NAME:LEEN MWENDWA

REG NO:SC211/0715/2018

BIT YEAR2 SEMESTER 2

MORNING STAR SCHOOL MANAGEMENT SYSTEM

PROJECT PREPARED BY:

GROUP 1

REPORT PREPARED BY:LEEN MWENDWA,SC211/0715/2018

March 2020

Table of Contents

SECTION ONE: PROJECT PROPOSAL

DECLARATION

The report is original work and has never been submitted to any other institution for examination purposes.

Name: LEEN MWENDWA REG NO:SC211/0715/2018

Signature……. Date…….

Supervisor Name: Dr GEOFFREY MARIGA
DATE……

Signature…….

ABSTRACT

Technology is advancing and replacing manual ways of doing things as a result of computerized machine and systems. Many organizations are interoperating technology into their day to day operation in order to cut down the costs.

Management systems are being developed to enable organizations to perform task more efficiently as well as without compromising on quality and integrity. This has made processing, storage and retrieval of information much easier. Security has also improved; this is to provide high standards of quality in terms of services provided.

DEDICATION

I dedicate this project to UBORA CAMPUS AND MORNING STAR ACADEMY

PREFACE

This document is mainly useful for everyone who intends to interact with a new system. In this report one will be able to know the problems and different challenges experienced before and also provide a user/admin manual on how to use the system.

ACKNOWLEGEMENT

Project is through due to help of many people but my greatest appreciation goes to the GROUP1 students taking a bachelor of science in information technology in their second year and DR GEOFFREY MARIGA.

BACKGROUND

GROUP 1 is a company dealing with school management, the company started as a small and now it has gain popularity thus attracting large numbers of customers. Company is faced with problem of lack of reliable system to record daily enrollments which make it difficult to track records. Need for the operation is to work on the operation in order to improve the services to our customers.

PROBLEM STATEMENT

Due to rapid growth of company I have noticed that there is need for a system which is going to replace old methods of student's registration during enrollment thus making work easier for the administration. Some of the consequences include

- ✓ Traffic to retrieve information about students or staff.
- ✓ Bulkiness of records in terms of keeping records.
- ✓ Lack of security due to lack of authentication.

PROPOSED SOLUTION

It is to develop a secure software system for the school management so as to make the operation easy and efficient.

OBJECTIVES

The main aim of this project is to develop a secure software system for effective management of the school that will be able to achieve the following objectives:

Project objectives

- ✓ To ensure quick access to student or staff information.
- ✓ Reduction of work force to both the staff and the users who are within the school.
- ✓ Increase customer satisfaction and transparency.

System objectives

- ✓ A system that will be confidential and secure processes of the school.
- ✓ A system that maintains correct database by providing an option to update the details of the students and staff in the school.
- ✓ A system that will provide authentication mechanisms to allow authorized users only to access.
- ✓ Ensuring that the system is user friendly.

SCOPE

The scope of this project is limited to the activities of students and staff, which includes securing the records of registration and adding, deleting, updating of the records of students and staff.

LITERATURE VIEW

This is the assessment of body of research that addresses a research question. The research should identify what is already known about the topic being studied. In today's world everyday students enroll to schools and they require registration .The staff is the administration ,the lecturers and the workers in various departments. Schools with such management systems are:

- Morning glory high school
- Ubora campus

METHODOLOGY

It means the method used to gather information or data which will help in developing the proposed system. These methodologies will help to identify the real requirements of the system. It also reduces any development risks which may arise during the system development.

What I would use:

- ✓ Observation ,this to check how the school is doing their registration.
- ✓ Interviewing, this will help to cover the ground quickly and also accurate information/reliable information
- ✓ Questionnaires, through asking of questions of what will be needed to be included into the system
- ✓ Through using of samples, a process of collecting a representative sample of documents, forms and records.

DEVELOPMENT METHODOLOGY

The choice of the system development is waterfall model. Since it is easy to understand, each phase is completed on time and also it is easier to implement since it is easier to plan. Its disadvantages are that it has higher risk that is if you make a mistake you cannot go back.

It consists of the following phases

- ✓ Planning which consist the work plan of the system
- ✓ Analysis - identifying of the problem
- ✓ Design- identify the system structure and behavior
- ✓ Implementation - installing of the system
- ✓ Maintenance checking of the errors

BUDGET AND RESOURCES

RESOURCE NAME	REQUIREMENTS/TYPE	PRICE
COMPUTER	HP	25000
PRINTER	JET PRINTER	10000
DATABASES	MS ACCESS	AVAILABLE
FLASHDISK	16GB	2000
DOCUMENATION SOFTWARE	VB.NET	AVAILABLE

PROJECT SCHEDULE

DESCRIPTION	DATE	DAY	TIME
PROJECT PROPOSAL	22/3/2020	MONDAY	3HOURS
SRS	25/3/2020	WEDNESDAY	1DAY
CODING	4/4/2020	WENESDAY/THURSDAY	2DAYS
TEST RESULTS	27/4/2020	FRIDAY	2HOURS
PRESENTATION	30/4/2020	Thursday	

SECTION TWO: SYSTEM REQUIREMENTS SPECIFICATION

SYSTEM REQUIREMENTS SPECIFICATION (SRS)

INTRODUCTION

This chapter contains the requirements specification and identification and definition for the project software application. The purpose of the software to be developed is its necessity for successful completion of the project and system quality upon completion. It will act as a template for the proper design of the application, the description contains the description of how the user will register new students and staff, update their records and delete them from the school when they leave.

PURPOSE

INTENDED PURPOSE/audiences

The intended audiences of GROUP 1 SCHOOL MANAGEMENT SYSTEM

- ✓ Developers- This document will be used by the developers in order to ensure they are developing the right project that meets the requirements provided by the SRS document
- ✓ Users -the document is referenced by the users in order to get familiar with the project idea.
- ✓ Testers- this is in order to have exact features and functions that will respond according to requirements and provided diagrams.
- ✓ Documentation writers- the document is used by the decision makers to know what features and in which way they have to explain so that the system will be understood by the users.
- ✓ End users use the system in order to know the exactly what they expect from the system.

SCOPE

SR document is basis on which proposed system referred to as inventory for LEEN'S MANAGEMENT SYSTEM. It covers the investigation techniques used, data collected and the analysis of the data collected. The main objective of this project is to develop secure software for effective management of GROUP 1 SCHOOL MANAGEMENT.

OVERALL DESCRIPTION

PRODUCT PERSPECTIVE

Proposed system will interrogateand computerize areas in the inventory management system including the departments with features such as , staff details, student details, academic results, and events scheduler, also added features also contain adding, deleting, saving and updating of records.

OPERATING ENVIRONMENT

The system is designed to operate on windows platform10, the computer should have a RAM capacity of at least 1GB or higher and hard capacity of 100 GB and higher. System requirements vb.net, printer and antivirus and also MS access to connect the databases.

DESIGN AND IMPLEMENTATION COSTRAINTS

The school will phase the following constraints;

- ✓ Hardware limitations - high costs of data conversions, which is converting all hardcopy records to softcopy and store in databases.
- ✓ Regulatory policies in terms of password for him/her to access the system.

USER DOCUMENATION

It will consist of several components usually expected of modern system including, tutorial, help pages and user manual which will contain sufficient information and instructions required to access and use the data system.

SYSTEM FEATURES

USER AUTHENICATION

User/admin authentication

Description and priority

It is where the users/admin who try to login into the system are evaluated whether they are ones whom they claim to be.

Responses:

User/admin activates the username entry and types his/her username and the system activates for user/admin password entry field to enter the password after he/she enters the password, press login button in the form and if the credentials are true the user/admin is allowed to enter into the system.

Functions requirements:

The software should provide a form which is described entry fields for username and passwords and checkboxes in which the user or admin must check-in, an interaction buttons in which the user/admin activates the system to check the validity of his credentials, and if the credentials are true the he/she will be allowed to the system else if they are not correct the a dialogue message box will be displayed try again.

The clear button will clear the textboxes if the user/admin mistakenly enters long password.

DATA BACKUP

The system should automatically perform a backup operation within the specified time, this should to ensure data is not lost in case the system of failure.

If the user tries to terminate e.g. deleting the record in databases or logging out before backing up then a warning message is sent to him or her to backup.

DATA ENTRY

The system should allow users to enter new records in databases and retrieve the required data in demand .

The system should provide empty spaces in form for the activity the user wants to perform bases on the details to activity he/she intends to accomplish. Then the user/admin presses the enter button for the system to process.

Form with appropriate entry fields for entering data

Appropriate interactions for each use case within e.g. buttons, textboxes

Validation mechanisms.

DATABASES ACCESS

This feature is designed to assist the system users to access/retrieve the information stored in the database. It will also provide access to earlier transactions which involve students who have been issued with books and titles of books in the library.

Stimulus/Response Sequences

User chooses which information to request the system for them system searches and validates the query and generates search details.

Functional Requirements

- ✓ The user shall be prompted to enter search details.
- ✓ The system shall validate and search the database for requested information.
- ✓ The system will provide a means of retrieving requested information from its database.
- ✓ The system shall generate search results.

