

MIMOSA MINE ONLINE DIET PRESCRIPTION EXPERT SYSTEM



MAKAITA TENDWA CHIKOKA

R132910T



MIMOSA MINE ONLINE DIET PRESCRIPTION EXPERT SYSTEM

MAKAITA TENDWA CHIKOKA

R132910T

Submitted in partial fulfilment of the requirements for the degree of

BSc Information Systems Honours Degree

Department Information Systems and Computer science in the

Faculty of science and technology at the

Midlands State University

Gweru

November 2014

Table of Contents

CHAPTER1

List Of Acronomy	4
List of Tables	7
1.0INTRODUCTION	9
1.1BACKGROUND OF THE STUDY	9
1.2ORGANOGRAM	11
1.3MIMOSA BUSSINESS STATEMENTS	12
1.4STATEMENT OF THE PROBLEM	12
1.5OBJECTIVE OF THE STUDY	13
1.6INSTRUMENTS.....	13
1.7JUSTIFICATION	14
1.8CONCLUSION.....	14
2.0 INTRODUCTION	15
2.1 BUSINESS VALUE	15
2.1.1 Business needs and functionality	15
2.2 FEASIBILITY STUDY	16
2.2.3 Hardware requisitions	17
2.2.4 Economic Viability	18
2.2.5 Cost Benefit Exploration-analysis	18
2.3 RISK ANALYSIS.....	22
2.4 PROJECT WORK PLAN AND SCHEDULE.....	23
2.5 CONCLUSION.....	24
3.0 INTRODUCTION	25
3.1 INFORMATION GATHERING	25
3.2 ANALYSIS OF CURRENT SYSTEM	28
3.2 DATA ANALYSIS.....	29
3.3WEAKNESS OF THE CURRENT SYSTEM.....	32
3.4EVALUATION OF ALTERNATIVES.....	32
3.4.1 In house Development	32
3.4.2 Outsourcing.....	33
3.5 REQUIREMENTS ANALYSIS	34
3.6 FUNCTIONAL REQUIREMENTS	34

3.7 CONCLUSION.....	38
4.0 INTRODUCTION	39
4.1 SYSTEM DESIGN	39
4.2 FUNCTIONALITY OF THE NEW SYSTEM.....	40
4.3 CONTEXT DIAGRAM.....	40
4.4 DATAFLOW DIAGRAM.....	42
4.5 SYSTEM ACHITECTURE DESIGN	44
4.6 THE PHYSICAL DESIGN ARCHITECTURE OF THE SYSTEM.....	46
4.7DATABASE DESIGN.....	47
4.7.1 Database architecture	49
4.7.2 Logical Design	49
4.7.3 Program Design	52
4.8 SYSTEM CLASS DIAGRAM	52
4.8.2 Interface Design	56
4.8.3 Input design.....	56
4.8.4 Output design.....	59
4.9 CONCLUSION.....	60
5.0 INTRODUCTION	61
5.1CODING	61
5.2 TESTING.....	62
5.2.1 Unit Testing	63
5.2.2 Module testing	64
5.2.3 Subsystem	65
5.2.4 System testing	65
5.2.5 Acceptance testing	65
5.3 TEST STRATEGIES.....	66
5.3.1 Verification	66
5.3.2 Validation.....	66
5.4 INSTALLATION	69
5.4.1 User training.....	69
5.4.2 File conversion.....	69
5.4.3 Sub system changeover.....	69
5.5 RECOMMENDATIONS	71
5.6 MAINTENANCE	71

5.6.1 Corrective maintenance.....	71
5.6.2 Preventive maintenance	71
5.6.3 Adaptive maintenance.....	72
5.7 RECOMMENDATIONS FOR MAINTENANCE.....	72
5.8 USER MANUAL.....	72
5.9 RECOMMENDATIONS	72
Bibliography	74
User Manual	75

List Of Acronomy

ARR	Accounting rate of return
GUI	Graphical user interface
ERR	Entity relationship diagram
LAN	Local Area Network
NPV	Net Present Value
ROI	Return On Investment
DFD	Data Flow Diagram

List of Figures

Fig 1.0 Map showing the geological location of Mimosa.....	7
Fig 1.1 Pie chart showing Mimosa Minerals	8
Fig 1.2Diagram showing hierachical structure	9
Fig 2.0 Hardware requisition	15
Fig 2.1 Software requisition.....	16
Fig 2.2 Cost benefit analysis.....	17
Fig 2.3 Project payback.....	18
Fig 2.4 Formula for ROI.....	19
Fig 2.5 Net Present value	20
Fig 2.6 Project Work Plan.....	22
Fig 2.7 Gantt chart	22
Fig 3.0 Context diagram of proposed system	28
Fig 3.1 Dataflow diagram for the proposed system.....	29
Fig 3.2Activity diagram for the proposed system.....	32
Fig 3.3 Enhanced activity diagram	34
Fig 4.0 Context diagram of existing system	39
Fig 4.1 Dataflow diagram of proposed system	41
Fig 4.2 Client server architcture.....	43
Fig 4.3 Physical design of proposed system	45
Fig 4.4 Ansi-sparc architecture	46
Fig 4.5 EER diagram.....	50
Fig 4.6 Class diagram of proposed system	51
Fig 4.7 Sequence Diagram(a)	52
Fig 4.8 Sequence Diagram(b)	53
Fig 4.9 Login structure.....	54
Fig 4.10 Longin Interface	55
Fig 4.11 Medical condition Explanation form.....	56
Fig 4.12 Diet Prescription form	57
Fig 5.1 System Testing Procedure	62
Fig 5.2Whitebox testing.....	62
Fig 5.3Blackbox testing	63
Fig 5.4 Longin Error display.....	66

Fig 5.5 Password Error display 66
Fig5.6 Field Selection error 67

List of Tables

Table 1 Interview Objectives27
Table 2 User table.....51
Table 3 Diet table.....51
Table 4 Condition table.....51

List of Appendices

Appendix A User Manual74
Appendix B Interview questions.....87
Appendix C Interview questions.....88
Appendix D Observation score sheet.....89
Appendix E Code Source90

1.0 INTRODUCTION

Diet prescriber is an expert system which streamlines or simplifies the task of a dietician. Traditionally diet prescription and nutritional evaluation procedures are possible by consulting the area experts, periodicals and text books. This system allows for individual nutritional assessment applying the anthropometric methods such as weight, height, blood group and other factors like stress, work load and classification of work in terms of mobility. The system also turns possible to prescribe and calculate diets and recommendation besides having an actual database which may be changed or modified. On the basis of available cumulative information, the system will display the name and the prescribed diet of the client. This system not only simplifies task of the dietician but also helps the client by providing information to modify by eradicating, subsidizing, or increasing specific nutrients in the diet as part of treatment for a disease or a clinical condition.

1.1 BACKGROUND OF THE STUDY

Mimosa Mining Company is entirely being owned by Mimosa investments limited. It is a Zimbabwean company jointly owned by Aquarius Platinum Limited (Australia) and Impala Platinum.

Mimosa Mining Company is situated on the Wedza Environmental Multipart on the Zimbabwean Prodigious Dyke east of Bulawayo and 33 ½km west of Zvishavane town.

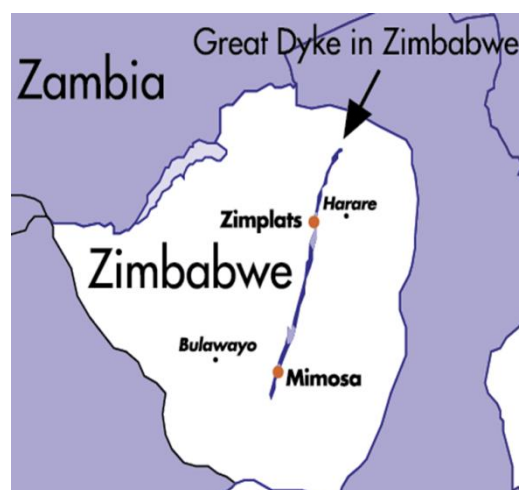


Fig 1.0 Map showing the geological location of Mimosa

The operations done at the mine comprises of a sleuth underground mine that is gained access to by a decline shaft, and a concentrator. The Mining Company is a mining and mineral

processing company. The concentrates extracted as ore is further sent to South Africa for further processing. Minerals that are economically vital to Mimosa are:

- Platinum Group Metal Minerals

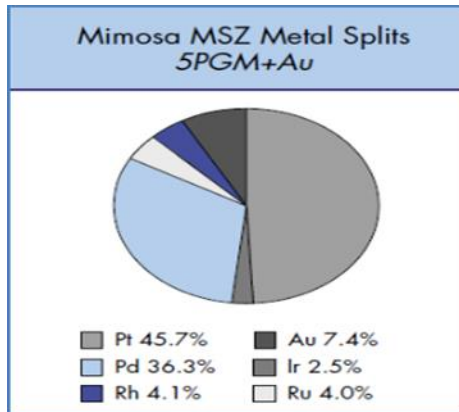


Fig 1.1 Pie chart showing Mimosa minerals

Containing the metals

- Platinum (Pt)
- Palladium (Pd)
- Rhodium (Rh)
- Iridium (Ir)
- Ruthenium (Ru)

- Base Metal Minerals

Containing the metals

- Base Metal Minerals
- *Containing the metals*
- Copper (Cu)
- Nickel (Ni)
- Gold (Au)

At the mine there is a clinic that provides health services for the mine employees as well as the beneficiaries. All employees are given medical aid cards by the company so no cash transactions are done at the clinic.

The clinic comprise of a few nurses, one radiographer, a pharmacist and one doctor. Mine employees often visit the clinic with problems that arise in long term due to the working environment of the mine. These problems comprise of hearing disorders, chest infections, eye

problems as well as back-aches despite other medical conditions that mainly occur in patients and require numerous nutrient alterations in short term. However this requires a specific therapeutic diet to make sure all diet necessities are included for example metabolic diets, cystic fibrosis, paediatric diabetes.

As it is a mining clinic, it is bound to provide around the clock support and service for the mine workers for them to be healthy and always alert for work.

1.2ORGANOGRAM

Mimosa mine has a stratified management structure. It has three directors as well as general managers and the rest of the management team in their respective hierarchical order.

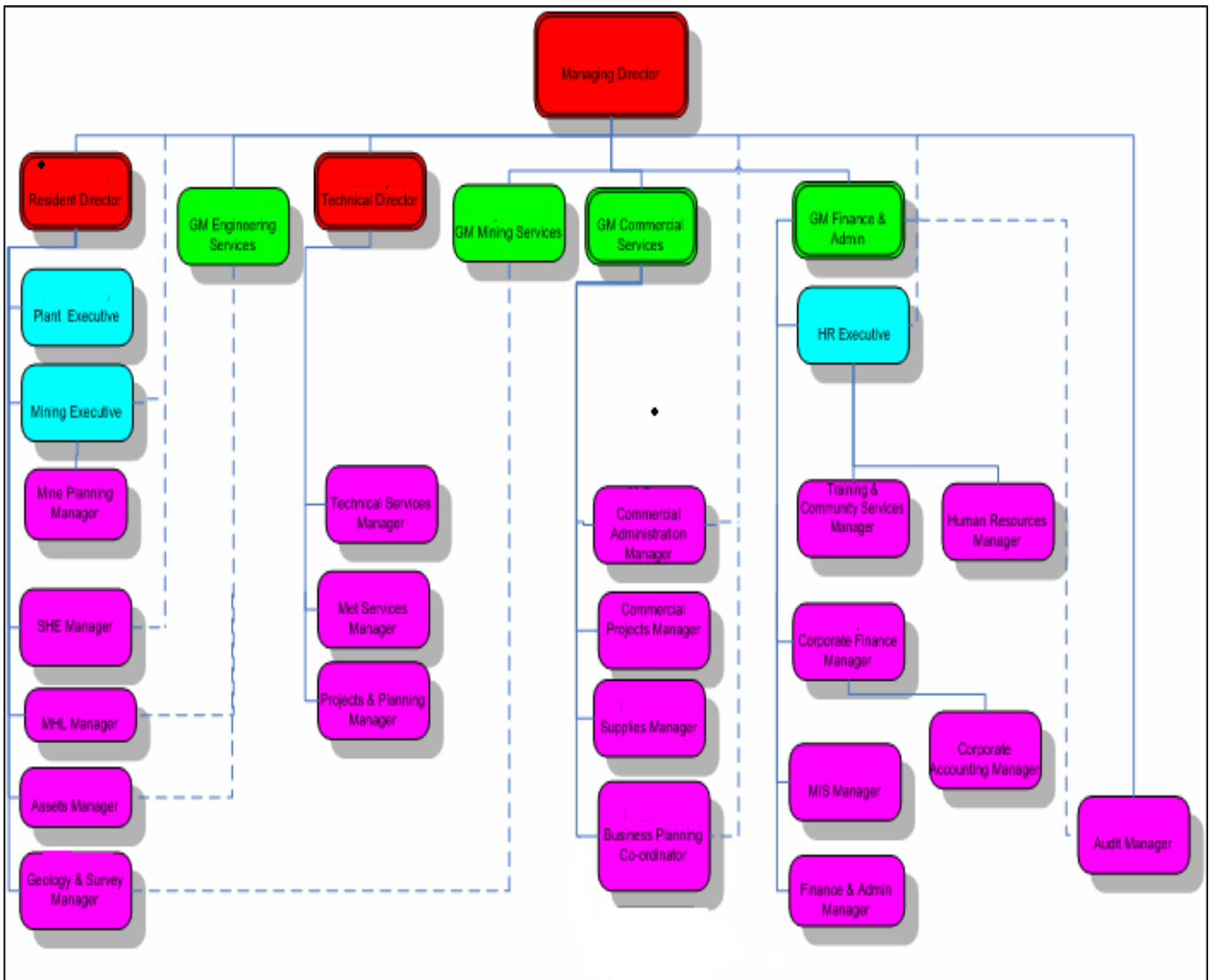


Fig 1.2 Diagram showing the company hierarchical structure

1.3 MIMOSA BUSINESS STATEMENTS

Vision

Ray Hackney (2010), Strategic Information Systems Management refers vision as what a company would like to accomplish, its core values and how it will operate to meet its targets.

Mimosa mining company vision is

- To remain in the bottommost cost quartile of Platinum Producers to deliver growth and greater yields to the Investors.

Mission

Mary Sumner (1998), Management Information Systems 4th Edition refers mission as a goal, purpose or reason for existence of a company.

To achieve the vision of Mimosa, a purposeful focus on the following themes is vital.

- To utilize resources efficiently and responsibly.
- To conduct business in an environmentally responsible manner.
- To safeguard the Health and Safety of all stakeholders.
- To continuously improve all systems and processes and recognize superior performance.
- To foster a mutually beneficial relationship with the community.

1.4 STATEMENT OF THE PROBLEM

Mimosa mining company has a part of its mission that strives to safeguard the health and safety of all its stakeholders in order to achieve its vision. In order to fulfil this mission, the mining company has a clinic that offers medical services to the mine employees although it does not have a proper hospital system.

As a mining company, it works in shifts thus day and night. Therefore at the mine there is only one doctor, one radiographer, a pharmacist, a few nurses and no specialist or dietician. However as there is a shortage of dieticians in Zimbabwe, many people lack the knowledge of how to handle their diet life. As the mine comprise of hard working men and women tirelessly day and night, these employees need to be guided or informed health wise on what to eat as they would be working in order to prevent sudden illnesses caused by food. There are many types of diets which can improve health conditions of patients and without a

knowledge base of these therapeutic diets it will be difficult to diagnose and prescribe them without being a specialist. Medical staff and the mine employees can utilise the combined expert services of a Medical doctor and a Dietician online. Therefore by introducing this proposed online diet prescription system, it will also diagnose patients with more than one medical condition i.e. a patient can be HIV positive, Diabetic and Hypertensive at the same time. The system will have the privilege to prescribe the therapeutic diet considering all the given medical conditions. The project is being designed keeping in mind the objective of enhancing the diet therapy process for the mine employees.

1.5 OBJECTIVE OF THE STUDY

Kevin Grant, David Edgar (1998), Management Information Systems, 4th edition refers objectives as specific outcomes that a system aims to achieve or accomplish within a time frame and having the resources. The objectives of the proposed system are:

- To calculate patients' nutritional requirements using standard equations based on assessments of the age, height, weight, nature of job and other relevant factors.
- To prescribe suitable diets to employees online.
- To inform mine healthcare professionals (SHE) as well as to educate the mine employees about food and nutrition issues.
- To provide researchers with a huge and up-to-date repository of information regarding various diets.
- To provide an expert system that acts in place of a dietician.

1.6 INSTRUMENTS

- C# - an object oriented programming language that is easy to configure and integrate with other programming languages.
- MYSQL - Accessible open source software that however can provide or initiate a database system that can be straightforwardly be networked.
- Apache Web Server – that provide easy connection and access to the database. It work well with other programming languages like Java and PHP.
- MS Office 2010 – will provide a good platform to type and compute some word documents for the system.

1.7JUSTIFICATION

Concise Oxford Dictionary (2001) defines justification as a way of explaining or convincing someone to support, believe or accept a certain occurrence. The Online diet prescription system will help achieve one of the company's missions that strive to continuously improve all systems and processes and recognize superior performance as well as improving the health and safety of the employees. By adapting to the new clinical system, it will help improve the employees' health there by resulting in happy and healthy works. The system will allow a therapeutic diet prescriber representing an innovative shift in the field of health and providing a repository knowledge base of Dieticians. The system benefits users in that it brings the services of a Dietician within the click of a button providing any health wise or dietary support they may need. By adapting to this system, information will reach everyone who is registered with the system timely and accurately providing an answer to shortcomings of the current system were by a dietician is not there at the mine clinic.

1.8CONCLUSION

The system developer clearly explained the problems of the current system and the merits of taking up the proposed system clearly outlining the objectives of the system. The next stage would be to plan for the proposed system so as to evaluate the viability of it being considered.

2.0 INTRODUCTION

The main reasons for carrying out the analysis phase are centred on striving to express how to come up with a viable project that will add value to the company operations. In this chapter, a

feasibility study is to be carried out in order to quantify the degree of worthiness of taking up the proposed system. A course of action is then drafted to execute and develop the system if the results would come positive thus the merits outweighing the de-merits

2.1 BUSINESS VALUE

Kevin Grant, David Edgar (1998), defines business value as the expected merits that an organisation will receive by taking up the project. These business core values must be understood and well established. A clarification to the management must be done about the feasibility and evaluation of the project as they fund the development of the system. The basic problem of business valuation is how to set a value on all the assets of a business, thus may include the intangibles. Several business valuation methods can be used to tackle the problem and attempt to determine a fair price for the business to be added value.

2.1.1 Business needs and functionality

Business needs

- An effective health online diet prescription for the mine employees.
- A health wise system that will meet the company's' mission statements and vision.

System functionality

The system should be able to partake the following functions

- Register users online
- The major system processes are to prescribe appropriate diets to users, plan appropriate diets to the users as well as calculate the body mass index.
- In terms of reports, the system should be able to generate a diet prescription report and generate custom reports for statistics of visits per given point in time.
- The system should provide a validation of inputs such as date violation thus it must not allow text where numbers are expected and the other way round.
- The administration tools should include creation of user accounts and their management as well as granting of privileges to users.
- The system should provide a certain number of trials when logging in thus 3times.

2.2 FEASIBILITY STUDY

According to *Kevin grant, David Edgar (1998)*, it is the way of analysing and evaluating the proposed system whether it is technically viable as well as being viable within the estimated costs and if it will be cost-effective. The study also attempts to decide whether the project can be developed within the targeted time limit. An acceptable project has merits that outweigh its de-merits.

Some interviews and questioners were done at the mine in order to evaluate the acceptance of the project. The following was the outcome;

Operational viability

- **The users** – these include the mine employees, the doctor and other health officials at the mine. They really recommended the development of the online diet prescription system as very viable and helpful as it would improve the health service delivery to the employees and everyone at large.
- **The management** – in this case the management include the IT department as well as the mine executives and directors. They were all confident about the new system as it will help attain their mission and vision of the company. The IT department was confident as well that it will be able to implement, maintain and constantly upgrade the system.

Feasibility overview

As the questionnaires were distributed around the mine, the results gathered clearly showed that the whole company is in support of the online diet prescription system. It however shows that the project is operationally feasible.

Organisational feasibility

The objective is to evaluate and survey if the company stakeholders are in support of the project. The results of the survey are as follows;

- **The employees** – as they are the major stakeholders of Mimosa mine, they really appreciated the system as it will improve their diet lives and occupational health in general.
- **The executives and managers** – they were satisfied by the system as it will help achieve their goals and missions. They fund the project and rated the system as a good initiative towards the digital world in health sector.
- **IT developers** – they are happy about the system as it would be a chance to show their skills and knowledge on terms of developing and maintaining systems.

Over view

The survey outlined that there is a possibility of organisational support in all areas as the stakeholders were happy about the system.

Technical viability

James O'Brian (2006), refers technical feasibility as a way of surveying whether the project is achievable technically. The study also looks at whether the company will meet the required hardware and software components needed to take up the project.

- **The system developers** – the technical team did a lot of courses and were exposed to different programming languages such as PHP, Java, C# as well as MySQL mentioning but a few.
- **Users** – all authorized computer users at the mine are computer literate there by making it easier to implement the online diet prescription system.
- **Upkeep** - the IT team at the mine has gone under several courses that include the safekeeping and maintenance of systems. Therefore include the disaster recovery plans.

The above results clearly show that the project is technically viable as well. There is a need to take up the project with doubt of technical or organisational feasibility.

2.2.3 Hardware requisitions

The table below illustrates the hardware requirements needed by the system.

Item	Specification
Processor	Core i7
Ram	4GB
Hard drive	500GB
Portable Media	External USB Hard Drive
Backup Media	Cloud server

Fig 2.0 Hardware requisition

Software Requisitions

Item	Specifications
Operating System	Windows 7 Premium
Browser	Mozilla Firefox or Opera mini
Database	MySQL
Webserver	Apache
Language	C#

Fig 2.1 Software requisition

2.2.4 Economic Viability

Ray Hackney(2010), suggest that economic viability is a process of analysing the project costs of operating and taking up the project against the profits that will be incurred when the project is fully running and developed. It looks again into the economic gains incurred before, during and after implementation. Risks must be considered, evaluated and weighed against other viable alternatives and using cost-benefit-analysis there should be able to weigh the benefits against the costs. The costs can be operational-costs or development costs.

Development cost

These types of expenses are incurred at the genesis of the project.

Operational cost

These expenses of this nature are experienced whilst operating the new system.

2.2.5 Cost Benefit Exploration-analysis

According to *Lucey T (1996), Costing 5th Edition*, defines cost benefit exploration-analysis as the expenses or costs that are going to be used against the rewards that the system is likely to

bring in the organisation. A viable system should be expensive at the initial stages of the project and yield constant benefits in the long run.

YEAR ►	2014	2015	2016
Benefits	USD	USD	USD
Service Value Addition	8 000	10 000	12 000
Increased Efficiency	7 000	10 000	15 000
Increased Security	9 000	11 000	14 000
Reduced Stationery	2 500	2 500	2 500
Total Benefits	13 000	33 500	43 500
Development Costs			
Labor (technical)	4 000	0	0
Licensed Software	800	0	0
Additional Hardware	1 800	0	0
User Training	1 500	0	0
Total Developmental Costs	8 100	0	0
Operational Costs			
Maintenance	5 000	5 000	10 000
Stationary	600	800	1 000
Other Costs	5 000	7 000	10 000
Total	10 600	12 800	21 000
Total Costs	18 700	12 800	21 000
Total Benefits	13 000	33 500	43 500
Profit/Loss	(5 700)	20 700	22 500

Fig 2.2 Cost benefit analysis table.

2.2.6 Investment evaluation

Methods that can be used to evaluate projects on the basis of cash flows are Return on investments and net profit.

