Case Allocation Form

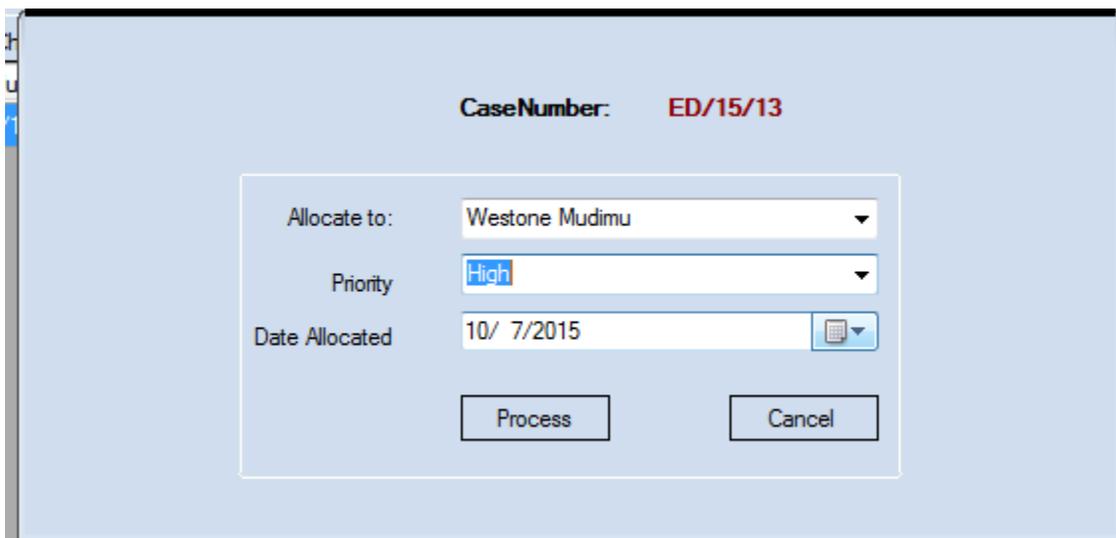
This form will display information of a case that needs to be allocated after searching by case number. Select the entry, then right click and the option to allocate the case will display. Click on the allocate button and an allocation screen displays. Using the drop down menu, you can select the officer to whom the case is allocated to. Once this has been done, click the process button.



The screenshot shows a search interface with the following elements:

- Select Search Criteria:** Radio buttons for "Case Number" (selected) and "Name".
- Search Input:** A text box containing "ED/15/13" and a "Load" button.
- Table:** A table with columns: CaseNumber, Name, Surname, Gender, Address, DateReferred, EmergencyAction, and Reas. The first row is highlighted in blue.
- Context Menu:** A menu is open over the first row, showing options: Allocate, Close Case, Delete, and Update.

CaseNumber	Name	Surname	Gender	Address	DateReferred	EmergencyAction	Reas
ED/15/13	Liberty	Svosve	Male	compensatio	1/1/2007	Refer for medical Assesment	1



The screenshot shows the Case Allocation Form for Case Number ED/15/13. The form contains the following fields and buttons:

- CaseNumber:** ED/15/13
- Allocate to:** A dropdown menu with "Westone Mudimu" selected.
- Priority:** A dropdown menu with "High" selected.
- Date Allocated:** A date picker showing "10/ 7/2015".
- Buttons:** "Process" and "Cancel".

Fig A8: Case allocation form

Case Assessment Form

The officer to whom the case is allocated will have to search through use of the case number.

The officer will then enter data gathered from case assessment.

This helps to gather information about a child so that we can identify his or her needs and make a plan to meet those needs.

Then after capturing all the details click save

Search Criteria
 Case Number ED/15/13 Name

Child

CaseNumber	Name	Surname	Gender	Address	DateReferred	EmergencyAction	ReasonOfRef
ED/15/13	Liberty	Svosve	Male	compensation house	1/1/2007	Refer for medical Assesment	1

Parents Champion Svosve

Survival Mother

General Health Good

Development Normal

Behavior disturbed

Social History leaving with a single parent

Educational N/A Other Organisations

Child's Comments the child is still young

Fig A9: Case assessment form

Care Plan Form

This form defines what action will be taken, by whom and when in order to make sure the child's needs are met and they are protected.

The Officer handling the case will determine the best course of action for the child and selects from a drop down menu the course of action relevant to the particular child. After selecting the preferred option click save.

Search Criteria

Case Number ED/15/13 Name

Child

CaseNumber	Name	Surname	Gender	Address	Date Referred
ED/15/13	Liberty	Svosve	Male	compensation house	1/1/2007

Case Plan

- Primary Care
 - Renunciation
 - Placement
- Secondary Care
 - Medical Assistance
 - Education
 - Counselling
 - Food
 - Supervision

Explain Case Plan: meeting the required needs

Date of Completion: 11/19/2015

Fig A10: Care plan form

Case Referral

This screen is used to transfer or refer a case to another strategic organization which has capacity to handle certain aspects of the case. For example the case below was referred to child line.

By clicking the process button the system will process the case and display the complete referral details for follow up purposes as displayed below:

The screenshot shows a software interface for case referrals. At the top, there is a 'Search Criteria' section with two radio buttons: 'Case Number' (selected) and 'Name'. Below this is a search input field and a 'Load Child' button. A table titled 'Child' displays a single record with columns for CaseNumber, Name, Surname, Gender, and Address. The record shows CaseNumber 'ED/15/13', Name 'Liberty', Surname 'Svosve', Gender 'Male', and Address 'comp'. Below the table is a 'Load Care Plan' button. A 'Details of Problem' section contains a 'Toggle All' button and a list of three checked items: 'Medical Assistance', 'Food', and 'Supervision'. The 'Reason of Referral' field contains the text 'test'. A 'Refer to:' dropdown menu is set to 'childline'. At the bottom, there are three buttons: 'Process', 'Clear', and 'Cancel'.

CaseNumber	Name	Surname	Gender	Address
ED/15/13	Liberty	Svosve	Male	comp

- Medical Assistance
- Food
- Supervision

Reason of Referral: test

Refer to: childline

REFERRAL FORM

To be completed in duplicate. One copy kept by referring organization, one by organization receiving the referral

Name of Child: LibertySvosve Date 10/7/2015
Date of Birth: 11/9/2006 12:00:00AM Sex Male
Name of Parent/Guardian: shushi svosve
Address: compensation house
Phone: 0776 635 737
Details of problem/need: Medical Assistance,Food,Supervision
Reason for referral: test
Referred By (Name): GarnetChiweshe Title 1
Organization: harare
Phone: 00263774352333
Referral sent to childline
Address: makoni
Phone: 0225 2741

Fig A11: Referral form

Case Review

This screen is used to check if the child's care plan is on track and continues to meet the child's needs. Search the case using the case number

Then add child's new needs, recommendations and date of the next review

Clicks save

Search Criteria

Case Number Name

Child

CaseNumber	Name	Surname	Gender	Address
ED/15/9	GODKNOWS	CHIMIZHE	Male	1479 LON

Assesed Abuse

- Abandonment
- Chronic illness
- Disability
- Juvenile Delinquency
- Neglect
- Other
- Out of School

Review Child's needs

still needs more support

Recommendations

the child need a wheelchair

Next Review Date

Fig A12: Case review

Case closure

This form is used to close or end a case after agreement between case management team, child and parents/caregiver. This is the point at which work with the child ends because situation is resolved or child dies. The child has to be aware of the case situation and also the case has to be approved in order to be closed. After capturing all relevant details, clicks save.

Search Criteria
 Case Number Name

Child

CaseNumber	Name	Surname	Gender	Address	DateReffered
ED/15/13	Liberty	Svosve	Male	compensation house	1/1/2007

Good

assessment of Curent Status

All needs are well taken off

Comment on Outstanding needs

The child's needs are all met

Reason Of Closure

Date of Notification:Child DateClosed Caregiver

Child aware? Type of Info Approver

Fig A13: Case closure

Reports

This activity views all types of reports generated in child protection case management system in excel format or pdf format for example case summary report and pending cases report.

The menu gives options of types of standard reports and the type of file the reader prefers to view in.

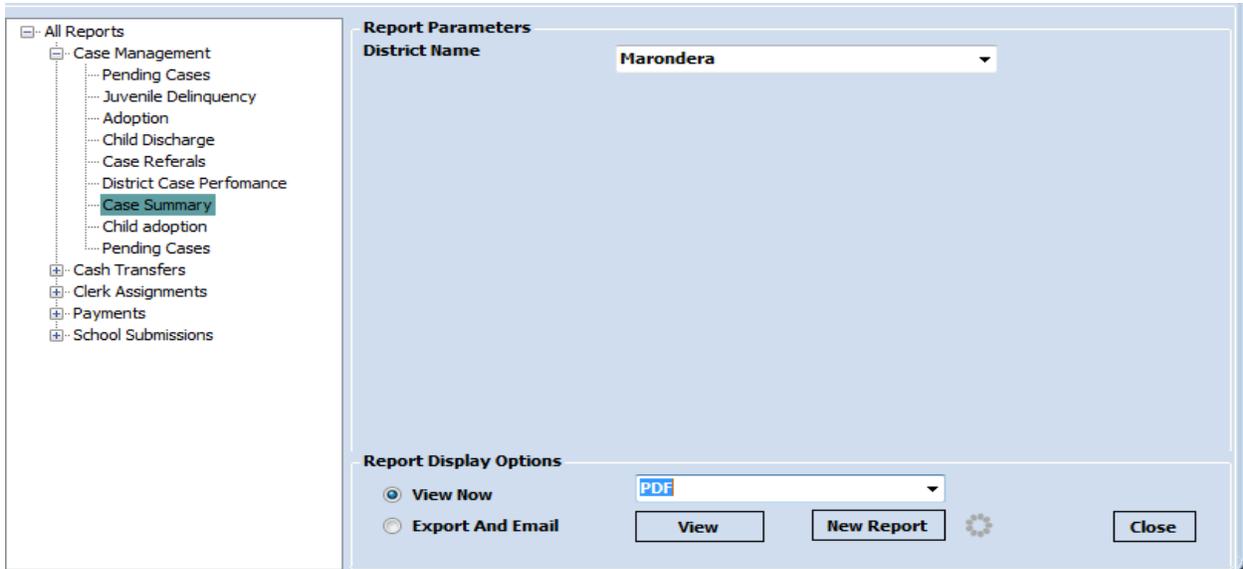


Fig A14: Report

Sample of a Report



Case Summary Report as at
10/7/2015



District	ReasonOfRef	Boys	Girls
Marondera	1	0	1
Marondera	6	5	4
Marondera	7	1	0
Marondera	9	1	0
Marondera	Neglect	0	1
Marondera		7.00	6.00
Grand Total		7.00	6.00

Fig A15: Report

Appendix B

Interview questions

INTERVIEW GUIDE FOR THE CHILD PROTECTION CASE MANAGEMENT SYSTEM

Section 1: INTERVIEW DETAILS

Province..... District.....

1. Interviewee’s position

2. Employee department

Male..... Female (Please tick appropriate)

Years in the current position.....

Section 2: INTERVIEW QUESTIONS

1. Can you please briefly describe the work that you do?

.....
.....

2. What type of information do you work with?

.....
.....

3. What processes are involved in collecting and storing the information?

.....
.....
.....
.....

.....
.....
4. May you please briefly explain how you process your information and the information retrieval system in place?

.....
.....
.....

5. Do you produce any reports? If yes, please describe the nature of reports that you produce and the frequency?

.....
.....
.....
.....

6. What constraints have you faced in the past in collection, documentation and retrieval of information?

.....
.....
.....

7. Looking at the way the current system is operating, what do you think could be done differently?

.....
.....
.....

8. What areas of the workflow process required to be completely overhauled for better efficiency?

.....
.....
.....

9. If a computerised system is to be developed, what in your view should it focus on in relation to the work flow process you described earlier?

.....
.....
.....
.....

Date: **Time**

Signature

Appendix C

Questionnaires

QUESTIONNAIRE FOR THE PRELIMINARY USABILITY REQUIREMENTS FOR CHILD PROTECTION CASE MANAGEMENT SYSTEM

My name is Garnet S Chiweshe a student at Midlands State University studying BSC Information Systems Honors Degree. One of the requirements for the study is a project report with a working system. If you agree, I would want to request you to spend a few minutes responding to a few questions to help me understand gaps in the current case management system for consideration of system improvements.

I hereby kindly request your assistance with the required information in this questionnaire. You have been selected to participate in this survey because of your knowledge, experience and contribution towards Case Management in Zimbabwe and you have used or have been in an environment where the system has been used. Your responses will be treated with uttermost confidentiality and will not be used for any other purpose outside the design of a case management MIS system as part of my study programme.

Thank you for your cooperation

Province..... District.....

1. Interviewee's position

2. Employee department

Male..... Female (Please tick appropriate)

Years in the current position.....

(TICK WHERE RELEVANT)

What can you say about the current system's performance?

EXCELLENT GOOD FAIR POOR

Do you think that the current system is reliable?

YES TO SOME EXTEND NO

Please Explain

.....
.....

What can you say about the current system's efficiency?

EXCELLENT GOOD POOR

How many cases do you attend per day?

1-5 5-10 10+

What type of technology do you use with the current system?

.....
.....

What are your views about the proposed system (Child Protection Case Management System)?

.....
.....

