

Hybrid Revenue Data System Analysis for Regal Insurance Zimbabwe



DE-VILLIERS NYAHUMA

R158520R

Hybrid Revenue Data System Analysis



By

DE-VILLIERS NYAHUMA

Submitted in partial fulfillment of the requirements for the degree of

BSc Honors Information Systems

Department of Computer Science and Information Systems

in the

Faculty of Science and Technology

at the

Midlands State University

Gweru

October 2017

SUPERVISOR:

MR MAMBOKO

Abstract

The purpose of this study is to come up with a system that allows online hybrid sales analysis. The main contributing reason for the development of this system is that the company has no particular system which is used for predicting sales. The research main objectives were to make a system which permits compilations of agents sales and calculating the revenue that have been collected at a particular time. Different alternatives were looked at and these include improving the current system, out-sourcing and in-house development for the solution. It has been noted that an in-house is the best idea because of its cost effectiveness and minimum risk associated. Selection of the alternative led to the beginning of the functional and non-functional requirements for the development of the system. Feasibility study was done to determine if it is feasible to develop the system after considering different types of feasibility study techniques. After considering these techniques, it has been noted that it was viable to continue with the development of the system and this has triggered data gathering. Different methods have been used to gather the data. Methods include interviews, questionnaires and observations. After having the data at hand, development started using PHP, MySQL and Dreamweaver platform. The system was tested using unit, module validation and verification tests. After testing the system was implemented using a parallel run method because of its advantages of comparing the data from both systems thereby being able to identify errors of the system. Maintenance procedures have been addressed which includes adaptive, corrective and preventive maintenance. It has been recommended to apply the three of them to the system.

Declaration

I De-Villiers Nyahuma, hereby declare that I am the sole author of this dissertation. I authorize the Midlands State University to lend this so that other institutions or individuals for the of scholarly research.

Signature _____

Date _____/_____/_____

Approval

This project entitled Regal Insurance Company done by De-Villiers Nyahuma meets the regulations governing to approval to proceed with Bachelor of Information Systems Honors Degree of the Midlands State University and is approved for its contribution to knowledge and literally presentation.

Supervisor : _____

Signature : _____

Date : ____/____/____

Acknowledgement

I would like to give thanks to friends and family for their moral support during the development of this project. Moreover, I want to acknowledge the great help from Mr. Mamboko for his support and guidance during the project development, many are the academic principles I have acquired in this period through him. Lastly and above all, many praises to the almighty God for the strength and enablement to see this project into completion.

Table of Contents

Abstract	i
Declaration.....	ii
Approval.....	iii
Acknowledgement	iv
Table of Contents.....	v
List of Tables	x
List of Figures.....	xi
CHAPTER 1: INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Organizational Background	1
1.3 Organizational Structure.....	2
1.3.1 Vision.....	3
1.3.2 Mission Statement	3
1.4 Objectives	4
1.5 Chapter Summary.....	5
CHAPTER 2: PLANNING PHASE	6
2.1 Introduction.....	6
2.2 Business value.....	6
2.3 Feasibility Study.....	7
2.3.1 Technical Feasibility	7
2.3.1.1 Hardware and software requirements for server	8
2.3.1.2 Hardware and software requirements for Workstations	8
2.3.1.3 Networking equipment	9
2.3.2 Economic Feasibility	9
2.3.2 (a) Development and Initial Costs.....	9
2.3.2(b) Operational costs:	10

2.3.2 (c) Benefits:.....	10
2.3.2.3 Cost Benefit Analysis (CBA).....	12
2.3.2.4 Return on Investment (ROI)	12
2.3.2.5 Payback analysis.....	13
2.3.2.6 Net present value (NPV).....	14
2.3.3 Operational Feasibility	15
2.3.4 Social Feasibility	15
2.4 Risk Analysis	16
2.5 Conclusion for Feasibility.....	17
2.6 Stakeholder Analysis	17
2.7 Develop Work Plan	18
2.7.1 Project schedule.....	18
2.7.2 Gantt chart.....	19
2.8 Chapter Summary.....	19
CHAPTER 3: SYSTEM ANALYSIS	20
3.1 Introduction.....	20
3.2 Data Gathering Methodologies	20
3.2.1 Interviews.....	20
3.2.1.1 Advantages of Interviews	20
3.2.1.2 Disadvantages of Interviews	21
3.2.2 Observations.....	22
3.2.2.1 Advantages of Using Observations	22
3.2.2.2 Disadvantages of using observations.....	22
3.2.3 Questionnaires.....	23
3.2.3.1 Advantages of questionnaires	23
3.2.3.2 Disadvantages of using questionnaires	23
3.2.4 Document review	23

3.2.4.1 Advantages of document review	24
3.2.4.2 Disadvantages of using document review	24
3.3 Analysis of present / existing system	24
3.3.1 Insurance Policy Sales	24
3.3.2 Revenue remittance	24
3.3.3 Data storage	25
3.3.4 Reporting and analysis.....	25
3.4 Process Analysis.....	25
3.4.1 Activity diagram of the current system	25
3.5 Data Analysis	26
3.5.1 Context diagram	26
3.5.2 Data flow diagram	27
3.6 Weaknesses of the existing system	29
3.7 Evaluation of Alternatives	29
3.7.1 Out-sourcing	30
3.7.1.1 Disadvantages of outsourcing	30
3.7.2 Improving the existing system	30
3.7.2.1 Reasons for not choosing or taking this alternative	31
3.7.3 In-house development.....	31
3.7.3.1 Advantages of In-house development	31
3.7.4 Selecting the best alternative	31
3.8 Requirements Analysis	31
3.8.1 Functional Requirements	32
3.8.1.1 Use case diagram.....	32
3.8.1.1.1 Insurance sale use case	33
3.8.2 Non-Functional requirements	33
3.9 Chapter Summary.....	35

CHAPTER 4: SYSTEM DESIGN.....	36
4.1 Introduction.....	36
4.2 System Design.....	36
4.2.1 Context Diagram of the proposed System	37
4.2.2 Data Flow Diagram for the proposed system	37
4.3 Architecture design	39
4.4 Physical design.....	39
4.5 Database design.....	40
4.5.1 Database schema	40
4.5.2 Data dictionary	42
4.5.3 Extended Entity Relationship Diagram (EERD).....	43
4.6 User Interface design.....	44
4.6.1 Log in design.....	44
4.6.1 Menu design.....	45
4.6.3 Input design.....	46
4.6.4 Output design	47
4.7 Pseudo code	48
4.8.1 Package diagram	50
4.8.2 Class diagram.....	51
4.8.3 Sequence diagram.....	52
4.9 Security Design	53
4.9.1 Physical Security	53
4.9.2 Network Security.....	54
4.9.3 Operational Security	54
4.9.4 Backup Design	54
4.9 Conclusion	55
CHAPTER 5: IMPLEMENTATION.....	56

5.1 Introduction.....	56
5.2 Coding	56
5.3 System Testing	56
5.3.1 Module testing.....	59
5.3.2 Sub system testing	60
5.3.3 System testing	60
5.3.3.1 System versus Objectives	61
5.4.1 Validation and verification	64
5.4.1.1 Validation.....	64
5.4.1.2 Verification	64
5.5 Installations	65
5.4.3 User training.....	65
5.4.4.2 .Parallel Operation:.....	67
5.4.4.3 Pilot Operations	67
5.4.4.4 Phased operation:	68
5.4.4.5 Recommended strategy.....	69
5.5 Maintenance	69
5.5.1 Corrective Maintenance.....	69
5.5.2 Adaptive Maintenance	70
5.5.3 Perfective Maintenance	70
5.5.4 Preventive maintenance	70
5.6 Conclusion	71
References	Error! Bookmark not defined.
APPENDIX A.....	72
APENDIX B	
APPENDIX C	
APPENDIX B: SNIPPET OF CODE	

List of Tables

Table 2.1 Hardware and software requirements for server

Table 2.2 Hardware and software requirements for users

Table 2.3 Networking equipment

Table 2.4 Estimated development and initial costs

Table 2.5 Operational costs

Table 2.6 Tangible and non-Tangible benefits

Table 2.7 Tangible benefits

Table 2.8 Cost benefit Analysis

Table 2.9: Risk Analysis

Table 2.10: Project schedule

Table 3.1 Evaluation of alternatives

Table 4.1 System users

Table 4.2 Policy

Table 4.3 Agent details

List of Figures

Figure 1.2

Figure 2.2 Calculated discounted cash flows and NPV.

Figure 2.3 Gantt chart

Figure 3.1 Insurance Sales activity flow diagram.

Figure 3.3 Context diagram for the existing system.

Figure 3.4 Data flow diagram for the existing system

Figure 3.5 Cash remittance use case

Figure 4.1 Context diagram for proposed system.

Figure 4.2 Data flow diagram for proposed system

Figure 4.3 Architectural design

Figure 4.4 Physical diagram

Figure 4.13 Database Architecture.

Figure 4.14 Extended Entity Relationship Diagram

Figure 4.5 Log in form

Figure 4.6 System Main window

Figure 4.7 policy transaction form

Figure 4.8 Adding new Agent

Figure 4.10 Package diagram

Figure 4.11 Class diagram

Figure 4.12 Sequence diagram

Figure 4.16 Backup design

Figure 5.1 System testing process

Figure 5.2 White box testing

Figure 5.3 Module testing

Figure 5.4 Preview of system objective 1

Figure 5.5 Preview of system objective 2

Figure 5.6 Preview of system objective 3

Figure 5.7 Preview of system objective 4

Figure 5.8 Illustration of system validation

CHAPTER 1: INTRODUCTION

1.1 Introduction

Advances in digital interactions, computation, and storage have created massive collections of data, capturing data and information of value to different sectors in the country. For example, search engine organizations such as Yahoo, Amazon, Microsoft and Google have formed an entirely new industry by capturing the information without restraint available on the World Wide Web and offered it to public in useful ways. These organizations collect trillions of bytes of data every single day and frequently add new services in retrieving the same information stored. Just as search engines have promoted e-business, insurance companies and banks have shifted to e-business and e-commerce and this has been transforming the activities of retail businesses and insurance. In Zimbabwe where the government has regulated that all insurance companies must trade online on a transaction platform called ICEcash, an electronic platform capable of handling more than 1 terabytes (1 billion bytes) of records, representing every single purchase record by Agents (around 150 thousand transactions per day) countrywide. Furthermore applying machine analysis to this information, companies can detect the sequence signifying the effectiveness of their price strategies and promotion campaigns, and enhanced their sales chains.

1.2 Organizational Background

Background of Organization

Regal Insurance Company (Pvt) Ltd is a Zimbabwean owned short-term Insurance Company with the Central Offices in city of Harare and 20 other branches within other cities in the country. It is fully registered in terms of the Insurance Companies Act with the Ministry of Finance and opened its doors in 2002.

Regal Insurance employs over six hundred (600) agents in its nationwide. The main business is insurance underwriting in classes of business inclusive of Personal Insurance, Business Insurance, Professional Indemnity, Motor Insurance, Engineering Insurance, Marine and Farming Insurance

Personal insurance this is a kind of insurance on individual bases rather than organisations or companies and covers or provides financial security of life and your family in case an

individual faces illness or injury or permanent disablement or death. It is also further divided into health insurance and life insurance. Health insurance covers the medical facilities of an individual and in case of death the family will get some money as a lamp some of what has been insured for by the individual to prevent the family to have poverty in their home. Life insurance covers the case of death, if a family depends on the individual who has insured. To prevent poverty, they will get a total package that has been insured for at the end of the year

Business insurance this protects the business from loses due to unexpected events during normal course of operations. They are different types of business insurance which are employee related risks, legal liability and property damage. Motor insurance so called automotive insurance this is an insurance that the insurer assumes that the risks of any loss the owner of the vehicle may damage the property as a result of an accident or theft .

Engineering insurance is an insurance that provides and safeguards the ongoing constructions, involving machines and equipment in the operations of the projects. Marine insurance covers the loss or damage of ships, property and terminals which are carried from and to destination. Farm insurance this is more of the needs of the farmer covering the equipment, crops, live stocks and property whether the land is being rented or owned. Furthermore it depends on whether it is a small or large scale farmer

1.3 Organizational Structure

Organizational structure is a term used to define a platform within an organization and how it is organized. Every job is identified, its purpose and where it reports to in the group company. According to Cronje (2002) an organizational structure is the framework, structure which defines the lines of authority within an organization. The organogram below describes the hierarchy of the company from the management to the operational team, it also entails how information flows in the organization. It defines roles that each individual will play in the company according to their qualifications. Below is the organization structure of Regal insurance (Pvt) Ltd.

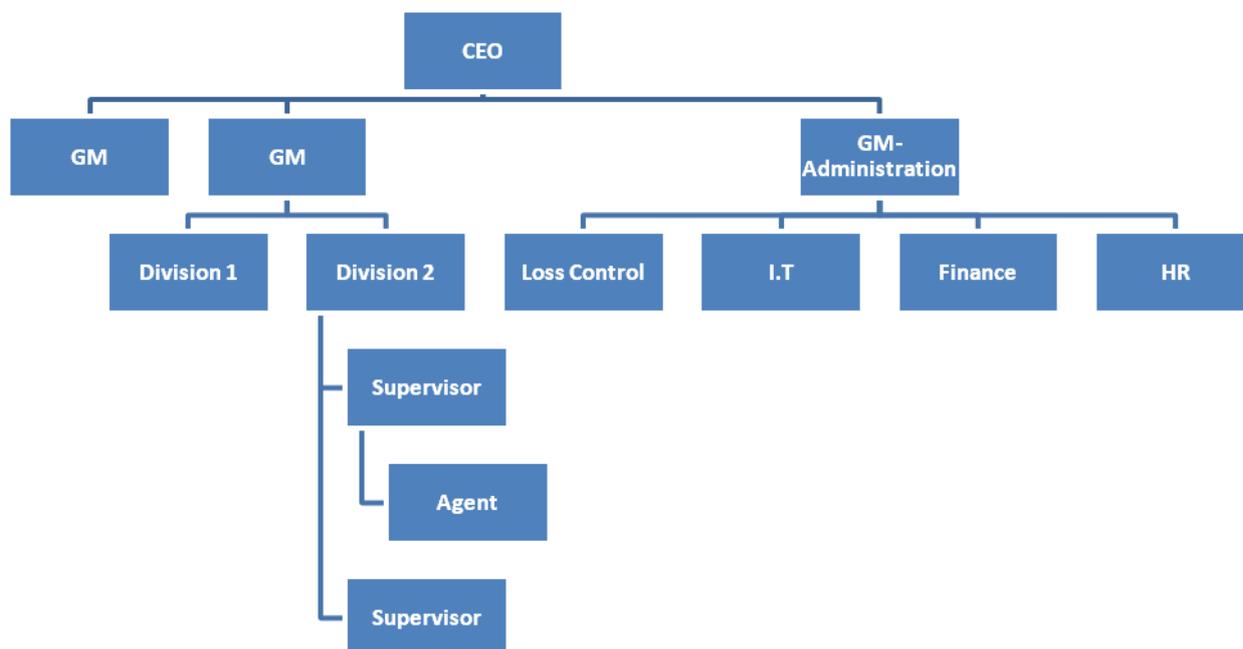


Figure 1.2 Organization structure

1.3.1 Vision

(Webster, 1992) a vision is a thought, objective, concept and is formed by the imagination and a description of how an organisation will be like in midterm and long term life span of an organisation. (Swift, 1954) suggested that a vision is the art of seeing things invisible. A vision is a mental picture depicted to anticipate future events and position of achievement on speculated goals. This gives an organisation a clear picture of where they are headed to.

To be the preferred provider of short term insurance solutions

1.3.2 Mission Statement

(Gibson, 1992) defines mission statements as a written declaration and a statement which is used to facilitate and communicate the organisation core purpose. (Shelly, 2016) suggested that, a mission statement outlines the purpose and direction for an organisation's activities and is the foundation of strategic planning.

Our mission is to provide innovative short term insurance and related services that are based on sound insurance principles.

Problem Statement

Sales analysis and forecasting is one of the most difficult areas of management, where a lot of experience, knowledge and specialized tools are required for accurate predictions. The main problem with a lot of companies in Zimbabwe is that they cannot afford to invest in technology for example SAP and Oracle Business Intelligence (BI) that will enable data analytics and this continues to pose a high risk of revenue leakages and loss. For many medium-big insurance companies, Regal Insurance being one of them, machine data analysis remains a nightmare and the recurring problem will cripple company growth. As a result of using inefficient data analytics ways, all revenue from Agents is not remitted in time and about 8% cannot be traced after a week. To address this problem, the goal is to design and develop a sales data analysis system which will help the company to have a deeper insight of sales, help to calculate profits and make better investment decisions. The implementation of this project will help the company to improve the methods of retaining markets, and target new markets so as to reach maximum efficiency in collecting revenue from Agents.

Aim

The main aim of the project is to create, design and build up a cost effective sales data analytics system that is capable of visualizing sales and predicting future sales for better decision making.

1.4 Objectives

To develop an automated system that:

- i) To enable machine analysis of history revenue data for decision-making.
- ii) To be able to predict future revenue based on pattern revenue inflow.
- iii) To enable program to be capable of random and custom revenue analysis as per agent.
- iv) To enable system to calculate revenue made by a single agent.
- v) To enable the data to be turned into insight.

Methods and Instruments

While aiming at coming up with a web based compliance database that will be used for communicating via an interface with Safety officers, Medical staff, middle and senior level management, with different backgrounds and information technology capabilities, the writer has proposed the following tools be used in developing the database:

- ❖ Programming language: PHP 5 or later. PHP is an advanced programming language that enhances interactive platform or interface and supports dominant database
- ❖ MySQL – an open source database server with a client/server implementation. It is fast, robust, easy to use and has the capability to store very large volumes of data. It is also designed for power, speed, and precision in critical task, heavy load use (MySQL, 2002,)
- ❖ Web Server: Apache 1.3.23 or later
- ❖ Operating system: Microsoft Windows 7 or later
- ❖ Database Management system tools for administrator: phpMyAdmin
- ❖ Querying language : SQL

Justification

The proposed Hybrid Sales Analytics assist to obtain value from the future savings or investments to be made in the analytics systems by giving front line servant leaders the data and information necessary to effectively supervise their Sales Agents. With the algorithmic content to be built available in this area, managers would be able to identify areas that need perfection, speed up the use in business and drive alignment. Through the insight gained, management can deepen Agent interaction by obtaining clear view of sales performance and this can avoid over-expectation and underperformance.

1.5 Chapter Summary

In this chapter, it has given us a background of Regal Insurance Company. Furthermore it elucidates the problem definition, aims, objectives and justification of the hybrid system. In the entire industry, sales companies are informed to work with more efforts yet less, by making sure they reach higher aggressive revenue reports and analytics were depending on pipeline and sales management. This new system aims to give revenue management with aggressive higher levels in the business through revenue format by opening up all the data hidden in the organization's whole system. More so with unlimited access to data that drives them to higher client appreciation and satisfaction at the lowest cost but high level revenue inflow.

CHAPTER 2: PLANNING PHASE

2.1 Introduction

For the purpose of this project and in context of business, this chapter will focus on ascertaining of business and determining the feasibility of the project. The analysis entails reasons as to why the project should be developed as well as paving the way forward for the development of the system. If the proposed system is not feasible, then the project will be aborted to avoid loss of resources. Proper planning together with good resource mobilizations will result in a successful feasibility study.

2.2 Business value

(Berman, 2007) defines business value as merely expenditure, reduction expansion of the business, maintaining operations, efficiency and speed. More so (Broadbent, 1998) emphasized on program/system accessibility, costs per business dealing and work station cost and times to implement the operational and system performance of the business. According to (Badiru, 2003) business value refers to the success of an organisation in using information to achieve its strategic objectives. This side of the organisation strategy to compromise the development and declaration of shared views of the business route and gain to be entailed overall from the business viewpoint.

Business value is net benefit that will be realized by the customer of a project, and can be measured in either monetary or non-monetary terms. It focuses mainly on clearly outlining measurable benefits over the costs, which will be accrued if the system is implemented successfully. An essential part of realizing the desired business value is having the right tools applied to drive quality and cost efficiency. The expected benefits from the project include:

- ❖ Increase forecasting accuracy and sales results when sales executives get up-to-the-minute information so they can accurately assess progress against sales goals
- ❖ Leveraged analytics - consistent data analysis and reporting of results provides a well-defined, standardized, and validated means of retrieving, calculating, and reporting data that eliminates transcription errors
- ❖ Reduced cycle time to perform data analysis. Time savings are frequently realized both in the time it takes to generate each report, and reviewing the results.

- Increase sales force effectiveness- fact-based insight into the effectiveness of the sales agents can help to take appropriate actions to ensure optimal team performance

2.3 Feasibility Study

(Mypoulos, 2005) in their research noted feasibility study as an official systems document which displays analysis outcomes, investigation, assessment for the system under research and concludes whether it is cost effective, profitable, technically feasible or not.. According to O. Brian (2006) defines feasibility study as to evaluate the viability of an idea, it is a logical assessment to determine if a project will be successful. (Killer, 2007) characterized possibility study as an investigation of the viability from claiming a thought concerning administrations more items. The clue may be to choose if an association ought try ahead and execute the undertaking or not.

Results from the analysis are used in defining if there is any presence of economic viability in developing the system under research and proceed to the implementation plan. A project feasibility in terms of technical and economical requirements, stakeholders' support and operational support are made and whether the benefits of the system are proportionate to the costs.

2.3.1 Technical Feasibility

According to (Whitten, 2007) clearly states whether the system can be developed prearranged improvements limitations in terms of all tools required and time between all the other rest. (Rosenblatt, 2011) defined Technical feasibility is performed to assess the expediency of the proposed solution and its technological requirements for setting the new system. It also serves to identify the appropriate hardware and software requirements for the development and implementation of the proposed system

According to feasibility analysis procedure the technical feasibility of the system is analysed and the technical requirements such as software facilities, procedures, and inputs are identified and this helps to determine if new technology is needed and if current technology can support it. The technical requirements of the proposed system were assessed with regards to the availability of the following:

❖ **Technical expertise**

The expertise with reference to the development of a system as well as with the help of project supervisor, it will be possible to develop a running system. The prospective end-users will be well trained on how the system works and all of them have the potential to become adept within the shortest possible time.

