**BINDURA UNIVERSITY OF SCIENCE EDUCATION**

**Faculty of Science Education**



**TUCKSHOP MANAGEMENT SYSTEM FOR CHIPADZE HIGH SCHOOL**

**For**

**Cathrine Kudhanda**

**B1645198**

**Supervisor Mr Chikwiriro**

***A PROJECT SUBMITTED IN TO THE EDUCATION DEPARTMENT IN PARTIAL FULFILLMENT TO THE REQUIREMENTS OF THE DIPLOMA SCIENCE EDUCATION.***

**Release Form**

Author: **B1645198. Cathrine Kudhanda**

Title : **Tuckshop management system for Chipadze high school**

Year granted : **2018**

Permission is hereby granted to Bindura University of Science Education library to produce single copies of this project and to lend or sell such copies for private, scholarly or scientific research purposes only. The author reserves other publication rights, and the project or extensive extracts from it may not be printed or otherwise reproduced without the author's written permission.

Permanent address: ……………………

Signed------------------------------------------

Date---------------------------------------------

**APPROVAL FORM**

The undersigned certify that they have supervised the student Cathrine Kudhanda dissertation entitled “**Tuck-shop Management System**” submitted in partial fulfilment of the requirements for the Diploma Science Education of Bindura University of Science Education.

…………………………………………… …………………………………………………

DATE

STUDENT NAME

………………………………………… ……………………………………………..

DATE

SUPERVISOR

………………………………………..... ……………………………………………

DATE

CHAIRPERSON

……………………………………….. ……………………………………………….

EXAMINER DATE

EXTERNAL

**DEDICATION**

This study is dedicated to the glory of God Almighty and to my loving husband…….., Mr and Mrs. Kudhanda May the good Lord continues to bless you all.

**ACKNOWLEDGEMENT**

First and foremost, I wish to thank the Almighty God, my parents and family, for giving me the opportunity and strength to complete the current study. I wish to express my sincere gratitude to those whose support made the completion of this study possible. In particular I would like to thank: Mr Chikwiriro, my supervisor, for his patience, wisdom, continuous support, encouragement and guidance. Kudhanda family, for their patience and assistance with literature sources. To all my friends Patience, Innocent, Florence, Kudzanai, Sydney and many more for their encouragements.

**ABSTRACT**

The Tuck-Shop Management System is software which can become the backbone for a billing and inventory system for small organizations. This software provides an uncomplicated system to run Tuck-Shop stores. This application could be very useful to small organizations. This application is inspired from current pen and paper based Tuck-Shop management systems. It will provide an easy and attractive interface so that the user can easily manage and utilize the application. Various other approaches were considered for this application.

Table of Contents

[Chapter 1: Problem Identification 10](#_Toc530212762)

[1.1 Problem definition 10](#_Toc530212763)

[1.2 Investigation and description of current system 10](#_Toc530212764)

[1.3 Statement of the problem 11](#_Toc530212765)

[1.4 System Objectives 11](#_Toc530212766)

[1.5 Limitations 11](#_Toc530212767)

[1.6 Scope/Delamination of the System 11](#_Toc530212768)

[1.7 Definition of Terms 12](#_Toc530212769)

[1.8 Conclusion 12](#_Toc530212770)

[Chapter 2: Requirements Specification 13](#_Toc530212771)

[2.1 Introduction 13](#_Toc530212772)

[2.2 Fact finding and analysis tools 13](#_Toc530212773)

[2.3 Data requirements 14](#_Toc530212774)

[2.4 Processing requirements 17](#_Toc530212775)

[2.5 Software requirements 18](#_Toc530212776)

[2.6 **Hardware requirements** 18](#_Toc530212777)

[Chapter 3: Design 19](#_Toc530212778)

[3.1 Introduction 19](#_Toc530212779)

[3.2 Interface Design 19](#_Toc530212780)

[3.3. System Design 23](#_Toc530212781)

[3.4. Use-Case Diagram 23](#_Toc530212782)

[3.4.1. A Use-case diagram 23](#_Toc530212783)

[3.5. System Overview 24](#_Toc530212784)

[3.6 Database design 26](#_Toc530212785)

[*3.7. Conclusion* 30](#_Toc530212786)

[Chapter 4: Coding and Testing 31](#_Toc530212787)

[4.1. Introduction 31](#_Toc530212788)

[4.2. Technical documentation 31](#_Toc530212789)

[4.2.1. Code for connecting to the database 31](#_Toc530212790)

[4.2.2. Code for system log in 31](#_Toc530212791)

[4.2.3. Code for student registration 32](#_Toc530212792)

[4.2.4. Code for updating student details 33](#_Toc530212793)

[4.2.5. Code for Product entry 34](#_Toc530212794)

[4.2.6. Code for Product List 35](#_Toc530212795)

[4.2.7. Code for supplier Entry 35](#_Toc530212796)

[4.2.8. Code for listing suppliers 36](#_Toc530212797)

[4.2.9 Code for batches entry 36](#_Toc530212798)

[4.3. Unit and system testing 38](#_Toc530212799)

[4.3.1. Unit Testing 38](#_Toc530212800)

[4.4. User testing 39](#_Toc530212801)

[4.5. Conclusion 41](#_Toc530212802)

[Chapter 5: Implementation 42](#_Toc530212803)

[5.1 Introduction 42](#_Toc530212804)

[5.2 Conversion Plan 42](#_Toc530212805)

[5.3 user manual 43](#_Toc530212806)

[5.4 User training plan 46](#_Toc530212807)

[Chapter 6: Evaluation and conclusion 47](#_Toc530212808)

[6.1 introduction 47](#_Toc530212809)

[6.2 Evaluation 47](#_Toc530212810)

[6.2.1 Achievements 47](#_Toc530212811)

[6.2.2 Limitations 47](#_Toc530212812)

[6.3 Opportunities for further development 48](#_Toc530212813)

[6.4 Conclusion 48](#_Toc530212815)

[References 49](#_Toc530212816)

**List of Tables**

[Table 1 Product 17](#_Toc530213172)

[Table 2 Supplier 17](#_Toc530213173)

[Table 3 Orders 17](#_Toc530213174)

[Table 4 Processing Table 18](#_Toc530213175)

[Table 5 Software requirements 19](#_Toc530213176)

[Table 6 Servers/Main Computers 19](#_Toc530213177)

[Table 7 Useraccout 29](#_Toc530213178)

[Table 8 suppliers 29](#_Toc530213179)

[Table 9 Product 30](#_Toc530213180)

[Table 10 Order 30](#_Toc530213181)

[Table 11 Sales 31](#_Toc530213182)

**List of Figures**

[Figure 1 Product Entry 5](#_Toc530214920)

[Figure 2 Supplier Entry 6](#_Toc530214921)

[Figure 3 Orders 6](#_Toc530214922)

[Figure 4 Login Form 10](#_Toc530214923)

[Figure 5 Home Page 11](#_Toc530214924)

[Figure 6 Add customer 12](#_Toc530214925)

[Figure 7 Add supplier 12](#_Toc530214926)

[Figure 8 Add batches 12](#_Toc530214927)

[Figure 9 Student list 13](#_Toc530214928)

[Figure 10 Supplier list 13](#_Toc530214929)

[Figure 11 Admin Activity 14](#_Toc530214930)

[Figure 12 Storekeeper Activity 15](#_Toc530214931)

[Figure 13 System Overview 15](#_Toc530214932)

[Figure 14 the Level O DFD 16](#_Toc530214933)

[Figure 15 First-level DFD for Storekeeper 16](#_Toc530214934)

[Figure 16 Second-level DFD for Storekeeper 17](#_Toc530214935)

[Figure 17 Entity-Relationship Model of the Database 18](#_Toc530214936)

[Figure 18 showing the results of missing information 29](#_Toc530214937)

[Figure 19 Displaying data on the forms 30](#_Toc530214938)

[Figure 20 Displaying of the output from the database 30](#_Toc530214939)

[Figure 21 Adding student details to the system 31](#_Toc530214940)

[Figure 22 Adding details of new book in database 31](#_Toc530214941)

[Figure 23 installing the software 34](#_Toc530214942)

[Figure 24 System is now installing 35](#_Toc530214943)

[Figure 25 Choose next 35](#_Toc530214944)

[Figure 26 Installation completed successfully 36](#_Toc530214945)

[Figure 27 Click Icon on the Desktop 36](#_Toc530214946)

# Chapter 1: Problem Identification

## Problem definition

The Chipadze High School Tuckshop is a Tuckshop that use a manual system in all their process, which is a slow process in doing everything in their shop, the manual system is also expensive because it also requires cupboard for keeping all the files and storage, its time consuming to search for a particular record . So the researcher is there to upgrade their system from manual to computerised system which is fast and safe on keeping the files

## 1.2 Investigation and description of current system

**Stock details file**

All the products supplied by supplier are put in a warehouse and the details of each item produced is recorded in the stock details file. Information recorded includes item name, item identification number (item id), and quantity.