Net profit

Lucey T (1996), defines it as the discrepancy between all the total expenses and total revenue of taking up a project over its life duration. The calculations of net profit for this project are bellow;

$$\begin{aligned} \text{a) Net Profit} &= \text{Total Benefits} - \text{Total Costs} \\ &= (13\,000 + 33\,500 + 43\,500) - 18\,700 \\ &= 90\,000 - 18\,700 \\ &= 71\,300 \end{aligned}$$

Payback period

Year	Project (USD) for Mimosa Mine
0	-18 700
1	13 000
2	33 500
3	43 500

Fig 2.3 Project payback

According to *Lucey T (1996)*, refers payback period as the time taken by a project to yield back the capital invested at the start of the project.

b) Payback Period:

To recover **USD18 700** which is the initial investment amount, it will take Total Amount of Year 1 which is \$13 000 + 5 700/33 500 (12mnths) = This would take 1Year and 2Months.

Return on investment

Lucey T (1996), defines return on investment as a method that is used to compare the net profit against the investment required. It is greatly used as a cost benefit evaluating tool.

ROI = (Average Annual Profit / Total Investment) * 100
where; Average Annual Profit = NP / # of years

Fig 2.4 Formulae for ROI according to Lucey T (1996),

Therefore for this project the calculations are as follows;

$$\begin{aligned}
 \text{Total Annual Profit} &= 90\,000 - 18\,700 \\
 &= 71\,300 \\
 \text{Total Average Annual Profit} &= 71\,300 / 3 \\
 &= \underline{23\,767} \\
 \text{Therefore Return On Investment} &= \frac{23\,767}{18\,700} \times 100\% \\
 &= \underline{127.09\%}
 \end{aligned}$$

The results show a positive ROI therefore thus the project is favourable.

Net present value

Lucey T (1996), defines net present value as

Net Present Value based on a discount factor of 15%:

Discount Factor = $1 / (1 + r)^t$; where r = Discount rate and t = time

Present Value = Value in Year x Discount Factor

Net Present Value = Total of Present Values

Below are the net present value calculations;

Year	Value in Year	Discount Factor	Present Value
------	---------------	-----------------	---------------

0	-18 700	1	- 18 700
1	13 000	0.87	11 304
2	33 500	0.76	25 331
3	43 500	0.66	26 602
Net Present Value			44 537

Fig 2.5 Net-Present Value of the project.

Over view of the economic feasibility

After critically analysing and evaluating the viability of the project economically, it hereby shows that the project is viable to undertake. The benefits of taking up the project outweigh the expenses that will be incurred.

2.3 RISK ANALYSIS

Ray Hackney(2010), strategic information systems management defines risk as analysing the possible factors that can hinder the success of the project. The risk evaluating process include risk identification, risk classification or risk prioritization and risk alleviation procedures.

People related risks – various software projects fail to meet requirements due to risks that are caused by lack of technical knowhow paying particular attention to the system developers. Due to pressure of other commitments at work, the system developers may fail to deliver the correct system specifications on time many projects fail to take off due to these kind of risks. This type of a risk can be mitigated by employing full time developers of the project such that their main focus would be to develop the project only.

Still on people related risks, some key developers of the project may leave the organisation whilst the project is half way through. This type of a risk can be mitigated by having contingency plans before taking up the project or just to have a stand-by skilled consultant for the project.

Time or deadline limits – various projects fail due to unrealistic time limits that are set at the beginning of the project. Due to this problem, the developers would take up the system at a fast pace so as to just meet the deadline but putting the project at stake. At the end the

developers produce a poorly developed project because of time. To solve this problem, there is need to closely monitor the project against time as it progresses. Realistic time schedules must be set at the beginning of the project is that every task is allocated each portion of time.

User resistance – this is the most fatal jeopardy that several systems face. People are resistance to change when it comes to taking up a new system. The users should be educated so that they appreciate the good that the new system will deliver to them. In order to eradicate this type of a problem, user involvement from the start to the end of the project in the new system would be the best.

Budget constraints - many projects when struck with this type of problem, they entirely abandon the project. This is caused by a failure in proper budgeting strategies or methods that are used at the beginning of the project. It may be greatly associated by changes in the economic costs of developing the project. A cost viable analysis of this project was done to evaluate the costs of developing this project. These costs however can be affected by economic changes such as inflation or exchange rates.

Project management risks – when taking up the project, authority should be assigned to the correct recipients using the proper communication channels. Tasks should be delegated on time and accountability of every action should be of high importance.

2.4 PROJECT WORK PLAN AND SCHEDULE

After analysing and getting the results that the project is feasible, a work plan should then be established. The work plan is there to show the general outline of the project activities to be done and their time periods. A work plan is completed by a Gantt chart that is used for activity scheduling. It is a road map for the developers to see what time frame is to be spent on a certain stage up until the project is done.

According to *Gary B. Shelly and Harry J. Rosenblatt (2010)*, *Systems Analysis and Design*, 8th Edition a Gantt chart may involve allocating activities to team members and the time frames.

On this project, the waterfall model is going to be used which involves the following stages.

ACTIVITY	START DATE	END DATE
Project Proposal	24.08.14	28.08.14
Planning	31.08.14	04.09.14
Analysis	07.09.14	18.09.14
Design	14.09.14	25.09.14
Coding	28.09.14	09.10.14
Testing	12.10.14	16.10.14
Implementation	19.10.14	30.10.14
Maintenance	02.11.14	On Going

Fig 2.6 Project work plan

The above table can be further illustrated using a chart below;

Tasks / Weeks	1	2	3	4	5	6	7	8	9	10	11	12
Project Proposal	█				█	█	█	█				
Planning		█			█	█	█	█				
Analysis			█	█	█	█	█	█				
Design				█	█	█	█	█				
Coding					█	█	█	█				
Testing					█	█	█	█				
Implementation					█	█	█	█	█	█		
Maintenance					█	█	█	█	█	█	█	█
Documentation	█	█	█	█	█	█	█	█	█	█	█	█

Fig 2.7 Gantt chart

2.5 CONCLUSION

After viably planning for the project in this chapter, the next stage would be to critically analysis the system as per the requirements. The system should now be critically scrutinised to have a clear understanding of how the proposed system will operate against the current system.

3.0 INTRODUCTION

Garry B Shelly and Harry J. Rosenblant (2010) systems analysis and design 8th Edition refers analysis phase as were by information about users wants and needs are gathered and build more models in-depth of what they can expect to accomplish with the new proposed system. This document defines management and user requirements, expenses and their alternative plans, and any recommendations that may be done. There is further understanding of the business requirements and a creation of a logical model of the new system. System requirements document is the end result of the analysis phase. To get an in-depth insight of what users require, some information gathering has to be carried out by the project developer.

3.1 INFORMATION GATHERING

Burkan, S., (2005), The Art of Project Management 1st Edition, refers information gathering as were by the detailed study of the current system is done for reasons of gathering facts about the current system as well as the merits and the short comings of the system. By gathering this information, there would be a better understanding of the current system and the user needs on the proposed system.

- (i) Participatory - The researcher assisted the mine health practioners and the workers to just have a firsthand experience and knowledge about the system. Through this engagement, the researcher managed to collect detail about the system operations.
- (ii) Surveys – *Dennis B.H Wixom and Roberta M. Roth (2012)* defines surveys as the method of critically collecting data in the form of questionnaires and communicating to many people getting their views upon a project.

Observation and document review.

Kenneth E. Kendall and Julie E. Kendall (2011) suggest that observation is critically examining or viewing a sequence or procedure and record the outcomes. This involves observing the processes that are involved from the start when login on to the system and viewing how the users use the system.

Advantages

- Provided direct information about behaviour of individuals and groups
- Provided good opportunities for identifying unexpected results
- Permitted the evaluator to enter into and understand situations or context of the system.
- Existed in unstructured, natural and flexible setting

- It was fast as the observer had to understand the situation without further explanation from anyone.

Disadvantages

- Affected the behaviour of the participants as they did not fully understand what the observer was doing.
- Data may be distorted by the selective perception of the observer.
- The observer had little control over the proceedings that took place.
- Behaviour which was observed may be abnormal due to participants not familiar with working under supervision.

Interviews

Frey J. H and Oishi S. M (1995) suggests that interview is a conversation that is done between two or more individuals where by questions are asked by an interviewer and the responses being given by an interviewee. In this fact finding technique all the interviewees are selected and appointments are made. The selected persons are then interviewed personally and their responses are recorded.

Interview conducted for knowledge gathering had the following objectives.

Knowledge gathering

Interviewees	Date	Time	Place	Objectives
SHE representatives	10.09.14	1000HRS	SHE Administration	-To know how they are currently working to improve the wellbeing of the mine employees. -To get user’s views and comments about the current system operatives.

Table:1 Interview objectives

Results for interview

The representatives outlined that:

- Their current work is tiresome because of shortage of professional health advisors as well as shortage of dieticians in Zimbabwe
- There was a suggestion to develop a computerised system.
- The system does not produce any reports making it difficult to make decisions.

Merits

- This method is good for probing unstructured questions.
- The researcher can adjust and re-phase questions depending on the user being interviewed and can reiterate in case a user happens to have misunderstood a question.
- The body language of the interviewee can easily be observed.
- Response rate is high since they is a scheduling of meetings
- Personal contacts are very important in getting the cooperation of the people involved in giving them the feeling of having made a substantial contribution towards the design of a new system

De-merits

- It is difficult to select interviewees, as this may hurt the feelings of the other people that would not have been selected for the interviews.
- It is difficult to analyse and compare results because of interviews tailored subjective nature.
- System users might not feel that open, hence they might not cooperate fully and put their ideas orally.
- This type of gathering method is time consuming therefore makes the process of data collection slow.
- The data collected depend upon the experience of the interviewers.

Questionnaire

Frey J. H and Oishi S. M (1995), defines questionnaires as the process of preparing standard questions that would be printed on paper and issued to the users of the system who are supposed to answer the questions on the spaces provided on the paper. The researcher distributed 10 questionnaires to users so that they fill in the information.

Merits

- They are regarded as less expensive compared to interviews
- They are convenient for scattered population.
- No prior arrangements are needed when conducting them.
- It has the advantage of containing pre-formulated questions and answers to which are essential for the development of the system under consideration

De-merits

- The papers can be poorly answered there by resulting in biased data collection.
- They are time consuming as results come after a long period of time as compared to interviews.
- Some people may answer the questions reluctantly because they are not supervised.

3.2 ANALYSIS OF CURRENT SYSTEM

The current system is currently manual were by a patient visits the doctor for diet advices if and only if the patient is seriously sick. Prior checks of health conditions are done at the mine when one joints the company. However people relax and as they would be working day and night for months or years that they tend to engage in improper and mislead diets until one gets sick then visit the clinic for consultation.

Requirements analysis

Requirements analysis according to *Lewis J. (2006) Fundamentals of Project Management 3rd Edition* is an iterative process that includes intermingling with the user, project team, and other stakeholders to attain a thorough understanding of each of the collected requirements and how they will affect the project.

Inputs include

- Name
- Age
- Gender
- Weight
- Height
- Stress Factor
- Food Allergies

Processes

- Diet Prescribing

Outputs

- Prescribed diet
- Patients' nutrition requirements.

Roles of the user include

- Login using username and password
- Register with the system
- Enter required details
- Post details
- View prescribed diet
- Logout of the system

Administrator

- Login using username and password
- Manages the users
- Runs reports
- Logout of the system.

3.2 DATA ANALYSIS

According to *Kenneth E. Kendall and Julie E. Kendall (2011)*, data analysis as a process of critically scrutinising gathered data, samples or models converting it to useful information. The developer used the context diagram and data flow diagram to analyse the current system.

Context diagram

Lewis, J., (2006) Fundamentals of Project Management 3rd Edition suggests that Context diagrams shows the communication or interaction between the system and the users. It however shows the whole system as a single unit process.

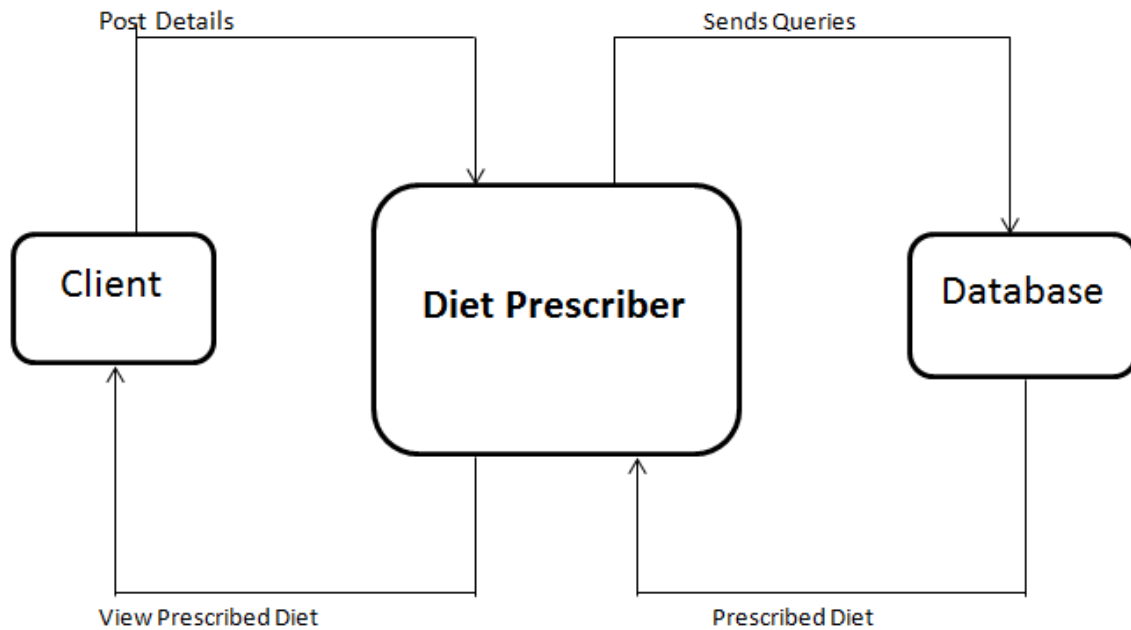


Fig 3.0 Context diagram of proposed system

Dataflow diagram

Lewis J. (2006) Fundamentals of Project Management 3rd Edition refers dataflow diagram as a process or sequence of data represented in the form of a diagram. The data on the diagram would be representing how the information or process of the system will follow.

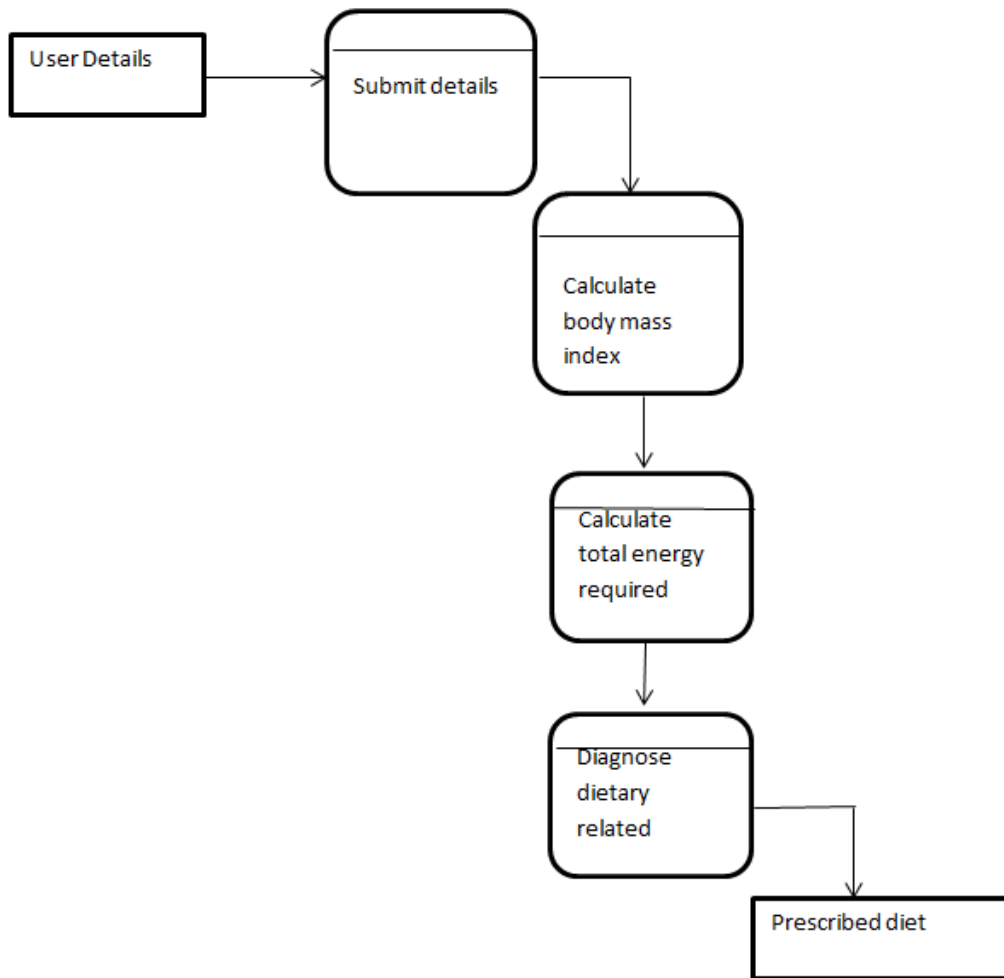


Fig 3.1 Dataflow diagram of proposed system.

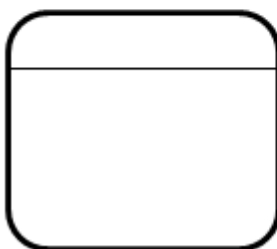
Key



Entity



Flow of Information



Process

3.3 WEAKNESS OF THE CURRENT SYSTEM

The existing system is a human being and can suffer from the following human errors and these may include

1. Tiredness
2. Memory limitations
3. Both the Dietician and the client should be present.

3.4 EVALUATION OF ALTERNATIVES

There are many systems evaluative that can be used for developing a computer system. These evaluation techniques help to choose the best alternative that would assist on reducing development expenses. The various alternatives are outlined below;

3.4.1 In house Development

Schwalbe, K (2006) refers to in house development as a process where by a system is developed within the company by the internal IT personnel's. Specific specifications of the users are taken into consideration and the system users feel the complete ownership of the system. Although developing a software internally within the company is challenging, the company will however be building its employee skills and experience. In house development keep up with the company budgetary constraints and the system will be implemented on the existing infrastructure.

In-house development has the following merits

- The company will assume total ownership of the new system as it would have been developed internally.
- It is regarded as less expensive as the resources and skills are acquired within the company.
- The new system will face less resistance from the users and management because it would have been internally developed.
- Maintaining of the system is easy as it would have been done internally.
- The user requirements would be clear making the users accept the system.

In house development has the following demerits

- Developing systems internally is usually for small projects

- The project can collapse if a key member resigns or withdraws at the middle of the project.
- If there is no proper coordination the system can be sub-standard.

3.4.2 Outsourcing

Concise Oxford Dictionary (2001) defines outsourcing as when a company subcontract or purchase services from an outside supplier. Another alternative is to approach outside companies or firms to handle part of the development workload on a short term or long term basis.

Outsourcing has the following merits

- The company and the developers can focus on other critical business functions.
- There is a chance to develop and implement new technology as the software would be outsourced.
- Costs can be predicted more accurately.
- Development cost can be reduced.
- The level a service can be specified
- There is less time to implement
- Requires less technical staff of the internal company since the vendors are the ones who will install and implement the system.

Outsourcing has the following demerits

- The company will not develop technical expertise in their own employees.
- The IT personnel's will not have control over the system because the system would have been done externally.
- Information can be vulnerable as it would have handed over to external organisation.
- The company may develop to depend on the organisation.
- Training cost is increased as there would be need to train the users to the new system.
- Maintenance of the new system will be difficult as there would require constant consultation from the developers of the system.

3.5 REQUIREMENTS ANALYSIS

Schwalbe, K, (2006), Information Technology Project Management, 6th Edition, refers requirements analysis as collecting information about what the user requires and defining, in the purest likely terms, the difficulty that the project is expected to answer. *Concise Oxford Dictionary (2001)* defines analysis as understanding the functions the project must perform users' business context and limitations, the presentation levels it must observe, and the external systems it must be companionable with. Interviews and cases are techniques that can be used to obtain this understanding. The results of the analysis are gathered in a formal requirements specification. The results serve as input to the next stage.

3.6 FUNCTIONAL REQUIREMENTS

Kenneth E. Kendall and Julie E. Kendall (2011), refers functional requirements as a set of cases that describe all the relations the users will have with the software. In the new system the following users shall be able to;

Clients;

- Login using the username and password.
- Register with the system.
- Enter required details.
- Post details.
- View prescribed diet.
- Logout of the system.

Administrator

- Login using username and password.
- Add new rules and diets.
- Manages the users.
- Manages the knowledgebase.
- Runs reports.
- Logout of the system.

Activity diagram

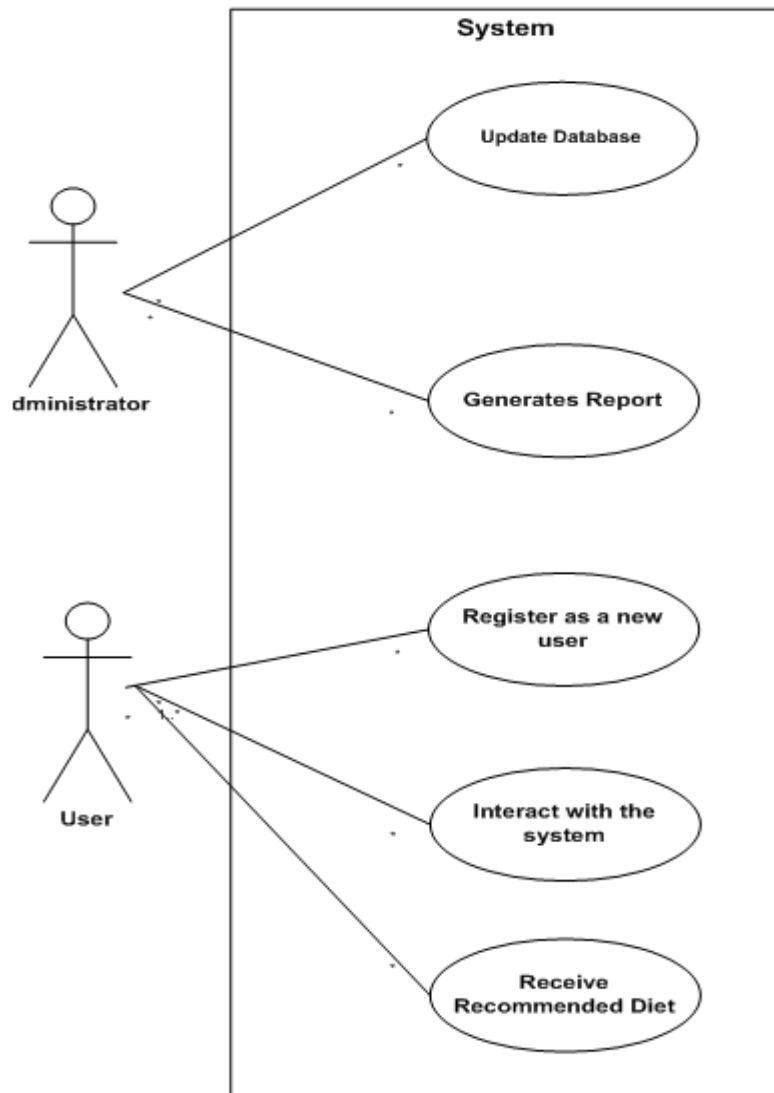
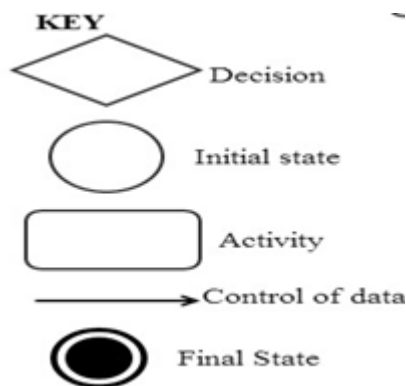


Fig 3.2 Activity diagram of proposed system

Enhanced activity diagram of the Diet prescriber expert system



Fig 3.3 Enhanced diagram for proposed system



3.6.1 Non-functional requirements

(i) User interface and human errors

- The system should be simple
- The user interface must be understandable thus almost self-explanatory.
- The system should have an allowance for self-recovery.
- The system must at most be user friendly.

