ZOU

FLEET MANAGER SYSTEM

Submitted in partial fulfilment of the requirements

For a Diploma in Information Technology

DECLARATION

I, hereby declare that I am the sole author of this project. I authorize Kushinga Phikelela Polytechnic to lend this project to other institutions or individuals for the purpose of academic research.

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

APPROVAL

This project entitled “Zou fleet manager” meets the regulations governing the award of a Diploma in Information Technology, and is approved for its contribution to knowledge and literal presentation.

Supervisor …………………………………………………

Date ………………………………………………………

ABSTRACT

This research’s main goal was to design and develop ZOU FLEET MANAGER SYSTEMfor Zimbabwe open university.The purpose of doing this research is to control the overall cost of operating and maintaining ZOU ‘S fleet of vehicles and, to maintain vehicles and equipment in a manner that extends their useful life, to control the growth in size of the fleet, to standardize the composition of the fleet and to accurately budget for maintenance and replacement costs.

In planning for the project, feasibility of the project was of interest as it related to the university’s business values and the benefits of developing such an application. It considered the economic, social, technical and operational feasibility analysis which helped to safely endorse the development of the ZOU FLEET MANAGER SYSTEM application as a necessity. An analysis of the existing system was thoroughly done through the implementation of some information gathering methodologies which gathered data for evaluation purposes. Analysis was done through activity diagrams, context and dataflow diagrams to clarify the nature of the existing system. Gathered information about the existing system concluded that a new system was to be developed given that the feasibility analysis was in support of the development of a new system.

**ACKNOWLEDGEMENT**

I would like to express my eternal gratitude to the Lord Almighty for all the guidance. My heartfelt thanks go to Kushinga Phikelela Polytechnic Information Technology Department staff with special mention of Mrs chabanga my supervisor, for their expertise, time and heartfelt dedication to the success of this project. Many thanks also go to Zimbabwe open university transport Department for allowing me to carry out the software project, family and friends for their continued support and encouragement.

Lastly, special thanks goes to my parents for their parental and financial support right from my nursery school up to this level.

Contents

[Chapter 1 1](#_Toc65855790)

[1.0 INTRODUCTION 1](#_Toc65855791)

[1.1 BACKGRAOUND OF STUDY 1](#_Toc65855792)

[1.1.1 ORGANISATIONAL STRUCTURE 3](#_Toc65855793)

[1.2 PROBLEM STATEMENT 3](#_Toc65855794)

[1.3 AIM 4](#_Toc65855795)

[1.4 OBJECTIVES 4](#_Toc65855796)

[1.5 JUSTIFICATION 5](#_Toc65855797)

[1.6 CONCLUSION 5](#_Toc65855798)

[Chapter 2 6](#_Toc65855799)

[2.0 INTRODUCTION 6](#_Toc65855800)

[2.1 WHY BUILD THE SYSTEM. 6](#_Toc65855801)

[2.2 BUSSINESS VALUE. 8](#_Toc65855802)

[2.3 FEASIBILITY STUDY. 8](#_Toc65855803)

[2.3.1 TECHICAL FEASIBILITY. 9](#_Toc65855804)

[2.3.1.1 TECHNICAL EXPERTISE 10](#_Toc65855805)

[2.3.1.1 HUMAN RESOURCES REQUIRED 11](#_Toc65855806)

[2.4.1 TECHNOLOGICAL FEASIBILITY. 11](#_Toc65855807)

[Table 2.4.1.1 HARDWARE REQUIREMENTS 12](#_Toc65855808)

[Table 2.4.1.2 SOFTWARE REQUIREMENTS 13](#_Toc65855809)

[2.5.1 ECONOMIC FEASIBILITY 14](#_Toc65855810)

[2.5.1 COSTS 14](#_Toc65855811)

[TABLE 2.5.1.1: PROJECT COSTS 16](#_Toc65855812)

[2.5.2.1 BENEFITS 16](#_Toc65855813)

[2.5.2.2 TANGIBLE BENEFITS 17](#_Toc65855814)

[2.5.2.2 INTANGIBLE BENEFITS 17](#_Toc65855815)

[TABLE 2.5.2.3 PROJECT BENEFITS 18](#_Toc65855816)

[2.5.3 COST BENEFIT ANALYIS 18](#_Toc65855817)

[TABLE 2.5.3.1 COST BENEFIT ANALYSIS 18](#_Toc65855818)

[2.5.3.2 RETURN ON INVESTMENTS (R.O.I) 20](#_Toc65855819)

[2.5.4 OPERATIONAL FEASIBILITY 21](#_Toc65855820)

[2.5.5 WORK PLAN 22](#_Toc65855821)

[2.5.5.1 PHASES ACTIVITIES. 25](#_Toc65855822)

[2.5.5. Gantt Chart. 25](#_Toc65855823)

[2.8 CONCLUSION 26](#_Toc65855824)

[Chapter 3 26](#_Toc65855825)

[3.0 INTRODUCTION 26](#_Toc65855826)

[3.1 A GENERAL MODEL OF DESIGN PROCESS 27](#_Toc65855827)

[3.2 INFORMATION GATHERING METHODOLOGIES 27](#_Toc65855828)

[3.2 INFORMATION GATHERING PROCESS 28](#_Toc65855829)

[3.2.1 OBSERVATION 28](#_Toc65855830)

[3.2.1 ADVANTAGIES OF OBSERVATION METHODOLOGIES 29](#_Toc65855831)

[3.2.2 DISADVANTAGIES OF OBSERVATION METHODOLOGIES 29](#_Toc65855832)

[3.3 DOCUMENTS REVIEW 30](#_Toc65855833)

[3.3.1 Advantages of documents review 30](#_Toc65855834)

[3.3.2 disadvantages of documents review 30](#_Toc65855835)

[3.4 interviews 31](#_Toc65855836)

[3.4.1 Advantages of interviews 31](#_Toc65855837)

[3.4.2 disadvantages of interviews 31](#_Toc65855838)

[3.5 focus groups 32](#_Toc65855839)

[3.5.1 advantages of focus groups 32](#_Toc65855840)

[3.5.2 disadvantages of focus groups 32](#_Toc65855841)

[3.6 Analysis of existing system 32](#_Toc65855842)

[3.6. 0 log book. 33](#_Toc65855843)

[3.6.1 process analysis 34](#_Toc65855844)

[3.6.1 table: process, inputs and outputs activities of the existing system. 34](#_Toc65855845)

[3.6.2 activity diagram of the old system 35](#_Toc65855846)

[3.7 Data analysis 35](#_Toc65855847)

[3.7.1 context diagram 36](#_Toc65855848)

[3.7.1.0 context diagram of the old system 36](#_Toc65855849)

[3.7.2 data flow diagram of the old system 36](#_Toc65855850)

[3.7.2 data flow diagram of the old system 37](#_Toc65855851)

[**3.7.3 weaknesses of the current system** 37](#_Toc65855852)

[**3.8 evaluation of alternatives** 38](#_Toc65855853)

[**3.8.1 outsourcing** 38](#_Toc65855854)

[**3.8.2 improving the current system** 39](#_Toc65855855)

[**3.8 .3 Development** 40](#_Toc65855856)

[**3.9 Requirements analysis** 41](#_Toc65855857)

[3.9.1 functional requirements (USE CASE DIAGRAM) 41](#_Toc65855858)

[**3.9.1 USE CASE DIAGRAM** 42](#_Toc65855859)

[**3.9.2 Nonfunctional requirements** 43](#_Toc65855860)

[**3.10 Conclusion** 45](#_Toc65855861)

[**Chapter 4** 46](#_Toc65855862)

[**4.0 Introduction** 46](#_Toc65855863)

[**Table: 4.0 general model of design process** 46](#_Toc65855864)

[**Table 4.0.1: General Summary of the Design phase.** 47](#_Toc65855865)

[**4.1 System design** 48](#_Toc65855866)

[**Table 4.1.1 System design elements and specifications** 49](#_Toc65855867)

[**4.2 Description of the proposed system** 50](#_Toc65855868)

[**Table 4.2.1: System Design Specification and Elements.** 53](#_Toc65855869)

[**4.3 CONTEXT DIAGRAM AND DFD OF THE PROPOSED SYSTEM** 53](#_Toc65855870)

[**Table 4.3.1: Context Diagram of the proposed system.** 54](#_Toc65855871)

[**4.3.2 Data Flow Diagram:** 54](#_Toc65855872)

[**Table 4.3.2: Data Flow Diagram of the proposed system.** 56](#_Toc65855873)

[**4.4 ARCHITECTURAL DESIGN** 56](#_Toc65855874)

[**4.4.1 Two-Tier Architecture:** 57](#_Toc65855875)

[**Table 4.4.1: Two-Tier Architecture Diagram of the proposed system.** 58](#_Toc65855876)

[**4.4.2 Limitations of a Two-Tier Approach** 58](#_Toc65855877)

[**4.4.3 Benefits of a Two-Tier Approach** 59](#_Toc65855878)

[**4.5 PHYSICAL DESIGN** 60](#_Toc65855879)

[***Table 4.5: Physical diagram of the proposed system.*** 60](#_Toc65855880)

[***The*** WAN connection is facilitated by a dedicated point to point link of fiber optic since the distance to the remote stations may spun up to more than 500km, hence only the fiber optic link can carter for long distance transmission problems. Fiber optics has a large capacity to carry high speed signals over longer distances without repeaters than other types of cables. The information carrying capacity increases with frequency. 60](#_Toc65855881)

[**4.5.1 DATABASE DESIGN** 62](#_Toc65855882)

[**a. Requirements Specifications:** 63](#_Toc65855883)

[Table 4.5.1: Database design stages 63](#_Toc65855884)

[b. **Database Design:** 63](#_Toc65855885)

[**1. Conceptual database design:** 64](#_Toc65855886)

[Table 4.5.2: Entity Relationship Diagram of the new system 65](#_Toc65855887)

[**2. Logical database design** 65](#_Toc65855888)

[**Database tables** 65](#_Toc65855889)

[***Table 4.5.3: users fields.*** 66](#_Toc65855890)

[***Table 4.5.4: vehicles fields.*** 67](#_Toc65855891)

[***Table 4.5.5: trips fields.*** 67](#_Toc65855892)

[***Table 4.5.6: audit trails fields.*** 67](#_Toc65855893)

[***Table 4.5.7: web notices fields.*** 68](#_Toc65855894)

[***Table 4.5.8: vehicle fields.*** 68](#_Toc65855895)

[**Table 4.5.9: departments fields.** 68](#_Toc65855896)

[**3. Physical database design** 69](#_Toc65855897)

[**4.6 PROGRAM DESIGN** 69](#_Toc65855898)

[**4.6.1 Package Diagram** 70](#_Toc65855899)

[**4.6.2 Class Diagram** 70](#_Toc65855900)

[**4.6.3 Sequence Diagram** 71](#_Toc65855901)

[**4.7 INTERFACE DESIGN** 72](#_Toc65855902)

[**4.7.1 INTERAFCE DESIGN STRUCTURE** 74](#_Toc65855903)

[**4.7.3 Main Menu design** 76](#_Toc65855904)

[**4.8 CONCLUSION** 77](#_Toc65855905)

[**Chapter 5** 78](#_Toc65855906)

[**5.0 introduction** 78](#_Toc65855907)

[**5.1 SYSTEM CODE** 78](#_Toc65855908)

[***5.1.1 PROGRAM CODE SEGMENTS*** 78](#_Toc65855909)

[***Code for adding vehicle*** 83](#_Toc65855910)

[**5.2 SYSTEM CODING** 90](#_Toc65855911)

[**5.3 PROGRAMMING STYLE** 90](#_Toc65855912)

[**5.4.0 TESTING** 92](#_Toc65855913)

[**5.4.1 SYSTEM TESTING** 92](#_Toc65855914)

[**5.5.1 TESTING PROCESS** 93](#_Toc65855915)

[**6.6.0 VALIDATION** 96](#_Toc65855916)

[**6.1.0 VERIFICATION** 97](#_Toc65855917)

[**CHAPTER 6** 98](#_Toc65855918)

[**6** **IMPLEMENTATION & POST IMPLEMENTATION PLAN** 98](#_Toc65855919)

[**6.0 SYSTEM INSTALLATION** 98](#_Toc65855920)

[**6.1** **SYSTEM CONVERSION** 99](#_Toc65855921)

[**6.1.1 PILOT CONVERSION** 99](#_Toc65855922)

[**6.1.2 DIRECT CONVERSION** 99](#_Toc65855923)

[**6.1.3 PHASED CONVERSION** 100](#_Toc65855924)

[**6.1.4 PARALLEL CONVERSION** 100](#_Toc65855925)

[**6.2.0 TRAINING** 101](#_Toc65855926)

[**6.3.1 MAINTENANCE** 101](#_Toc65855927)

[**6.3.2 SYSTEM MAINTENANCE** 102](#_Toc65855928)

[**6.3.2.1 CORRECTIVE MAINTENANCE** 103](#_Toc65855929)

[**6.3.2.2 ADAPTIVE MAINTENANCE** 103](#_Toc65855930)

[**6.3.2.3 PERFECTIVE MAINTENANCE** 104](#_Toc65855931)

[**7.3** **SECURITIES** 104](#_Toc65855932)

[**7.3.1 SOFTWARE APPLICATION BACKUP SERVICES** 104](#_Toc65855933)

[**6.4.2 HARDWARE BACKUP SERVICES** 104](#_Toc65855934)

[**6.4.3 SYSTEM REVIEW** 104](#_Toc65855935)

[**6.5 INSTALLATION** 105](#_Toc65855936)

[**6.6 USER TRAINING** 105](#_Toc65855937)

[**CHAPTER 7** 107](#_Toc65855938)

[**7.0** **SUMMARY, CONCLUSION & RECOMMENDATIONS** 107](#_Toc65855939)

[**7.1 SUMMARY** 107](#_Toc65855940)

[**7.1.1 PROJECT REVIEW** 107](#_Toc65855941)

[**7.1.2 CHALLENGES ENCOUNTERED** 108](#_Toc65855942)

[**7.2 CONCLUSION** 108](#_Toc65855943)

[**7.3 RECOMMENDATIONS** 108](#_Toc65855944)

[**CHAPTER 8** 109](#_Toc65855945)

[**8.0 BIBLIOGRAPHY** 109](#_Toc65855946)

[2002, 9, pp. 334-355. 110](#_Toc65855947)

# Chapter 1

## INTRODUCTION

ZOU transport managers face an increasing burden of paper work from their business of being stewards of company vehicles. Storing paper files is no longer good enough. It does not enable remote staff to access documents, and records are often inaccurate or incomplete leaving the operator exposed and open to prosecution.

The solution incorporates a central repository that stores a wide variety of documents relating to the fleet. Thus the author is proposing to design a system called fleet manager for the named company above. The proposed system has the following features and benefits:

* Storage and management of driver’s procedure documentation.
* Remote web access for local fleet managers of vehicle documentation.
* Rapid access to key documents in the event of an inspection of vehicle accident.

The author also learnt that the university has also adopted a vehicle replacement program which determines the replacement dates for vehicles. The criteria for replacement include age, usage and maintenance costs. In general, vehicles are replaced at five years or 100,000kilometers. All new purchases for vehicles are part of the budget cycle, so this system would also want to give a hand on such a program with functions and measures that smoothens its process.

## 1.1 BACKGRAOUND OF STUDY

According to Harold R. Kerzner (2003), background of study refers to information essential to understanding of a problem or situation. It can be well understood as the statement identifying where the study well emanated from.

The Zimbabwe Open University (ZOU) is a unique institution of higher learning offering Open and Distance e-Learning (ODeL) education and it is the only ODeL University in the country. It started off in 1993 as the Centre for Distance Education under the University of Zimbabwe and has since grown into a fully fledged university.

In order to sufficiently cater for students throughout Zimbabwe and globally, ZOU adopted a highly decentralized structure of opening Regional Centre’s in all the ten (10) administrative provinces of Zimbabwe as well as a Virtual Region Centre for all international students.

Organizations are constantly in need of efficient systems that will enable them handle daily operations such as;

* How to drive sustainable savings?
* How to do with fewer resources?
* How to optimize transportation costing via flexible routing.
* How to improve performance execution and how to ensure on time deliveries, leverage transportation best practices are constantly asked.

With the fact that it is possible to solve such problems using a computer system and careful programming ,software developers have done this using algorithms that synchronizes with the transportation operations of established involved. This need for efficient communication and co-ordination of the movement and operations under Zimbabwe Open University to the outside organization necessitates automated transport system.

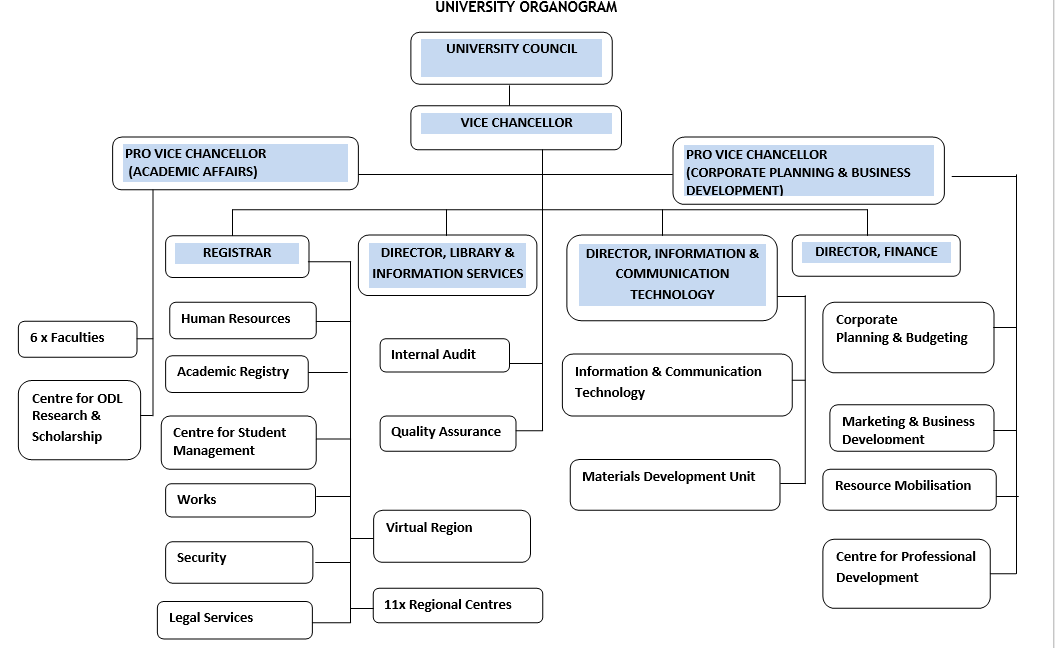
ZOU’s open and distance learning model offers a number of advantages to both learners and to providers of opportunities for learning. Problems such as distance and time, which are barriers to conventional learning, are overcome in open and distance learning.