DATA ANALYSIS

Description and Priority

The system would provide a data analysis tool. This would be for analysis of existing data for professional reporting and faster decision making. Analysis would use charts (such as pie charts, bar charts etc.)

Stimulus/Response Sequences

This system would activate the analysis interaction provided by the system and specify the data range within which data should be analyzed.

Functional Requirements

- ✓ An interface where analysis report would be displayed.
- ✓ A form level analysis interaction.
- ✓ Code for generating the analysis.

EXTERNAL INTERFACES REQUIREMENTS

USER INTERFACES

In order the user to interact with the system. This would help the user to master the functionality of the system of the system quickly and have the optimal performances within the system. This includes the checkboxes, buttons, textboxes, labels, combo boxes

HARDWARE INTERFACES

The system would use basic hardware components of the system, keyboard and mouse for inputting data, printer for hard copy outputs, processing to be done by processor and storage to be done automatically.

SOFTWARE INTERFACES

Windows 10 pro

NONFUNCTIONAL REQUIREMENTS

Performances requirements

Security performances in terms of password for authentication

SECTION THREE: SYSTEM DESIGN SPECIFICATION

INTRODUCTION

This phase describes desired features and operation of the proposed system in detail including data flow diagrams, entity relationship diagram, context diagram, flowcharts, process diagrams and documentation.

SCOPE

The scope of this document is as far as the design of the proposed system is concerned and is not concerned with the system which is to be interfaced with the system such as the personnel system. Therefore, the system is concerned only with the operations of the school.

OBJECTIVE

To come up with the most optimal design this will be used to guide the construction of the proposed system in the successive stage. The system will provide the services of the registering students, new staffs, records of books in the library and finance issues.

SYSTEM OVERVIEW

System will provide the following services

- ✓ User authentication a way to log in and out of the system.
- ✓ Add, save, update, and delete new students /staff .
- ✓ Safeguard data integrity.

DESIGN CONSIDERATIONS

Should operate in a window 10 OS, user should be able to identify their input in the proposed time and the system should reflect the requirements that they requested.

SYSTEM ARCHETICTURE

COMPONENT	PURPOSE
USER/ADMIN VERIFICATION	It verifies who is trying to access into the system
Task addition	allow the user/admin to add a particular thing into the system
Task updating	It is where a particular thing is updated and saved into the system
Task deletion	It is where only the admin can delete a particular record where the user cannot

LOGICAL DESIGN

CONTEXT DIAGRAM

Login details

Use case name	login	View menu
precondition	Actor must enter the password	The information must be in advance
Post condition	Actor is allowed to submit the request	Information details are displayed

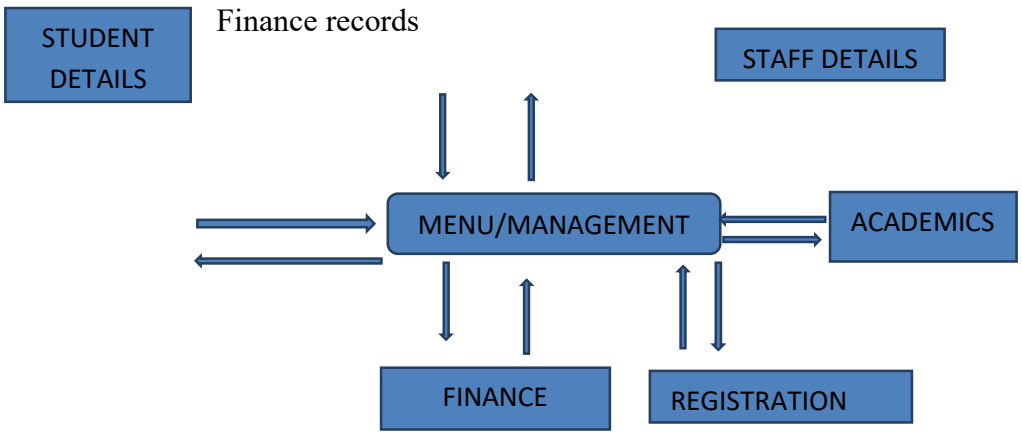
Staff details

Student details

Academic details

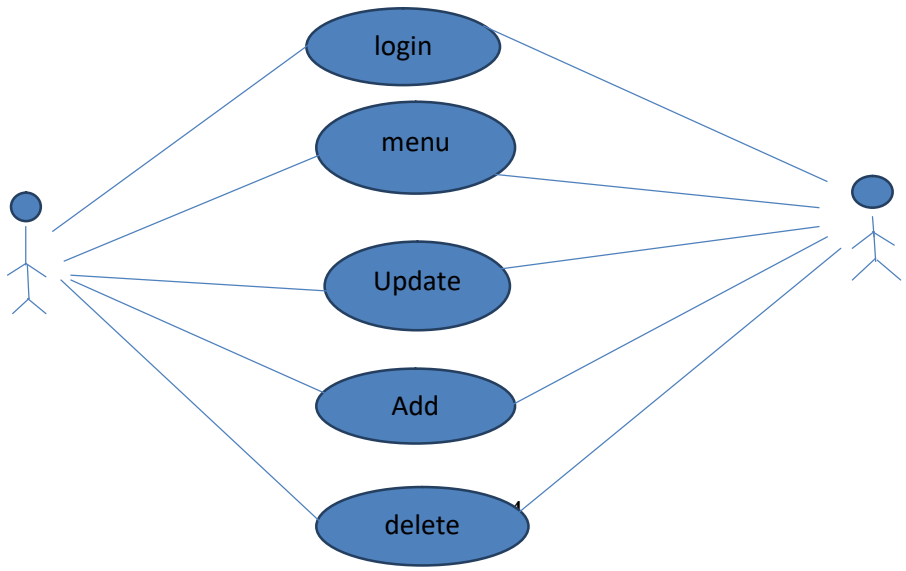
Registration details

Academic details



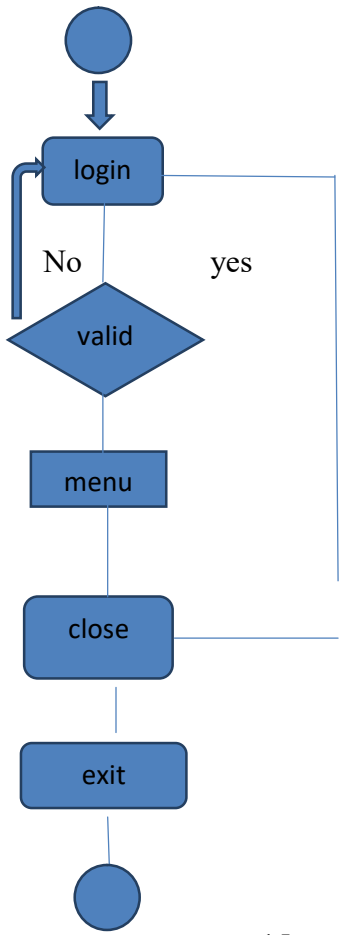
USECASE DIAGRAM

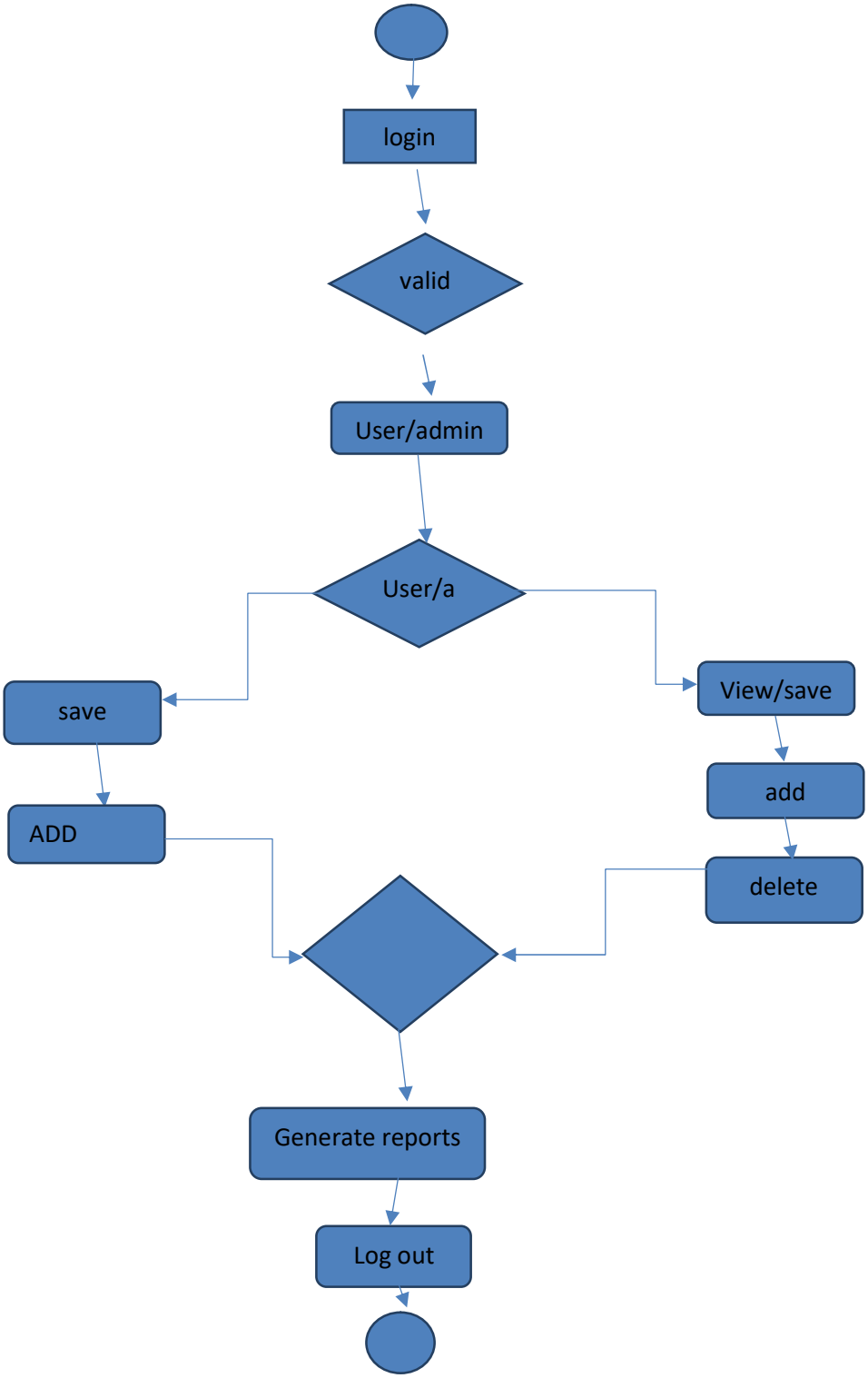
Use case represents the functionality of the system. It focuses on the behavior of the system and external points of view where the actors are outside the boundary of the system, where areas use case are inside the boundary.