❖ **Hardware and software requirements**

At least Windows 7 Operating System or later is the recommended environment as these versions appropriately support networking. The analysis carried out indicated that the organisation is in possession of the recommended hardware and much of the software and this makes the solution feasible.

2.3.1.1 Hardware and software requirements for server

The table below summarises the hardware and software specifications.

Table 2.1 Hardware and software requirements for server

Hardware requirements
Intel core i5 @ 3.0 GHz Processor
4 GB RAM
1 TB HDD
Software requirements
Apache Web server
MySQL Database server
phpMyAdmin database management tool
PHP programming language

2.3.1.2 Hardware and software requirements for Workstations

Table 2.2 Hardware and software requirements for users

Workstations Hardware Specifications
Intel core 2 duo @ 2.0 GHz Processor
4 GB DDR3 RAM
500 GB HDD
Microsoft Windows 7 or later
Microsoft office suite 2010 or later
Microsoft Edge 10 / Mozilla Firefox 36 or later web browser

2.3.1.4 Networking equipment

Table 2.3 Networking equipment

Networking Specifications
12 port Ethernet Switch
Router and Firewall
UTP cables CAT 6/ Wifi

2.3.2 Economic Feasibility

According to (Rosenbalt, 2009) they said it is the predictable gain of the future system that umbrellas the approximate cost and is considered the total cost owning and continue to support and is an ongoing maintenance costs and acquisition costs. (Rosenblatt, 2011) a project is economically feasible if its benefits outweigh its costs. Economic feasibility measures the cost-effectiveness of a project or solution For the purpose of this project, a cost-benefit analysis as well as the payback period will be some of the ways of evaluation.

2.3.2(a) Development and Initial Costs

The following costs were incurred while coming up with the system:

- i. **Initial Costs:** These are costs associated with starting-up the new computerized solution including acquiring needed hardware and software.
- ii. **Development costs** On this part outlined the costs that are incurred during the development of the system for example salaries to systems development team and training costs.

Table 2.4 Estimated development and initial costs

Description	Quantity	Unit Costs \$	Total Cost \$
Hardware			
Pentium 4 Lenovo Desktop set	10	600	6000
Main Server	1	1500	1500
Switch	2	400	800
Backup Media	1	300	300
HP laser Jet colour printer	2	300	600

Software			
Full Dreamweaver CS5 package	1	--	--
Windows 2008 R2 Server	1	550	550
Microsoft Office 2010 suite (<i>sunk cost</i>)	1	300	300
MacAfee Antivirus	10 users	500	500
Labour- Installations	-	2000	2000
User Training	-	1200	1200

2.3.2(b) Operational costs:

These are costs incurred after the system is implemented and continue to be incurred while the system is in use. These include system maintenance costs, annual licensing costs and compliance costs among others. The below table illustrates the operational costs for the period of the coming three years.

Table 2.5 Operational costs

Description	Year 1	Year 2	Year 3	Total
	\$	\$	\$	\$
System maintenance	1,000	1,200	1,500	3,700
Training costs	1,200	600	500	3,300
Stationary and consumables	1,000	1,000	900	2,900
Conversion Costs	1,000	-	-	1,000
Licensing fees	500	500	500	1,500
Sundry costs	2,000	1,500	1,000	4,500
Total for each year	6,700	4,800	4,400	15,900

2.3.2 (c) Benefits:

The following benefits will accrue to the firm due to the implementation of the proposed project.

i. Tangible Benefits:

These are benefits which a monetary value can be attached.

ii. Intangible Benefits

Intangible benefits are those that are related to the qualitative aspect which the system is expected to produce and these cannot be easily measured in monetary terms.

Table 2.6 Tangible and non-Tangible benefits

Tangible Benefits	Non-Tangible Benefits
1. Provide information to all level management through comprehensive and flexible reports	i) A better corporate image.
ii) Upgrades in the upgrade for data integument on every one clients through the utilization of secure international ID approach.	2. Improved work performance.
3. Maximise resource utilisation and profitability	
4. Better focusing accuracy and revenue results	
5. Improvement in agent responsiveness based on revenue inflow for each agent	

The following table illustrates tangible benefits.

Table 2.7 Tangible benefits

Benefits	YEAR 1	YEAR 2	YEAR 3	TOTAL
	\$	\$	\$	\$
Reduction in wages and salaries	15000	15000	17000	47,000
Reduction in agent expenses	6000	7500	8000	21,500

The usual quantifiable feasibility indicators used in the economic analysis are the cost-benefit analysis, net present value and payback period.

2.3.2.3 Cost Benefit Analysis (CBA)

Brent (2006) said that CBA will be those present worth of constantly on profits short of what everyone costs, subject will specified imperatives.. Cost–Benefit Analysis (CBA), sometimes called benefit–cost analysis (BCA), is a systematic process for calculating and comparing benefits and costs of a project.

Table 2.8 Cost benefit Analysis

Year	YEAR		YEAR		YEAR		Total
	1		2		3		
Benefits	USD		USD		USD		
Tangible	21,000		22,500		26,000		
Total benefits		21,000		22,500		26,000	69,500
Costs	--		--		--		
Operational costs	20,400		4,800		4,400		
Total costs		20,400		4,800		4,400	29,600
*Net benefits		600		17,700		21,600	39,900

**Net benefits = Total benefits - Total costs*

Cost Benefits Evaluation Techniques

There are three methods that are used to assess CBA of the project development and justification is prepared as to necessity of the project and is necessary to be carried out in the organisation. The techniques are Return on Investment, Net Present Value and Pay Back Period.

2.3.2.4 Return on Investment (ROI)

According to (Lucey,2011) Return on Investments' are the gains that are realised through the measuring of what has been gained against what has been invested. Return on Investment can be defined as an execution measure used to assess the effectiveness of a speculation or with think about the effectiveness of a number about diverse ventures. With ascertain ROI, those profit (return) for a speculation may be separated by those expense of the investment;

those outcome will be communicated Concerning illustration An rate alternately An proportion.

$$\text{ROI} = \frac{(\text{Average net benefits} - \text{Total costs})}{\text{Total Development costs}}$$

Average net benefit, refers to the benefits gained from a deal. If the total invested revenue does not meet the cost and producing a constructive ROI then it has to be discarded and for look for other opportunities with a positive ROI

ROI for the project (for financial year 2017) is:

$$\begin{aligned} \text{ROI} &= \frac{(\$39900 - \$29600)}{\$29600} \times 100\% \\ &= \underline{\underline{34.79\%}} \end{aligned}$$

After the calculation of ROI, it has a positive value so the project is worth investing in, but if it had a negative value it would not have been worth investing but to opt for other alternatives.

2.3.2.5 Payback analysis

Payback period according to (Schwable, 2006) describes it as a period and the total time taken to recover, in terms of revenue inflow. According to (Tanenbaum, 1990) defined payback time as the amount of time it takes to break even or to settle up the primary investment. A payback (PBP) period is the time a project can generate cash to recoup its initial outlay. If a deal doesn't contain a short PBP, or if there is another option with a shorter PBP, there for the deal not be taken into consideration. From the PBP analysis below, the project will need less than one year to payback for itself entirely and this makes it worthwhile to continue with the project.

Year	Cash flow (USD)	Cumulative balance (USD)
0	(13,700)	(13,700)
1	600	(13,100)
2	17,700	4,600
3	21,600	26,200

Figure 2.1 Project cash flows

Payback period calculation:

$$\frac{\text{Cost of Project}}{\text{Annual Cash Inflow}} = \text{Payback Period}$$

Payback period = [13,700 / 21000] *12 months

= **7 months 2 week**

2.3.2.6 Net present value (NPV)

Schwalbe (2006) defines Net Present Value as a method of calculating the expected net monetary gain or loss from a project by discounting all expected future cash inflows and outflows to the present point in time. It is based on view that a dollar today is better than a dollar to be received the next year. This is an assessment method which takes into description of the prosperity of the investment and the time frame of the ready money flowing from that which is produced by discounting future cash flows with a percentage called discount rate. If NPV is negative the project should be discarded, but it is positive, then it is worthwhile to continue with the project. The present value of any future cash flow is obtained by applying the following formula:

$$\text{Present Value} = \frac{\text{Value in year } t}{(1 + r)^t}$$

There for “r” is discount rate

‘ t ’ is the numeral value of years into the potential period revenue inflow .

Year	Cash flow (US\$)	Discount factor @10%	Discounted cash flow (USD)
0	(13,700)	1.000	(13,700)
1	600	0.9091	545.46
2	17,700	0.8642	15,296.34
3	21,600	0.7513	16,228.08
NPV			\$ 18,370.14

Figure 2.2 Calculated discounted cash flows and NPV.

The project has a positive net present value so it is worth investing into.

2.3.3 Operational Feasibility

Kendal (1996) postulates that operational feasibility refers to the evaluation which analyses the willingness and ability of stakeholders to operate the proposed system and how well the system operates. According to (Hughes, 1994) defines operational feasibility is concerned with whether business is current operating environment is capable of supporting the new, be consistent with the activities needed by the system.

This part of feasibility study assesses whether the system will be used if it is developed and implemented. Here an evaluation is done on the effects of each alternative on the routine operations of the organization. Management has to be heavily involved in this area of the feasibility study because management understands the operations of the organization. This type of analysis will focus on the changes affecting staff, whether users will accept the changes and to what extent they need to change the way they do business. How the operations will be interrupted while setting up the system and whether any changes in one section of the organization adversely affect another section is again a consideration during this type of feasibility study. With the organization's much placed emphasis on quality occupational health to its workforce, this project has gained management support and more importantly all users accepted the changes to be made and no major interruptions will be experienced during system set up, making this project to be operationally feasible.

2.3.4 Social Feasibility

Social feasibility seeks to assess the impact to the company's society and find out whether it is possible to impact the society positively through the new system (Rosenblatt, 2011). Some social and ethical issues will be reinforced by legislation. Ethical issues to consider are copyright legislation, codes of practice relating to contract management and privacy legislation.

To the community the organization will employ and train the young trainee graduate on how to become an agent for the organization. By introducing the education and training it makes the young man and woman to have focus towards achieving goals. It is going to reduce the unemployment rate in the country and it still continues to employ new trainee agents. This has reduced the crime rates in the country especially communities that have branches of

Regal insurance. Youngsters who drink and take drugs and end up committing crime have reduced since they have something to do for a living. The agencies of Regal insurance have been trying all their best to get as many clients as possible by visiting remote regions and it has been an inspiration the youngsters who are growing up. Furthermore Regal insurance has donated some funds to Falcon seven foundation a charity organization which takes care of the orphans and widows in Gutu area. More so from the profit the organization would have made it will facilitate sports in different region of the country. Promotion on free training to trainee graduates, they do not have to pay any fees but will be well experienced. There for this will reduce the crime rate since most youngsters will be very busy and have something to do for a living.

To a lesser extend this new system is going to make other employees lose their jobs if they don't make any revenue. The unemployment will increase since the system will be making some estimates of hybrid sales. The crime rate will be increased but in a different platform which is a more professional in criminal acts.

The government will also gain from the profits made by the organization in terms of tax and reduced unemployment rate. By reducing unemployment rate it has promoted, this may reduce work for the police force of Zimbabwe and wasting time looking for criminals. The bill for looking for criminals is so high. The government will channel the money towards infrastructure development from the tax the organizations would have paid. This serves time for the rest of the nation but focus on developing the country.

2.4 Risk Analysis

A risk is an event that might occur, and will, if it does occur, affect negatively the successful completion of the project. Risk management is the process of identifying risk, assessing risk, and taking steps to reduce risk to an acceptable level. An effective risk management process is an important component of a successful software project. The table below summarizes the risks associated with the project and solutions to mitigate those risks.

Table 2.9: Risk Analysis

Type of Risk	Severity	Risk Impact	Solution
Data loss	High	Loss of Sales data when server crashes or unauthorised deletions	Regular Data backup, Use of passwords and access
Insufficient expertise and skills within the	Medium	Development of a system with wrong user functions and delays project implementation	Training programs, recruiting individuals with top talent, assigning tasks to key
Shortfall in hardware and software	Low	Delays project Development	Contractual agreements with suppliers, analyses of other organizations performance
Constant change of requirements by users	Medium	There will be an increase in time required to complete the project increase in the scope of the project which will require more days to	User involvement in Requirements specification, incremental prototyping
Computer viruses	Medium	Delays the development as viruses may affect the code	Use updated Antivirus software

Technical Risk

(Mckee, 2001) Pointed out that when new project contain unique technical elements or unproven technology they are being developed under significant technological risk. Furthermore this is the risk that the system may not meet the user's expectation in under the client interface recognizing that those clients might need different sorts for system with altered interface.

2.5 Conclusion for Feasibility

In conclusion on the analyses of the feasibility study of the anticipated system and recommendations has been approved and to continue the progress of this system proposed. Benefit outweighs the initial costs within a short space of time, so the project is worth investing.

2.6 Stakeholder Analysis

In every project, there will be stakeholders who express their different interests and these interest must be analysed so that the stakeholders will; achieve a common goal.

i) The Organization Management

The organization management is wants the project to reduce operational costs and the system to be used as a tool aiding in decision-making. They are saying the project should be cost saving in the long run.

ii) The User/ Customer

The users are concerned with user friendliness and easy to use of the system. Again, they expect the project to speed up their day-to-day operations. They recommend that online help should be provided.

iii) The Developer

Developers typically use the architecture as a reference for developing the system and/or assembling system components. In this project, system developers have highlighted that it must be easy to add some modules if need be since they are concerned with extendibility and maintainability of the system.

i) System Maintainer

The maintainers are mostly concerned with how easy it will be to change the system in the future that can be initiated by new/different user requirements and business requirements. System maintainers highlighted that maintenance should have minimum impact on users.

2.7 Develop Work Plan

A project plan involves the identification of tasks and the allocation of resources to those tasks. The project will follow the system development life cycle and thus will depict a Waterfall Model. The Waterfall Model was chosen as best model because verification is done at each stage, the progress made is easy to trace and its association with stringent documentation which increases system visibility.

2.7.1 Project schedule

The project schedule describes the dependencies between activities, the estimated time required to reach each milestone and the allocation of people to activities. The total time for the development and allocation for every single phase is below and followed by a Gantt chart.

Table 2.10: Project schedule

ID	Activity Name	Duration	Start	Finish
1	Project Proposal	1 week	Mon 25/02/17	Fri 3/03/17
2	Planning	1 week	Mon 06/03/13	Fri 10/03/17
3	System Analysis	3 weeks	Mon 13/03/17	Fri 31/03/17
4	System Design	3 weeks	Mon 03/04/17	Fri 21/04/17
5	Implementation	1 week	Mon 24/04/17	Fri 28/04/17
6	Maintenance	2 weeks	Mon 01/05/17	Fri 12/05/17
7	Project Documentation	11 weeks	Mon 25/02/17	Fri 12/05/17

The activity table was used to produce a Gantt chart shown below, (figure 2.3).

2.7.2 Gantt chart

ID	Activity	Predecessors	Time in weeks													
			1	2	3	4	5	6	7	8	9	10	11			
1	Project proposal	-														
2	Planning	1														
3	Analysis	2														
4	Design	3														
5	Implementation	4														
6	Maintenance	5														
7	Documentation	1,2,3,4,5,6,7														

Figure 2.3 Gantt chart

2.8 Chapter Summary

In this chapter, the reasons for building were discussed together with the business value expected from implementing the system. A feasibility study conducted helps to decide whether to continue with the project or not which at the end the project was deemed feasible. Finally, the chapter highlighted the project plan outlining the start, finish dates and the order of scheduled project tasks.

CHAPTER 3: SYSTEM ANALYSIS

3.1 Introduction

A system analysis can be described as a problem-solving technique that decomposes a system into its constituent pieces for studying how well those component parts interact to accomplish their purpose (Hoffer et al, 2002). This chapter shall discuss the applicable research methods used for the study, how the research was conducted, the findings and goes on to highlight the analysis of the existing system.

3.2 Data Gathering Methodologies

For the purpose of study, a number of different research methods were engaged. There are four methods used which include questionnaires, document reviews, interviews and observations and have been used as tools to collect information of the present contractor workflow running system. The four-phase methodology enabled the analyst to access information about the contractor engagement and management framework from key informants within three domains (Medical section, Projects and SHEQ departments). All forms of information collection were necessary to ensure that the technical report was fully informed.

3.2.1 Interviews

Interviews were conducted with two Finance staff members and others from admin department. The interview data was to compliment the data collected during document review and observation. The key informant interviews were expected to comment on the key components about revenue management system as well as the procedures presently being offered, the part done by a various users, and mechanisms recognized to make sure there is fulfilment in the organisations and also the revenue remittance period to the insurance company.

3.2.1.1 Advantages of Interviews

(Kendell,2002) defines an interview as a data gathering technique in which the researcher getting face to face with targeted interviewee. There are rewards of using the platform of interviews include:

Interviewees are used to obtain different opinions from the interviews and how they feel about the old system they are using, informal procedures and company goals. To some extend opinions are more important than facts. As the analysts dig deeper looking into opinions

much more than facts, he discovers more problems by users that need to be addressed urgently.

The interviews are used as a way of capturing how interviewees feel because they understand the researcher. The analyst allowed the interviewees to explain and give a feedback of what is important in the organisation for them for the purpose of achieving organisation goals.

There are three types of interviews namely structured interviews, semi structured interviews and unstructured interviews. The researcher had the privilege to obtain to conduct all the three types.

Structured Interviews: Appointments were made and face to face interviews of this nature were conducted, one with students and the other with dissertation supervisors. The analyst had written questions well prepared for the interviews. The interviewees were asked closed questions and they did not have any chance to detour the asked questions. This saved a lot of valuable time and the analyst managed to acquire only the necessary data. Given the case that the analyst was a distant from the targeted interviewees, these interviews could otherwise have been conducted using technologies such as telephone calls video conferencing.

Unstructured interviews are a type of interview which is a direct opposite of structured interview. The researcher got into the interview with an unordered set of questions; all the questions he posed to the interviewees were heard generated. Further questions were taken from the responses given by interviews. Interviews were free to provide answers in any way they deem necessary and were not restricted on the amount of data to supply. This help a lot since some interviews provided with some data which could otherwise have been forgotten if it was structured interview.

Semi-structured interviews: This type of interviews is a concatenated of the above the two. The analysts set interview questions but will occasionally detour from them to ask off head. Agents and supervisors as well as project coordinators underwent this interview. Benefits of both structured and unstructured interviews were ripped using this semi-structured interview.

3.2.1.2 Disadvantages of Interviews

- ❖ Not all possible stakeholders were consulted, given the overall project point in time. More so due to shortage of enough time to do as many as possible interviews, the forecaster of the project was unable to solicit the necessary information.

- ❖ Some information could not be disclosed for security reasons and fear of information abuse.

3.2.2 Observations

The observations were carried out by visiting the Finance department and Sales Agents sites, directly observing how they process an insurance policy transaction. These findings were recorded and analysed to determine design criteria. This technique also allowed for a visual opportunity to observe what was currently used in the system to confirm sale, to reconcile total sales of the day or given period. (Steele,1996) postulates that observation is a way of gathering data by watching behaviour, event, or noting characteristics in their natural setting. Observation is a data gathering in which the research is very much involved in viewing, watching, studying, recording and as well as analysing the way the system under observation functions (Blaxter,2006).

3.2.2.1 Advantages of Using Observations

- ❖ Observation qualifies as scientific enquiry when it is specifically designated to answer a research question (Schindler,1998)
- ❖ It was helpful to directly see what sales agents do rather than relying on what they say they do
- ❖ Observations are used to increase data about the assessors and the surrounding environment that is not reachable throughout the other method. The analyst has got a clear picture of what is done and which is not elucidated or documented (Kendell, 2002).
- ❖ These observations were relatively not expensive as compared to different fact-finding methods. This is because other techniques usually require substantially more employees' release time and copying expenses.

3.2.2.2 Disadvantages of using observations

Despite the convenience of observations, some drawbacks of using observations were noted for example system activities took place at odd times, causing a scheduling inconvenience for the system analyst.

- ❖ Preskill(2001) postulates that the responsibilities being looked into were matters to different types of interruptions and it has been very difficult when it comes to time. Many employees usually have feelings and being uncomfortable during observation, they may unwittingly perform differently when being observed. The experiments done by Elton Mayo, the Hawthorne Experiments actually proved this.

3.2.3 Questionnaires

Questionnaires consist of different questions investigating on how the respondents narrate to and control business agents in the company. Survey method was crucial in gathering information from the users of current system and determining the user needs and requirements for the new system. In gathering this information, the team compiled a list of changes that need to be made in integral and technological provisions for the new system. Questionnaires were developed using close-ended questions to increase the response rate.

3.2.3.1 Advantages of questionnaires

(Westbrook,2000) describes that various questionnaires maybe responded quickly and insurer's wider ground. Some other benefits are:

- ❖ Questionnaires are easiest way and means of getting information from a large group of people and were used so as to carry exploratory study and to gauge overall opinion before the systems project is given any specific direction.
- ❖ Reduces chance of evaluator bias because the same questions were asked to all respondents
- ❖ Some people feel more comfortable responding to a questionnaires than participating in an interview

3.2.3.2 Disadvantages of using questionnaires

- ❖ In some of the questions, there were differences in interpretations of questions and this compromised the validity of the information given by the respondents and it has not always been possible to identify all corroborating information. (Westbrook, 2000)
- ❖ Questionnaires tend to be inflexible. There was no opportunity for the systems analyst to obtain voluntary information from individuals or to reword questions that may have been misinterpreted and also data was not easy to interpret.