(ii) Error handling

The system should have an error handling for the following

- Data capturing
- Analysis of data or user details

(iii) Security Issues

- The system should be secure such that only authorized users should have access to the user name and password.
- Access levels should vary with different users to protect company information.
- The password must be long enough to match and meet the company password policy.
- An antivirus system shall be deployed to manage all the machines accessing the server.
- Backup using IBM tapes and storing them in safes or other places from the site

(iv) System efficiency and through put

The proposed system should

- Allow for quick upload and download of information as well as availability of data whenever needed.
- Should always be available.
- The system should show some improvement in terms of time and efficiency.
- There should be reduction of operational costs because of the minimization on the overtime of the IT personals.

(v) Quality service

Quality is very essential when it comes to systems therefore it should be maintained.

The system should be a pace setter in terms of quality.

(vi) Improved client service

To build and improve customer satisfaction and goodwill, the system must be able to serve customers at the comfort of their workplaces.

3.7 CONCLUSION

After having a thorough analysis about the system, the author would recommend that the system should be developed in house. In house is likely to provide and produce a good quality system quickly and it can be used and accepted by the users.

4.0 INTRODUCTION

The following step that follows after finally analysing user requirements and specifications is designing a working system on a stable platform. In this chapter different issues are to be outlined thus include how the system will be developed configured as well as installed. Generally the system design shows a comprehensive outline of computer based solutions, the architectural design, database design as well as the interface design.

4.1 SYSTEM DESIGN

According to *James O Brian(2006)*, system design is where the user requirements are changed into workable software structures. A number of activities are done thus include designing the user interfaces, the knowledge base representation, hardware, reasoning techniques and other programs. The other aim of design is to have a clear view of the system using diagrams that are clearly explained. As has been pointed out in the last chapter, the new system should accomplish and try to solve the shortfalls of the current system.

The design of the new system should cater for the following;

Efficiency

Procedures should be done and run in a short period of time. *Ray Hackney(2010)*, *Strategic Information Systems Management* describe efficiency as the extent where by effort or time is fully exploited for the projected task or purpose.

User friendly

A good system or project should have minimum consultation as well as support.

Functionality

The system should address the required functionalities specified in the user requirements and these should be able to be modified to meet the user requirements.

Security

According to the *Concise Oxford Dictionary (2001)* security is a process or measure that is done so safeguard information or any other form of private ownership. The system should cater for privacy and confidentiality. Only authorised users are to have the access to use the system. User name and passwords are to be used so as to ensure security. Data access and audit trails should be provided by the system. Password should be encrypted.

Performance

The system should function according to the requirements and set standards.

Maintainability

Mary Sumner (1998), management Information Systems 4th Edition propagates maintainability as the ease with which any maintenance activity can be done on a certain equipment. The proposed system should be easy to maintain thus involves even upgrades in terms of technological changes in the business or computer environment.

4.2 FUNCTIONALITY OF THE NEW SYSTEM

The new system shall be comprised of the following modules;

- **Client module:** This module keeps complete information about clients. Every client has to fill in his or her personal data when submitting the problem and will automatically start to enjoy expert services of a Dietitian upon registration.
- **Diet prescriber module:** This module is responsible for providing the prescribed diets to clients when provided with details.

The **Support** module has several sub modules and these are:

- (i) **Body Mass Index Calculator:** This module calculates Body Mass index using clients' Information.
- (ii) **Nutrients Calculator:** This module only calculates required nutrients using Body Mass Index and other relevant factors.
- (iii) **Data Transition:** This module converts all the calculations to a diet prescription.

Administrator Module:

The module keeps complete information about the clients. This module is used to add, delete or update the user information thus however manage user accounts. The administrator is as well responsible of managing the entire system including changing system settings.

4.3 CONTEXT DIAGRAM

Jeffrey .A Hoffer: Morden (2011), defines context diagram as a diagrammatic representation

of the system. The new system will differ from the current system in that it will eliminate the manual process. The context diagram shows the system boundaries, external entities and the entire overview of the system and how it relates to the users of the system.

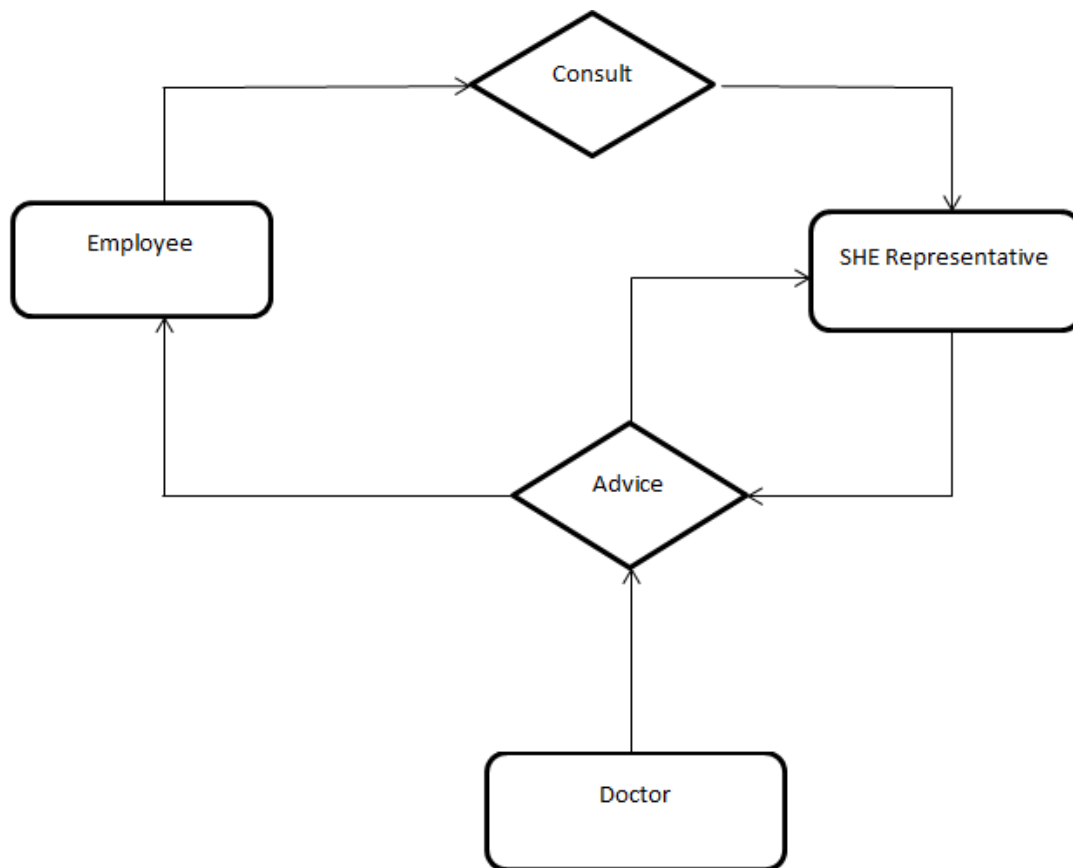
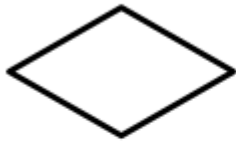


Fig 4.0 Context diagram of existing system.

Key



Relationship



Entity



Flow of information

4.4 DATAFLOW DIAGRAM

According to *Whitten and Bentley, (2007)*, Systems Analysis and Design, DFD is a tool for structured analysis. It graphically shows the movement of data and systems components between them.

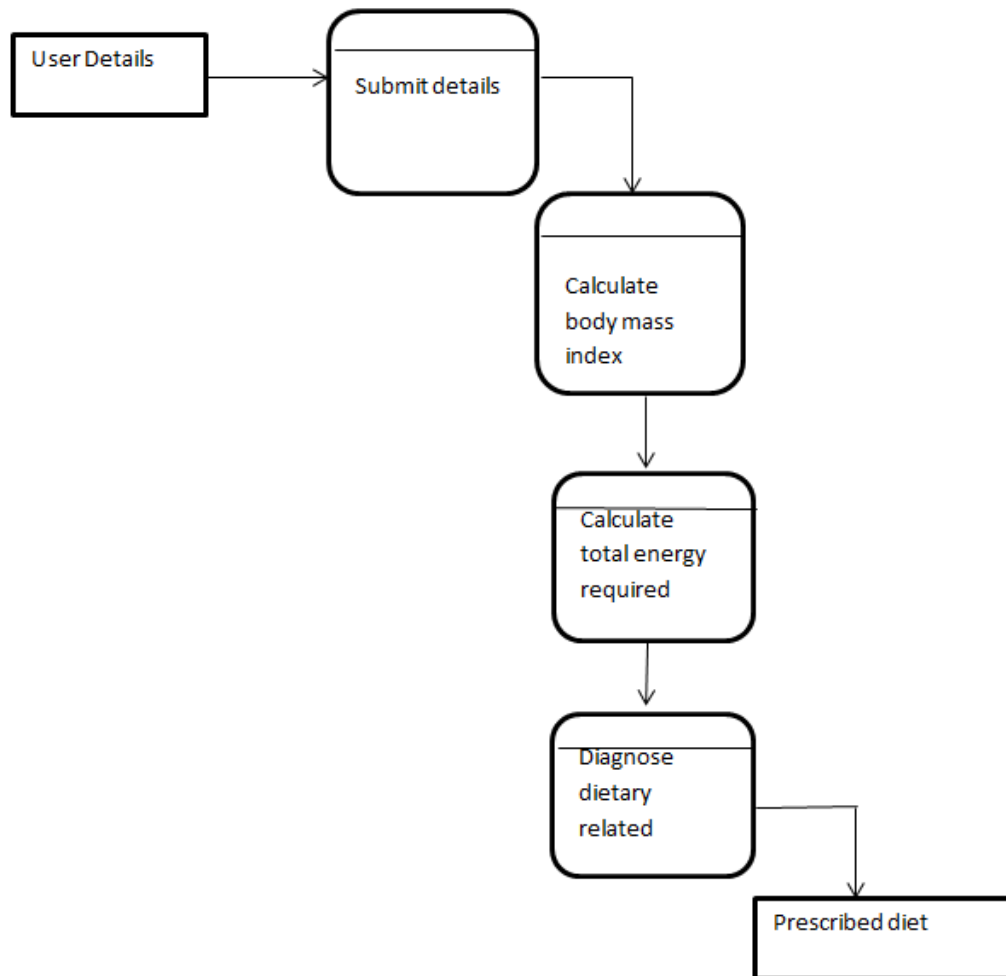
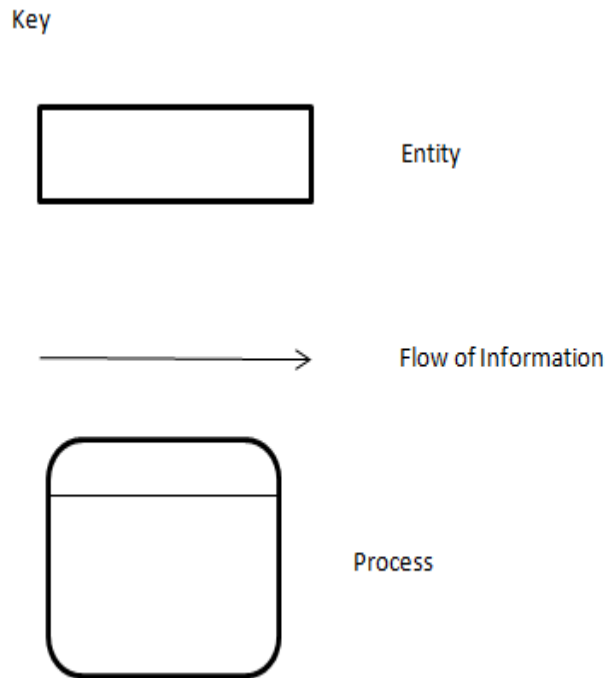


Fig 4.1 Dataflow diagram of proposed system.



4.5 SYSTEM ARCHITECTURE DESIGN

The system that is going to be implemented will conform to a client or sever setup where different clients from within and outside the company will have to access the same database through network connections. This is the architecture proposed project. The following will make up the system.

Client machines

Any machine with access to the internet and has a web browser will access the diet prescriber.

Database server

The database server will be MySQL server. The package facilitates the creation of a database, using C# codes in creation Windows Applications rights.

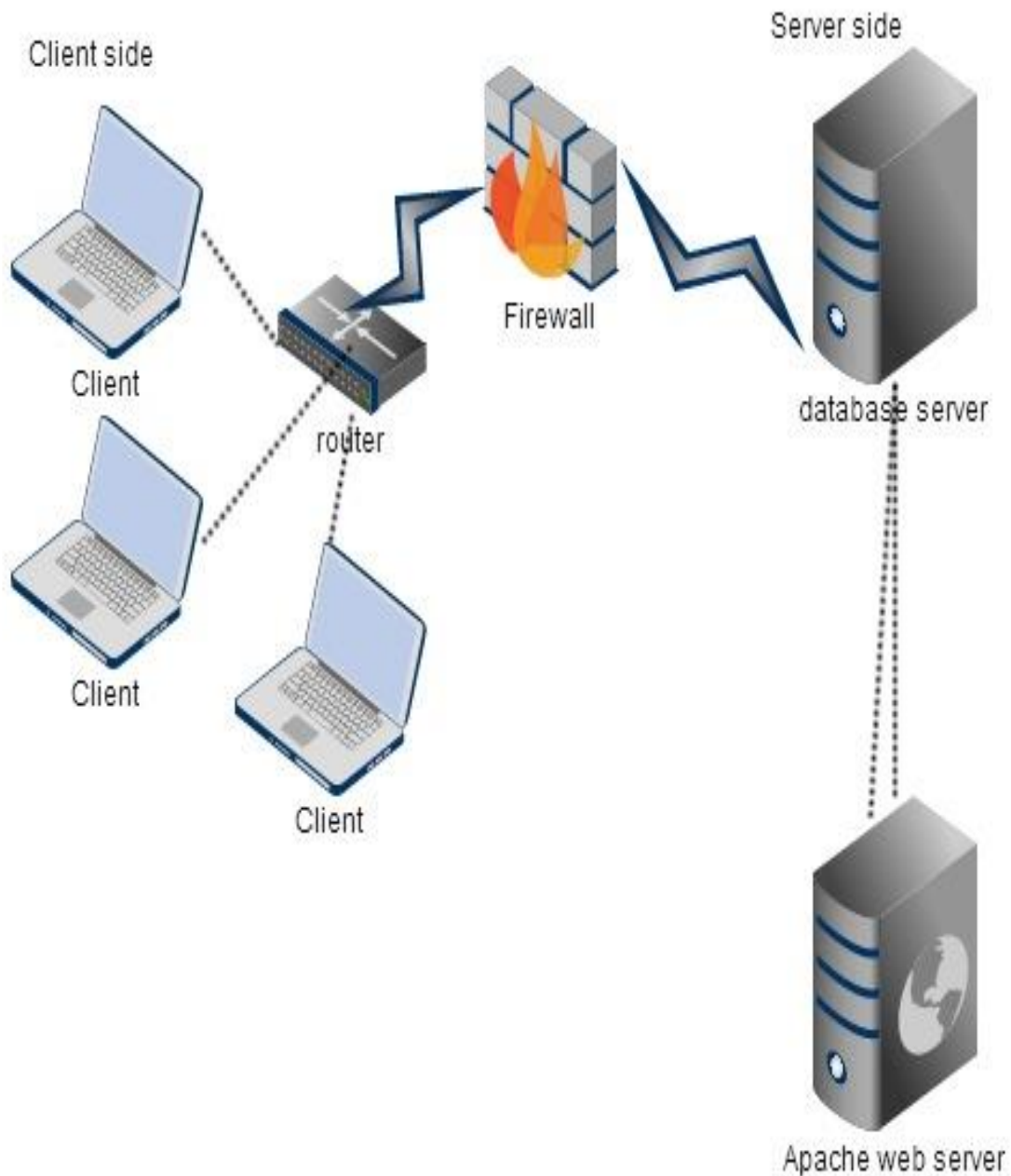


Fig 4.2 Client server architecture

Sommerville. I, (2004), Software testing, refer client server architecture as systems that divide the processes between a central server and one or more networked clients. In this situation the clients server architecture is build or developed using MySQL connectivity database. The system will include the following

Database server

All the data and information is stored here that is used or required. To ensure data consistency and integrity, all important and necessary modifications will be carried out in the database. The proposed system uses MySQL.

Clients machines

They are user workstations that are connected to the internet to allow the users to connect to the system.

4.6 THE PHYSICAL DESIGN ARCHITECTURE OF THE SYSTEM

Whitten and Bentley (2001), refer the physical design as the design of the hardware and software platforms which the proposed system is going to be based on. The proposed system may require a network connection so that other people may be from the Harare branch will be connected sharing a centralized database. A centralized database will enhance security of the entire system hence no data can be hacked. The system will however be run on the existing local area network. The configurations should be accurate to make sure the system runs smoothly. A firewall is installed in between the system server and client computers so as to prevent viruses that may be acquired through the internet.

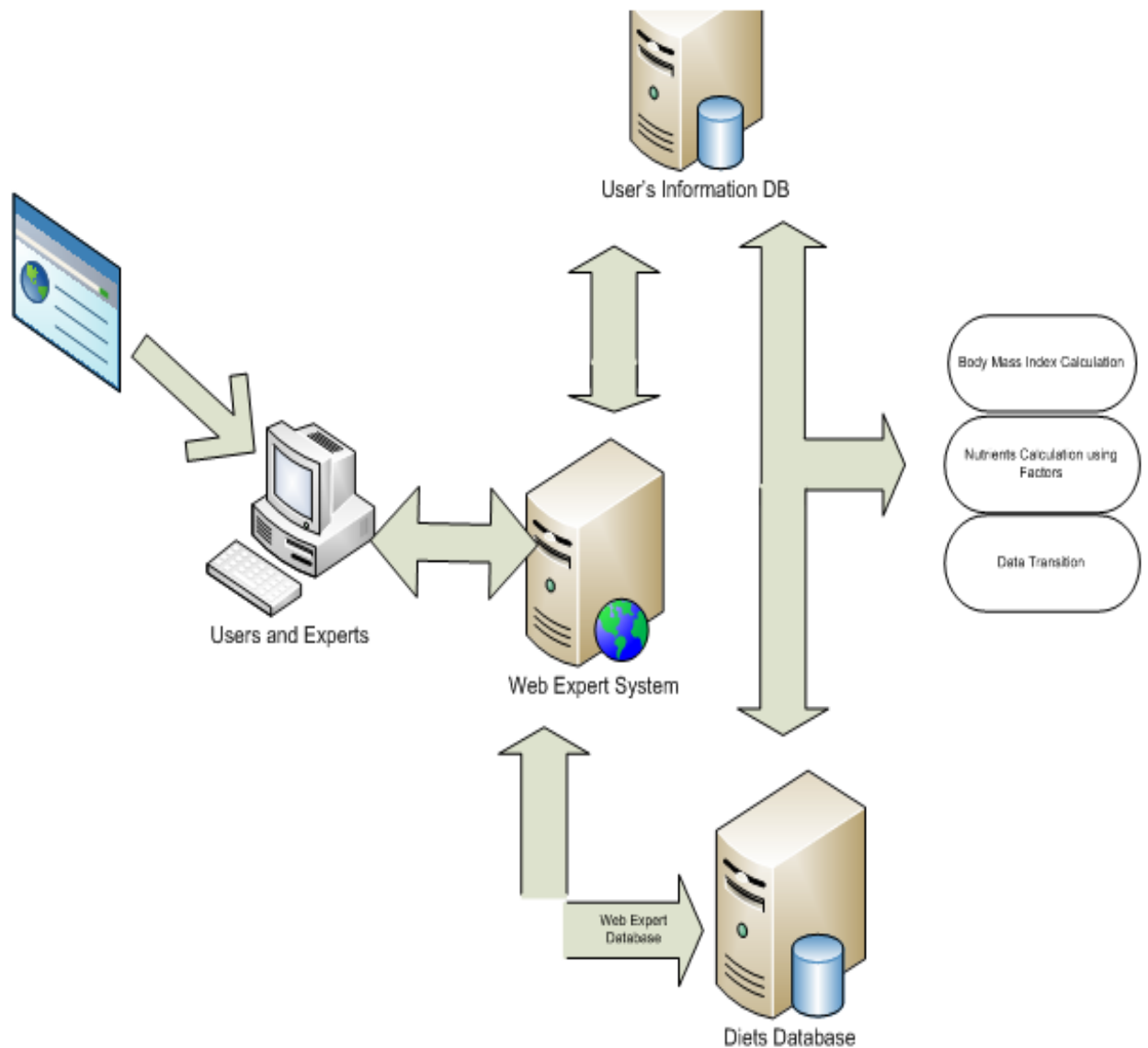


Fig 4.3 Physical design of the proposed system

4.7 DATABASE DESIGN

Sommerville(2011), argues that the core of the entire system is the database. All the information that the system will be working on will be stored in the database in this case which happens to be the MySQL database. Critical planning as well as the design of the database structure will determine how well the entire system will operate. Having shown the pictorial relationship that exists between various entities involved in the system, the table below illustrate the entities thus the full information about each and every entity and how it is composed. The designing of the proposed system should meet the user requirements. For the system to yield success, the data repository should make sure there is data consistency integrity and aim at reducing data redundancy. The database should as well be efficient in its processing thus there is a need to

seriously design the database. The proposed system will adopt the database management system than a file based system because there is integration of files therefore eliminating data redundancy.

By using database management, security is improved thus through the use of strict measures such as password policies. The administrator would be the one possessing the rights to use it.

It gives room for data backups thus in case of disasters or unplanned events. The other reason to use DBMS would be have a data consistent database that will be validated as data is inputted into the system.

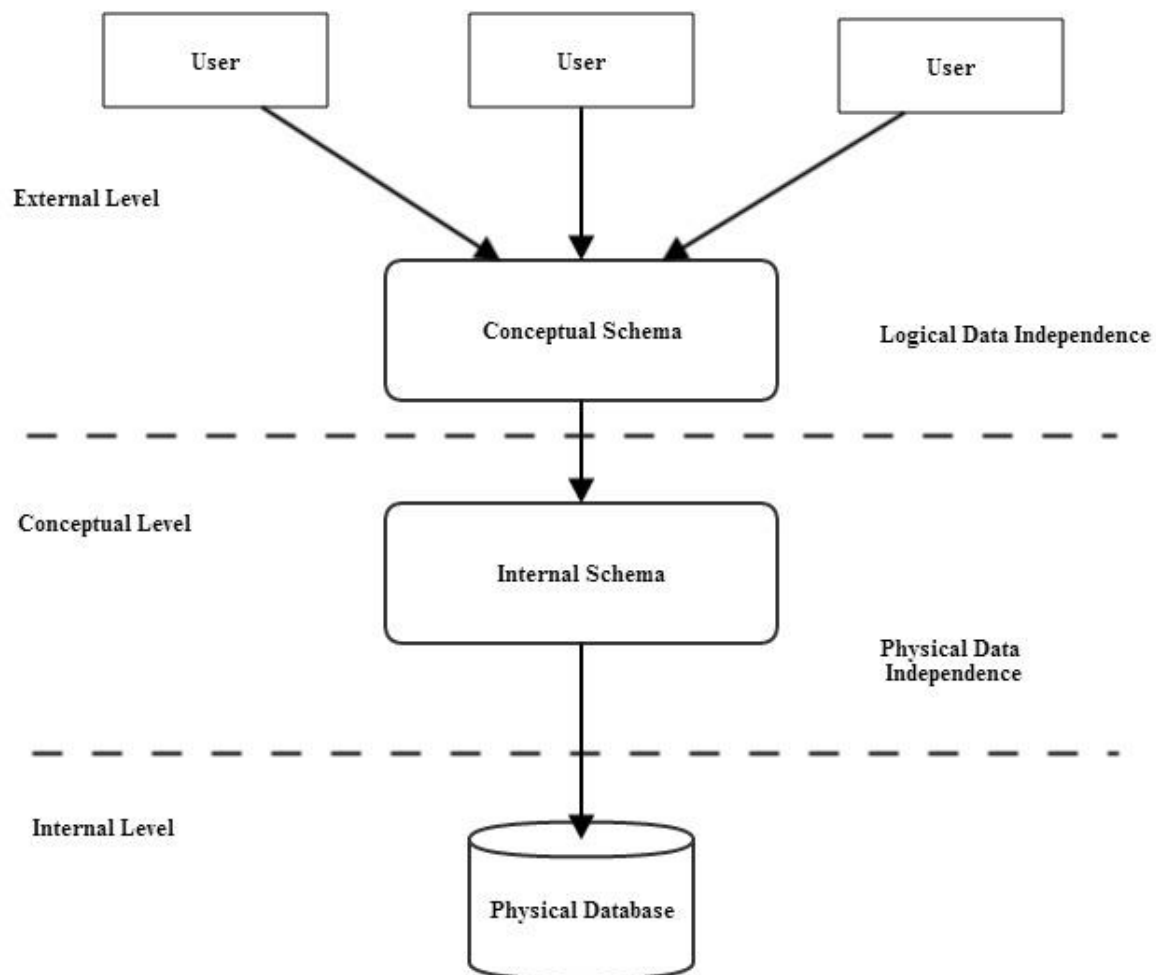


Fig 4.4 Ansi-sparc architecture

External view

The level defines how data is stored and it is the lowest of all the levels. The level is meant to meet the individual interests. On this level, the users and the administrators will be having different opinions over the proposed system.