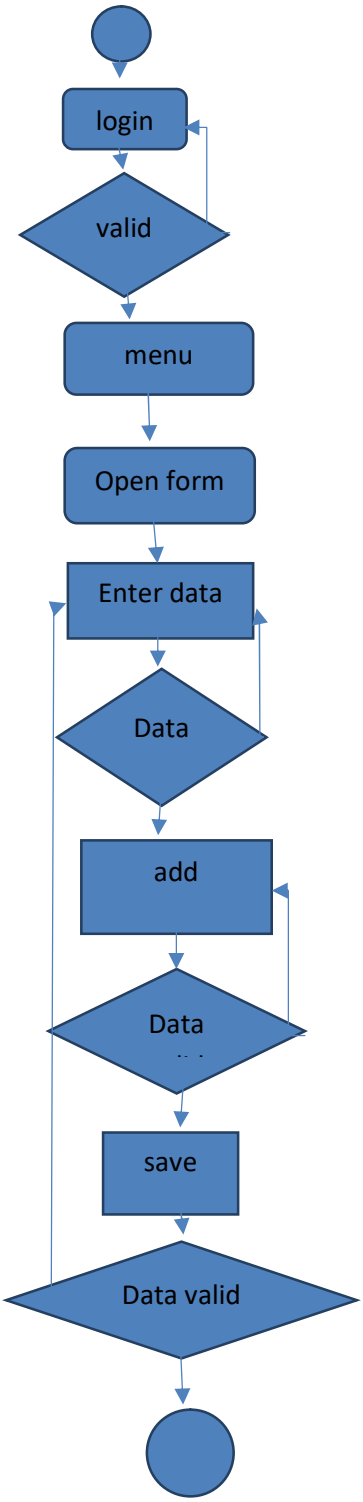
actors	User/admin	User/admin
actions	Actor submits a request; actor make a request he/she is authorized that is after checking in he/she must the appropriate password for him/her to login NB: actor who are not privileged cannot view the data base or login	Actor enters the view criteria for the information

ACTIVITY DIAGRAMS

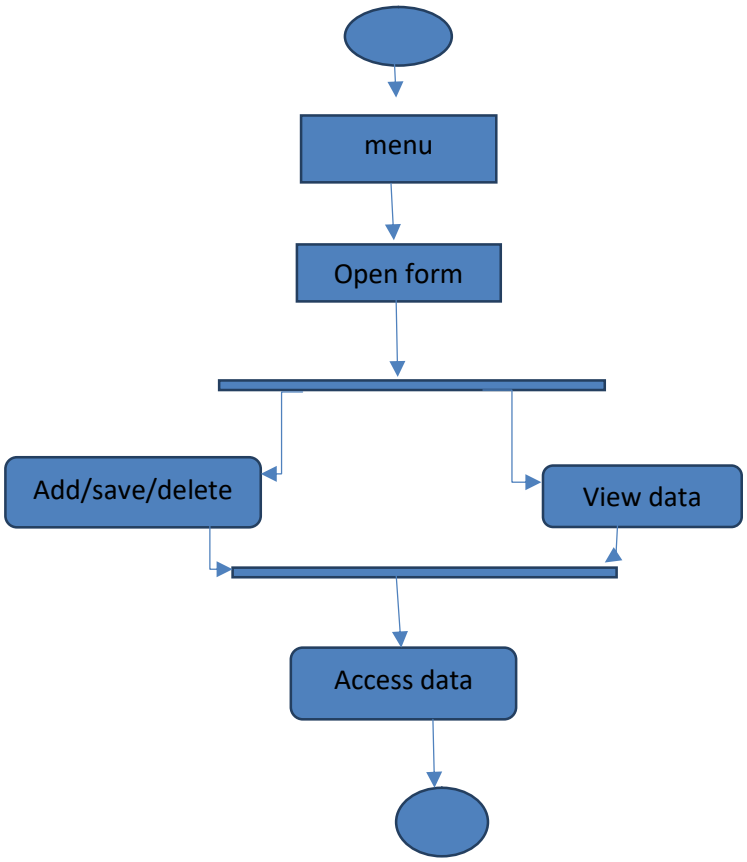



login
valid
User/admin
User/a
save
ADD
View/save
add
delete
Generate reports
Log out

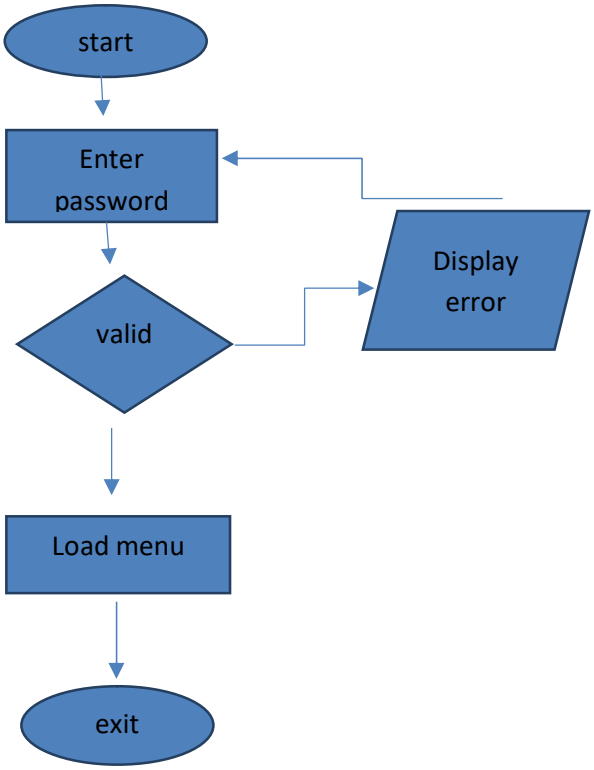
Databases update/save/delete



Menu



FLOWCHARTS



DATABASES DESIGN

Entities and attribute identification

Staff details

Student details

Books details

Medicine details

Finance details

ATTRIBUTES

STAFF DETAILS: STAFF ID, NAME, AGE, ADRESS, GENDER, STATUS, EMAIL, JOINING DATE,SALARY

STUDENT DETAILS:STUDENT ID, NAME,AGE, ADRESS, GENDER, STATUS, EMAIL, DATE ENROLLED , RESULTS

MEDICINE DETAILS: MEDICINE ID, MEDICINE NAME, CATEGORY, PRICE, MANUFACTURE DATE, EXPIRY DATE, STUDENT TREATED WITH IT

FINANCE DETAILS: STUDENT NAME, STUDENT ID, CONTACT, PAYMENT DATE/ TRANSACTION,AMOUNT

BOOK DETAILS: BOOOK TITLE,THE PUBLISHER ,AUTHOR,CATEGORY,YEAR OF PUBLICATION,BOOK ID

Database design schema

It easily implements using the relational data structures, a relational data structure consists of one or more two dimensional databases referred to as relation.