3.2.4 Document review

The process of document review involves gathering information by checking available documents which may be in-house to a system or company. Checking available documents helps to get clear picture about the philosophy, past and function of the system being evaluated and company in which it is operating. Documents are papers and are hard copies which includes reports, policy documents and books, procedures (cash remittance) and cashing forms. This material was supplemented by information collected through the interview process.

3.2.4.1 Advantages of document review

(Westbrook, 2000) cited some advantages of using document review among them are:

In the organisation, they keep records of every agent and memos send to every branch which made it very clear for the interviewee and thus clearing a better source of deep insight information of the organisation

- ❖ Not clear picture of records of every agent, so by document review it showed that there was some information missing which made it easy for the superiors to see that there is need for this project to be carried out because they could not have an up to date record of every client were he or she on the fields. Coming up of different issues not realised by other means has showed the importance of the system to be developed and this will bring up the behind picture of a program which is not directly observed
- ❖ Data enclosed in current document is separately confirmable by checking with the other copy the agent keeps for record purposes.

3.2.4.2 Disadvantages of using document review

- ❖ Usually data may not be applicable, unavailable, not organized if not incomplete or inaccurate (Kendell, 2002)
- ❖ Accountability and control of data is very difficult and thus depends on the data provided in the files to be assessed on value and source usability.

3.3 Analysis of present / existing system

A thorough analysis of the current system was done in order to have a good foundation of information for input to the proposed project. From the fact findings it showed that the current system being used by Regal insurance was not fully computerised and not integrated.

3.3.1 Insurance Policy Sales

Insurance products e.g passenger insurance and agricultural insurance are currently being sold by manually recording in receipt books. The Agent then receives cash/ Ecocash transfer and the client receives a receipt and policy forms. Only motor vehicle insurance is sold online and the transaction history will be reported the following day.

3.3.2 Revenue remittance

Cash is remitted to the Insurance branch office or can be collected from the agent by the supervisor. The agent may also make a transfer via Ecocash, or swiping on point of sale terminals (POS).

3.3.3 Data storage

Sales records and client details are kept in box files which are used as a database. The excel documents have little sales history for the agents. Data in the hard copies include the daily gross sales by Agents, the commissions and other related sales expenses. The same information may be kept as soft copies in users' desktop computers.

3.3.4 Reporting and analysis

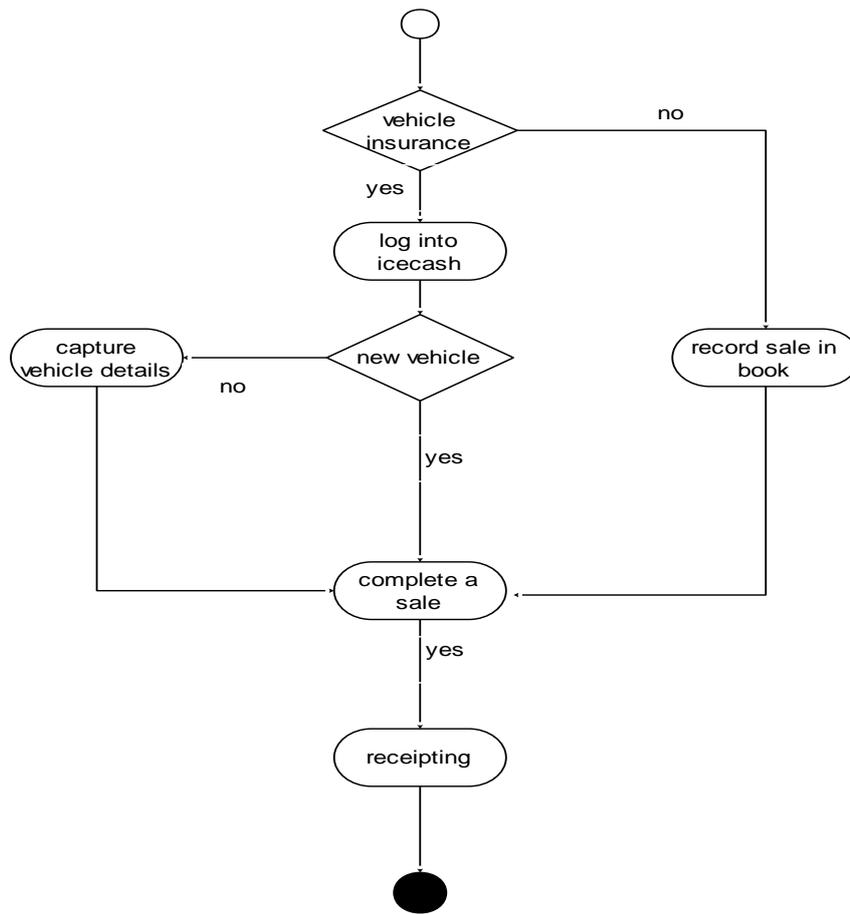
Reports are manually generated from the insurance cover note sales done manually by recording in receipt books. These reports are also filed as hard copies in box files. Only vehicle insurance sales can be analysed by sorting excel file to get a custom report. This is done by uploading the Microsoft excel file into sorting application called Agecon, which then lists Agents according to gross sales. This sales history (report) will then be used to enable collection of revenue from Agents.

3.4 Process Analysis

(Andersen, 1999) defines that a process can be described as a sound series of linked dealings that convert all input being turned into output. Activity diagrams are useful for analysing the processes by explaining the actions taking place. The major explanation to use them is to make sure they replicate the workflow behind the actual system being developed.

3.4.1 Activity diagram of the current system

Activity diagrams are used to describe a complex system in terms of smaller functional components called activities. An activity represents a dynamic process during which objects are manipulated and transformed. The diagram modelled workflow patterns involved from medical examinations for employee at medical centre for either an initial or refresh medical test to induction after which the contractor will then commence work.



	Initial State		Decision		Control of data
	Process/ Activity				Final state

Figure 3.1 Insurance Sales activity flow diagram.

3.5 Data Analysis

3.5.1 Context diagram

Choubey (2012) defines a context diagram as a simplified depiction of the movement of data through the various structures of an information systems in organisation. In order to understand well the system under consideration as a single high-level process showing the relationships it has with other external entities (systems, organizational groups and external data stores), a context diagram will be used. It simply shows the system of interest, the external entities with which it interacts with, and the data flows between the system and the external entities. Below is a typical context diagram of the current system.

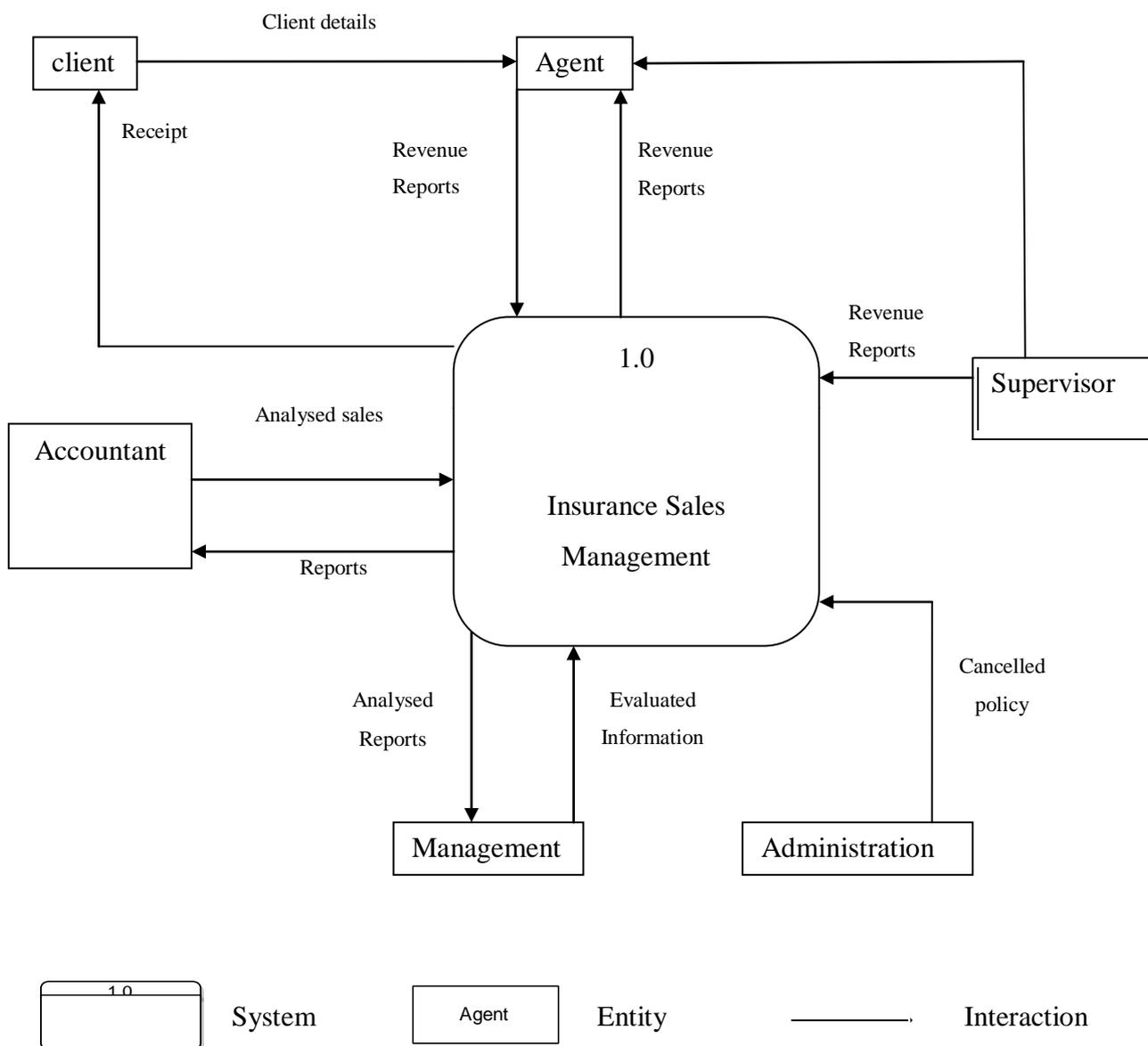
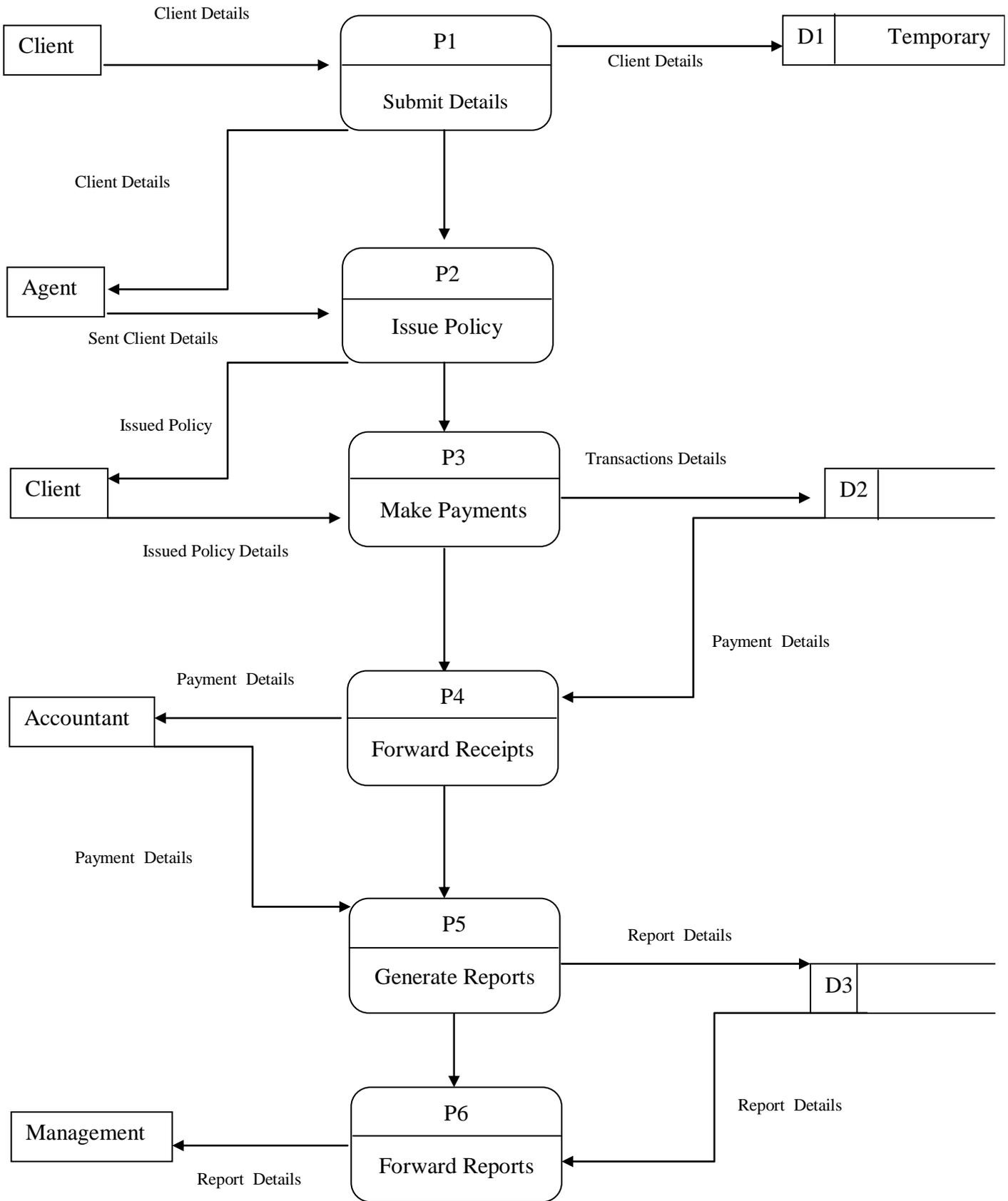


Figure 3.3 Context diagram for the existing system.

3.5.2 Data flow diagram

Data flow diagrams (DFDs) are a way of representing a system's business processes, the flow of data into and out of those processes, and the flow of data between the system and the external agencies with which it interacts. Williams (1998) suggested that a level1 data flow is a diagrammatical presentation of the relationship between the system entities, processes and data stores using data flows to show data being transmitted from one point to the other. The figure below shows the data flow diagram of the existing system.



Key

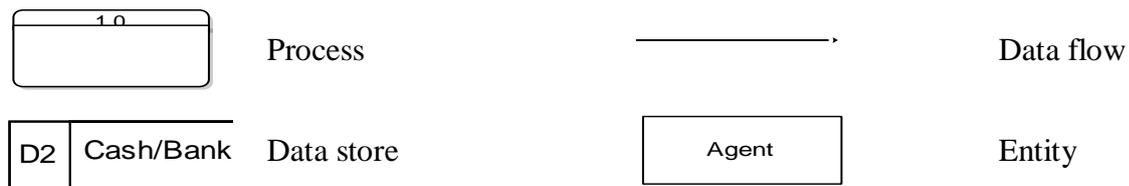


Figure 3.4 Data flow diagram for the existing system

3.6 Weaknesses of the existing system

The current system has quite a number of weaknesses as identified.

- ❖ The manual sale and recording of transactions is prone to many human errors and creates a loophole for fraudulent activities for example reproduction of fake receipt books.
- ❖ Sales information availability depends on how the department keeps it. Some cashing forms are destroyed after the supervisor finishes collecting cash from the agents so the system fails to provide comprehensive sales and remittance history when reference is needed.
- ❖ Tracking of debts (uncollected funds) is very difficult for a very big number of agents as some who make seldom sales may be forgotten and these results in bad debts.
- ❖ Manual analysis of sales history from disparity systems and processes for about 500 agents nationwide is very difficult and cumbersome.
- ❖ Remittance of cash is taking long as some sales may remain unknown until the agents came to declare the sale.

3.7 Evaluation of Alternatives

For the purpose of this study, different alternative project system were proposed and evaluated. The outcome of feasibility study assisted us to select the best and most appropriate alternative that reduces the costs and shows the possible alternative to servant leaders for an absolute decision to be made. The alternatives listed below were taken into considerations (Table 3.1).

- ❖ In-house development
- ❖ Improving the present system
- ❖ Out-sourcing

Table 3.1 Evaluation of alternatives

Alternative	Costs (\$)
Outsourcing	20 500
Improving current system	13 000
In-house development	20 400

3.7.1 Out-sourcing

Marian (2011) suggested that outsourcing is contracting an external developer to handle the project or lead the internal team in developing the system. This is when the company purchase from a confirmed outside trader or involving individuals or organisations who are able to design and administer the technological solution as per specifications.

3.7.1.1 Disadvantages of outsourcing

- ❖ Reliance on vendor’s technical support to resolve issues
- ❖ Risk of exposing confidential data to a third-party
- ❖ These packages are very expensive.
- ❖ The vendor may be simply unqualified for the project or may fail to develop the system that can address the current needs.

3.7.2 Improving the existing system

Alex (2007) defines improving the existing means adding features to the already standing water treatment plant. Another alternative to deal with the present problems was to improve the current manual system. Improving the contractor management system would involve labelling all the files and arrange them in order of dates for example years and records can be indexed for easy reference. Few people would be recruited to capture and maintain the records thereby relieving the work load of the present workforce. A safe room will be needed to store old files than destroying them; this will function as the archive. A procedure to collect cash from agents will be revised so that every agent will remit all funds in the shortest period of time to avoid loss of funds.

3.7.2.1 Reasons for not choosing or taking this alternative

- ❖ This can only be short term solution because as the agents continue to increase, the weaknesses of the current system will tend to resurface in the long run.
- ❖ For this paper system to work better, it requires many workers (division of labour) which will end up increasing costs and without doubt duplication of data.

3.7.3 In-house development

In- house development means the organisation chooses to develop its own technological system from the available resources, or purchase, possibly customize, and implement a software package.

3.7.3.1 Advantages of In-house development

- ❖ Complete control of the solution.
- ❖ Tailored to unique business needs
- ❖ Ownership of the software code
- ❖ Meet constraints of existing systems and existing technology
- ❖ Development team is very close to the end user, resulting in a high quality product for the end user. This results in a much greater “specialised” application.
- ❖ Business knowledge of the development team increases, reducing the time required for specification and design, as knowledge of the problem domain is pre-existing.
- ❖ By using active technology and capital it becomes cost effective when designing the whole system in-house.

3.7.4 Selecting the best alternative

After a thorough evaluation, the option for in-house development was adopted as the best. The most important considerations were total cost of ownership (Table 3.1), satisfying unique business requirements and to make use of the available internal resources and capabilities.

3.8 Requirements Analysis

System requirements are one of the major focus areas which need to be carefully (Summerville,1998). Requirements analysis is the method of expanding the system’s properties and may appear from the arrangement of different sectors for purpose of developing a system which will meet preset user expectations and supplies the basis for rapid and speedy development. Requirements are divided into non-functional and functional needs.

Functional desires are tasks which the system must sustain, whilst non- functional needs are limitations on different attributes of these responsible tasks.

3.8.1 Functional Requirements

The system functionality requirements are those which will specify its function or behaviour (Robertson, 2012). Functional requirements are processes and observable tasks that must be done by the system being development. (Bredemeyer, 2001) explained what the system is supposed to do, functions that are going to be captured in use cases, the behaviours that are going to be analysed by sketch sequence diagrams, most probably trace to single chunks of a system program.

Proposed system should be able to function in some of the following ways:

❖ Online sales and monitoring

The new system should support online sales of all insurance products by agents and allow people in office to view transactions as they happen.

❖ Reports

Display sale reports on a dashboard according to agent, location or region. Reports should gross sales

❖ Sale prediction

A predictive analysis based on sales pattern by region or agent.

❖ Real time Sales analytics

Comprehensive sales data analytics as according to segmented markets by region, or product sold and to enable agent sales profile analysis.

3.8.1.1 Use case diagram

The system's behaviour in respect to the users can be best presented by a use case diagram (Rena-Peraki, 2008). The use case drawing has 4 fundamentals:

- a) The participants, who are individuals or a department. Each participant is connected with different use cases, each one of these will be describing the player wants to do with the system.
- b) Use cases, is actually represented by ovals and their names on it that explains a cycle of behaviour that will be executed (e.g. the doctor can carry out medical test).

- c) The positions that represent types of associations among a player and a use case or involving two use cases. The '<<include>>' relationship, for example, is a fixed relationship among two use cases which is denoted by the invocation of a use case by a different one.
- d) The program borders which is in a rectangle form that includes every use cases which specify all the system's functions.