**Supplier details file**

The company also keeps records of all the companies that supply the products required. The information kept includes suppliers’ Identification number, suppliers’ name, and suppliers’ address

**Order details file**

When products are to be bought from suppliers, an order is first made. The order specifies order Id, item name, quantity, order date. The supplier name and supplier Identification number which are used to update this file are retrieved from the supplier details file.

***Purchases details file***

When a purchase has been made, the clerk records the purchases details in this file. Information recorded includes invoice number, order Identification number (order id), quantity bought, supplier Identification number (supplier id) etc. The order Id and item name used to update this file are retrieved from the order details file whilst supplier Id and suppler name are retrieved from the supplier details file

***Sales details form***

When a customer purchases Maheu from the shop details of the Maheu bought are recorded in the sales details file. The information recorded includes receipt number, customer name, item bought

## 1.3 Statement of the problem

**The system that is being used now has the following problems**

* Miscalculations of sales as they will be lots of papers to calculate.
* It can be time consuming especially when there are more customers on the counter.
* Misplacement of some of the documents as they will be lots of paper work done every day.

## 1.4 System Objectives

The proposed computerized tuckshop management system will be able to:

* Validate input purchases, sales and order figures so that correct figures are sent for processing
* Make accurate calculations on items purchased, costs and items sold, through the use of formulas
* Update stock levels in the stock details file when item are purchased or sold

## 1.5 Limitations

Due to time and financial constraints, the software that is developed covers only the aspect of Chipadze High School.

## 1.6 Scope/Delamination of the System

The project is going to;

* Reduce the transcription errors during data entry into the books of records.
* Easy amendments and updating of file records without messing up the books of records.
* Provide file security through the use of passwords and user IDs.
* Instantly update of the quantity in stock and automatically remind management when a particular item is running out of stock
* To maintain the records of the tuckshop, customers, suppliers with great ease. The stock of the products that are available must be updated from time to time
* Help them in report generation
* Automatically make calculations of daily, weekly or monthly total sales made by the Tuckshop

## 1.7 Definition of Terms

**Microsoft Access**- is database management system

**Authentication** - The state of a user being logged in.

**Authorization** - The state of a user having sufficient permission in order to perform a desired task.

## 1.8 Conclusion

The Tuckshop system is proposed for Chipadze high school. The proposed work describes an automated system that progresses from the traditional method of manual buying and selling of goods to an automated process, by providing controlled access to the resources. There is going to be an analysis of similar existing systems in different tuckshops.

# Chapter 2: Requirements Specification

## 2.1 Introduction

Requirement analysis is the way to understand the requirements of the users of the system. It explores the expectations of the user about the proposed system. Requirements are actually the descriptions on how the system should work and interact with the user of the system. The efficiency of the final product depends on the accuracy of the requirement analysis.

## 2.2 Fact finding and analysis tools

Educational research uses a variety of research instruments like interviews, observations and questionnaires for the purpose of this research questionnaires shall be used. Interviews will also be used to compliment the questionnaires.

Questionnaires

Oppenheim (2009) refers to the questionnaires as any data collection instrument; attitude scales projective techniques or rotting saddles. The questionnaires will be chosen because of its advantages: The results of the questionnaires can usually be quickly and easily quantified by either a researcher or through the use of a software package. Large amounts of information can be collected from a large number of people in a short period of time and in a relatively cost effective way. Questionnaires can also be carried out by the researcher or by any number of people with limited effect to its validity and reliability.

It is an instrument that can be given to respondents within a short space of time. Questionnaires help reduce costs in terms of money and effort. They also serve to ensure confidentiality because the respondents are not asked to indicate their class or names.

Although the questionnaire might prove straight forward information there can be hitches which can be uncounted within the techniques which are employed if questionnaires are postponed so respondents may not cooperate and not return them if they are reminded. The language may be misunderstood. This may lead to unwanted responses being given in open ended questions and inadequate answers when questionnaires are posted. In this case the researcher will overcome some of these problems by physically administering the questionnaires. Omission will be noticed as she will be collecting the questionnaires and the respondents will kindly be asked to finish answering the questionnaires to ensure clarity of the questionnaires and that respondents would not object to answering any question.

**Interviews**

The research instrument will be used in this study because of its main advantage of being adaptable. The researcher will use this instrument to cross check the validity of the questionnaires.

This is a personal interaction between the interviewer and the interviewee. The advantages of an interview over using questionnaires is that the interviewer gets to observe facial expression and verbal cues which may either be supporting or belying the spoken word hence having more validity. Interviews are however more time consuming than questionnaires which are mainly targeting the views of the users of the Tuck-shop System at Chipadze high school.

## 2.3 Data requirements

Product Code

Product Name

Description

Price

Supplier

Figure 1 Product Entry

Name

Surname

Phone

Address

Figure 2 Supplier Entry

Total Amount

orderID

ProductID

Pice

Quntity

Figure 3 Orders

**DATA STRUCTERS AND FILE ORGANISATION**

Table 1 Product

|  |
| --- |
| Field name field type field size format examples |
| Pid Numeric 5 1 |
| Pname Text 15 ZapNax |
| Pdiscription Text 15 500g red |
| Price Currency $0.50 |
| Quantity 1 30 1 |

Table 2 Supplier

|  |
| --- |
| Field name field type field size format examples |
| Sid Numeric 4 1 |
| Name Text 30 TM Supermarket |
| Phone Number 10 077928288 |
| Address Text 90 Robert Mugabe |

Table 3 Orders

|  |
| --- |
| Field name field type field size format examples |
| OrderID Numeric 6 1 |
| PID Numeric 5 2 |
| SID Numeric 5 2 |
| Price Currency $90.00 |
| Quantity Numeric 8 23 |
| TotalAmount Currency $2 070.0 |

## 2.4 Processing requirements

The following users were consulted and outlined their expectations:

Table 4 Processing Table

|  |  |
| --- | --- |
| **USER** | **EXPECTATION(S)** |
| Storekeep | * A system that shows the total amount of sales and total number of products in the stock * A user-friendly interface * A system that can reduce the number of entries to be made each time a transaction occurs. * An automatically updating system. |
| Administration & Management | A more secure system so that easy data loss is avoided.  A portable system that takes up less disk space and requires minimum resources so that some costs such as stationary costs are eliminated. |

Functional and non-functional requirements

Functional requirements;

The system must be able to;

* Support real time updating of all functions
* Allow staff to initiate a database search from any point in the system
* Provide for the use of function /hot keys for frequently used functions
* Validate isbn-10 and isbn-13

Non-functional requirements;

* Usability
* Availability & Reliability
* Backup & Disaster recovery
* Privacy & Security
* Accessibility

2.5 Software requirements

Table 5 Software requirements

|  |  |
| --- | --- |
| **Software** | **Purpose** |
| Visual Basic 2010 or 2015 | Designing and coding the new system |
| Microsoft Office 2007 or later | Creation of Reports and Database storage |
| Antivirus | Securing the system from virus attacks |

2.6 Hardware requirements

**HARDWARE REQUIREMENTS**

* **Input**

Keyboard- For capturing library members’ information and books information.