Table Name	Field Name	Key	type	description	Required
Staff details	STAFF_ ID	Primary key	Number	Unique identifies the staff	Yes
	STAFF_ Name		Text	Identifies the staff name	Yes
	AGE		number	Is the age of the staff	yes
	ADDRESS		Text	Where he/she is located	Yes
	STATUS		TEXT	Whether he/she is married or not	Yes
	GENDER		text	The equality either male/female	no
	EMAIL		text		yes

	JOINING DATE		Date/time	When he/she start working in the company	yes
	SALARY		currency	What he/she is paid at the end of month	yes
Table Name	Field Name	Key	type		Required
Student details	student_ ID	Primary key	Number	Uniquely identifies the student who was registered	Yes
	student_ Name		Text	Identifies the name of the student	Yes
	AGE		number	The age of the customer/patient	yes
	ADDRESS		Text	Where he /she is located	Yes
	STATUS		TEXT	Married or not	Yes
	GENDER		text	Whether is he or she	no
	EMAIL		text		yes
	Total price		currency	Captures the total transaction done	no
	Date checked		Date/time	The date when he or she was treated	yes

Table Name	Field Name	Key	Length	Required	description
MEDICINE DETAILS	Medicine ID	Primary key	Number	Yes	Uniquely captures the medicine

	MEDICINE NAME		text	Yes	Captures the name of the medicine
	CATEGORY		Date	Yes	Captures the group in which the medicine is grouped
	EXPIRY DATE		Number	Yes	The date of which the medicine is expected to be no use
	MANUFACTURE DATE		Time	No	The date in which the medicine was manufactured
	Student given		Number	No	The person issued the medicine
	Date issued		Date	No	Date given to the student
	SUPPLIER ID	Foreign key	NUMBER	YES	Is the person who supplied the medicine and uniquely identifies the supplier

Table Name	Field Name	Key	Length	description	Required
BOOK DETAILS	BOOK ID	Primary key	Number	Uniquely identifies the book	Yes

	BOOK NAME		TEXT	Captures the book name	Yes
	AUTHOR		TEXT	The writer of the book	Yes
	DATE OF ISSUE		DATE	When the book was issued	Yes
	PUBLISHER		TEXT	The one who published the book	No
	YEAR OF PUBLICATION		NUMBER	The year the book was published	No
	CATEGORY		TEXT	The category in which the book is in.	No

Table Name	Field Name	Key	Length	description	Required
FINANCE DETAILS	STUDENT NAME			The name of the student	Yes
	Student_ ID	Foreign key	Number	The student who made a transaction	Yes
	Amount paid		Text	The amount paid by the student	Yes
	Date paid		date	The date in which the fees was paid	yes

E.R.D

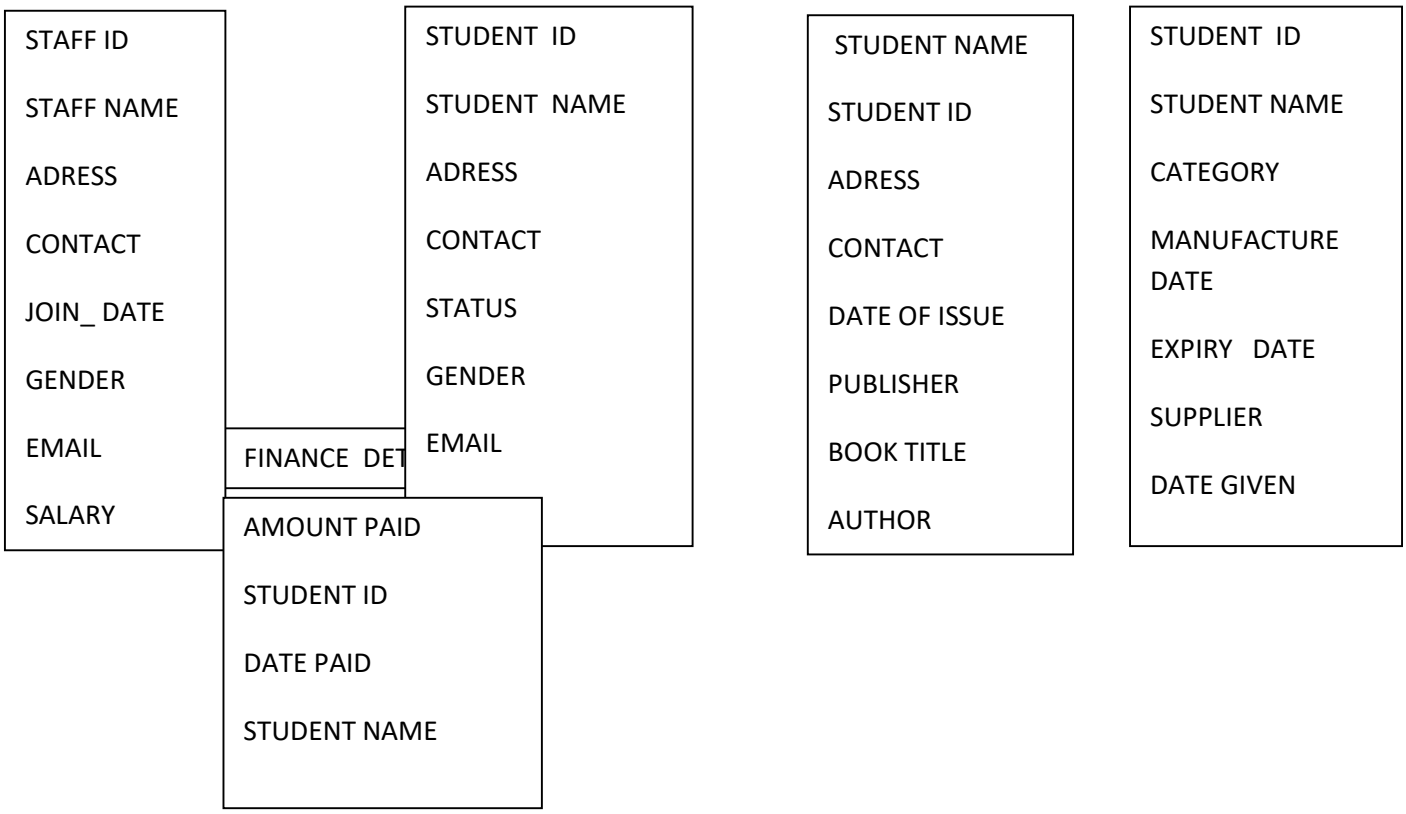
It is a technique of organizing and documenting a system's data. An entity model represents the relationship between different entities or fields in a system while entity relationship involves drawing of the system entities and showing how they relate to each other.