How do you secure your data with the current system?

.....
.....

What features would you want to see in a new and improved system?

.....
.....
.....

Thank you for taking time to answer these questions. Your participation is greatly appreciated

Appendix E: Code Snippet

Login Code

```
InitializeComponent();
txtUserName.Text = User.GetLastLoggedInUserInRegistry();
GMSPACK.User.RegisterInStartup(true);

private bool prompt;
public bool UserPrompted
{
    get { return prompt; }
}

string UserPassword, UserName;
string Section = null;

UserName = this.txtUserName.Text;
UserPassword = EncryptionDecryption.Encrypt(this.txtUserPassword.Text,
"fffff", "tttt", "MD5", 4, "@1B2c3D4e5F6g7H8", 256);
User.SetLoggedInUserInRegistry(txtUserName.Text);

GMSPACK.User.LogIn(UserName, UserPassword);

User.SetLoggedInUserInRegistry(txtUserName.Text);
//Alert that user is authenticated
GMSPACK.GlobalClass.isAuthenticated = true;
//LogProcessID();
int DaysLeft = PasswordPolicy.DaysLeftForPasswordExpiry();

if (DaysLeft <= 3)
{
    if (DaysLeft <= 0)
    {
        prompt = true;
        MessageBox.Show("Your Password expired today. Please change your
password.", "Password Expiry", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        frmChangePassword cp = new frmChangePassword();
        cp.UserPrompted = prompt;
        cp.Show();
        this.Hide();
        return;
    }

    else
    {
        MessageBox.Show("Your Password will expire in " + DaysLeft + "
days. Please consider changing your password.", "Password Expiry", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
}
}
```

```

        #region LOAD Controls
        //instantiate main Form
        frmMain fm1 = new frmMain();
        fm1.Show();
        #endregion
        //hide login Form
        this.Hide();
    }

    catch (Exception ex)
    {
        GMSPACK.GlobalClass.DisplayMessageBox(ex, "Security Policy");
        txtUserPassword.Clear();
        txtUserPassword.Focus();
    }

    private void CloseMenuItem_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Do you really want to close me?",
            "Are you sure?", MessageBoxButtons.YesNo,
            MessageBoxIcon.Exclamation,
            MessageBoxDefaultButton.Button2) == DialogResult.Yes)
        {
            Application.Exit();
        }
    }
    private void txtUserPassword_TextChanged(object sender, EventArgs e)
    {
        txtUserPassword.PasswordChar = '*';
    }

    private void btnUserLoginCancel_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}

```

User group

```

public partial class frmUserGroup : DevComponents.DotNetBar.Office2007RibbonForm
{
    // private DMSEntities context = null;

    public frmUserGroup()
    {
        InitializeComponent();
    }

    private void btnClear_Click(object sender, EventArgs e)

```

```

    {
        Utilities.ClearTextControls(this);
    }

    private void btnCancel_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void btnSave_Click(object sender, EventArgs e)
    {

```

Add user

```

        get { return user; }
        set { user = value; }

    public frmAddUsers()
    {
        InitializeComponent();

        private void btnCancel_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnClear_Click(object sender, EventArgs e)
        {
            Utilities.ClearTextControls(this);
        }

        private void validate()
        {
        }

        private void btnSave_Click(object sender, EventArgs e)
        {

        }

        private void frmUsers_Load(object sender, EventArgs e)
        {
            Utilities.LoadComboBox(cboGroup, "spGroupCbo");
            Utilities.LoadComboBox(cboStatus, "spStatusCbo");
        }

        private void txtPassword_MaskInputRejected(object sender,
MaskInputRejectedEventArgs e)

        private void txtPassword_TextChanged(object sender, EventArgs e)
        {
            txtPassword.PasswordChar = '*';
        }

        private void txtConPassword_TextChanged(object sender, EventArgs e)

```

```

    {
        txtConPassword.PasswordChar = '*';
    }
}

```

Param

```

    Unknown = 0, TextBox = 1, DatePicker, CheckBox, ComboBox };
    List<Control> ControlDetails = new List<Control>();
    // List<tblReportParameter> sysParams = new List<tblReportParameter>();
    ArrayList paramValues, sysParam;
    DataSet ds;
    bool insert = true;
    private void btnCancel_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void btnClear_Click(object sender, EventArgs e)
    {
        Utilities.ClearTextControls(tlpControls);
        insert = true;
        btnSave.Text = "Insert";
        btnDelete.Visible = false;
    }

    private void RenderControl(Label varLabel, Control varControl)
    {
        ControlDetails.Add(varControl);
        tlpControls.Controls.Add(varLabel, 0, ControlDetails.Count - 1);
        tlpControls.Controls.Add(varControl, 1, ControlDetails.Count - 1);
    }
    private Control getControl(ControlType control, string recordsource)
    {
        if (control == ControlType.TextBox)
        {
            TextBox varControl = new TextBox();
            varControl.Height = 21;
            varControl.Width = 249;
            return varControl;
        }
        else if (control == ControlType.CheckBox)
        {
            CheckBox varControl = new CheckBox();
            return varControl;
        }
        else if (control == ControlType.ComboBox)
        {
            ComboBox varControl = new ComboBox();
            varControl = setComboDetails(varControl, recordsource);
            varControl.Height = 21;
            varControl.Width = 249;
            return varControl;
        }
    }
}

```

```

else if (control == ControlType.DatePicker)
{
    DateTimePicker varControl = new DateTimePicker();
    varControl.Height = 21;
    varControl.Width = 249;
    return varControl;
}
else
{
    MessageBox.Show("Unknown control type");
    return null;
}
}
private ComboBox setComboDetails(ComboBox varControl, string recordsource)
{
    ArrayList param = new ArrayList();
    param.Clear();
    //Get source of drop down list:
    param.Add(recordsource);
    try
    {
        DataSet dsCombo = DataAccess.execSelProcedure("spCboGetDropDownList",
param);

        //Get source of drop down list:
        varControl.DataSource = dsCombo.Tables[0];
        varControl.DisplayMember = "Name";
        varControl.ValueMember = "ID";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        varControl = null;
    }
    //Set the value of each combo box:
    varControl.ValueMember = "ID";
    varControl.DisplayMember = "Name";
    return varControl;
}
private void LoadControls(DataSet dsColSettings, DataSet dsValues)
{
    GRIDPACK.ClearGrid(dgvParameters);
    //Use the returned result set to load dgvParameters:
    foreach (DataRow row in dsColSettings.Tables["Results"].Rows)
    {
        //Create a new control:
        DataGridViewColumn col = new DataGridViewColumn();
        DataGridViewComboBoxColumn colCombo = null;
        DataSet dsCombo; //Used for retaining values for
DataGridViewComboBoxcontrols:

        //Get the type of control type defined for this control from the
controlType field:
        DataGridViewCellTypes colType =
(DataGridViewCellTypes)Enum.Parse(typeof(DataGridViewCellTypes),
row["ColumnType"].ToString());
        ControlType controlType;
        //Determine the CellTemplate to use:
        switch (colType)

```

```

{
    case DataGridViewCellTypes.GridTextBox:
        col.CellTemplate = new DataGridViewTextBoxCell();
        controlType = ControlType.TextBox;
        break;
    case DataGridViewCellTypes.GridComboBox:
        //The control to which we will attach the
DataGridViewComboBoxCell cell template:
        colCombo = new DataGridViewComboBoxColumn();

        //Create object to set cell template details for combo control:
DataGridViewComboBoxCell cellCombo = new
DataGridViewComboBoxCell();
        cellCombo.DisplayStyle =
DataGridViewComboBoxDisplayStyle.Nothing;

        //Set the CellTemplaet of the DataGridViewComboBoxcontrol:
        colCombo.CellTemplate = cellCombo;

        col = (DataGridViewColumn)colCombo;
        controlType = ControlType.ComboBox;

        break;
    case DataGridViewCellTypes.GridDateTimePicker:
        CalendarColumn calCol = new CalendarColumn();
        col = (DataGridViewColumn)calCol;
        controlType = ControlType.DatePicker;
        break;
    case DataGridViewCellTypes.GridCheckBox:
        controlType = ControlType.CheckBox;
        break;
    default:
        // col.CellTemplate = new DataGridViewCheckBoxCell();
        controlType = ControlType.Unknown;
        MessageBox.Show("Unknown ControlType ");
        break;
}

//Set the control name:
col.Name = row["ColumnFieldName"].ToString();

//Set data property:
col.DataPropertyName = row["ColumnFieldName"].ToString();

//Set the control header text:
col.HeaderText = row["Caption"].ToString();
string ControlName=col.Name;
string LabelText=col.HeaderText;
string recordSource = row["DropDownListSource"].ToString();
//Add control to dgvParameters:
dgvParameters.Columns.Add(col);
//Add controls to panel:
Label varLabel = new Label();
varLabel.Text = LabelText;
varLabel.Name = "lbl"+varLabel.Text;
Control varControl = getControl(controlType, recordSource);
varControl.Name = ControlName;
RenderControl(varLabel, varControl);

```

```

if ((bool)row["HideColumn"].Equals(true))
{
    //Hide control:
    dgvParameters.Columns[col.Name].Visible = false;
    //Hide control & Label:
    tlpControls.Controls[ControlName].Visible = false;
    tlpControls.Controls[varLabel.Name].Visible = false;
}

//Populate the combo box with values from the database:
if (colType == DataGridViewCellTypes.GridComboBox)
{
    //Set DataGridViewComboBoxcontrol datasource:
    if (colCombo != null)
    {
        paramValues.Clear();

        //Get source of drop down list:
        paramValues.Add(row["DropDownListSource"].ToString());

        dsCombo = DataAccess.execSelProcedure("spCboGetDropDownList",
paramValues);

        colCombo.DataSource = dsCombo.Tables["Results"];
        colCombo.DisplayMember = "Name";
        colCombo.ValueMember = "ID";

        //Set the value of each row with a combo box:
        foreach (DataRow dgvRow in dsValues.Tables[0].Rows)
        {
            // dgvParameters.Rows[0].Cells[col.Name].Value =
dgvRow[row["ColumnFieldName"].ToString()];
            //dgvParameters.AllowUserToDeleteRows = true;
            break;
        }
    }
}

dgvParameters.Visible = true;
//dgvParameters.KeyUp();
dgvParameters.AutoGenerateColumns = false;
dgvParameters.DataMember = "Results";

dgvParameters.DataSource = dsValues;
}

private void twwParams_DoubleClick(object sender, EventArgs e)
{
    btnSave.Text = "Insert";
    btnDelete.Visible = false;
    insert = true;
    string spName;
    paramValues = new ArrayList();
    //DataGridViewComboBoxColumn colCombo = null;

```

```

        DataSet
        dsColSettings, //For the individual control settings of each system
parameter table:
        dsValues; //For retaining values stored in the database tables of each
system parameter:

        try
        {
            Utilities.selectParent(tvwParameters);
            //Retrieve the system parameter object attached to the selected node:
            paramValues.Add(tvwParameters.SelectedNode.Text);

            //Retrieve the select query with the values for the parameter table:
            dsValues = DataAccess.execSelProcedure("spCaseGetParamTableValues",
paramValues);

            //Clear Panel if there are any controls already being displayed:
            ControlDetails.Clear();
            tlpControls.Controls.Clear();
            tlpControls.RowStyles.Clear();

            //Clear parameter values list:
            paramValues.Clear();

            //Get caption name of the selected node:
            paramValues.Add(tvwParameters.SelectedNode.Text);

            //Use caption name to get the controls of the underlying base table:
            dsColSettings = DataAccess.execSelProcedure("spCaseGetParamTableColumns",
paramValues);

            if (dsColSettings.CreateDataReader().HasRows.Equals(true))
            {
                LoadControls(dsColSettings, dsValues);
            }
        }
        catch (Exception ex)
        {
            DisplayMessages.DisplayErrorMessage(ex.Message, "Error loading system
parameters");
        }
        finally
        {
            paramValues.Clear();
            // ds.Clear();
        }
    }

    private void frmParameters_Load(object sender, EventArgs e)
    {
        btnSave.Text = "Insert";
        btnDelete.Visible = false;
        tvwParameters.Nodes.Clear();
        string spName = "spCaseGetParamTables";

        try

```

```

    {
        ds = DataAccess.ExecuteCMD(spName);
        sysParam = new ArrayList();

        foreach (DataRow row in ds.Tables[0].Rows)
        {
            //Create node:
            TreeNode node = new TreeNode(row["Caption"].ToString());

            //Instantiate SystemParameter object in node tag:
            node.Tag = new GMSSystemParameter((int)row["ID"],
row["Caption"].ToString(), row[2].ToString(), row[3].ToString());

            //Load tree nodes:
            tvwParameters.Nodes.Add(node);

            //Populate system parameters array list:
            sysParam.Add(node.Tag);
        }
        tvwParameters.SelectedNode = null;
    }
    catch (Exception ex)
    {
        DisplayMessages.DisplayErrorMessage(ex.Message, "Error loading system
parameters");
    }
}

private void btnNewParam_Click(object sender, EventArgs e)
{
    try
    {
        //this.Hide();
        if (!GMSForm.IsAlreadyInstantiated(typeof(frmNewParam)))
        {
            frmNewParam NewParamForm = new frmNewParam();
            NewParamForm.FormClosed += new
FormClosedEventHandler(NewParamForm_Closed); //add event handler

            NewParamForm.Tag = this.GetType().Name; //tag the parent form's type
to the child

            NewParamForm.TopLevel = false;
            NewParamForm.Parent = this;
            NewParamForm.Parent = this;
            NewParamForm.Location = new
System.Drawing.Point((NewParamForm.Parent.Width - NewParamForm.Width) / 2,
(NewParamForm.Parent.Height - NewParamForm.Height) / 2);
            NewParamForm.Show();
            NewParamForm.BringToFront();
            NewParamForm.WindowState = FormWindowState.Maximized;
        }
    }
    catch (Exception ex)
    {
        DisplayMessages.DisplayErrorMessage(ex.Message, "Error loading system
parameters");
    }
}