Conceptual View

The level describes the relationship that exists between the data itself that is actually stored in the database. This is also referred to as the community view or the data relationship model. The data at this level is stored in tables. Queries are defined and how they link with each other.

Internal View

This view expresses the highest level of abstraction. The main motive of this level is to simplify the users' interaction with the database. The interaction is made easy by a provision of a user-friendly interface that the user can contemplate. Records in the database can be viewed using tables, reports or forms.

4.7.1 Database architecture

Database management system works basically with logical components such as tables and functions, views and procedures. The data in the proposed system will be organised into logical components and be visible. The reasons for using relational database are

- It is suitable for all platforms
- DBMS are good at managing data
- By using common characters within the database, data can be matched.
- It supports various data requirements.
- It supports database manipulation operations which include retrieval, deletion and updates.
- Due to improved concurrence access, different users can be able to access the same table in the database concurrently.

4.7.2 Logical Design

Having shown the pictorial relationships that exist between various entities involved in the system, the following table gives the details of the entities that is, full information about each and every entity and how it is composed. Below is a data dictionary showing critical tables in the database

Diet (DID, dietName, ProteinRequired, CarbohydratesRequired, WaterRequired, Restrictions, AdditionalInfo)

User (username, password)

User Info (id, name, surname, age, sex, height, address, password, username, date)

Condition (CID, ConditionName)

User

FIELDNAME	DATATYPE	SIZE
USERNAME	Short Text	(20)
PASSWORD	Short Text	(15)
DEPARTMENT	Text	(20)

Table 2 User table

Diet

FIELDNAME	DATATYPE	SIZE
DID	Integer	(11)
DIETNAME	Integer	(11)
PROTEINREQUIRED	Integer	(5)
CARBOHYDRATESREQUIERD	Integer	(5)
FATSREQUIRED	Integer	(5)
WATERREQUIRED	Integer	(5)
RESTRICTIONS	Text	

Table 3 Diet table

Condition

FIELD	DATATYPE	SIZE
CONDITION ID	Medium integer	(9)
CONDITIONNAME	Varchar	(50)

Table 4 Condition table

Entity Relationship Diagram

James O' Brien, (1996), Management Information Systems, 4th Edition, suggests that an entity-relationship diagram is a specialized graphic that demonstrates the interrelationships between entities in a database. To represent information, symbols are used and these include boxes that represent entities, diamonds that represent relationships and ovals that represent attributes.

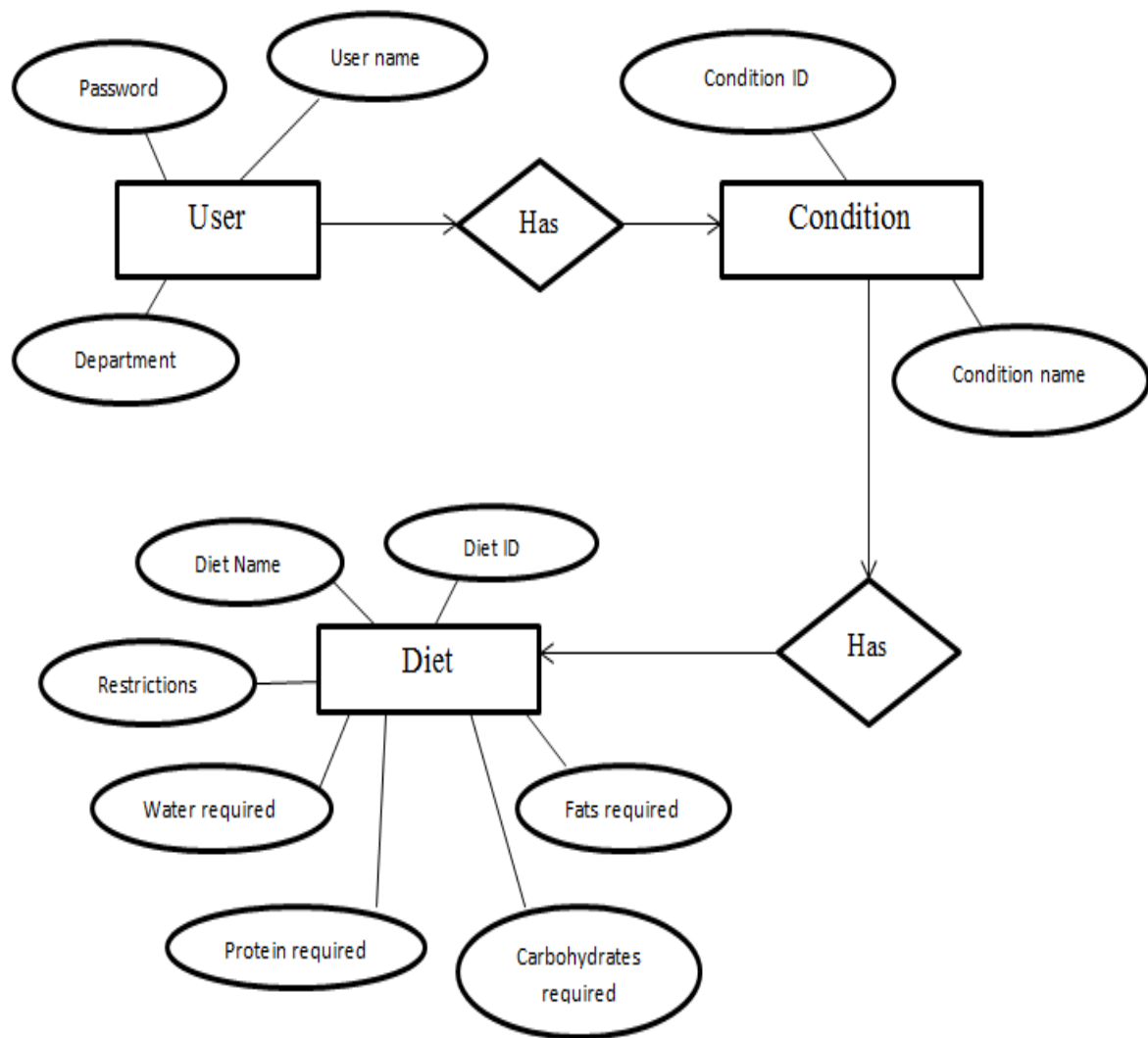


Fig 4.5 EER diagram

4.7.3 Program Design

The process which involves the design of modules and how these modules interact with each other using UML diagrams which include class diagrams is called program design. A correctly documented program is easy to maintain in future. Some of the characteristics that describe a good program include reusability, loose coupling, extensibility, ease of maintenance and minimum complexity. The program design also shows how the software data and functions interact with each other.

4.8 SYSTEM CLASS DIAGRAM

Whitten and Bentley (2001), argues that a system class diagram shows the attributes of the classes, class of the system, operations as well as interrelationships between the users and the systems. A class diagram can be referred to as an object oriented development and design

tool. A rectangle is an icon that is used to show a class. The class is divided into three sections that include class, attributes and the operations.

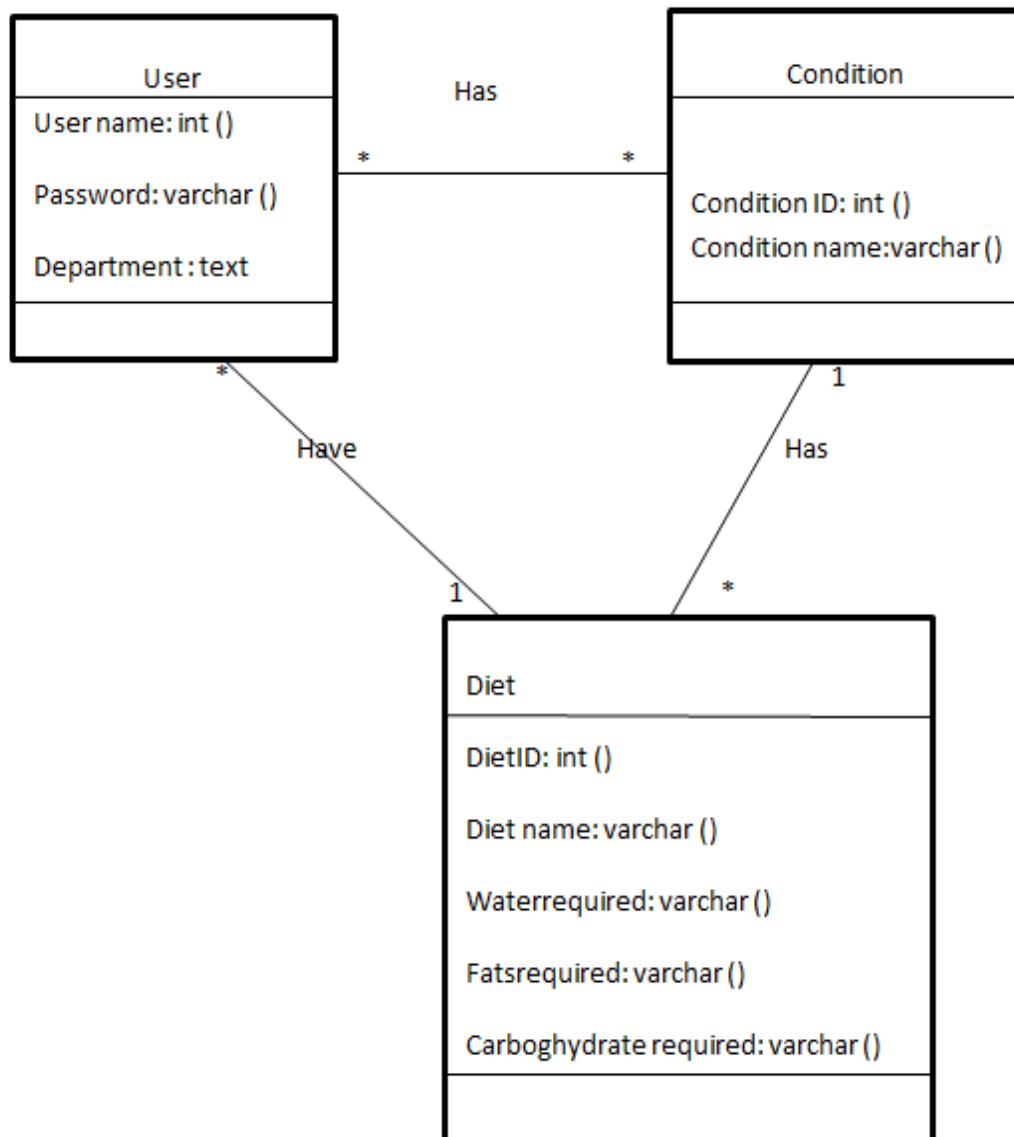
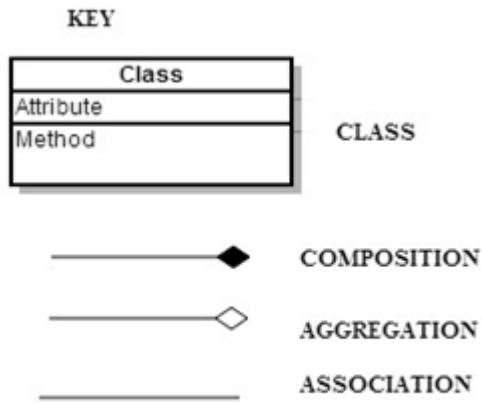


Fig 4.6 Class diagram of proposed system



4.8.1 Sequence diagram

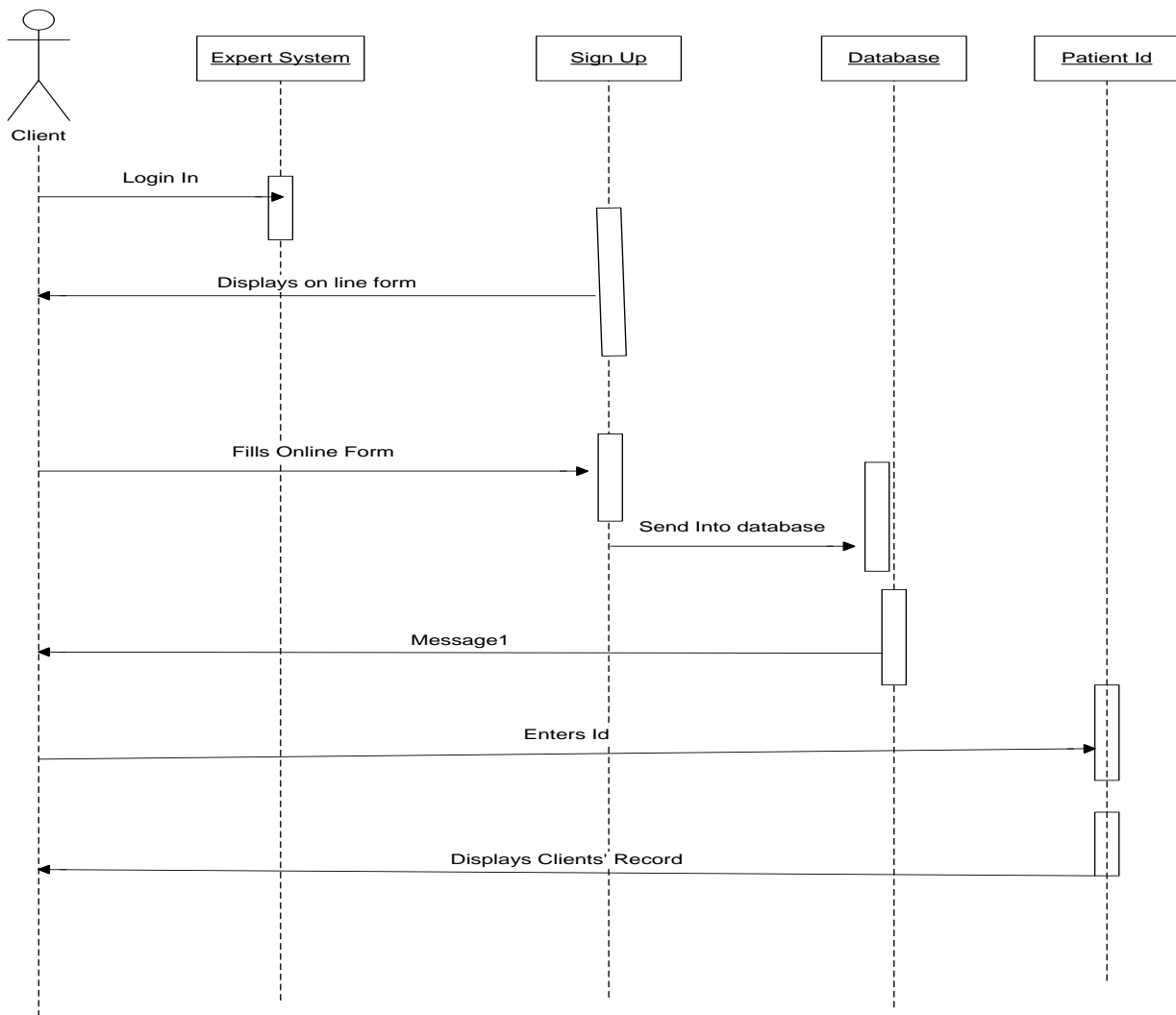


Fig4.7 (a) showing sequence diagram for registering with the Expert system.

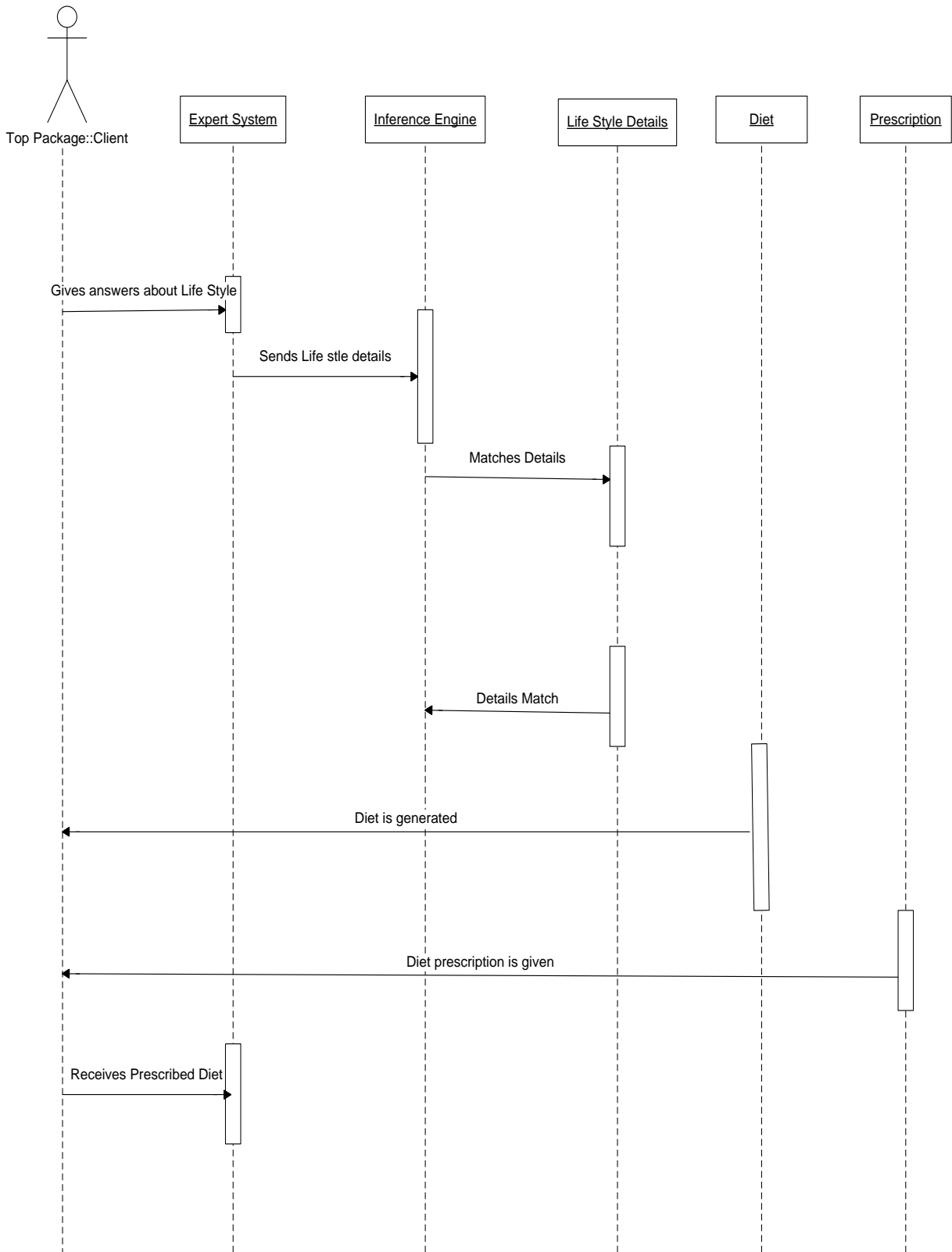


Fig 4.7 (b) Sequence diagram showing the continuation from registry.

4.8.2 Interface Design

James O, Brien (1996), suggest that in interface design that is where there is human interaction with the computer. There is communication between the two. The interface designs often take into consideration the capabilities, the needs as well as the experience of the system users. It goes through the application of design principles that include user guidance, user diversity, user familiarity, consistency and recoverability.

4.8.3 Input design

This stage, input forms would be designed. The forms are responsible for allowing users to input data into the system. Although these platforms do not show the complexity of the system, they however present a high level of abstraction of the new system. Its main objective is to present a user friendly environment for the proposed system. Input designing have certain procedures and these may include;

- Information or data that will be captured will be validated as it enter into the system before it reaches the database.
- A constant color format for all the control buttons should be observed.
- Titles are to be used for guidance throughout the system.