Zimbabwe Open University relies heavily on its vehicles for the success of its business and rendering services to its valued customers who are very much dispersed across the face of Zimbabwe. Therefore the mismanagement of the fleet will adversely impact the success of ZOU since its business great relies on transport and needs mobility for its lectures and students, hence the need of ZOU Fleet Manager as a means to ensure guarantee of service from all known vehicles. The following are some of the reasons why this system is of value to the daily business of ZOU towards the achievement of its goals:

* Flexibility to meet ever-changing needs and hence will remain of value to the company.
* Creation of a single database as opposed to the current system where information is fragmented across different subsystems (Provinces, Head Office)
* Increase in worker satisfaction resulting in increased better service delivery.
* Analysis will be very easy in proposed system as it is automated .

## 1.1.1 ORGANISATIONAL STRUCTURE

According to McGrill company’s organizational structure is “Division of labour and patterns of coordination, communication, workflow, and formal power that direct organizational activities”. Organizational structure refers to the hierarchy of an organization and how the components of this hierarchy work together to achieve the objectives of the company. An organogram below depict the organizational structure of Zimbabwe Open University.



## 1.2 PROBLEM STATEMENT

This is a summary of the problem to be solved. This statement illustrates the scope of the problems to be solved by the project that is to be undertaken. It emphasizes on the scope of what does this project need to do for you by pin pointing to us. The statement will point to the next step which is planning what to do about the differences that is the planning is what is to be done, Bentley, L (2007) to address the problems. The problems being faced with the current operations are noted and then refined to form a list of separate problems that need to be addressed. The project Problem Definition, list of problems to be addressed by the project that have been noted are as follows.

* Increasing burden of paper work.
* Poor access to key documents in the event of an inspection or vehicle incident.
* Invisibility of vehicles’ historical safety checks and records
* No remote access for local fleet managers of vehicle documentation.
* Inaccurate or incomplete vehicle and drivers records leaving the operator exposed and open to prosecution.
* High overall cost of operating and maintaining ZOU’S fleet of vehicles

## 1.3 AIM

ZOU Fleet Manager’s primary objective is to control the overall cost of operating and maintaining ZOU ‘S fleet of vehicles and, to maintain vehicles and equipment in a manner that extends their useful life, to control the growth in size of the fleet, to standardize the composition of the fleet and to accurately budget for maintenance and replacement costs.

## 1.4 OBJECTIVES

**According to Selah (2009), objectives are a specific result that a person or system aims to achieve within a time frame and with available resources. According to the business dictionary, objectives are basic tools that underlie all planning and strategic activities which serve as the basis for creating policy and evaluating performance.**

The proposed system seeks to achieve the following objectives

* To create a data warehouse of all trip requests made by departments for decision making.
* To calculate depreciation of vehicles and suggesting its disposal price.
* To keep the record of existing vehicles, type, and colour, date of purchase, picture and life span.

## 1.5 JUSTIFICATION

Surely any organization with a might fleet like ZOU will embrace such a system because it is like a packaged remedy for all its problems as far as monitoring vehicles individually is concerned. This will surely save a lot a time and costs for the power utility and allow it to shift attention to other major core business which may need more attention. Hence the study and research in this proposed field is a great investment with huge returns for both long and short term periods. In a nut shell the following are benefits drawn from the adoption of the proposed system:

Save Costs:

There are several substantial reductions of cost through enhanced efficiency in operations and reduction of clerical tasks which were supposed to be done to create and keep records for the whole fleet and unnecessary inventory.

Long Term Advantages:

Fleet managers need a comprehensive, usable and powerful system that helps them manage and make strategic decisions at a fleet-wide level for short and long-term strategies.

Simplify daily work:

The solution is unique in using a highly visual interface for both viewing and organizing resources and information. Fleet Manager is a user-friendly and simple solution which makes it easier and quicker to allocate resources (vehicles) as well as to monitor and continuously optimize operations for ZOU.

## 1.6 CONCLUSION

Zimbabwe Open University is a very big company just as highlighted above with innumerable fleet of vehicles and the current system which is used to manage it cannot effectively take care of that business without a system like ZOU Fleet Manager. The research done by the author has also proved that the present problems highlighted earlier are enough evidence to call for adoption of the new system and importance of this study. The justification phase has also displayed the merits of the study and this has wonderfully exhibited the glorious operation under this proposed system.

# Chapter 2

## 2.0 INTRODUCTION

Planning is deciding in advance - what to do, when to do & how to do. It bridges the gap from where we are & where we want to be”. A plan is a future course of actions. It is an exercise in problem solving & decision making. Planning is determination of courses of action to achieve desired goals.

This chapter is focused on why build the system, a critical analysis of whether it is worth-while building this system and a clear plan of how proceed with the research study. It critically tests to see whether the development of the system should be continued; there is a focus on the system's business value, its contribution to the organization which are tangibles and intangibles, if the organization will experience any losses due to the investing into this system. This chapter will focuses on the feasibility of the system technically if there is sufficient technical know-how and expertise. Economically if the organization will benefit materially, gain a monetary benefit from the introduction of the system directly or indirectly.

This chapter aims to:

1. Provide a comprehensive and innovative introduction of ZOU FLEET MANAGERProject scope.

2. To equip any potential users and ZOU management with comprehensible information to:

* Identify, formulate and appraise this project.
* Determine the feasibility of undertaking this project as an investment.

## 2.1 WHY BUILD THE SYSTEM.

The process of purchasing, keeping track of, and maintaining vehicles records can be a time-consuming process, and for large fleets, it can be complex. The following evidence emphasize why a system like Fleet Manager should be build:

* SIMPLIFIES MANAGEMENT.

Fleet Management is a task which allows companies which rely on transportation in their business to remove or minimize the risks associated with vehicle investment, improving efficiency, productivity and reducing their overall transportation and staff costs. Fleet Manager as a system can take all these functions under its simple scope and allow storage of its important data in a systematic way which is always available to the managers such that it makes work enjoyable.

* AVOIDANCE THE RISK OF MISSING PAPER DOCUMENTS

The manual system which is currently in use for ZOU is not secure enough for the storage of its fleet information. Many at times important documents are not found at the time they are most needed thus leaving valuable company assets vulnerable to fraudsters and its employees to prosecution thus marring company image. However through the use of a database the proposed system is set to eradicate this problem and ensure safety and security for the fleet documents and always available to rightful personnel.

* VEHICLE DIAGNOSTICS

Some fleet management systems have vehicle diagnostic capabilities. By connecting to an on-board computer, you can keep track of information such as mileage and fuel consumption. From such information the system is able to make conclusions concerning the health state of the vehicle that is whether a massive service (over-haul) or a light service is required.

* REPLACEMENT AND LIFECYCLE MANAGEMENT

In order to optimize company fleets, it is important to know the lifecycles of each vehicle so as to know when to buy new vehicles and when others need to be retired. These tasks can encompass all operations from vehicle acquisition to disposal. With a system such as ZOU Fleet Manager these functions are easily done with only a click of a button.

* MANAGEMENT OF INCIDENT LOGS OR UNIT FAILURES.

Fleet Manager can help ZOU management to do their accounting more accurately, and can help them optimize the size of their fleet. It can help working faster, be more efficient, and make fewer mistakes. This function include tasks such as driver/vehicle profiling, trip profiling, dispatching, vehicle efficiency and keeping record of vehicle trips, destinations, time and the driver for that journey.

* RAPID ACCESS TO KEY DOCUMENTS IN THE EVENT OF AN INSPECTION OR VEHICLE INCIDENT.

Comprehensive reports offered by the system facilitate for easy access of important vehicle documents just on time unlike the manual system which takes ages to browse through its archives thus working at the disadvantage of the success of the company.

## 2.2 BUSSINESS VALUE.

According to Sward. D, (2006), he defined business value as “the benefits that can be achieved by embarking on the project that will increase the goodwill of the business in terms of efficiency and effectiveness.”

Zimbabwe open university relies heavily on its vehicles for the success of its business and rendering services to its valued customers who are very much dispersed across the face of Zimbabwe. Therefore the mismanagement of the fleet will adversely impact the success of ZOU since its business great relies on transport and needs mobility for its lectures and students, hence the need of ZOU Fleet Manager as a means to ensure guarantee of service from all known vehicles. The following are some of the reasons why this system is of value to the daily business of ZOU towards the achievement of its goals:

* Flexibility to meet ever-changing business needs and hence will remain of value to the company.
* Improve business productivity and morale amongst users since it will be faster, accurate, user-friendly and reliable system.
* Creation of a single database as opposed to the current system where information is fragmented across different subsystems (Provinces, Head Office)
* Reduction of costs incurred due to the use of stationery.
* Improved decision making process.
* Increase in worker satisfaction resulting in increased better service delivery.
* Analysis will be very easy in proposed system as it is automated

## 2.3 FEASIBILITY STUDY.

The feasibility study’s main goal is to assess or evaluate the economic, technical, social, operational viability of the ZOU Fleet Manager as an investment opportunity for Zimbabwe open university. This study needs to answer the question: “Does the idea make economic sense?” The study provides a thorough analysis of the proposed system, including a look at all the possible roadblocks that may stand in the way of the project’s success. Alvarez (2007) defines feasibility analysis as a measure of economic, technical, operational and organizational feasibility against the businesses current possessions. This analysis also outlines project budget or Cost benefit Analysis. The outcome of the feasibility study indicates whether or not to proceed with the proposed venture. If the results of the feasibility study are positive, then the author can proceed to develop a business plan.

If the results show that the project is not a sound business idea, then the project will not be pursued. Although it is difficult to accept a feasibility study that shows these results, it is much better to find this out sooner rather than later, when more time and money would have been invested and lost.

It is tempting to overlook the need for a feasibility study. Often, the committee undertaking this study may face resistance from potential members on the need to do a feasibility study. Many people will feel that they know the proposed venture is a good idea, so why carry out a costly study just to prove what they already know? The feasibility study is important because it forces the system developers to put its ideas on paper and to assess whether or not those ideas are realistic. It also forces the programmers to begin formally evaluating which steps to take next.

## 2.3.1 TECHICAL FEASIBILITY.

Technical assessment of a proposed system consists of evaluating the required functionality against the hardware, software and technical skills (human resources) available.

Norman, R, J (1999), he cites that “Technical Feasibility is the measure of the practicality of a specific technical information system solution and the availability of technical resources. It measures the extent to which the system can be successfully designed, developed and implemented given all technical constraints”

When an organization has a strategic Information System plan, this is likely to place limitations on the nature of solutions that might be considered. The constraints will in fact influence the cost of the solution and this must be taken into account in the Cost Benefit Analysis.

Technical feasibility can be evaluated by answering the following;

* We currently possess the necessary technology?
* Is the proposed technology or solution practical?
* Do we possess the necessary technical expertise, and is the schedule reasonable?
* Is relevant technology mature enough to be easily applied to our problem?
* Assuming that required technology is practical, there is need to ask whether or not it is easily available from ICT suppliers?
* If the technology is available, does it have the capacity to handle the solution?
* If the technology is not available, can it be acquired through other means?

## 2.3.1.1 TECHNICAL EXPERTISE

This is established if there are individuals best suited for the development of the system before a project commences. The organization has three options to choose from:

* Consulting other software developing companies
* Training one of its employees
* Employing new IT personnel well versed with system development

The table below illustrates the human resources required the development, implementation and maintenance of the proposed system.

|  |  |  |  |
| --- | --- | --- | --- |
| Vacancy | Number of personnel | Qualification and experience | Type of contract |
| System administrators | 3 | HND/Degree in IT from a reputable institution with 2 years experience | Permanent |
| Database administrators | 3 | HND/Degree in IT from a reputable institution with 2 years experience in MYSQL and Oracle databases | Permanent |
| Programmers | 2 | A reputable and sound contractor with knowledge in PHP,Java and C++ | Temporary |

## 2.3.1.1 HUMAN RESOURCES REQUIRED

Zimbabwe open university is a renowned institute of higher learning with a vast amount resources and the above human resource requirements for the commencement of the project is just a drop in an ocean. This means that the project has passed its first test for feasibility however further test need to be done for other aspects as well that’s technological feasibility.

## 2.4.1 TECHNOLOGICAL FEASIBILITY.

This is mostly concerned with the hardware and software requirements for the development and implementation of this new system. This part of the study seeks to unveil blinders on the following concerns:

* What type of equipment and technology is needed for UNIVERSITY to run this system?
* Who are the potential suppliers of this equipment? Where are they located? What sort of service and warranties do they provide? How long will it take to acquire the equipment and begin operations?
* What type of structural facility is needed if there are any special storage requirements?

|  |  |  |  |
| --- | --- | --- | --- |
| HARDWARE | RECOMMENDED SPECIFICATIONS | QUANTITY | SUPPLIER DETAILS |
| Application server | 4Gig RAM,320HDD SATA,3GHz | 1 | Micro technologies  Harare  0715660503 |
| DELL desktop machines | 2Gig RAM, 1.8Gz, 160HDD SATA-RAID, DVD/RW | |  |  | | --- | --- | | 15 |  | | Global Computers  Harare  0735025241 |
| |  |  | | --- | --- | | Database server |  | | 4Gig RAM, 2 x 320HDD, 3GHz+ Dual Core Processor | 1 | Micro technologies  Harare  0715660503 |
| |  | | --- | | Backup sever | | 4Gig RAM, dual core processor 3,2 GHz speed, 2x320HDD | 1 | Micro technologies  Harare  0715660503 |
| |  |  | | --- | --- | | Printers |  | | HP Laser Jet | 5 | Micro technologies  Harare  0715660503 |
| |  | | --- | | UPS | | APC 180Watts 300V 230V (I/O) with standby time 6Hrs | 1 | Micro technologies  Harare  0715660503 |
| Network switch | 24port CISCO Switch | 1 | Micro technologies  Harare  0715660503 |
| Ethernet Cable | 100 meter reel | 1 | Micro technologies  Harare  0715660503 |
| RJ45s AND CRIMBLING TOOLS | Made in German | 3 pockets | Micro technologies  Harare  0715660503 |

## Table 2.4.1.1 HARDWARE REQUIREMENTS

The above is the list of hardware, specifications and quantity required for the new system, fortunately ZIMBABWE OPEN UNIVERSITY has got other machines which it already has so there may be only need for the new employees otherwise other employees will use their old machines thus cutting the budget a great deal. More over the suppliers of these devices although they are different coordination is not a problem because of ZOU’s high level telecommunications, so there is

no hindrance on the aspect of suppliers. This means that the system has passed the second test. The next assessment is on the software requirements.

|  |  |  |  |
| --- | --- | --- | --- |
| Software system | SOFTWARE | QAUNTITY | SUPPLIER DETAILS |
| Windows 8 | 15 users | NAME:Refresh computers  Location:Harare  Contacts:0715668503 |
| Apache server | 3 users | NAME:Refresh computers  Location:Harare  Contacts:0715668503 |
| Oracle virtual box | 1 | NAME:cherryl computers  Location:Harare  Contacts:0715668503 |
| Application development software | PHP version 7.3.11 | 2 | DOWNLOAD ON INTERNET |
| Sublime | 2 | DOWNLOAD ON INTERNET |
| photoshop | 2 | DOWNLOAD ON INTERNET |
| Xampp server | 2 | DOWNLOAD ON INTERNET |
| Java NET beans 8.0.2 | 2 | DOWNLOAD ON INTERNET |
| Microsoft office 2013 | 15 users | NAME:cherryl computers  Location:Harare  Contacts:0715668503 |
| Antivirus software | ESET Nodpoint 6 | 15 users | NAME:Refresh computers  Location:Harare  Contacts:0715668503 |

## Table 2.4.1.2 SOFTWARE REQUIREMENTS

Table 2.45.2 contains the software requirements for the new system including the quantities and suppliers. Just like the hardware this list can easily be financed from the company cash coffers, therefore the system software requirements cannot bar the adoption of the new system.

Therefore the conclusion is that the new system is technically feasible and management is willing and able to acquire the above specified requirements, developing, support and maintenance staff can be easily sourced, system scope is moderate.

## 2.5.1 ECONOMIC FEASIBILITY

According to Castro et al (2002) he propagates economic feasibility as, “a method to find out whether the benefits expected from the new system outweighs the incurred costs of carrying out the project”. For economically feasible systems the system should overall result in an excess of benefits over costs to pass the economic feasibility study hence otherwise a cash-flow sensitive organization may discredit it due to its demand for more money resources but may be forced to conduct further study based on the value it may bring the

The ZOU FLEET MANAGER system is a value adding system which will add material value to the business by enhancing the current systems and reduce risks which will potentially cause losses which can be converted into great money values.

The economic feasibility therefore is favorable when the net benefit is positive and unfavorable when the net benefit is negative. This part of feasibility study gives the top

Management the economic justification for the new system. This is an important input to management because very often the top management does not like to get confounded by the various technicalities that bound to be associated with a project of this kind. A simple economic analysis that gives the actual comparison of costs and benefits is much more meaningful in such cases. The economic face of the proposed project was to the effect that it could be pursued with the net benefit being extremely positive. However, many of the auxiliary benefits of the system could not be quantified but efforts were made to make some estimation for the purpose of costing.

## 2.5.1 COSTS

Costs are expenses, outlays or losses arising from developing and using a system. For software projects these costs are classified into two classes that are the development costs and implementation costs.

Development costs are meant to cover the expenses incurred during the development of the system and these can include:

* Communication costs.
* Salaries, traveling and research costs for the development team.
* Stationery costs.

Implementation/Maintenance costs are meant to cover for the maintenance of the system. These costs will include the software licenses, which is a one off payment, and the annual license, which is paid quarterly every year for example anti-viruses. The maintenance costs will also include upgrades to the system and any other modifications to be done to the system. The maintenance costs also cover the support costs, which will be conducted using telephones, or site visits.

|  |  |  |
| --- | --- | --- |
| NARRATION | QUANTITY | AMOUNT |
| DEVELOPMENT COSTS |  | US$ |
| HP desktop machines | 5 | 2650.00 |
| Printer | 4 | 660.00 |
| System software | 1 | 230.00 |
| Switch | 1 | 309.00 |
| Eset antivirus | 1 | 60.00 |
| Hired consultants |  | 150.00 |
| Labour |  | 300 |
| UPS | 1 | 95.00 |
| Database server | 1 | 3,115.00 |
| Application server | 1 | 3,040.00 |
| Application software | 1 | 210.00 |
| TOTAL: |  | 10,819.00 |
|  |  |  |
| IMPLEMANTITIONS/ MAINTAINANCE  COSTS |  |  |
| HP Desktop machine | 10 | 5,300.00 |
| Computer room upgrades |  | 1,040.00 |
| Network cables |  | 60.00 |
| Training |  | 40.00 |
| Backup server |  | 920.00 |
| Labour |  | 40.00 |
| TOTAL: |  | 7,400.00 |
|  |  |  |
|  |  |  |

## TABLE 2.5.1.1: PROJECT COSTS

## 2.5.2.1 BENEFITS

Benefits are the advantages received from installing and using this system and in this case they are economic in nature because of one or more of the following:

* Cost-saving nature
* Cost-avoidance nature
* Improved-service-level nature
* Improved-information nature

Cost-saving property leads to reductions in administrative and operational costs. A reduction in the size of the clerical staff used in the support of an administrative activity in this case the management of vehicles is an example of a cost-saving benefit.

Cost-avoidance property eliminates future administrative and operational costs. No need to hire additional staff in future to handle an administrative activity is an example of a cost-avoidance benefit.

Improved-service-level properties are those where the performance of a system is improved by a new computer-based method. Querying vehicle information in a database in fifteen minutes rather than an hour is an example of this third type of benefit.