3.8.1.1.1 Insurance sale use case

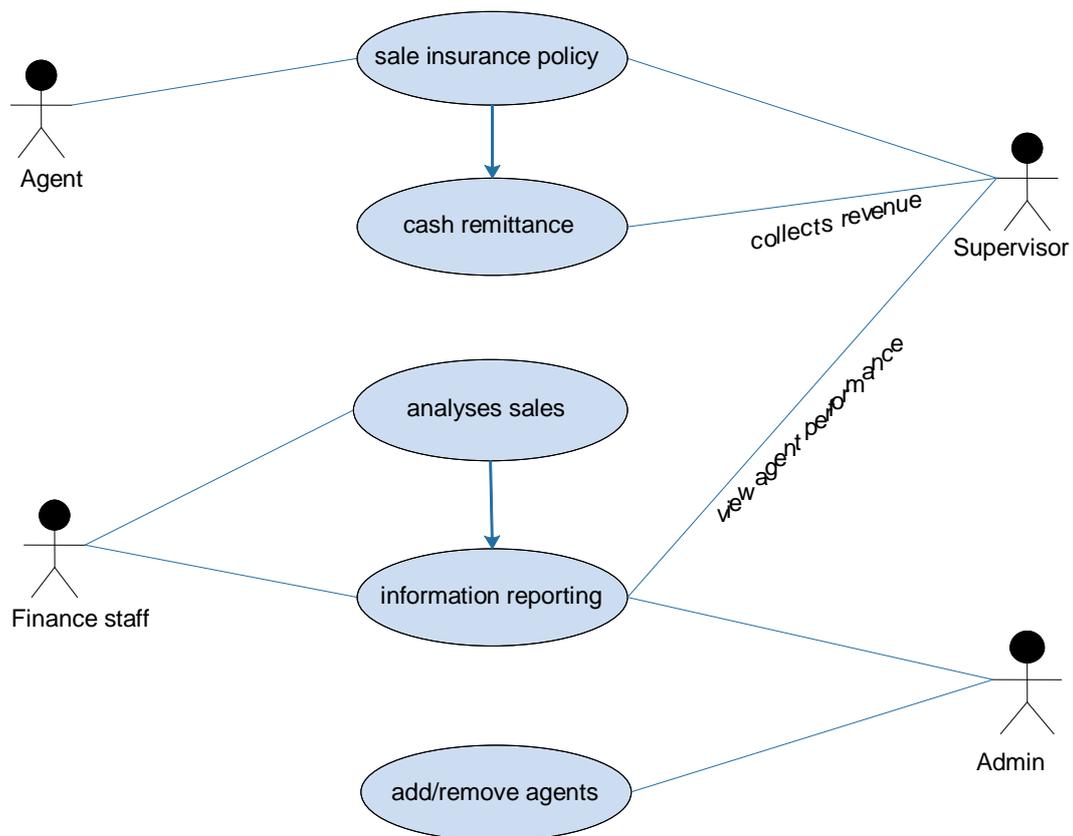


Figure 3.5 Cash remittance use case

3.8.2 Non-Functional requirements

A system's non-functional requirements are those which describe how the system works, thus how it carries out its processes (Robertson, 2012). Non-functional these are requirements that are worldwide limitations on a software, that is, they are intrinsic worth or values that the program being designed must have, but which are not the tasks that are being computed by the program, for example maintainability, development costs, performance, reliability, operational costs, robustness and portability (Bredemeyer, 2001).

Furthermore they usually called software qualities and the following were noted:

❖ **Interface requirements**

The system should be user friendly and a additional function for help assistance. It should be compatible with other systems and it has a Graphical User Interface (GUI).

❖ **Performance requirements**

There must be a simultaneous handling of many transactions, large storage capacity source, through put and excellent response time. Robustness and reliability of information should be available when needed and integrity of data must be maintained by the system.

❖ **Security**

The whole system must be able to withstand catastrophic events like viruses. It must be difficult to hack and an authenticated mechanism, encryption of passwords and rights to access must be for the system users.

❖ **Operating requirements**

When a new system is developed and is too large and it may cause inconvenience when it comes to maintenance. Highly skilled personnel are required to support the system. Characteristics of the new system:

Modifiability: The ability to put more function in the future.

Compatibility: Must be compatible with other programs.

Evolvability: It must support and be capable of exploitation in the modern world technology.

❖ **Economic requirements**

Technological cost: The hardware, software and migration should be in the umbrella of the budget. The system should limit immediate and future long term cost.

❖ **Lifecycle requirements**

Maintainable

Enhancebility

Portable

3.9 Chapter Summary

The chapter highlighted a lot data gathering methodologies and findings that were used for analysis of an existing program. More so it explored different development alternatives, after in house project development, there was the excellent option. The process and information analysis laid out in the chapter, represents the graphical/sketch view of the whole system, its information flow and processes which were used as the foundation of the discussion of requirement analysis and system weaknesses.

CHAPTER 4: SYSTEM DESIGN

4.1 Introduction

This chapter highlights the approach to the creation of the proposed Insurance Sales Analysis System. It provides an understanding and procedural details necessary for implementing this system by defining the architectural, process and user interface design and also that will be required to implement the system. (Rudolph, 2010) postulates, will generate specifications that highlight the actual solution. The system specifications that were stated in chapter 3 will be further improved and allocated into the system and database design specifications. Parts of the system such as the login page must be functional (David, 2009).

4.2 System Design

(Springer, 2015) postulates, the system design refers to the system functionality and how it operates. According to (Whitten et al, 2004) system design must be flexible enough to be altered if the need arises to add features. This is the development of a blue print or the conceptual model of the sales analytics system that shows the hardware and software architecture, components, modules, interfaces, and data for a computer system to satisfy specified requirements. Software design sits at the core of software engineering process. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities –design, code and test that is required to build and verify software. The proposed system will incorporate the following modules:

❖ **Online insurance sales**

Insurance policies for all offerings will be sold online regardless on the geographical place. Every Agent will access the portal via a web in order to sell insurance.

❖ **Revenue remittance**

This module involves online revenue collection and debt calculation for any outstanding amount form the agent.

❖ **Sale prediction**

The system will be able to predict future sales based on trend analyses. This can help to forecast sales and based on market conditions.

❖ **Real-time data monitoring that influences behavior**

Online tracking and analysis of sales will be done anytime by a person from the office. The dashboard will highlight graphically the summaries and any report may be drawn from the statistics. Real-time monitoring and visualization is fundamentally changing the relationship of insurers and the agent. By monitoring their sales, Agents may learn more about themselves, and insurance companies can leverage the data to influence behaviors.

4.2.1 Context Diagram of the proposed System

The context diagram below (Figure 4.1) shows the processes and activities to be included in the computerized system

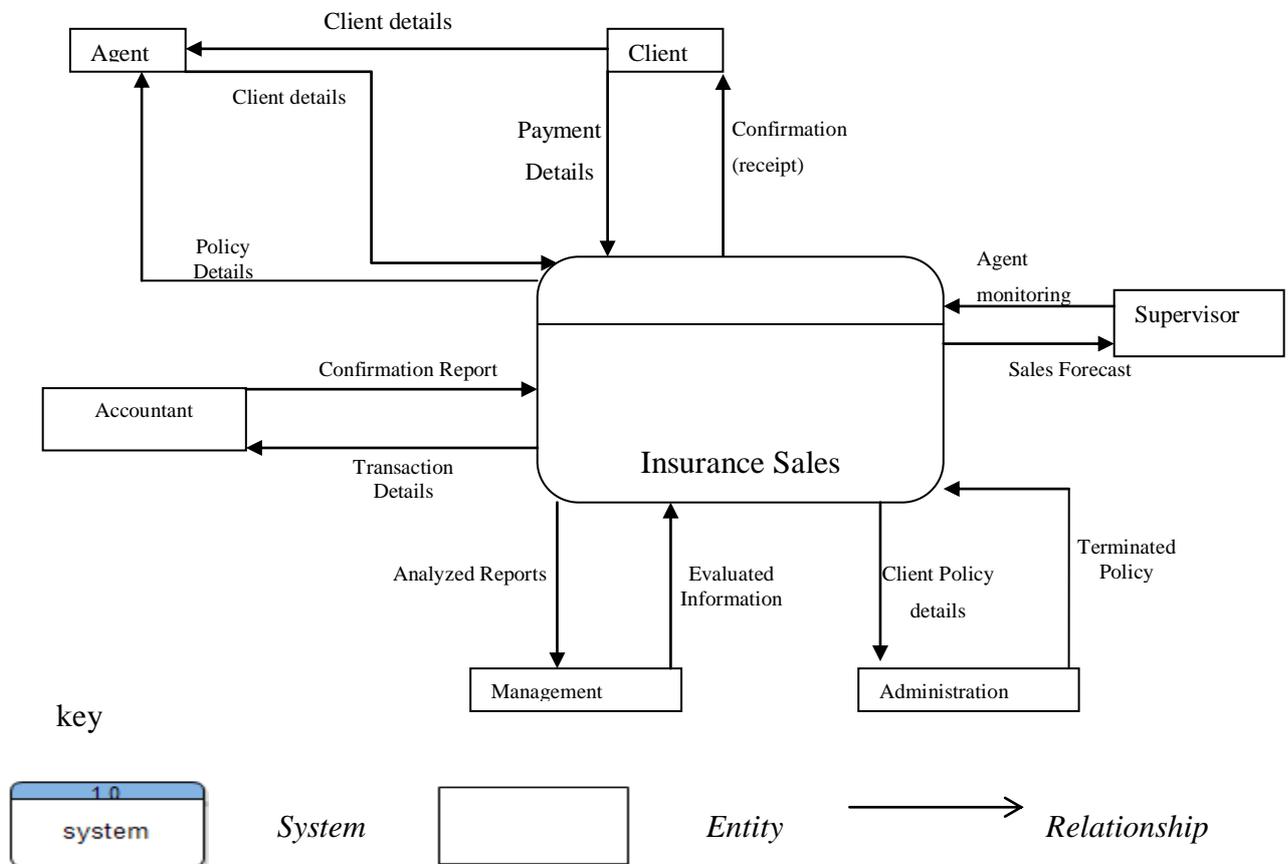


Figure 4.1 Context diagram for proposed system.

4.2.2 Data Flow Diagram for the proposed system

A Data flow diagram (DFD), also known as a bundle chart, is used to express system requirements in a graphical form. Data flow diagram is a diagrammatical presentation and the relationships between processes, system entities and information stores using data flows to reflect data being transmitted from one point to the next (Williams, 1998). A DFD depicts information flow and transform that are applied as data moves from input. Use of DFD helps to clarify the system requirements. This is a diagrammatic representation of data flow within

the system from the agent to the section head of operations. It mainly gives an overview of the system and its functionalities however without getting into much detail (Workinger, 2014). The DFD of the proposed system is shown in figure below (Figure 4.2).

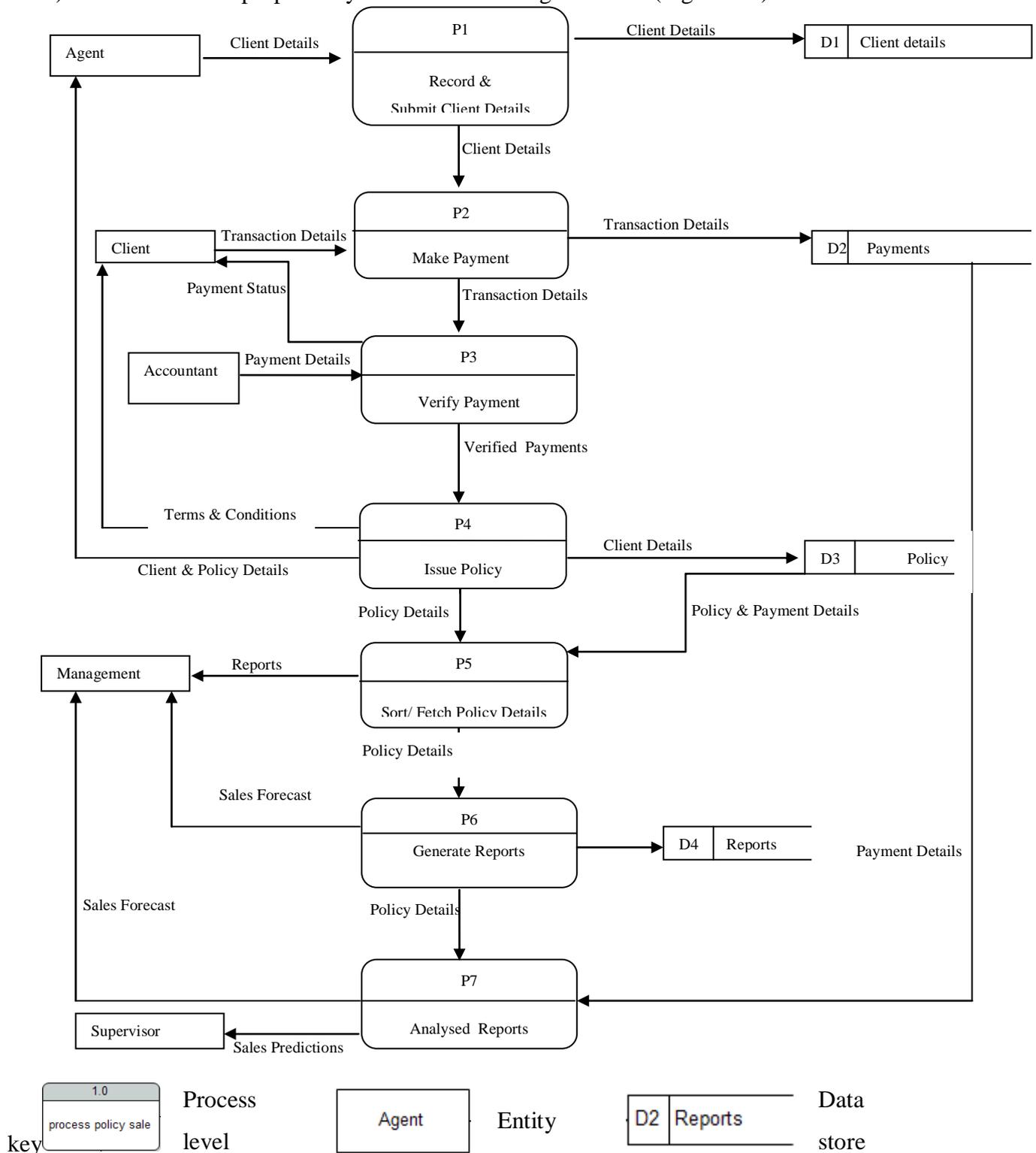


Figure 4.2 Data flow diagram for proposed system

4.3 Architecture design

(Thalheim, 2001) postulates, the diagram below shows a conceptual model defining how the system is structured and how it will behave. This architectural design aims to reduce the difficulties of understanding complex system designs (Firesmith et al, 2008). The system architecture specifies how the system's individual components will interact to efficiently implement the business solution. An architectural design can be defined as a concept that focuses on the components or elements of a system and unifies them into a coherent and functional whole in order to achieve the set objectives. The whole system architecture will be a client-server, mainly consisting of front-end workstations and back-end system server. The system architecture arrangement diagram of the proposed sales analytics system is as shown in Fig. 4.3.

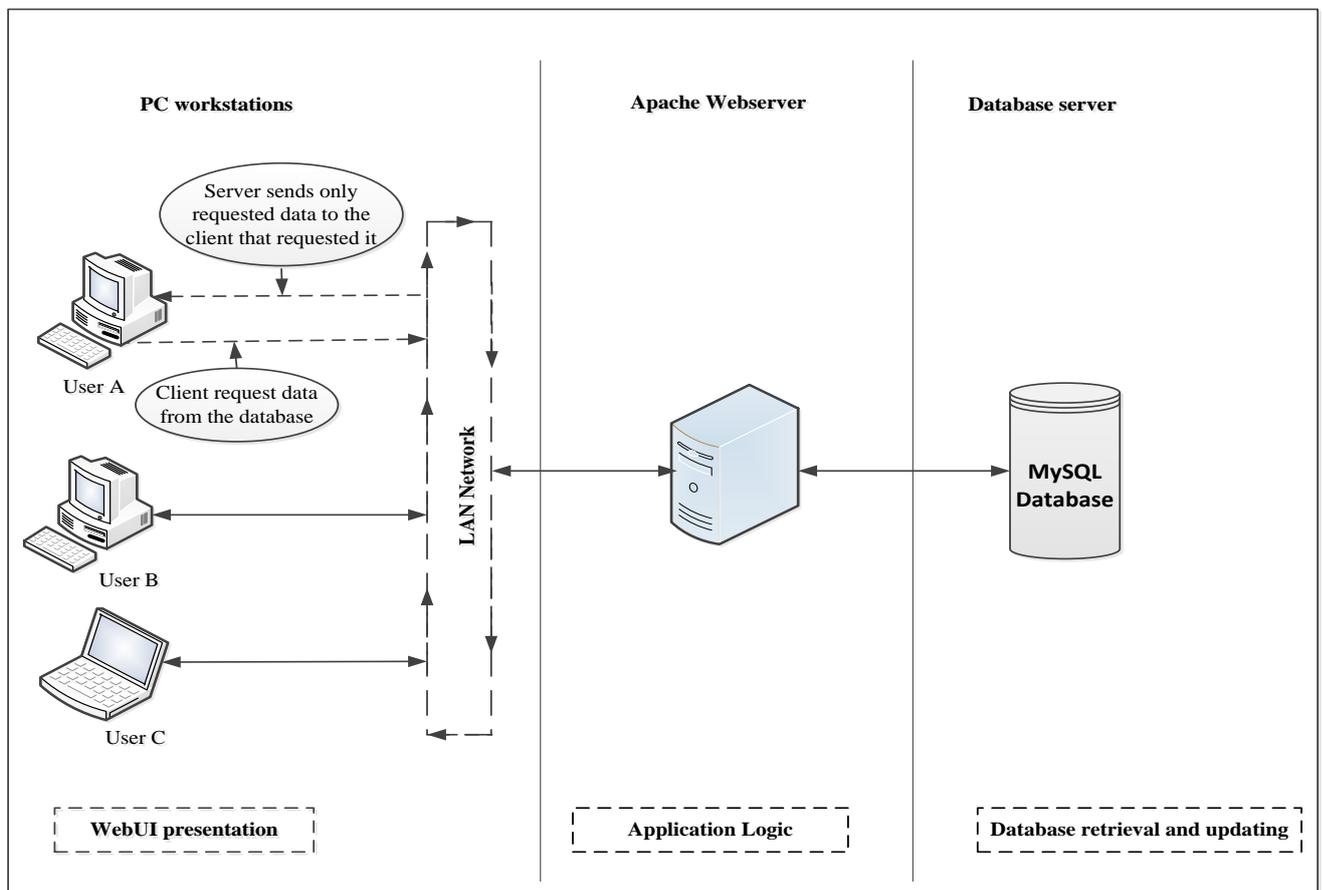


Figure 4.3 Architectural design

4.4 Physical design

Williams (1998), physical design of data is concerned with conversion of user requirements gathered using various data gathering techniques and ideas into a functional system. The physical design relates to the actual input and output processes of the system. This is laid

down in terms of how data is input into a system, how it is verified/ authenticated, how it is processed, and how it is displayed as output. The physical design of a system is its graphical representation of a system which is responsible for showing all of the system's entities, both external and internal (Levin, 2015). The diagram describes the technical environment of the proposed system, which encompasses the hardware that is going to be used and the way it is designed (location of the hardware) and the way in which it is going to be networked. The most important aspect is that of the interaction between the hardware and the software under development. The system will be integrated in an already existing LAN/ WAN infrastructure.

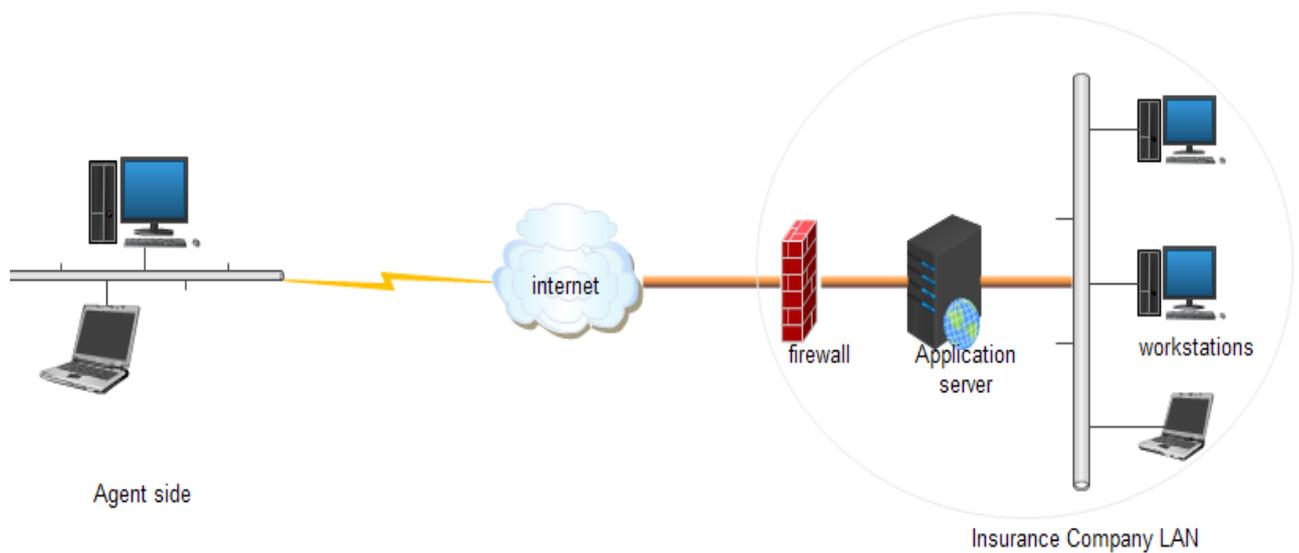


Figure 4.4 Physical diagram

4.5 Database design

Toby et al (2011), a database is a class of grouped stored information that serves a need of various users inside one or more organizations. For any information management and processing system, database is the backbone or the source of data for the system. Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database.

4.5.1 Database schema

The data architecture design is made up of standards policies and models that govern the data to be collected, its arrangement and integration into the system (Prashant, 2009). This is a way of representing data and storing them in layers which are known as schemas. The

database is designed from three different viewpoints known as Schema. (Lightstone et al, 2007).The generalised architecture of a database system is called the ANSI/SPARC (American National Standards Institute - Standards Planning and Requirements Committee)model

ANSI/SPARC three-tier database architecture is shown in the fig.