STAFF DETAILS

STUDENT DETAILS

LIBRARY DETAILS

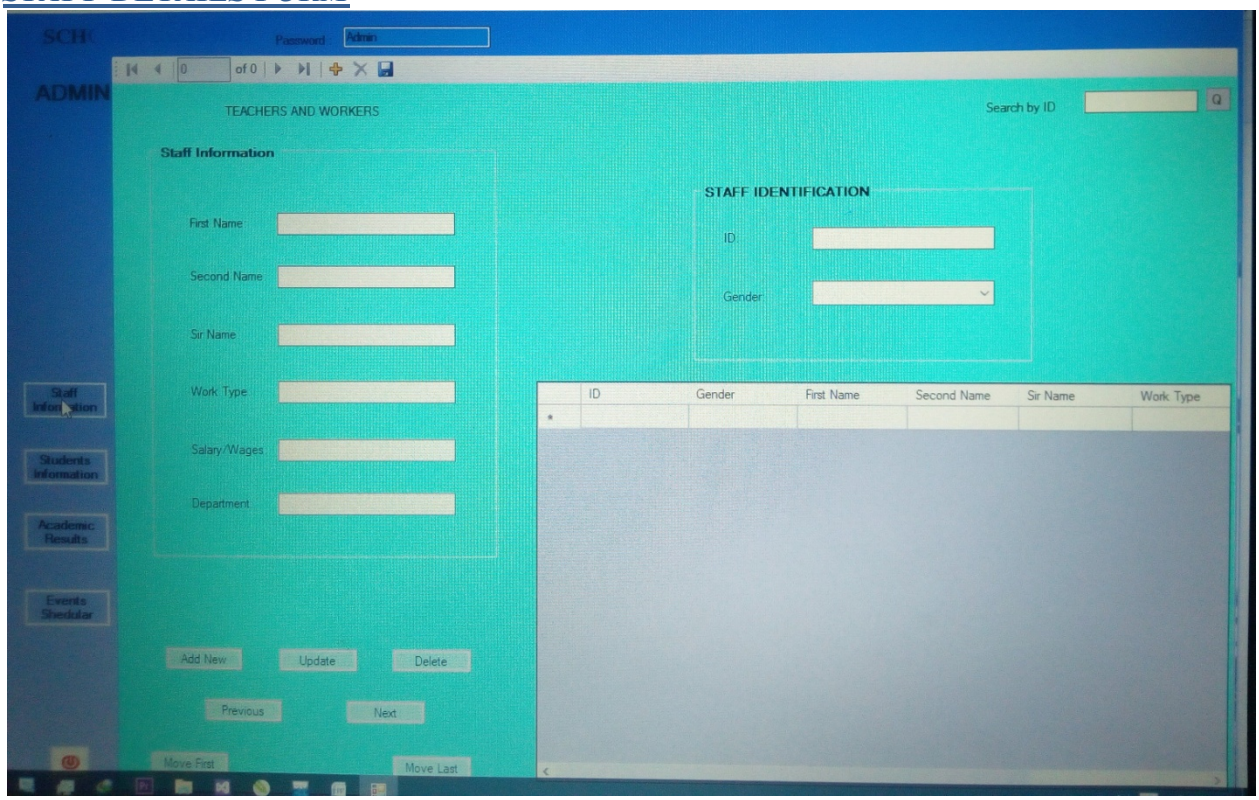
MEDICINE DETAILS



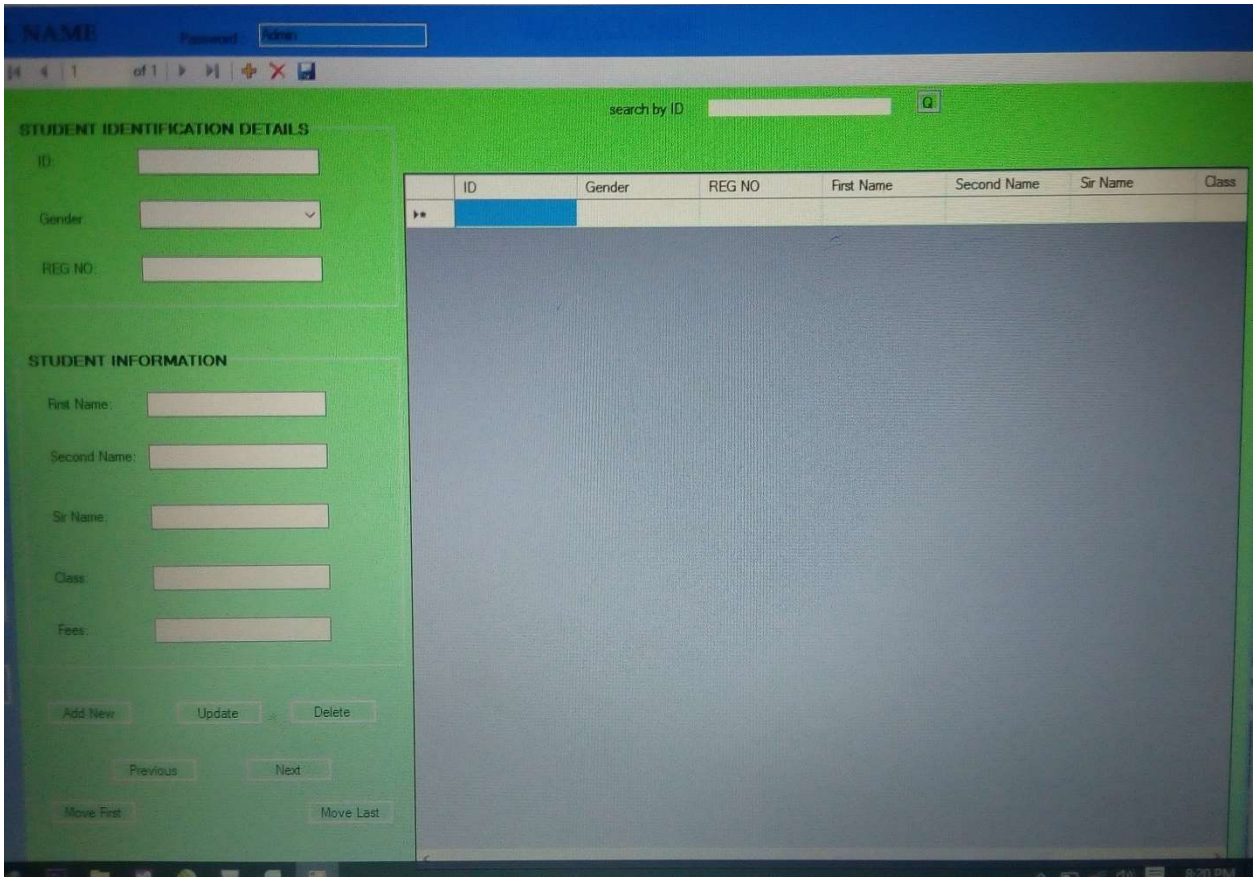
INPUT /OUTPUT SCREEN DESIGN

This is designed to allow the user to add information freely and easily and be able to switch between a number of forms. The forms below are used for displaying both input and output.