```

```

    }
}
private void NewParamForm_Closed(object sender, EventArgs e)
{
    //TODO: Add code to execute when New Param form is closed:
    frmParameters_Load(this, e);
}
private string CreateSqlStatement( out string statusMessage)
//private string CreateSqlStatement(DataGridViewCellCancelEventArgs e, out string
statusMessage)
{
    int ControlCount = countControls();
    try
    {
        DataSet ds = (DataSet)dgvParameters.DataSource;
        int idCol = 0;
        bool idColChecked = false;

        string sqlStr;
        GMSSystemParameter sys =
(GMSSystemParameter)twvParameters.SelectedNode.Tag; //SystemParameter object contains
name of table to be affected:
        int eofCounter; //Used to check if eof reached in foreach loop below:

        string ID = null;

        //Loop through all the cells of the DataGridView object:
        foreach (Control ctr in tlpControls.Controls)
        {
            // if (ctr.Name == "msrepl_tran_version")
            //{

            //     break;
            //}
            if (ctr.Name == "ID")
            {

                break;
            }
        }

    }

    //Add new Data:
    if (insert)
    {
        DataSet dsi = (DataSet)dgvParameters.DataSource;
        DataTable dt = dsi.Tables[0];
        //Create INSERT statement:

        sqlStr = "INSERT INTO " + sys.TableName + " (";
        eofCounter = 0;

        #region Column Names foreach loop
        //Get the controls of the tables to be updated:
        foreach (Control ctr in tlpControls.Controls)
        {

            foreach (Control controls in ctr.Controls)

```

```

        {
            if (ctr is Label)
                continue;
        }

is CheckBox)
if(ctr is TextBox||ctr is ComboBox ||ctr is DateTimePicker||ctr
{
    //Skip inserting into the ID control:
    if (ctr.Name.Equals("ID")) /* NB:The ID control's header text
must ALWAYS = "ID": */
    {
        eofCounter++;
        // eofCounter = eofCounter + 1;// Controls++;
        continue;
    }
    if (ctr.Name.Equals("msrepl_tran_version"))
    {

        eofCounter++;
        //eofCounter = eofCounter + 1;
        // Controls++;
        continue;
    }

    //Retrieve all controls and add them to paramList:
    sqlStr = sqlStr + ctr.Name;

    //Increment counter:
    eofCounter++;

    //Determine whether the control added is the first or the last
one:

    //Check if control is not the last control:
    if (eofCounter < ControlCount) /* eofCounter must already have
been incremented when it gets here: */
    {
        sqlStr = sqlStr + ", ";
    }
    else if (eofCounter.Equals(ControlCount))//control is the last
control:
    {
        sqlStr = sqlStr + ") ";
    }
    //}
}
//Get the corresponding values to be inserted into each control:
}
#endregion

sqlStr = sqlStr + "VALUES (";
eofCounter = 0;

```

```

#region control Values foreach loop
//Get the values of the every control in the row:
foreach (Control ctr in tlpControls.Controls)
{
    if (ctr is TextBox || ctr is ComboBox || ctr is DateTimePicker ||
ctr is CheckBox)
    {
        // int ControlCount = countControls();
        //Skip inserting into the ID control:

        if (ctr.Name.Equals("ID"))
        {
            eofCounter++;
            //eofCounter = eofCounter + 1;
            continue;
        }
        if (ctr.Name.Equals("msrepl_tran_version"))
        {

            eofCounter++;
            //eofCounter = eofCounter + 1;
            // Controls++;
            continue;
        }

        if
(ctr.Name.ToLowerInvariant().Equals("CreatedBy".ToLowerInvariant()))
        {
            sqlStr = sqlStr + GMSPACK.GlobalClass.CurrentUser.UserID;
        }
        else if
(ctr.Name.ToLowerInvariant().Equals("ModifiedBy".ToLowerInvariant()))
        {
            sqlStr = sqlStr + GMSPACK.GlobalClass.CurrentUser.UserID;
        }
        else if
(ctr.Name.ToLowerInvariant().Equals("WorkStationIP".ToLowerInvariant()))
        {
            sqlStr = sqlStr + "" + GMSPACK.Session.GetLocalIPAddress() +
"";
        }
        else if
(ctr.Name.ToLowerInvariant().Equals("Modified".ToLowerInvariant()))
        {
            sqlStr = sqlStr + "" + DateTime.Now.ToString() + "";
        }
        //Get the values to be inserted into each control in the database
row. Include Date values, string values and null values in double quotes '':
        else if (ctr.Text.GetType().Equals(typeof(string)) ||
ctr.Text.GetType().Equals(typeof(DateTime)) || ctr.Text.Equals(System.DBNull.Value))
        //If the control is of type Combobox:
        {
            if (ctr.GetType().Equals(typeof(ComboBox)))
            {
                var cbo = ctr as ComboBox;
                sqlStr = sqlStr + "" + cbo.SelectedValue.ToString() +
""; //Use the EditedFormattedValue() method

```

```

        }
        else
        {
            sqlStr = sqlStr + "" + ctr.Text + "";
        }
    }
    else if (ctr.Text.GetType().Equals(typeof(bool))) //Convert bool
values to 1/0:
    {
        int bit;
        if ((bool)ctr.Text.Equals(true))
            bit = 1;
        else
            bit = 0;

        sqlStr = sqlStr + bit;
    }
    else
        sqlStr = sqlStr + ctr.Text;

    //Increment counter:
    eofCounter++;

    //Determine whether the control added is the first or the last
one:

    //Colcontrolcolumn is not the last control:
    if (eofCounter < ControlCount)
    {
        sqlStr = sqlStr + ", ";
    }
    else if (eofCounter.Equals(ControlCount))//control is the last
control:
    {
        sqlStr = sqlStr + ")";
    }

}
    //Get the corresponding values to be inserted into each control:
}
#endregion

    statusMessage = "saved";
}
else //ID control not null:
{
    //Create UPDATE statement:

    sqlStr = "UPDATE " + sys.TableName + " SET ";
    eofCounter = 1;

    #region foreach loop
    //Get the values of the every control in the row:
    foreach (Control ctr in tlpControls.Controls)
    {
        foreach (Control controls in ctr.Controls)
        {
            if (ctr is Label)

```

```

        continue;
    }

    if (ctr is TextBox || ctr is ComboBox || ctr is DateTimePicker ||
ctr is CheckBox)
    {
        //Get the value of the ID control separately:
        if (ctr.Name.Equals("ID"))
        {
            idCol = (int.Parse)(ctr.Text);
            eofCounter++;
            //eofCounter = eofCounter + 1;
            idColChecked = true;
            continue;
        }
        if (ctr.Name.Equals("msrepl_tran_version"))
        {

            eofCounter++;
            // eofCounter = eofCounter + 1;
            // Controls++;
            idColChecked = true;
            continue;
        }

        if
(ctr.Name.ToLowerInvariant().Equals("Createdby".ToLowerInvariant()))
        {
            eofCounter++;
            continue;
        }
        else if
(ctr.Name.ToLowerInvariant().Equals("ModifiedBy".ToLowerInvariant()) && eofCounter <
ControlCount)
        {
            sqlStr = sqlStr + ctr.Name + " = " +
GlobalClass.CurrentUser.UserID + ", ";
            eofCounter++;
            continue;
        }

        else if
(ctr.Name.ToLowerInvariant().Equals("WorkStationIP".ToLowerInvariant()) && eofCounter <
ControlCount)
        {
            sqlStr = sqlStr + ctr.Name + " = " + "'" +
GMSPACK.Session.GetLocalIPAddress() + "'" + ", ";
            eofCounter++;
            continue;
        }
        else if
(ctr.Name.ToLowerInvariant().Equals("Modified".ToLowerInvariant()) && eofCounter <
ControlCount)
        {
            sqlStr = sqlStr + ctr.Name + " = " + "'" +
DateTime.Now.ToString() + "'" + ", ";

```

```

        eofCounter++;
        continue;
    }

    //Last cell not reached and the ID control has been checked.
Append a ", " at the end of the string:
    else if (eofCounter < ControlCount && idColChecked == true)
    {
        if (ctr.Text.GetType().Equals(typeof(string)) ||
ctr.Text.GetType().Equals(typeof(DateTime)) ||
ctr.Text.GetType().Equals(typeof(System.DBNull)))
        {

            string cellValue;

            //If the cell is of type DropDownComboBoxcontrol:
            if (ctr.GetType().Equals(typeof(ComboBox)))
            {
                var cbo = ctr as ComboBox;
                cellValue = cbo.SelectedValue.ToString(); //Use
the SelectedValue() method:
            }
            else //Cell is not of type DropDownComboBoxcontrol:
            {
                cellValue = ctr.Text.ToString(); //Use the Value()
method:
            }

            sqlStr = sqlStr + ctr.Name + " = '" + cellValue + "', ";
        }
        else if (ctr.Text.GetType().Equals(typeof(bool)))
        {
            int bit;
            if ((bool)ctr.Text.Equals(true))
                bit = 1;
            else
                bit = 0;

            sqlStr = sqlStr + ctr.Name + " = " + bit + ", ";
        }
        else
        {
            sqlStr = sqlStr + ctr.Name + " = " + ctr.Text + ", ";
        }
    }
    else //Last cell reached. Don't append ", " at the end of the
string:
    {

        if
(ctr.Name.ToLowerInvariant().Equals("ModifiedBy".ToLowerInvariant()))
        {
            sqlStr = sqlStr + ctr.Name + " = " +
GlobalClass.CurrentUser.UserID;
        }
        else if
(ctr.Name.ToLowerInvariant().Equals("WorkStationIP".ToLowerInvariant()))
        {

```

```

        sqlStr = ctr.Name + " = " + "'" +
GMSPACK.Session.GetLocalIPAddress() + "'";
    }
    else if
(ctr.Name.ToLowerInvariant().Equals("Modified".ToLowerInvariant()))
    {
        sqlStr = sqlStr + ctr.Name + " = " + "'" +
DateTime.Now.ToString() + "'";
    }

    else if ((ctr.Text.GetType().Equals(typeof(string)) ||
ctr.Text.GetType().Equals(typeof(DateTime))) ||
ctr.Text.GetType().Equals(typeof(System.DBNull)))
    {
        string cellValue;

        //If the cell is of type DropDownComboBoxcontrol:
        if (ctr.GetType().Equals(typeof(ComboBox)))
        {
            var cbo = ctr as ComboBox;
            cellValue = cbo.SelectedValue.ToString(); //Use
the SelectedValue() method:
        }
        else //Cell is not of type DropDownComboBoxcontrol:
        {
            cellValue = ctr.Text.ToString(); //Use the Value()
method:
        }

        sqlStr = sqlStr + ctr.Name + " = '" + cellValue + "' ";
    }
    else if (ctr.Text.GetType().Equals(typeof(bool)))
    {
        int bit;
        if ((bool)ctr.Text.Equals(true))
            bit = 1;
        else
            bit = 0;

        sqlStr = sqlStr + ctr.Name + " = " + bit + " ";
    }
    else
        sqlStr = sqlStr + ctr.Name + " = " + ctr.Text + " ";

    //There must be at least two more controls before this comma
is added to avoid SQL syntax errors:
    if ((ControlCount - eofCounter) > 1)
        sqlStr = sqlStr + ", ";
    }
    //Increment counter:
    eofCounter++;
}
}
sqlStr = sqlStr + "WHERE " + sys.TableIdentityColumn + " = " +
idCol.ToString());
#endregion

statusMessage = "updated";

```

```

    }
    return sqlStr;
}
catch (Exception ex)
{
    throw ex;
}
}
private string CreateSqlStatement(DataGridViewCellCancelEventArgs e, out string
statusMessage)
{
    int ControlCount = countControls();
    try
    {
        DataSet ds = (DataSet)dgvParameters.DataSource;
        int idCol = 0;
        bool idColChecked = false;

        string sqlStr;
        GMSSystemParameter sys =
(GMSSystemParameter)twvParameters.SelectedNode.Tag; //SystemParameter object contains
name of table to be affected:
        int eofCounter; //Used to check if eof reached in foreach loop below:

        string ID = null;

        //Loop through all the cells of the DataGridView object:
        foreach (Control ctr in tlpControls.Controls)
        {
            if (ctr.Name == "ID")
            {
                break;
            }
            if (ctr.Name.Equals("msrepl_tran_version"))
            {
                break;
            }
        }
    }

    //Add new Data:
    if (insert)
    {
        DataSet dsi = (DataSet)dgvParameters.DataSource;
        DataTable dt = dsi.Tables[0];
        //Create INSERT statement:

        sqlStr = "INSERT INTO " + sys.TableName + " (";
        eofCounter = 0;

        #region Column Names foreach loop
        //Get the controls of the tables to be updated:
        foreach (Control ctr in tlpControls.Controls)
        {
            foreach (Control controls in ctr.Controls)