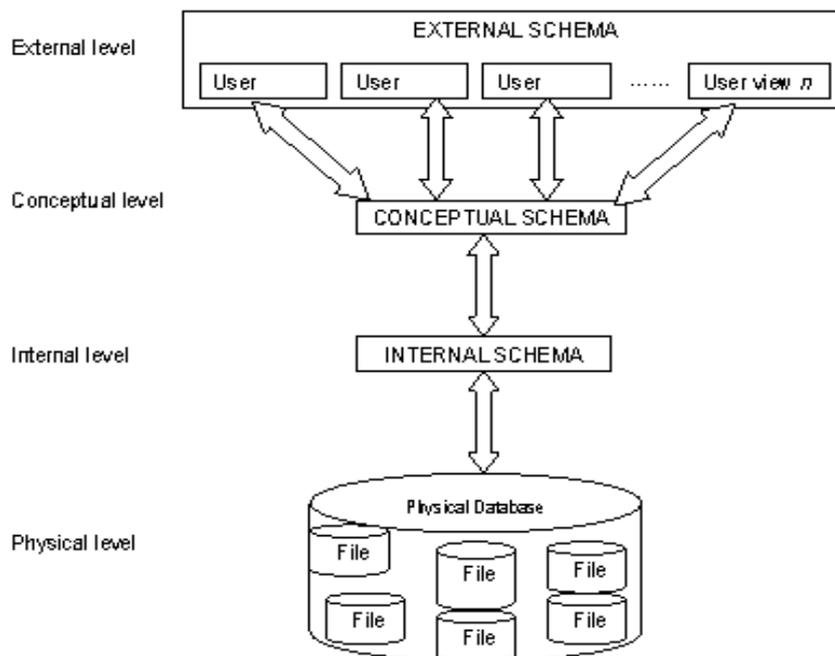


Figure 4.5 Database Architecture.

i. External level

The user’s view of the database is customised to his/her interests. They only view the data that is relevant to them. In this department can only view the appointments done by their project department. The SHEQ trainer also can only view schedule times only for him or her. At this level there is high level abstraction. Information requested by the user will only be displayed and used at this level and the rest of the data remains concealed (Storey, 2006). Howe (2011) suggested that application level portray the local view of the database needed by the application program or programs.

ii. Conceptual level

The conceptual schemas describe the stored information in-terms of the data model of DBMS (Gupta, 2007). This is the community view of the database; it describes the data stored in the database and the relationships among the data or between the database entities are explained

and the conceptual schema also describes the conceptual view not only represents the whole database but continuously checks for integrity and consistency in the whole database (Aït-Ameur, 2007).

iii) Internal Level

This level mainly depicts the way the RDBMS and OS perceive the data. It describes how data is stored in the database. This level goes hand in hand with the internal view and the internal schema which shows all records in the database. The internal schema describes the representation of attributes and how to access them in the database (Fankam, 2008).

4.5.2 Data dictionary

System Users table

Table 4.1 System users

Field	Type	Description	Null	Extra
userID	varchar(10)	userID	No	primary key
Username	varchar(10)	user's username	No	
<u>Password</u>	varchar(20)	Password	No	
Level	int(2)	access level	No	

Insurance Policy

Table 4.2 Policy

Field	Type	Description	Null	Extra
<u>policy_id</u>	bigint(10)	appointment entry number	No	auto_increment
Name	varchar(10)	employment number	No	
Customer_Id	varchar(20)	Project department	No	
Notification	Time	time left work	No	
Date	varchar(10)	clinic visit date	No	
Period	Text	Appointment type	No	

Agents

Table 4.3 Agent details

Field	Type	Description	Null	Extra
<u>AgentNumber</u>	varchar(10)	Agent Account Number	No	primary key
Name	Text	Agent's name	No	
Surname	Text	Agent's last name	No	
Branch	varchar(10)	Agent's Branch	No	
National Id	varchar(15)	national ID number	No	
mobileNumber	varchar(15)	phone number	No	
Address	varchar(40)	physical address	No	
Level	Text	Agent user type	No	
Marital status	Text	marital status	No	
Gender	Text	gender	No	
Location	varchar(20)	location	No	
Business	varchar(20)	Agent's company name	No	

4.5.3 Extended Entity Relationship Diagram (EERD)

The Extended Entity-Relationship (EER) model is a conceptual (or semantic) data model, capable of describing the data requirements for a new information system in a direct and easy to understand graphical notation (Mylopoulos, 2004). An entity-relationship shows information in a system represented in a graphical format and how entities are related to the whole information system (Paul, 2004). The conceptual ER model normally defines master reference data entities that are commonly used by the organization.

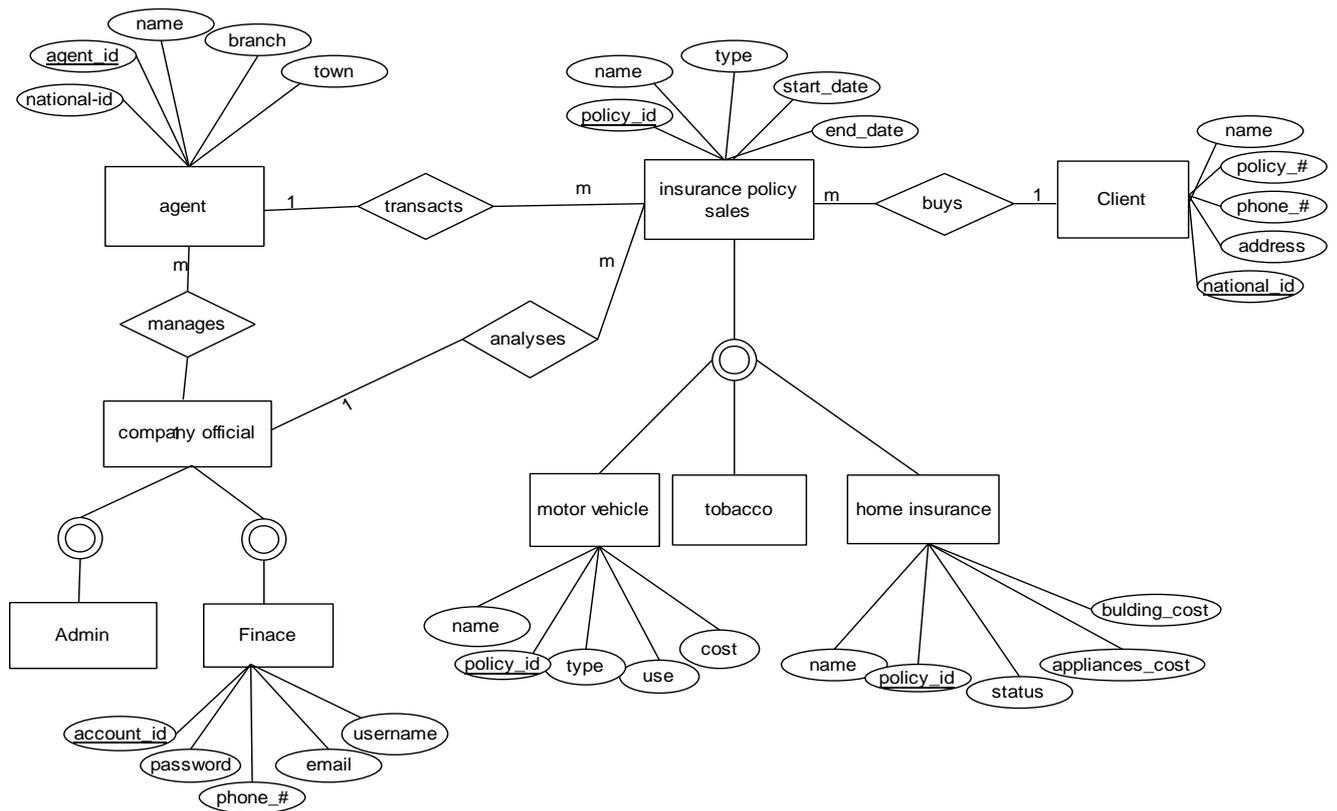


Figure 4.6 Extended Entity Relationship Diagram

4.6 User Interface design

Designing the system interface includes two parts. One is to design the user interfaces and the other is to design the interfaces that interact with other systems, which are integral to the functioning of the system. A system's user interface for the most part goes for making the system as usable to the user as possible hence the arrangement of pages or forms will be connected to the database of the system (Norman, 2002). The design process is achieved with the focus on the user's experience and interaction; hence it involves identifying the primary user interface elements. Primary windows are those windows that the user will interact with most. The goal of user interface design is to make the user's interaction as simple and efficient as possible. There is likewise the need to take a gander at issues like how the system will fit into the user's daily operations or activities (Wolf, 2012).

4.6.1 Log in design

The log in form will be presented for every user before getting started using the system. Every user will have his/her unique log in details depending on the level of use.

The image shows a login form for a system named 'SALES PRO'. At the top, there is a grey header bar with the text 'SALES PRO' in orange. Below the header, there are two input fields: the first is labeled 'username' and has a small person icon on the right; the second is labeled 'password' and has a key icon on the right. Below these fields is a 'sign in' button. At the bottom left, there is a link that says 'click here if you forgot password'.

Figure 4.6 Log in form

4.6.1 Menu design

The main form is the first form presented to a user after successfully log in. it contains the links to all the system modules for example the link to contractor management form. A menu, is a rundown of options which are composed such that the user can without much of a stretch make choices and navigate to the intended page or interface (Dix, 1998). Where a user is restricted from accessing the link, the message box will be displayed

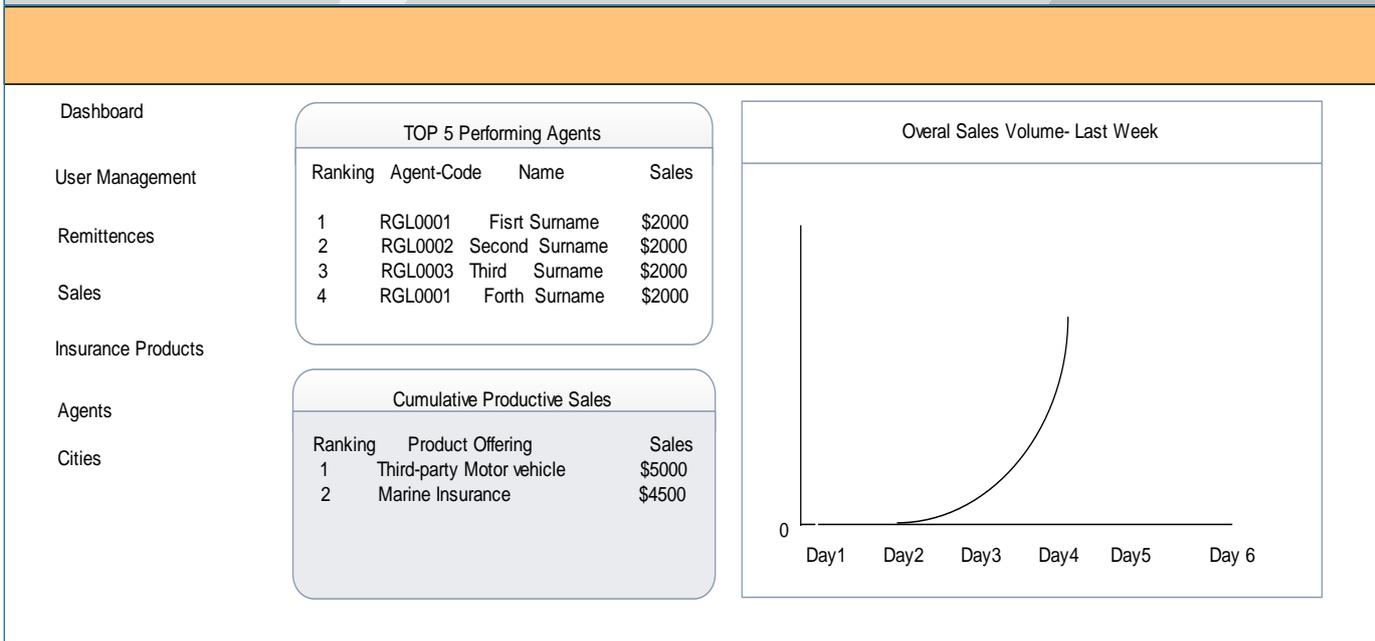


Figure 4.7 System Main window

4.6.3 Input design

The interface should enable automated collection and input of data for example online insurance policy sales. Figure 4.12 is a design of registering a new system user.

Policy Sale

Product Name

Period to ensure

Customer Name

Customer Phone number

Customer ID

Send SMS notification Yes No, not now

Figure 4.8 policy transactionform

❖ Create new Agent account

Agent profiles are created after approved application by the manager. All relevant information will be required to enable valid account.

The image shows a web form titled "New Agent". It contains the following fields from top to bottom: First Name, Surname, Phone number, National ID, Email address, User type, City, Location, and Business Address. Each field is represented by a text input box. At the bottom of the form are two buttons: "Create" and "Back".

Figure 4.9 Adding new Agent

4.6.4 Output design

Output design focuses on designing all the output forms in the system. These includes records of all the data, sales results and the supervisor that have logged in that week or month (Norman, 2002). The system will have a portal to handle key output data in form of reports separated automatically through preset functions such as data of a specific period (for example number of policies sold according agent)

❖ Sales transaction report

The figure below (figure 4.10) illustrates how induction training and (or) medical tests reports will be displayed.

Transaction	Agent Name	To remit by	Total Sales	Paid	Amount Owing
15	ABC	3 June 2017	\$700	\$690	\$10
16	Ronald	4 June 2017	\$500	\$350	\$150

17	Divillers	4 June 2017	\$1000	\$900	\$100
----	-----------	-------------	--------	-------	-------

4.7 Pseudo code

Pseudo code is a simpler format of representing the actual code that has been used in developing the system (Zobel, 2013). Pseudo code is an outline of a program, written in a form that can easily be converted into real programming statements (Johnson, 2012). Pseudocode consists of short, english phrases used to explain specific tasks within a program's algorithm. The goal of writing pseudo code, then, is to provide a high-level description of an algorithm which facilitates analysis and eventual coding but at the same time suppresses many of the details that vanish with asymptotic notation.

Pseudo code for user log in

Start

Select system

Enter username and password

If (username and password) valid Then

Log in

Main Dashboard show

 Application menu

 System Settings

Else

Login failure

End if

End

Pseudo code for Main menu Form show

Start

If (System Settings) Then

Configuration window show

Agents

Users

Products

Configurations

End if

Else

If (Agents) Then

Agents menu show

Active

Add new

Deactivated

Else

If (Products) Then

Schedules window form show

Add new insurance product

Active products

View all products

Else

If (Configurations) Then

System admin form show

Settings

End if

Else

If (reports) Then

Reports window show

Listing Products
Active policies
Retired offerings/products
Show dashboard for sales
Show predictive analysis
End if

Else

If (log out) Then

Log out

Log in form show

End if

Else

Unauthorised user show

Go back to main window

End

4.8 Program design

The system provides the following main functions: online insurance policy sales and tracking and analysis of those sales, and reporting. The conceptual modelling of the system is depicted by class diagrams, package and sequence diagrams.

4.8.1 Package diagram

Package diagrams are constructed in order to easily see the groups of packages and how they are linked. (Goodwin, 2015). Package diagrams visualize packages and depicts the dependency, import, access, generalization, realization and merge relationships between them. Package diagram enables you to gain a high level understanding of the collaboration among model elements through analyzing the relationships among their parent package. This also helps explain the system's architecture from a broad view. The packages are displayed with a tabbed folder with the package's name written on it. A package diagram is a Unified Modeling Language (UML) structure diagram showing the system's packages and

dependencies between the packages. A package or namespace clusters semantically related elements which are likely to change together (Sparks, 2011). There is a dependency from one package to another if any class in the first package has a dependency to any class in the other package. The User Interface Package for example contains the classes that interact with the user for data entry, results presentation and user log in. The figure below shows the package diagram of the proposed system.

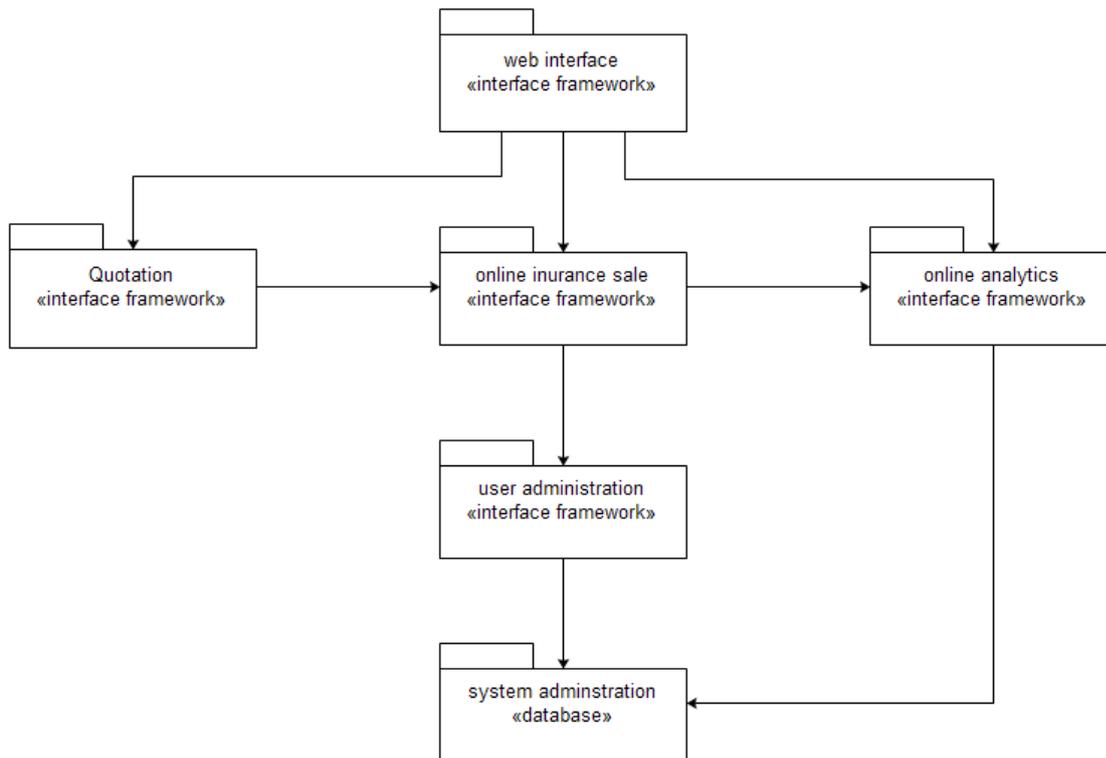


Figure 4.10 Package diagram

4.8.2 Class diagram

A class diagram shows a set of classes, interfaces, associations and generalizations. Classes are depicted as boxes with three sections, the top one indicates the name of the class, the middle one lists the attributes of the class, and the third one lists the methods. A class diagram within the Unified Modeling Language (UML) describes a static type structure diagram showing a system's attributes, operations and relationships between methods (Geoffrey, 2011). The diagrams are used in conceptual, data modelling and detailed modelling which then translates the model into programmable code (Fowler, 2003).

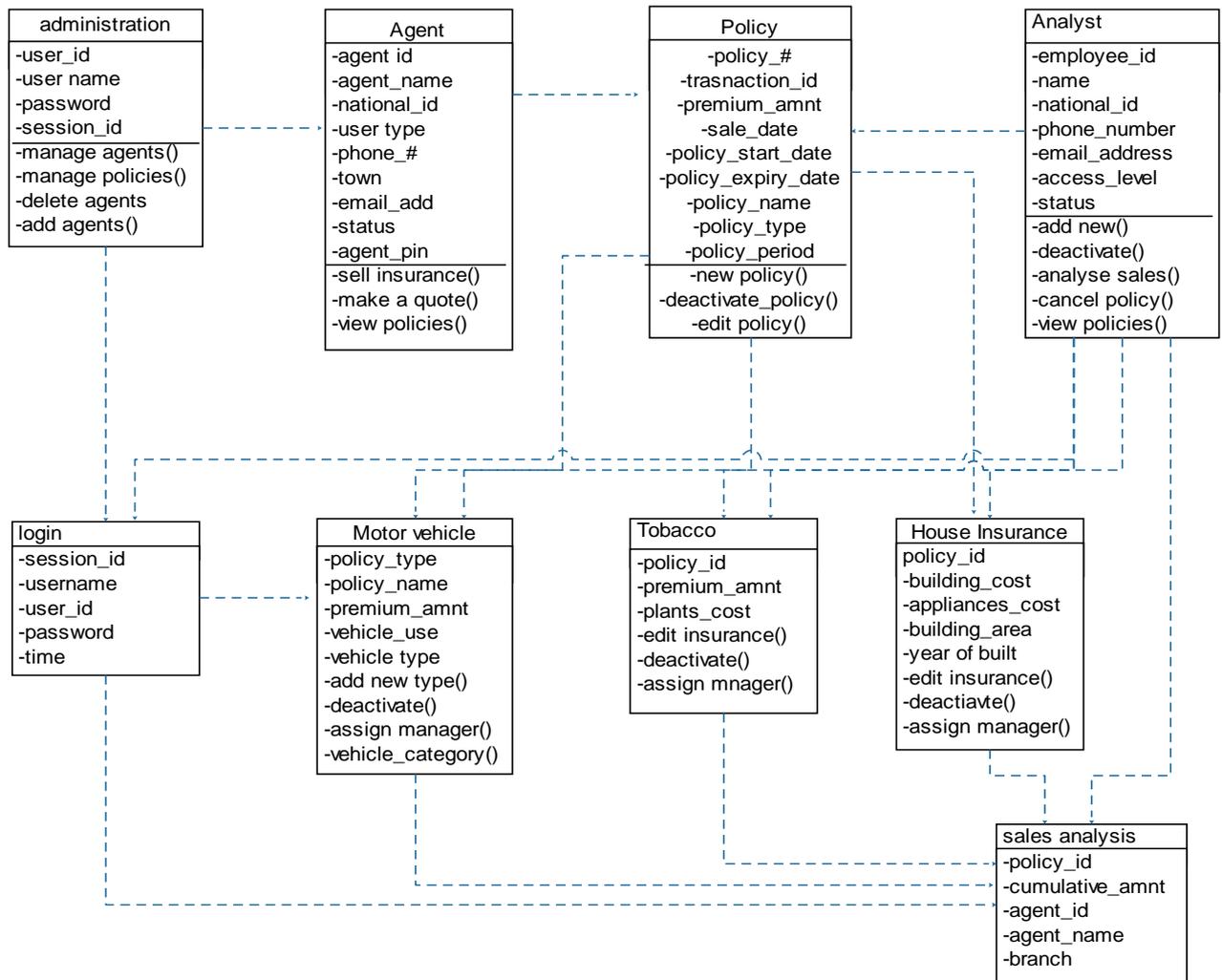


Figure 4.11 Class diagram

4.8.3 Sequence diagram

(Martin, 2003) postulates, a sequence diagram or interaction diagram shows how methods function with together and the way in which they work. A sequence diagram in a Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Key parts of a sequence diagram are (a) *participant*: an object or entity that acts in the sequence diagram and (b) *message*: communication between participant objects. Sequence diagrams provide a high level view of control flow patterns through the system. The figure below shows a sequence diagram for the new system.