Mouse – selection of options

Bar code reader – to capture book id and ISBN

* **Output**

High Definition (HD) Monitor – Quality and crystal clear display of records

Laser Printer –For printing of reports when needed by the administration

Table 6 Servers/Main Computers

|  |  |
| --- | --- |
| **Server Component** | **Specifications** |
| Processor | Intel Pentium 4 or better |
| System Memory | At least 1GB RAM |
| Storage Media | 80GB Hard Disk Drive or better, CD RW |

# Chapter 3: Design

## 3.1 Introduction

System design is the most important phase of the system development lifecycle. It is a solution, a how to approach to the creation of a new system. It provides the understanding and procedural details necessary for implementing the system recommended in feasibility study. Emphasis is on translating the performance requirements in to design specifications. The design phase is a transition from a user oriented document (system proposal) to a document oriented to the programmers or database personnel. A system design gives accurate system performance. Several structural tools are used for design purpose of the candidate system

## 3.2 Interface Design

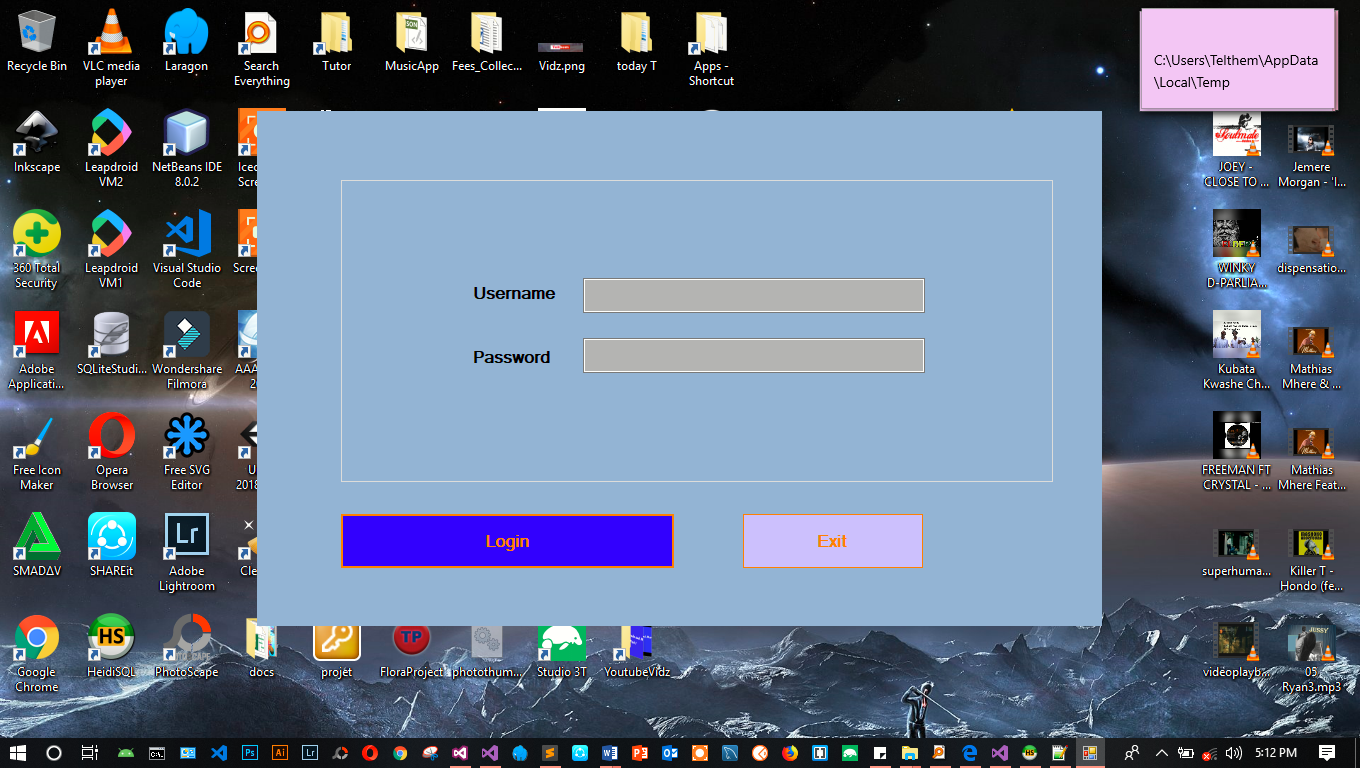


Figure 4 Login Form

The Tuck-shop system has optimum level of security. Storekeeper need to provide a username for identification purposes and a password for authentication so that only authorized member’s access the Tuck-shop system



Figure 5 Home Page

The home page interface in figure 2 above displays the menu for every activity Storekeeper should work on. There is a menu for add products, register student and other