```

```

        {
            if (ctr is Label)
                continue;
        }

is CheckBox)
        if(ctr is TextBox||ctr is ComboBox ||ctr is DateTimePicker||ctr
        {
            //Skip inserting into the ID control:
            if (ctr.Name.Equals("ID")) /* NB:The ID control's header text
must ALWAYS = "ID": */
            {
                eofCounter++;
                // Controls++;
                continue;
            }

            //Retrieve all controls and add them to paramList:
            sqlStr = sqlStr + ctr.Name;

            //Increment counter:
            eofCounter++;

            //Determine whether the control added is the first or the last
one:

            //Check if control is not the last control:
            if (eofCounter < ControlCount) /* eofCounter must already have
been incremented when it gets here: */
            {
                sqlStr = sqlStr + ", ";
            }
            else if (eofCounter.Equals(ControlCount))//control is the last
control:
            {
                sqlStr = sqlStr + ") ";
            }
            //}
        }
        //Get the corresponding values to be inserted into each control:
    }
    #endregion

    sqlStr = sqlStr + "VALUES (";
    eofCounter = 0;

    #region control Values foreach loop
    //Get the values of the every control in the row:
    foreach (Control ctr in tlpControls.Controls)
    {
        if (ctr is TextBox || ctr is ComboBox || ctr is DateTimePicker ||
ctr is CheckBox)
        {
            // int ControlCount = countControls();
            //Skip inserting into the ID control:

```

```

        if (ctr.Name.Equals("ID"))
        {
            eofCounter++;
            continue;
        }

        if
(ctr.Name.ToLowerInvariant().Equals("Createdby".ToLowerInvariant()))
        {
            sqlStr = sqlStr + GMSPACK.GlobalClass.CurrentUser.UserID;
        }
        else if
(ctr.Name.ToLowerInvariant().Equals("ModifiedBy".ToLowerInvariant()))
        {
            sqlStr = sqlStr + GMSPACK.GlobalClass.CurrentUser.UserID;
        }
        else if
(ctr.Name.ToLowerInvariant().Equals("WorkStationIP".ToLowerInvariant()))
        {
            sqlStr = sqlStr + "" + GMSPACK.Session.GetLocalIPAddress() +
"";
        }
        else if
(ctr.Name.ToLowerInvariant().Equals("Modified".ToLowerInvariant()))
        {
            sqlStr = sqlStr + "" + DateTime.Now.ToString() + "";
        }
        //Get the values to be inserted into each control in the database
row. Include Date values, string values and null values in double quotes '':
        else if (ctr.Text.GetType().Equals(typeof(string)) ||
ctr.Text.GetType().Equals(typeof(DateTime)) || ctr.Text.Equals(System.DBNull.Value))
        //If the control is of type Combobox:
        {
            if (ctr.GetType().Equals(typeof(ComboBox)))
            {
                var cbo = ctr as ComboBox;
                sqlStr = sqlStr + "" + cbo.SelectedValue.ToString() +
""; //Use the EditedFormattedValue() method
            }
            else
            {
                sqlStr = sqlStr + "" + ctr.Text + "";
            }
        }
        else if (ctr.Text.GetType().Equals(typeof(bool))) //Convert bool
values to 1/0:
        {
            int bit;
            if ((bool)ctr.Text.Equals(true))
                bit = 1;
            else
                bit = 0;

            sqlStr = sqlStr + bit;
        }
        else
            sqlStr = sqlStr + ctr.Text;

```

```

        //Increment counter:
        eofCounter++;

        //Determine whether the control added is the first or the last
one:

        //Colcontrolcolumn is not the last control:
        if (eofCounter < ControlCount)
        {
            sqlStr = sqlStr + ", ";
        }
        else if (eofCounter.Equals(ControlCount))//control is the last
control:
        {
            sqlStr = sqlStr + ")";
        }

    }
    //Get the corresponding values to be inserted into each control:
}
#endregion

    statusMessage = "saved";
}
else //ID control not null:
{
    //Create UPDATE statement:

    sqlStr = "UPDATE " + sys.TableName + " SET ";
    eofCounter = 1;

    #region foreach loop
    //Get the values of the every control in the row:
    foreach (Control ctr in tlpControls.Controls)
    {
        foreach (Control controls in ctr.Controls)
        {
            if (ctr is Label)
                continue;

        }

        if (ctr is TextBox || ctr is ComboBox || ctr is DateTimePicker ||
ctr is CheckBox)
        {
            //Get the value of the ID control separately:
            if (ctr.Name.Equals("ID"))
            {
                idCol = (int.Parse)(ctr.Text);
                eofCounter++;
                idColChecked = true;
                continue;
            }

            if
(ctr.Name.ToLowerInvariant().Equals("Createdby".ToLowerInvariant()))
            {

```

```

        eofCounter++;
        continue;
    }
    else if
(ctr.Name.ToLowerInvariant().Equals("ModifiedBy".ToLowerInvariant())) && eofCounter <
ControlCount)
    {
        sqlStr = sqlStr + ctr.Name + " = " +
GlobalClass.CurrentUser.UserID + ", ";
        eofCounter++;
        continue;
    }
    else if
(ctr.Name.ToLowerInvariant().Equals("IsDeleted".ToLowerInvariant())) && eofCounter <
ControlCount)
    {
        sqlStr = sqlStr + ctr.Name + " = " + 1 + ", ";
        eofCounter++;
        continue;
    }

    else if
(ctr.Name.ToLowerInvariant().Equals("WorkStationIP".ToLowerInvariant())) && eofCounter <
ControlCount)
    {
        sqlStr = sqlStr + ctr.Name + " = " + "" +
GMSPACK.Session.GetLocalIPAddress() + "" + ", ";
        eofCounter++;
        continue;
    }
    else if
(ctr.Name.ToLowerInvariant().Equals("Modified".ToLowerInvariant())) && eofCounter <
ControlCount)
    {
        sqlStr = sqlStr + ctr.Name + " = " + "" +
DateTime.Now.ToString() + "" + ", ";
        eofCounter++;
        continue;
    }

    //Last cell not reached and the ID control has been checked.
Append a ", " at the end of the string:
    else if (eofCounter < ControlCount && idColChecked == true)
    {
        if (ctr.Text.GetType().Equals(typeof(string)) ||
ctr.Text.GetType().Equals(typeof(DateTime)) ||
ctr.Text.GetType().Equals(typeof(System.DBNull)))
        {

            string cellValue;

            //If the cell is of type DropDownListcontrol:
            if (ctr.GetType().Equals(typeof(ComboBox)))
            {
                var cbo = ctr as ComboBox;
                cellValue = cbo.SelectedValue.ToString(); //Use
the SelectedValue() method:

```

```

    }
    else //Cell is not of type DropDownListcontrol:
    {
        cellValue = ctr.Text.ToString(); //Use the Value()
    }

    sqlStr = sqlStr + ctr.Name + " = '" + cellValue + "', ";
}
else if (ctr.Text.GetType().Equals(typeof(bool)))
{
    int bit;
    if ((bool)ctr.Text.Equals(true))
        bit = 1;
    else
        bit = 0;

    sqlStr = sqlStr + ctr.Name + " = " + bit + ", ";
}
else
{
    sqlStr = sqlStr + ctr.Name + " = " + ctr.Text + ", ";
}
}
else //Last cell reached. Don't append ", " at the end of the
string:
{
    if
(ctr.Name.ToLowerInvariant().Equals("ModifiedBy".ToLowerInvariant()))
    {
        sqlStr = sqlStr + ctr.Name + " = " +
GlobalClass.CurrentUser.UserID;
    }
    else if
(ctr.Name.ToLowerInvariant().Equals("WorkStationIP".ToLowerInvariant()))
    {
        sqlStr = ctr.Name + " = " + "" +
GMSPACK.Session.GetLocalIPAddress() + "";
    }
    else if
(ctr.Name.ToLowerInvariant().Equals("Modified".ToLowerInvariant()))
    {
        sqlStr = sqlStr + ctr.Name + " = " + "" +
DateTime.Now.ToString() + "";
    }
    else if
(ctr.Name.ToLowerInvariant().Equals("IsDeleted".ToLowerInvariant()))
    {
        sqlStr = sqlStr + ctr.Name + " = " + "" + 1 + "";
    }

    else if ((ctr.Text.GetType().Equals(typeof(string)) ||
ctr.Text.GetType().Equals(typeof(DateTime))) ||
ctr.Text.GetType().Equals(typeof(System.DBNull)))
    {
        string cellValue;

```

```

//If the cell is of type DropDownComboBoxcontrol:
if (ctr.GetType().Equals(typeof(ComboBox)))
{
    var cbo = ctr as ComboBox;
    cellValue = cbo.SelectedValue.ToString(); //Use
the SelectedValue() method:
}
else //Cell is not of type DropDownComboBoxcontrol:
{
    cellValue = ctr.Text.ToString(); //Use the Value()
method:
}

sqlStr = sqlStr + ctr.Name + " = '" + cellValue + "' ";
}
else if (ctr.Text.GetType().Equals(typeof(bool)))
{
    int bit;
    if ((bool)ctr.Text.Equals(true))
        bit = 1;
    else
        bit = 0;

    sqlStr = sqlStr + ctr.Name + " = " + bit + " ";
}
else
    sqlStr = sqlStr + ctr.Name + " = " + ctr.Text + " ";

//There must be at least two more controls before this comma
is added to avoid SQL syntax errors:
if ((ControlCount - eofCounter) > 1)
    sqlStr = sqlStr + ", ";
}
//Increment counter:
eofCounter++;
}
}
sqlStr = sqlStr + "WHERE " + sys.TableIdentityColumn + " = " +
idCol.ToString();
#endregion

statusMessage = "updated";
}
return sqlStr;
}
catch (Exception ex)
{
    throw ex;
}
}
private int countControls()
{
    int countCB = 0;
    int countTB = 0;
    int countDTP = 0;
    int countCBO = 0;
    foreach (Control ctr in tlpControls.Controls)

```

```

    {
        if (ctr.GetType() == typeof(CheckBox))
        {
            countCB++;
        }
        else if (ctr.GetType() == typeof(TextBox))
        {
            countTB++;
        }
        else if (ctr.GetType() == typeof(DateTimePicker))
        {
            countDTP++;
        }
        else if (ctr.GetType() == typeof(ComboBox))
        {
            countCBO++;
        }
    }

    int controls = countTB + countCB + countDTP + countCBO;
    return controls;
}
private void btnSave_Click(object sender, EventArgs e)
{
    try
    {
        string statusMessage, sqlStatement; ;

        paramValues.Clear();
        foreach (Control ctr in tlpControls.Controls)
        {
            dgvParameters.ClearSelection();
            if ((ctr.Text == null || ctr.Text.ToString() == "") &&
ctr.Visible)
            {
                ctr.Focus();
                ctr.BackColor = Color.Azure;
                MessageBox.Show("Enter data in " + ctr.Name + " field, do not
leave it blank. The data was not saved", "System Parameters", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
                return;
            }
        }

        if (GlobalClass.DisplayMessageBox("Are you sure you want to save this
record?", "Save Confirmation", MessageBoxButtons.YesNo,
MessageBoxIcon.Question).Equals(DialogResult.Yes))
        {
            //Construct sql string
            sqlStatement = CreateSqlStatement( out statusMessage);
            //sqlStatement = CreateSqlStatement(new
DataGridViewCellCancelEventArgs(0, row.Index), out statusMessage);
            //Add sql string to paramValues ArrayList:
            paramValues.Add(sqlStatement);
        }
    }
}

```

```

        if (DataAccess.execProcedure("spExecCMD", paramValues) > 0)
        {
            GlobalClass.DisplayMessageBox(" successfully Saved", "System
Parameters.", MessageBoxButtons.OK, MessageBoxIcon.Information);
            Utilities.ClearTextControls(tlpControls);
            insert = true;
        }
        else
        {
            GlobalClass.DisplayMessageBox(" Failed to save", "System
Parameters.", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}
catch (Exception ex)
{
    GlobalClass.DisplayMessageBox(ex, "Error Updating Row");
}
}

private void dgvParameters_DoubleClick(object sender, EventArgs e)
{
}

private void dgvParameters_DataError(object sender,
DataGridViewDataErrorEventArgs e)
{
}

private void dgvParameters_CellClick(object sender, DataGridViewCellEventArgs e)
{
}

private void dgvParameters_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)
{
    try
    {
        insert = false;
        btnSave.Text = "Update";
        btnDelete.Visible = true;
        foreach (Control ctr in tlpControls.Controls)
        {
            if (ctr is TextBox || ctr is ComboBox || ctr is CheckBox || ctr is
DateTimePicker)
            {
                if (ctr.GetType().Equals(typeof(ComboBox)))
                {
                    var cbo = ctr as ComboBox;
                    cbo.Text =
dgvParameters.Rows[e.RowIndex].Cells[ctr.Name].EditedFormattedValue.ToString();
                }
            }
        }
    }
}