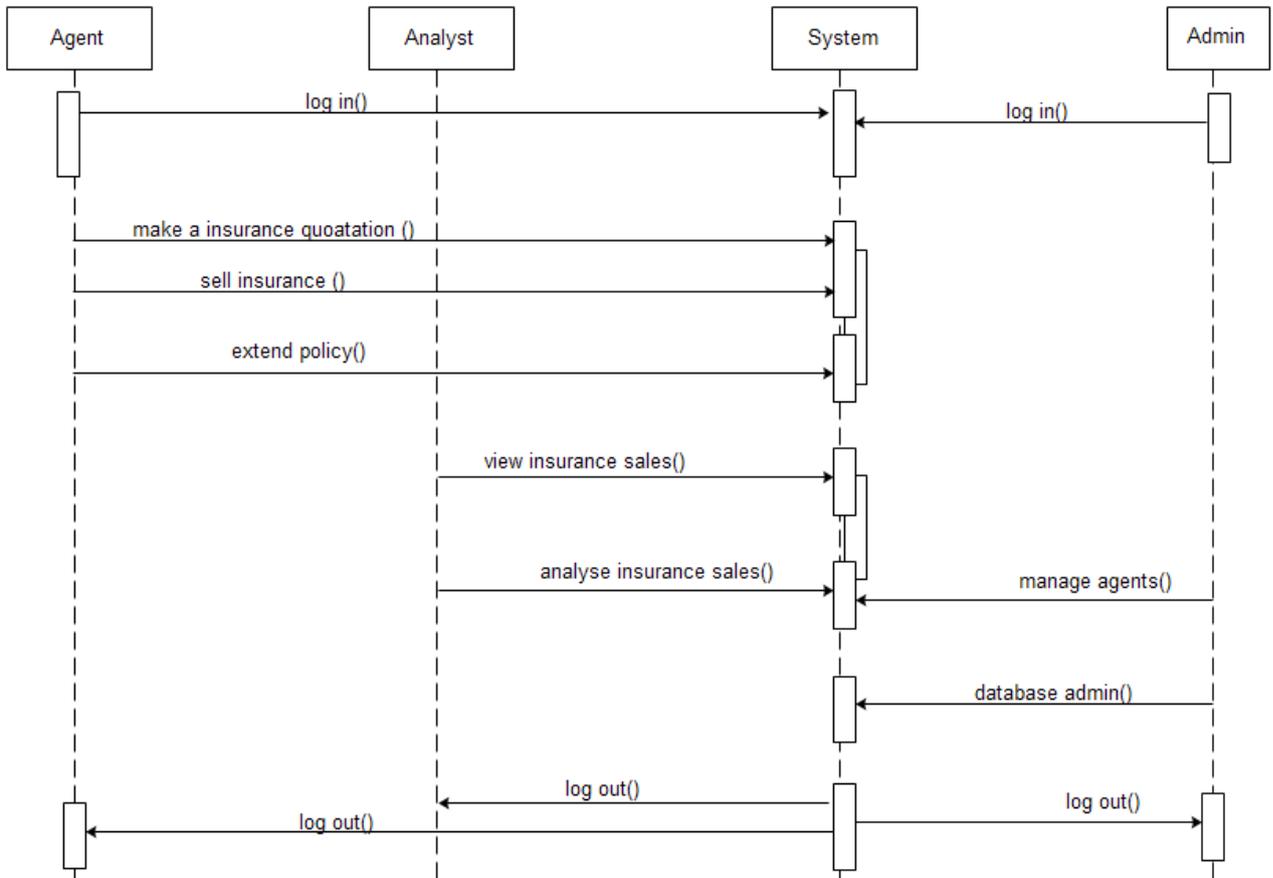


Figure 4.12 Sequence diagram

4.9 Security Design

Security design is an approach is commonly used for building a more secure system. The security of the software is a key feature for that should be thought about when building up a product (Daniel, 2014).

4.9.1 Physical Security

Physical security is the protection of personnel, computer hardware, software and computer networks from physical actions and events that could cause serious loss or damage to organizational data. This includes protection from fire, flood, natural disasters, burglary, theft, vandalism and terrorism. This ensures that all the hardware components associated with the system are stationed in a location that is theft free and free from damage by nature or the surrounding (Jannsen, 2014).

4.9.2 Network Security

This ensures that security levels of both hardware and software with the network of the system is secured to avoid accessing information to the data shared over the network. The Arduino Ethernet shield has an Ethernet port which must be deactivated when inactive or unplugged from a recognized mac or IP address (Harmening et al, 2009). The web server is protected with either an HTTPS socket or login credentials avoiding unauthorized access to the data. The network administrator traces IP address that access the web server for accountability purposes.

4.9.3 Operational Security

Operational security guarantees that data stored within the system is safe from unsafe and unauthorised hands which may be potential threats to the information (Andrew, 2009). The data must be safe from misuse by others in the system. When the data is entered into the system it must be consistence and free from some alteration except for authorized personnel.

4.9.4 Backup Design

The purpose of the systems backup is to provide a means to: (1) restore the integrity of the computer systems in the event of a hardware/software failure or physical disaster, and (2) provide a measure of protection against human error or the inadvertent deletion of important files. The IT personnel will be responsible for carrying out system backup operations on the MySQL server weekly, fortnightly and monthly.

The backup strategy was designed to be able to:

- ❖ Recover from data loss in all circumstances like hard drive failure, virus attacks, theft, accidental deletes or data entry errors, sabotage, fire, flood, earth quakes and other natural disasters.
- ❖ Recover to an earlier state if necessary like due to data entry errors or accidental deletes.
- ❖ Recover as quickly as possible with minimum effort, cost and data loss.
- ❖ Require minimum ongoing human interaction and maintenance after the initial setup.
Hence able to run automated or semi-automated.

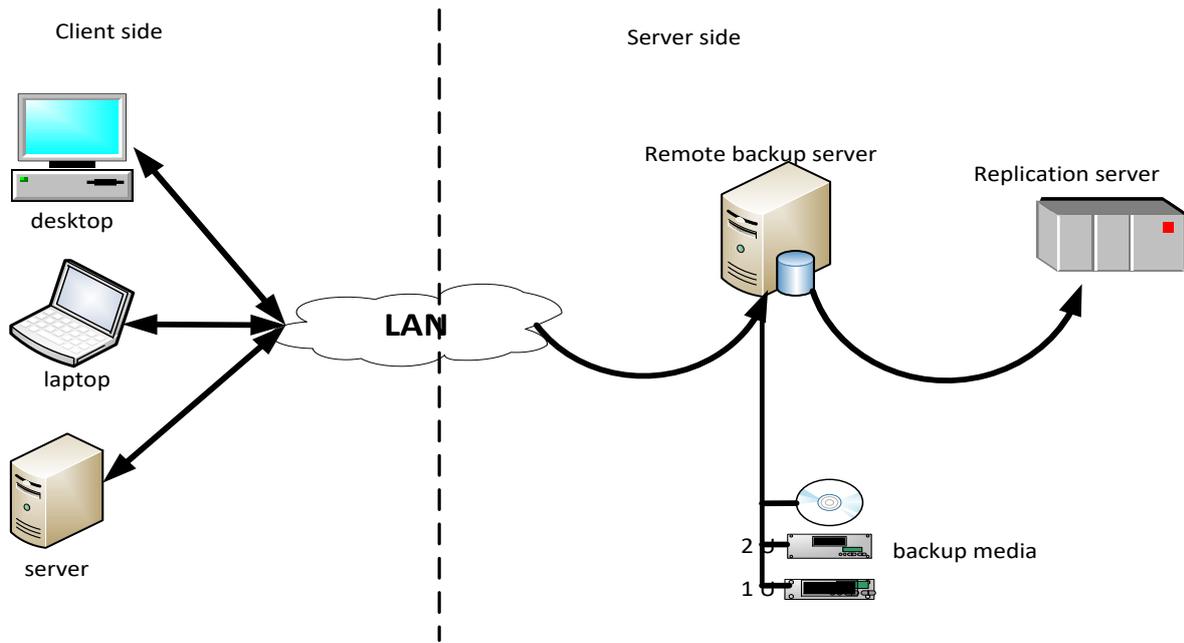


Figure 4.13 Backup design

4.9 Conclusion

This chapter mainly focuses on a detailed description on the designing of the proposed solution like the layout of the hardware and software, description of the data input and output and the structure of the proposed system. Most of the goals of the system were addressed and illustrated by the use of data flow and entity relationship diagrams. The program, logical, interface, input and output design paved the way for implementation phase, which is the actual development of the proposed solution.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter highlights the system maintenance process of defining how the technological solution will be built, tested and maintained to ensure that it is operational and used (Burd, 2007). The purpose of system implementation can be summarized as follows: making the new system available to a prepared set of users (the deployment), and positioning on-going support and maintenance of the system within the performing organization (the transition). At a finer level of detail, deploying the system consists of executing all steps necessary to educate the users on the use of the new system, placing the newly developed system into operation.

5.2 Coding

“Coding is the process of designing, writing, testing, debugging, and maintaining the source code of computer programs” (Milner, 2010). Computer codes are set of instructions which are used to create and design user applications as solutions to the real world struggle (Bebbington, 2014). Dreamweaver CS5, MySQL and SQL statements are used for development of the software package (Hartmann, 2009). During this process, coding is done with future maintenances in the mind. In order to come up with easy and smooth software construction, some techniques used include using a coding standard that support coding reviews and unit testing and avoiding complex or hard to understand language structures. The following are considered during coding activity:

- (a) Creating understandable source code for example naming conventions
- (b) Use of classes, enumerated types, named constants
- (c) Uses of controlled structures
- (d) Source code organisation (into statements, classes, and other structures).

5.3 System Testing

(Watson, 2007) suggested that at the University of Georgia, testing occurs when “a system is put into a testing environment to check for errors and bugs.” System testing is aimed at evaluating an attribute or capability of a program or system and determining that it meets its

required results. . This testing can be determine whether the system is presented to the supervisor or persons accountable (Prasad, 2008). Testing is usually performed for the following purposes:

- ❖ To improve quality.- As computers and software are used in critical applications, the outcome of a bug can be severe. Debugging, a narrow view of software testing, is performed heavily to find out design defects by the programmer and get them fixed.
- ❖ For Verification & Validation (V&V) - Testing is heavily used as a tool in the V&V process. Testers can make claims based on interpretations of the testing results, which either the product works under certain situations, or it does not work (Jiantao, 1999).
- ❖ For reliability estimation- Software reliability has important relations with many aspects of software, including the structure, and the amount of testing it has been subjected to. Testing can serve as a statistical sampling method to gain failure data for reliability estimation.

The following diagram shows the testing levels during the testing process.

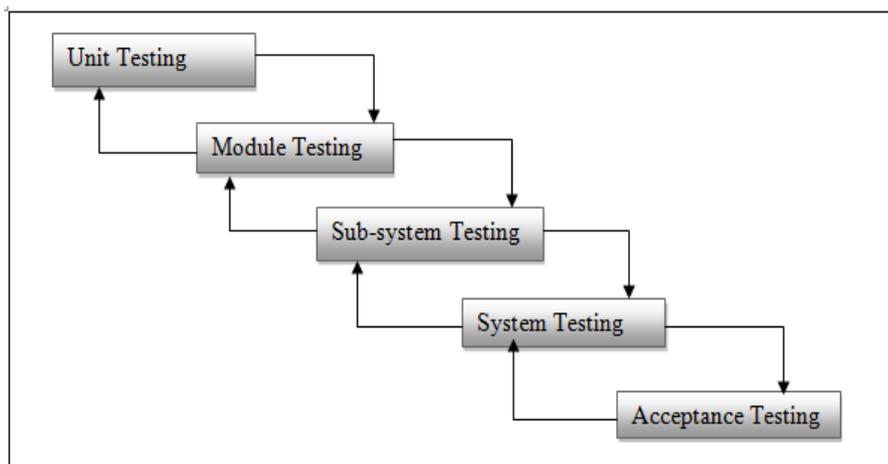


Figure 5.1 System testing process

Levels of System testing

Unit testing

Unit testing concentrates on each unit for example component, class or object of the software as implemented in the source code (Pressman, 2010). Unit testing is when person facets of the program are tested to see if they are good enough for the proposed use (Noel, 2015). A series of stand-alone tests are conducted during and each test examines an individual

component that is new or has been modified. The purpose is to validate that each unit of the software performs as designed. Unit testing is performed by using the White Box Testing method.

White box testing

White box testing is done to test the core operation of the system, testing to check if smaller changes on one facet of the program have an impact on different modules (Chauhun, 2010) This is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system. The following screen shot is a log in test to see if it works.

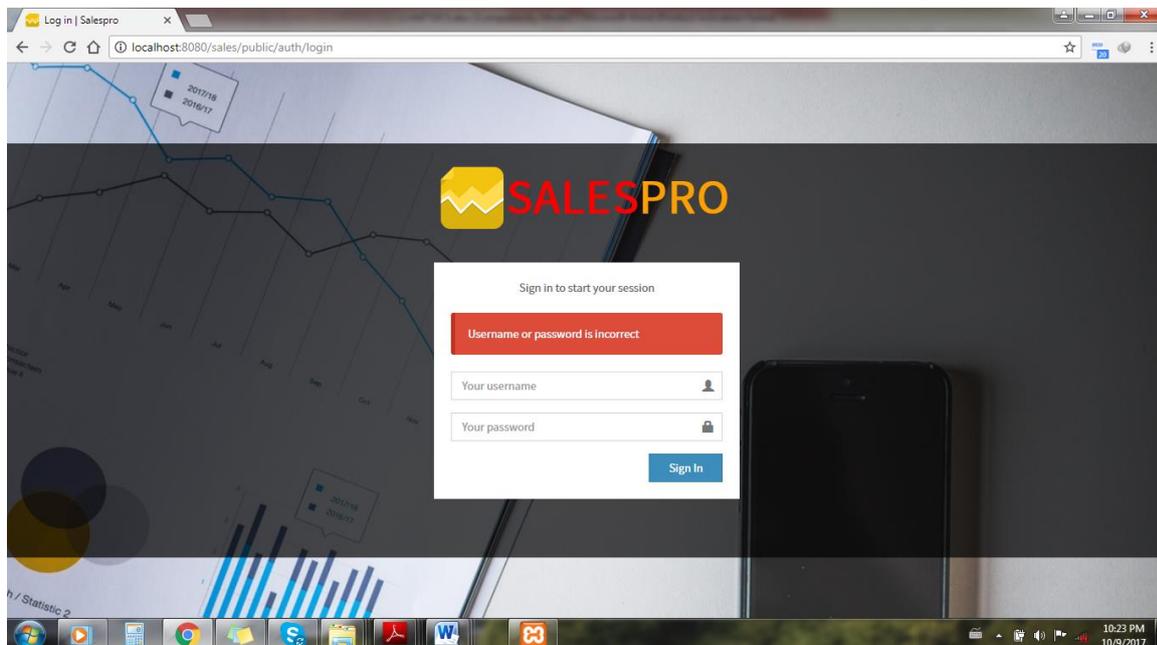


Figure 5.2 White box testing

Black-box Testing

Unlike white-box testing, black-box testing looks at the product concentrating on its usefulness as opposed to its inward structure. As such, this sort of testing does not consider the structure of the program, rather it concentrates on what the program does consequently the other name given to this kind of testing is functional testing (Saleh, 2009). It looks at how practical a product is, depending on the expressed objectives. In black-box testing, the operations team behind completing the testing procedure does not have the information of the interior system or structure of the product being tried. Information about the systems inputs

and outputs is known however how the system procedures the inputs to create the outcomes is not known.

5.3.1 Module testing

A module is any collection of executable program statements that can be called from any other module in the program and has the potential of being independently compiled. A module, which can be a closed subroutine, a procedure or a subprogram, is the primary unit of a program's structure. Module testing is, therefore, a process of testing the individual subprograms, subroutines or procedures in a program. Its purpose is to check if the module contradicts the system internal specifications (Myers, 1979). Module testing makes the debugging tasks much easier because it only involves a single module and, when an error is found, the error should be located within that particular module. The person who is testing a module should be the one who knows the internal details of that module - namely, the programmer who has coded the module.

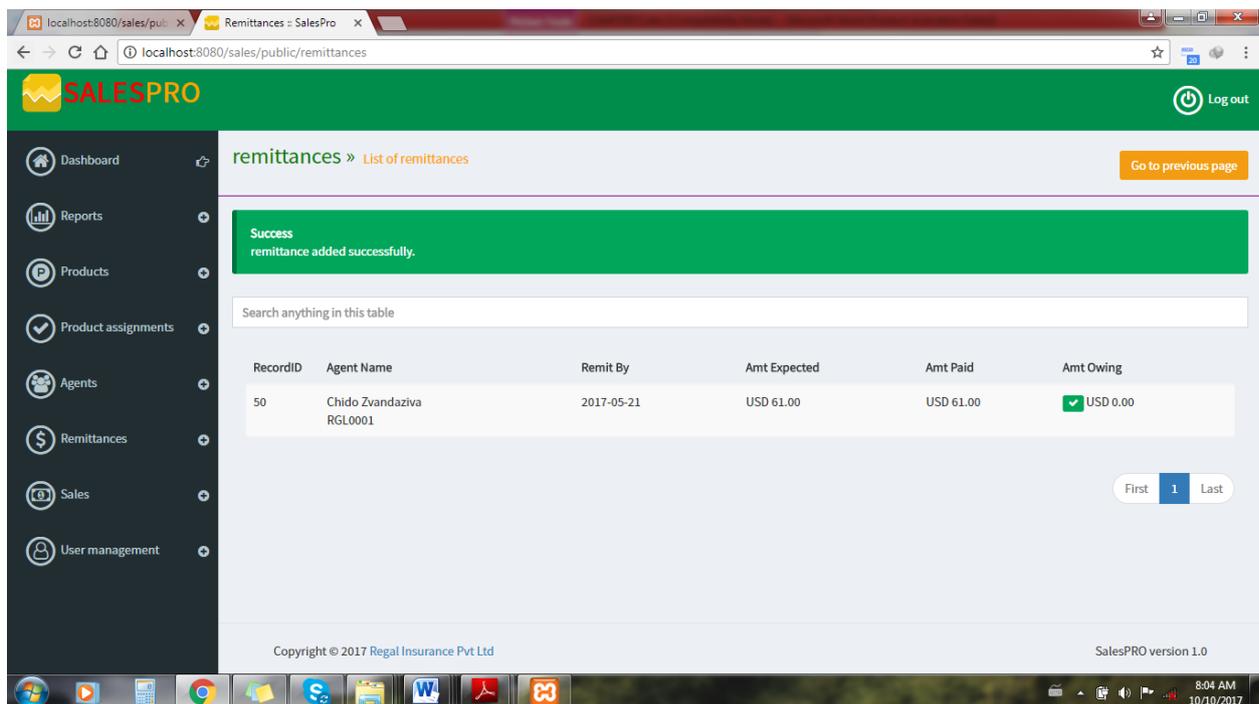


Figure 5.3 Module testing

5.3.2 Sub system testing

Subsystem testing often called integration testing is a level of the software testing process where individual units which define subsystem are combined and tested as a group (McDonald, 2012). Integration testing is performed as a prelude to system testing. The test is often done on both the interfaces between the components and the larger structure being constructed, if its quality property cannot be assessed from its components. Integration testing examines all the components and modules that are new, changed, affected by a change, or needed to form a complete system (John, 2005). During the process of developing the system when two or more units are ready, they are assembled and sub system testing is performed for example, whether the insert form will add data to the database.

5.3.3 System testing

System Testing tends to affirm the end-to-end quality of the entire system. System test is often based on the functional/requirement specification of the system. Non-functional quality attributes, such as reliability, security, and maintainability, are also checked (Saleh, 2009). System Testing tests all components and modules that are new, changed, affected by a change, or needed to form the complete application. The system test may require involvement of other systems but this should be minimized as much as possible to reduce the risk of externally-induced problems. The emphasis in system testing is validating and verifying the functional design specification and seeing how all the modules work together.

Acceptance testing

Acceptance Testing is a level of the software testing process where a system is tested for acceptability and it is concerned with the fully integrated system, involving users, customers, or a third party designated by the customer. User Acceptance Testing (UAT) also called Beta testing or end-user testing comprises functional tests, interoperability tests, and regression tests (Cimitile, 2000). Acceptance Testing is done when the completed system is handed over from the developers to the customers or users. The purpose of acceptance testing is rather to give confidence that the system is working than to find errors and to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. Usually, Black Box Testing method is used in acceptance testing. (Koopman, 2011) propounded that the purpose of acceptance test is check if the system meets all requirements to be used by a customer and it is usually the last checkpoint before shipping a system and

might be performed on all systems to check for hardware defects/manufacturing defects, not just software design problems.

5.3.3.1 System versus Objectives

The whole system is tested against all the stated objectives to check for conformance and usability.

The aim of testing the system at this level is to find out whether the system is performing as expected so that we can determine if the system will be able to meet the user requirements.

❖ Objective 1

vi) To be able to analyse historical sales data for decision-making.