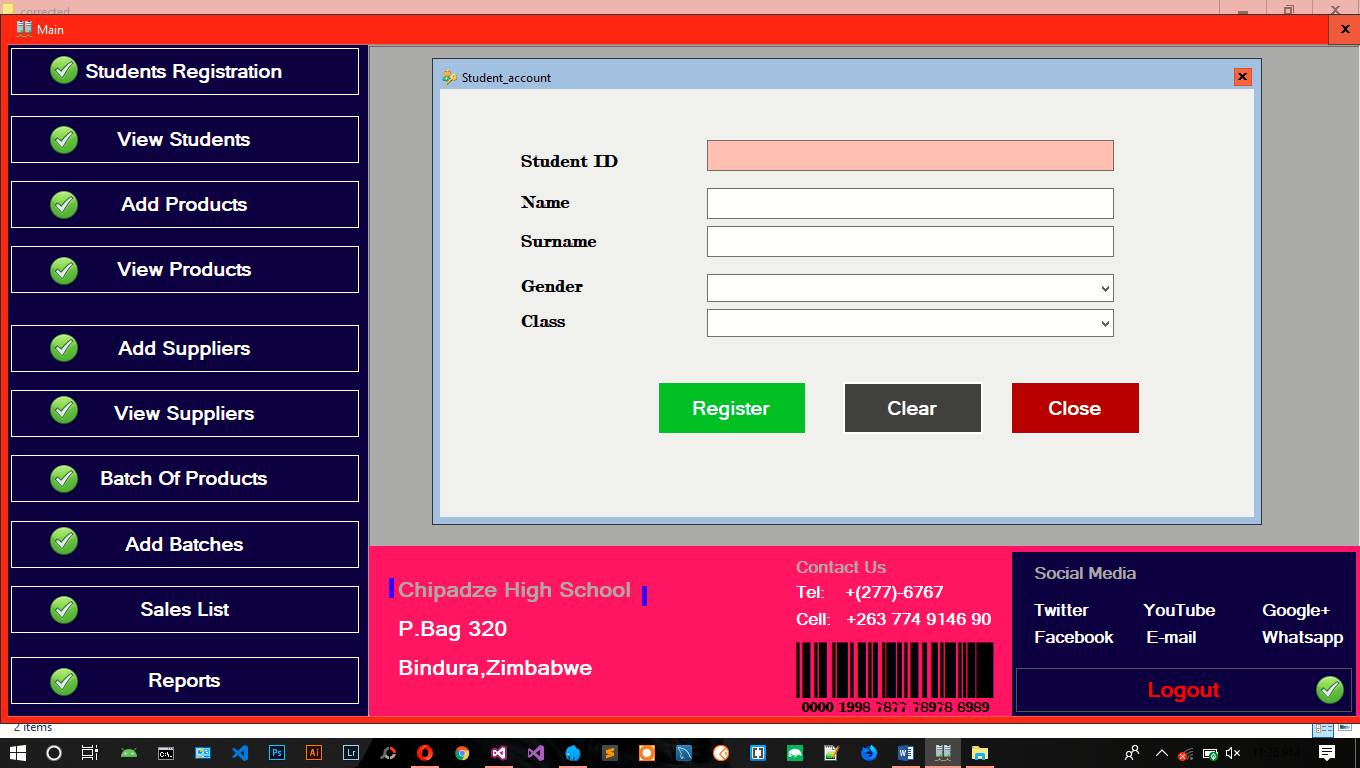


Figure 6 Add customer

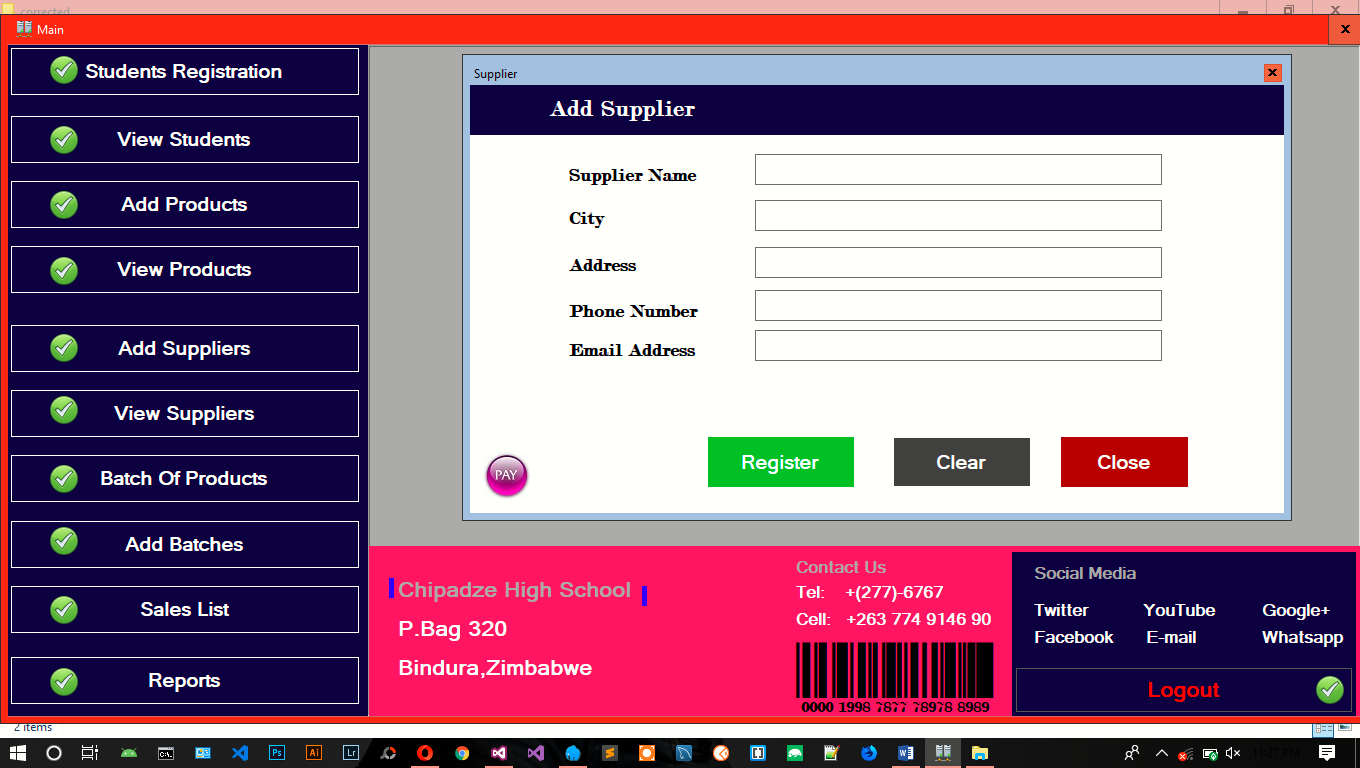


Figure 7 Add supplier

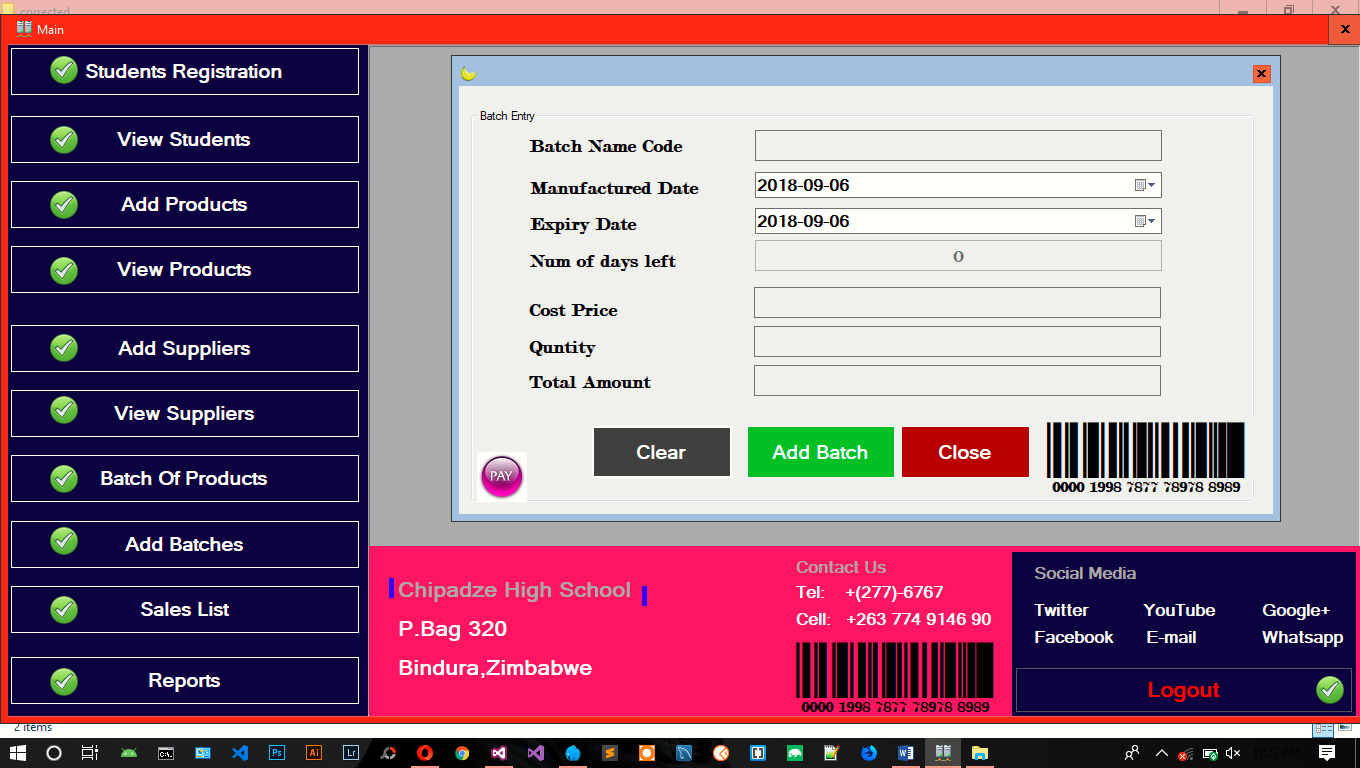


Figure 8 Add batches

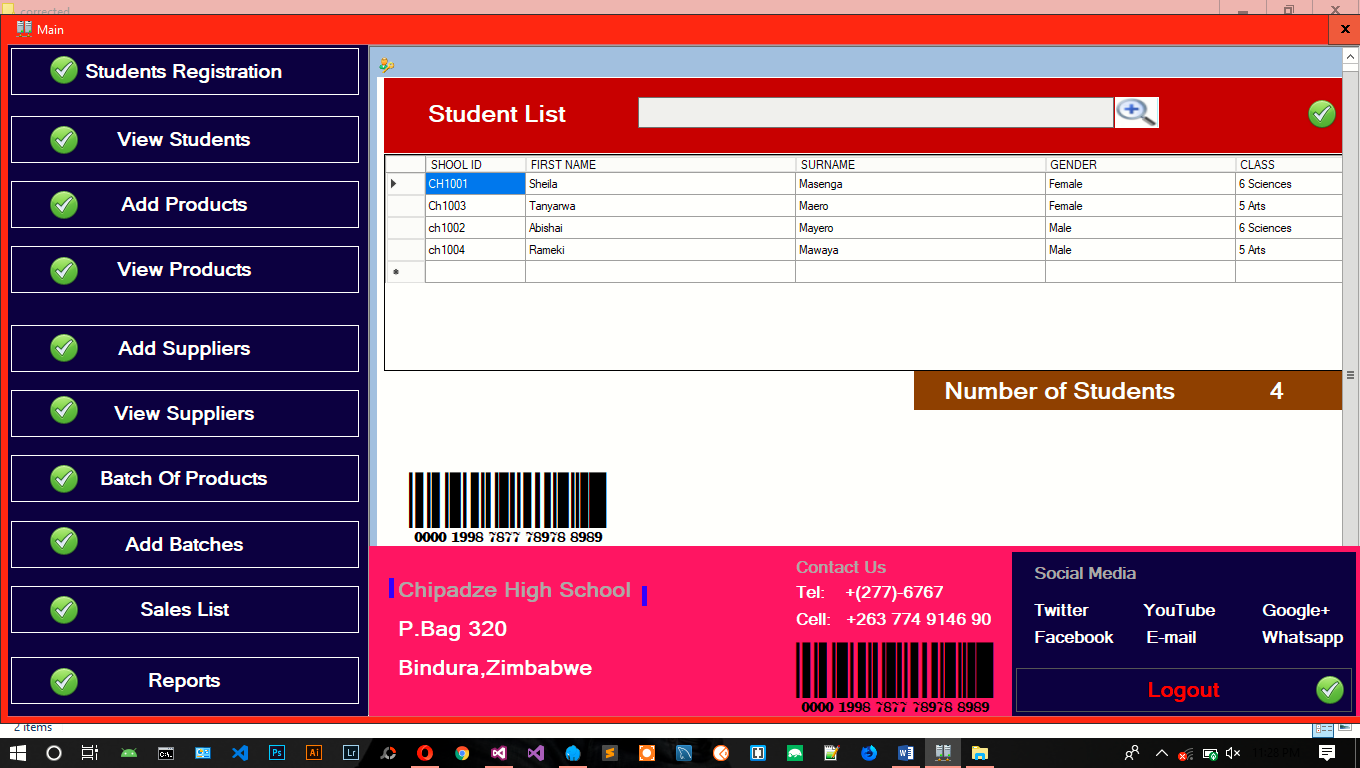


Figure 9 Student list

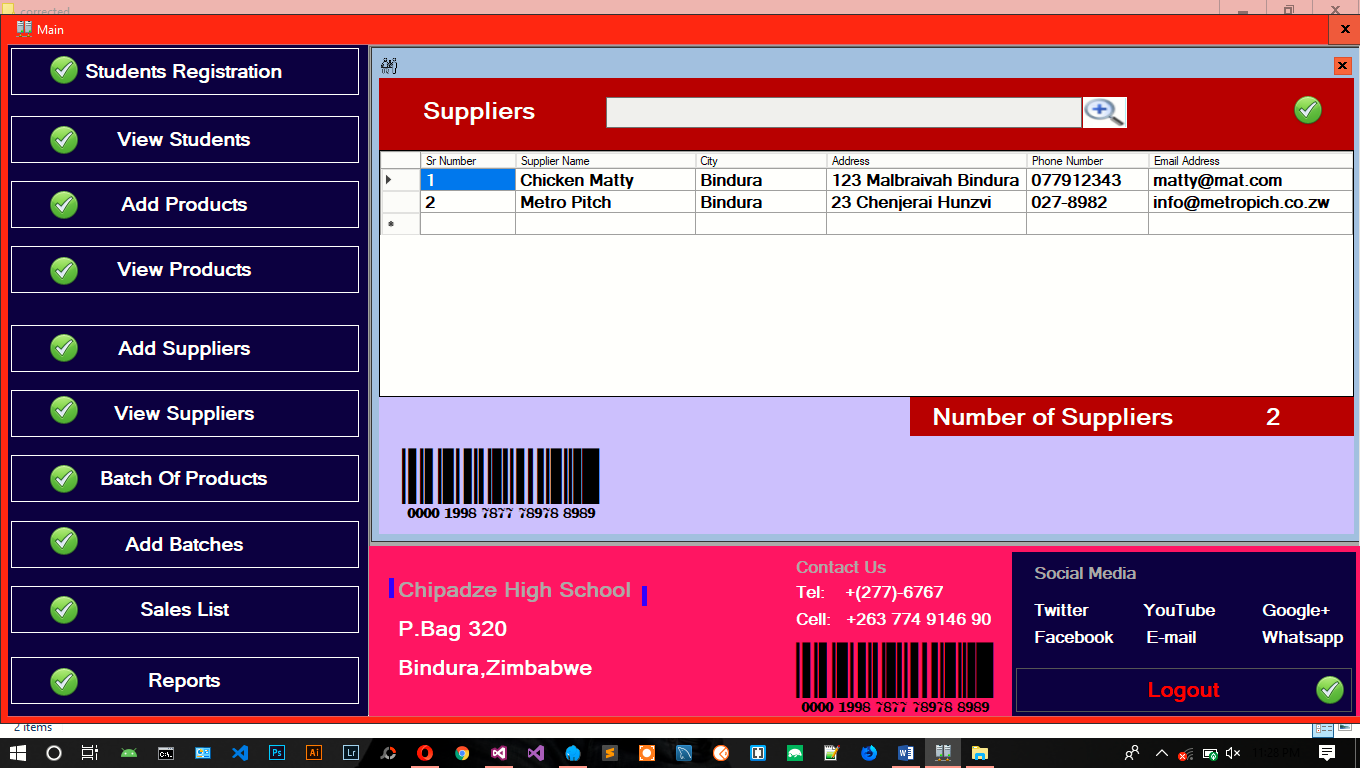


Figure 10 Supplier list

## 3.3. System Design

This phase focuses on the design of the proposed system, the design being based on the findings of the analysis of the current system. It gives an outline of the physical design, the architecture design, interface design, database design and program design. It outlines all the design needed for the proposed system. After exploring and documenting all the processes involved in the existing system it is now time to design the proposed system. This involves giving an outline of how the proposed system is going to be developed, configured and deployed. In this phase researcher will outline the Physical Design, Systems Architecture and Database design, Interface Design, Program Design and Test Design.

## 3.4. Use-Case Diagram

To clearly outline a set of scenarios that describes an interaction between a user and the system,