Login structure

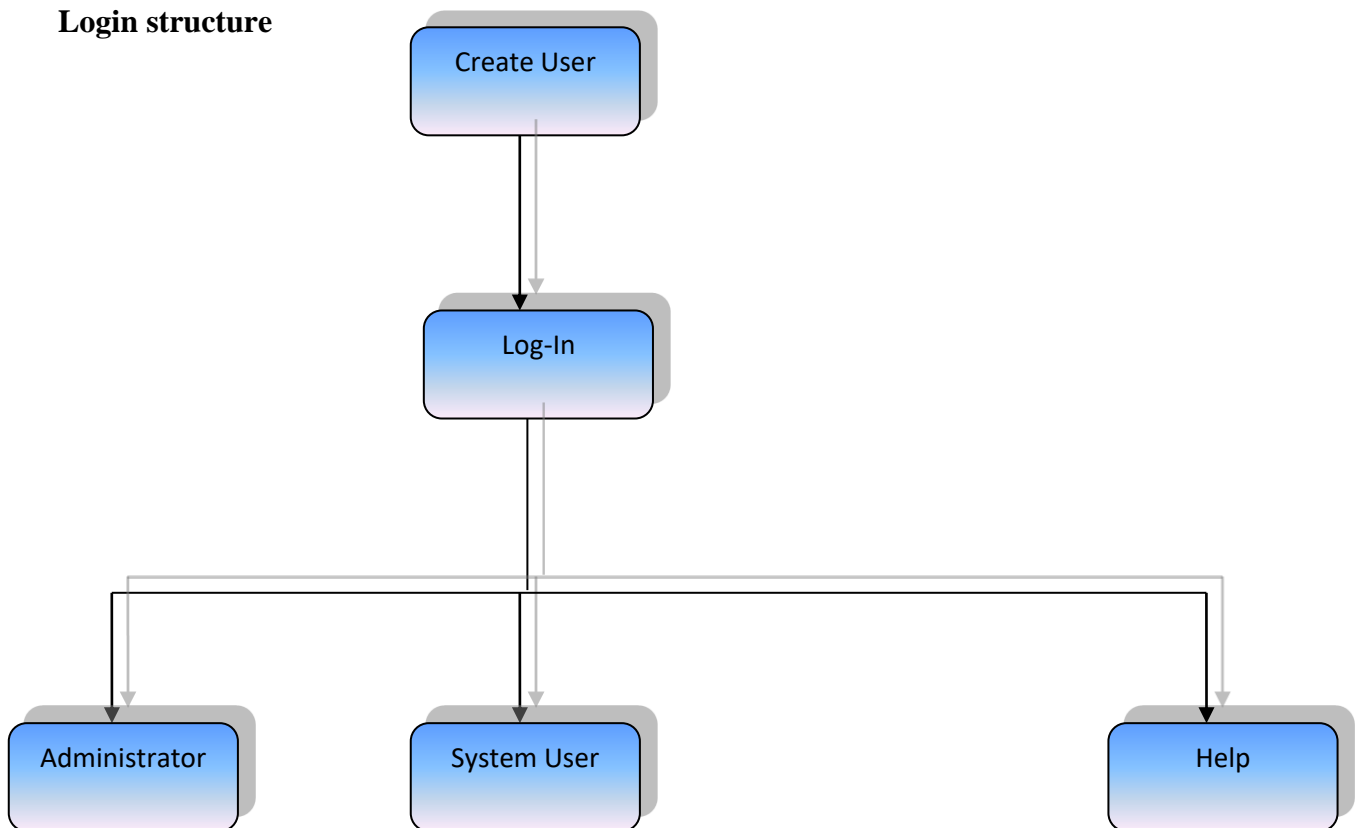
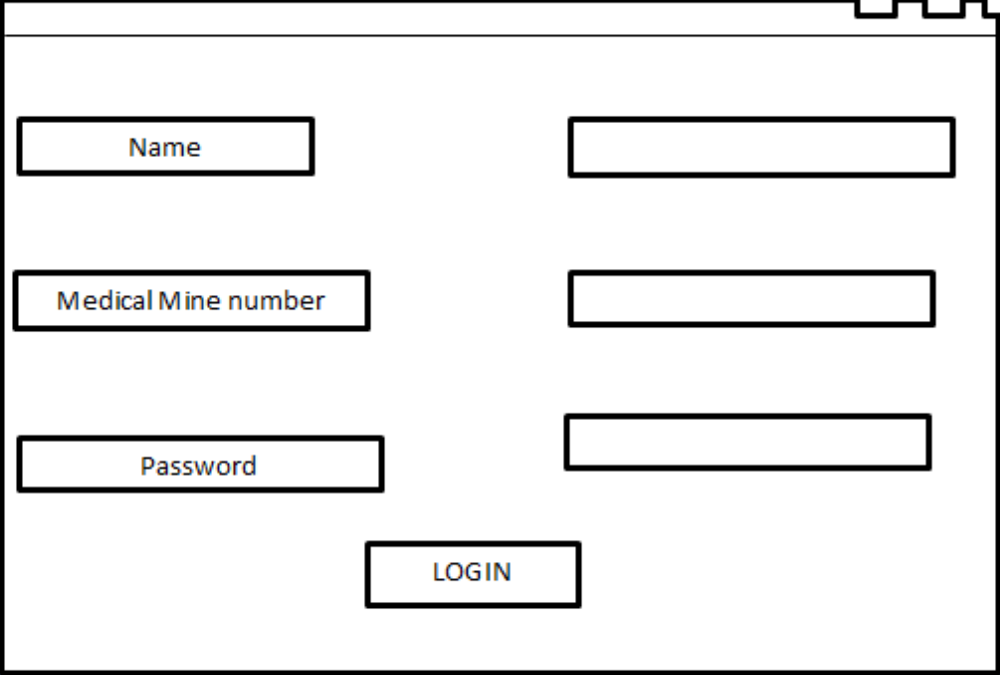


Fig 4.8 Login structure

Login form

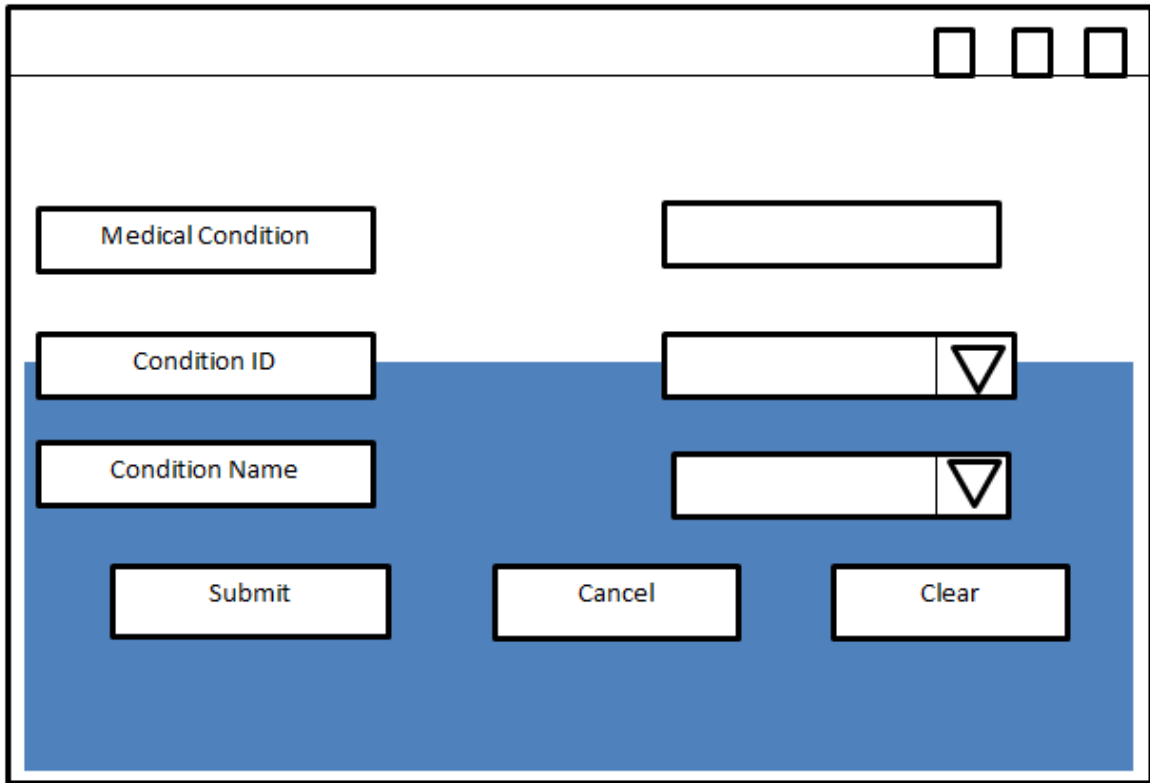


The image shows a login form interface within a window. The window has a blue border and three small white squares in the top right corner. The form itself is white and contains three input fields stacked vertically on the left side. The first field is labeled 'Name', the second 'Medical Mine number', and the third 'Password'. To the right of each label is an empty rectangular input box. Below these fields is a single button labeled 'LOGIN'.

Fig 4.9 Login interface

Condition explanation form

This is where the user explains the medical condition and specifies the name and ID of the problem.



The image shows a web form titled "Medical condition explanation form". It features a header bar with three window control icons (minimize, maximize, close) on the right. The form contains three input fields: "Medical Condition" (a text box), "Condition ID" (a text box with a dropdown arrow), and "Condition Name" (a text box with a dropdown arrow). Below these fields are three buttons: "Submit", "Cancel", and "Clear". The bottom portion of the form is highlighted with a blue background.

Fig 4.10 Medical condition explanation form

Diet prescriber form

On this form, the diet name and ID will be specified based on the condition explained before. The diet prescriber will then prescribe appropriate diet for the user and provide explanations.

Fig 4.11 Diet prescription form

4.8.4 Output design

Jeffrey A Hoffer Morden(2011) suggest that output design is the process of creating a platform that displays information to the users. The output is displayed or produced after the user has entered information into the system. A general form is created that will help display the inputted data. The output can be presented in different forms such as hard copy thus printed paper or soft copy thus message on the computer screen. To make sure there is an effective output design, the output must be delivered on time meeting the user request, it must be meaningful thus being understandable and should be of quality. The output presented by the proposed system is the prescribed diet information and instructions that come out after the user has stated the medical condition.

Prescribed diet produced by the expert system

This report provides the dietary information that the patient would have enquired through stating the medical conditions.

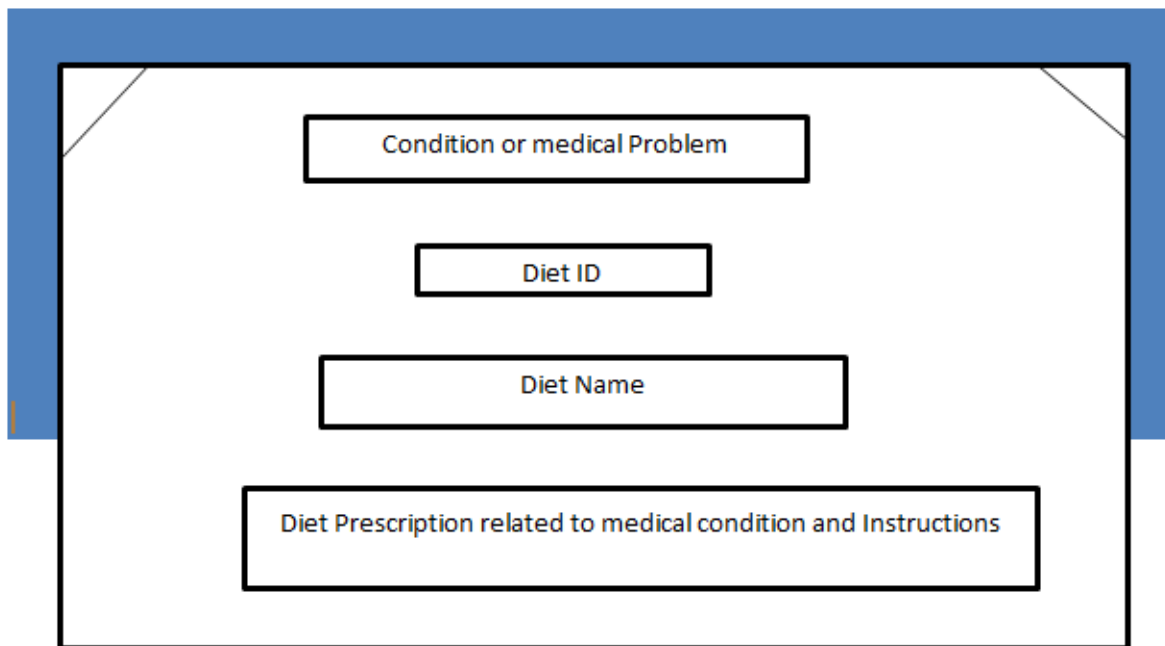


Fig4.12 Diet prescription output form

4.9 CONCLUSION

After all the analysis, plans and proposed designs for the system, there is a need now to convert the ideas into a working system that will address the problems sited by the current system. Under this phase, the environments and platforms that the proposed system will use were defined. The entire system has been designed in a way that administrators will be able to maintain as well as the users to be able to use the system with little to no consultation for help.

5.0 INTRODUCTION

Bentley and Whitten, (2007), suggest that the implementation process of a system include functions or activities such as testing, installing the software, maintaining it as well as training the users about the system. In this level, the system would be tried and tested to evaluate any hiccups on the new system so that recommendations of corrections will be done before the installation process. Various testing sessions are performed to ensure all errors are highlighted and corrected.

5.1 CODING

Kenneth E. Kendall and Julie E. Kendall (2011), argues that coding comprise of program instructions that will be converted into certain instructions that will then be executed by the computer system. First to be constructed was the system database and Xampp thus include MySQL and Apache were used. The programming languages used on the system is C#.

5.1.2 PSEUDO CODES

Login

Check whether there is connection to the database,

If connected

Enter username and password,

If incorrect then

Re-enter the username and password correctly,

If correct then

Go to main menu,

Else if

Go to Admin main menu,

Else if

Go to user main menu,

Else if

Exit the program,

End if

Adding a new user

Enter Admin main menu,

Input user details,

Validate the input (all fields filled),

If invalid information is entered

Display error message,

Else

Save user information,

Display approval,

End

Diet prescriber

Enter medical condition,

Enter body weight details,

Check input of details,

If correct

Else

Receive diet prescription details

5.2 TESTING

Concise oxford dictionary (2001) refers testing as a survey that is done to provide stakeholders with relevant information about the product quality as well as the service under test. To ensure that the system meets user requirements, a number of testing techniques should be done. To ensure that all components of the database meet their functionality, intensive testing was done. The general testing process is illustrated by the figure below.

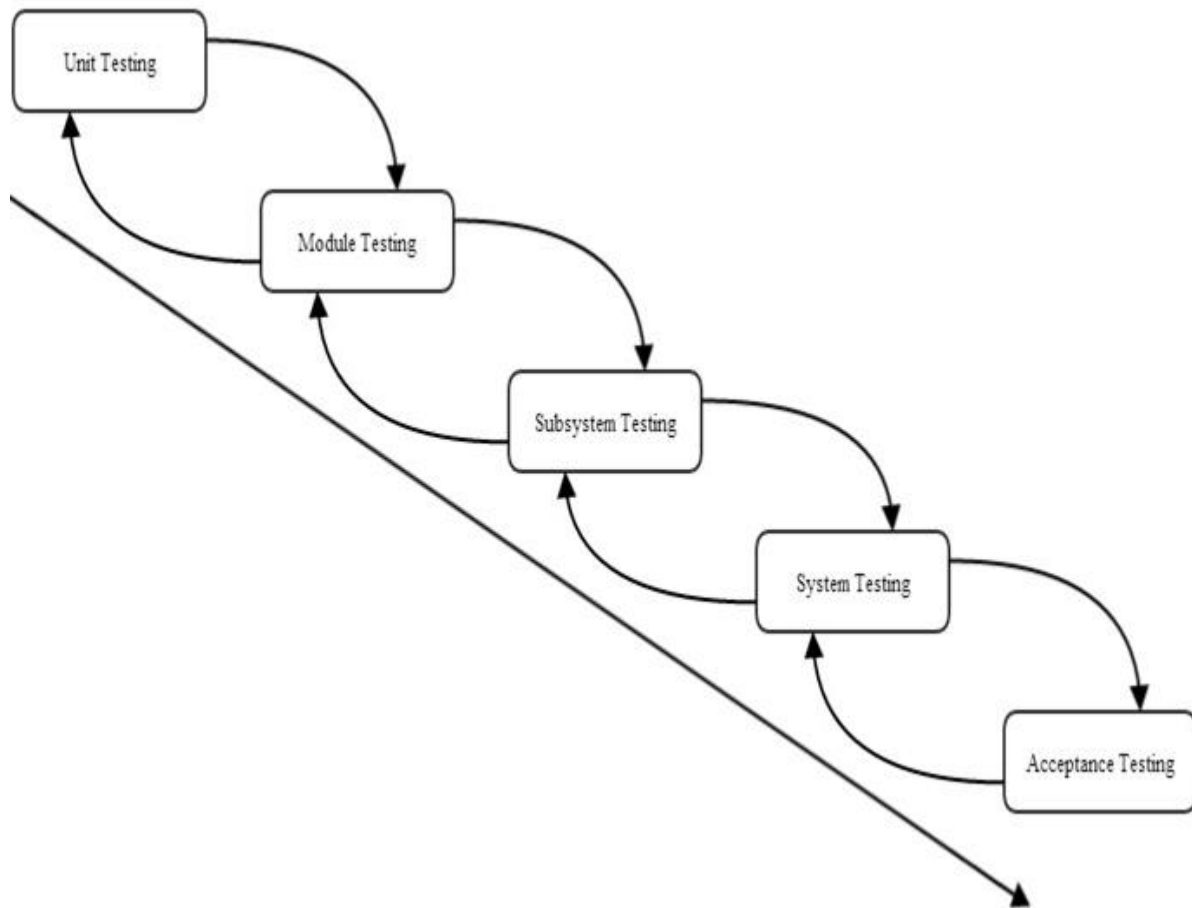


Fig 5.1 System Testing Procedure according to Bentley and Whitten (2007) System analysis and Design.

5.2.1 Unit Testing

Hammond et al (2005) argues that unit testing concentrates on a single unit for testing and these may include a module of a function. The main objective of testing these modules individually is to make sure each single unit performs according to its specified tasks. The unit test can be separated into two forms and these are explained below.

White box test

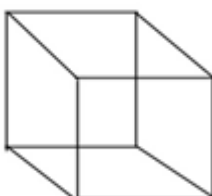


Fig 5.2 white box testing

It is a way of testing the code and design documents as well as analyzes and possibly manipulates the internal state of the entity put to test. It is known as glass box testing thus a security testing technique that can be used to validate whether code implementation follows planned design. It focuses at the internal structure of the system. In order to test the intended and as well as unintended software behaviour, white box testing include analyzing the dataflow, information flow, control flow and exception and error handling.

Black box test



Fig 5.3 Black box testing

Hammond et al (2005) on the other hand refers black box texts as the way of testing where by the tester uses the inputs and then observe the outcomes although would not be able to observe the inner core of the module being tested. To perform this type of testing, there is no need to possess the knowledge of the core design or the code. Its top objective is to test the software application or product for the requirements or its functionality. It is also known as the functional testing technique. The main concern is the functionality and not the implementation of the software.

5.2.2 Module testing

Bentley and Whitten (2007), defines module testing as the entire testing of code objects that are produced by the compiler in use when built from the source. Using this method, it combines dependent components and therefore testing them wholly. A pool of events or functions would be tested and unit testing will also be done using the stated objectives. The intended goal is to test the functionality of the modules as well as their reliability. Patient prescribed diet form was tested for their dependency with user condition specifier form module. The idea was to find out if the system was sending correct prescriptions per person intended to receive the diet instructions and information.

5.2.3 Subsystem

According to the *Concise oxford dictionary (2001)* subsystem is a smaller version of the big system. This is where by all the modules are gathered and tested. These modules would have been combined into subsystems. Subsystems ensure that the streams are correct. Interface mismatches are detected as well as the exercise of the interface between programs. The prescription form at the dispensary was tested in accordance to the doctors' prescription module. When the prescribed medication message is delivered to the dispensary, an acknowledgement that the message is delivered is sent back to the doctor for conformation. The system yield positive results by performing this function.

5.2.4 System testing

Jeffrey A. Hoffer: Morden (2011), refers system testing as the testing of the entire information system as well as the processing situations. To come up with a system, the sub modules above were tested. In order to avoid errors that would obstruct the functionality of the system, the system was tested. The system worked properly and data was used and verified.

5.2.5 Acceptance testing

The end users test the system so as to see whether it is acceptable or not. The end users are the ones who take up this testing. The testing is the one that will determine the success and the catastrophe of the proposed system. It comprise of beta and alpha testing.

➤ Beta testing

The proposed system is tested using the actual procedures and data by the doctors and the nurses who will be using the system. Like any other systems, errors and omissions on the new system will be seen or noticed. The process is done now and again until the organisation and the users accept that the proposed system is ready for usage.

➤ Alpha testing

The proposed system is then delivered and presented to the organisation as well as the project stakeholders. They will have a feel of the system testing for errors and any hiccups then report the errors that they would have discovered. The system development team will then correct the errors that would have been noticed.

5.3 TEST STRATEGIES

According to *James O'Brian (2006)*, System design, suggest that test strategies are there to make sure that there is the correct functionality of programs and systems. Some errors such as syntax and logic errors were discovered through using code reviews as well as structured walkthroughs.

5.3.1 Verification

According to *Hammond et al (2005)*, verification is checking if the system was established correctly. The system was tested and checked to see if it meets the user requirements and specifications. Verification can be assessed by acceptance testing.

5.3.2 Validation

According to *Jeffrey A. Hoffer: Modern (2011)*, System Analysis and Design, refers validation as checking and evaluating the system for the right and intended functionality. In this case if a field is not completed for example medical condition description; the system will refuse to give the patient a prescription because nothing will be specified. If a user does not enter a valid password, the field below highlights.

Login Error

Healthy Diet for a Healthier Lifestyle

Login

Here is The Online Diet Prescriber

- To assist the dietician.
- To calculating clients' nutritional requirements using standard equations based on assessments of the height, weight, stress, age and other relevant factors.
- To prescribe suitable diets to clients.
- Diagnose diet related medical conditions given symptoms.
- To remind the possible dietary therapy and nutritional calculations to the dietician on the basis of given factors (to overcome human errors like diligence, versatile, tiredness).
- To educate other healthcare professionals (doctors, nurses, biochemists, social workers, care workers and community workers) about food and nutrition issues.
- To provide researchers a huge and up-to-date repository of information regarding various diets.

To get registered click [here](#). If you are already registered log in below

Email

Password

Login [Forgot my password?](#)

ACCESS DENIED. User account not found.

Enter symptoms and click Search Search

Search Results

Copyright © M Chikoko 2014

Fig 5.4 Login error display

The user tried to enter into the system without using a password or username. However the system denied because all the fields are not filled in and the password is incorrect. Access denied user not found is the message that pops up.

Password creation error

Home Page Diet Prescription System Data System Administration MIS Reports Login > Patient > System Admin > Users > Renewal

Healthier Lifestyle

LOGIN CREDENTIALS RENEWAL

User Name M Chikoko

Login Email mchikoka@mimoso.co.zw

Old Password

Login Email mchikoka@mimoso.co.zw

New Password

Re-enter Password

Question What is your favourite color?

Answer red

Save

Simple passwords are dis-abled. Strong password must contain minimum of 4 characters and satisfy 3 of the rules. It must contain Special Characters or Upper case or Lower case or Numerics.

Logout << Back Logout

Copyright © Malizita Chikoko 2014

Fig 5.5 Password error display

The user entered a short password and did not meet the password policy. The password policy at Mimosa mine states that a password must contain a minimum of four characters and it should satisfy three of the rules that include containing special characters, upper case or lower case and numeric.

Field selection error

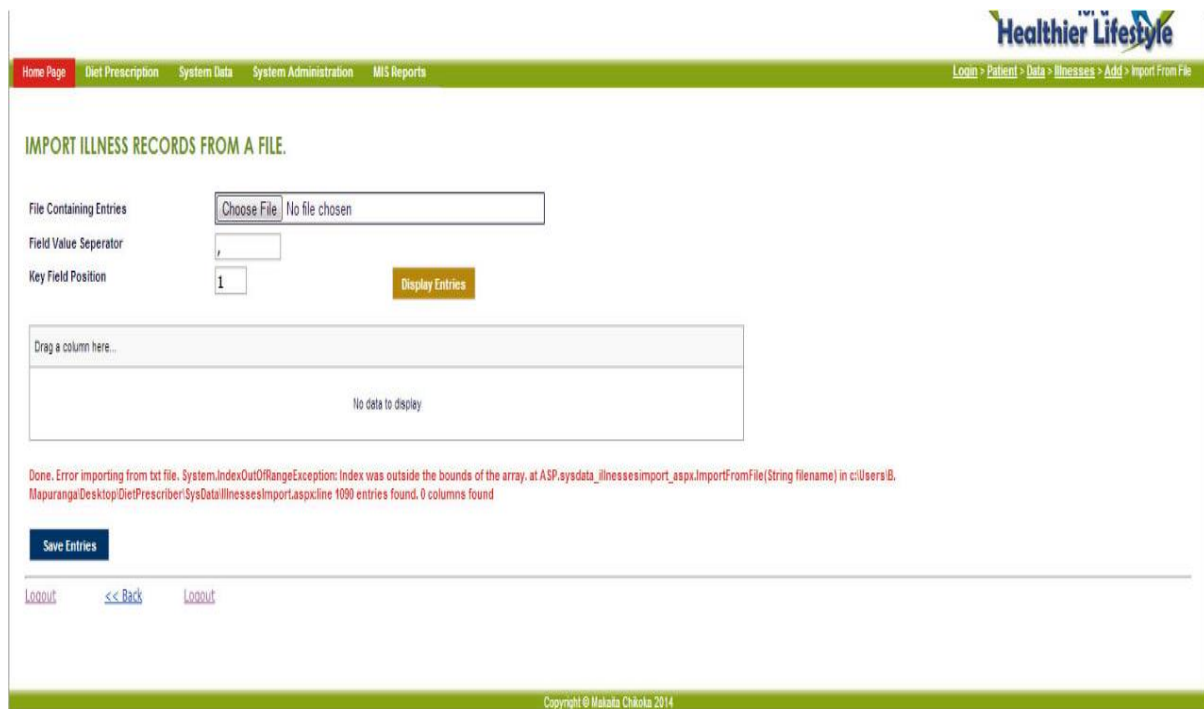


Fig 5.6 Field selection error display

The user did not select a record file of illness from the previous session and tried to continue. Error importing from text file is the message that pops up.