```

```

        else
        {
            ctr.Text =
dgvParameters.Rows[e.RowIndex].Cells[ctr.Name].Value.ToString();
        }
    }
}
catch (Exception ex)
{
    GlobalClass.DisplayMessageBox(ex.Message, "Binding Parameters");
}
}

private void btnDelete_Click(object sender, EventArgs e)
{
    insert = false;
    {
        DataGridViewRow row = dgvParameters.SelectedRows[0];
        row.Cells["IsDeleted"].Value = true;
        if (dgvParameters.IsCurrentRowDirty.Equals(true))
        {
            try
            {
                string statusMessage, sqlStatement; ;

                paramValues.Clear();

                if (GlobalClass.DisplayMessageBox("Are you sure you want to
delete this record?", "Delete Confirmation", MessageBoxButtons.YesNo,
MessageBoxIcon.Question).Equals(DialogResult.Yes))
                {

                    //Construct sql string
                    // sqlStatement = CreateSqlStatement(out statusMessage);
                    sqlStatement = CreateSqlStatement(new
DataGridViewCellCancelEventArgs(0, row.Index), out statusMessage);
                    //Add sql string to paramValues ArrayList:
                    paramValues.Add(sqlStatement);

                    if (DataAccess.execProcedure("spExecCMD", paramValues) > 0)
                    {
                        GlobalClass.DisplayMessageBox("Record deleted
successfully.", "System Parameters.", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }
                    else
                    {
                        GlobalClass.DisplayMessageBox("Record failed to delete.",
"System Parameters.", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }
                }
            }
            else
            {
                }
            }
        }
    }
}
catch (Exception ex)

```



```

    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex, "Error loading form");
    }
}

private void twvPermissions_AfterCheck(object sender, TreeViewEventArgs e)
{
    try
    {
        if (e.Action.Equals(TreeViewAction.ByMouse))
        {
            GMSTreeView.ToggleAllChildNodes(e.Node, e.Node.Checked);
            GMSTreeView.CheckParentNode(e.Node);
        }
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex, "TreeView Error");
    }
}

private void cmdToggle_Click(object sender, EventArgs e)
{
    try
    {
        GMSTreeView.ToggleTreeViewNodes(tvwPermissions);
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex, "Error toggling tree nodes");
    }
}

private void optUserGroup_CheckedChanged(object sender, EventArgs e)
{
    //Disable cboUserGroup if optUserGroup is not checked, else enable it:
    if (optUserGroup.Checked == false)
    {
        cboUserGroup.Enabled = false;
    }
    else
    {
        cboUserGroup.Enabled = true;
    }
}

private void optUser_CheckedChanged(object sender, EventArgs e)
{
    //Disable cboUser if optUser is not checked, else enable it:
    if (optUser.Checked == false)
    {
        cboUser.Enabled = false;
    }
    else
    {
        cboUser.Enabled = true;
    }
}

```

```

    }
}

private void cmdExpandAll_Click(object sender, EventArgs e)
{
    if (!AllNodesExpanded)
    {
        tvwPermissions.ExpandAll();
        AllNodesExpanded = true;
        cmdExpandAll.Text = "Collapse All";
    }
    else
    {
        tvwPermissions.CollapseAll();
        AllNodesExpanded = false;
        cmdExpandAll.Text = "Expand All";
    }
}

private void cboUser_SelectedValueChanged(object sender, EventArgs e)
{
    try
    {
        if (IsFormLoading)
        {
            return;
        }
        else
        {
            tvwPermissions.Nodes.Clear();

            //Reload grid
            GMSTreeView.LoadNodes(tvwPermissions, arrMenuItem); //Load base nodes:

            //User selects group
            paramValues = new ArrayList();

            //Get the set permissions for that group from the database:
            paramValues.Add(cboUser.SelectedValue);

            //Get permissions DataSet:
            DataSet ds = DataAccess.execSelProcedure("spCaseGetUserPerms",
paramValues);

            int usergroupID = (int)DataAccess.executeScalar("spGetUserGroupID",
paramValues);

            if (ds.Tables[0].Rows.Count > 0)
            {
                LoadUserGroupPerms = false; //Don't load user group permissions

                //Get user group id
                cboUserGroup.SelectedValue = usergroupID;

                LoadUserGroupPerms = true;

                //Load permissions
                GMSTreeView.LoadGroupPermissions(ds, tvwPermissions.Nodes);

                //Remove the nodes that are not checked
                //GMSTreeView.RemoveUncheckedNodes(tvwPermissions);
            }
        }
    }
}

```

```

    }
    else //load group permissions if non are available
    {
        paramValues = new ArrayList();
        paramValues.Add(usergroupID);
        //cboUserGroup.SelectedValue = usergroupID;

        DataSet data =
        DataAccess.execSelProcedure("spCaseGetUserGroupPerms", paramValues);

        GMSTreeView.LoadGroupPermissions(data, tvwPermissions.Nodes);
    }
}
}
catch (Exception ex)
{
    Utilities.ServiceLog(ex.Message);
}
}

private void cboUserGroup_SelectedValueChanged(object sender, EventArgs e)
{
    if (IsFormLoading)
    {
        return;
    }
    else
    {
        if (LoadUserGroupPerms)
        {
            tvwPermissions.Nodes.Clear();

            //Reload grid
            GMSTreeView.LoadNodes(tvwPermissions, arrMenuItem); //Load base nodes:

            //User selects group
            paramValues = new ArrayList();

            //Get the set permissions for that group from the database:
            paramValues.Add(cboUserGroup.SelectedValue);

            //Get permissions DataSet:
            DataSet ds = DataAccess.execSelProcedure("spCaseGetUserGroupPerms",
paramValues);

            GMSTreeView.LoadGroupPermissions(ds, tvwPermissions.Nodes);
        }
    }
}

private void cmdApply_Click(object sender, EventArgs e)
{
    if (GlobalClass.DisplayMessageBox("Are you sure you want to save these
permissions?", "Save Confirmation", MessageBoxButtons.YesNo,
MessageBoxIcon.Question).Equals(DialogResult.Yes))
    {
        try
        {

```

```

        if (optUserGroup.Checked)
        {
            //SavePermissions(tvwPermissions.Nodes);
            BulkCopy.BulkCopySavePerms(tvwPermissions.Nodes,
(int)cboUserGroup.SelectedValue, PermissionType.UserGroup, "dbo.tblCaseUserPermissions",
"dbo.tblCasePermissions");
            GlobalClass.DisplayMessageBox("Permissions updated successfully
in the database.", "Save Confirmation");
        }
        else if(optUser.Checked)
        {
            //SaveUserPermissions(tvwPermissions.Nodes);
            BulkCopy.BulkCopySavePerms(tvwPermissions.Nodes,
(int)cboUser.SelectedValue, PermissionType.User, "dbo.tblCaseUserPermissions",
"dbo.tblCasePermissions");
            GlobalClass.DisplayMessageBox("Permissions updated successfully
in the database.", "Save Confirmation");
            refreshToolStripMenuItem_Click(this, e);
        }
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex, "Error saving permissions");
    }
}

private void SavePermissions(TreeNodeCollection nodes)
{
    try
    {
        foreach (TreeNode node in nodes)
        {
            paramValues.Clear();

            //Retrieve the node's GSMMenuItem:
            GSMMenuItem mnuItem = (GSMMenuItem)node.Tag;

            //Add parameters:
            paramValues.Add(cboUserGroup.SelectedValue);
            paramValues.Add((int)mnuItem.MenuItemID);
            paramValues.Add(node.Checked);

            //Commit changes to database:
            if (DataAccess.execProcedure("spCaseInsSavePerms", paramValues) > 0)
                counter++;

            //Recursively loop through all child nodes if they are present:
            if (node.Nodes != null)
                SavePermissions(node.Nodes);
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

```

private void SaveUserPermissions(TreeNodeCollection nodes)
{
    try
    {
        foreach (TreeNode node in nodes)
        {
            paramValues.Clear();

            //Retrieve the node's GSMMenuItem:
            GSMMenuItem mnuItem = (GSMMenuItem)node.Tag;

            //Add parameters:
            paramValues.Add(cboUser.SelectedValue);
            paramValues.Add((int)mnuItem.MenuItemID);
            paramValues.Add(node.Checked);

            //Commit changes to database:
            if (DataAccess.execProcedure("spCaseInsSaveUserPerms ", paramValues)
                > 0)
                counter++;

            //Recursively loop through all child nodes if they are present:
            if (node.Nodes != null)
                SaveUserPermissions(node.Nodes);
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

private void refreshToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (optUser.Checked)
    {
        cboUser_SelectedValueChanged(this, e);
    }
    else if (optUserGroup.Checked)
    {
        cboUserGroup_SelectedValueChanged(this, e);
    }
}

private void refreshUserPermissionsToolStripMenuItem_Click(object sender,
EventArgs e)
{
    try
    {
        if (optUser.Checked)
        {
            if (GlobalClass.ShowDialogBox("Are you sure you want to reset
this user's permissions to the default user group permission settings?", "Save
Confirmation", MessageBoxButtons.YesNo,
MessageBoxIcon.Question).Equals(DialogResult.Yes))
            {
                //Reload user group permissions
                cboUserGroup_SelectedValueChanged(this, e);
            }
        }
    }
}

```

```

        RemoveUncheckedNodes(tvwPermissions.Nodes);

        //Save the group permissions for the currently selected user
        //BulkCopy.BulkCopySavePerms(tvwPermissions.Nodes,
(int)cboUser.SelectedValue, PermissionType.User);

        //Refresh Treeview
        //refreshToolStripMenuItem_Click(this, e);
        //GlobalClass.DisplayMessageBox("Permissions successfully reset
in the database.", "Save Confirmation");
    }
}
}
catch (Exception ex)
{
    throw ex;
}
}

public void RemoveUncheckedNodes(TreeNodeCollection nodes)
{
    try
    {
        //Loop through all the nodes in the TreeNodeCollection:
        foreach (TreeNode tn in nodes)
        {
            //Remove unchecked node:
            if (!tn.Checked)
                tn.Remove();
            else //Node is checked:
            {
                //Loop through child nodes and remove any unchecked children:
                if (tn.Nodes.Count > 0)
                    RemoveUncheckedNodes(tn.Nodes);
            }
        }
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex, "Error removing nodes");
    }
}

private void cmdClose_Click(object sender, EventArgs e)
{
    this.Close();
}
}
}
}

```

Case intake

```

public partial class frmIntakeAllocation :
DevComponents.DotNetBar.Office2007RibbonForm

```

```

{
    public frmIntakeAllocation()
    {
        InitializeComponent();
    }
    private string CaseNumber = null;
    DataTable dtPar, dtSi;
    private void frmIntakeAllocation_Load(object sender, EventArgs e)
    {
        try
        {
            //Utilities.LoadComboBox(cboEmergencyAction, "spEmergencyAction");
            Utilities.LoadComboBox(cboRefRelationship, "spRelationship");
            Utilities.LoadComboBox(cboParGender, "spGenderCbo");
            Utilities.LoadComboBox(cboSibRelationship, "spRelationship");
            Utilities.LoadComboBox(cboParentRelationship, "spRelationship");
            Utilities.LoadComboBox(cboSibGender, "spGenderCbo");
            Utilities.LoadComboBox(cboDecision, "spCboDecisionType");
            //Utilities.LoadComboBox(cboFurtherAction, "spEmergencyAction");
            Utilities.LoadComboBox(cboProvince, "spCboProvince");
            Utilities.LoadComboBox(cboChildGender, "spGenderCbo");
            Utilities.LoadParentView(tvwAction, "spEmergencyAction");
            Utilities.LoadComboBox(cboReasons, "spCboCaseNature");
            Utilities.LoadParentView(tvwFurtherAction, "spEmergencyAction");
            //Utilities.LoadComboBox(cboRefReason, "spCboCaseNature");
            // Utilities.LoadComboBox(cboUserName, "spUserCbo");
            //txtOtherReasons.Visible = true;
            //cbOtherReasons.Checked = false;
            //txtOtherReasons.Hide()// = true;
            dtPar = new DataTable();
            initialiseParTable(dtPar);
            dgvParents.DataSource = dtPar;
            dtSi = new DataTable();
            initialiseSiTable(dtSi);
            lblspecify.Visible = false;
            txtOther.Visible = false;
            dgvSiblings.DataSource = dtSi;
        }
        catch (Exception ex)
        {
            GlobalClass.DisplayMessageBox(ex.Message, "Loading Values");
        }
    }

    private void tpChildInfo_Click(object sender, EventArgs e)
    {

    }

    private void cboProvince_SelectedIndexChanged(object sender, EventArgs e)
    {
        try
        {
            ArrayList param = new ArrayList();
            param.Add(cboProvince.Text);
            Utilities.LoadComboBox(cboDistrict, "spCboDistrictPro", param);
        }
        catch (Exception ex)

```

```

    {
        GlobalClass.DisplayMessageBox(ex.Message, "Loading Values");
    }
}

private void cboDistrict_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        CaseNumber = null;
        lblCaseNumber.Text=null;
        ArrayList param = new ArrayList();
        param.Add(cboDistrict.Text);
        Utilities.LoadComboBox(cboWard, "spCboWardDist", param);
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex.Message, "Loading Values");
    }
}

private void btnSave_Click(object sender, EventArgs e)
{
    try
    {
        ArrayList paramR = new ArrayList();

        ArrayList paramA = new ArrayList();
        TreeNodeCollection nodesA = tvwAction.Nodes;
        foreach (TreeNode pn in nodesA)
        {
            if (pn.Checked == true)
            {
                paramA.Add(pn.Text);
            }
        }