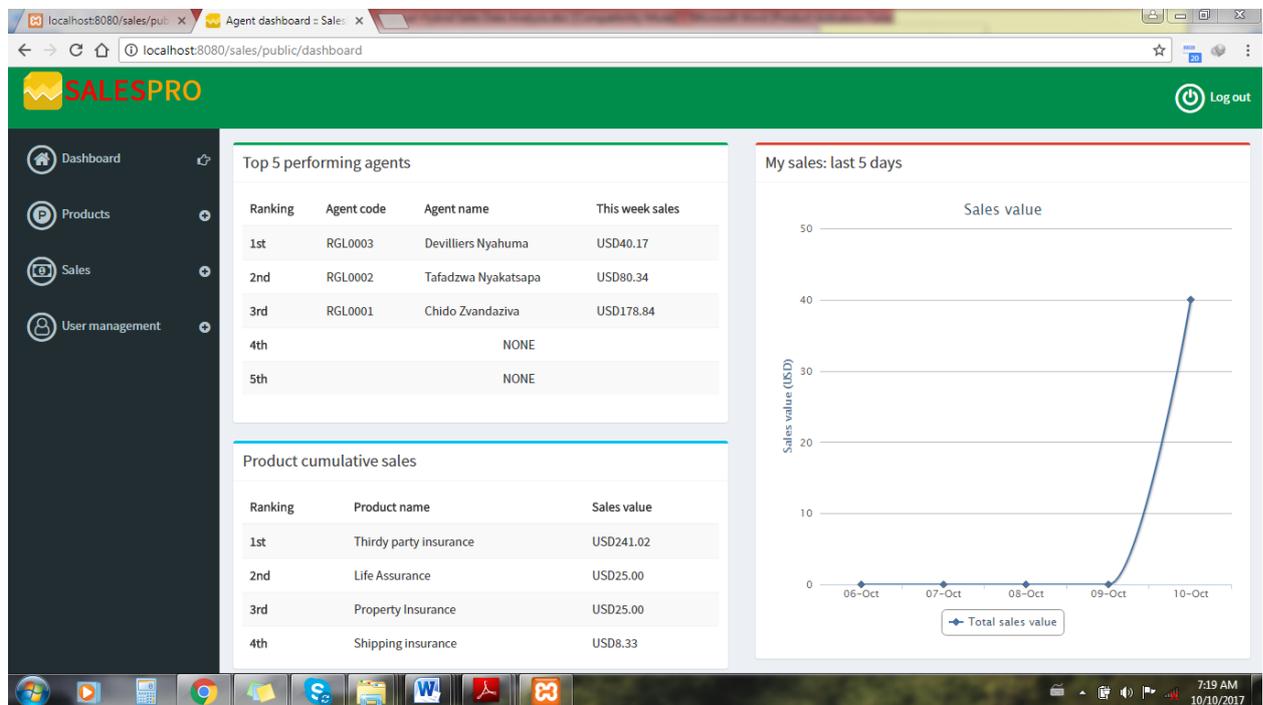


Figure 5.4 Preview of system objective 1

System solution

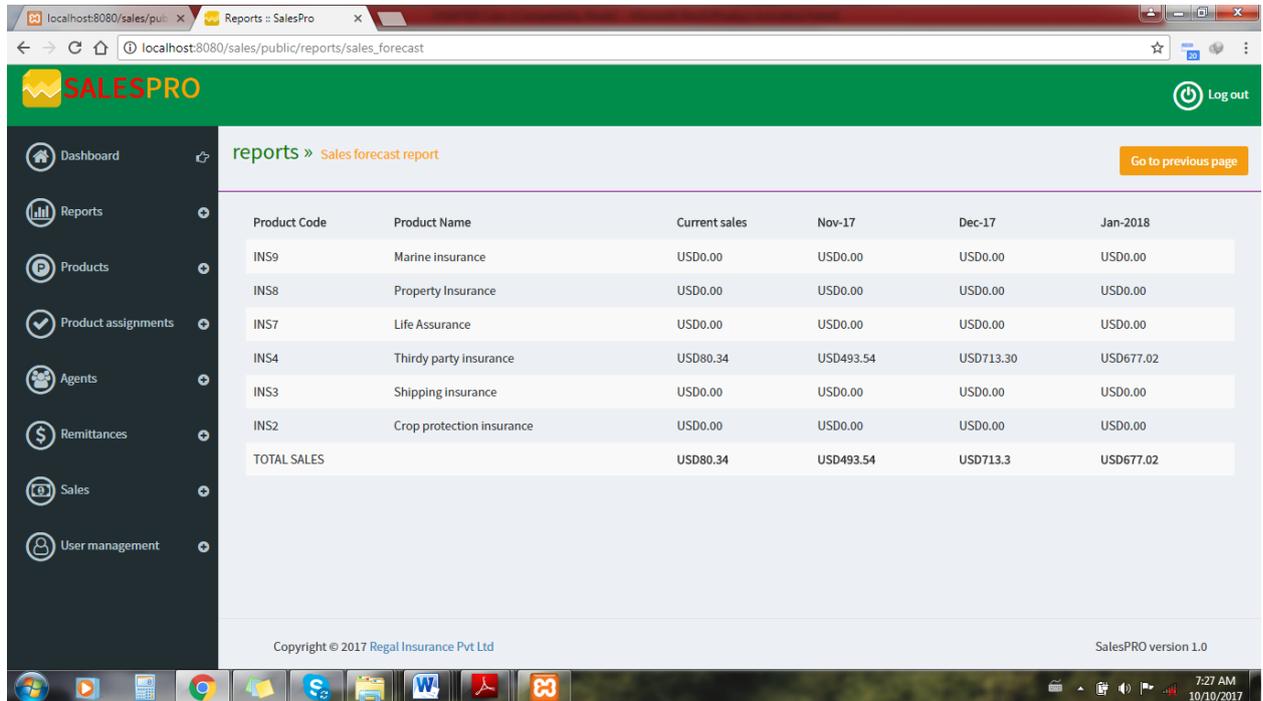
The system can analyse sales data by agent and compute total sales for the past week and rank the cumulative sales based on product offering.

❖ Objective 2

To be able to forecast future sales based on current sales performance.

System solution

The system can randomly forecast sales for all product offerings based on current sales data. Below is the screen shot.



The screenshot shows a web browser window displaying the SalesPRO application. The page title is "reports > Sales forecast report". A sidebar on the left contains navigation links for Dashboard, Reports, Products, Product assignments, Agents, Remittances, Sales, and User management. The main content area displays a table with the following data:

Product Code	Product Name	Current sales	Nov-17	Dec-17	Jan-2018
INS9	Marine insurance	USD0.00	USD0.00	USD0.00	USD0.00
INS8	Property Insurance	USD0.00	USD0.00	USD0.00	USD0.00
INS7	Life Assurance	USD0.00	USD0.00	USD0.00	USD0.00
INS4	Thirdy party insurance	USD80.34	USD493.54	USD713.30	USD677.02
INS3	Shipping insurance	USD0.00	USD0.00	USD0.00	USD0.00
INS2	Crop protection insurance	USD0.00	USD0.00	USD0.00	USD0.00
TOTAL SALES		USD80.34	USD493.54	USD713.3	USD677.02

At the bottom of the page, there is a copyright notice: "Copyright © 2017 Regal Insurance Pvt Ltd" and the version number "SalesPRO version 1.0".

Figure 5.5 Preview of system objective 2

❖ Objective 3

To be able to randomly analyse and track sales on custom dates as per agent.

System solution

Reports are produced showing all the insurance sales done by the agent in a specified period of time.

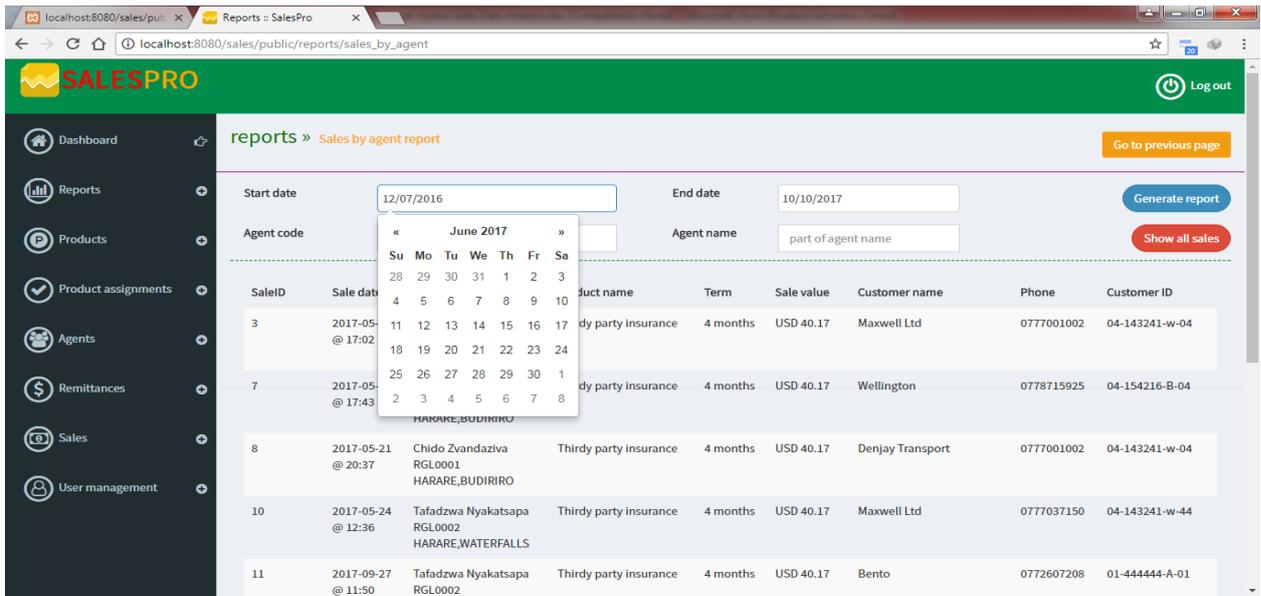


Figure 5.6 Preview of system objective 3

❖ Objective 4

To make informed decisions from the sales performance data.

System solution

Based on sales analysis, top performing agents and product offering are easily computed. From this statistics the company may make wise decisions on allocating limited sales resources for example POS machines to top performing agents. Poor performing agents may also be retired when necessary or a strong promotional budget may be recommended.

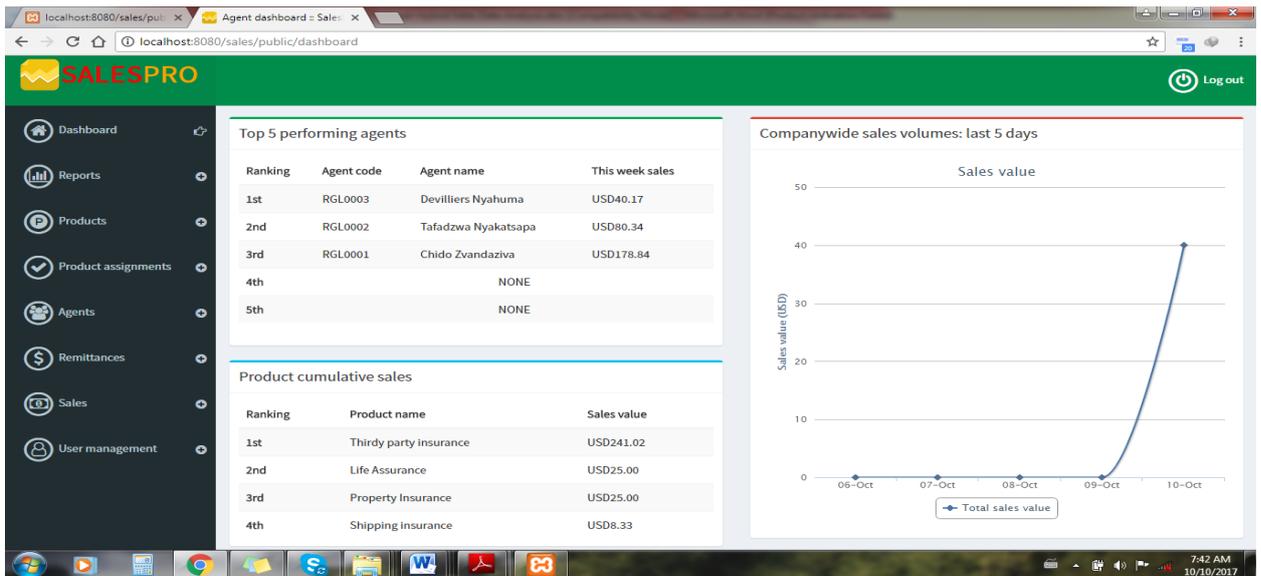


Figure 5.7 Preview of system objective 4

5.4.1 Validation and verification

System validation and verification (V&V) is regarded as a coherent discipline and both validation and verification are needed throughout the lifecycle. "Software V&V is a systems engineering discipline which evaluates the software in a systems context, relative to all system elements of hardware, users, and other software" (Dolores, 2011).

5.4.1.1 Validation

The objective of this phase is to check whether the right system is being developed for the specified problem. The new system is compared with the user requirements to ensure that it meets the requirements. It involves making subjective assessments of how well the (proposed) system addresses a real-world need. Validation is achieved through the comparison of data entered and the output. This is to ensure that the data is captured as required and the output is as expected. Validation mainly checks values entered in the textboxes if they corresponds the specified field type.

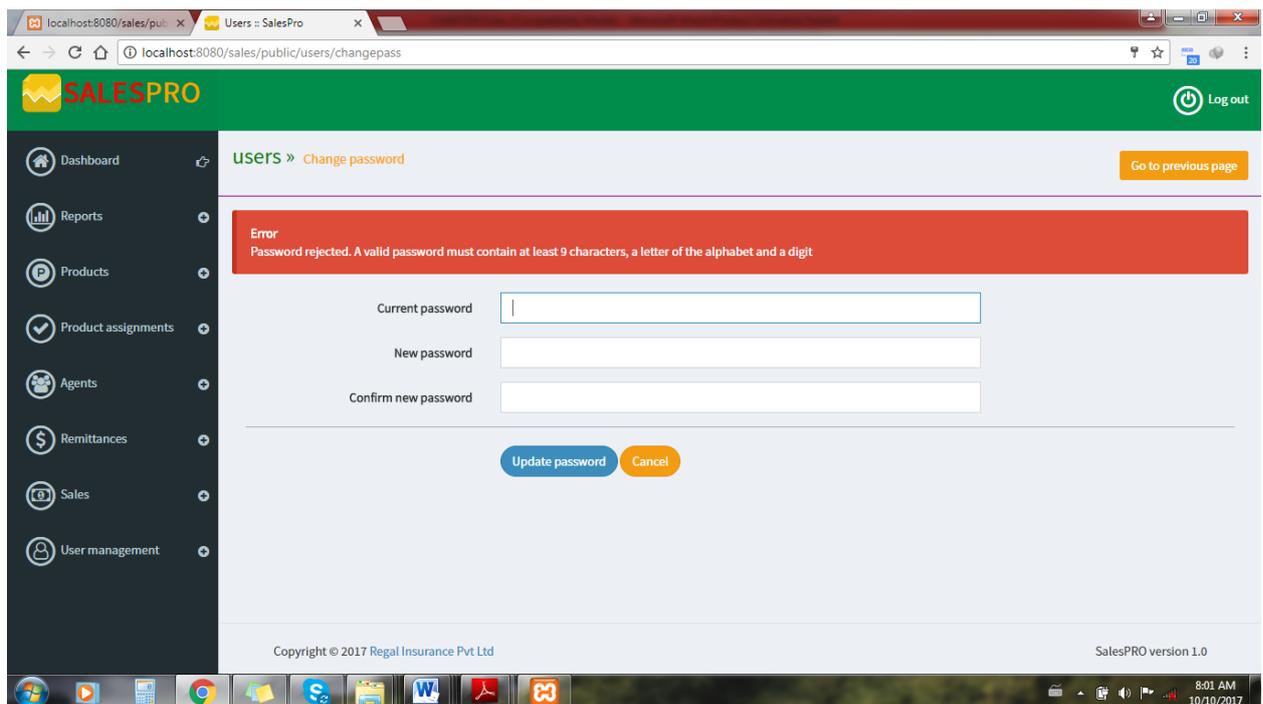


Figure 5.8 Illustration of system validation

5.4.1.2 Verification

This is concerned with building the system right, whether the system is well-engineered, error-free, and so on (Serendipity, 2012). Verification is the comparing of the realised data

against the reasons it was made for (Engel,2010). Verification will help to determine whether the software is of high quality, but it will not ensure that the system is useful. It is concerned with whether we are building a system in conformance with the expectations of the important user. The system exhibited consistencies and correctness in its execution thus meeting its objectives.

5.5 Installations

In this section, it gives us instructions on how to install the new system and ensure that it is running and ready to produce the required services to the users. This is going to include hardware and software configurations at the current site of doing work. The following are steps for the installation of the system

- i) Hardware Installations.
- ii) System Installations
- iii) User Training
- iv) File Conversions

Hardware installation

On hardware installations it includes setting-up of necessary hardware for the proposed new system that is installation of the new servers, the printers, the computers and all the available network installations required. The Information Technology staff for the organisation will be responsible for backing up and maintenance if necessary.

Systems Installation

This is when the application software is loaded into the computer hard disk to start to work.

The installation process is described below.

- i) Install XAMPP and create a root folder named salespro in C:\xampp\htdocs\
- ii) Install the system files to the root folder. (in C:\xampp\htdocs\salespro)
- iii) Create and map the system data base

5.4.3 User training

Training is a process of transferring measured and defined knowledge and skills (Wills, 1998). User training involves preparing the system end users to have a knowhow of how to

interact with the system before operations. This also includes involves the IT personnel learning how to troubleshoot the system in the event of problems arising as users continue to use the system. In-house training will be conducted within a week's period for all users and to make training easier, user manuals will be provided (see Appendix). The three main groups for training are users, managers, and IT staff. The training will be conducted on different days at the organization's premises.

IT personnel- IT personnel will support the system during whole of its life cycle, so they will familiarise on troubleshooting the system.

Users – Agents will be trained to sell insurance online and how to manage personal information

Finance department- will be trained on how to process remittance from agents and reporting

System Change-over

This process of inserting the new information system to online and retiring of the old or ancient system is known as system change-over. Changeover is also known as cut over or even go live and is a process of moving from use of the old system to the new system (Cannon, 2011). There are currently four(4) change-over ways which are :

Direct Cut-over

Direct cut-over approach has caused the change-over from old system migrating to the new system and has to occur abruptly when the new system becomes operational. It is least expensive but at the same time involves more risks compared to other change-over methods.

Advantages of Direct cut-over

- i) Low operational costs due to the maintenance of only a single recordkeeping system
- ii) Straightforward implementation all staff are using the same system at all times
- iii) Clear direction for policy

Disadvantages of Direct Cut-over

The method of the new system change-over involves higher risks of the whole system failure and it is preferable for (CSP) commercial software packages. So if the system fails in health centre then it is going to be difficult to store data of employees who would have visited health

centre. And if the storage is not done properly then there will be some incorrect reports and monitoring of the employee's occupational health and this will not have been properly done.

5.4.4.2 Parallel Operation:

The Parallel Operation Changeover method directly require that the old and new information systems operate at a full scale for a specified period of time. Information is put to both systems and the output generated by the new system is then compared with the one equivalent to the output from the first old program. When the management, users, and Information Technology group are now satisfied that the current new system operates correctly and perfectly then the old system will automatically be terminated. It is the most expensive change-over method and has lower risks.

Advantages

- i. The advantages of the parallel system is it has lower risk of system failures, so all the other tasks can be done properly at the health centre. If the new system does not work well or properly, then the health centre can start to use the old manual system that it has been using as a backup until an appropriate change has been rectified. It gives us a clear view and picture of the old system versus the new system.
- ii. In this system it allows some errors to be corrected as soon as they are discovered or brought to light before the new system made fully operational.
- iii. There will be no disturbance or disruption of the operations in the whole department, since the old system will still running in case of emergence the new system encounters serious problems.

Disadvantages

As we know parallel system is the most costly changeover method as both old and new systems operate fully for specified period and we also know that the budget of health centre is also low so it will be difficult for health centre to follow this changeover process.

5.4.4.3 Pilot Operations

This pilot change-over method involves implementations of the complete new system and at a selected location of a organisation. The direct cut-over method and operating are both of the

systems for only pilot site. The first group that uses the newly installed system first is also called the Pilot Site. By limiting the implementation to a pilot site reduce the risk of system to fail as compared with a less expensive than the parallel system.

Advantages of Pilot Operation

Pilot operation is combination of both direct cutover parallel operation, which restricts the implementation to a pilot site and reduces risk and of system failure as compared with a direct cut-over method.

Operating system only at pilot site is less expensive than parallel operation for entire health centre and all health centres.

If we use the parallel approach to finish the implementation then that would mean the changeover period can be shorter if program proves successful at the pilot site so most of time will be consumed at health centre in the implementations of the new system.

Disadvantages

This method is also expensive as compared to direct cut-over.

5.4.4.4 Phased operation:

The phased operation changeover method involves implementing the new system in stages, or modules. We can implement each subsystem by using any of the other three changeover methods. In this approach risk of errors or failures is limited to the implemented module only as well as it is less expensive than the full parallel operation.

Advantages

As we know in this method we have to implement the new system in stages, or modules, which is less prone to risk of system failure or errors at health centres, as failure is limited to the implemented module only.

It is also less expensive than parallel system because we have to work only with one part of system at a time.

Disadvantages

As the system, which we are implementing, involves various phased operation like treatment, registration, clinic visit authorisation etc so it can cost more than the pilot approach.

5.4.4.5 Recommended strategy

As we can determine from above information, the parallel conversion strategy is the best because it is less risk. There will be no disruption of operations in the department since users can switch to the old system in the event that the new system encounters problems.

5.5 Maintenance

System maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes or to adapt the product to a modified environment (Khaled, 2006). Maintenance is a very important process that needs to be maintained if the system has to survive longer while delivering perfect goods. As software systems age, it becomes increasingly difficult to keep them 'up and running' without maintenance. The maintenance of system is important because:

- ❖ A system will continue to change and evolves as it is used in the long run.
- ❖ The changes will be arising with time while changing and evolving as it is used by the users.
- ❖ The changes will then be arising from request by users due to the problems. Reports from all operating departments or groups who would have identified the bugs in the system that must be corrected and fixed.

There are four types of system maintenance and these are corrective maintenance, perfective maintenance, adaptive maintenance and préventive maintenance.