Improved-information benefits are where computer based methods can provide better information for decision-making. For example this new system can display complete and comprehensible information for each and every vehicle such that the management can make informed decision concerning disposal or acquisition of a vehicle.

There are two types of benefits that were identified that are the tangible benefits and the intangible benefits that the system is going to provide.

## 2.5.2.2 TANGIBLE BENEFITS

These are benefits that are quantifiable and can be expressed in figures and these include:

* Increased revenues due to reduction in corruption.
* The proposed system will result in less paperwork hence less stationary expenses.
* Increased data security through use of passwords.
* Reduction of record keeping and storage costs through use of the hard drive and large storage mediums, such as, CDs and DVDs.
* The use of the database will make searching for vehicle details faster and efficient than before

## 2.5.2.2 INTANGIBLE BENEFITS

These are the benefits believed to be difficult and impossible to quantify and these include

* Better service delivery to clients
* Improved employee morale
* Goodwill
* Improved organizational standards and professionalism.
* Effective decision making by management

After all cost and benefits were identified, cost-benefit-analysis was carried out to determine the cost effectiveness of the project.

|  |  |
| --- | --- |
| BENEFITS: | VALUES(US$) |
| Reduced stationery cost | 50.00 |
| Reduced telephone cost | 350.00 |
| Reduced security cost | 223.00 |
| Reduced overtime cost | 300.00 |
| Reduced labour cost | 2,800.00 |
| Expected increase in revenue | 25,819.00 |
| TOTAL BENEFITS | 29,542.00 |

## TABLE 2.5.2.3 PROJECT BENEFITS

## 2.5.3 COST BENEFIT ANALYIS

“Cost-benefit analysis is the exercise of evaluating a planned action or project by determining what net value it will have for the company”, <http://www.inc.com>.

Basically, a cost-benefit analysis finds, quantifies, and adds all the positive factors. These are the benefits. Then it identifies, quantifies, and subtracts all the negatives, the costs. The difference between the two indicates whether the planned action is advisable. The real key to doing a successful cost-benefit analysis is making sure to include all the costs and all the benefits and properly quantify them. It is the fundamental assessment behind virtually every business decision, due to the simple fact that business managers do not want to spend money unless the benefits that derive from the expenditure are expected to exceed the costs. As companies increasingly seek to cut costs and improve productivity, cost-benefit analysis has become a valuable tool for evaluating a wide range of business opportunities, such as major purchases, organizational changes, and expansions.

|  |  |  |
| --- | --- | --- |
| NARRATION | DR | CR |
| INCOME: |  |  |
| Total Benefits(table project benefits.) |  | 29,542.00 |
| Less cost: |  |  |
| Total costs(table :project costs) |  | (18,219.00) |
| EXCESS OF INCOME OVER EXPENDITURE: |  | 11,323.00 |

## TABLE 2.5.3.1 COST BENEFIT ANALYSIS

From the Cost-Benefit Analysis results obtained, it can be noted that benefits outweigh the costs hence we can consider proceeding with a project. In order to choose between projects we need to take timing of the costs and benefits into account as well as the benefits relative to the size of the investment. One common method for comparing projects on the basis of their cash flow forecasts is Return of Investment (ROI).

## 2.5.3.2 RETURN ON INVESTMENTS (R.O.I)

According to Alvarez (2007), “Return on Investment determines the lifetime profitability of different investments”. A positive ROI shows that a project is viable as there will be a return to the initial investment while a negative ROI shows that the project is not viable as there will be no return to the initial investment. Since the ROI is highly positive, this suggests that the project is viable.

A performance measure used to evaluate the efficiency of an investment or to compare the efficiency of a number of different investments. To calculate ROI, the benefit (return) of an investment is divided by the cost of the investment; the result is expressed as a percentage or a ratio.

ROI = ( Gain from investment – cost of investment)

cost of investment

Return on investment is a very popular metric because of its versatility and simplicity. That is, if an investment does not have a positive ROI, or if there are other opportunities with a higher ROI, then the investment should be not be undertaken.

However this flexibility has a downside, as ROI calculations can be easily manipulated to suit the user's purposes, and the result can be expressed in many different ways. When using this metric, one must make sure to understand what inputs are being used.

Using the figures above our ROI will be:

(29,542.00-18,219.00) \* 100 = 62.15%

18,219.00

The ROI is positive therefore the project is a good investment opportunity.

## 2.5.4 OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, [corporate culture](https://en.wikipedia.org/wiki/Corporate_culture) and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, reducibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are to be realized. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

This feasibility looks at how the system will affect the stake holders of the organization. The PIECES framework can help in identifying operational problems to be solved, and their urgency. PIECES is an acronym for:

* Performance: Does the operation of the new system provide adequate throughput and response time?
* Information: Does the operation of the new system provide end users and managers with timely, pertinent, accurate and usefully formatted information?
* Economy: Does the operation of the new system provide cost-effective information services to the business? Could there be a reduction in costs and/or an increase in benefits?
* Control: Does the operation of the new system offer effective controls to protect against fraud and to guarantee accuracy and security of information?
* Efficiency: Does the operation of the new system makes maximum use of available resources, including people, time, and flow of forms?
* Services: Does the operation of the new system provide reliable service? Is it flexible and expandable?

The operations will be affected positively by the introduction of the new system because the computerized system is faster and effective as compared to the current manual system. Generally business operations are made easy and this motivates employees hence they will be eager to learn the new system. It also involves the training of staff at the national traffic police to understand the structure and the operation of the system.

## 2.5.5 WORK PLAN

The key to a successful project is in the planning. Creating a project plan is the first thing one should do when undertaking any kind of project. Often project planning is ignored in favor of getting on with the work. However, many people fail to realize the value of a project plan in saving time, money and many problems. A project plan outlines in specific detail how a project will be conducted, who will work on which part, and when and in what order each part will be accomplished.

The system development life cycle (SDLC) will be used, and waterfall model has been adopted as the appropriate development methodology for the proposed system. The following is a list of activities found in the waterfall methodology:

Project proposal

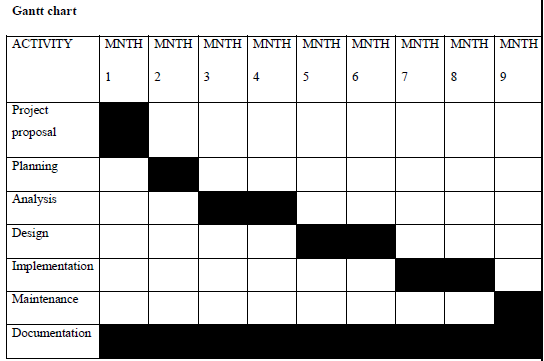
* Purpose
* To notify the top managers about the need of a new system, its objectives, the drawbacks of the current system and the tools required for the development of that new system.
* Principal Tools
* Good communication skills.
* Project planning
* Purpose
* Deciding whether the proposed system is needed, or whether the existing manual system needs modification. If “yes” then determine rough costs and benefits.
* Principal Tools
* Fact-gathering techniques such as interviewing the organization’s management and proposed visitors. Cost-benefit analysis that ascertains whether the benefits are worth the associated costs.
* Analysis
* Purpose
* Determining the purpose, target audience, and proposed content and functionality for the entire system. In short, deciding what is needed.
* Principal Tools
* Fact-gathering techniques.
* Cost-benefit analysis.
* Preliminary system architecture models.
* Design
* Purpose
* Designing the system (both organizationally and visually) that fulfils the requirements determined in the analysis phase.
* Principal Tools
* Site architecture models.
* MySQL database
* xamppServer 7.0
* Notepad++
* Image editors such as Adobe Photoshop or Fireworks for creating site graphics.
* Coding and Testing
* Purpose
* Building and testing the system. Construction and testing can’t be separated into independent phases, since testing should begin as soon as the first few lines of code are written. That way, problems are identified early in the process, rather than at the end when the entire system might end up needing major modifications to correct those problems.
* Principal Tools
* Notepad++
* Image editors such as Adobe Photoshop or Fireworks for creating site graphics.
* Implementation
* Purpose
* Uploading the system to the server, performing final site-wide tests.
* Principal Tools
* FTP or its equivalent.
* Maintenance
* Purpose
* Repairing, upgrading, and overhauling the system as necessary.
* Principal Tools

All of the tools mentioned in the prior phases.

The table and Gantt chart below shows the dates upon which each task is going to commence and end.

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Start | End | Duration |
| Proposal Project | 19/01/20 | 31/01/20 | 1mnth |
| Planning | 01/02/20 | 27/01/20 | 1mnth |
| Analysis | 02/03/20 | 30/04/20 | 2mnth |
| Design | 04/05/20 | 26/06/20 | 2mnth |
| Implementation | 02/07/20 | 28/08/20 | 2mnth |
| Maintenance | 02/09/20 | 29/09/20 | Ongoing |
| Documentation | 19/01/20 | 29/09/20 | 9mnth |

## 2.5.5.1 PHASES ACTIVITIES.



## 2.5.5. Gantt Chart.

## 2.8 CONCLUSION

The process of feasibility is for the idea of the new proposed system and after considering all the constraints, the feasibility shows that the benefits outweighed the costs of development. After the planning phase and the project being approved, deemed feasible and a work plan laid down the recommendation is to proceed with the development to the next chapter three for a detailed analysis of the current system.

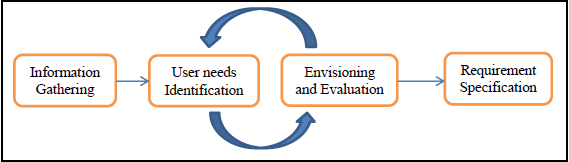
# Chapter 3

## INTRODUCTION

The purpose of this chapter is to obtain a thorough and detailed understanding of ZOU ‘s needs as defined in the first chapter and captured in the second chapter, and to break it down into discrete requirements, which are then clearly defined. During system requirements analysis process, the framework for the proposed and approved system is developed, providing the foundation for all future design and development efforts and also creating a detailed functional specification defining the full set of system capabilities to be implemented, along with accompanying data and process models illustrating the information to be managed and the processes to be supported by that system.

The functional specification will evolve throughout this phase of the SDLC as detailed business requirements are captured, and as supporting process and data models are created, ensuring that the eventual solution provides the users with the functionality they need to meet their stated business objectives.

“Analysis is a problem solving technique that decomposes a system into component pieces for the purpose of studying how well those component parts work and interact to accomplish their purpose”., - Management Information Systems, - Prof. J. Laudon.



## AGENERAL MODEL OF DESIGN PROCESS

The analysis phase of the software project life cycle produces the detailed requirements and system architecture specifications for a project. Most importantly, it establishes what the end user wants the system to do. During this phase, models of the system are to be developed. The author will use these models to ensure that he understands the system requirements during the design phase and that the project meets the client's expectations. The extensive information gathering and analysis of this phase provides extensive details on each of the aspects of the project. The final stage of the analysis phase is to organize this information into documents that will guide the work during the rest of the project. The three important deliverables created during this phase are:

* The business requirements report.
* The conceptual systems design plan.
* The strategy documents.

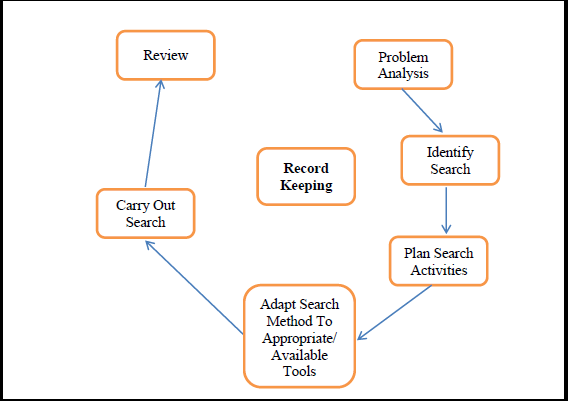
The business requirements report lays out the Zimbabwe Open University criteria of the project and includes any work associated with fleet management. The conceptual systems design plan is the transition from business requirements to the more detailed technical design. The plan contains high-level architecture diagrams like context diagrams and use case diagrams. The strategy documents describe the overall direction of the project in a number of areas, including alternative evaluation and implementation. These are direction-setting documents that can be worked on with the analyst and provide the overall direction to the more detailed planning that occurs later in the design phase.

## INFORMATION GATHERING METHODOLOGIES

Information gathering methodologies are a set of procedures used in making systematic observations or otherwise obtaining data, evidence, or information as part of a research project or study.

The Information system designed for an organization must meet the requirements of the end users of that organization. To obtain what an end user expects from the Information System the designer must gain complete knowledge of the organization’s working. It is important for the developer to know the information gathering techniques so that no information is overlooked and the nature and functions of an organization are clearly understood. The main purpose of gathering information is to determine the information requirements of an organization. Information requirements are often not stated precisely by management. It is the analyst’s responsibility to prepare some precise system requirements specifications (SRS), which is easily understood by users, since it is a vital document before starting a project.

The following procedure broadly defines an effective information gathering strategy:



## 3.2 INFORMATION GATHERING PROCESS

The following methodologies were used for data collection which was used for the design and development of ZOU Fleet Manager:

* Observations
* Documents review
* Interviews
* Focus group

## OBSERVATION

A qualitative method that provides descriptive information about what happens in a program event, including the environment or context, activities, processes, and discussions. Seeing and listening are key to observation. Observation provides the opportunity to document activities, behavior and physical aspects without having to depend upon people’s willingness and ability to respond to questions. It makes the analyst to get as close to the real system as possible.

It involves watching, following and recording activities as they are performed and then interpret those observations and draw conclusions. It works well when the system analysis is more interested in behavior than perception of the users. Observing user in the field is always the best way of determine their usability requirement.

Observations can be classified into two categories namely participant and non-participant observation. Participant observation has the following characteristics:

* Observer participates actively for an extended period of time
* May require that observer to live or work in that area
* Assumes that observer will become accepted member of the group or community

Non-participant observation has got the following features:

* Observer is an eavesdropper
* Someone who attempts to observe people without interacting with them without their knowledge that they are being observed

## ADVANTAGIES OF OBSERVATION METHODOLOGIES

* Allows viewing of what users actually do in context and may discover unnoticed processes.
* Collect data where and when an event or activity is occurring
* Does not rely on people’s willingness to provide information
* Evaluator observation contributes information from a different perspective than that of program participants and staff.

## DISADVANTAGIES OF OBSERVATION METHODOLOGIES

* Time consuming to perform.
* Susceptible to observer bias.
* Hawthorne effect – people usually perform better when they know they are being observed.
* Does not increase understanding of why people behave the way they do.
* With participant observation, a skilled facilitator is needed to help participants present a critical evaluation.

## DOCUMENTS REVIEW

Document review is a way of collecting data by reviewing existing documents. The documents may be internal to a program or organization. Documents may be hard copy or electronic and may include reports, program logs, performance ratings, funding proposals, meeting minutes, newsletters, and marketing materials. They provide valuable information about the extent to which current systems meet user needs and can identify potential usability problems to avoid in the new system. The document review process provides systematic procedure for identifying, analyzing, and deriving useful information from these existing documents.

## Advantages of documents review

* Effective means of identifying current problems, possible new features and acceptance criteria.
* Information contained in extanal document(s) is independently verifiable.
* The document review process can be done independently, without needing to solicit extensive input from other sources.
* Document review is typically less expensive than collecting the data on your own.
* Good source of background information
* Provides a behind-the-scenes look at a program that may not be directly observable (Unobtrusive)
* May bring up issues not noted by other means

## disadvantages of documents review

* May lead to including too many new functions or make system too similar to the old one.
* Information in the document(s) may represent a perspective that is not aligned with the needs assessment project.
* Obtaining and analyzing necessary documents can be a time consuming process.
* You are not able to control the quality of data being collected and must rely on the information provided in the document(s) to assess quality and usability of the source(s).
* Information may be inapplicable, disorganized, unavailable, or out of date
* Could be biased because of selective survival of information

## interviews

Frey and Oishi (1995:01) define it as "a purposeful conversation in which one person asks prepared questions (interviewer) and another answers them (respondent)" Interviews can have one of two basic structures. They can be either structured (closed interview style) or unstructured (open interview style). Open-ended or unstructured interviews are defined as an informal interview, not structured by a standard list of questions. Fieldworkers are free to deal with the topics of interest in any order and to phrase their questions as they think best. Closed or structured interviews are defined as a social survey where the range of possible answers to each question is known in advance. Often, possible answers are listed on the form so that the interviewer simply marks the appropriate reply in each case. This approach is much more standardized using a prearranged list of answers for the respondent to choose from. There is little freedom for flexibility, due to the fixed question order. Each person is given the same questions therefore being uniform.

Due to the explosive growth of new communication forms, such as computer mediated communication (for example e-mail and chat boxes), other interview techniques can be introduced and used within the field of qualitative research.

## Advantages of interviews

* With open-ended interviews, the interviewer may obtain rich details and new insights.
* Interviews allow the interviewer to ask the respondent for additional information.
* The respondent is able to raise the issues she or he feels are important and express ideas in her or his own words.
* They are useful to obtain detailed information about personal feelings, perceptions and opinions
* Ambiguities can be clarified and incomplete answers followed up

## disadvantages of interviews

* Open-ended interviews may require an outside evaluator for assistance with methods and analysis.
* They can be very time-consuming: setting up, interviewing, transcribing, analyzing, feedback, reporting
* They can be costly
* Different interviewers may understand and transcribe interviews in different ways.

## 3.5 focus groups

Open-ended interview with a group of similar respondents who engage in discussion about a specific topic under the direction of an interviewer. The interviewer is usually an outsider to the organization and may take a directive or unobtrusive role.

## 3.5.1 advantages of focus groups

* Like open-ended interviews, the participants are able to raise their own issues that they feel are important.
* Unlike one-on-one interviews, focus groups allow the interviewer to be less intrusive, and discussion can take its own direction – people are often more candid and spontaneous in a group discussion.
* Discussion among focus group participants can generate new information and raise new issues providing a range of responses with useful information.

## 3.5.2 disadvantages of focus groups

* Focus groups should be run by a skilled facilitator and often by an outsider to the program.
* Since focus groups work best with 4 to 10 people, organizing groups and motivating people to attend may be difficult.
* The group dynamic may silence some, especially dissenting opinions.
* May be expensive and time-consuming to conduct and transcribe, and require either a trained note taker or transcription of tapes.
* Usually not quantitative and usually not generalizable samples.

## Analysis of existing system

ZIMBABWE OPEN UNIVERSITY is currently using a log book system to manage and keep record of its fleet. Each and every vehicle owns its copy of the book in which every driver enters his or her name, journey details like destination, initial mileage, final mileage and the amount of fuel drawn including all services done to that vehicle. Below is a copy of a log book page.



## 0 log book.

At the end of the season that book is compiled by the transport department to analyze the use and possible condition of that vehicle and again that log book can be used also to launch investigations concerning probable vehicle misuse such as personal use of vehicles and precious resources like fuel. Experience has proved that most drivers may use company vehicles for personal gain and business like public transportation and stealing fuel from the company pumps in the name of a vehicle only to later on divert the resource.