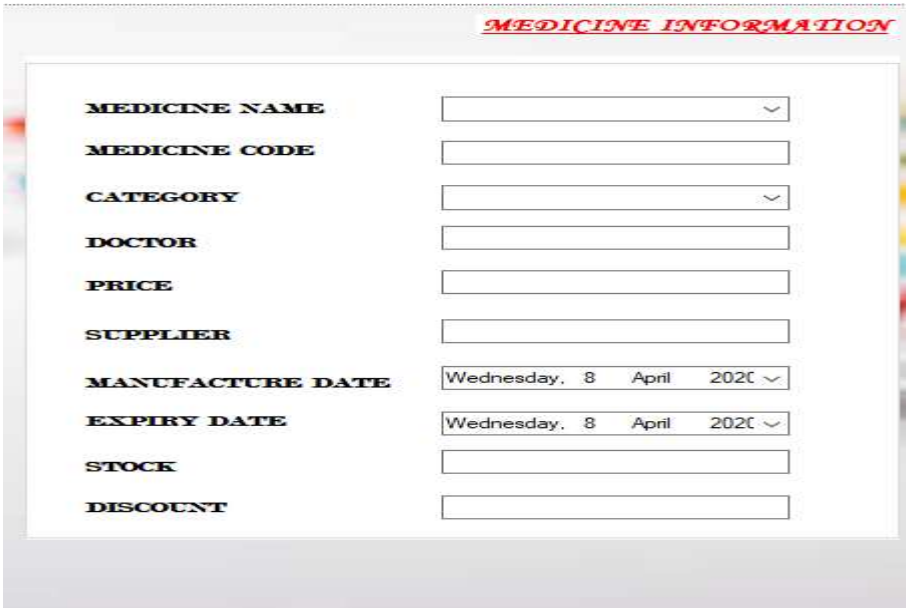
STAFF DETAILS FORM



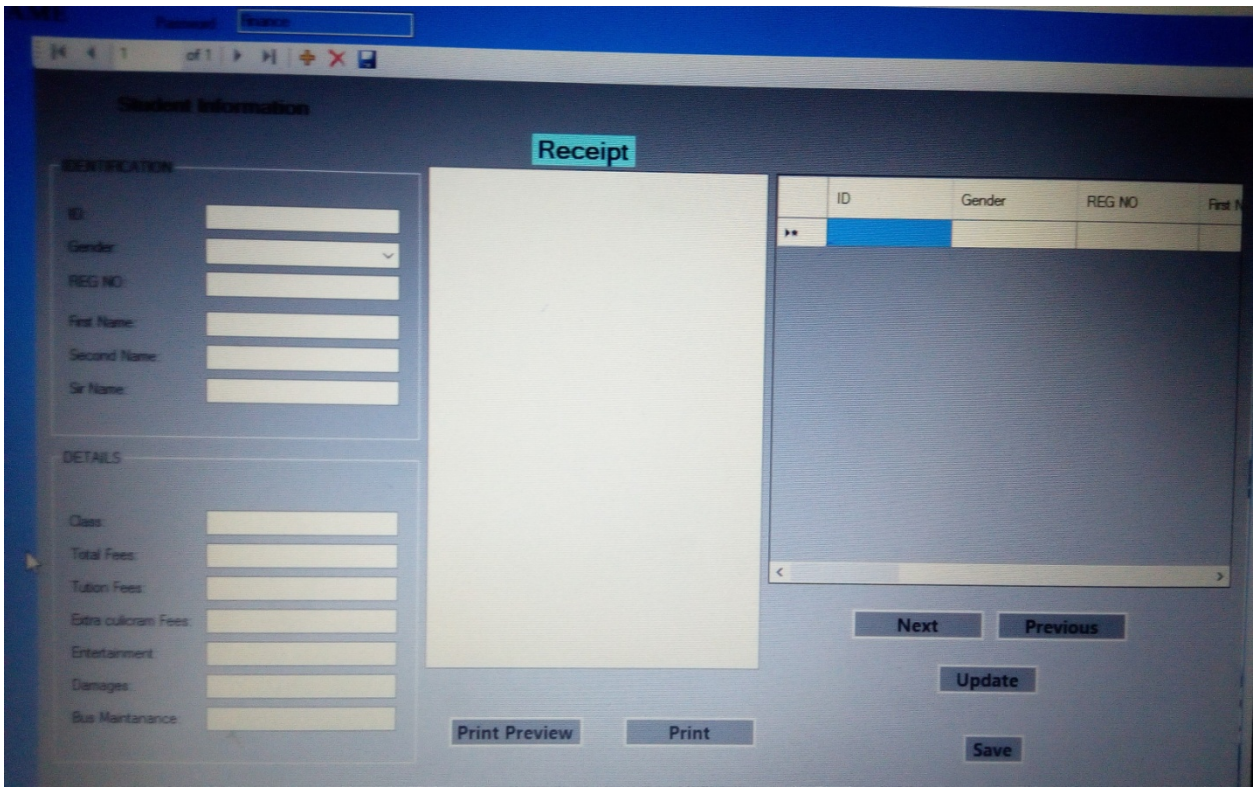
STUDENT DETAILS



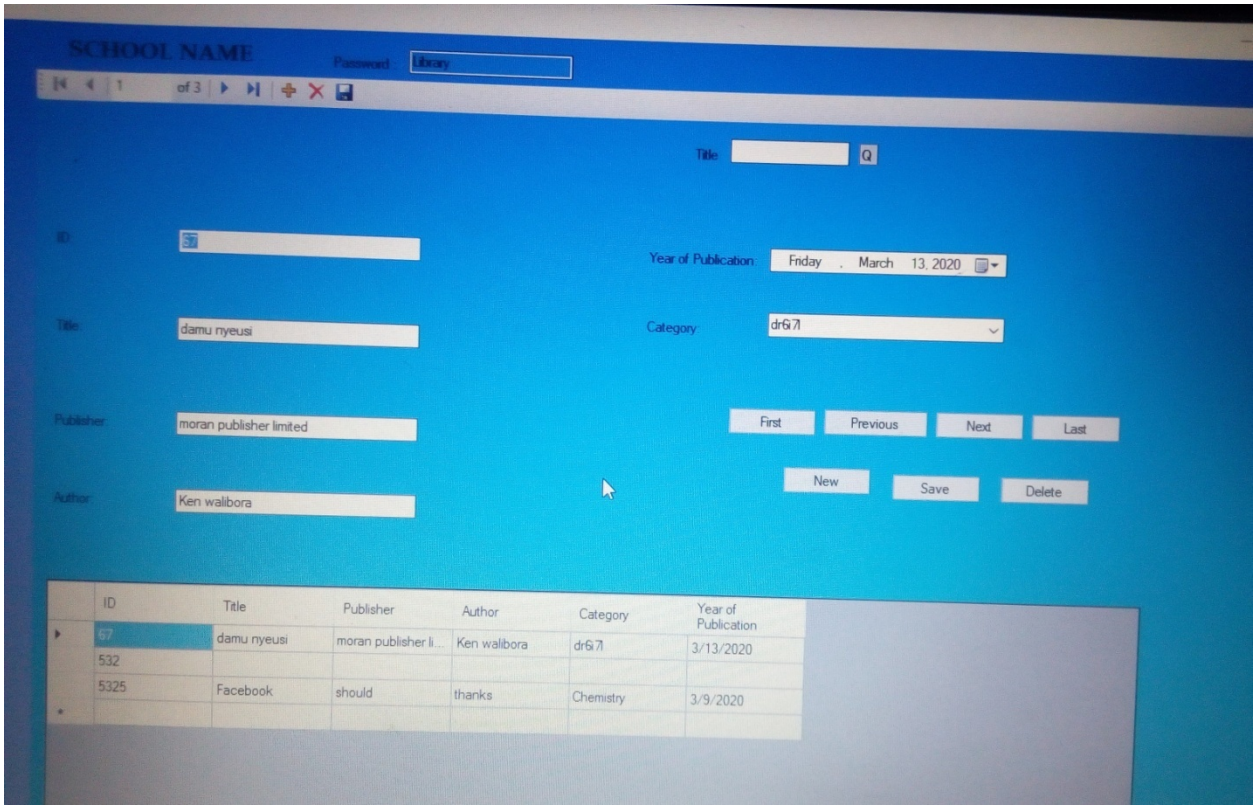
MEDICINE DETAILS



FINANCE DETAILS



LIBRARY /BOOKS DETAILS



SECTION FOUR:TEST PLAN SPECIFICATION

INTRODUCTION

This chapter outlines the tests that were performed on the GROUP 1 school management system and the evaluation of the system based on the inputs to the system and results obtained.

OBJECTIVES

- ✓ To ensure that the system program is free from errors
- ✓ To ensure that the group has adequate documentation and instructions to operate the system.
- ✓ To guarantee that the end users can successfully interact with the system.
- ✓ To find out whether the components of the system interfaces are working correctly.
- ✓ Test data should deliberately include incorrect data in order to test the validation and control procedures

STATEMENT SCOPE

The system will be tested on variety features such as:

- ✓ Authentication of the users attempting to log on to the system
- ✓ Ability of the system to search, delete, save and update records stored in the database.
- ✓ Ability if the system to validate entity integrity.

MAJOR COSTRAINTS

- ✓ Time for completion .
- ✓ Limited budget
- ✓ Software only runs in windows application 10

TEST PLAN

Test planning is also the practice of coming up with strategies, schedule and models to be used to test the software and products.

Testing will be carried out on various parts of the system using different testing methods and the results will be analyzed and evaluated. Test plan outlines the types of tests, the expected outcome, and comments on the actual test results

Test number	**test area**	**Purpose**	**Expected results**	**Comments**
1.	interface	Degree to which the user interacts with the system	Easy to interact with	The degree of interactivity with the system should be high
2.	Login system	Security purposes	Access granted if the user name and password are correct	Only authorized users can log in
3.	Main system	System navigation	Navigate with ease	The menu option should be easy to use
4.	code	System operation	No errors when this system runs	Coding should be in consisted with the

				design
5.	validation	System accuracy	Only valid entries are in the system	Reject invalid entries.

TESTING STATEGY

This involves checking if the school system meets and conforms to system requirement specifications checking for bugs and virus in the system.

UNIT TESTING

This test individual component of the system to show how they interoperate to enhance system for performance and correctness. An example of such components includes the following: text boxes, labels, command button and other dynamic fields and tables are validated so as to generate the required output.

User name- the fields will have to prompt a username, which gives the user rights.

Password-the field will only accept the valid password. It will not just accept any other texts.

Numerical fields- it will only accept numerical values.

INTEGRATION TESTING

This is an orderly progression of testing in which software elements, hardware elements or both are combined and tested until the entire system has been integration.

VALIDATION TESTING

This section includes a discussion of the order of validation by software function. First of all, the school management system is started, it prompts the user to enter the user name and password which upon acceptance allows the user access to the whole system.

TEST SCHEDULE

Type of testing	No. of Hrs.	Start date	Completion date
Unit Testing	3	27/3/2020	28/3/2020
Integration Testing	2	28/3/2020	29/3/2020
Validation Testing	1	29/3/2020	29/3/2020

TEST PROCEDURE FOR INTEGRATION
Integration testing can be executed in two ways:

TOP DOWN INTEGRATION
This is an incremental approach where modules are integrated by moving downwards through the hierarchy. The top-down integration strategy verifies major control or decision points early in the test process, which aids in the identification of major system errors or omissions early. This method also allows the early demonstration of system functionality, which is a confidence builder for both the developer and the customer.

BOTTOM UP INTEGRATION
This technique begins construction and testing with modules at the lowest level. The low-level modules are tested and then integrated into a cluster is tested. Next, different clusters that interface together are tested and this process continues until we reach the top module.

SECTON FIVE:IMPLEMENTATION
INTRODUCTION
The main purpose of this document is to clearly describe the activities and procedures for successful changeover from manual booking to the new newly developed school management system.

OUTCOME
The expected outcome implementation includes:

- ✓ Fast and reliable inventory capturing system.
- ✓ Minimal interruption of the daily activities of the school.
- ✓ Maximum resources utilization and minimal time wastage.

IMPLEMENTATION METHODS
There are four possible implementation methods.

Parallel Running
The old and new systems run simultaneously for an agreed period of time and results from the two systems are compared. Once the user has complete confidence in the new system, the old system is abandoned.

Direct Changeover
Implement the new system completely and withdraw without any sort of parallel running at all. It demands thorough testing and well – planned file creation and training strategies

PROPOSED IMPLEMENTATION METHODS
It will be better to use direct changeover method of implementation. This is because of security because one can fall back to the old system should the new system bring problems or fail.

IMPLEMENTATION PROCESS

Acceptance testing

An acceptance testing will be carried out before implantation to ensure that the system runs acceptably in the user environment and that the system met its user requirements this test will be carried out in the users' platform.

Data conversion

It includes movement of data held in the files to the hardware through the data entry of the system in other words conversion of data in hard copy into soft copy.

Execution

It involves the actual operation or running of the new system.

Review

System developer and the agency will analyze and compare the existing system with the new system.

IMPLEMENATION RESOURCES

Hardware

- ✓ PC
- ✓ 3.0GHZ processor speed
- ✓ Hard disk at least 256MB
- ✓ 512MB RAM
- ✓ Hp Printer

Software Requirement

- ✓ Windows 10 operating system
- ✓ Microsoft visual studio 6.0 for running the system
- ✓ Microsoft word 2007 for documentation
- ✓ Microsoft access 2007 for database creation
- ✓ Antivirus software to get rid of viruses

User training

This involves training the users and other relevant stakeholder on how to use and operate the new system efficiently and effectively. It will be done so that the system users will know how the system runs. This will enable them to be user friendly with the system.