5.5.1 Corrective Maintenance

Corrective maintenance is the reactive modification of a software product performed after delivery to correct discovered faults (Martin, 2009). A defect can result from design errors, logic errors and coding errors. Defects are also caused by data processing errors and system performance errors. All these errors, sometimes called 'residual errors' or 'bugs', prevent the software from conforming to its agreed specification. Examples of corrective maintenance include correcting a failure to test for all possible conditions or a failure to process the last record in a file (McClure, 1983).

5.5.2 Adaptive Maintenance

Adaptive maintenance is the modification of a software product performed after delivery to keep software usable in a changed or changing environment (Mahgiary, 2004). Adaptive maintenance is a process or path under which a system is set to adjust the change inside the program or in the system environment and corrective maintenance is a method of correcting a malfunction in the system (Shaun 2009). (Wright, 2007) states that Adaptive maintenance is continuation that allows software to familiarize to a change in the system environment. This shall be undertaken to provide for changes that might happen to the operational environment of the user. This is also done to advance the system or add enhancements so as to acclimatize it to the environment changes, for example there might be a need to develop a bigger size of the database, to cater for business branches.

5.5.3 Perfective Maintenance

This is a modification about a software product after delivery to improve performance or maintainability and preventive maintenance is performed for the purpose of preventing problems before they occur (Bendolpher, 2006). Perfective maintenance mainly deals with accommodating to new or changed user requirements. Perfective maintenance concerns functional enhancements to the system and activities to increase the system's performance or to enhance its user interface (VanVliet, 2000).

5.5.4 Preventive maintenance

Preventive maintenance concerns activities aimed at increasing the system's maintainability, such as updating documentation, adding comments, and improving the modular structure of the system (VanVliet, 2000). The long-term effect of corrective, adaptive and perfective changes increases the system's complexity. As a large program is continuously changed, its complexity, which reflects deteriorating structure, increases unless work is done to maintain or reduce it. The change is usually initiated from within the maintenance organization with the intention of making programs easier to understand and hence facilitating future maintenance work (Grubb, 1996). Examples of preventive change include restructuring and optimizing code and updating documentation.

Recommendations for further work

This project demonstrated that sales management and analysis can be improved by developing a hybrid system with various functions like sales ranking and predictions that are not part of some old industry technology solutions. In this project the sales predictions and Agent sales ranking are some major improvement to sales management and tracking especially in an expanding agent footprint. Due to time constraints the system was not developed as fully-fledged software the analyst have made recommendations that can be implemented as further work. The future work required to extend the system may include the following:

- a) Auto suspension to the Agent account which has balance of at least 50% of the sales not cashed.
- b) The sales data to show form of payment i.e. Cash, EcoCash and Bank and any discount auto-calculation when the premium is high. This will be effectively handled when the company is running promotions.

5.6 Conclusion

Throughout this study, the concept of online sales management and analytics has been reflected. The development of a Hybrid Sales Analytics (SalesPro) was achievement in improving insurance sales tracking and monitoring through the use of sales ranking dashboards, cumulative group total sales and future sales predictions which can be graphed on the dashboard. This chapter discussed the system testing and implementation in length. It presents the basic testing strategies which are top-down strategy and bottom-up strategy, the various levels of system testing which are unit testing, module testing, sub system testing and integration testing. A direct cutover was used as the conversion strategy and during the process; users were trained to use the new system. The maintenance strategies to be used are corrective, adaptive, perfective and preventive maintenances. The primary purpose of implementing this system was to analyse historical insurance sales data availability in order to manage well the remittance, and agent performance in various product offerings. As part of further work, some features can be added to this application such as auto suspension of Agent accounts which have the cumulative balance of at least 50% of the agent sales not remitted.

References

- Abdallah Al-Sukairi, (2004) Advanced Database Systems Lecture Notes, King Fahd University of Petroleum & Minerals.
- Ben Linders (2016). “How Database Administration Fits into DevOps”
- Beyond-Davies, Paul (2004) Database System. Basingstoke, UK: Palgrave: Houndmills ISBN 1403916012
- Cronje (2002) Management Principles: A Contemporary Edition for Africa, Published by Juta & Companies.
- Coronel, Carlos; Morris, Steven; Rob, Peter (2011). Database Systems: Design, Implementation, and Management (9th ed.). Cengage Learning. Chapter 5. ISBN 978-0-538-46968-5.
- Dolores R. Wallace and Roger U. Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standards, Fujii, NIST Special Publication 500-165)
- Elmasri R and Navathe S. (2011), Fundamentals of database systems 6th Edition, Pearson Addison –Wesley
- Elmasri, Ramez; Navathe, Shamkant B. (2011). Fundamentals of Database Systems (6th ed.).
- Fowler (2003) UML Distilled: A Brief Guide to the Standard Object Modeling Language
- Fankam, C, (2008) OntoDB2: Support of Multiple Ontology Models within Ontology. In: Proceedings of the EDBT 2008 PhD Workshop. Co-located with the 11th International Conference on Extending Database Technology
- Gary B. Shelly, Harry J. Rosenblatt, (2011) Systems Analysis and Design,
- Gibson C. (1992) Evaluation of second generation, New York : Berkeley
- Gibson M and Hughes C. (1994) System Analysis and Design .A comprehensive Methodology with case, Massachusetts Boyd and Fraser
- Harold Kerzner, (2012) Project Management: A Systems Approach to Planning, Scheduling and Controlling
- Hannu Jaakkola and Bernhard Thalheim. (2011) “Architecture”-driven modelling methodologies. Anneli Heimburger et al. IOS Press. P.98

Hoffer J. A Jeffrey A, Joey F, George, Josephs Valaich (2002) Modern systems Analysis and Design, 3rd Edition , Prentice Hall, Inc

Hughs Bob and Cotterell Mike (1999), Software Project Management, 2nd Edition , McGraw- Hill Publishing company, UK

Institute of Electrical and Electronics Engineers, Inc (IEEE). (2006), IEEE Standard for Software Maintenance. IEEE Std 1219-2006 , New York.

International Journal of Computer Science and Information Security (IJCSIS) (2009) Vol. 4, No. 1 & 2, A Step towards Software Corrective Maintenance: Using RCM model

Jean, S., Pierra, G., Aït-Ameur, Y (2007). Domain Ontologies: a Database-Oriented Analysis. In: Filipe, J., Cordeiro, J., Pedrosa, V. (eds.) WEBIST 2005 and WEBIST 2006. LNBIP, vol. 1, pp. 238–254. Springer, Heidelberg

Justin Zobel (2013). "Algorithms" in Writing for Computer Science (second edition). Springer. ISBN 1-85233-802-4.

Kendell K.E. and Kendell J.E. (2002) Systems Analysis and Design, Published by Pearson Education, Asia.

Koopman Philip (2011) Embedded Software Testing, Carnegie Mellon

Lightstone Sam, Toby J. Teorey and Tom Nadeau, (2007) Physical Database Design: The Database Professional's Guide to Exploiting indexes views storage.

Miriam Webster (1992) Business Management forum Published by Federal Street Press America

Mittal, Prashant (2009) “Data Architecture Standards” Global India Publications. Page 314.

Pearson/Addison Wesley. Chapters 8 and 9. ISBN 0-136-08620-9

Post, G., Anderson, D., (2006). Management information systems: Solving business problems with information technology. (4th ed.). New York: McGraw-Hill Irwin.

Rajaraman V. (2002) Analysis and Design of Information Systems Published by Prentice Hall of India , New Delhi

Russ-Eft D. and Preskill H.(2001) , Evaluation in organisations, New York Basic Books

Ruth Malan and Dana Bredemeyer,(2001) white paper Defining Non-Functional Requirements, Bredemeyer Consulting <http://www.bredemeyer.com/papers>

Shelly .A (2016) Build a Better Vision Structure, Rowman and Little field.

Sparks, Geoffrey. "Database Modelling in UML". Retrieved 8 September 2011

Scott Workinger (2014)“Introduction to Transformational Systems Engineering”. INCOSE Enchantment Chapter, Albuquerque.

Sugumaran, V., Storey, V.C(2006). The role of domain ontologies in database design: An ontology management and conceptual modeling environment. ACM Transactions on Database Systems (TODS) 31(3), 1064–1094

Takang, A.A., and Grubb, P.A. (1996). Software Maintenance Concepts and Practice. Thompson Computer Press London, UK.

Van Vliet, H. (2000). Software Engineering: Principles and Practices, 2nd Edition, Wiley & Sons Publishing, West Sussex, England.

Wright, Joe; Jim Harmening (2009) "15" Computer and Information Security Handbook Morgan Kaufmann Publications Elsevier Inc p. 257

Whitten, Jeffrey L, Bentley, Lonnie D, Dittman, Kevin C (2004). Fundamentals of system analysis and design methods.

Westbrook, L. (2000) Identifying and analysing user needs: a complete handbook and ready-to-use assessment workbook. Neal-Schuman,

Ziv Baida (October 2012) Stakeholders and Their Concerns In Software Architectures, Vrije Universiteit, Amsterdam, The Netherlands. Accessed on http://eprints.ucl.ac.uk/744/1/1.7_stake.pdf

(<http://www.visual-paradigm.com/product/vpuml/tutorials.jsp>) accessed on 14 September, 2012

<http://www.mysql.com>

<http://www.caes.uga.edu/center/caed/>

<http://www.cs.cmu.edu/afs/cs/project/edrc-ballista/www/> accessed on 24 September 2012

<http://www.cs.jmu.edu/users/foxcj/cs555/Unit12/Testing/index.htm> slide show of software testing. Accessed on 24 September 2012

APPENDIX A

User manual

This user manual will guide the users to log in the system for their first time, and will be a reference guide for many of the daily operations.

System Login

1. Log in to <http://127.0.0.1:8080/sales/> on web browser to access the system.

(Recommended browsers are Mozilla Firefox and Google chrome)

2. Use these details to log as Admin:

Username: **admin**

Password: **xxx**

User management

3. To create a new Agent, Click Agents > Create new Agent then a window will be displayed to enter new agent details. Then click  button to save the new record.
4. To create a new system user Click User Management > Create New user.

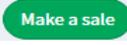
On the new user window, fill in all details and select the user type, then click the  button to save the record.

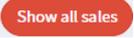
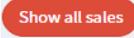
Operations

5. To assign an Agent products to sell, navigate to Products Assignment > create new product assignment. Search Agent Name the tick any of the available select box to assign the product list.

Click the  button to save the entries

6. To process a sale, you must be logged in as an agent. Go to Sales > Sell a product.

Fill in the form with the required information and click  button to process the transaction

7. To process new funds remittance, log on as Analyst, then navigate to Remittances > New Funds remittances, a form will appear requesting you to select the Agent name.
8. When you select the Agent name, another window will be displayed with the gross total owing and the text field for the amount to be paid on that particular time. Enter the text fields accordingly and click the **Process** button.
9. To produce a report for Total Agents, go to Reports>Agent listing report. You can customise the report by selecting the Registration year, city or suburb and the click  or  to show all Agents.
10. To produce a report for Sales by Product, go to Reports >Sales by product type. You can customise the report by product name, start date and end date and click the  or  to show all sales.
11. To produce a Sales report by Region, go to Reports >Sales by Region. You can customise the report by city name, suburb name and click the  or  to show all sales.
12. To produce a Sales Remittance report, go to Reports >funds remittance report. You can customise the report by selecting the remittance condition, start date and end date, then click the  or  to show all remittances.
13. To show the sales forecast for the next three months, log as Analyst. Navigate to Reports >Sales forecast report. The forecast will be populated for each product offering for the next three months.
14. To change your password, while you are logged in click user management > Change my password. On the window put your current password and the new password. Click the update password button.

APPENDIX B: INTERVIEW CHECKLIST

Interview Questions

Supervisor

1. What kind of data do you need to record every day?

2. How long do you take recording this data?

Agent

1. Do you enjoy working with the current manual system and if so what are the strengths?

2. What problems are you facing using the current system?

3. What view or opinion do you have about migrating from the current system?

4. What expectations of the proposed system do you suggest?

Checklist Section A: Staff

1. How do you rate the current system?

Excellent Good Fair Poor

2. Given the choice to migrate from the current system to the new system, would you choose to migrate?

Yes No

If No, what may be the reason _____

3. Do you often experience very busy days and huge workloads during your operations?

Very often Sometimes All the time

4. Have some of employees complained about the current system?

Yes No

If YES, what were the complaints?

5. What problems are you facing with the current system?

Section B: Clients

1. Are you satisfied with the current system that you have been using all along?

Yes No

If YES, give the reason

2. How do you rate the current system?

Excellent Good Fair Poor

3. Are you pleased with the processing speed of the current system?

Yes No

If No what are the suggestions? _____

APPENDIX C: QUESTIONAIRES

04 March 2017

Dear Participant

I am De-Villiers Nyahuma and I am a student at Midlands State University. For my final project, I am developing a hybrid revenue data system analysis for Regal Insurance company it will be able to achieve better results each day.

I am inviting you to participate in this research study by completing the attached surveys. The following questionnaire will require approximately two days completing. In order to ensure that all information will remain confidential, please do not include your name. If you choose to participate in this project, please answer all questions as honestly as possible and return the completed questionnaires promptly. Participation is strictly voluntary and you may refuse to participate at any time.

Thank you for taking the time to assist me in my educational endeavors.

Yours Sincerely

De-Villiers Nyahuma

+263777946399

APPENDIX D: SNIPPET OF CODE

a. Product Assignment

```
<?php
class Controller_ProductAssignments extends Controller_Base
{
    public function action_index()
    {
        $data['productassignments'] = Model_Productassignment::find('all');
        $this->template->title = "Productassignments";
        $this->template->content = View::forge('productassignments/index', $data);
    }
    public function action_view($id = null)
    {
        is_null($id) and Response::redirect('productassignments');
        if ( ! $data['productassignment'] = Model_Productassignment::find($id) )
        {
            Session::set_flash('error', 'Could not find productassignment #' . $id);
            Response::redirect('productassignments');
        }
        $this->template->title = "Productassignment";
        $this->template->content = View::forge('productassignments/view', $data);
    }
    public function action_create()
    {
        if (Input::method() == 'POST')
        {
            Debug::dump($_POST); exit;

            $val = Model_Productassignment::validate('create');

            //validate product selection, at least one should be selected
            if ($val->run())
            {
                //lets insert the selected products
                $products = $_POST['product'];
                $comm = $_POST['comm'];

                $count = 0;
                for($x = 0; $x < count($products); $x++){

                    // @todo fix irregularities that occur if you skip one
                    product inbetween

                    $comm_value= $comm[$x]/100;
                    if(empty($products[$x])) continue;
                }
            }
        }
    }
}
```

```

        $productassignment
Model_Productassignment::forge(array(
        'agent_id' => Input::post('agent_id'),
        'product_id' => $products[$x],
        'commission' => $comm_value,
    ));
    if($productassignment->save()) $count++;
}
if ($count > 0)
{
    Session::set_flash('success', "productassignment added
successfully. $count products have been allocated ");

    Response::redirect('productassignments');
}

else
{
    Session::set_flash('error', 'Could not create
productassignment. Please try again');
}
}
else
{
    Session::set_flash('error', $val->error());
}
}

View::set_global('products',Model_Product::query()->where('is_active',1)-
>get());
$this->template->title = "Productassignments";
$this->template->content = View::forge('productassignments/create');
}

public function action_edit($id = null)
{
    is_null($id) and Response::redirect('productassignments');

    if ( ! $productassignment = Model_Productassignment::find($id))
    {
        Session::set_flash('error', 'Could not find productassignment #.$id);
        Response::redirect('productassignments');
    }

    $val = Model_Productassignment::validate('edit');

    if ($val->run())
    {
        $prod = $_POST['product'];
        $comm = $_POST['comm'];
    }
}

```

```

//Debug::dump($prod[0]); exit;
//$productassignment->agent_id = Uri::segment(3);
$productassignment->product_id = $prod[0];
$productassignment->commission = $comm[0];

if ($productassignment->save())
{
    Session::set_flash('success', 'productassignment updated
successfully.');
```

Response::redirect('productassignments');

```

}

else
{
    Session::set_flash('error', 'Could not update productassignment.
Please try again.');
```

}

```

else
{
    if (Input::method() == 'POST')
    {
        $productassignment->agent_id = $val->validated('agent_id');
        $productassignment->product_id = $val-
>validated('product_id');
        $productassignment->commission = $val-
>validated('commission');
```

Session::set_flash('error', \$val->error());

```

    }
    $this->template->set_global('productassignment', $productassignment,
false);
}

$this->template->title = "Productassignments";
$this->template->content = View::forge('productassignments/edit');
```

}

```

public function action_delete($id = null)
{
    is_null($id) and Response::redirect('productassignments');
```

if (\$productassignment = Model_Productassignment::find(\$id))

```

    {
        $productassignment->delete();
        Session::set_flash('success', 'productassignment deleted successfully.');
```

}

```

    else
    {
```

```

        Session::set_flash('error', 'Could not delete productassignment . Please
try again');
    }
    Response::redirect('productassignments');
}
}

```

b. Insurance sale

```

<?php
class Controller_Sales extends Controller_Base
{

    public function action_index()
    {
        $sales = Model_Sale::find('all');
        $data['sales'] = $sales;
        $this->template->title = "Sales";
        $this->template->content = View::forge('sales/index', $data);
    }

    public function action_create()
    {
        if (Input::method() == 'POST')
        {
            $val = Model_Sale::validate('create');

            if ($val->run())
            {
                list($uid) = Auth::get_user_id();
                //@todo forge a proper method to find agent id
                $agent_id = $uid;

                //then find agent id from the given user id.
                $tmpagent = Model_Agent::query()->where('user_id',$uid)-
>get_one();

                $agent_id= $tmpagent->id;

                //sale value: get from product record
                $prod = Model_Product::find(Input::post('product_id'));

                $months = Input::post('months');
                $annual = $prod->annual_premium;
                $sale = round(($annual/12)*$months,2);

                //@todo calculate commission payable, for this agent, and this
product
                $comm_value=
Model_Commission::compute_payable($agent_id,Input::post('product_id'), $sale);

```

```

//@todoTo create a message here egReg for Altezza ADJ2030
successful. Insurance Code is 20034567
$msg = "You have successfully paid for your insurance.Thank
you!";

```

```

$sale = Model_Sale::forge(array(
    'agent_id' => $agent_id,
    'product_id' => Input::post('product_id'),
    'months_insured' => $months,
    'sale_value' => $sale,
    'cust_name' => Input::post('cust_name'),
    'cust_phone' => Input::post('cust_phone'),
    'cust_identification' => Input::post('cust_identification'),
    'notification_msg' => $msg,
    'prefer_sms' => Input::post('notify'),
    'sms_sent' => 0,
));

```

```

if ($sale and $sale->save())
{

```

```

    // @todo record
    $props = array(
        'sale_id' => $sale->id,
        'comm_value' => $comm_value
    );
    Model_Commission::forge($props)->save();

```

```

    //@todosendsms notification
    $notify = Input::param('notify');

```

```

    $result = 0;
    if($notify == 1){

```

```

        $to = '263'. (int)(Input::param('cust_phone'));
        $msg_ready = urlencode($msg);

```

```

        $api_res =
"http://app.vipersms.co/api?senderid=SALESPRO&cmd=send&username=salesproapi&pass
word=pass123&gsm=".$to."&msg=".$msg_ready;

```

```

        $result = @file_get_contents($api_res);
    }

```

```

    //@todo update the sale record to indicate that the SMS
was sent or not

```

```

    if($result > 0 ){
        $sale->sms_sent = 1;
        $sale->save();
    }

```

```

//done

```

```

        Session::set_flash('success', 'sale made successfully. ');
        Response::redirect('sales');
    }

    else
    {
        Session::set_flash('error', 'Could not create sale. Please
try again');
    }
}
else
{
    Session::set_flash('error', $val->error());
}
}

$this->template->title = "Sales";
$this->template->content = View::forge('sales/create');

}

public function action_delete($id = null)
{
    is_null($id) and Response::redirect('sales');

    if ($sale = Model_Sale::find($id))
    {
        $sale->delete();

        Session::set_flash('success', 'sale deleted successfully. ');
    }
    else
    {
        Session::set_flash('error', 'Could not delete sale . Please try again');
    }
    Response::redirect('sales');
}
}
}

```

c. Sale remittance

```

<?php echo render('utils/search_able'); ?>
<div class="col-md-12 col-sm-12 col-xs-12">
    <div class="x_panel">

        <div class="x_content">

            <?php if ($remittances): ?>
            <table class="table table-striped" id="tbl-paginated">

```

```

        <thead>
            <tr>
                <th class="col-md-1">RecordID</th>
                <th class="col-md-3">Agent Name</th>
                <th class="col-md-2">Remit By</th>
                <th>Amt Expected</th>
                <th>Amt Paid</th>
                <th>Amt Owing</th>
            </tr>
        </thead>
        <tbody>
<?foreach ($remittances as $remittance): ?>                <tr>
<td><?php echo $remittance->id; ?></td>
                <td><?php echo $remittance->agent->first_name.' '$remittance-
>agent->last_name.'<br/>'.strtoupper($remittance->agent->agent_code); ?></td>
                <td><?php echo $remittance->remit_by; ?></td>
                <td><?php echo 'USD ' .number_format($remittance-
>amt_expected,2); ?></td>
                <td><?php echo 'USD ' .number_format($remittance->amt_paid,2);
?></td>
                <td>
                    <?php if($remittance->amt_owing > 0): ?>
                        <span class="label label-danger"><i class="fa fa-
dollar"></i></span>
                    <?php else: ?>
                        <span class="label label-success"><i class="fa fa-
check"></i></span>
                    <?phpendif; ?>
                    <?php echo 'USD ' .number_format($remittance-
>amt_owing,2); ?>
                </td>
            </tr>
        <?phpendforeach; ?> </tbody>
</table>

<?php else: ?>
<div class="alert alert-danger">
    <p>There are no Remittances. </p>
</div>
<?phpendif; ?>                </div>
</div>
</div>

```