However the success of the system sorely depends on the truthfulness and integrity the drivers to take care of the log book. Mostly the log books of several vehicles went missing and others entered wrong or incomplete information thus giving the management a hard time. Missing of a log book means the loss of several years’ information which cannot be recovered anywhere and hence loss of life time data. It has also been proven that the system has several loop holes for fraud for example a diver may misplace the log book intentionally in order to cover-up for his allegations thus making any investigations impossible.

## 3.6.1 process analysis

Process analysis is step-by-step breakdown of the phases of a process, used to convey the inputs, outputs, and operations that take place during each phase. A process analysis can be used to improve understanding of how the process operates, and to determine potential targets for process improvement through removing waste and increasing efficiency.

Process analysis is an approach that helps managers to improve the performance of their business activities. It can be a milestone in continuous improvement. The process boundaries are defined by the entry and exit points of inputs and outputs of the process. The table below shows the input, output and processing activities of the old system:

|  |  |  |
| --- | --- | --- |
| INPUTS | PROCESSES | OUTPUTS |
| * Journey details e.g. purpose, destination and distance.   Fuel drawn i.e. amount and place where it is drawn.  Service done to the vehicle including cost.   * Names and signature. | * Entering data * Analyzing data for investigation purposes * Disposal of a vehicle * Acquiring a new vehicle | * Reports * Archives and files * Total expenses |

## 3.6.1 table: process, inputs and outputs activities of the existing system.

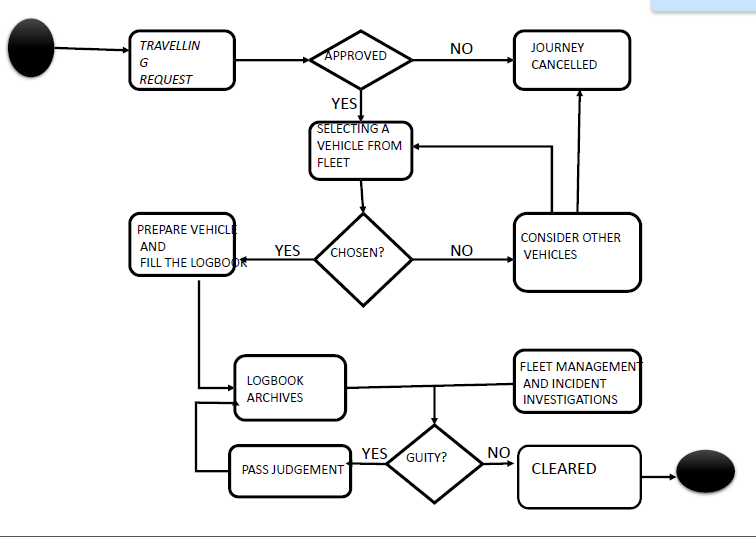
KEY

STATE

START/STOP

FLOW

DECISION



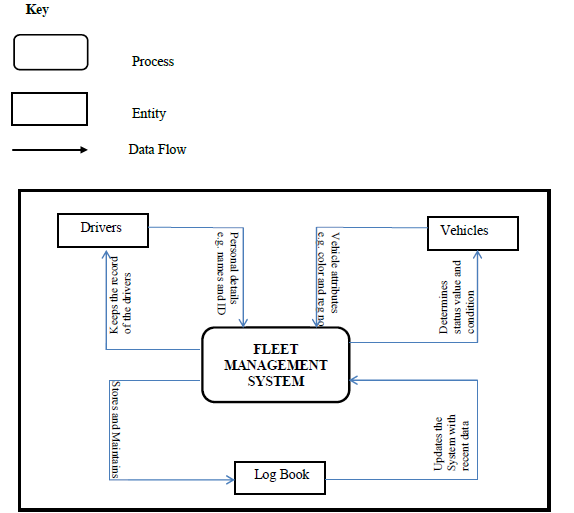
## 3.6.2 activity diagram of the old system

## 3.7 Data analysis

“Data analysis is a body of methods that help to describe facts, detect patterns, develop explanations, and test hypotheses.”- Macintosh HD June 10, 1996.

Analysis of data is used to inspecting, cleaning, transforming, and modelling data with the goal of highlighting useful information, suggesting conclusions, and supporting decision making. Two diagrams are used in this section namely context diagram and data flow diagram.

## 3.7.1 context diagram

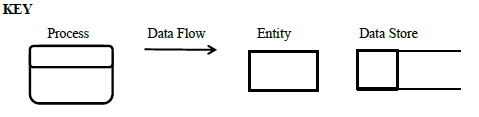


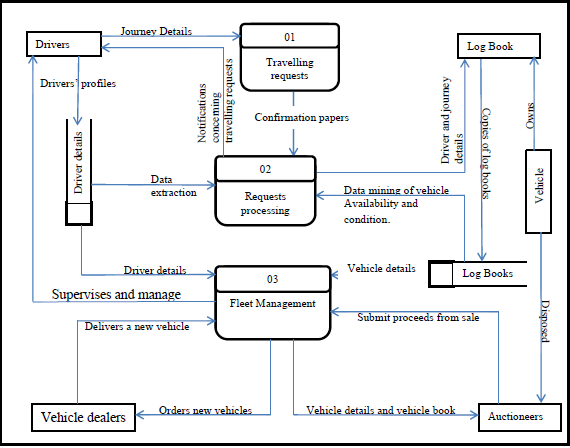
## 3.7.1.0 context diagram of the old system

The Context Diagram shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities (log book, drivers and vehicles).

## 3.7.2 data flow diagram of the old system

A data flow diagram (DFD) is a tool for functional analysis that shows the general flow of information through a system or program. DFDs reveal relationships among and between the various components in a program or system. DFDs are an important technique for modeling a system’s high-level detail by showing how input data is transformed to output results through a sequence of functional transformations. DFDs consist of four major components: entities, processes, data stores, and data flows.





## 3.7.2 data flow diagram of the old system

## **3.7.3 weaknesses of the current system**

The current fleet management system has the following drawbacks

* The current system is time consuming since it is manual in all its aspects.
* Evaluations and analysis of the whole fleet is difficult and almost impossible because there is no central data repository.
* Probability of losing and missing important vehicle document is high due to poor security of the system archives where the data documents are stored.
* High stationery cost.
* The system promotes centralization of data to local stations.
* Facilitates corruptive practices due to the intensity of loopholes.

## **3.8 evaluation of alternatives**

There are options which could also have been considered as an alternative to the development of this system, such alternatives are to be revised now in this section:

* Outsourcing
* Improvement
* Development

## **3.8.1 outsourcing**

Outsourcing is the act of one company contracting with another company to provide services that might otherwise be performed by in-house employees. Often the tasks that are outsourced could be performed by the company itself, but in many cases there are financial advantages that come from outsourcing. Typically, the function being outsourced is considered non-core to the business.

Drawbacks on outsourcing

* Outsourcing may increase the risk of information leakage, reduce confidentiality, as well as introduce additional privacy and security concerns
* There is also the danger of not being able to control some aspects of the company, as outsourcing may lead to delayed communications and project implementation. Project implementation timelines may suffer as a result.
* Any sensitive information is more vulnerable, and a company may become very dependent upon it’s outsource providers, which could lead to problems should the outsource provider back out on their contract suddenly.
* Acute when the work is being done in a different country (offshored), since that involves language, cultural and time zone differences.
* Outsourcing IT reduces or completely eradicates direct communication between organization and clients. Limited communication impedes the relationship building process, which may lead to the overall dissatisfaction of the organization and client.
* Organizations that outsource IT services run a risk of receiving poor quality work.
* High charging rates.

## **3.8.2 improving the current system**

Act of enhancing or making better in terms of quality, value or usefulness. This can be by making ideas, objects or processes more desirable by adding or removing components. The term can be also be applied to people as well, via methods such as performance reviews which are meant to try and improve an employee in some manner.

Drawbacks of improving the current system

* Possibility that the upgrade will worsen the product. Upgrades can also worsen a product subjectively.
* Upgrades of hardware involve a risk that new hardware will not be compatible with other pieces of hardware in a system.
* The main disadvantages are that you might get some new bugs, some of which could even stop you from using older data files.
* More expensive than developing a new system.
* Weaknesses of the old system will resurface.

## **3.8 .3 Development**

This is the development of a system from a company’s internal resources and pool of IT resources. This may involve hiring of experience technical team and pooling of funds to finance the development of the system.

Merits of development

* Legal costs are reduced significantly like licenses and training costs incurred if the system outsourced.
* Greater and better functionality since the analyst is more acquainted with the business needs and environment.
* Records are kept in a database that is easier to access and safe for storage.
* Implementation will be easier since the personnel responsible for the system development will be readily available.
* It minimizes changes in the business procedures and policies.
* User satisfaction is enhanced.
* Quality Control
* When development takes place in-house, the hands-on coding and testing process translates into the internal team’s system familiarity and results in lower long-term maintenance expenses and more professional application support.
* In-house Design Gives you Control
* Increased Flexibility

## **3.9 Requirements analysis**

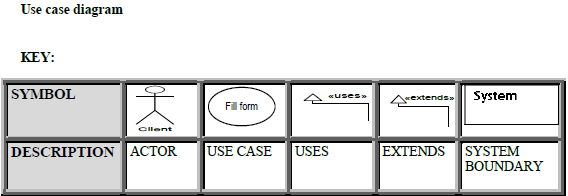
“Software engineering task bridging the gap between system requirements engineering and software design. Provides software designer with a model of system information, function and behavior” (Bruce R. Maxim).

The purpose of Requirements Analysis is to obtain a thorough and detailed understanding of the business need and to break it down into discrete requirements, which are then clearly defined and reviewed. Requirement analysis can be categorized into two phases namely:

* Functional requirements
* Non-functional requirements

## 3.9.1 functional requirements (USE CASE DIAGRAM)

Requirements that define those features of the system that will specifically satisfy a user’s need, or with which the users will directly interact with.





## **3.9.1 USE CASE DIAGRAM**

Use cases are used during the analysis phase of a project to identify and partition system functionality. They separate the system into actors, use cases and associations. Associations between actors and use cases are indicated in use case diagrams by solid lines.

An association exists whenever an actor is involved with an interaction described by a use case. Associations are modeled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case. Actors represent roles that are played by users of the system. Those users can be humans, other computers, pieces of hardware, or even other software systems. Use cases describe the behavior of the system when one of these actors sends one particular stimulus. This behavior is described textually. It describes the nature of the stimulus that triggers the use case; the inputs from and outputs to other actors, and the behaviors that convert the inputs to the outputs.

## **3.9.2 Nonfunctional requirements**

The plan for implementing non-functional requirements is detailed in the architecture. In general, functional requirements define what a system is supposed to do whereas non-functional requirements define how a system is supposed to be. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements".

Non-functional requirements can be divided into two main categories:

1. Execution qualities, such as security and usability, which are observable at run time.

2. Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system.

Security

* Login requirements - access levels.
* Password requirements - length, special characters, expiry, recycling policies
* Inactivity timeouts – durations, actions

Audit:

* Audited elements – what business elements will be audited?
* Audited fields – which data fields will be audited?
* Audit files characteristics - before image, after image, user and time stamp.

Performance:

* Response times - application loading, screen open and refresh times.
* Processing times – functions, calculations, imports, exports.
* Query and Reporting times – initial loads and subsequent loads.

Capacity:

* Throughput – how many transactions per hour does the system need to be able to handle?
* Storage – how much data does the system need to be able to store?
* Year-on-year growth requirements

Availability:

* Hours of operation – when is it available? Considering weekends, holidays, and maintenance times.
* Locations of operation – where should it be available from, what are the connection requirements?

Reliability:

* Mean Time Between Failures – What is the acceptable threshold for down-time? e.g. one a year, 4,000 hours
* Mean Time To Recovery – if broken, how much time is available to get the system back up again?

Integrity:

* Fault trapping (I/O) – how to handle electronic interface failures.
* Bad data trapping - data imports, flag-and-continue or stop the import policies.
* Data integrity – referential integrity in database tables and interfaces
* Image compression and decompression standards

Recovery:

* Recovery process – how do recoveries work, what is the process?
* Recovery time scales – how quickly should a recovery take to perform?
* Backup frequencies – how often is the transaction data, set-up data, and system (code) backed-up?
* Backup generations - what are the requirements for restoring to previous instance(s)?

Compatibility: the system is supposed to be:

* Compatible with shared applications – What other systems does it need to talk to?
* Compatible with 3rd party applications – What other systems does it have to live with amicably?
* Compatible on different operating systems – What does it have to be able to run on?
* Compatible on different platforms – What are the hardware platforms it needs to work on?

Maintainability: the system is supposed to:

* Conform to architecture standards – What are the standards it needs to conform to or have exclusions from?
* Conform to design standards – What design standards must be adhered to or exclusions created?
* Conform to coding standards – What coding standards must be adhered to or exclusions created?

Usability: the system is supposed to:

* Look and feel standard - screen element density, layout and flow, colors, UI metaphors, keyboard shortcuts.
* Internationalized / localized requirements – languages, spellings, keyboards, paper sizes.

Documentation:

* Required documentation items and audiences for each item.

## **3.10 Conclusion**

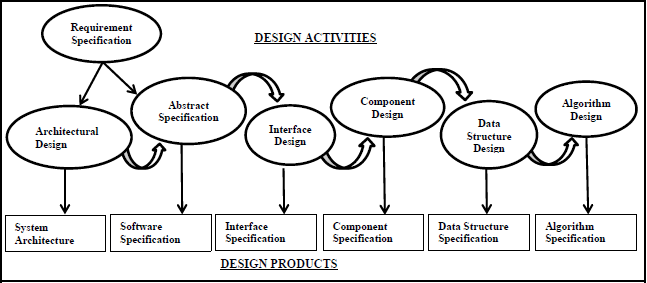
This marks the end of the analysis phase of the old ZOU fleet management system and the beginning of the designing of the new system (ZOU Fleet Manager). Vital data has been collected using various data collecting methodologies and that data played a pivotal role in the analysis of the old system using various diagrammatic and logical methods. As a result, the loopholes of the old system were exposed at the same time unveiling the inevitable need for the new system. The evaluation of alternatives has also revealed the need of the in-house development of the new system against the popular and attractive options of outsourcing and upgrading the existing system (improvement) hence endorsing the proposed plan. The next stage is the design phase whereby the conceptual appearance of the new system is going to be exhibited.

# **Chapter 4**

## **4.0 Introduction**

The starting point is the requirements document delivered by the analysis phase and mapping those requirements into architecture. The requirements identified in the Analysis Phase will be transformed into a system design document that accurately describes the design of the ZOU Fleet Manager system and that will be used as an input to system development in the next phase. That architecture defines the components, their interfaces and behaviors. Therefore, this chapter describes a plan to implement the requirements, it represents the ``how'' phase. Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are to be established.

The system designer is not going to arrive at the finished design immediately. Design development is iterative through a number of different versions. The starting point is informal design which is refined by adding information to make it consistent and complete as shown in the figure below:



## **Table: 4.0 general model of design process**

The table below illustrates some of the major activities which are going to be met during this stage:

|  |  |
| --- | --- |
| ***DESIGN PHASE ATTRIBUTES*** | ***DESCRIPTION*** |
| Major function:  Output:  Principal tools:  Personnel and Tasks: | To design a new ZOU Fleet Management system that will fulfill the requirements of the users and mitigate the shortfalls of the older version of the system. |
| The design specification document. |
| Data dictionary, data flow diagrams, ER or Tables and EER diagram, process specifications, data models, system models, prototyping, system flowcharts, Package and class, collaboration or sequence diagram. |
| * The analyst evaluates and orders all necessary hardware and software * The analyst transforms the functional diagrams of the analysis phase into the hierarchical diagrams of the design phase * The analyst designs the user interface, including input and output formats. * The analyst incorporates system security measures into the design. * The analyst determines the staffing requirements and designs procedures and workflow. * The analyst designs the required database. * Users, managers, and supervisor review the design |

## **Table 4.0.1: General Summary of the Design phase.**

## **4.1 System design**

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development.

Kenneth C Laudon: Management Information Systems, “System design refers to detailing how a system will meet the information requirements as determined by the system analysis. The design of an information system is the overall plan or model for that system. Each design represents a unique blend of technical and organizational components.”

The table below illustrates major design specifications and their elements as described in the previous chapter.

|  |  |  |  |
| --- | --- | --- | --- |
| ***SPECIFICATIONS*** | |  |  | | --- | --- | |  | ***ELEMENTS*** | |
| OUTPUTS:  INPUTS:  USER INTERFACE:  DATABASE DESIGN:  PROCESSING:  CONTROLS:  SECURITY:  TRAINING:  CONVERSION: | * Medium * Content * Timing |
| * Origins * Flow * Data entry |
| * Simplicity * Efficiency * Logic * Feedback * Errors |
| * Logical data model * Volume and speed requirements * File organization and design * Record specification |
| * Computations * Program modules * Required reports * Timing of outputs |
| * Input controls(character, limit, reasonableness) * Processing controls(consistency, record counts) * Output controls(totals, samples of output) * Procedural controls(passwords, special forms) |
| * Access controls * Catastrophe plans * Audit trails |
| * Select training techniques * Develop training modules * Identify training facilities |
| * Transfer files * Initiate new procedures * Select testing method * Cut over to new system |

## **Table 4.1.1 System design elements and specifications**

## **4.2 Description of the proposed system**

This section outlines the purpose of the Fleet Manager system as well as how it will be used by its various users in Zimbabwe open university. It offers in-depth descriptions of the services available to each user.

ZOU Fleet Manager will have the following functionality illustrated in the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***FUNCTION*** | ***DESCRIPTION*** | ***INPUTS*** | ***PROCESSING*** | ***OUTPUT*** |
| FLEET RECORDS STORAGE: | It maintains a database containing all the records of existing vehicles for example date of purchase, current millage, accident records, and depreciation. Management of vehicles includes creation, modification and deletion of the vehicle records within the system. The  Fleet Manger will add each vehicle in the fleet into the system using the application’s web-based interface. Once a vehicle record has been added into the system, it is automatically assigned a unique identifier. | Initial and final mileage, cost of purchase and rate of depreciation. | Calculation of distance travelled, calculation of net book value and saving of entered data into the database. | Displaying of reports clearly exhibiting useful information for the effective monitoring of the fleet |
| DRIVER PROFILE MAINTAINANCE: | The system is supposed to keep an up to date repository of all drivers’ details like ID number, names, discipline records and next of kin. The Fleet Manger can add, modify and delete vehicle  drivers from the system. | Drivers’ personal information, incident reports e.g. accidents. | Saving and protection of the data in the database through the use of access levels and user authentication. | Report displaying correct and clear drivers’ personal information and record of behavior. |
| FLEET DISPATCHER | Facilitates the disposal of all appropriate vehicles which have reached the maximum mileage permissible. | Records of the vehicle mileage from the database. | Checking and comparing the current mileage with the maximum mileage set. | Prompt report to dispose the outdated vehicle. |
| VIEW STATISTICS | A report based on vehicle statistics will include: a list of all the routes driven in the vehicle average speed; vehicle driver; and estimated mileage. | Trips made and destinations, driver and distance travelled. | Compilation of data from different tables in the database and presenting it in one view. | Report of individual vehicle statistic. |
| VIEW VEHICLES | Through the application the system user is presented with a list of all active vehicles (vehicles currently being deemed as running). | Vehicle profile and image. | Linking a vehicle profile with its appropriate image. | Reports displaying profiles of the vehicles |
| ACCIDENT  ADMINISTRATION | This module records the details of the accident and manages the accident claims from and to the third party. | Vehicle and driver involved an accident and the police report. | Saving the incident information and referencing it to the correct vehicle and driver involved. | Reports showing all accident data when queried. |

## **Table 4.2.1: System Design Specification and Elements.**

## **4.3 CONTEXT DIAGRAM AND DFD OF THE PROPOSED SYSTEM**

**Context Diagram:**

Jeffrey A. Hoffer: Modern System Analysis and Design, “a context diagram is an overview of an organizational system that shows the system boundaries, external entities that interact with the system and major information flows between the entities and the system.”