5.4 INSTALLATION

According to *Whitten and Bentley, (2007)*, System Analysis and Design, installation is defined as getting the proposed or developed system to use. The users will now be adopting the new system. The new software is installed on the hardware transferring from the old system to the new system. The main activities or functions at this stage are the conversion of files and the training of the users. System change procedures for example the pilot, parallel and direct operations are carried out during this stage.

5.4.1 User training

Training for users was planned and developed. It included the entire system entities. The purpose of the training was to conduct and familiarize the end users with the new proposed system. It is vital as well for the management to be familiar with the new system in order to be able to read and evaluate the generated reports produced by the system as well as help them for evaluation purposes. Training is carried out so that the people intended to use the system can operate and understand the functions.

5.4.2 File conversion

Gary B. Shelly and Harry J. Rosenblatt (2010) refer file conversion as a process were by the users verify the results of the testing process and then send the files to the new system. The conversion of the files will be done on weekends or after working hours as the users would be using the files during working hours.

5.4.3 Sub system changeover

Summerville. I, (2004), Software Testing, refers system changeover as technically replacing the old system with the new system. Many approaches can be used for system changeover and can include;

Direct changeover

Using this method, the old system stops to function and is replaced straightaway with the new system.

Merits

- Duplication of work is minimized thus it is efficient
- One system will be functioning at a time thus it becomes less expensive

De-merits

- The proposed system might not be completely correct.
- Operational details have to be carefully followed as well as carefully planning and testing.
- When errors are identified after the changeover process, it would be difficult to make the system operational.

Parallel running

Summerville. I, (2004), Software Testing defines parallel running as running both the old and new system together for a short while then later stops the old system and let the new system function fully.

Merits

- Using this method, a backup option exists therefore making it less risky as results can be verified.

De-merits

- It is costly as both systems will be functioning at the same time for some period of time.
- For systems that are not similar, this method cannot be adopted.

Pilot operation

Summerville. I, (2004), Software Testing, define pilot operation as running both the current as well as the proposed system at once but in a single unit location. The other section of the company will carry on working with the previous system while the other part would use the new system.

Merits

- Costs are moderate as selected part of the organisation will be operating on two systems at once.
- There is a lower risk of failure as the proposed system would be installed at a pilot site.

Phased changeover

Using this method, the system is executed in steps across the organisation. Part of the system is given to the organisation. It is however less costly as the system will be introduced and implemented in stages rather than all at once. Risk is moderate as it is only limited to the implemented module.

5.5 RECOMMENDATIONS

The system produced a high level of user satisfaction. The new proposed system proved to be highly maintainable as well as possessing better security and control measures. The users recommended the system as being user friendly as well as efficient. However the users responded very well to the training. The parallel running installation method was suggested and chosen because of its identified benefits although it may require a lot of work.

5.6 MAINTENANCE

According to *James O'Brian (2006)*, maintenance is a continuous way of checking, assessing as well as changing the functioning information systems to make necessary enhancements. A continuous maintenance is needed on the new system to make sure it continues to conform to the user and system specifications. Continuous reviews and upgrades are to be done on the new system after periodically checking for environmental changes as well as the specifications. System maintenance is periodically and an ongoing process. Below are maintenance types;

5.6.1 Corrective maintenance

Mary Sumner (1998) refers corrective maintenance as a method that is used to rectify a failure that happened or is in the process of happening. The method may involve repairing, restoring or replacing components. Regular inspection is to be carried out in order to identify errors in time as well as failures for corrective maintenance.

5.6.2 Preventive maintenance

This type of method according to *Mary Sumner (1998)* is carried out constantly or periodically according to system requirements so as to monitor some conditions on the system. Life prolonging tasks and condition monitoring are programmed at consistent intervals.

5.6.3 Adaptive maintenance

According to *Mary Sumner (1998)* argues that this type allow the software to adjust into the system environment. It has the capability to operate the software in a changed environment although it does not offer new skills.

5.7 RECOMMENDATIONS FOR MAINTENANCE

If the system is to last or live for long, continuous and effective maintenance is to be carried out. As the system is used, changes are seen and may be considered. Change requests due to some identified problems or users whom have noticed some hiccups in the system should be fixed.

5.8 USER MANUAL

This is a manual documentation that provides guidance or procedures of the system to the user and shows how to use the system. The interface and basic structure of the system is shown on this user manual.

See appendix for user manual

5.9 RECOMMENDATIONS

Before using the system, users should familiarize and understand the user manual. Users should however maintain system security and follow all protocols required by the system. Passwords should be confidential and not to be shared to any unauthorized personnel. Maintenance levels should be kept and done constantly so as to keep the standards for the system as well as maintain the performance.

5.10 LIMITATIONS

On developing the system, some hiccups were uncouncted as well as in the implementation of the objectives. A lot of time was needed to develop the system therefore due to scarcity of it, the system was developed hurriedly to meet the company requirements at the expense of what the market really wants.

5.11 CONCLUSION

A functional manual is needed for the users so as to have a continuous operation on the system. After passing through all these stages, the project will now be functional.

Bibliography

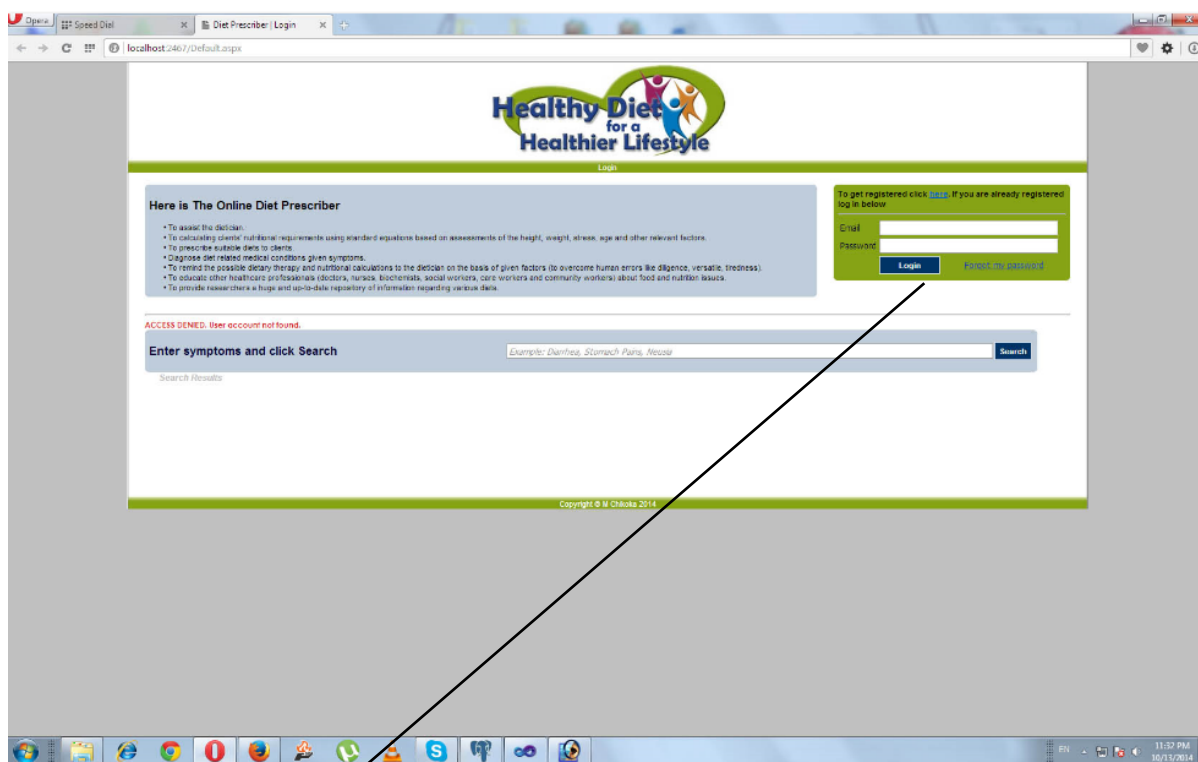
1. Alan Dennis, Barbara Haley Wixom, Toberta M. Roth, System Analysis and Design, 5th Edition John Wiley & Sons, Inc.,
2. Concise Oxford Dictionary (2001)
3. Dennis, L., (2010) Systems analysis and design 5th Edition, McGraw-Hill
4. Frey J.H & Oishi S.M (1995), How to conduct interviews by telephone and in person
5. Gary B. Shelly and Harry J. Rosenblatt (2010), Systems Analysis and Design, 8th Edition
6. Hammond et al (2005), Management of Information Systems, Imperial college London.
7. James O' Brien, (1996), Management Information Systems, 4th Edition, Prentice Hall, London.
8. Jeffrey A. Hoffer: Morden (2011), System Analysis and Design, Prentice Hall
9. Kenneth E. Kendall and Julie E. Kendall (2011), Systems Analysis and Design, 8th Edition, Prentice Hall
10. Kevin grant, David Edgar (1998), Management Information Systems, 4th edition
11. Lewis, J., Fundamentals of Project Management 3rd Edition (2006), Worksmart Publishing Company.
12. Lucey T (1996), Costing, 5th Edition, Aldine Place London
13. Mary Sumner (1998), Management Information Systems 4th Edition, London
14. Ray Hackney(2010), Strategic Information Systems Management
15. Schwalbe, K, (2006), Information Technology Project Management, 6th Edition, Cengage learning
16. Sommerville. I, (2004), Architectural Design, retrieved from <http://www.comp.lancs.ac.uk/computing/resources/IanS/SE7/Presentations/PDF/ch11.pdf>
17. Sommerville. I, (2004), Software Testing, retrieved from <http://www.comp.lancs.ac.uk/computing/resources/IanS/SE7/Presentations/PDF/ch23.pdf>.
18. Sommerville. I, (2004), Verification and validation, retrieved from <http://www.comp.lancs.ac.uk/computing/resources/IanS/SE7/Presentations/PDF/ch22.pdf>
19. Whitten and Bentley, (2007), System Analysis and Design, McGraw-Hill Company.

Appendix A User Manual

Installation and User Guide Mimosa Mine Diet Prescription Expert System

Login form

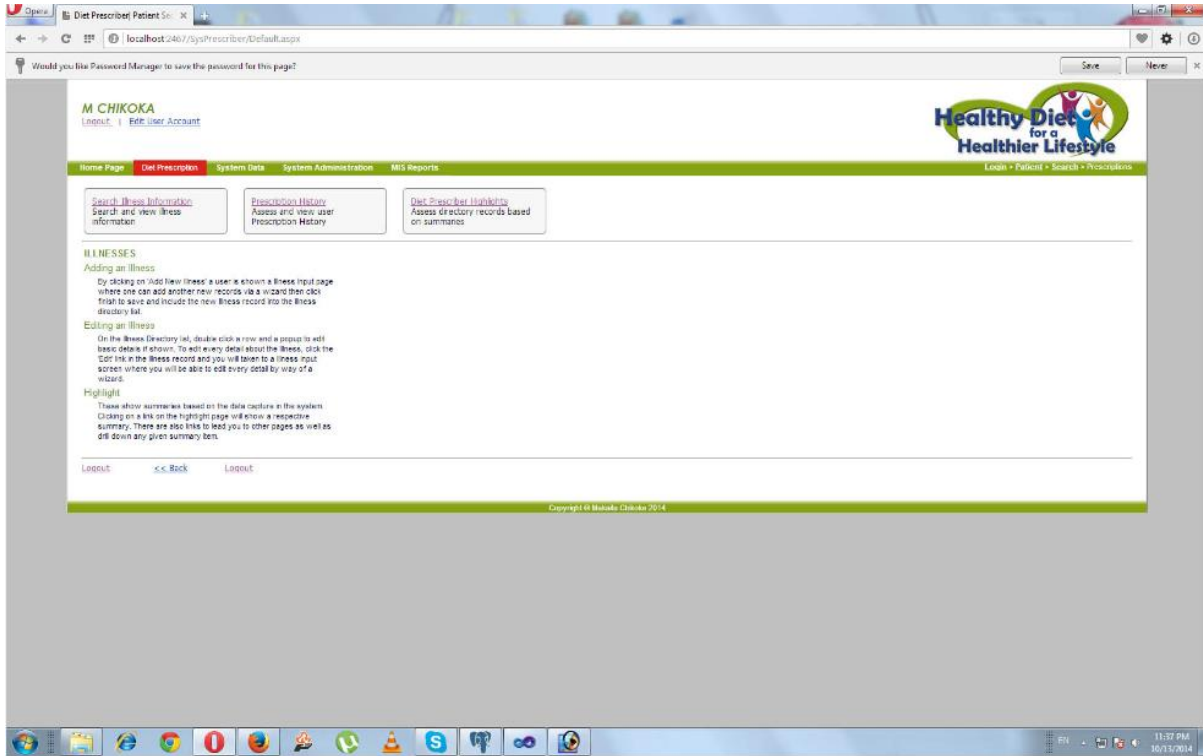
The login page is used by each user using the system to login into the system. In this case the users include the administrator and the users who are the mine employees. Each user has his/her own password and username they use to login. Passwords and username allocated to each user are different from each other because of their different assess levels. After each user login, this will take us to the home page of each user.



Submit username
and password and
click login

Main menu

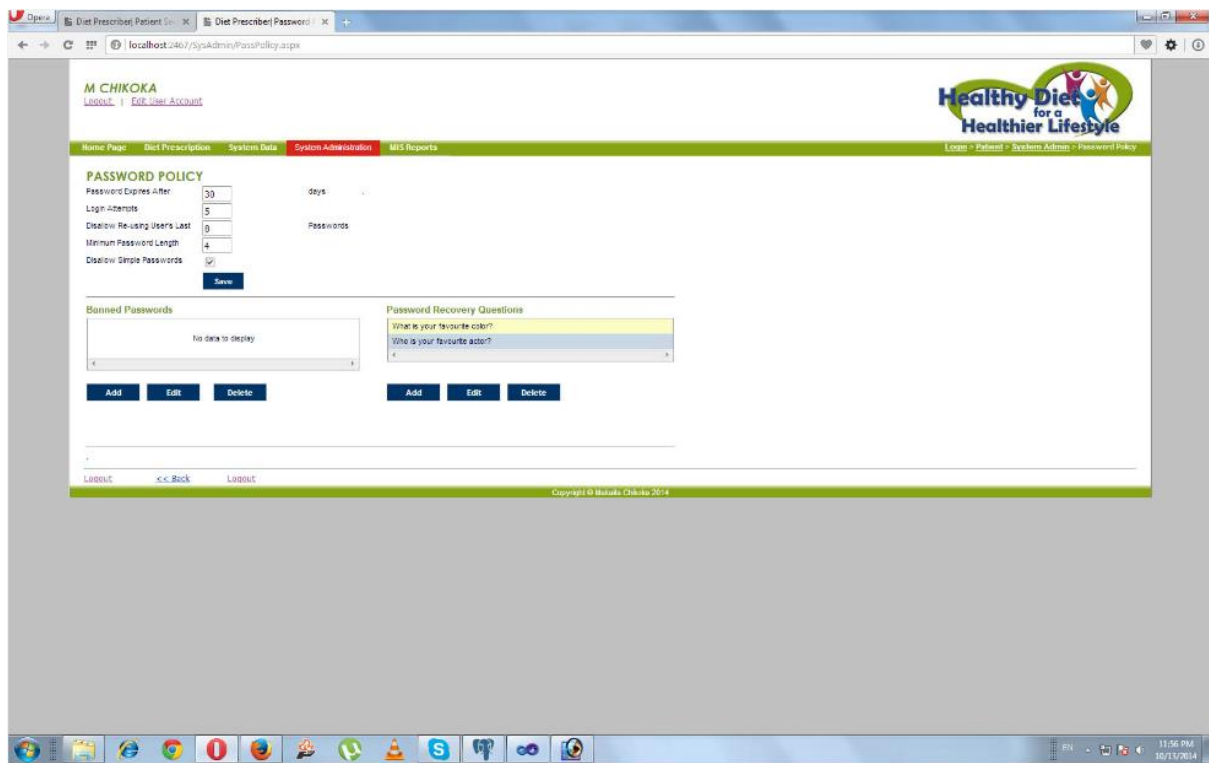
This is the page shown after the administrator or user login successfully into the system.



Password Policy

The system password policy state that

1. Passwords expires after 30days
2. Login attempts are limited to 5
3. The system disallow re-using users last 8 passwords
4. The minimum password length is 4 characters



Login credential renewal

Passwords expire after 30days therefore they should be renewed. It should meet the password policy as shown below.

M CHIKOKA
Logout | Edit User Account

Home Page | Diet Prescription | System Data | System Administration | MIS Reports

Healthy Diet for a Healthier Lifestyle
Logout | Patient | System Admin | Users | Renewal

LOGIN CREDENTIALS RENEWAL

User Name: M Chikoka
Login Email: mchikoka@mimos.co.zw
Old Password:

Login Email: mchikoka@mimos.co.zw
New Password:
Re-enter Password:

Question: What is your favourite color?
Answer: red

Simple passwords are dis-abled. Strong password must contain minimum of 4 characters and satisfy 3 of the rules i.e must contain Special Characters or Upper case or Lower case or Numeric.

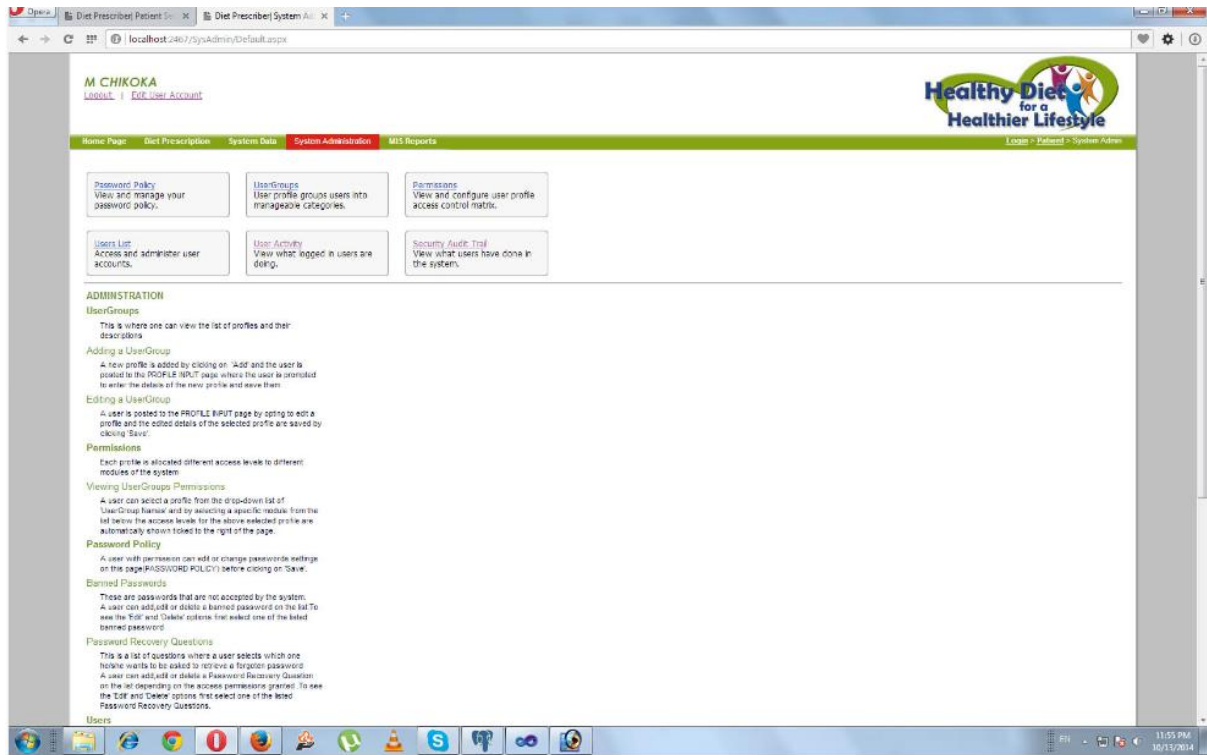
Logout | < Back | Logout

Copyright © Mimos MCHIKOKA 2014

Simple passwords are dis-abled.
Strong passwords must contain
minimum of 4 characters & satisfy 3
of the rules i.e contain special
characters, upper or lower case and
numeric.

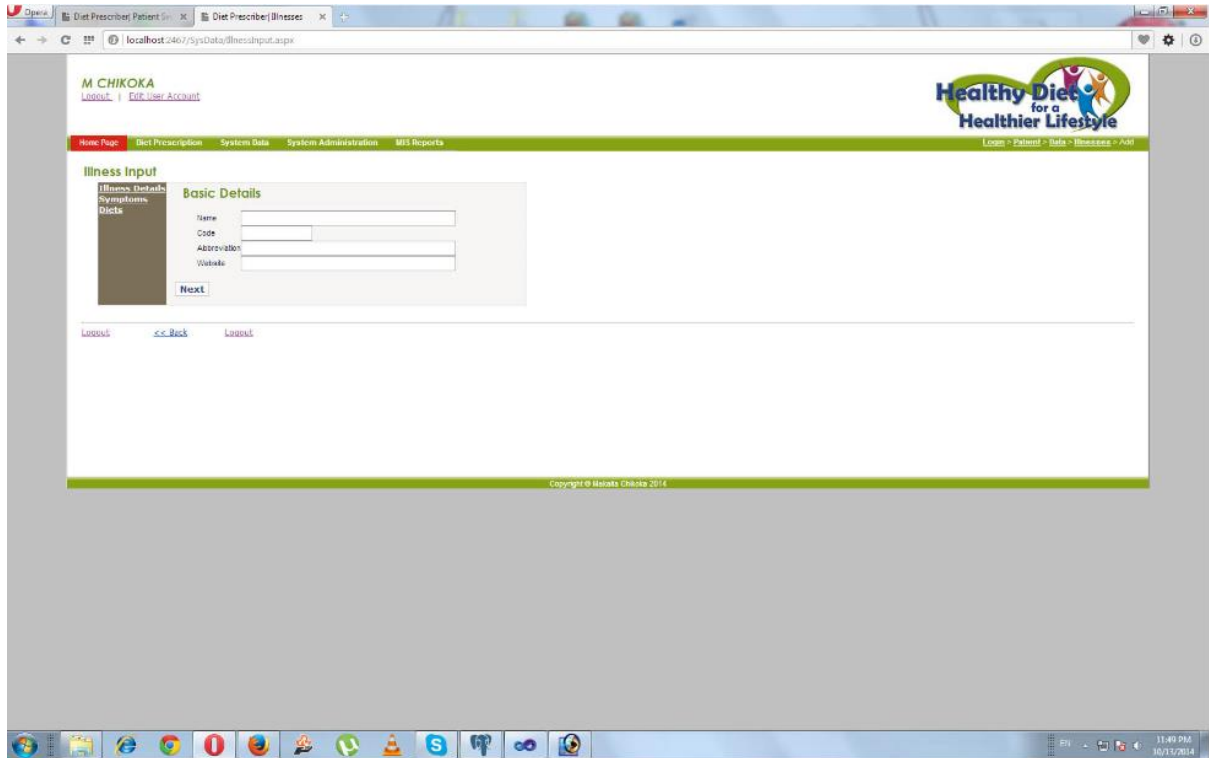
Administrator

This is the page shown after the administrator login successfully into the system. The administrator is able to add a new user as well as give permissions for each user. The administrator can also view all the users entered into the system and is able to edit if there are any changes.