### 3.4.1. A Use-case diagram

First administrator will login into the system. The click adds student, products, sales or update product. Admin has the right to update the products, students and delete them.

Figure 11 Admin Activity

Figure 12 Storekeeper Activity

## 3.5. System Overview

The architectural design specifies the major components of the application and how they communicate with each other. The Tuckshop will consist of a database and a user interface

Supplier Management

Order Management

Product Management

Login Management

User System Management

Customer Management

Figure 13 System Overview

Order

Make Orders

Tuck-shop Management System

Make Payment

Supply purchased products

Make Payment

Figure 14 the Level O DFD

**FIRST-LEVEL DFD FOR STOREKEEPER**

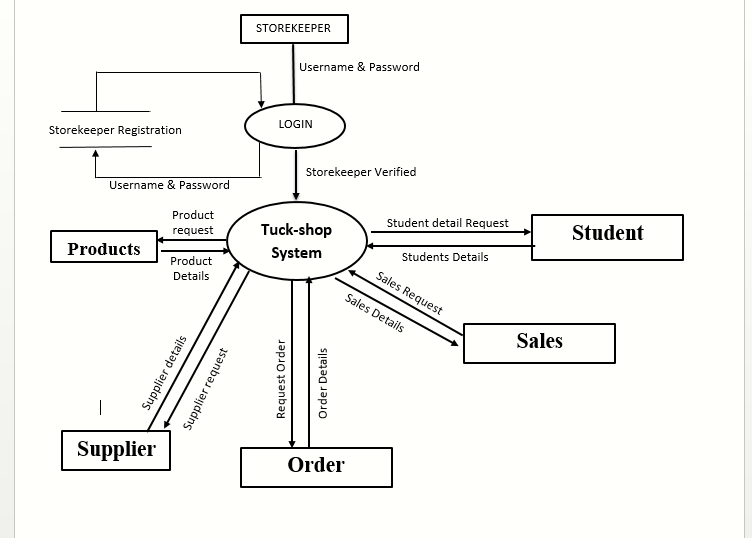


Figure 15 First-level DFD for Storekeeper

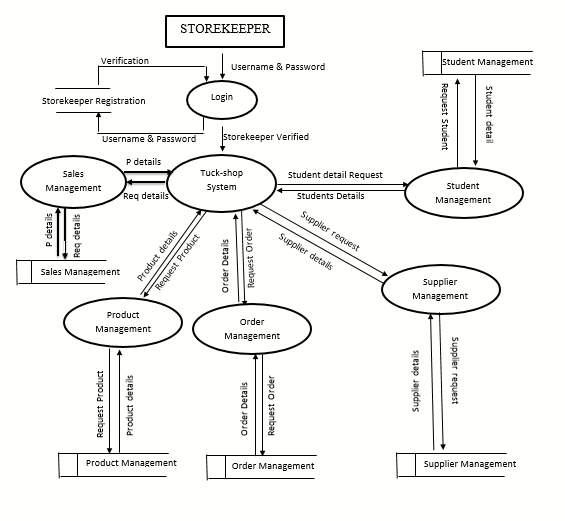


Figure 16 Second-level DFD for Storekeeper

## 3.6 Database design

Microsoft Access database is a relational database management system that is well known for its reliability and dynamics when it comes to database management. It offers good security for database, at the same time allowing for rapid insertion, deletion and updating of entries into the database.