        ///////////////decision////////////////////////////////////
        ArrayList paramD= new ArrayList();
        TreeNodeCollection nodesD= tvwFurtherAction.Nodes;
        foreach (TreeNode pd in nodesD)
        {
            //foreach (TreeNode cn in pn.Nodes)
            //{
            if (pd.Checked == true)
            {
                paramD.Add(pd.Text);
            }
            //}
        }

        string BirthRecord = null;
        //string ReasonOfRef = string.Join(",",
        (string[])paramR.ToArray(Type.GetType("System.String")));
    }
}

```

```

string Province = cboProvince.SelectedValue.ToString();
string District = cboDistrict.SelectedValue.ToString();
string Ward = cboWard.SelectedValue.ToString();
string Name = txtNameOfChild.Text;
string Surname = txtSurname.Text;
DateTime DOB = dtpChildDOB.Value;
string Address = txtChildAddress.Text;
DateTime DateReffered = dtpDateofRef.Value;
string EmergencyAction = string.Join(",",
(string[])paramA.ToArray(Type.GetType("System.String")));
string Referee = txtRefName.Text;
string Relationship = cboRefRelationship.SelectedValue.ToString();
string RefAge = txtRefAge.Text;
string RefAddress = txtRefAddress.Text;
string RefPhone = txtRefPhone.Text;
//string ReasonOfRef = ""; // cboRefReason.SelectedValue.ToString();
string ReasonOfRef =
cboReasons.SelectedValue.ToString();//string.Join(",",
(string[])paramR.ToArray(Type.GetType("System.String")));
string Gender = cboChildGender.SelectedValue.ToString();
//string AllocatedTo = cboUserName.Text;
//DateTime DateAllocated = dtpDateAllocated.Value;
if (rdoEst.Checked == true)
{
    BirthRecord = "Esimated";
}
else
{
    BirthRecord = "Documented";
}
//add Child
ArrayList param = new ArrayList();
param.Add(Province);
param.Add(District);
param.Add(Ward);
param.Add(Name);
param.Add(Surname);
param.Add(Gender);
param.Add(BirthRecord);
param.Add(DOB);
param.Add(Address);
param.Add(DateReffered);
param.Add(EmergencyAction);
param.Add(Referee);
param.Add(Relationship);
param.Add(RefAge);
param.Add(RefAddress);
param.Add(RefPhone);
//param.Add(CaseNumber);
param.Add(ReasonOfRef);
//param.Add(AllocatedTo);
//param.Add(DateAllocated);
DataSet message= DataAccess.execSelProcedure("spCaseIntake", param);
CaseNumber = message.Tables[0].Rows[0][0].ToString();
//Add Parent
for (int i = 0; i < dgvParents.Rows.Count; i++)
{
    string pGender = dgvParents.Rows[i].Cells["Gender"].Value.ToString();

```

```

        string pDOB = dgvParents.Rows[i].Cells["DOB"].Value.ToString();
        string pName = dgvParents.Rows[i].Cells["Name"].Value.ToString();
        string Occupation =
dgvParents.Rows[i].Cells["Occupation"].Value.ToString();
        string pAddress =
dgvParents.Rows[i].Cells["Address"].Value.ToString();
        string Phone = dgvParents.Rows[i].Cells["Phone"].Value.ToString();
        string Deceased =
dgvParents.Rows[i].Cells["Deceased"].Value.ToString();
        string Parent =
dgvParents.Rows[i].Cells["Relationship"].Value.ToString();
        ArrayList parame = new ArrayList();
        parame.Add(CaseNumber);
        parame.Add(pGender);
        parame.Add(pName);
        parame.Add(Parent);
        parame.Add(pDOB);
        parame.Add(Occupation);
        parame.Add(pAddress);
        parame.Add(Phone);
        parame.Add(Deceased);
        DataAccess.execSelProcedure("spCaseInsParent", parame);
    }
    //add siblings
    for (int i = 0; i < dgvSiblings.Rows.Count; i++)
    {
        string Nam = dgvSiblings.Rows[i].Cells["Name"].Value.ToString();
        string Rel =
dgvSiblings.Rows[i].Cells["Relationship"].Value.ToString();
        string Age = dgvSiblings.Rows[i].Cells["Age"].Value.ToString();
        string sAddress =
dgvSiblings.Rows[i].Cells["Address"].Value.ToString();
        string sGender =
dgvSiblings.Rows[i].Cells["Gender"].Value.ToString();
        ArrayList para = new ArrayList();
        para.Add(CaseNumber);
        para.Add(Nam);
        para.Add(Rel);
        para.Add(Age);

        para.Add(sAddress);
        para.Add(sGender);
        DataAccess.execSelProcedure("spCaseInsSiblings", para);
    }
    //add Decision

    string Decision = cboDecision.Text;//paramD
    //string Reason= txtWhyDecision.Text;
    string Action = string.Join(",",
(string[])paramD.ToArray(Type.GetType("System.String")));//
    string DetailedAss=txtAssessment.Text;
    //string DetEmergencyAction = txtDetEmergencyAction.Text;
    ArrayList par = new ArrayList();
    par.Add(CaseNumber);
    par.Add(Decision);
    //par.Add(Reason);
    par.Add(Action);
    par.Add(DetailedAss);

```

```

        //par.Add(DetEmergencyAction);
        DataAccess.execSelProcedure("spCaseInsDecision", par);

        SqlConnection sqlcon= DataAccess.getConnection();
        DisplayMessages.DisplayInfoMessage("Successfully Saved :"+CaseNumber,
"Saved");
        //sqlcon.InfoMessage += new
        SqlInfoMessageEventHandler(myConnection_InfoMessage);
        Utilities.ClearTextControls(this);
        //GRIDPACK.ClearGrid(dgvParents);
        //GRIDPACK.ClearGrid(dgvParents);

    }
    catch (Exception ex)
    {
        DisplayMessages.DisplayErrorMessage(ex.Message, "could not save");
    }

}
void myConnection_InfoMessage(object sender, SqlInfoMessageEventArgs e)
{
    DisplayMessages.DisplayInfoMessage(e.Message, "Saved");
}
private void btnClear_Click(object sender, EventArgs e)
{
    Utilities.ClearTextControls(this);
}

private void btnCancel_Click(object sender, EventArgs e)
{
    this.Close();
}

private void btnUpdate_Click(object sender, EventArgs e)
{
    try
    {
        DataRow dr;

        dr = dtPar.NewRow();

        dr["Name"] = txtParName.Text.Trim();
        dr["DOB"] = dtpParDOB.Value.ToShortDateString().Trim();
        dr["Gender"] = cboParGender.Text.Trim();
        dr["Relationship"] = cboParentRelationship.Text.Trim();
        dr["Occupation"] = txtParOccupation.Text.Trim();
        dr["Address"] = txtParAddress.Text.Trim();
        dr["Phone"] = txtParPhone.Text.Trim();
        dr["Deceased"] = dtpParDateDeceased.Value.ToShortDateString().Trim();
        dtPar.Rows.Add(dr);
        Utilities.ClearTextControls(gbParent);
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex.Message, "Grid Error");
    }
}
}

```

```

private void initialiseParTable(DataTable dt)
{
    try
    {
        dt.Columns.Add(new DataColumn("Name"));
        dt.Columns.Add(new DataColumn("Gender"));
        dt.Columns.Add(new DataColumn("Relationship"));
        dt.Columns.Add(new DataColumn("DOB"));
        dt.Columns.Add(new DataColumn("Occupation"));
        dt.Columns.Add(new DataColumn("Address"));
        dt.Columns.Add(new DataColumn("Phone"));
        dt.Columns.Add(new DataColumn("Deceased"));
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex.Message, "Loading Values");
    }
}

private void initialiseSiTable(DataTable dt)
{
    try
    {
        dt.Columns.Add(new DataColumn("Name"));
        dt.Columns.Add(new DataColumn("Gender"));
        dt.Columns.Add(new DataColumn("Age"));
        dt.Columns.Add(new DataColumn("Relationship"));
        dt.Columns.Add(new DataColumn("Address"));
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex.Message, "Loading Values");
    }
}

private void tpParentsGuardian_Click(object sender, EventArgs e)
{
}

private void btnAdd_Click(object sender, EventArgs e)
{
    try
    {
        DataRow dr;

        dr = dtSi.NewRow();

        dr["Name"] = txtSibName.Text.Trim();
        dr["Age"] = txtSibAge.Text.Trim();
        dr["Gender"] = cboSibGender.Text.Trim();
        dr["Relationship"] = cboSibRelationship.Text.Trim();
        dr["Address"] = txtSibAddress.Text.Trim();
        dtSi.Rows.Add(dr);
        Utilities.ClearTextControls(gbSiblings);
    }
    catch (Exception ex)

```

```

        {
            GlobalClass.DisplayMessageBox(ex.Message, "Grid Error");
        }
    }

private void cboWard_SelectedIndexChanged(object sender, EventArgs e)

private void cboRefReason_SelectedIndexChanged(object sender, EventArgs e)

private void rdoOtherReasons_CheckedChanged(object sender, EventArgs e)

private void cbOtherReasons_CheckedChanged(object sender, EventArgs e)

private void txtRefAge_KeyPress(object sender, KeyPressEventArgs e)
{
    ValidateControls.StrictNumber(sender, e);
}

private void txtSibAge_KeyPress(object sender, KeyPressEventArgs e)
{
    ValidateControls.StrictNumber(sender, e);
}

private void cboReasons_SelectedValueChanged(object sender, EventArgs e)
{
}

private void groupBox1_Enter(object sender, EventArgs e)
{
}

private void cboReasons_SelectionChangeCommitted(object sender, EventArgs e)
{
    if (cboReasons.GetItemText(cboReasons.SelectedItem).ToString() == "Other")
    {
        lblspecify.Visible = true;
        txtOther.Visible = true;
    }
    else
    {
        lblspecify.Visible = false;
        txtOther.Visible = false;
    }
}

private void txtOther_TextChanged(object sender, EventArgs e)
{
    Utilities.expandTextBox(this.txtOther, this);
}

private void txtAssessment_TextChanged(object sender, EventArgs e)
{
    Utilities.expandTextBox(this.txtAssessment, this);
}

```

```

    }
}
}

```

Case allocation

```

        InitializeComponent();
    {
private bool cont = false;
private string caseNumber = null;
private string searchCriteria = null;
string value = null;
string CaseID = null;
public string CaseNumber

        set { caseNumber = value; }
    }
public bool Cont
{
    set { cont=value;}
}
private void frmCaseAllocation_Load(object sender, EventArgs e)
{
    if(cont==true)
    {

    }
    rdoCaseNumber.Checked = true;
}

private void groupBox1_Enter(object sender, EventArgs e)
{

}

private void groupBox2_Enter(object sender, EventArgs e)
{

}

private void txtCaseNumber_Leave(object sender, EventArgs e)
{

}

private void rdoCaseNumber_CheckedChanged(object sender, EventArgs e)
{
    if(rdoCaseNumber.Checked==true)
    {
        txtName.Enabled = false;
        txtCaseNumber.Enabled = true;
        searchCriteria = "number";
    }
}

```

```

        else if (rdoName.Checked == true)
        {
            txtName.Enabled = true;
            txtCaseNumber.Enabled = false;
            searchCriteria = "name";
        }
    }

private void rdoName_CheckedChanged(object sender, EventArgs e)
{
    if (rdoCaseNumber.Checked == true)
    {
        txtName.Enabled = false;
        txtCaseNumber.Enabled = true;
        searchCriteria = "number";
    }
    else if (rdoName.Checked == true)
    {
        txtName.Enabled = true;
        txtCaseNumber.Enabled = false;
        searchCriteria = "name";
    }
}

private void btnLoad_Click(object sender, EventArgs e)
{
    try
    {
        loadChildGrid.loadGrid(this.value, this.searchCriteria,
this.rdoCaseNumber, this.rdoName, this.txtCaseNumber, this.txtName, this.dgvCaseDetails);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void dgvCaseDetails_MouseDown(object sender, MouseEventArgs e)
{
    try
    {
        if (e.Button == MouseButtons.Right)
        {
            CaseID = null;
            var hti = dgvCaseDetails.HitTest(e.X, e.Y);
            dgvCaseDetails.ClearSelection();
            dgvCaseDetails.Rows[hti.RowIndex].Selected = true;
            cmsAllocations.Show(dgvCaseDetails, e.Location);
            CaseID =
dgvCaseDetails.Rows[hti.RowIndex].Cells["ID"].Value.ToString();
            caseNumber=
dgvCaseDetails.Rows[hti.RowIndex].Cells["CaseNumber"].Value.ToString();

```

```

    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void tsmiAllocate_Click(object sender, EventArgs e)
{
    try
    {
        if (!GMSForm.IsAlreadyInstantiated(typeof(frmAllocateTo)))
        {
            frmAllocateTo frmAt = new frmAllocateTo();
            frmAt.Tag = this.GetType().Name; //tag the parent form's type to the
child
            frmAt.TopLevel = false;
            frmAt.Parent = this;
            frmAt.Location = new System.Drawing.Point((frmAt.Parent.Width -
frmAt.Width) / 2, (frmAt.Parent.Height - frmAt.Height) / 2);
            frmAt.CaseID= CaseID;
            frmAt.CaseNumber = caseNumber;
            frmAt.BringToFront();
            frmAt.Show();
            // frmAt.WindowState = FormWindowState.Maximized;