The context diagram is going to establish initial ZOU Fleet Manager’s scope and its big picture under scrutiny. A project’s scope defines what aspect of the business a system or application is supposed to support and how the system being modeled must interact with other systems and a business as a whole. In the information systems framework, scope is defined as the communication focus from the system owner’s perspective. It is documented with a context diagram.

As shown in the context diagram, the main purpose of the system is to process fleet information in response to management and drivers’ demands.

MANAGEMENT

DRIVER

Reports for the fleet, drivers and incidents.

Incident reports

Data mining

Hearing results

Personal profile

Trip information

***ZOU FLEET MANAGER***

01

Driver details, fleet data, accident information

Queries response

Vehicle acquisition

Vehicle calculate depreciation

DATABASE

FLEET

Data queries

Vehicle disposal

## **Table 4.3.1: Context Diagram of the proposed system.**

## **4.3.2 Data Flow Diagram:**

The following data flow diagram (DFD) reveals relationships among and between the various components and functions of ZOU Fleet Manager. DFDs are an important technique for modeling a system’s high-level detail by showing how input data is transformed to output results through a sequence of functional transformations.

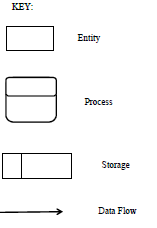
When it comes to conveying how information data flows through systems (and how that data is transformed in the process), data flow diagrams (DFDs) are the method of choice over technical descriptions for three principal reasons.

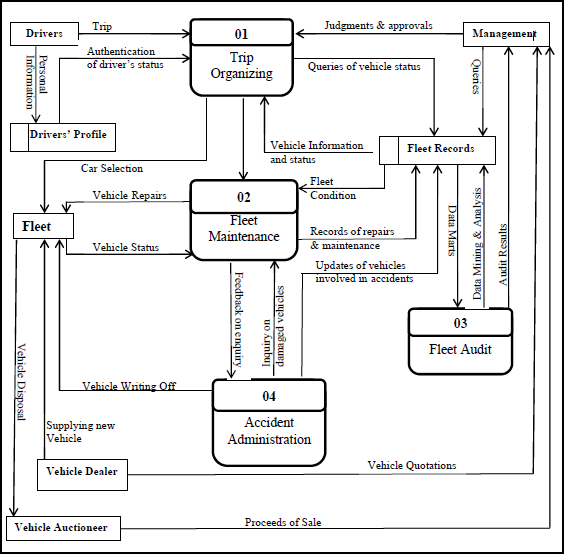
* DFDs are easier to understand by technical and nontechnical audiences.
* DFDs can provide a high level system overview, complete with boundaries and connections to other systems.
* DFDs can provide a detailed representation of system components.

DFDs represent the following:

* External devices sending and receiving data.
* Processes that change that data.
* Data flows themselves.
* Data storage locations.

The following is a data flow diagram of the ZOU Fleet Manager system.





## **Table 4.3.2: Data Flow Diagram of the proposed system.**

## **4.4 ARCHITECTURAL DESIGN**

Joel Henry: Software Project Management, “System architecture design is how the software product is portioned and how the partitions work together to provide functionality.”

Leszek A Maciaszek: Requirements Analysis and System Design, “Architecture design is the description of the system in terms of its modules (components).”

The architectural design involves the layered organization of classes and packages, the assignment of processes to computing facilities, reuse, and component management.

Architectural design resolves the issues with regard to a multi-tier physical architecture as well as with regard to multi-layer logic architecture.

Design comparisons and decisions are hotbeds for ego bumping on the scale of sumo wrestling. Each software engineer knows the best software design, even though none of their designs are the same. Decisions concerning architecture designs will be based upon design qualities such as:

Maintainability: The ease or difficulty of maintaining a network, including correcting defects, improving performance, or adding functionality.

Extendibility: The ease or difficulty of adding function of adding functionality to the system including its network architecture.

Performance: The ability to execute functions fast enough to meet performance goals. (Response time is important to an application.)

Security: The authorization of access to data in a network, which is controlled by the network administrator.

Safety: Isolation of safety-critical components e.g. servers and access points.

Availability: Include redundant components in the architecture so as to make accessibility easier.

Portability: Is a characteristic attributed to a computer program if it can be used in operating systems other than the one in which it was created without requiring major rework.

Testability: the ease or difficulty of testing a software product in order to uncover defects and support corrective activities

These network qualities are influenced by factors such as: Coupling: The degree of connections between classes. It measures the class interdependence. The weaker the coupling the bet

Cohesion: The degree of inner self-determination of the class. It measures the strength of the class independence. A highly cohesive class performs one action or achieves a single goal. The stronger the cohesion the better.

Modularity: The characteristic of a network that is divided into a set of units that support ease of network change, promote system understanding, and reduce complexity.

Information hiding (Encapsulation): the characteristic of a network in which details of a specific implementation are not visible outside a portion or segment of that network. This is achieved through the creation of virtual networks popularly known as V-LANS.

Insulation: the characteristic of design and code that allows changes within one portion of the design or a code unit to have no effect on other parts

The architectural design includes decisions about the solution strategy for the client and server aspects of the system. The architectural design is also concerned with the selection of a solution strategy and with modularization of the system. The solution strategy needs to resolve client (user interface) and server (database) issues as well as any middleware needed to “glue” client and server processes.

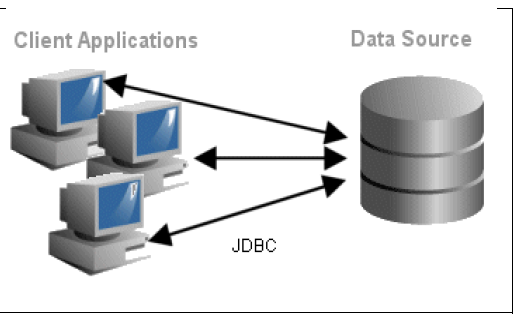
Network Architecture Design

Due to several factors considered the designer is going to use the two-tier network architecture to implement the ZOU Fleet Manager network.

The end goal is to build a network for convergence––one that has the extensibility to support a variety of new applications in a highly available and secure fashion. While the two-tier architecture represents a simplification of previous designs, it does not represent a compromise in terms of quality, availability, security or management. As the name indicates, the two-tier architecture is a collapsing of layers into intelligent core and unified access

## **4.4.1 Two-Tier Architecture:**

The two-tier architecture is like a client server application. The direct communication takes place between client and server. There is no intermediate between client and server.



## **Table 4.4.1: Two-Tier Architecture Diagram of the proposed system.**

## **4.4.2 Limitations of a Two-Tier Approach**

The two-tier network architecture may not be appropriate for every enterprise. Some enterprises may be limited by physical or organizational constraints. A two-tier network design is ideally suited for enterprises with greenfield environments or with sufficient space and modularity within the building to handle the consolidation and changes necessary to migrate from a three-tier approach. Enterprises with long distances between core and access may find it more cost-effective to include additional network layers to ensure reach of traffic flows. An example of this may be a large manufacturing campus with distributed production locations. In other cases, organizations that occupy historical buildings with older cable plants or restrictions on physical infrastructure changes may also be limited in their ability to cleanly migrate to a two-tier design.

While a two-tier design should decrease the amount of space required for the physical housing of switches, some building layouts may not have the flexibility of providing space where it is required. As a general rule, enterprises with a physical separation between the data center or access layer and the core that is greater than 100 meters for copper runs and 300 meters for multi-mode fiber, could be faced with the relatively more expensive option of using higher powered optics to cover the greater distances. This additional cost should be weighed carefully against the cost of supporting additional network layers. Fundamentally, the largest issues prohibiting two-tier architecture are distance, age of cable plants and the flexibility of the building structure. Assuming none of these are of major concern, the enterprise is best served by collapsing layers and simplifying overall network design.

Performance will be reduced when there are more users. This design is used frequently in decision support systems where the transaction load is light. Two tier software architectures require minimal operator intervention. The two tier architecture works well in relatively homogeneous environments with processing rules (business rules) that do not change very often and when workgroup size is expected to be fewer than 100 users, such as in small businesses.

The most important limitation of the two-tier architecture is that it is not scalable, because each client requires its own database session. The two tier design will scale-up to service 100 users on a network. It appears that beyond this number of users, the performance capacity is exceeded. This is because the client and server exchange "keep alive" messages continuously, even when no work is being done, thereby saturating the network.

Implementing business logic in stored procedures can limit scalability because as more application logic is moved to the database management server, the need for processing power grows. Each client uses the server to execute some part of its application code, and this will ultimately reduce the number of users that can be accommodated.

The two tier architecture limits interoperability by using stored procedures to implement complex processing logic (such as managing distributed database integrity) because stored procedures are normally implemented using a commercial database management system's proprietary language. This means that to change or interoperate with more than one type of database management system, applications may need to be rewritten. Moreover, database management system's proprietary languages are generally not as capable as standard programming languages in that they do not provide a robust programming environment with testing and debugging, version control, and library management capabilities.

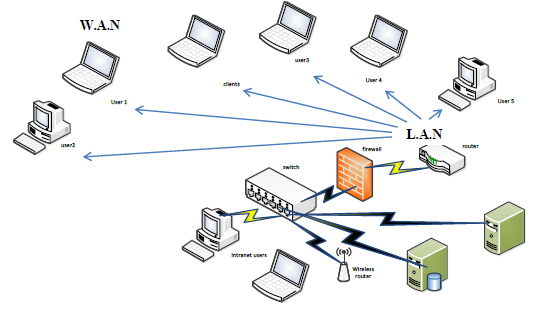
Two tier architectures can be difficult to administer and maintain because when applications reside on the client, every upgrade must be delivered, installed, and tested on each client. The typical lack of uniformity in the client configurations and lack of control over subsequent configuration changes increase administrative workload. The two tiered architecture is not effective running batch programs. The client is typically tied up until the batch job finishes, even if the job executes on the server; thus, the batch job and client users are negatively affected.

## **4.4.3 Benefits of a Two-Tier Approach**

Zimbabwe open university will benefit from a lower network acquisition cost by requiring fewer switches. Installation and maintenance costs are decreased due to the added simplicity of the two-tier design. A simplified network improves the IT organization’s ability to innovate at the application layer and deliver next generation IP applications such as IP Telephony with quality and consistency. Development Issues e.g. Simple structure, Easy to setup and maintain. Business logic and database are physically close, which provides higher performance.

## **4.5 PHYSICAL DESIGN**

Physical design is the hardware layout, where cables are routed, where routers, servers, and workstations are located within a building. Fortunately, Zimbabwe Open University has got an existing infrastructure through which it connects to the internet, so the new system is going to adopt such architecture to implement its network. The following image exhibits the layout of the architecture.



**L.A.N**

**W.A.N**

## ***Table 4.5: Physical diagram of the proposed system.***

## ***The*** WAN connection is facilitated by a dedicated point to point link of fiber optic since the distance to the remote stations may spun up to more than 500km, hence only the fiber optic link can carter for long distance transmission problems. Fiber optics has a large capacity to carry high speed signals over longer distances without repeaters than other types of cables. The information carrying capacity increases with frequency.

The LAN is comprised of the following devices:

* Router
* Firewall
* Switch
* Wireless router
* Application server
* Network cables (CAT 6)

Router

A router is a device that is going to forward data packets between the ZOU LAN and its remote stations, creating an overlay internetwork. A router is connected to two or more outside data lines as shown above and linking them with the LAN. When a data packet comes in on one of the lines, the router reads the address information in the packet to determine its ultimate destination. Then, using information in its routing table or routing policy, it directs the packet to the next network on its journey. Routers perform the "traffic directing" functions on the Internet. A data packet is typically forwarded from one router to another through the networks that constitute the internetwork until it gets to its destination node.

Firewall

A firewall can either be software-based or hardware-based and is used to help keep a network secure. Its primary objective is to control the incoming and outgoing network traffic by analyzing the data packets and determining whether it should be allowed through or not, based on a predetermined rule set. The firewall will build a bridge between the LAN that is assumed to be secure and trusted, and another network, usually an external (inter)network, such as the Internet, that is not assumed to be secure and trusted.

Switch

A network switch is a computer networking device that connects network segments or network devices. It commonly refers to a multi-port network bridge that processes and routes data at the data link layer (layer 2) of the OSI model. A switch in an Ethernet-based LAN reads incoming TCP/IP data packets or frames containing destination information as they pass into one or more input ports.

Wireless router

A wireless router is a device that performs the functions of a router but also includes the functions of a wireless access point and a network switch. It is commonly used to provide access to the Internet or a computer network. It does not require a wired link, as the connection is made wirelessly, via radio waves. This device will be used by users of laptops in the organization.

Application server

This is the hosting machine for ZOU Fleet Manager and its database. That is where all the fleet management transactions are going to be done and saved. An application will provide the system with services such as security, data services, transaction support, load balancing, and management of large distributed systems.

Network cables (CAT 6)

Category 6 cable (Cat 6), is a cable standard for Gigabit Ethernet and other network physical layers. Cat 6 features more stringent specifications for crosstalk and system noise. The cable standard provides performance of up to 250 MHz and is suitable for 10BASE-T, 100BASE-TX (Fast Ethernet), 1000BASE-T/1000BASE-TX (Gigabit Ethernet) and 10GBASE-T (10-Gigabit Ethernet).

## **4.5.1 DATABASE DESIGN**

Ryan K. Stephens: Understanding Database Fundamentals, “Database design is the process of converting business objects into tables and views. It is the process of actually fashioning artistically or skillfully a database basing on a business model.”

Business model components such as entities and attributes are converted into tables and columns. Constraints are added to columns where necessary in order to enforce data and referential integrity. Views of tables might be created in order to filter the data that a user sees, or to simplify the query process. After the design of a database is complete, the entire business model (business processes, rules, and entities) will have been converted into a functional database in which corporate data can be stored, modified, and easily retrieved.

A good database is determined as seen through the eyes of the customer, the end user, the database administration team, and management. There are many hallmarks of a good database, the most common of which are:

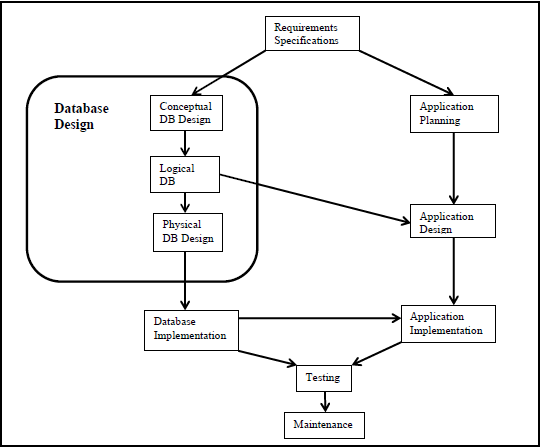
* Data storage needs having been met
* Data is readily available to the end-user
* Data being protected through database security
* Data being accurate and easy to manage
* Overall database performance being acceptable
* Having a minimized amount of redundant data stored

The following are milestones in form of stages which the author is going to pass through until the full implementation of the database.

## **a. Requirements Specifications:**

This has already been done in the previous chapter and in the preliminary stages of this chapter through the use of context diagrams, data flow diagrams of both the old and new systems and also the data gathering process. That data is going to be used to come up with the blue print of the desired database throughout the designing process. The major goal in requirements specification is to:

* Collect the data used by the organization,
* Identify relationships in the data,
* Identify future data needs,
* And determine how the data is used and generated.



## Table 4.5.1: Database design stages

## b. **Database Design:**

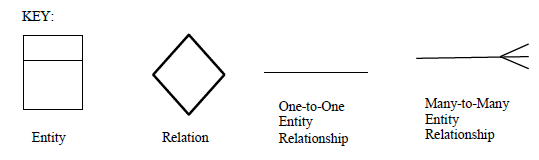
The requirements specification has provided with a high-level understanding of the organization, its data, and the processes that the author must model in the database. Database design involves constructing a suitable model of this information. Since the design process is

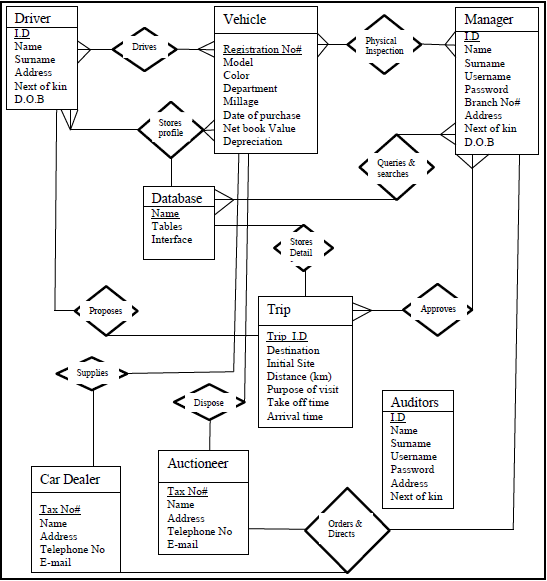
complicated, especially for large databases, database design is divided into three phases:

## **1. Conceptual database design:**

C.J Date: Introduction to Database Systems 7th edition, “Conceptual database design is a process of deciding what information is to be held in the database- in other words, to identify the entities of interest to the enterprise and to identify the information to be recorded about those entities.”

The conceptual database design uses data requirements and produces entity relationship diagrams (ERDs) and should represent all the data requirements and format. The following is the ERD of the new system.





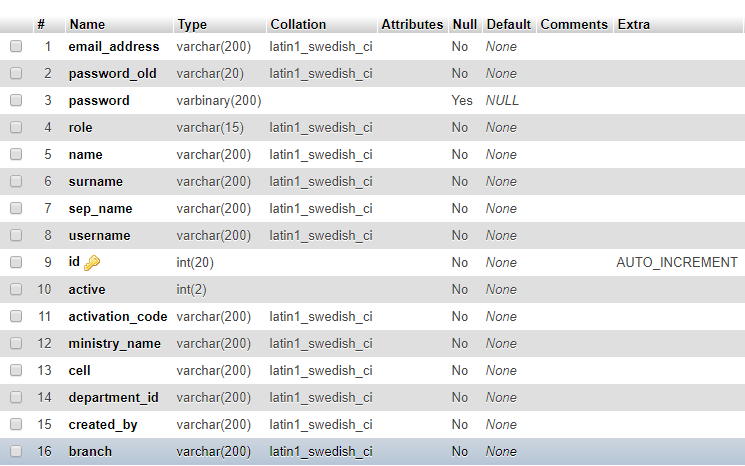
## Table 4.5.2: Entity Relationship Diagram of the new system

## **2. Logical database design**

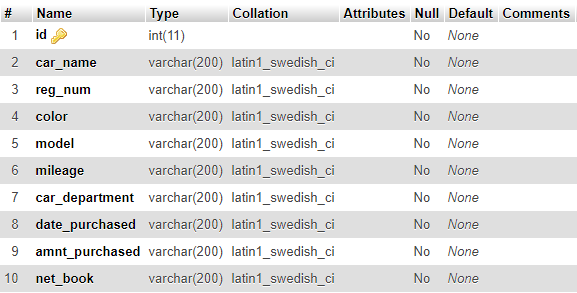
The logical database design transforms the conceptual data model into a format understandable by a DBMS. It is not concerned with efficient implementation but rather with refinements to the conceptual data model. The refinements preserve the information content of the conceptual data model while enabling implementation on DBMS. The logical database design consists of two refinement activities which are conversion and normalization. The conversion activity transforms ERDs into table designs using conversion rules. The normalization activity removes redundancies in a table using constrains or dependencies among columns.