**Entity-Relationship Model of the Database**

Admin

Sales

Product

Useraccount

Supplier

Order

Customer

Manage

Supply

Handle

Figure 17 Entity-Relationship Model of the Database

Table 7 Useraccout

|  |  |  |
| --- | --- | --- |
| **Data Item** | **Type** | **Description** |
| id | Int | Used to track user |
| role | Text | Track role of each user |
| username | Text | Username must be unique |
| password | Text | The password of user |

Table 8 suppliers

|  |  |  |
| --- | --- | --- |
| **Data Item** | **Type** | **Description** |
| id | Int | Used to track user |
| Company name | Text | Company’s name |
| address | Text | Company’s address |
| Phone number | Integer | To contact Company’s |

Table 9 Product

|  |  |  |
| --- | --- | --- |
| **Data Item** | **Type** | **Description** |
| id | Int | Used to track user |
| ProductName | Text | Product name |
| price | Integer | Price for product |
| Qunty | integer | Quanty |
| Supplier\_id | Int | Supplier id foreign key |

Table 10 Order

|  |  |  |
| --- | --- | --- |
| **Data Item** | **Type** | **Description** |
| id | Int | Order number |
| Productid | Int | Product name (FK) |
| Supplier\_id | Int | Supplier (FK) |
| price | Integer | Price for products |
| Total\_amount | Currency | Amount for all products |
| Payment\_mode | Text | Eco Cash or bank transfer |

Table 11 Sales

|  |  |  |
| --- | --- | --- |
| **Data Item** | **Type** | **Description** |
| id | Int | invoice number |
| Productid | Int | Product name (FK) |
| customer\_id | Int | Customer id (FK) |
| price | Integer | Price for products |
| quantity | Int | Number of products purchased |

## 3.7. Conclusion

The analysis process is useful in presenting justification of why it is necessary to build a new system by analyzing the current system and bring out it weakness and also helped in spelling out the data requirements of the new system so as to the best data requirements during the design phase which is the next chapter.

# Chapter 4: Coding and Testing

## 4.1. Introduction

The objective of this chapter is to build and test a reliable information system with trained users ready to benefit as expected from use of the system. Coding is the physical creation of the system programs. The system will be programmed using VB.Net which is a scripting language and My SQL for Database Management System.

## 4.2. Technical documentation

This section presents the system code of the system according to the functions to be performed.

### 4.2.1. Code for connecting to the database

Imports MySql.Data.MySqlClient

Module Db\_Conn

Dim MysqlConn As New MySqlConnection

Public Function myconn() As MySqlConnection

Return New MySqlConnection(My.Settings.mydbConnection)

End Function

End Module

### 4.2.2. Code for system log in

If txtUser.Text = "" Then

lblError.Text = "Username is required Please"

lblError.Visible = True

ElseIf txtPass.Text = "" Then

lblError.Text = "Password is required Please"

lblError.Visible = True

Else

sql = "SELECT \* FROM `users` WHERE `username`='" & txtUser.Text & "' AND `password`='" & txtPass.Text & "' "

bookfindthis(sql)

If GetNumRows() = 1 Then

MsgBox("Logged in Successfully", MsgBoxStyle.Information)

LoadSingleResult("login")

txtPass.Text = ""

txtUser.Text = ""

### 4.2.3. Code for student registration

If btnReg.Text = "Register" Then

'select all records in the database

tuckfindthis("SELECT COUNT(\*) FROM `customer` WHERE `studID`='" & txtID.Text & "'")

'check if the id is already exist in the database

If GetCount() > 0 Then

MsgBox("Student ID Number is already exist Please try something new!", MsgBoxStyle.Critical)

Else

'checking for empty textboxes and show error

If txtID.Text = "" Then

MsgBox("Student ID Number is Required!", MsgBoxStyle.Critical)

ElseIf txtName.Text = "" Then

MsgBox("Name is Required!", MsgBoxStyle.Critical)

ElseIf txtSurnm.Text = "" Then

MsgBox("Surame is Required!", MsgBoxStyle.Critical)

ElseIf cmbgender.Text = "" Then

MsgBox("Please select gender!", MsgBoxStyle.Critical)

ElseIf cbClass.Text = "" Then

MsgBox("Please select class!", MsgBoxStyle.Critical)

'if there no errors occur

Else

'then insert record into database now

issucess = product\_insert("INSERT INTO `customer` (`studID`, `name`, `surname`,`gender`, `class`) VALUES ('" & txtID.Text & "','" & txtName.Text & "', '" & txtSurnm.Text & "', '" & cmbgender.Text & "', '" & cbClass.Text & "')")

'if successfully

If issucess = True Then

MsgBox("New student has been added successfully!", MsgBoxStyle.Information)

StudentList.Show()

Me.Hide()

txtID.Text = ""

txtName.Text = ""

txtSurnm.Text = ""

Else

' No student has been added!

MsgBox("No student has been added!", MsgBoxStyle.Critical)

txtID.ForeColor = Color.Red

txtName.ForeColor = Color.Red

txtSurnm.ForeColor = Color.Red

End If

End If

End If

ElseIf btnReg.Text = "Update" Then

issucess = product\_insert("UPDATE `customer` SET `name`='" & txtName.Text & "',`surname`= '" & txtSurnm.Text & "',`gender`='" & cmbgender.Text & "',`class`= '" & cbClass.Text & "' WHERE `studID`='" & txtID.Text & "'")

If issucess = True Then

MsgBox("New student has been updated successfully!", MsgBoxStyle.Information)

StudentList.Show()

Else

MsgBox("No student has been updated!", MsgBoxStyle.Critical)

End If

End If

### 4.2.4. Code for updating student details

issucess = book\_insert("UPDATE `student\_details` SET `name`='" & txtFname.Text & "',`surname`= '" & txtSurname.Text & "',`dateOfBirth`='" & txtDOB.Text & "',`gender`= '" & cboGender.Text & "',`nationality`= '" & txtNational.Text & "'," &

"`class`='" & cboClass.Text & "',`term`= '" & cboTerm.Text & "',`address`='" & txtAddress.Text & "',`fatherName`= '" & txtFather.Text & "', `occupation`='" & txtOcpy.Text & "',`phone`='" & txtPhone.Text & "' WHERE `regNumber`='" & txtReg.Text & "'")

If issucess = True Then

MsgBox("Student has been updated successfully!", MsgBoxStyle.Information)

StudentsList.Show()

Me.Hide()

txtReg.Clear()

txtFname.Clear()

txtSurname.Clear()

txtDOB.ResetText()

cboGender.ResetText()

cboClass.ResetText()

cboTerm.ResetText()

txtNational.Clear()

txtAddress.Clear()

txtFather.Clear()

txtOcpy.Clear()

txtPhone.Clear()

Else

lblError.Text = "No student has been updated!"

lblError.Visible = True

### 4.2.5. Code for Product entry

If btnReg.Text = "Add Product" Then

'checking for empty textboxes and show error

If txtPName.Text = "" Then

MsgBox("Product Name Is Required!", MsgBoxStyle.Critical)

ElseIf cmbsupler.Text = "" Then

MsgBox("Please Select supplier", MsgBoxStyle.Critical)

ElseIf cbBatch.Text = "" Then

MsgBox("Please Select batch name!", MsgBoxStyle.Critical)

ElseIf txtPrds.Text = "" Then

MsgBox("Please give the description Of product", MsgBoxStyle.Critical)

ElseIf txtPrice.Text = "" Then

MsgBox("Price Of item Is required!", MsgBoxStyle.Critical)

'if there no errors occur

Else

'then insert record into database now

issucess = product\_insert("INSERT INTO `products` (`ProdName`, `SupplierID`, `ProductDesc`,`batchID`, `cost\_price`) VALUES ('" & txtPName.Text & "','" & supplierID.Text & "', '" & txtPrds.Text & "', '" & batchid.Text & "', '" & txtPrice.Text & "')")

'if successfully

If issucess = True Then

MsgBox("New Product has been added successfully!", MsgBoxStyle.Information)

Product\_List.Show()

Me.Hide()

txtPName.Text = ""

txtPrds.Text = ""

txtPrice.Text = ""

Else

' No student has been added!