Methods of training

- ✓ User manuals: Users will be provided with user manual of the new system as part of the system documentation that is well explained and illustrated on how best the system can best used. This will be helpful in case of difficult using the system.
- ✓ On job training: Users will be given some tasks to perform on their own following given instruction. In case of any mistake or error done or difficulties encountered they will be corrected and guided on how to avoid them. This will be cheap and faster method of training.

PROBLEMS ENCOUNTERED

There was a time constraint because it took a lot of time to install and setup the new system.

There was also a big difficulty at first explaining to the system end users on how to go about running the new system.

CONCLUSION

Locally developed software is still not widely embraced by many organizations in Kenya but by the look of current statistics, they are getting a hang on it. Emphasis on use of locally developed software in organizations should be put across to be able to achieve vision 2030 vision.

SECTION SIX

USER MANUAL

GENERAL INFORMATION

The aim of this guide is to help users operate the software productively with minimum reading. It will also be used to guide the users while using the program. It is an important tool in trouble shooting when working with the system and when installing the program. The LEEN'S school management system is easy to install and implement. This manual will show the general interface of the system, how it works to meet the needs of the user and how the various operations are carried out using the system.

SYSTEM OVERVIEW

The LEEN'S school management system is system that is used to keep all the records of the student and staff. The system starts with the login form then finally it loads the whole system after user/admin authentication.

PROJECT PREFERNCES

- ✓ Piet Westerndorp, Carel Jansen (1996) Interface Design& Document Design

AUTHORIZED PERMISSION

The only authorized users permitted onto the school management system are the management and the personnel who should operate the system only. Apart from that no users are permitted onto the system.

USER/ADMIN ACCESS LEVELS

The system will only have a user or admin access level where it will be share between the management and the personnel responsible for operation of the system.

ORGANIZATION OF MANUAL

The system manual is organized in such a way that it starts form the beginning that is the system installation to the bottom details. This is called the top-down approach.

GETTING STARTED

This section provides a general walkthrough of the system from initiation of the system throughout to exit it

LOG IN

This screen will be used to verify users/admin and grant access to the system .Only valid users /admin with a valid username and password are allowed to login to the system. If you enter the details incorrectly, then the system will not grant you access to the system.

Codes for the login

```
PublicClassForm1
Dim form AsNewForm2



PrivateSub Button1_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)

End Sub

PrivateSub Button2_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)


End Sub

PrivateSub Button3_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
```

```
End Sub

PrivateSub CheckBox1_CheckedChanged(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles CheckBox1.CheckedChanged
If CheckBox1.Checked = TrueThen
        TextBox2.UseSystemPasswordChar = False
Else
        TextBox2.UseSystemPasswordChar = True
EndIf
End Sub

PrivateSub Label1_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Label1.Click

End Sub

PrivateSub Label2_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Label2.Click

End Sub

PrivateSub TextBox1_TextChanged(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs)

End Sub

PrivateSub TextBox2_TextChanged(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs)

End Sub

PrivateSub LinkLabel1_LinkClicked(ByVal sender AsSystem.Object, ByVal e
AsSystem.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked

End Sub

PrivateSub BunifuFlatButton1_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton1.Click


If CheckBox2.Checked Then
Dim USERNAME AsString
Dim PASSWORD As Integer
        USERNAME = TextBox1.Text
        PASSWORD = TextBox2.Text
If (USERNAME = "Admin"And PASSWORD = "2001") Then
```

```
MessageBox.Show("login successful")

Form2.Show()
Me.Hide()



Else
MessageBox.Show("Try again")

EndIf

ElseIf CheckBox3.Checked Then
Dim USERNAME AsString
Dim PASSWORD As Integer
        USERNAME = TextBox1.Text
        PASSWORD = TextBox2.Text
If (USERNAME = "user"And PASSWORD = "1234") Then
MessageBox.Show("login successful")

Dim form AsForm2
            form = Form2

form.BunifuFlatButton1.Enabled = False
form.BunifuFlatButton4.Enabled = False
form.BunifuFlatButton5.Enabled = False
DimformuAsForm4
DimformmAsForm5
Dim tuna AsForm8
            tuna = Form8


formu = Form4
formu = Form5
formu.Button6.Enabled = False
formu.Button6.Enabled = False
tuna.Button7.Enabled = False


Form2.Show()
Me.Hide()
```

```
Else
MessageBox.Show("Try again")

EndIf


EndIf

End Sub

PrivateSub BunifuFlatButton2_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton2.Click
    TextBox1.Clear()
    TextBox2.Clear()


End Sub

PrivateSub BunifuCheckbox1_OnChange(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs)

End Sub

PrivateSub BunifuFlatButton3_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton3.Click
Application.Exit()


End Sub

PrivateSub TextBox1_TextChanged_1(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles TextBox1.TextChanged

End Sub
```

```
PrivateSub BunifuPictureBox1_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuPictureBox1.Click

End Sub

PrivateSub CheckBox2_CheckedChanged(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles CheckBox2.CheckedChanged
If CheckBox2.Checked = TrueThen
        CheckBox3.Checked = False
ElseIf CheckBox3.Checked = TrueThen
        CheckBox2.Checked = False

EndIf
End Sub

PrivateSub CheckBox3_CheckedChanged(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles CheckBox3.CheckedChanged
If CheckBox3.Checked = TrueThen
        CheckBox2.Checked = False
ElseIf CheckBox2.Checked = TrueThen
        CheckBox3.Checked = False

EndIf
End Sub

PrivateSub TextBox2_TextChanged_1(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles TextBox2.TextChanged

End Sub

PrivateSub Form1_Load(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
HandlesMyBase.Load

EndSub
EndClass
```

MENU

After logging in one is able to access the menu which contains several buttons.

```
PublicClassForm2

PrivateSub Button4_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
```

```
EndSub

PrivateSub Button6_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)

EndSub

PrivateSub Button7_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)

EndSub

PrivateSub Button1_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)

EndSub

PrivateSub Button2_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)


EndSub

PrivateSub Button3_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)

EndSub

PrivateSub Button4_Click_1(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)

EndSub

PrivateSub Button5_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)

EndSub

PrivateSub Button8_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)

EndSub

PrivateSub BunifuFlatButton1_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton1.Click
Form3.Show()
Me.Hide()
EndSub
```

```
PrivateSub BunifuFlatButton2_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton2.Click
Form4.Show()
Me.Hide()
EndSub

PrivateSub BunifuFlatButton3_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton3.Click
Form5.Show()
Me.Hide()
EndSub

PrivateSub BunifuFlatButton4_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton4.Click
Form6.Show()
Me.Hide()
EndSub

PrivateSub BunifuFlatButton5_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton5.Click
Form7.Show()
Me.Hide()
EndSub

PrivateSub BunifuFlatButton6_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton6.Click
Form8.Show()
Me.Hide()

EndSub

PrivateSub BunifuFlatButton7_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton7.Click
Form1.Show()
Me.Hide()
EndSub

PrivateSub BunifuFlatButton8_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton8.Click
IfMsgBox("Are you sure you want to quit?", MsgBoxStyle.YesNo) = MsgBoxResult.YesThen
Application.Exit()
EndIf
EndSub
```

```
PrivateSub BunifuPictureBox1_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs)


EndSub
EndClass
```

STAFF DETAILS

```
PublicClassForm2

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
With Form12
            .TopLevel = False
            Panel1.Controls.Add(Form12)
            .BringToFront()
            .Show()
EndWith
EndSub

PrivateSub STUDENTSBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)


EndSub


PrivateSub STUDENTSBindingNavigatorSaveItem_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs)


EndSub


PrivateSub STUDENTSBindingNavigatorSaveItem_Click_2(ByVal sender As System.Object, ByVal
e As System.EventArgs)

EndSub


PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
With Form4
            .TopLevel = False
            Panel1.Controls.Add(Form4)
            .BringToFront()
            .Show()
EndWith
EndSub
```

```
PrivateSub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
With Form5
            .TopLevel = False
            Panel1.Controls.Add(Form5)
            .BringToFront()
            .Show()
EndWith
EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click

EndSub

PrivateSub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
Form1.Show()
Me.Close()

EndSub

PrivateSub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Me.Close()
EndSub

PrivateSub Panel1_Paint(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs)

EndSub

PrivateSub Panel1_Paint_1(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Panel1.Paint

EndSub
EndClass
```

STUDENT DETAILS

```
PublicClassForm2

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
With Form12
            .TopLevel = False
            Panel1.Controls.Add(Form12)
            .BringToFront()
            .Show()
EndWith
EndSub

PrivateSub STUDENTSBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)


EndSub
```

```
PrivateSub STUDENTSBindingNavigatorSaveItem_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs)


EndSub


PrivateSub STUDENTSBindingNavigatorSaveItem_Click_2(ByVal sender As System.Object, ByVal
e As System.EventArgs)

EndSub


PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
With Form4
            .TopLevel = False
            Panel1.Controls.Add(Form4)
            .BringToFront()
            .Show()
EndWith
EndSub


PrivateSub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
With Form5
            .TopLevel = False
            Panel1.Controls.Add(Form5)
            .BringToFront()
            .Show()
EndWith
EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click

EndSub

PrivateSub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
Form1.Show()
Me.Close()

EndSub

PrivateSub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Me.Close()
EndSub

PrivateSub Panel1_Paint(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs)

EndSub

PrivateSub Panel1_Paint_1(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Panel1.Paint
```

```
EndSub
EndClass
```