        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void tsmiCloseCase_Click(object sender, EventArgs e)
{
    if (!GMSForm.IsAlreadyInstantiated(typeof(frmCaseClosure)))
    {
        frmCaseClosure frmcc = new frmCaseClosure();
        frmcc.Tag = this.GetType().Name; //tag the parent form's type to the
child
        frmcc.TopLevel = false;
        frmcc.Parent = this;
        frmcc.Location = new System.Drawing.Point((frmcc.Parent.Width -
frmcc.Width) / 2, (frmcc.Parent.Height - frmcc.Height) / 2);
        frmcc.InitialStage = true;
        frmcc.CaseNumber = caseNumber;
        frmcc.BringToFront();
        frmcc.Show();
        frmcc.WindowState = FormWindowState.Maximized;

    }
}

```

```

    }

    private void tsmiDelete_Click(object sender, EventArgs e)
    {
        try
        {
            if (GlobalClass.DisplayMessageBox("Are you sure you want to delete this
record?", "Delete Confirmation", MessageBoxButtons.YesNo,
MessageBoxIcon.Question).Equals(DialogResult.Yes))
            {
                ArrayList param = new ArrayList();
                param.Add(CaseID);
                DataAccess.execSelProcedure("spDeleteCase", param);
                loadChildGrid.loadGrid("", this.searchCriteria,
this.rdoCaseNumber, this.rdoName, this.txtCaseNumber, this.txtName, this.dgvCaseDetails);
                GlobalClass.DisplayMessageBox("Record deleted successfully.",
"System Parameters.", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }
        catch (Exception ex)
        {
            GlobalClass.DisplayMessageBox(ex, "Error Deleting Row");
        }
    }
}

```

Case Assessment

```

private string searchCriteria = null;
string value = null;
public frmCaseAssessment()
{
    InitializeComponent();
}
private bool cont = false;
private string caseNumber;
public string CaseNumber
{
    set { caseNumber = value; }
}
public bool Cont
{
    set { cont = value; }
}

private void btnLoad_Click(object sender, EventArgs e)
{
    try
    {
        loadChildGrid.loadGrid(this.value, this.searchCriteria,
this.rdoCaseNumber, this.rdoName, this.txtCaseNumber, this.txtName, this.dgvCaseDetails);
    }
    catch (Exception ex)

```

```

        {
            MessageBox.Show(ex.Message);
        }
    }

    private void rdoCaseNumber_CheckedChanged(object sender, EventArgs e)
    {
        try
        {
            selectCriteria.criteria(this.rdoCaseNumber, this.rdoName, this.txtName,
            this.txtCaseNumber, this.searchCriteria);
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void rdoName_CheckedChanged(object sender, EventArgs e)
    {
        try
        {
            selectCriteria.criteria(this.rdoCaseNumber, this.rdoName, this.txtName,
            this.txtCaseNumber, this.searchCriteria);
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void frmCaseAssessment_Load(object sender, EventArgs e)
    {
        try
        {
            rdoCaseNumber.Checked = true;
            Utilities.LoadComboBox(cboDevelopment, "spCboDevelopment");
            ArrayList param = new ArrayList();
            param.Add(GlobalClass.CurrentUser.loginName);
            Utilities.LoadComboBox(cboAssessedBy, "spDistrictUsers", param);
            Utilities.LoadCheckList(clstOtherorganisations, "spDistrictPartners",
            param, false);
            // Utilities.LoadComboBox(cboBehavior, "spCboBehavior");
            //Utilities.LoadComboBox(cboEducational, "spLeveOfEducationCbo");
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void btnProcess_Click(object sender, EventArgs e)
    {
        try {
            string caseNumber =
            dgvCaseDetails.CurrentRow.Cells["CaseNumber"].Value.ToString();

```

```

ArrayList param = new ArrayList();
param.Add(caseNumber);
//param.Add(txtFamily.Text);
param.Add(txtSurvival.Text);
param.Add(txtGeneralHealth.Text);
param.Add(cboDevelopment.Text);
param.Add(txtBehavior.Text);
param.Add(txtSocialHistory.Text);
param.Add(txtEducational.Text);
param.Add(txtChildComments.Text);
param.Add(txtConclusion.Text);
param.Add(dtLastAssessment.Value);
param.Add(cboAssessedBy.Text);

Utilities.BindControls(this, "spInsAssessment", param);

for (int i = 0; i < clstOtherorganisations.CheckedItems.Count; i++)
{
    ArrayList param2 = new ArrayList();
    string userid = clstOtherorganisations.CheckedItems[i].ToString();
    param2.Add(caseNumber);
    param2.Add(userid);
    DataAccess.execSelProcedure("spCaseInsOtherOrg", param2);
}

DisplayMessages.DisplayInfoMessage("Sucesfully saved", "Saved");
Utilities.ClearTextControls(this);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void txtSurvival_TextChanged(object sender, EventArgs e)
{
    Utilities.expandTextBox(this.txtSurvival, this);
}

private void txtGeneralHealth_TextChanged(object sender, EventArgs e)
{
    Utilities.expandTextBox(this.txtGeneralHealth, this);
}

private void txtBehavior_TextChanged(object sender, EventArgs e)
{
    Utilities.expandTextBox(this.txtBehavior, this);
}

private void txtSocialHistory_TextChanged(object sender, EventArgs e)
{
    Utilities.expandTextBox(this.txtSocialHistory, this);
}

```

```

private void txtEducational_TextChanged(object sender, EventArgs e)
{
    Utilities.expandTextBox(this.txtEducational, this);
}

private void txtOtherOrganisations_TextChanged(object sender, EventArgs e)
{
    //Utilities.expandTextBox(this.txtOtherOrganisations, this);
}

private void txtChildComments_TextChanged(object sender, EventArgs e)
{
    Utilities.expandTextBox(this.txtChildComments, this);
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    Utilities.expandTextBox(this.txtConclusion, this);
}

private void panel13_Paint(object sender, PaintEventArgs e)
{
}

private void txtParents_TextChanged(object sender, EventArgs e)
{
    Utilities.expandTextBox(this.txtParents, this);
}

private void panel19_Paint(object sender, PaintEventArgs e)
{
}

private void txtSurvival_KeyPress(object sender, KeyPressEventArgs e)
{
    this.SelectNextControl(txtGeneralHealth, true, true, true, true);
}

private void txtParents_KeyPress(object sender, KeyPressEventArgs e)
{
    this.SelectNextControl(txtSurvival, true, true, true, true);
}

private void txtGeneralHealth_KeyPress(object sender, KeyPressEventArgs e)
{
    this.SelectNextControl(cboDevelopment, true, true, true, true);
}

private void dgvCaseDetails_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        string CaseNumber =
dgvCaseDetails.CurrentRow.Cells["CaseNumber"].Value.ToString();

```

```

        ArrayList param = new ArrayList();
        param.Add(CaseNumber);
        DataSet ds = DataAccess.execSelProcedure("spViewCaseAssessments", param);
        dgvCaseDetails.DataSource = ds;
        if (ds.Tables[0].Rows.Count > 0)
        {
            txtEducational.Text = ds.Tables[0].Rows[0]["Educational"].ToString();
            txtSurvival.Text = ds.Tables[0].Rows[0]["Survival"].ToString();
            txtGeneralHealth.Text =
ds.Tables[0].Rows[0]["GeneralHealth"].ToString();
            cboDevelopment.Text = ds.Tables[0].Rows[0]["Development"].ToString();
            txtBehavior.Text = ds.Tables[0].Rows[0]["Behavior"].ToString();
            txtSocialHistory.Text =
ds.Tables[0].Rows[0]["SocialHistory"].ToString();
            txtChildComments.Text =
ds.Tables[0].Rows[0]["ChildComments"].ToString();
            txtConclusion.Text = ds.Tables[0].Rows[0]["Conclusion"].ToString();
            cboAssessedBy.Text=
ds.Tables[0].Rows[0]["lastAssessedBy"].ToString();
            dtLastAssessment.Text =
ds.Tables[0].Rows[0]["Dateassessed"].ToString();
        }

        else
        { MessageBox.Show("No Case Assessment details have been recorded for this
case"); }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void dgvCaseDetails_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
}

private void btnCancel_Click(object sender, EventArgs e)
{
    this.Close();
}

private void btnClear_Click(object sender, EventArgs e)
{
    txtEducational.Text = "";
    txtSurvival.Text = "";
    txtGeneralHealth.Text = "";
    cboDevelopment.Text = "";
    txtBehavior.Text = "";
    txtSocialHistory.Text = "";
    txtChildComments.Text = "";
    txtConclusion.Text = "";
}

private void dgvCaseDetails_MouseDown(object sender, MouseEventArgs e)

```

```

    {
        try
        {
            if (e.Button == MouseButtons.Right)
            {
                // CaseID = null;
                var hti = dgvCaseDetails.HitTest(e.X, e.Y);
                dgvCaseDetails.ClearSelection();
                dgvCaseDetails.Rows[hti.RowIndex].Selected = true;
                cmsAssessment.Show(dgvCaseDetails, e.Location);
                //CaseID =
                dgvCaseDetails.Rows[hti.RowIndex].Cells["ID"].Value.ToString();
                caseNumber =
                dgvCaseDetails.Rows[hti.RowIndex].Cells["CaseNumber"].Value.ToString();
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

```

Case Outstanding

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using DataManager;
using System.Collections;
using GMSPACK;

namespace CaseManager.Forms
{
    public partial class frmOutStandingCases :
    DevComponents.DotNetBar.Office2007RibbonForm
    {
        public frmOutStandingCases()
        {
            InitializeComponent();
        }

        private void frmOutStandingCases_Load(object sender, EventArgs e)
        {
            ArrayList param = new ArrayList();
            string user = User.GetLastLoggedInUserInRegistry();
            DataTable dt = new DataTable();
            param.Add(user);
            DataSet ds = DataAccess.execSelProcedure("spNS_GetMessages", param);
            if (ds.Tables[0].Rows.Count > 0)

```

```

        {
            dgvPendingIssues.DataSource = ds;

            dgvPendingIssues.DataMember = "Results";
        }
    }
}

```

Case Care Plan

```

private string searchCriteria = null;
string value = null;
private bool AllNodesExpanded = false;
private bool initialStage = false;
public bool InitialStage
{
    set { initialStage = value; }
}
private string caseNumber;
public string CaseNumber
{
    set { caseNumber = value; }
}
public frmCasePlan()
{
    InitializeComponent();
}

private void rdoName_CheckedChanged(object sender, EventArgs e)
{
    selectCriteria.criteria(this.rdoCaseNumber, this.rdoName, this.txtName,
this.txtCaseNumber, this.searchCriteria);
}

private void rdoCaseNumber_CheckedChanged(object sender, EventArgs e)
{
    selectCriteria.criteria(this.rdoCaseNumber, this.rdoName, this.txtName,
this.txtCaseNumber, this.searchCriteria);
}

private void btnLoad_Click(object sender, EventArgs e)
{
    try
    {
        loadChildGrid.loadGrid(this.value, this.searchCriteria,
this.rdoCaseNumber, this.rdoName, this.txtCaseNumber, this.txtName, this.dgvCaseDetails);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

```

private void frmCasePlan_Load(object sender, EventArgs e)
{
    ArrayList param = new ArrayList();
    try
    {
        DataSet dsv=new DataSet();
        DataSet values = new DataSet();

        if (initialStage == true)
        {
            rdoCaseNumber.Checked = true;
            txtCaseNumber.Text = caseNumber;
            param.Add(caseNumber);
            dsv = DataAccess.execSelProcedure("spPlanlistParam", param);
            string value = dsv.Tables[0].Rows[0]["Parent"].ToString();
            values = Strings.ArrayListToDataSet(value);
            //Utilities.LoadParentView(tvChecklist, values, param);
            LoadTreeView(tvChecklist, "spPlanlist");
            loadChildGrid.loadGrid(this.caseNumber, this.searchCriteria,
this.rdoCaseNumber, this.rdoName, this.txtCaseNumber, this.txtName, this.dgvCaseDetails);
        }
        else
        {
            rdoCaseNumber.Checked = true;
            Utilities.LoadTreeView(tvChecklist, "spPlanlist");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
private void LoadTreeView(TreeView tvw, string spName)
{
    ArrayList param = new ArrayList();
    DataSet values = new DataSet();
    DataSet dsv = new DataSet();
    bool exists = false;
    param.Add(caseNumber);
    dsv = DataAccess.execSelProcedure("spPlanlistParam", param);
    string value = dsv.Tables[0].Rows[0]["Parent"].ToString();
    values = Strings.ArrayListToDataSet(value);
    DataSet ds = DataAccess.execProcedure(spName);

    List<string> treeNode = (from r in ds.Tables["Results"].AsEnumerable()

                            select
r.Field<string>("Parent")).Distinct().ToList();

    foreach (string parent in treeNode)
    {
        tvw.Nodes.Add(parent, parent);
        tvw.Nodes[parent].Tag = parent;
    }
}

```