MsgBox("No product has been added!", MsgBoxStyle.Critical)

txtPName.ForeColor = Color.Red

txtPrds.ForeColor = Color.Red

txtPrice.ForeColor = Color.Red

End If

End If

End If

### 4.2.6. Code for Product List

Private Sub Product\_List\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

tuckfindthis("SELECT `PID`, `ProdName`,`ProductDesc`, `cost\_price` FROM `products`")

LoadData(ProductsGrid, "products")

End Sub

**End Sub**

### 4.2.7. Code for supplier Entry

If btnReg.Text = "Register" Then

'checking for empty textboxes and show error

If txtName.Text = "" Then

MsgBox("Name is Required!", MsgBoxStyle.Critical)

ElseIf txtCity.Text = "" Then

MsgBox("City is Required!", MsgBoxStyle.Critical)

ElseIf txtAddress.Text = "" Then

MsgBox("Please select gender!", MsgBoxStyle.Critical)

ElseIf txtNumber.Text = "" Then

MsgBox("Please mobile is required!", MsgBoxStyle.Critical)

ElseIf txtEmail.Text = "" Then

MsgBox("Please email address is required!", MsgBoxStyle.Critical)

'if there no errors occur

Else

'then insert record into database now

issucess = product\_insert("INSERT INTO `supplier` (`supplierName`, `City`, `SupplierAddress`,`Phone`, `Email`) VALUES ('" & txtName.Text & "','" & txtCity.Text & "', '" & txtAddress.Text & "', '" & txtNumber.Text & "', '" & txtEmail.Text & "')")

'if successfully

If issucess = True Then

MsgBox("New supplier has been added successfully!", MsgBoxStyle.Information)

SupplierList.Show()

Me.Hide()

txtName.Text = ""

txtCity.Text = ""

txtNumber.Text = ""

txtEmail.Text = ""

### 4.2.8. Code for listing suppliers

Private Sub SupplierList\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

tuckfindthis("SELECT `supplier\_id`, `supplierName`,`City`, `SupplierAddress`, `Phone`, `Email` FROM `supplier`")

LoadData(supplierGrid, "suppliers")

End Sub

### 4.2.9 Code for batches entry

Private Sub btnReg\_Click(sender As Object, e As EventArgs) Handles btnReg.Click

If btnReg.Text = "Add Batch" Then

'checking for empty textboxes and show error

If txtNam.Text = "" Then

MsgBox("Batch Name is Required!", MsgBoxStyle.Critical)

ElseIf txtBnum.Text = "" Then

MsgBox("Please select date manufactured and expiry date!", MsgBoxStyle.Critical)

ElseIf txtCost.Text = "" Then

MsgBox("Cost is required!", MsgBoxStyle.Critical)

ElseIf txtQunty.Text = "" Then

MsgBox("Quntity is required!!", MsgBoxStyle.Critical)

ElseIf txtTotalM.Text = "" Then

MsgBox("Please make you provide qntity and price!", MsgBoxStyle.Critical)

'if there no errors occur

Else

'then insert record into database now

issucess = product\_insert("INSERT INTO `batches` (`batch\_name`, `manufactered\_date`, `date\_expiry`,`nmbrofdaysleft`, `cost\_price`, `quntity`, `total\_amount`) VALUES ('" & txtNam.Text & "','" & dtpSt.Text & "', '" & dtpEnd.Text & "', '" & txtBnum.Text & "', '" & txtCost.Text & "', '" & txtQunty.Text & "', '" & txtTotalM.Text & "')")

'if successfully

If issucess = True Then

MsgBox("New batch has been added successfully!", MsgBoxStyle.Information)

Batch\_List.Show()

Me.Hide()

txtNam.Text = ""

txtCost.Text = ""

txtQunty.Text = ""

txtTotalM.Text = ""

Else

' No student has been added!

MsgBox("No batch has been added!", MsgBoxStyle.Critical)

txtNam.ForeColor = Color.Red

txtCost.ForeColor = Color.Red

txtQunty.ForeColor = Color.Red

txtTotalM.ForeColor = Color.Red

End If

End If

ElseIf btnReg.Text = "Update" Then

issucess = product\_insert("UPDATE `batches` SET `batch\_name`='" & txtNam.Text & "',`manufactered\_date`= '" & dtpSt.Text & "',`date\_expiry='" & dtpEnd.Text & "',`nmbrofdaysleft`= '" & txtBnum.Text & "',`cost\_price`= '" & txtCost.Text & "',`quntity`= '" & txtQunty.Text & "',`total\_amount`= '" & txtTotalM.Text & "' WHERE `batch\_name`='" & txtNam.Text & "'")

If issucess = True Then

MsgBox("Batch has updated successfully!", MsgBoxStyle.Information)

StudentList.Show()

Else

MsgBox("No Batch has been updated!", MsgBoxStyle.Critical)

End If

End If

End Sub

**Code for listing batches**

Private Sub Batch\_List\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

tuckfindthis("SELECT `bID`, `batch\_name`,`manufactered\_date`, `date\_expiry`, `nmbrofdaysleft`, `cost\_price`, `quntity`, `total\_amount` FROM `batches`")

LoadData(BatchGrid, "batches")

End Sub

## 4.3. Unit and system testing

### 4.3.1. Unit Testing

This is a testing procedure in which the single components of computer software are tested. It is a software development process that focuses on inspecting the proper operation of system units individually and independently. The rationale is to certify that each unit of the computer software performs as intended by isolating a selected section of the code and ascertain its accuracy. Furthermore, unit testing is preferred because it is assumed that it is easier to recognize an error in a single module than the whole system. In most cases unit testing is done using a single or few system inputs to produce a single output. Unit testing is done using black-box and white-box testing procedure.