## **Database tables**

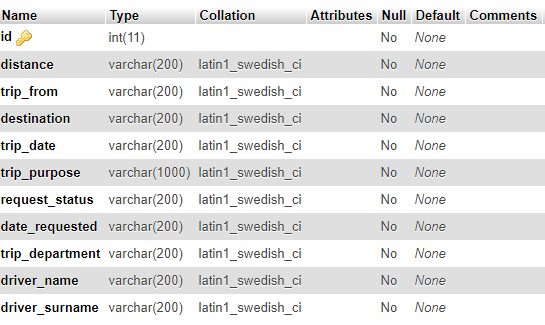
Ryan K. Stephens: Database Design 2001, “A table is the primary unit of physical storage for data in a database.” When a user accesses the database, a table is usually referenced for the desired data. Multiple tables might comprise a database; therefore a relationship might exist between tables. Because tables store data, a table requires physical storage on the host computer for the database. The following are tables which are going to be available in system database.



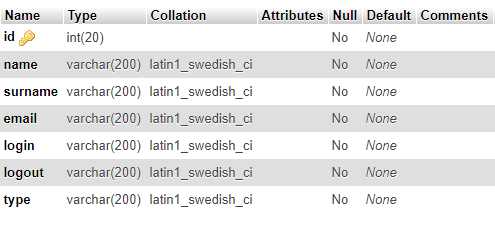
## ***Table 4.5.3: users fields.***



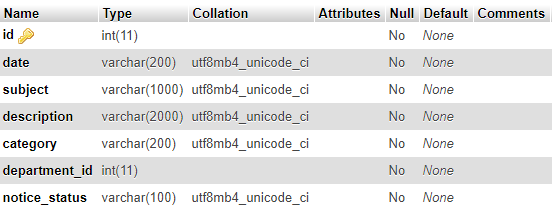
## ***Table 4.5.4: vehicles fields.***



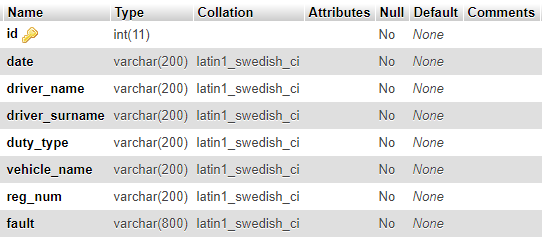
## ***Table 4.5.5: trips fields.***



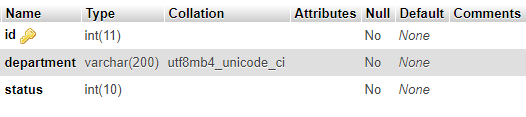
## ***Table 4.5.6: audit trails fields.***



## ***Table 4.5.7: web notices fields.***



## ***Table 4.5.8: vehicle fields.***



## **Table 4.5.9: departments fields.**

## **3. Physical database design**

The physical database design is concerned with efficient implementation and better performance of the database. An efficient implementation minimizes response time without using too many resources such as disk space and main memory. Two main activities in this stage are about indexes and data placement. An index improves performance on retrievals and data placement is concerned with how data is clustered or located close together on a disk.

The following is a three level architecture of the new system.

Application Layer

Conceptual Layer

Physical Layer

Internal View 1

Internal View 2

Internal View 3

Conceptual Schema

Internal Schema

Stored Database

Stored Database

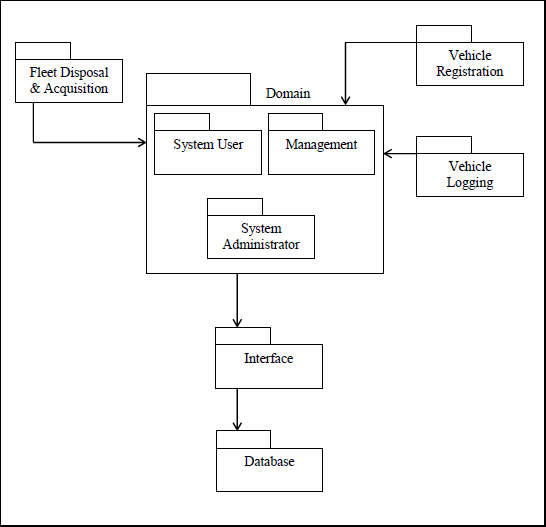
Stored Database

## **4.6 PROGRAM DESIGN**

Program design refers to the process of developing and structuring what goes into a project before it begins, during redesign, or at any other point when the system analyst have an opportunity to assess whether the project is achieving its objectives and whether its effectiveness can be improved. Design determines what results we want, based on an assessment of the needs and resources available, and what interventions are most likely to achieve them. Program design is most effectively achieved through the use of various models like package, class, and sequence diagrams and such tools are also going to be implemented the designing of this system’s program.

## **4.6.1 Package Diagram**

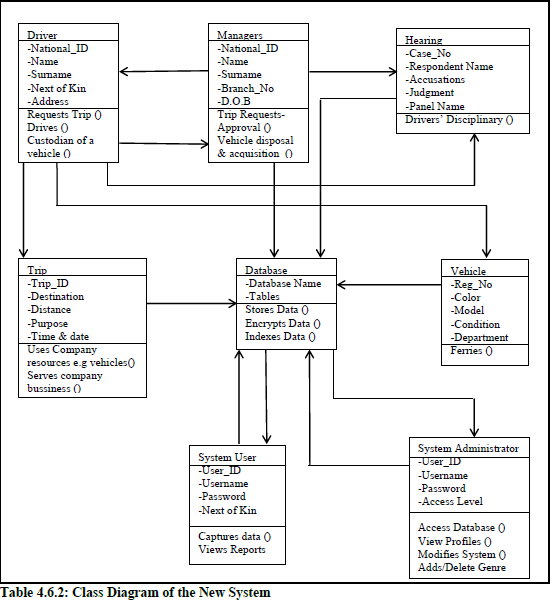
A package is a grouping of modeling elements under an assigned name. The services that a package provides are the result of the services provided by its internal parts that is its classes for example on the domain package in the diagram below has three sub-packages with different functions and those functions determine the kind of services offered by the domain package.



***Table 4.6.1: Package Diagram of the New System***

## **4.6.2 Class Diagram**

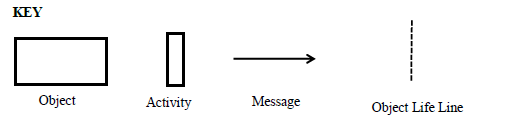
It is a brainstorming diagram comprised of lines, entities and attributes which depict relationships or associations among program classes. The important concept covered in this diagram is COUPLING which is the quantitative measure of the degree to which classes are connected one to another. As classes and components become more interdependent, coupling increases, so the important objective is to keep coupling as low as possible.

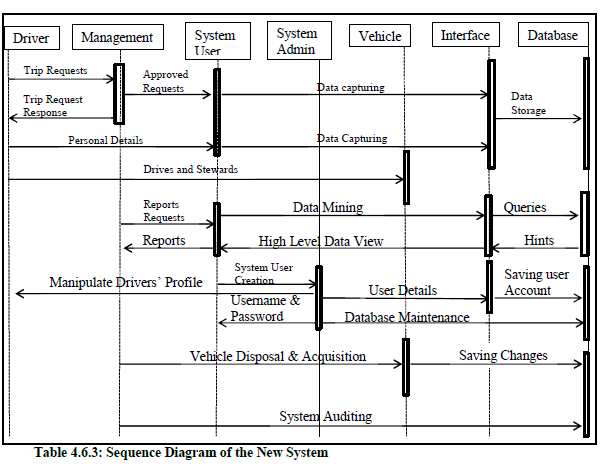


## **4.6.3 Sequence Diagram**

Sequence diagrams are created as a design activity, they describe implementation detail and how the objects of the emerging design would enact the use case through messages and collaboration.

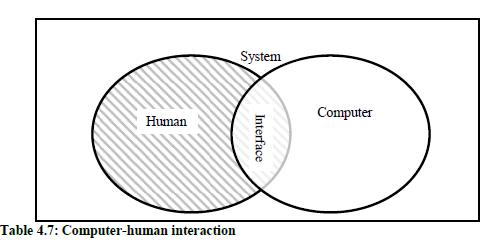
Sequence diagrams show the sequence of events between collaborating objects. Interaction is set of messages in some behavior that are exchanged between objects across links. Process begins with a driver presenting trip requests to the management which in turn gives feedback concerning approval or disapproval of the requested trip and incase of approval then the sequence begins up to the storage of trip details in the system database.





## **4.7 INTERFACE DESIGN**

Deborah J Mayhew: Principles and Guidelines in Software User Interface, “Interface is the means through which two subsystem (the computer and the human) communicate with one another”. Basically the human is flexible or adaptable and most computer systems are not, inputs must be made in a particular format and outputs are predefined thus for a computer system a human can learn and adapt while the computer cannot. This puts the burden of successful interaction totally on the user hence then need for a computer-human interface. The table below illustrates the relationship between humans, system and computer.



Interface design will take account of the needs, experience and capabilities of the system users through the application of the following design principles although not all principles are applicable to all designs.

|  |  |
| --- | --- |
| ***Principle*** | ***Description*** |
| User familiarity | The interface should use terms and concepts which are drawn from the experience of the people who will make most use of the system. |
| Consistency | The interface should be consistent in that, wherever possible, comparable operations should be activated in the same way. |
| Minimal surprise | Users should never be surprised by the behavior of a system. |
| Recoverability | The interface should include mechanisms to allow users to recover  From errors. |
| User guidance | The interface should provide meaningful feedback when errors  occur and provide context-sensitive user help facilities. |
| User diversity | The interface should provide appropriate interaction facilities for  different types of system user. |

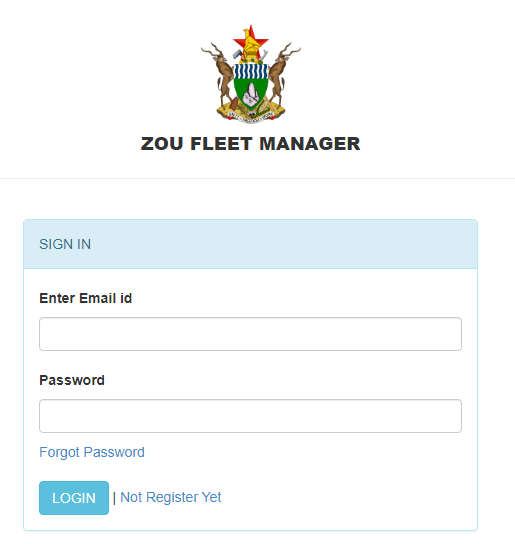
## **4.7.1 INTERAFCE DESIGN STRUCTURE**

The graphical user interface (GUI) is going to be the standard middle-man between the database and the users. It is going to be comprised of windows, menus and pointers which are going to aid user navigation through the required functionalities. Such an interface will facilitate better input designs and output designs e.g. login form and reports.

**4.7.2 Input design**

The major aim of this design is to facilitate a user friendly way data capturing in a validated format so that data integrity of the database maybe promoted easily and consistently. The

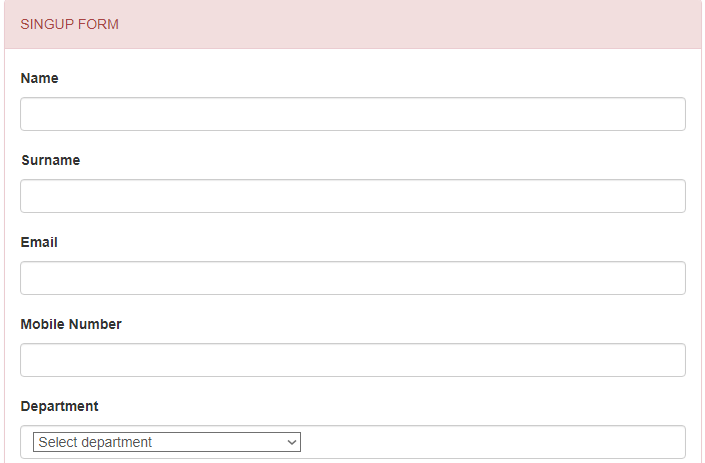
input design also includes hot keys or short cuts for easy maneuver accessibility of various resources for example the tab key for moving to the next input box. Most importantly all input boxes will be vividly labeled so that there may be no mismatch or confusion on the part of the system user. Error messages are also going to be put in place in order to serve the purpose of sign posts diction giver for wandering users. Below are the blue prints of the major input forms of the new system.



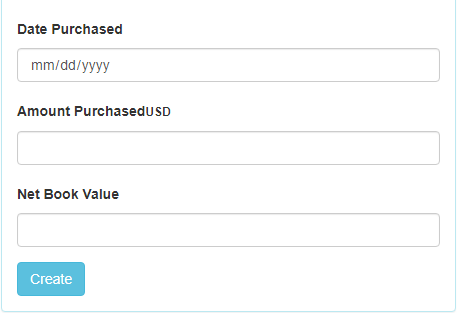
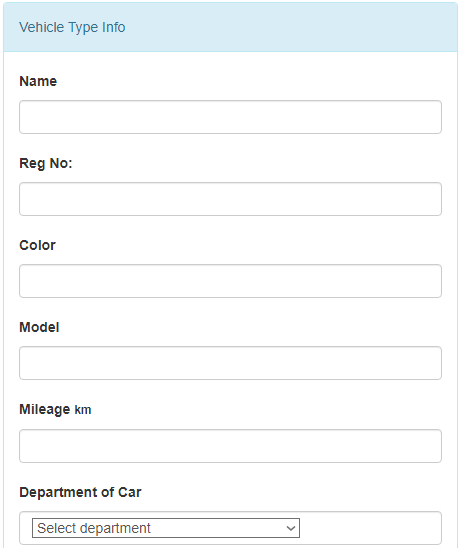
***Table 4.7.21: User Login Form***

**Fig 4.7.21** above shows the typical appearance to be of the non-administrator user login form, basically comprised of two input boxes and their corresponding labels and two commanding buttons, one for signing in the authenticated user and the other for clearing the text boxes and exiting the current page to the home page. An administrator will login separately from the general users; rather he will have his distinct login form.

Another input form is the “New User” form. Basically the form allows addition of a new system user and his access level. Its appearance is as shown below.



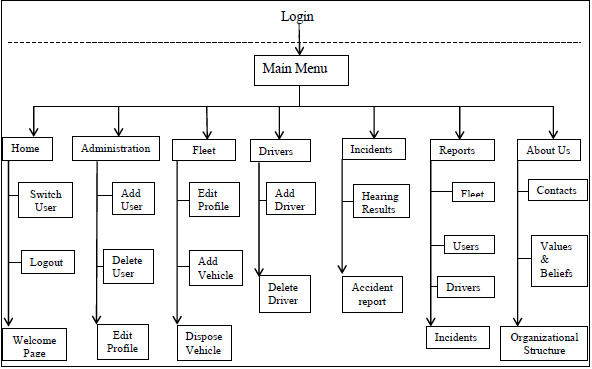




**Fig 4.7.23** is the vehicle login form where the main transaction of the system is performed. This form serves to capture all vehicle details concerning all the business trips that vehicle will carry out.

## **4.7.3 Main Menu design**

After a successful login the user will be directed to the main menu page which contains all the functionalities links of the ZOU Fleet Manager. This page will display several sub-menus which are designed to accommodate more data without compromise on the presentation of information. The basic structure of this page is as shown in the table below.



***Table 4.7.3 Main Menu Structure***

**4.7.4 Output design**

Output design involves specifying how production of on-screen reports and paper-based reports will occur. System output is the most important and direct source of information to the user. Efficient, intelligible output design improves the system relationships with the user and help in decision making. Another form of output considered is the hardcopy from the printer. The following is a typical design of one of the reports to be produced by the system.

## **4.8 CONCLUSION**

The basic structure and architecture of the new system has been developed and the next stage is the actual implementation of the system to produce something tangible. This document is going to be the guiding blue print during the coding phase.

# **Chapter 5**

## **5.0 introduction**

Various designs were laid out in the previous chapters and a workable one was decided on which shall now be implemented. There has also been given an outline of how the input and output of the proposed system will be like. In this chapter, the developer exhibits sample codes that will help in getting what expected output from the system and the various ways in which the new system can be implemented and these are discussed in this section.

## **5.1 SYSTEM CODE**

In this section we will discuss the programming language that will be used for development, programming style, data storage, connection method, processing method as well as the input and output methods.