```

//List<string> Parent = treeNode;
foreach (DataRow dr in ds.Tables["Results"].Rows)
{
    // string parent in treeNode
    TreeNode child = new TreeNode();
    child.Tag = dr["ChildID"].ToString().Trim();
    child.Text = dr["Child"].ToString().Trim();
    //tvw.Nodes["Parent"].Nodes.Add(child);
    tvw.Nodes[dr["Parent"].ToString()].Nodes.Add(child);
    //foreach (DataRow drv in values.Tables["Results"].Rows)
    //{
        //exists = values.Tables[0].Select().ToList().Exists(row =>
row["Parent"].ToString().ToUpper() == child.Text);
        String searchNode = child.Text;
        bool contains = values.Tables[0].AsEnumerable()
            .Any(row => searchNode ==
row.Field<String>("Parent"));
        if (contains)
        {
            child.Checked = true;
        }
    //}
}

}
private void btnSave_Click(object sender, EventArgs e)
{
    try
    {
        ArrayList parame = new ArrayList();
        TreeNodeCollection nodes = tvChecklist.Nodes;
        foreach (TreeNode pn in nodes)
        {
            foreach (TreeNode cn in pn.Nodes)
            {
                if (cn.Checked == true)
                {
                    parame.Add(cn.Text);
                }
            }
        }
        string CasePlan = string.Join(", ",
(string[])parame.ToArray(Type.GetType("System.String")));
        string CaseNumber=
dgvCaseDetails.CurrentRow.Cells["CaseNumber"].Value.ToString();
        DateTime DateOfCompletion = dtpDateCompletion.Value;
        string CaseExplanation = txtExplainCarePlan.Text;
        ArrayList param = new ArrayList();

        param.Add(CaseNumber);
        param.Add(CasePlan);
        param.Add(DateOfCompletion);
        param.Add(CaseExplanation);
    }
}

```

```

        DataAccess.execSelProcedure("spInsCasePlan", param);
        DisplayMessages.DisplayInfoMessage("Successfully Saved :" + CaseNumber, "Case
Plan");
        if (initialStage)
        {
            this.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

private void btnExpandAll_Click(object sender, EventArgs e)
{
    if (!AllNodesExpanded)
    {
        tvChecklist.ExpandAll();
        AllNodesExpanded = true;
        btnExpandAll.Text = "Collapse All";
    }
    else
    {
        tvChecklist.CollapseAll();
        AllNodesExpanded = false;
        btnExpandAll.Text = "Expand All";
    }
}

private void btnToggleAll_Click(object sender, EventArgs e)
{
    try
    {
        GMSTreeView.ToggleTreeViewNodes(tvChecklist);
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex, "Error toggling tree nodes");
    }
}

private void btnCancel_Click(object sender, EventArgs e)
{
    this.Close();
}

private void btnClear_Click(object sender, EventArgs e)
{
    Utilities.ClearTextControls(this);
}
}

```

Case Referral

```

string searchCriteria = null;
string value = null;
public frmCaseReferral()
{
    InitializeComponent();
}

private void frmCaseReferral_Load(object sender, EventArgs e)
{
    try
    {
        rdoCaseNumber.Checked = true;
        Utilities.LoadComboBox(cboReferTo, "spPartnerCBO");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void rdoName_CheckedChanged(object sender, EventArgs e)
{
    selectCriteria.criteria(this.rdoCaseNumber, this.rdoName, this.txtName,
this.txtCaseNumber, this.searchCriteria);
}

private void rdoCaseNumber_CheckedChanged(object sender, EventArgs e)
{
    try
    {
        selectCriteria.criteria(this.rdoCaseNumber, this.rdoName, this.txtName,
this.txtCaseNumber, this.searchCriteria);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void btnLoad_Click(object sender, EventArgs e)
{
    try
    {
        loadChildGrid.loadGrid(this.value, this.searchCriteria,
this.rdoCaseNumber, this.rdoName, this.txtCaseNumber, this.txtName, this.dgvCaseDetails);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void btnLoadProblem_Click(object sender, EventArgs e)
{
    try
    {

```

```

        ArrayList param = new ArrayList();
        DataSet dsv = new DataSet();
        DataSet values = new DataSet();
        string caseNumber =
dgvCaseDetails.CurrentRow.Cells["CaseNumber"].Value.ToString();
        param.Add(caseNumber);
        dsv = DataAccess.execSelProcedure("spPlanlistParam", param);
        string value = dsv.Tables[0].Rows[0]["Parent"].ToString();
        values = Strings.ArrayListToDataSet(value);
        Utilities.LoadParentView(tvwAction, values, param);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void btnToggleAll_Click(object sender, EventArgs e)
{
    try
    {
        GMSTreeView.ToggleTreeViewNodes(tvwAction);
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex, "Error toggling tree nodes");
    }
}

private void btnProcess_Click(object sender, EventArgs e)
{
    ArrayList parame = new ArrayList();
    TreeNodeCollection nodes = tvwAction.Nodes;
    foreach (TreeNode cn in nodes)
    {
        if (cn.Checked == true)
        {
            parame.Add(cn.Text);
        }
    }
    string CaseProblem= string.Join(",",
(string[])parame.ToArray(Type.GetType("System.String")));
    string ReasonOfRef = txtReason.Text;
    string Organisation = cboReferTo.SelectedValue.ToString();
    string caseNumber =
dgvCaseDetails.CurrentRow.Cells["CaseNumber"].Value.ToString();
    int User = GMSPACK.GlobalClass.CurrentUser.UserID;
    ArrayList param = new ArrayList();

    param.Add(caseNumber);
    param.Add(CaseProblem);
    param.Add(ReasonOfRef);
    param.Add(User);
    param.Add(Organisation);
    DataAccess.execSelProcedure("spInsCaseReferral", param);
    ArrayList caseNo = new ArrayList();
    caseNo.Add(caseNumber);

```

```

        DisplayMessages.DisplayInfoMessage("Successfully Saved :" + caseNumber,
"Saved");
        Emails.PrintReport("Case_Referral", "rptCaseReferral", "spRptCaseReferral",
caseNo);
    }

    private void btnClear_Click(object sender, EventArgs e)
    {
        Utilities.ClearTextControls(this);
    }

    private void btnCancel_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}

```

Case Review

```

string searchCriteria = null;
string value = null;
public frmCaseReview()
{
    InitializeComponent();
}

private void frmCaseReview_Load(object sender, EventArgs e)
{
    try
    {
        rdoCaseNumber.Checked = true;
        Utilities.LoadParentView(tvwAction, "spCboCaseNature");
        //Utilities.LoadParentView(tvwReasons, "spCboCaseNature");
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex.Message, "Loading Values");
    }
}

private void cboProvince_SelectedIndexChanged(object sender, EventArgs e)

private void cboDistrict_SelectedIndexChanged(object sender, EventArgs e)

private void cboCaseNumber_SelectedIndexChanged(object sender, EventArgs e)

private void dd_Click(object sender, EventArgs e)
{
}

```

```

private void label18_Click(object sender, EventArgs e)
{
}

private void label1_Click(object sender, EventArgs e)
{
}

private void button2_Click(object sender, EventArgs e)
{
    Utilities.ClearTextControls(this);
}

private void btnSave_Click(object sender, EventArgs e)
{
    try
    {
        ArrayList paramA = new ArrayList();
        TreeNodeCollection nodesA = tvwAction.Nodes;
        foreach (TreeNode pn in nodesA)
        {
            if (pn.Checked == true)
            {
                paramA.Add(pn.Text);
            }
        }
        string Assessed = string.Join(",",
(string[])paramA.ToArray(Type.GetType("System.String")));
        string CaseNumber=
dgvCaseDetails.CurrentRow.Cells["CaseNumber"].Value.ToString();
        ArrayList param = new ArrayList();
        param.Add(CaseNumber);
        param.Add(Assessed);
        param.Add(txtReviewChildNeeds.Text);
        param.Add(txtRecommendations.Text);
        param.Add(dtpNextReviewDate.Value.ToShortDateString());
        DataAccess.execSelProcedure("spInsCaseReview", param);
        DisplayMessages.DisplayInfoMessage("Successfully saved", "Saved");
        Utilities.ClearTextControls(this);
    }
    catch (Exception ex)
    {
        DisplayMessages.DisplayErrorMessage(ex.Message, "could not save");
    }
}

private void tvwAction_AfterSelect(object sender, TreeViewEventArgs e)
{
}

```

```

        private void rdoName_CheckedChanged(object sender, EventArgs e)
        {
            selectCriteria.criteria(this.rdoCaseNumber, this.rdoName, this.txtName,
this.txtCaseNumber, this.searchCriteria);
        }

        private void rdoCaseNumber_CheckedChanged(object sender, EventArgs e)
        {
            selectCriteria.criteria(this.rdoCaseNumber, this.rdoName, this.txtName,
this.txtCaseNumber, this.searchCriteria);
        }

        private void btnLoad_Click(object sender, EventArgs e)
        {
            try
            {
                loadChildGrid.loadGrid(this.value, this.searchCriteria,
this.rdoCaseNumber, this.rdoName, this.txtCaseNumber, this.txtName, this.dgvCaseDetails);
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void btnReviewCasePlan_Click(object sender, EventArgs e)
        {
            if (!GMSForm.IsAlreadyInstantiated(typeof(frmCasePlan)))
            {
                frmCasePlan frmcp = new frmCasePlan();
                frmcp.Tag = this.GetType().Name; //tag the parent form's type to the
child
                frmcp.TopLevel = false;
                frmcp.Parent = this;
                frmcp.Location = new System.Drawing.Point((frmcp.Parent.Width -
frmcp.Width) / 2, (frmcp.Parent.Height - frmcp.Height) / 2);
                frmcp.InitialStage = true;
                frmcp.CaseNumber =
dgvCaseDetails.CurrentRow.Cells["CaseNumber"].Value.ToString(); ;
                frmcp.BringToFront();
                frmcp.Show();
                frmcp.WindowState = FormWindowState.Maximized;
            }
        }
    }
}

```

Case closure

```

namespace CaseManager.Forms
{
    public partial class frmCaseClosure : DevComponents.DotNetBar.Office2007.RibbonForm
    {

```

```

private string searchCriteria = null;
string value = null;
public frmCaseClosure()
{
    InitializeComponent();
}
private bool initialStage = false;
public bool InitialStage
{
    set { initialStage = value; }
}
private string caseNumber;
public string CaseNumber
{
    set { caseNumber = value; }
}
private void frmCaseClosure_Load(object sender, EventArgs e)
{
    try
    {
        if (initialStage == true)
        {
            rdoCaseNumber.Checked = true;

            txtCaseNumber.Text = caseNumber;
            loadChildGrid.loadGrid(this.caseNumber, this.searchCriteria,
this.rdoCaseNumber, this.rdoName, this.txtCaseNumber, this.txtName, this.dgvCaseDetails);

        }
        rdoCaseNumber.Checked = true;
        //Utilities.LoadComboBox(cboProvince, "spCboProvince");
        Utilities.LoadComboBox(cboChildAware, "spCboChildAware");
        Utilities.LoadComboBox(cboTypeOfInfo, "spCboChildInfo");
        Utilities.LoadComboBox(cboUser, "spUserCbo");
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex.Message, "Loading Values");
    }
}

private void comboBox3_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void cboProvince_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void button3_Click(object sender, EventArgs e)
{
    this.Close();
}

private void button1_Click(object sender, EventArgs e)

```

```

    {
        MouseEventArgs me = (MouseEventArgs)e;
        try
        {
            string Number =
dgvCaseDetails.CurrentRow.Cells["CaseNumber"].Value.ToString();

            ArrayList param = new ArrayList();
            param.Add(Number);
            param.Add(dtpDateClosed.Value.ToShortDateString());
            param.Add(txtAssessment.Text);
            param.Add(txtOutstandingNeeds.Text);
            param.Add(txtReasonOfClosure.Text);
            param.Add(dtpDonChild.Value.ToShortDateString());
            param.Add(dtpDonCareGiver.Value.ToShortDateString());
            // param.Add(cboChildAware.Text);
            param.Add(cboTypeOfInfo.Text);
            param.Add(cboUser.ValueMember.ToString());
            param.Add(cboChildAware.Text);

            DataAccess.execSelProcedure("spInsCaseClosure", param);
            DisplayMessages.DisplayInfoMessage("Successfully saved", "Saved");
            Utilities.ClearTextControls(this);
            initialStage = false;
        }
        catch (Exception ex)
        {
            DisplayMessages.DisplayErrorMessage(ex.Message, "could not save");
        }
    }

private void btnClear_Click(object sender, EventArgs e)
{
    Utilities.ClearTextControls(this);
}

private void rdoName_CheckedChanged(object sender, EventArgs e)
{
    try
    {
        selectCriteria.criteria(this.rdoCaseNumber, this.rdoName, this.txtName,
this.txtCaseNumber, this.searchCriteria);
    }
    catch (Exception ex)
    {
        GlobalClass.DisplayMessageBox(ex.Message, "Loading Values");
    }
}

private void rdoCaseNumber_CheckedChanged(object sender, EventArgs e)
{
    try
    {
        selectCriteria.criteria(this.rdoCaseNumber, this.rdoName, this.txtName,
this.txtCaseNumber, this.searchCriteria);
    }
}

```

```
        catch (Exception ex)
        {
            GlobalClass.DisplayMessageBox(ex.Message, "Loading Values");
        }
    }

    private void btnLoad_Click(object sender, EventArgs e)
    {
        try
        {
            loadChildGrid.loadGrid(this.value, this.searchCriteria,
            this.rdoCaseNumber, this.rdoName, this.txtCaseNumber, this.txtName, this.dgvCaseDetails);
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}
```