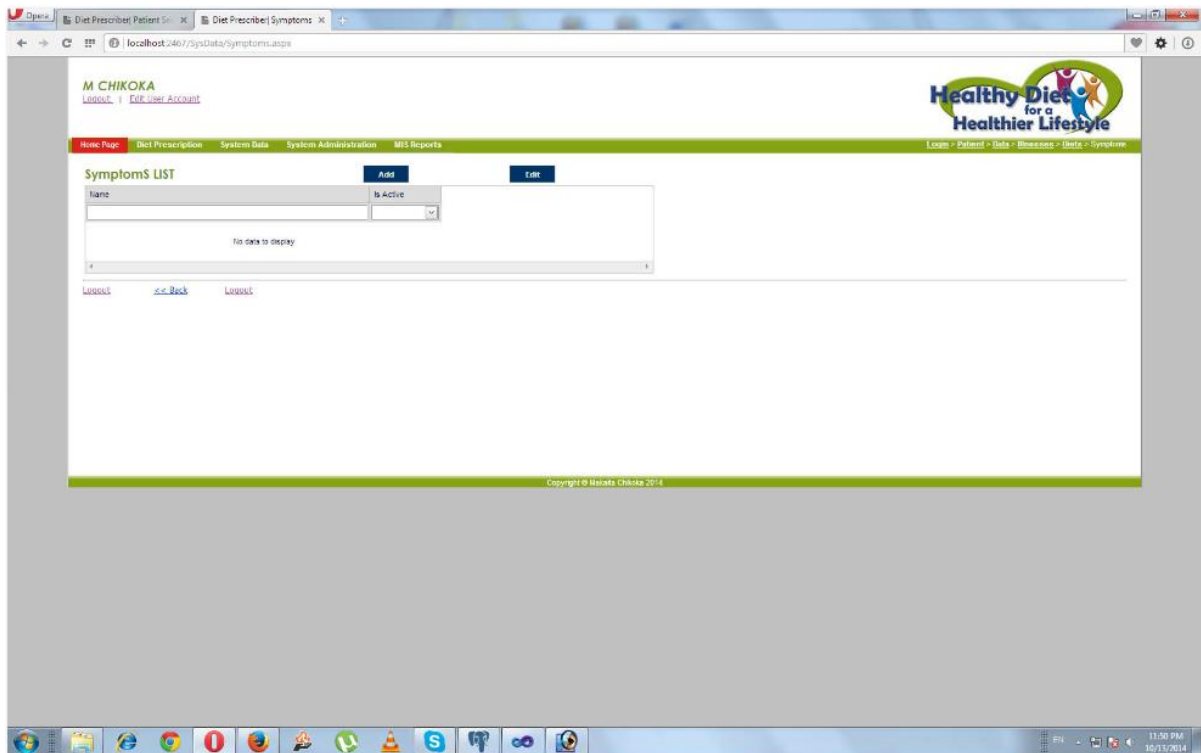
Condition description page

The users enter their illness and descriptions of the symptoms and press enter.

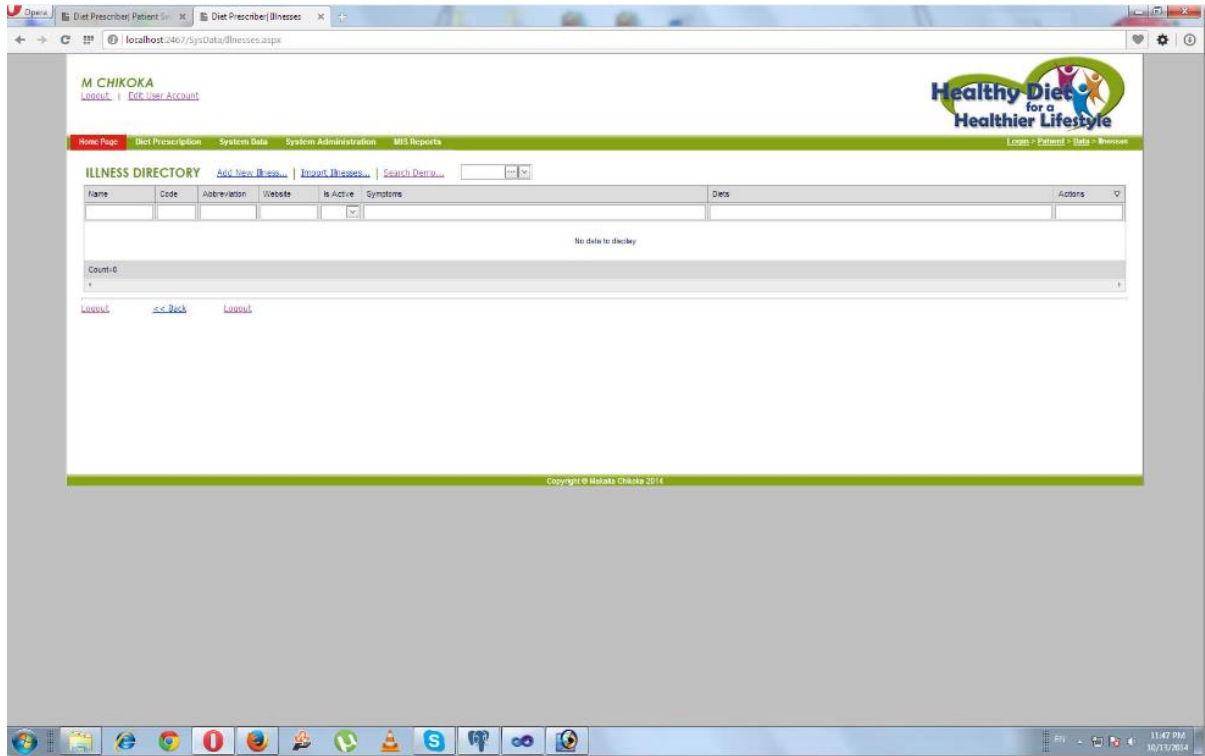


Symptoms list page

Here the various symptoms are displayed for the user to choose. But however if there is no symptom related, the user would have to type in the description of the symptoms on the space provided.

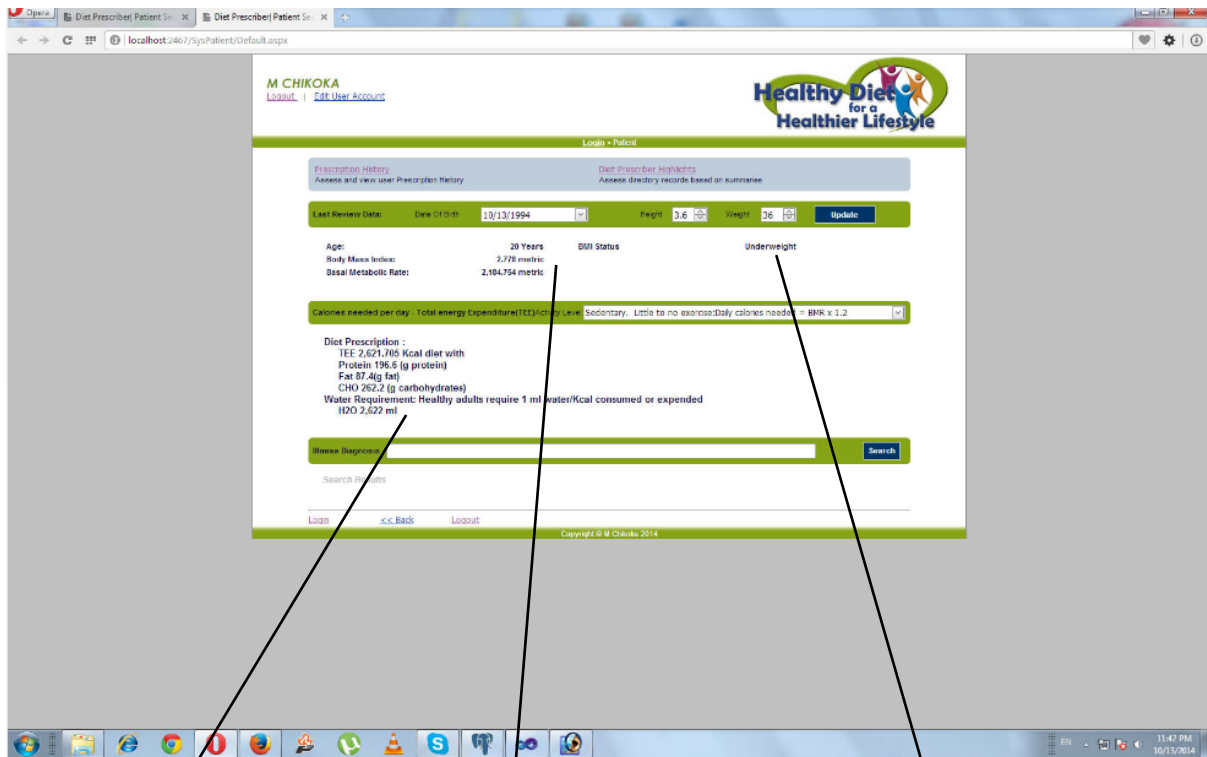


When the user press enter, the form bellow pops out prompting the user to view if there were similar symptoms specified before.



Diet prescription highlight

Here the user views the prescribed diet according to the entered symptoms or illness.



Prescribed diet details showing mineral contents and amounts

Showing age the calculated BMI

Detecting whether the user is underweight or overweight

Activity Reports

The administrator has the privilege to view the activity reports of the system. He/she is able to view the date and time the user login and even the PC that was used.

USER NAME	EMAIL	LOGINAT	CURRENT USER ACTION	DATE AND TIME	PCNAME
M Chikola	mchikola@mimosa.co.zw	13/10/2014 22:36:44	Diet Prescriber User Activity	13/10/2014 23:07:09	127.0.0.1 127.0.0.1

Shows the time the user login to the system

Current date and time

Shows the PC number the user was using

User profile list

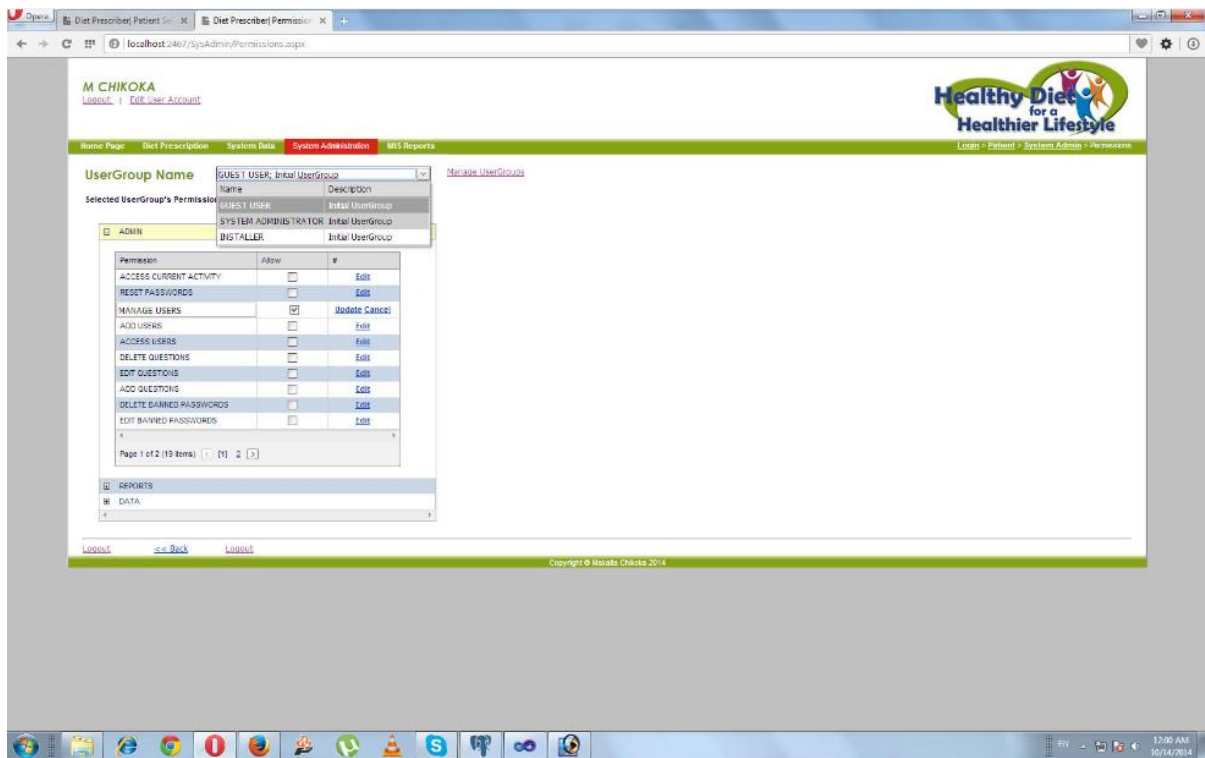
The administrator is able to view the user profile list thus whether it's a guest at the mine or a permanent employee. This form is editable only to the administrator.

The screenshot shows the M CHIKOKA web application interface. At the top left, the user is logged in as 'M CHIKOKA' with links for 'Logout' and 'Edit User Account'. The top right features the 'Healthy Diet for a Healthier Lifestyle' logo and a navigation path: 'Login > Patient > System Admin > UserGroups'. The main navigation bar includes 'Home Page', 'Diet Prescription', 'System Data', 'System Administration' (highlighted), and 'MIS Reports'. The 'USER-PROFILES LIST' section contains a table with columns 'NAME' and 'DESCRIPTION', and buttons for 'Add' and 'Edit'. The table lists three users: 'GUEST USER', 'SYSTEM ADMINISTRATOR', and 'INSTALLER', all associated with the 'Initial UserGroup'. Below the table are 'Logout', '<< Back', and 'Logout' links. The footer contains the copyright notice 'Copyright © Malala Chikoko 2014'.

NAME	DESCRIPTION
GUEST USER	Initial UserGroup
SYSTEM ADMINISTRATOR	Initial UserGroup
INSTALLER	Initial UserGroup

User group permission form

Here the administrator has the right to give permissions to a certain group of the users as well as resetting passwords for those who have attempted to login more than 5 times.



Audit trail form

The system allows an audit trail process were by all the modified data thus deleted or updated is displayd. It shows the modifier as well as the date and time. This form will help when preparing for the internal or external audits that take place now and again at the mine.

The screenshot displays the 'M CHIKOKA' audit trail interface. The page title is 'M CHIKOKA' with a logo for 'Healthy Diet for a Healthier Lifestyle'. The navigation menu includes 'Home Page', 'Diet Prescription', 'System Data', 'System Administration', and 'MIS Reports'. The 'System Administration' menu is active, showing 'Login', 'Patient', 'System Admin', 'Users', 'Current Activity', and 'Audit Reports'.

The audit trail form shows a table of system changes with the following data:

Audit ID	Date	Username	Work Station	Action	Description	XB	Plaf	Rev Data
1	2014/05/13 22:38:44	mchikoka@rencos.co.zw	127.0.0.1	Modified	Users	1	DCName=	DCName=127.0.0.1
2	2014/05/13 22:40:22	mchikoka@rencos.co.zw	127.0.0.1	Modified	Users	1	Height=1.6	Height=2.0
3	2014/05/13 22:40:31	mchikoka@rencos.co.zw	127.0.0.1	Modified	Users	1	Weight=60	Weight=41
4	2014/05/13 22:40:38	mchikoka@rencos.co.zw	127.0.0.1	Modified	Users	1	Weight=41	Weight=38

The interface also includes a 'Logout' button, a '< Back' button, and a 'Logout' button. The footer of the page reads 'Copyright © Makoto Chikoko 2014'.

Appendix B Interview Questions

Interviewer

Interviewee

Date

Time

Questions to Employee:

Question 1:

What do you think about introducing a diet prescriber expert system for the mine?

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

Question2:

What changes do you expect to be integrated in the proposed system?

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

Question 3:

What kind of problems do you encounter with the current system?

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

Question 4:

Will the new diet prescriber system benefit the organisation as a whole?

YES

NO

If yes explain:

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

Appendix C Interview Questions

Interviewer
Date

Interviewee
Time

Questionnaire to the User

Question1:

Are you satisfied with the current system?

YES **NO**

If NO state reasons and recommendations

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

Question 2 :

What problems are being faced by the use of the current system?

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

Question 3 :

Do you think introducing the new diet prescriber system will improve the standards of health delivery at the mine?

YES **NO**

If YES give reasons

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

Question 4 :

Is the Doctor always available on both shifts for the consultation?

YES

NO

If NO explain why

.....

.....

.....

.....

Appendix D Observation score sheet

Observation guide schedule.

DATE

OBSERVER

TIME

DEPARTMENT

OBSERVATION

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

CONCLUSION

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



STAMP

Appendix E Source Code

User Actions

```
using System;
using DevExpress.Xpo;
using DevExpress.Data.Filtering;
using DietPrescriber.AppClasses.SysPrescriber;
using System.Data;
using DevExpress.Xpo.DB;

namespace DietPrescriber.AppClasses.SysAdmin {

    public class Users : SysAuditObject {

        public Users(Session session)
            : base(session) {
            // This constructor is used when an object is loaded from a persistent
storage.
            // Do not place any code here.
        }

        public override void AfterConstruction() {
            base.AfterConstruction();
            // Place here your initialization code.
            IsLocked = false;

            Gender = "Female";
            DateOfBirth = DateTime.Now.Date.AddYears(-20);
            Height = 1.6; // metres
            Weight = 68; //kgs
            ReviewDate = DateTime.Now.Date;
        }

        [Delayed, Association("UserGroup-Users")]
        public UserGroups UserGroup {
            get { return (UserGroups)GetProperty("UserGroup"); }
            set { SetProperty("UserGroup", value); }
        }

        public string DistinctName {
            get { return Name + ": " + Email; }
        }

        public string Name {
            get { return GetProperty<string>("Name"); }
            set { SetProperty("Name", value == null ? "" : value.Replace(", ", "-
")); }
        }

        public string PhoneNumbers {
            get { return GetProperty<string>("PhoneNumbers"); }
            set { SetProperty("PhoneNumbers", value); }
        }

        // Diagnostic Input
        public string Gender {
            get { return GetProperty<string>("Gender"); }
            set { SetProperty("Gender", value == null ? "" : value); }
        }

        public DateTime DateOfBirth {
            get { return GetProperty<DateTime>("DateOfBirth"); }
            set { SetProperty("DateOfBirth", value); }
        }
    }
}
```

```

}
public double Height { //meters
    get { return GetPropertyValue<double>("Height"); }
    set { SetPropertyValue("Height", value); }
}
public double Weight { // Kgs
    get { return GetPropertyValue<double>("Weight"); }
    set { SetPropertyValue("Weight", value); }
}
public DateTime ReviewDate {
    get { return GetPropertyValue<DateTime>("ReviewDate"); }
    set { SetPropertyValue("ReviewDate", value); }
}

// Diagnostic Output
public int Age {
    get { return DateTime.Now.Date.Subtract(DateOfBirth).Days / 365; }
}
public double BodyMassIndex {
    get {
        // BMI = weight (kg)/ (height (m)) 2
        return Weight / (Height * Height);
    }
}
public string BodyMassIndexStatus {
    get {
        double ans = BodyMassIndex;
        if (ans < 18.5) return "Underweight";
        else
            if (ans < 24.9) return "Normal weight";
            else
                if (ans < 29.9) return "Overweight";
                else
                    return "Obese";
    }
}
public double BasalMetabolicRate {
    get {
        if (Gender.ToLower().Contains("f")) {
            // BMR = 447.593 + ( 9.247 x weight in kg ) + ( 3.098 x height in
            cm ) - (4.330 x age in years )
            return 447.593 + (9.247 * Weight) + (3.098 * Height * 100) -
            (4.330 * Age);
        } else {
            //BMR = 88.362 + ( 13.397 x weight in kg ) + ( 4.799 x height in
            cm ) - ( 5.677 x age in years )
            return 88.362 + (13.397 * Weight) + (4.799 * Height * 100) -
            (5.677 * Age);
        }
    }
}

public string Email { //Use this one for loginName
    get { return GetPropertyValue<string>("Email"); }
    set { SetPropertyValue("Email", value); }
}
public string Pass {
    get { return GetPropertyValue<string>("Pass"); }
    set { SetPropertyValue("Pass", value); }
}

```

```

    }
    DateTime fExpDate;
    public DateTime ExpDate {
        get { return fExpDate; }
        set { SetPropertyValue("ExpDate", ref fExpDate, value); }
    }
    public string Question {
        get { return GetPropertyValue<string>("Question"); }
        set { SetPropertyValue("Question", value); }
    }
    public string Answer {
        get { return GetPropertyValue<string>("Answer"); }
        set { SetPropertyValue("Answer", value); }
    }

    bool fIsLocked;
    public bool IsLocked {
        get { return fIsLocked; }
        set { SetPropertyValue("IsLocked", ref fIsLocked, value); }
    }
    public string PCName {
        get { return GetPropertyValue<string>("PCName"); }
        set { SetPropertyValue("PCName", value); }
    }

    public bool LoggedIn;
    public DateTime LogInAt;
    public string LastUserAction;
    public DateTime LastPing;
    public DateTime LastSync;
    public bool SysMaint;
    public string Username;

    protected override void OnSaving() {
        base.OnSaving();

        if (Session.FindObject<Users>(CriteriaOperator.Parse("[Oid]<>" +
            Oid.ToString() + " AND Upper([Email])='" + Email.ToUpper() + "'")) != null)
            throw new Exception("The specified email was already taken up. Please
try a different email.");
        if (UserGroup == null)
            throw new Exception("User profile can not be null.");

        try {
            if (SysModule.HttpSessionState != null)
                Username = SysModule.HttpSessionState["username"].ToString();
        } catch { }
    }
}
}
}

```

Diet Prescriber

```

using System;
using DevExpress.Xpo;
using DevExpress.Data.Filtering;
using DevExpress.Xpo.DB;
using DietPrescriber.AppClasses.SysData;
using DietPrescriber.AppClasses.SysAdmin;

```



```

namespace DietPrescriber.AppClasses.SysPrescriber {

    public class Searches : XPLiteObject {
        public Searches(Session session)
            : base(session) {
            // This constructor is used when an object is loaded from a persistent
storage.
            // Do not place any code here.
        }
        public override void AfterConstruction() {
            base.AfterConstruction();
            // Place here your initialization code.
            Tokens = "";
        }

        [Delayed, Association("Search-SearchItems")]
        public XPCollection<SearchItems> Items {
            get { return GetCollection<SearchItems>("Items"); }
        }

        [Key(true)]
        public int Oid;

        public string Reference;
        [Size(500)]
        public string Request;
        public int NextRow;
        public int TotalRows;

        public DateTime SearchDate;
        public decimal CutOff, AvgRelevance;
        [Size(1000)]
        public string Tokens;
        [Size(1000)]
        public string LastResponse = "";
        public bool IsRelated = false;

        // Methods
        protected override void OnSaving() {
            base.OnSaving();

            // Validate
            try { Tokens = Tokens.Trim(','); } catch { }

            // Initialisation
            if (Oid < 1) {
                SearchDate = DateTime.Now;
            }
        }

        protected override void OnDeleting() {
            base.OnDeleting();
            Session.Delete(Items);
        }

        public static void ClearCache(string pName) {
            try {
                Session xpoSession = SysModule.XpoNewSession("XpoLayer");
                XPCollection<Searches> xpcData = new
XPCollection<Searches>(xpoSession);
                xpcData.Criteria = CriteriaOperator.Parse("Items[Name='" + pName +
"'].Count>0");
            }
        }
    }
}