Coding

The designer used:-

* Php for the connection of my modules to the database.
* HTML and CSS for styling the WEB pages
* JavaScript for validation of all the forms.
* MySql for the database queries

## ***5.1.1 PROGRAM CODE SEGMENTS***

Code for login, making changes, adding vehicle

<?php

session\_start();

include ("includes/connection.php");

//login with roles

// constant time string compare

function isEqual($str1, $str2)

{

$n1 = strlen($str1);

if (strlen($str2) != $n1) {

return false;

}

for ($i = 0, $diff = 0; $i != $n1; ++$i) {

$diff |= ord($str1[$i]) ^ ord($str2[$i]);

}

return !$diff;

}

if(isset($\_POST['login'])){

$email=mysqli\_real\_escape\_string($con,$\_POST['inputEmail']);

$pass=mysqli\_real\_escape\_string($con,$\_POST['inputPassword']);

$result = mysqli\_query($con, "SELECT \* FROM user\_registration WHERE email\_address='".$email."'");

$row = $result->fetch\_assoc();

$db\_hash = $row['password']; // field with the password hash

$given\_hash = crypt($pass, $db\_hash);

if (isEqual($given\_hash, $db\_hash)) {

//if (isEqual(0, 0)) {

// user password verified

//$result = mysqli\_query($con, "SELECT \* FROM user\_registration WHERE email\_address='".$email."' AND password='".$pass."'");

$test= $row['role'];

$active = $row['active'];

$id = $row['id'];

$name = $row['name'];

$surname = $row['surname'];

// checking to see if the email exists

$\_SESSION['email\_address']=$email;

if($test=='ADMIN')

{

if($active == 2){

$\_SESSION['email\_address']=$email;

$\_SESSION['last\_login\_time']= time();

$nguva = date('d-m-Y h:i:sa');

$query = "insert into logbook (name, surname, email, login) values ('$name','$surname','$email','$nguva')";

$insert = mysqli\_query($con, $query);

echo "<script>window.open('admin/index.php','\_self')</script>";

}

}

if($test=='DEPARTMENT\_TEAM')

{

if($active == 1){

$\_SESSION['email\_address']=$email;

$\_SESSION['last\_login\_time']= time();

echo "<script>window.location.href='users/aspiring\_application.php'</script>";

}else if($active == 2){

$\_SESSION['email\_address']=$email;

$\_SESSION['last\_login\_time']= time();

echo "<script>window.location.href='department/index.php'</script>";

}

}

if($test=='MANAGER')

{

if($active == 1){

$\_SESSION['email\_address']=$email;

$\_SESSION['last\_login\_time']= time();

echo "<script>window.location.href='users/aspiring\_application.php'</script>";

}else if($active == 2){

$\_SESSION['email\_address']=$email;

$\_SESSION['last\_login\_time']= time();

echo "<script>window.location.href='manager/index.php'</script>";

}

}

if($test=='DRIVER')

{

if($active == 1){

$\_SESSION['email\_address']=$email;

$\_SESSION['last\_login\_time']= time();

echo "<script>window.location.href='users/aspiring\_application.php'</script>";

}else if($active == 2){

$\_SESSION['email\_address']=$email;

$\_SESSION['last\_login\_time']= time();

echo "<script>window.location.href='driver/index.php'</script>";

}

}

if($test=='FINANCE')

{

if($active == 1){

$\_SESSION['email\_address']=$email;

$\_SESSION['last\_login\_time']= time();

echo "<script>window.location.href='users/aspiring\_application.php'</script>";

}else if($active == 2){

$\_SESSION['email\_address']=$email;

$\_SESSION['last\_login\_time']= time();

echo "<script>window.location.href='finance/index.php'</script>";

}

}

}else{

echo "<script>alert('Email or password is not correct, try again!')</script>";

//echo "<script>window.location = 'login.php'</script>";

}

}

?>

## ***Code for adding vehicle***

<?php

include('../session.php');

$today = date('d-m-Y h:i:sa');

?>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" />

<meta name="description" content="" />

<meta name="author" content="" />

<title>ZOU FLEET MANAGER</title>

<!-- BOOTSTRAP CORE STYLE -->

<link href="assets/css/bootstrap.css" rel="stylesheet" />

<!-- FONT AWESOME STYLE -->

<link href="assets/css/font-awesome.css" rel="stylesheet" />

<!-- CUSTOM STYLE -->

<link href="assets/css/style.css" rel="stylesheet" />

<!-- GOOGLE FONT -->

<link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css' />

</head>

<?php

//initialize variables

$date = "";

$department = "";

//if post button is clicked

if(isset($\_POST['save'])){

$car\_name=mysqli\_real\_escape\_string($con,$\_POST['car\_name']);

$reg\_num=mysqli\_real\_escape\_string($con,$\_POST['reg\_num']);

$color=mysqli\_real\_escape\_string($con,$\_POST['color']);

$model=mysqli\_real\_escape\_string($con,$\_POST['model']);

$mileage=mysqli\_real\_escape\_string($con,$\_POST['mileage']);

$car\_department=mysqli\_real\_escape\_string($con,$\_POST['car\_department']);

$date\_purchased=mysqli\_real\_escape\_string($con,$\_POST['date\_purchased']);

$amnt\_purchased=mysqli\_real\_escape\_string($con,$\_POST['amnt\_purchased']);

$net\_book=mysqli\_real\_escape\_string($con,$\_POST['net\_book']);

$email = $login\_session\_email\_address;

$query = "INSERT INTO `vehicle`(`car\_name`, `reg\_num`, `color`, `model`, `mileage`, `car\_department`, `date\_purchased`, `amnt\_purchased`, `net\_book`)

values ('$car\_name','$reg\_num','$color','$model','$mileage','$car\_department','$date\_purchased','$amnt\_purchased','$net\_book')";

mysqli\_query($con, $query);

if($query){

echo "<script>

alert('You have successfully added a vehicle.');

</script>";

//redirecting to index page after saving info to the database

}

}

?>

<body>

<!------MENU SECTION START-->

<?php include('includes/header.php');?>

<!-- MENU SECTION END-->

<div class="content-wra

<div class="content-wrapper">

<div class="container">

<div class="row pad-botm">

<div class="col-md-12">

<h4 class="header-line">Add vehicle</h4>

</div>

</div>

<div class="row">

<div class="col-md-6 col-sm-6 col-xs-12 col-md-offset-3"">

<div class="panel panel-info">

<div class="panel-heading">

Vehicle Type Info

</div>

<div class="panel-body">

<form role="form" method="post">

<div class="form-group">

<label>Name</label>

<input class="form-control" type="text" name="car\_name" autocomplete="off" required />

</div>

<div class="form-group">

<label>Reg No:</label>

<input class="form-control" type="text" name="reg\_num" autocomplete="off" required />

</div>

<div class="form-group">

<label>Color</label>

<input class="form-control" type="text" name="color" autocomplete="off" required />

</div>

<div class="form-group">

<label>Model</label>

<input class="form-control" type="text" name="model" autocomplete="off" required />

</div>

<div class="form-group">

<label>Mileage <small>km</small></label>

<input class="form-control" type="text" name="mileage" autocomplete="off" required />

</div>

<div class="form-group">

<label>Department of Car</label>

<div class="form-control">

<select class="span3" name="car\_department" >

<option value = "Select department" class="form-control">Select department</option>

<?php

$get\_countries = mysqli\_query($con, "select \* from department");

while($row1 = mysqli\_fetch\_assoc($get\_countries)){

if($department==$row1['id']){

echo"<option selected = 'selected' value = ".$row1['id'].">"

.$row1['department']."</option>";

}

else{

echo"<option value = ".$row1['id'].">".$row1['department']."</option>";

}

}

?>

</select>

</div></div>

<div class="form-group">

<label>Date Purchased</label>

<input class="form-control" type="date" name="date\_purchased"autocomplete="off" required />

</div>

<div class="form-group">

<label>Amount Purchased<small>USD</small></label>

<input class="form-control" type="text" name="amnt\_purchased" autocomplete="off" required />

</div>

<div class="form-group">

<label>Net Book Value</label>

<input class="form-control" type="text" name="net\_book" autocomplete="off" required />

</div>

<button type="submit" name="save" class="btn btn-info">Create </button>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- CONTENT-WRAPPER SECTION END-->

<?php include('includes/footer.php');?>

<!-- FOOTER SECTION END-->

<!-- JAVASCRIPT FILES PLACED AT THE BOTTOM TO REDUCE THE LOADING TIME -->

<!-- CORE JQUERY -->

<script src="assets/js/jquery-1.10.2.js"></script>

<!-- BOOTSTRAP SCRIPTS -->

<script src="assets/js/bootstrap.js"></script>

<!-- CUSTOM SCRIPTS -->

<script src="assets/js/custom.js"></script>

</body>

</html>

## **5.2 SYSTEM CODING**

The proposed system is developed using PHP programming language. This is because PHP is the most popular scripting language on the web. It is used to enhance web pages. With PHP and MySQL i created username and password login pages, check details from a form, create forums, picture galleries and a whole lot more. It is even possible to create PDF documents on the fly.

PHP is known as a server-side language. That's because the PHP doesn't get executed on your computer, but on the computer you requested the page from. The results are then handed over to you, and displayed in your browser. So since we want a web based system PHP would be the best language to use.

## **5.3 PROGRAMMING STYLE**

There are many approaches to system development that can be used but we will use the Structured Approach to Analysis and Design in our implementation. In implementing this approach, we will use the Top-down Approach together with Modular programming.

**5.3.1 The Top-Down Approach**

This approach allows us to ascertain overall organizational objectives along with ascertaining how they are best met in the overall system. The system will be divided into subsystems and their requirements.

Advantages of the Top-down approach

* It allows us to avoid the chaos of attempting to design the system “all at once”.
* It gives us the ability to have separate systems analysis teams working in parallel on different but necessary subsystems.
* It eliminates losing sight of the system as a result of getting so mired in detail.

Disadvantages

* There is a danger that the system will be divided into the wrong subsystems.
* Once subsystem divisions are made, their interfaces may be neglected or ignored.
* The subsystems must be reintegrated, eventually.

5.3.2 The Top-Down Design of the proposed system

The proposed system will be decomposed into six subsystems, each of which will be divided into subtasks. The six subsystems of the proposed system will be as follows:

1. File – This will enable creation of new records and saving of backup files.
2. Queries – This will allow the querying of database tables to produce relevant data.
3. View/Edit – This will allow the viewing and editing of records.
4. Print – This will allow the printing of reports, all account details. The print function will be enabled by the browser’s print button.
5. Security options – This will enable users to change their passwords as well as the information systems manager to add new users to the system.
6. Exit – This will allow users to exit the system by showing the log out dialogue box.

The proposed system will make use of **Modular programming** and the Top-down approach. Modular programming concept is useful for the top-down approach. Once the top-down design approach is taken, we will break the whole system into logical, manageable sized modules or subprograms in order to use modular programming techniques.

Advantages of modular programming

* Modules are easier to write and debug.
* Tracing an error in a module is less complicated.
* Modules are easier to maintain.
* Modules are easier to grasp because they are self-contained subsystems.
* It is to modify or insert modules.
* As individual programmers and design, we can be given different modules and work independently of each other.

## **5.4.0 TESTING**

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum amount of efforts.

Software Testing is the process of executing a program or system with the intention of detecting errors (Brien,1996). Testing is usually done for to improve quality, for Verification and Validation (VV) and for reliability estimation. Testing of the system was done so that errors can be discovered and fixed before the system is put to use. This is very crucial so that we make sure that the system is perfect and does not give problems to the end users.

Two classes of inputs are provided to test the process

1. A software configuration that includes a software requirement specification, a design specification and source code.
2. A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

## **5.4.1 SYSTEM TESTING**

System testing is testing that was conducted on the complete, integrated On Transport management System to evaluate the system’s compliance with its specified requirements. System testing took as its input, all of the integrated software components. The aim behind system testing was to detect defects both within the inter-assemblages and also within the system as a whole.

## **5.5.1 TESTING PROCESS**

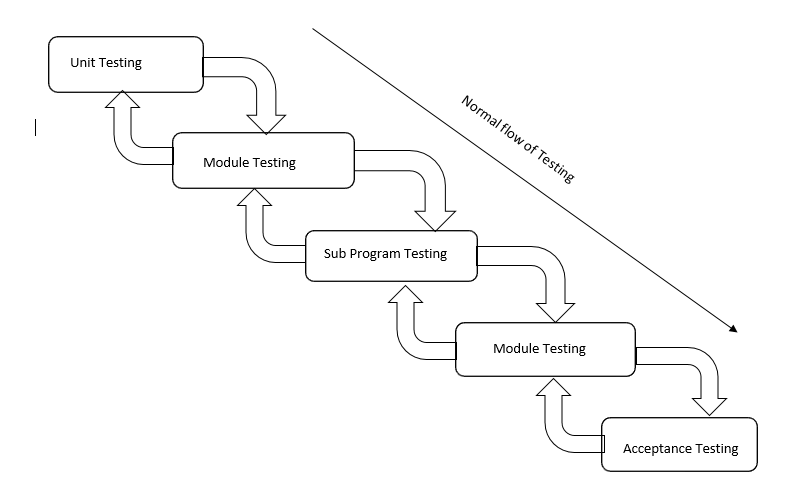


Figure 5.5.1 Testing process

**5.5.1.1 Unit Testing**

During unit testing each form on the system was tested to see if it performed as per requirement. Each component of the system was tested individually to ensure its functionality and integrity.

A test case design method that uses the control structure of the procedural design to derive test cases. By using white box testing methods, the developer could derive test cases that:

* Guarantee that all independent paths within a module have been exercised at least once.
* Exercise all logical decisions on their true and false sides.
* Execute all loops at their boundaries and within their operational bounds.
* Exercise internal data structures to ensure their validity.

During unit testing some errors were raised and all of them were rectified and handled well. The result was quiet satisfactory and it worked well.

**5.5.1.2 Module Testing**

In this case each of the four modules, each module was created separately and tested separately then linked to form one. This testing strategy was instrumental in the understanding of all the validation that needed to be coded as per module. Module testing was rather the test of related unit components of the system. Execution paths, error-handling paths, normal, abnormal and extreme data were thoroughly tested. The idea behind was to check if the system was updating the accounts before the customer left. Also to find out if the program would make subtractions in database once the customer pays his or her bill. The system was capable of producing the systems objectives and therefore concluded to be working properly.

***Subsystem Testing***

System modules that were related formed system subsystems these were tested for data, stress, and error integrity. Database was also tested as a subsystem of its own. This is also called link testing. It tests a collection of modules, which have been integrated into subsystems. It ensures job streams are correct. There is detection of interface mismatches and demanding exercises of the interface between programs.

**5.5.1.3 System Testing**

System testing is the testing of all the integrated modules that makes up the system to check if it was performing the required functions. The system is able to add new records of student. All queries are executed by the system.

**5.5.1.4 Integration Testing**

Integration testing is a system technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design. Bottom-up integration is the traditional strategy used to integrate the components of a software system into functioning whole. Bottom-up integration consists of unit test followed by testing of the entire system. A sub-system consists of several modules that communicated with other defined interface. The system was done the integration testing. All the modules were tested for their compatibility with other modules. They test was almost successful. All the modules coexisted very well, with almost no bugs. All the modules were encapsulated very well so as to not hamper the execution of other modules.

**5.5.1.5 Security Testing**

It attempts to verify that protection mechanisms built into a system will in fact protect it from improper penetration. The system’s security must of course be tested from in vulnerability form frontal attack.

**5.5.1.6 Black Box Testing**

Black box testing is done to find out the following information as shown in below:

1. Incorrect or missing functions.
2. Interface errors.
3. Errors or database access.
4. Performance error.
5. Termination error.

The mentioned testing is carried out successfully for this application according to the user’s requirement specification.

**5.5.1.7 Test Data Output**

After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

**5.5.1.8 Performance testing**

Testing conducted to evaluate the compliance of a system or component with specified requirements. For example a performance requirement might state that the system automatically updates in less than 3 seconds.

**5.5.1.9 Usability testing**

Testing to evaluate the extent to which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component. This can be done by all system users interacting with the system.

**5.5.1.10 Acceptance testing**

The goal of system testing is to confirm that the system is complete, meets the organizations’ needs that prompted the system to be developed, and is acceptable to users. The system is tested with data supplied by users rather than simulated data. Acceptance testing helped reveal errors and omissions in the system requirements definition because real data tests the system in different way from the test data. Acceptance testing also revealed requirements problems where the system’s facilities did not really meet the user’s needs. Acceptance testing is done in two stages: alpha and beta testing.

## **6.6.0 VALIDATION**

To ensure that students always provide valid input, a set of validation controls are added to the update web form and login page. The validation controls provide an easy-to-use but powerful way to check for errors and where necessary, display messages to the user (Microsoft). By using validation controls custom error information is displayed to the user. The type of validation controls used in the development of the ZOU FLEET MANAGER System is:

* Required Field Validator

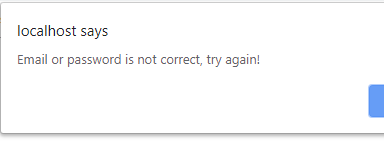
This validation control ensures that the user does not omit the entry of required information during the registration process. Since most of the information requested is compulsory, this control is used frequently. If a user omits a required entry, an asterisk will appear along the required field to prompt the user to complete.

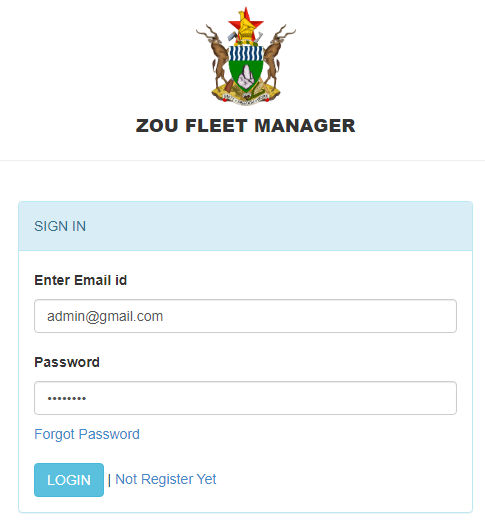
* Regular Expression Validator

This validation control is used to check that an entry made by a student matches a pattern defined by a regular expression. This type of validation enables the checking for predictable sequences of characters such as addresses and date on the update form.

The required information will be flagged by asterisks. The asterisks represent different validation controls and on the right are the properties for a single validation control.

Scenario 1

The system will pop up a message box below.



## **6.1.0 VERIFICATION**

This was also intensively done. This is whereby we looked at the system to see whether we had developed the correct system. This means that the system was checked to see if it was meeting the customer specifications and requirements. The system might be running and fully functional but not meeting the user requirements. We managed to check and analyse system representation using static techniques to check on requirements documents, design diagrams, program source code and inspections. We tested the system with some data and compared the results with already known results. We used the white box method for verification where the tests were conducted to ensure that the internal operation of the system performed according to specifications and all the internal components had been adequately exercised.

**6.2.0 CONCLUSION**

This penultimate chapter outlined the implemented application and how it was tested. Possible appropriate tests have been done to assess system but this does not automatically mean that a flawless system has been developed by the researcher. There are always undiscovered errors since testing can only be done up to the tester’s cognitive ability. The ultimate chapter addresses the challenges faced by the researcher in view of project accomplishments and quality assessment and gives recommendations for further work.

# **CHAPTER 6**

# **IMPLEMENTATION & POST IMPLEMENTATION PLAN**

## **6.0 SYSTEM INSTALLATION**

Installation is when the developed system is being installed. Users are changing from using the old manual system to using the new system. In this phase the required software is installed on the appropriate hardware converting from the as-is system to the to-be system. Users are moved from using the old system to using the new system. The system is installed within the Zimbabwe Open university network as defined within the design documentation.

Steps For Application Software Installation

* The system is installed from the software C.D
* Install system to path. (It is recommended that the system be installed to the program files folder.)
* Verify that the system is installed properly that is all folders are installed.
* Add the system data source.
* Connect data source to the database on the main server housing the system database.

Database Installation

The database management system (DBMS) will use MySQL Server. The DBMS is installed on a central server, which will serve all the other terminals and thus is the Central database

## **SYSTEM CONVERSION**

According to Cliff (1994) system conversion is a procedural process where our system replaces the old system and this will take effect after the establishment of the operational environment and training. It is technical process where system replaces the old system and will take effect after the establishment of the operational environment and training. It had discovered that there are several methods that can be used to install the new system but the following are recommended

* Pilot Conversion
* Direct Conversion
* Phased Conversion
* Parallel Conversion.

### **6.1.1 PILOT CONVERSION**

With this conversion, the system is installed to a small section in the organization, for testing purposes. That is to say that the system will be used by a selected group of users who will further assess its acceptability and functionality. Cost is relatively moderate since only one location runs both systems. Risk is also relatively moderate as risk of failure is reduced to the pilot site.