FINANCE DETAILS

```
PublicClassForm8

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
With Form9
            .TopLevel = False
            Panel1.Controls.Add(Form9)
            .BringToFront()
            .Show()
EndWith
EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
With Form7
            .TopLevel = False
            Panel1.Controls.Add(Form7)
            .BringToFront()
            .Show()
EndWith

EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
Me.Close()

EndSub
EndClass
```

LIBRARY DETAILS

```
PublicClassForm7

PrivateSub Label12_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Label12.Click

EndSub

PrivateSub DateTimePicker2_ValueChanged(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles DateTimePicker2.ValueChanged

EndSub

PrivateSub GroupBox2_Enter(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles GroupBox2.Enter
```

```
EndSub

PrivateSub Button6_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button6.Click
IfMsgBox("Are you sure you want to CLOSE the program?", MsgBoxStyle.YesNo) =
MsgBoxResult.YesThen
Form2.Show()
Me.Hide()
EndIf
EndSub

PrivateSub Button5_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button5.Click
IfMessageBox.Show("Do you really want to delete the DATA?", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Warning) =
Windows.Forms.DialogResult.YesThen
BOOKBindingSource.RemoveCurrent()


EndIf


EndSub

PrivateSub Form7_Load(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
HandlesMyBase.Load
'TODO: This line of code loads data into the 'BOOKDataSet.ISSUE' table. You can move, or
remove it, as needed.
Me.BOOKTableAdapter.Fill(Me.BOOKDataSet.issue)

EndSub

PrivateSub Button2_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button2.Click
bookBindingSource.AddNew()

EndSub

PrivateSub Button3_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button3.Click
bookBindingSource.MoveNext()

EndSub

PrivateSub Button4_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button4.Click
```

```
If DateTimePicker2.Text <= DateTimePicker3.Text Then
MessageBox.Show("year of publishing should not be equal OR greater to year of issue")


Else
BookBindingSource.EndEdit()
BookTableAdapter.Update(BOOK Dataset)
MessageBox.Show("YOUR INFO HAS BEEN SAVE SUCCESFUL")


EndIf



EndSub

PrivateSub Button1_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button1.Click
Dim quantity AsInteger
DimmedicinepriceAsInteger
Dim discount AsInteger
Dim vat AsInteger
        quantity = TextBox7.Text
medicine price = TextBox8.Text
        discount = TextBox9.Text
        vat = TextBox10.Text

        TextBox11.Text = ((quantity * medicineprice) - (discount + vat))

EndSub

PrivateSub TextBox12_TextChanged(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles TextBox12.TextChanged
        TextBox12.Text = TextBox11.Text

EndSub

PrivateSub TextBox11_TextChanged(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles TextBox11.TextChanged
        TextBox12.Text = TextBox11.Text
EndSub

PrivateSub Button7_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button7.Click
Dim search AsInteger = TextBox13.Text
```

```
Me.BOOKTableAdapter.FillBysearchBOOKNo(Me.BOOKataSet.ISSUE, search)

EndSub

PrivateSub RichTextBox1_TextChanged(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles RichTextBox1.TextChanged


EndSub

PrivateSub BunifuFlatButton1_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton1.Click
    RichTextBox1.AppendText("TOTAL PURCHASE CALCULATION" + vbNewLine)
    RichTextBox1.AppendText("-----------------------------------------------------" + vbNewLine)
    RichTextBox1.AppendText("PURCHASE NO=" + TextBox4.Text + vbNewLine)
    RichTextBox1.AppendText("MEDICINE CODE=" + TextBox5.Text + vbNewLine)
    RichTextBox1.AppendText("MEDICINE NAME=" + ComboBox1.Text + vbNewLine)
    RichTextBox1.AppendText("QUANTIY=" + TextBox7.Text + vbNewLine)
    RichTextBox1.AppendText("PRICE=" + TextBox8.Text + vbNewLine)
    RichTextBox1.AppendText("DISCOUNT=" + TextBox8.Text + vbNewLine)
    RichTextBox1.AppendText("VAT=" + TextBox8.Text + vbNewLine)
    RichTextBox1.AppendText("TOTAL AMOUNT=" + TextBox8.Text + vbNewLine)
    RichTextBox1.AppendText("-----------------------------------------------------" +
vbNewLine)
    RichTextBox1.AppendText("-----------------------thank you----------------------" +
vbNewLine)
EndSub

PrivateSub Button8_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)


EndSub

PrivateSub BunifuFlatButton2_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton2.Click
    RichTextBox1.Clear()
EndSub

PrivateSub BunifuFlatButton3_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handles BunifuFlatButton3.Click
    PrintPreviewDialog1.ShowDialog()


EndSub
```

```
PrivateSub PrintDocument1_PrintPage(ByVal sender AsSystem.Object, ByVal e
AsSystem.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
Dim font1 AsNewFont("arial", 16, FontStyle.Regular)
e.Graphics.DrawString(RichTextBox1.Text, font1, Brushes.Black, 100, 100)

EndSub
EndClass


MEDICINE DETAILS
PublicClassForm5

PrivateSub Button9_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button9.Click
IfMsgBox("Are you sure you want to quit?", MsgBoxStyle.YesNo) = MsgBoxResult.YesThen
Form2.Show()
Me.Hide()
EndIf

EndSub

PrivateSub Form5_Load(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
HandlesMyBase.Load
'TODO: This line of code loads data into the 'MEDICINETABLEDataSet.MEDICINE' table.
You can move, or remove it, as needed.
Me.MEDICINETableAdapter.Fill(Me. MEDICINETABLEDataSet.MEDICINE)

EndSub

PrivateSub Button1_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button1.Click
Dim search AsString = TextBox6.Text

Me. MEDICINETableAdapter.FillBysearchMEDICINECODE(Me. MEDICINETABLEDataSet.
MEDICINE, search)
EndSub

PrivateSub Button11_Click (ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button11.Click
MEDICINEBindingSource.AddNew()

EndSub

PrivateSub Button4_Click (ByVal sender AsSystem.Object, By Val e AsSystem.EventArgs)
Handles Button4.Click
MEDICINEBindingSource.MoveNext()
```

```
EndSub

PrivateSub Button5_Click (By Val sender AsSystem.Object, By Val e AsSystem.EventArgs)
Handles Button5.Click
MEDICINEBindingSource.MovePrevious()

EndSub

PrivateSub Button10_Click (By Val sender AsSystem.Object, By Val e AsSystem.EventArgs)
Handles Button10.Click
If DateTimePicker1.Text <= DateTimePicker2.Text Then
MessageBox.Show("manufacture date should not be equal or greater than Expiry date")
Else

MEDICINETableAdapter.Update(MEDICINETABLEDataSet)
MEDICINEBindingSource.EndEdit()



EndIf


EndSub

PrivateSub Button6_Click (By Val sender AsSystem. Object, By Val e AsSystem.EventArgs)
Handles Button6.Click
IfMessageBox.Show("Do you really want to delete the DATA?", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Warning) = Windows.
Forms.DialogResult.YesThen

MEDICINEBindingSource.RemoveCurrent()
EndIf


EndSub

PrivateSub DataGridView1_CellContentClick (By Val sender As System. Object, By Val e As
System. Windows. Forms.DataGridViewCellEventArgs) Handles
DataGridView1.CellContentClick

EndSub

PrivateSub DateTimePicker1_ValueChanged (By Val sender As System. Object, By Val e
AsSystem.EventArgs)

EndSub
```

```
PrivateSub ComboBox4_SelectedIndexChanged (By Val sender As System. Object, By Val e
AsSystem.EventArgs) Handles ComboBox4.SelectedIndexChanged

    ComboBox4.Select()

EndSub
EndClass
```

SECTION SEVEN: CONCLUSION

This project addresses the benefits of a computerized system would bring as opposed to manual system. It highlights issues related to human computer interaction (HCI)

ACHIEVEMENT

After undertaking this project for the last 3 months I have been able to widen and improve my knowledge and skills application development. I have acquired experience in project management in such variables like project scheduling and planning, time management.

CHALLENGES

During the project I have encountered some challenges which are worth mentioning;

- Coding was a bit time consuming
- System testing-the testing environment wasn't favorable.

DESIGN METHOD

Though the waterfall method helped a lot, it was a drawback requirement drift. After planning at each stage of the project it had to be ensured that it was kept within time, it was difficult however, to long each stage would take due to little experience of writing a project of this scale.

RECCOMEDATIONS

The management should be in a position to respond to the demand of the system by providing full support both in their approval or momentary terms.

The management should make sure to offer in-house training early in advance to facilitate the worm reception of the new system.

The system to be used may only be practically installed if all the hardware and software requirements, which were stipulated in the Proposal, are fully met.

REFERENCES

- ✓ Software Project Management Plan *Jacksonville State University Computing and Information Sciences*, Jacksonville State University, 2003
- ✓ Ian Somerville, 2001 *Software Engineering* Pearson Education Limited, London