```

```

        if (xpcData.Count > 0) {
            xpoSession.Delete(xpcData);
        }
    } catch (Exception ex) {
        SysModule.LogInfo("Err ClearCache for " + pName + " " + ex.Message);
    }
}

public static Searches GetByReference(string pReference) {
    return
SysModule.XpoNewSession("XpoLayer").FindObject<Searches>(CriteriaOperator.Parse("[Reference]='" + pReference + "'"));
}

static string areaFilter = "";
static bool canFilterByArea = true;
public static XPCursor LoadData(Session pSession, string pCriteria) {
    // Filter Active Only
    pCriteria = (String.IsNullOrEmpty(pCriteria) ? "[IsActive]=True" :
"[IsActive]=True AND (" + pCriteria + ")");
    SortProperty[] sortProps = {
        new SortProperty("[Oid]", SortingDirection.Ascending)
    };
    return
        new XPCursor(
            pSession,
            typeof(Illnesses),
            CriteriaOperator.Parse((String.IsNullOrEmpty(areaFilter) ||
canFilterByArea == false ? pCriteria : "(" + areaFilter + ") AND " + pCriteria)),
            sortProps
        );
}

public static Searches LoadResults(string pReference, string pRequest, bool
pUseFullPhrase) {
    // Clean up
    pRequest = Soundex.CleanupPhrase(pRequest, ' ').ToUpper().Trim();
    // Load up
    Searches recSrch = GetByReference(pReference);
    if (recSrch != null) {
        if (pRequest == recSrch.Request && !recSrch.IsRelated) {
            // Index is out of bounds OR less than Hour ago =>> reset index +
date
            if ((pRequest == recSrch.Request) || (recSrch.NextRow >=
recSrch.TotalRows) || (recSrch.SearchDate < DateTime.Now.AddMinutes(-10)))
                recSrch.NextRow = 0;
            // Prepare to a refresher
            if (recSrch.SearchDate > DateTime.Now.AddMinutes(-2)) // newer
that 2 mins ago
                if (recSrch.Items.Count > 0) {
                    recSrch.SearchDate = DateTime.Now;
                    return recSrch;
                }
        }
        recSrch.Request = pRequest;
        recSrch.Session.Delete(recSrch.Items);
        recSrch.SearchDate = DateTime.Now;
        recSrch.CutOff = 0;
        recSrch.AvgRelevance = 0;
        recSrch.NextRow = 0;
        recSrch.IsRelated = false;
    } else {

```

```

        recSrch = new Searches(SysModule.XpoNewSession("XpoLayer")) {
Reference = pReference, Request = pRequest, CutOff = 0 };
    }

    // Format Search Message
    string prevSearch = "", search = "", pluralSearch = "";
    search = (recSrch.Request.Replace(" IN ", " ").Replace(" IN,", "
").Replace(",IN,", " ").Replace("\t", " ").Trim().Replace(";", ","));
    string[] words = search.Split(',');
    int cntFound = 0; // words.Length - 1

    prevSearch = search = "";
    recSrch.Tokens = ",";
    recSrch.Save();
    decimal sumRelevance = 0, ptsActual = (2 + words.Length -
recSrch.Request.Split(',').Length), ptsPhrase = 3, sumItems = 0;
    //SysModule.LogInfo("Initialising Search for: " + search);
    SearchItems recItem = null;
    foreach (string w in words) {
        if (w.Trim() == "") continue;
        search = w.Trim();
    redo:
        // Prepare to Search Check Aliases
        search = Aliases.GetSubstitute(recSrch.Session, search);
        if (!Soundex.CanEncode(search))
            if (search.Length < 4 || (!decimal.TryParse(search, out tempNum)))
        {
            prevSearch = search;
            continue;
        }
        if (("," + recSrch.Tokens).Contains(", " + search + ",")) continue;
        XPCursor xpcRecs = null;
        ptsPhrase = search.Contains(" ") ? 5 : 0;
        recSrch.Tokens += search + ",";

        #region Illnesses
        // Do Illnesses
        xpcRecs = LoadData(recSrch.Session,
String.Format("Upper([Name])Like'{0}' OR Upper([Code])Like'{0}' OR
Upper([Abbreviation])Like'{0}'", search));
        recSrch.TotalRows += xpcRecs.Count;
        int prevIllOid = 0;
        foreach (Illnesses b in xpcRecs) {
            if (prevIllOid == b.Oid) continue; // Just get the First as the
Default Contacts unless a single full word is supplies
            prevIllOid = b.Oid;
            recItem = new SearchItems(recSrch.Session) { Search = recSrch,
Name = b.Name, Abbreviation = b.Abbreviation, Code = b.Code, Symptoms = b.Symptoms,
Diets = b.Diets, Relevance = DoPoints(b.Code, ptsActual + ptsPhrase), Phrase = search,
IllOid = b.Oid, Field = "Name" };
            recItem.Save();
            cntFound++;
            sumRelevance += recItem.Relevance;
            // if (topScore < recItem.Relevance) topScore = recItem.Relevance;
        }
        if (xpcRecs.Count <= 0) {
            xpcRecs = LoadData(recSrch.Session,
String.Format("Upper([DistinctName])Like'{0} %'", search));
            recSrch.TotalRows += xpcRecs.Count;
            prevIllOid = 0;
            foreach (Illnesses b in xpcRecs) {

```

```

        if (prevIllOid == b.Oid) continue; // Just get the First as
the Default Contacts unless a single full word is supplies
        prevIllOid = b.Oid;
        recItem = new SearchItems(recSrch.Session) { Search = recSrch,
Name = b.Name, Abbreviation = b.Abbreviation, Code = b.Code, Symptoms = b.Symptoms,
Diets = b.Diets, Relevance = DoPoints(b.Code, ptsActual + ptsPhrase), Phrase = search,
IllOid = b.Oid, Field = "Name" };
        recItem.Save();
        cntFound++;
        sumRelevance += recItem.Relevance;
        // if (topScore < recItem.Relevance) topScore =
recItem.Relevance;
    }
    if (search.Length >= 3)
        if (xpcRecs.Count <= 0) {
            xpcRecs = LoadData(recSrch.Session,
String.Format("Upper([DistinctName])Like'{0}%' ", search));
            recSrch.TotalRows += xpcRecs.Count;
            prevIllOid = 0;
            foreach (Illnesses b in xpcRecs) {
                if (prevIllOid == b.Oid) continue; // Just get the
First as the Default Contacts unless a single full word is supplies
                prevIllOid = b.Oid;
                recItem = new SearchItems(recSrch.Session) { Search =
recSrch, Name = b.Name, Abbreviation = b.Abbreviation, Code = b.Code, Symptoms =
b.Symptoms, Diets = b.Diets, Relevance = DoPoints(b.Code, ptsActual + ptsPhrase),
Phrase = search, IllOid = b.Oid, Field = "Name" };
                recItem.Save();
                cntFound++;
                sumRelevance += recItem.Relevance;
                // if (topScore < recItem.Relevance) topScore =
recItem.Relevance;
            }
        }
    }
    if (search.Length > 3) {
        xpcRecs = LoadData(recSrch.Session,
String.Format("Upper([DistinctName])Like'%{0}%' ", search));
        recSrch.TotalRows += xpcRecs.Count;
        prevIllOid = 0;
        foreach (Illnesses b in xpcRecs) {
            if (prevIllOid == b.Oid) continue; // Just get the First as
the Default Contacts unless a single full word is supplies
            prevIllOid = b.Oid;
            recItem = new SearchItems(recSrch.Session) { Search = recSrch,
Name = b.Name, Abbreviation = b.Abbreviation, Code = b.Code, Symptoms = b.Symptoms,
Diets = b.Diets, Relevance = DoPoints(b.Code, ptsActual + ptsPhrase), Phrase = search,
IllOid = b.Oid, Field = "Name" };
            recItem.Save();
            cntFound++;
            sumRelevance += recItem.Relevance;
            // if (topScore < recItem.Relevance) topScore =
recItem.Relevance;
        }
    }
}
#endregion

#region Symptoms
xpcRecs = LoadData(recSrch.Session,
String.Format("Upper([Symptoms])Like'%,{0},%' ", search));
recSrch.TotalRows += xpcRecs.Count;
foreach (Illnesses b in xpcRecs) {

```

```

        recItem = new SearchItems(recSrch.Session) { Search = recSrch,
Name = b.Name, Abbreviation = b.Abbreviation, Code = b.Code, Symptoms = b.Symptoms,
Diets = b.Diets, Relevance = DoPoints(b.Code, (ptsActual + ptsPhrase)), Phrase =
search, IllOid = b.Oid, Field = "Symptoms" };
        recItem.Save();
        cntFound++;
        sumRelevance += recItem.Relevance;
        // if (topScore < recItem.Relevance) topScore = recItem.Relevance;
    }
    if (xpcRecs.Count < 1 && (search.Length >= 3)) {
        xpcRecs = LoadData(recSrch.Session,
String.Format("Upper([Symptoms])Like'%,{0}%", search));
        recSrch.TotalRows += xpcRecs.Count;
        foreach (Illnesses b in xpcRecs) {
            recItem = new SearchItems(recSrch.Session) { Search = recSrch,
Name = b.Name, Abbreviation = b.Abbreviation, Code = b.Code, Symptoms = b.Symptoms,
Diets = b.Diets, Relevance = DoPoints(b.Code, (ptsActual + ptsPhrase) / 2), Phrase =
search, IllOid = b.Oid, Field = "Symptoms_" };
            recItem.Save();
            cntFound++;
            sumRelevance += recItem.Relevance;
            // if (topScore < recItem.Relevance) topScore =
recItem.Relevance;
        }
        if (xpcRecs.Count < 1 && (search.Length >= 3)) {
            xpcRecs = LoadData(recSrch.Session,
String.Format("Upper([Symptoms])Like'%{0},%' ", search));
            recSrch.TotalRows += xpcRecs.Count;
            foreach (Illnesses b in xpcRecs) {
                recItem = new SearchItems(recSrch.Session) { Search =
recSrch, Name = b.Name, Abbreviation = b.Abbreviation, Code = b.Code, Symptoms =
b.Symptoms, Diets = b.Diets, Relevance = DoPoints(b.Code, (ptsActual + ptsPhrase) /
2), Phrase = search, IllOid = b.Oid, Field = "_Symptoms" };
                recItem.Save();
                cntFound++;
                sumRelevance += recItem.Relevance;
                // if (topScore < recItem.Relevance) topScore =
recItem.Relevance;
            }
        }
    }
}
#endregion

#region Diets
xpcRecs = LoadData(recSrch.Session,
String.Format("Upper([Diets])Like'%,{0}%", search));
recSrch.TotalRows += xpcRecs.Count;
foreach (Illnesses b in xpcRecs) {
    recItem = new SearchItems(recSrch.Session) { Search = recSrch,
Name = b.Name, Abbreviation = b.Abbreviation, Code = b.Code, Symptoms = b.Symptoms,
Diets = b.Diets, Relevance = DoPoints(b.Code, (ptsActual + ptsPhrase)), Phrase =
search, IllOid = b.Oid, Field = "Diets" };
    recItem.Save();
    cntFound++;
    sumRelevance += recItem.Relevance;
    // if (topScore < recItem.Relevance) topScore = recItem.Relevance;
}
if (xpcRecs.Count < 1 && (search.Length >= 3)) {
    xpcRecs = LoadData(recSrch.Session,
String.Format("Upper([Diets])Like'%,{0}%", search));
    recSrch.TotalRows += xpcRecs.Count;
    foreach (Illnesses b in xpcRecs) {

```

```

        recItem = new SearchItems(recSrch.Session) { Search = recSrch,
Name = b.Name, Abbreviation = b.Abbreviation, Code = b.Code, Symptoms = b.Symptoms,
Diets = b.Diets, Relevance = DoPoints(b.Code, (ptsActual + ptsPhrase) / 2), Phrase =
search, IllOid = b.Oid, Field = "Diets_" };
        recItem.Save();
        cntFound++;
        sumRelevance += recItem.Relevance;
        // if (topScore < recItem.Relevance) topScore =
recItem.Relevance;
    }
    if (xpcRecs.Count < 1 && (search.Length >= 3)) {
        xpcRecs = LoadData(recSrch.Session,
String.Format("Upper([Diets])Like'%'{0},%'", search));
        recSrch.TotalRows += xpcRecs.Count;
        foreach (Illnesses b in xpcRecs) {
            recItem = new SearchItems(recSrch.Session) { Search =
recSrch, Name = b.Name, Abbreviation = b.Abbreviation, Code = b.Code, Symptoms =
b.Symptoms, Diets = b.Diets, Relevance = DoPoints(b.Code, (ptsActual + ptsPhrase) /
2), Phrase = search, IllOid = b.Oid, Field = "_Diets" };
            recItem.Save();
            cntFound++;
            sumRelevance += recItem.Relevance;
            // if (topScore < recItem.Relevance) topScore =
recItem.Relevance;
        }
    }
}
}
#endregion

prevSearch = search;
ptsActual--;
if (cntFound > 0) {
    sumItems += cntFound;
    cntFound = 0;
    if (pluralSearch != search)
        recSrch.CutOff += ptsActual;
}
if (Soundex.FixPlural(search, out pluralSearch))
    if (("," + recSrch.Tokens).Contains("," + pluralSearch + ",") ==
false) {
        search = pluralSearch;
        ptsActual++;
        goto redo;
    }
}
// Build AveRelevance
try {
    if (sumItems > 0) {
        recSrch.AvgRelevance = Math.Round(sumRelevance / (sumItems + 1),
3);
        // recSrch.AvgRelevance = (recSrch.AvgRelevance + topScore + 1) /
2;
    } else recSrch.AvgRelevance = 1;
} catch { }
try { recSrch.Save(); } catch (Exception ex) {
SysModule.LogInfo(ex.ToString()); }

SysModule.LogInfo("Search Tokens: " + recSrch.Tokens);
return recSrch;
}

private XPView VwItems {

```

```

        get {
            XPView vw = new XPView(Session, typeof(SearchItems));
            vw.Criteria = CriteriaOperator.Parse("[Search.Reference]=" +
Reference + "");
            vw.AddProperty("Ill0id", CriteriaOperator.Parse("Ill0id"), true);
            vw.AddProperty("Name", CriteriaOperator.Parse("Name"), true);
            vw.AddProperty("Code", CriteriaOperator.Parse("Code"), true);
            vw.AddProperty("Symptoms", CriteriaOperator.Parse("Symptoms"), true);
            vw.AddProperty("Diets", CriteriaOperator.Parse("Diets"), true);
            vw.AddProperty("Relevance", CriteriaOperator.Parse("Sum(Relevance)",
false);
            vw.AddProperty("Count", CriteriaOperator.Parse("Count()"), false);

            vw.Sorting.Add(new SortProperty("Code", SortingDirection.Ascending));
            vw.Sorting.Add(new SortProperty("Relevance",
SortingDirection.Descending));
            vw.Sorting.Add(new SortProperty("Name", SortingDirection.Ascending));
            vw.Sorting.Add(new SortProperty("Symptoms",
SortingDirection.Ascending));
            vw.Sorting.Add(new SortProperty("Diets", SortingDirection.Ascending));

            return vw;
        }
    }

    public void ApplyCutOff(XPView vw, bool takeItemsAbove) {
        if (vw.Count > 0) {
            if (takeItemsAbove) {
                // Use The Higher
                if (CutOff < AvgRelevance) vw.Filter =
CriteriaOperator.Parse("[Relevance]>" + AvgRelevance.ToString("##0.0#"));
                else vw.Filter = CriteriaOperator.Parse("[Relevance]>" +
CutOff.ToString("##0.0#"));
            }
            if (vw.Count <= 0) {
                if (CutOff < AvgRelevance) vw.Filter =
CriteriaOperator.Parse("[Relevance]>=" + AvgRelevance.ToString("##0.0#"));
                else vw.Filter = CriteriaOperator.Parse("[Relevance]>=" +
CutOff.ToString("##0.0#"));
                // Check to Use the Lower
                if (vw.Count <= 0) {
                    if (CutOff > AvgRelevance) vw.Filter =
CriteriaOperator.Parse("[Relevance]>" + AvgRelevance.ToString("##0.0#"));
                    else vw.Filter = CriteriaOperator.Parse("[Relevance]>" +
CutOff.ToString("##0.0#"));
                }
                if (vw.Count <= 0) {
                    if (CutOff > AvgRelevance) vw.Filter =
CriteriaOperator.Parse("[Relevance]>=" + AvgRelevance.ToString("##0.0#"));
                    else vw.Filter =
CriteriaOperator.Parse("[Relevance]>=" + CutOff.ToString("##0.0#"));
                }
            }
        }
        } else {
            // Use The Higher
            if (CutOff < AvgRelevance) vw.Filter =
CriteriaOperator.Parse("[Relevance]<=" + AvgRelevance.ToString("##0.0#"));
            else vw.Filter = CriteriaOperator.Parse("[Relevance]<=" +
CutOff.ToString("##0.0#"));
            if (vw.Count <= 0) {
                if (CutOff < AvgRelevance) vw.Filter =
CriteriaOperator.Parse("[Relevance]<" + AvgRelevance.ToString("##0.0#"));
            }
        }
    }
}

```



```

static decimal tempNum = 0;
static decimal DoPoints(string pRankCode, decimal pPoints) {
    tempNum = pPoints;
    try {
        if (!String.IsNullOrEmpty(pRankCode))
            if
(decimal.TryParse((pRankCode.ToUpper().Trim().Trim('R')).TrimStart('0'), out tempNum))
{
                tempNum = pPoints + (5 - (tempNum * 2));
            } else
                tempNum = pPoints;
        } catch { }
    }
    return tempNum;
}

static string TrimSymptoms(string pSymptoms, out int pLenRemainder) {
    pSymptoms = pSymptoms.Replace(" ", "").Trim();
    if (pSymptoms.Length > 12) {
        string[] arr = (pSymptoms.Replace("/", ",")).Split(',');
        string temp = "";
        int cnt = 0;
        foreach (string num in arr) {
            if (num.Trim().Length < 4) continue;
            temp += ", " + num.Trim();
            if ((++cnt) > 1) break;
        }
        if (cnt > 1) pSymptoms = temp;
    }
    // Finalise
    pSymptoms = (pSymptoms.Length < 30 ? pSymptoms : pSymptoms.Substring(0,
29)).Trim(',').Trim().Trim(',');
    pLenRemainder = 30 - pSymptoms.Length;
    return pSymptoms;
}
}
}
}

```

Password Policy

```

using System;
using DevExpress.Xpo;
using DietPrescriber.AppClasses.SysPrescriber;
using DevExpress.Data.Filtering;

namespace DietPrescriber.AppClasses.SysAdmin {

    public class PassPolicy : SysAuditObject {
        public PassPolicy(Session session)
            : base(session) {
            // This constructor is used when an object is loaded from a persistent
storage.
            // Do not place any code here.
        }

        public override void AfterConstruction() {
            base.AfterConstruction();
            // Place here your initialization code.
        }

        int fPassExpiry;
        public int PassExpiry {

```

```

        get { return fPassExpiry; }
        set { SetPropertyValue("PassExpiry", ref fPassExpiry, value); }
    }
    int fLogAttempts;
    public int LogAttempts {
        get { return fLogAttempts; }
        set { SetPropertyValue("LogAttempts", ref fLogAttempts, value); }
    }
    int fPassHistory;
    public int PassHistory {
        get { return fPassHistory; }
        set { SetPropertyValue("PassHistory", ref fPassHistory, value); }
    }
    int fPassLength;
    public int PassLength {
        get { return fPassLength; }
        set { SetPropertyValue("PassLength", ref fPassLength, value); }
    }
    bool fCheckSimilarity;
    public bool CheckSimilarity {
        get { return fCheckSimilarity; }
        set { SetPropertyValue("CheckSimilarity", ref fCheckSimilarity, value); }
    }

    public string Username;

    public static PassPolicy Default {
        get { return SysModule.XpoDefaultSession.FindObject<PassPolicy>(null); }
    }

    protected override void OnSaving() {
        base.OnSaving();

        //Validate here
        if (Oid < 1) {
        }

        try { Username = SysModule.HttpSessionState["username"].ToString(); }
    catch { }
    }
}

```

System Reports

```

using System;

/// <summary>
/// Summary description for SysStructures
/// </summary>
namespace DietPrescriber.AppClasses {
    public static class SysStructures {
        public struct ExportTypes {
            public const string Csv = "Csv";
            public const string Pdf = "Pdf";
            public const string Rtf = "Rtf";
            public const string Xls = "Xls";

            public static string[] ToStringArray() {
                string[] arr = {
                    Csv,

```

```

        Pdf,
        Rtf,
        Xls
    };
    return arr;
}
}

public struct ScopeTypes {
    public const string Trenching = "Trenching";
    public const string NonTrenching = "Non-Trenching";
    public const string UnDefined = "Un-defined";
    public const string Archived = "Archived";
    public const string Stalled = "Stalled";

    public static string[] ToStringArray() {
        string[] arr = {
            Trenching,
            NonTrenching,
            UnDefined,
            Archived,
            Stalled,
            "All " + Trenching,
            "All"
        };
        return arr;
    }
}

public struct DocTypes {
    public const string Product= "Product";
    public const string Others = "Others";

    public static string[] ToStringArray() {
        string[] arr = {
            Product,
            Others
        };
        return arr;
    }
}

public struct Reports_General {
    public const string ExecutiveSummary = "Executive Summary";
    public const string ExecutiveSummaryDetailed = "Detailed Executive
Summary";
    public const string NumOfProductsByStatus = "Number of Products Per
Status"; // "Number of Products Completed", "Number of Products Outstanding";
    public const string PortfolioAnalysis = "Portfolio Analysis";

    public const string ListIllnessLinks = "Illness Links List";
    public const string ListProducts = "Products List";
    public const string ListProductsAppraised = "Products Received List";

    public static string[] ToStringArray() {
        string[] arr = {
            ExecutiveSummary,
            ExecutiveSummaryDetailed,
            NumOfProductsByStatus,
            PortfolioAnalysis,

```

```

        ListIllnessLinks,
        ListProducts,
        ListProductsAppraised
    };

    return arr;
}

public struct Reports_Activity {
    public const string ActivitiesExecutiveSummary = "Executive Summary";
    public const string ProductsActivitySlips = "Products-Activity Slips";
    public const string ProductsActivitiesUpdated = "Products Activities
Updated";

    public static string[] ToStringArray() {
        string[] arr = {
            ActivitiesExecutiveSummary,
            ProductsActivitySlips,
            ProductsActivitiesUpdated
        };

        return arr;
    }
}

public struct Reports_Product {
    public const string ProductsExecutiveSummary = "Executive Summary";
    public const string ProductsProductsummary = "Products Summary";
    public const string NumOfProductsBySymptoms = "Number of Products By Build
Status"; // "Number of Products Completed", "Number of Products Outstanding";
    public const string NumOfProductsByLastActivity = "Number of Products By
Last Activity"; // "Number of Products Completed", "Number of Products Outstanding";
    public const string CummulativeProjectsAppraised = "Cumulative Projects
Appraised";

    public const string ListIllnessLinks = "Illness Links List";
    public const string ListIllnessLinksDetailed = "Illness Links List -
Detailed";

    public const string ListProducts = "Products List";
    public const string ListProductsAppraised = "Products Received List";
    public const string ListProductsWithoutProducts = "Products Without
Products";

    public static string[] ToStringArray() {
        string[] arr = {
            ProductsExecutiveSummary,
            ProductsProductsummary,
            NumOfProductsBySymptoms,
            NumOfProductsByLastActivity,
            CummulativeProjectsAppraised,

            ListIllnessLinks,
            ListIllnessLinksDetailed,
            ListProducts,
            ListProductsAppraised,
            ListProductsWithoutProducts
        };

        return arr;
    }
}

public struct Reports_Graphs {
    // S0

```

```

        public const string IllnessesPerProductCategory = "Illnesses Per
ProductCategory";
        public const string ProductsPerCategory = "Products Per ProductCategory";

        // Proj
        public const string ProductsPerSymptoms = "Products Per Symptoms";
        public const string ProductsPerLastAction = "Products Per LastAction";
        public const string ProductsCompleted = "Products Completed/Billable";

        // Act
        public const string ActivitiesInProgressProducts = "Activities In Progress
- Products";
        public const string ActivitiesCompleted = "Key-Activities Completed";

        public static string[] ToStringArray() {
            string[] arr = {
                // S0
                IllnessesPerProductCategory,
                ProductsPerCategory,

                // Products
                ProductsPerSymptoms,
                ProductsPerLastAction,
                ProductsCompleted,

                // Activities
                ActivitiesInProgressProducts,
                ActivitiesCompleted
            };
            return arr;
        }
    }
}

```