Advantages

* Moderate costs as only a chosen site would be running two systems at once
* A moderate risk of failure as the new system is only installed at the pilot site.
* The developed system is placed in actual site and tested before actual use to check if the system operates according to the set aims and objectives. System processes are checked to see if they perform according to the user requirements.

### **6.1.2 DIRECT CONVERSION**

Direct changeover is when the old system ceases to function abruptly and the new system takes over usually over a weekend or during a slack period. The Old system will stop operating as all the users will have been updated with the new system. This strategy has a relatively low cost of implementation, however there is an imminently high risk if the new system fails to live up to expectations or does not provide better functionalities compared to the old one.

Advantages

* Efficient method in so far as it minimizes duplication of work
* Less costs as only one system would be in operation

Disadvantages

* New system may not be entirely correct
* It is difficult to make the system operational when some errors are identified after changeover and need correction
* Requires careful planning , testing and attention to operational detail

### **6.1.3 PHASED CONVERSION**

On phased conversion the new system is installed in different stages. This is done while the old system is being slowly phased out. Users can easily adjust to the new system as there is no rush in implementing it. Cost is relatively moderate because the system is implemented in stages. Risk is also relatively moderate because the risk is limited to the module being implemented.

### **6.1.4 PARALLEL CONVERSION**

Parallel conversion involves running the two systems together at the same time. This gives the users a better background and backup to refer back to if the new system fails to live up to the expected standards. The implementation costs are relatively high as both systems are in operation for the changeover period, Risk is relatively low.

Advantages

* Low risk as a result can be verified and a back-up option exists

Disadvantages

* Relatively high costs as both systems are in operation at the same time for some time
* Method cannot be used for systems which are not similar

**6.1.5. Recommendation on changeover strategy (pilot changeover)**

From the above changeover methodologies considering the type of data and risk that the system might pose if a smooth transition may at any time fail, the recommended method would be a pilot changeover. The system will be used by a subsection of the entire organization who will further examine and asses the system functionality and acceptability.

6.1.5.1. Reasons for choosing pilot changeover strategy

* When the new system fails only one part will be affected; the rest will still be functional.
* Staff will be trained in one area only which is much quicker and less costly than other change over strategies.

6.1.5.2. Conclusion drawn from changeover analysis strategies.

* From the above it can be concluded that pilot changeover is the conversion strategy that suits best the system considering factors such as relative costs and impact of failure which have a huge impact to the organization.

## **6.2.0 TRAINING**

Training of the user will be done on a personal level as few people hence the intricacy of the system can be taught to the users faster and more effectively at the terminals.

Training is done at two levels

* Module level: this is for the particular modules that concern the particular users.
* System level. This is for management who must appreciate the development of the system and its function. Users who have access rights to all modules also must be versed with the entire system as it functions.

## **6.3.1 MAINTENANCE**

Software maintenance is the process of modifying a software system or component after delivery to correct faults, improve performances or other attributes, or adapt to a changed environment. Once a system enters a period known of its cycle as maintenance, which represents the greatest expenditure of effort and resources, the information needs of an organization constantly changes in response to external and internal factors.

External factors that may require changes include the business climate, imposition of rules and regulations by government. Internal factors include changes in user requirements and responses to methodological and technological changes. In addition, all systems contain inadequacy and errors overloaded during system development.

## **6.3.2 SYSTEM MAINTENANCE**

Maintenance is an important process to be given good attention if the system is to live longer while delivering goods. The system maintenance is important because:

* A system continues to change and evolves as it is used.
* The changes will be arising, change and evolve as it is used.
* The changes will be arising from change request due to the problems.
* Reports from operating groups who identifies the bugs in the system that must be fixed.
* Changes will be arising from users.
* There are three different types of system maintenance, which are:
* Corrective maintenance
* Perfective maintenance
* Adaptive maintenance



***Figure 6.3.2 System Maintenance***

### **6.3.2.1 CORRECTIVE MAINTENANCE**

This is concerned with correcting discovered and reported error. Errors range from wrong implementation (dividing instead of multiplying) to completely not performing. Once an error has been reported, investigations are to be launched to establish the root and cause of the errors. As soon as these are established, a design is mapped out for how to effectively correct the error, followed by the correction (implementing the corrective design). Once a correction has been made, weekly reviews or follow ups should be made to make sure that everything is in place, and to make sure that no new bugs were introduced during the fixing. All the correction activities from the investigations to the implementation are documented.

### **6.3.2.2 ADAPTIVE MAINTENANCE**

Like any other system, the **ZOU FLEET MANAGER SYSTEM** operates in an ever-changing environment. The changing of the environment also detects that the system be accordingly changed. During adaptive maintenance, the situation or condition calling for a change is analyzed to get a better understanding, leading to designing how to integrate or introduce the design into the existing system, and finally implementing the change. Once a change has been implemented, monthly reviews, depending on the size of the change, are to be done to make sure that the implementation was correct, and to make sure that no bugs were introduced during the implementation of the change. All adaptive maintenance change activities, from analysis all the way to the implementation should be documented.

### **6.3.2.3 PERFECTIVE MAINTENANCE**

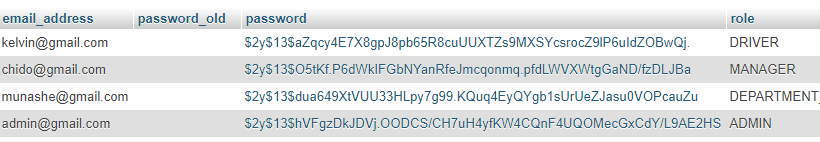
During development it is not always the case that the best implementation is implemented. It’s not by design that the best implementations are not implemented, but at times situations such as schedule constraints or even ignorance, detect that it be like that. During perfective maintenance the goal is to implement a better version of what already in place, as well as to add missing functionalities. This is achieved by first assessing whether it is worth the effort to perfect the system to those extents. If it is, then design specifications should be drafted and when approved implemented. As with any other maintenance activity reviewing is done. All activities ranging from analysis to implementation are to be recorded.

## **SECURITIES**

### **7.3.1 SOFTWARE APPLICATION BACKUP SERVICES**

The system developer will provide the application backup service. For system code related problems, system code debugging and modifications the organization will need to consult the developer or any computer software developer who is familiar with the PHP language. As for the operating system backup service the operating system software supplies will provide this.

Sensitive information is encrypted. E.g. this password



### **6.4.2 HARDWARE BACKUP SERVICES**

Computer hardware backup service, which may be needed, is hardware servicing, repair, and maintenance.

### **6.4.3 SYSTEM REVIEW**

Both the system developer and the system users do this. It is a review of the system performance and this will be done annually. The review will be based on determining the necessary changes needed to be done to the system so as to make it fulfill the new user’s needs, which may have up due to changes in the user-operating environment.

## **6.5 INSTALLATION**

6.5.1 Steps for Application Software Installation

* The system is installed from the software C.D
* Install system to path. (It is recommended that the system be installed to the program files folder.)
* Verify that the system is installed properly that is all folders are installed.
* Add the system data source.
* Connect data source to the database on the main server housing the system database.

6.5.2 Database Installation

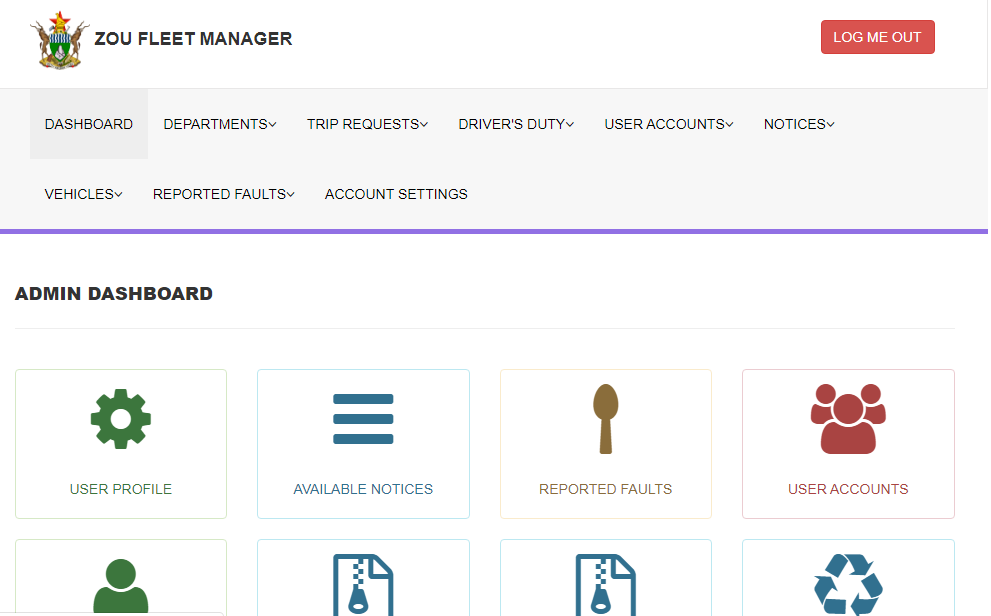
The database management system (DBMS) will use MySQL Server. The DBMS is installed on a central server, which will serve all the other terminals and thus is the Central database

## **6.6 USER TRAINING**

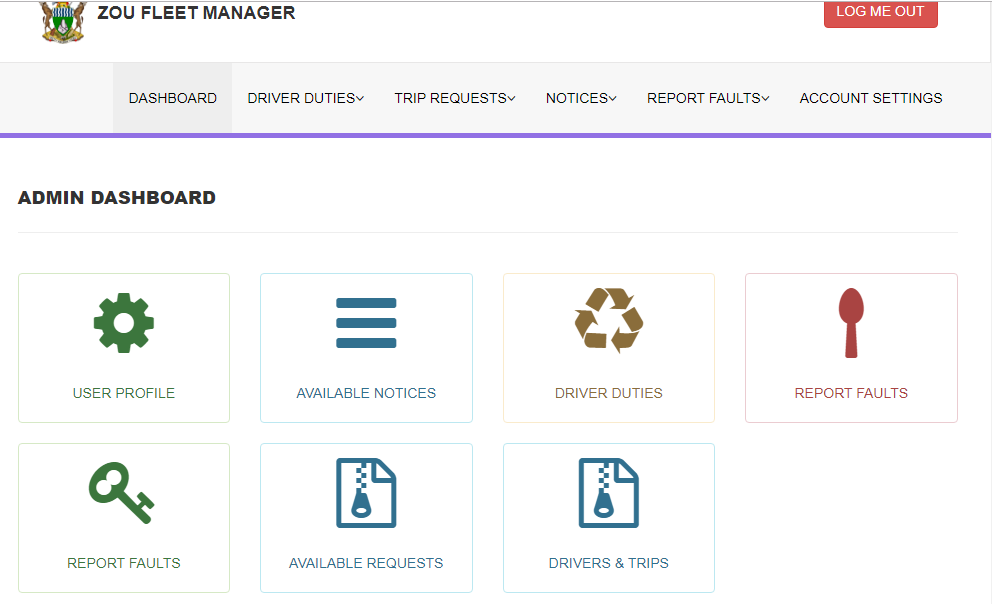
**6.6.1 User Manual**

Logging into the system you enter you user name and password and select the group of users you are in.

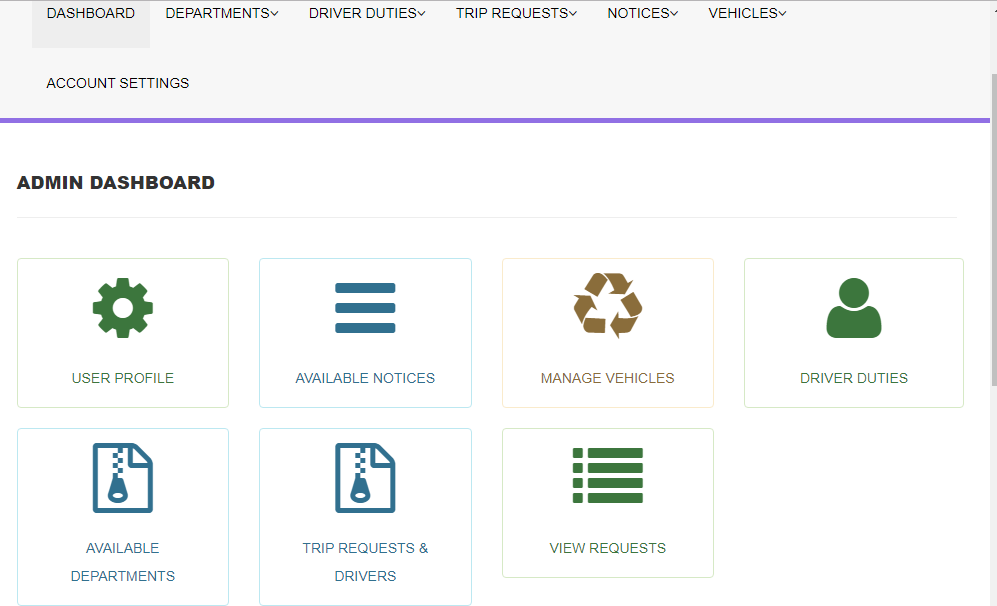
6.6.1.1 Administrator main



7.6.1.2 driver main



6.6.1.2 manager Main



# **CHAPTER 7**

## **SUMMARY, CONCLUSION & RECOMMENDATIONS**

## **7.1SUMMARY**

A summary of the whole research project and evaluation of the objectives is discussed in this chapter. It gives an insight to the development of the ZOU FLEET MANAGER SYSTEM. It also outlines and discusses the challenges faced during the development of this system and finally gives the recommendations for the future works.

### **7.1.1 PROJECT REVIEW**

Project review is necessary to ensure that the system meets the specified objectives and is operating properly. Critical areas taken into consideration are:

* System quality and effectiveness
* System reliability and maintainability
* User satisfaction
* Accuracy, completeness and timeliness of output
* Adequacy of system controls and security

### **7.1.2 CHALLENGES ENCOUNTERED**

Although a lot of positive achievements have been noted some of the challenges that were faced during the project were as follows:

* Limited application domain knowledge: Having little appreciation for the fleet management field meant that the system developer/ knowledge engineer had to learn a lot of terminology from scratch. This was aggravated by limited availability of up to date literature on the implementation of a decision support system of this nature, and adequate resources that were required for the project.
* Gaining proficiency in the use of some of the tools: A lot of time was spent learning and which was the adopted inference engine. Also a lot of time was spent experimenting on programming language and MySQL as the system developer had little in-depth knowledge of the two.
* Lack of knowledge on Support Systems Design at the beginning of the project work: The modules ‘Web Development’ and ‘Database Systems Concepts and Design’ were done at the time that the project was expected to be underway. These modules were vital for the understanding of programming and backend design which was required for the project.

## **7.2 CONCLUSION**

The development of the ZOU FLEET MANAGER is a demonstration of a fruitful academic experience at Kushinga Phikelela Polytechnic. It eluded some key factors in the industry such as a sense of responsibility, time management and planning within limited resources. Given the fact that all the project objectives were met in a constrained time frame and limited resources it can be concluded that the project was a success. This system has a lot of room for expansion and only marks the beginning of automated expertise in the transport management field.

## **7.3 RECOMMENDATIONS**

A lot of further improvements can be made to the ZOU FLEET MANAGER so as to improve its functionality and broaden the scope of the current system. The system can be improved to be more informative by including discussion forums for the workers, including information on how to keep a good moral relation with a company and can be made to be accessed via (short messages services).

* High level and sophisticated codes such JavaScript, cookie and XML as well as extensions such Web Assist and Developer Toolbox can be used in order to enhance the system functionally and also rectify all the weaknesses and defects of the system.
* The search function of the system will be enhanced in order to give the message when the record was not found in the database. Delete function can also be redesigned and enhanced in order to give the confirmation message together with the record to be deleted before deletion from the database.
* System can be configured with email functionality in order to verify and send the password to users when they forgot their passwords.
* It is thus essential for users to attend refresher training courses.
* The programmers must come up with system improvements from time to time to ensure that the system adapts well to the dynamic information technology environment.

# **CHAPTER 8**

## **8.0 BIBLIOGRAPHY**

Catanio, J., Yoo, J. , Ashish, G. (2003). “Relationship Analysis: A research plan for enhancing

System Analysis for Web Development”, Proceedings of the 36th Annual Hawaii

International Conference on 6-9 Jan 2003, pp 10-14.

Dennis, A. Wixom, B. H. and Tegarden, D. (2005). Systems Analysis and Design with UML

Version 2.0 An Object-Oriented Approach, 2nd Edition, USA: John Wiley and Sons,

Inc.

Friedman, J., (2005). “Has the Holy Grail of the Paperless Office Been Achieved?” Journal of

Information Management & Technology, Cimtech, St. Albans, ROYAUME-UNI (1992)

(Revue), pp.30-31.

Gottschalk, T. (2003), “Evaluation for Distance Educator”. University of Idaho Engineering

Outreach.

Muirhead, B. (2006). Effective online teachers. i-Manager’s Journal of Educational

Technology, 1(4), 14-21.

Quintero, A. M. R., Valderrama, J. T. (2002). “Separating the navagational aspect, Distributed

Computing Systems Workshops”, Proceedings 22nd International Conference on 2-5

July 2002, pp 419-423.

Sherry, L., (1996). “Issues in Distance Learning”, International Journal of Educational

Telecommunications, New York: Academic Press, 1996, 1 (4): 337-365.

Zhang, P., Benbasat, I., Carey, J., Davis, F., Galletta, D. and Strong, D (2002). Human-

Computer Interaction Research in the MIS Discipline, Communications of the AIS,

## 2002, 9, pp. 334-355.

Blind, (2001, 475 updating), Bozeman & Bretschneider, (1986, 475); Caudle, Gorman & Newcomer

(1991).Strategic Information System Planning in U. S.

Kruchten, P.(2004). The Rational Unified Process: An introduction (3rd ed.),Addison Wesley

Luke Wroblewski & Esa M. Rantanen (2009). Design Considerations,for web-based

Applications.

Post, G., & Anderson, D.(2006). Management information systems: Solving business problems

with information technology. (4th ed.). New York: McGraw-Hill Irwin.

Roper,M. (1994) Software Testing. New York: McGraw-Hillrwin.

Schwalbe, K. (2005). Information technology project management. (4th ed.). New York:

Thoinson.

Steven Alter (1996). "Basic Ideas for Understanding Information Systems" in Information

Systems: A Management Perspective, 2nd Edition, Benjamin Cummings, 1-9, 1996.

Blanchard, B. S., & Fabrycky, W. J.(2006). Systems engineering and analysis (4th ed.) New

Jersey: Prentice Hall.

c Bajgoric,Nijaz.(l999). Web-to-Host Connectivity Tools in Information Systems, Systems

Development Management

Conallen, J.(2003). Building Web Applications with UML, (Second Edition), Addison Wesley.

QI Connolly, T., & Begg, C.(2005). Database Systems, (2nd Edition), Addison Wesley.

Cem Kaner. (2009). "Activities in software testing education: a Lstructt~re,fomra pping learning

Objectives to activity designs " Software Testing Education Workshop (International Conference on Software Testing), Denver.

Elizabeth Levin & Catherine Pocknee; & Gabriella Pretto, (2009).The challenges of establishing

an internship program: Policy, expectations and workloads. Swinburne University of Technology.