* **Missing information**

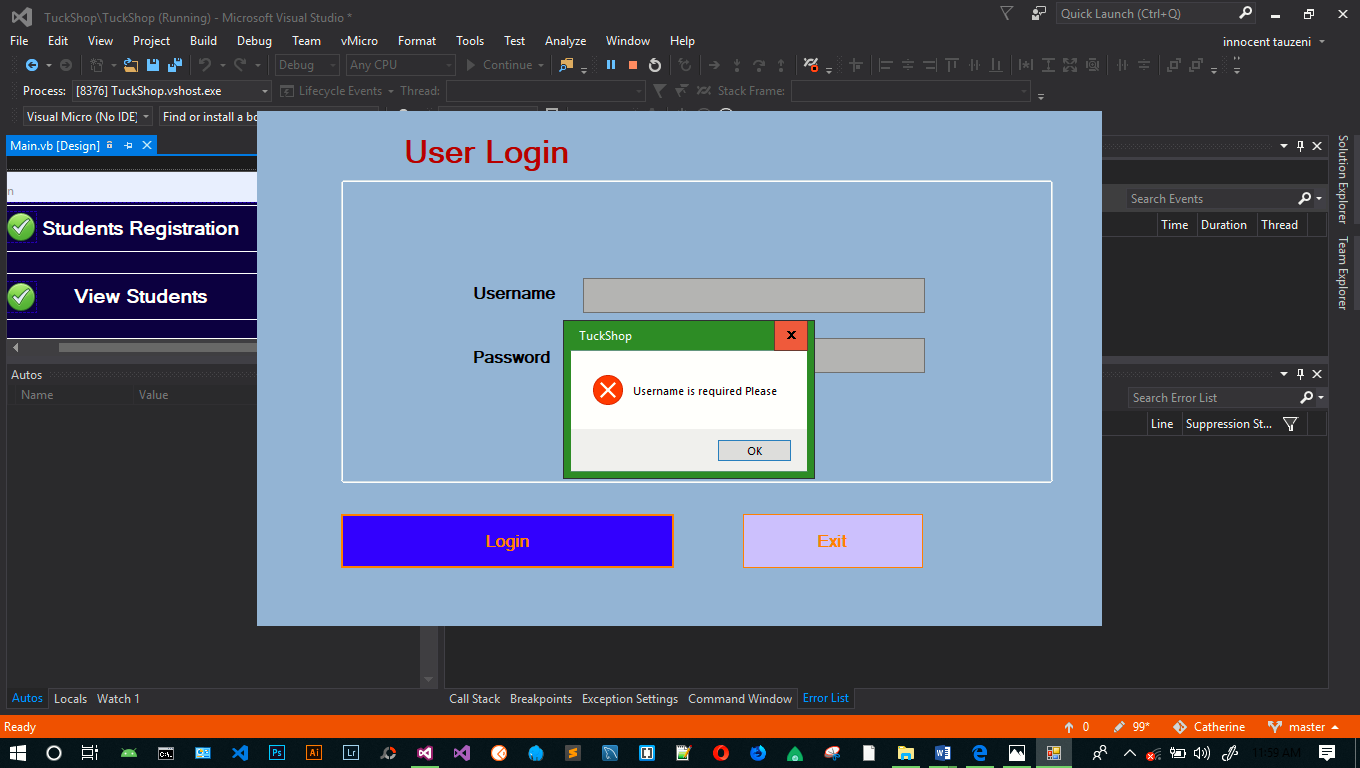
****

Figure 18 showing the results of missing information

**Display data on forms**

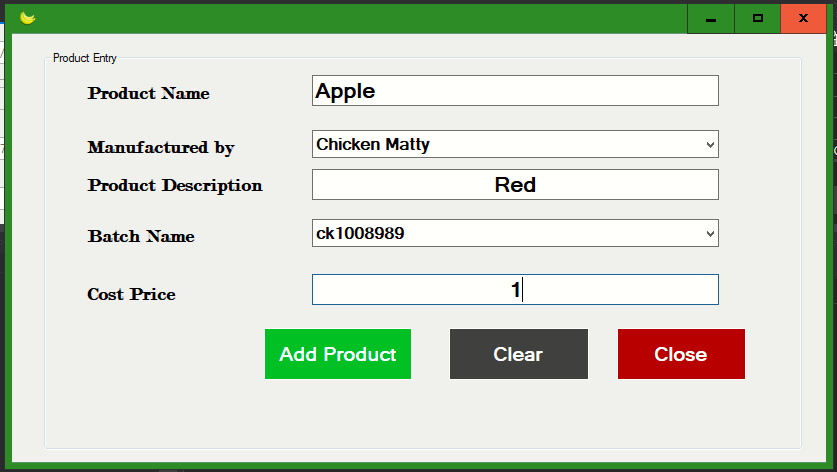


Figure 19 Displaying data on the forms

**Displaying of the output from the database**

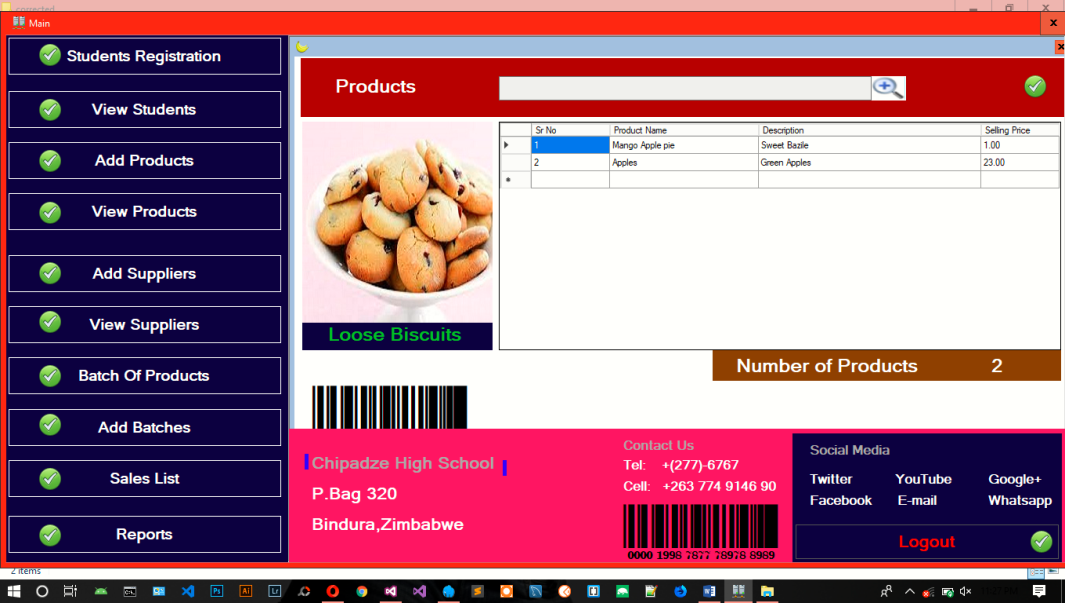


Figure 20 Displaying of the output from the database

## 4.4. User testing

User testing was guided by the objectives of the project defined in Chapter 1 as follows:

* **Adding student details to the system**

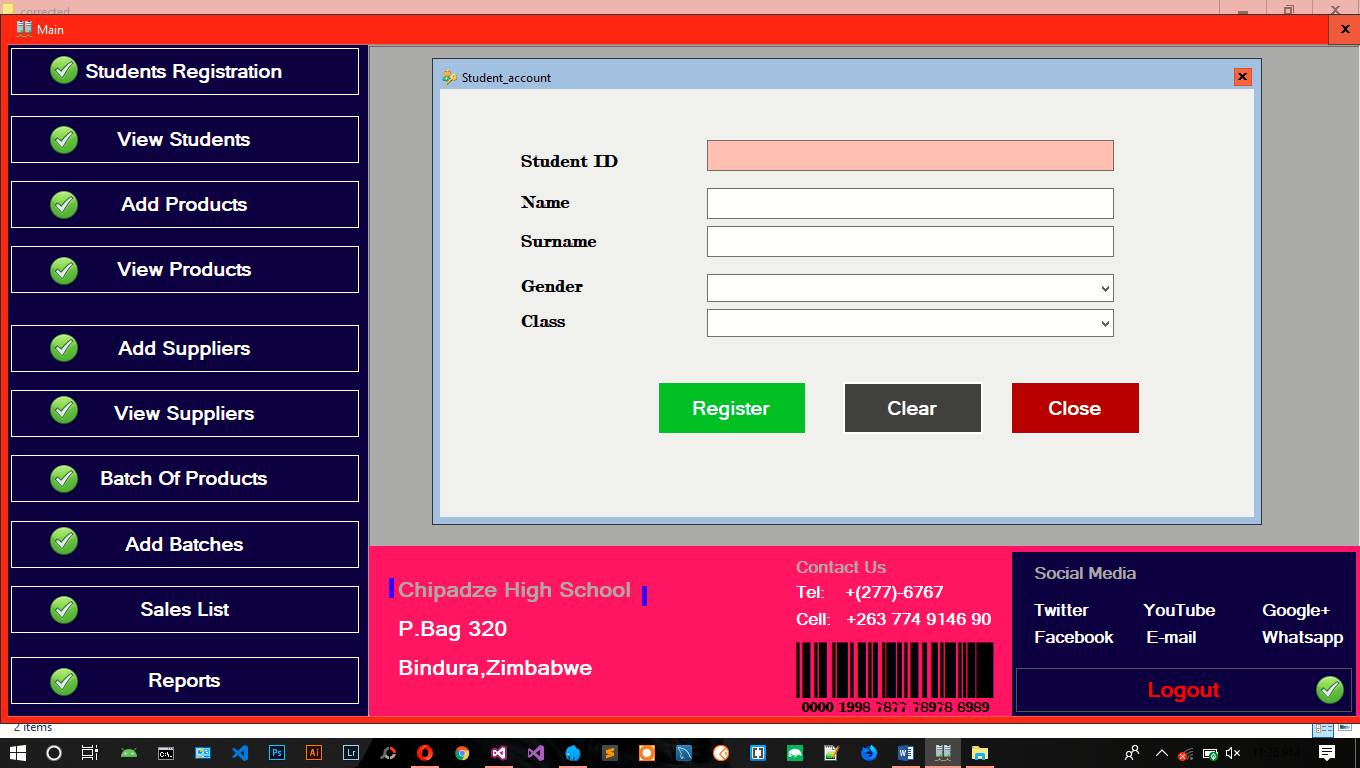


Figure 21 Adding student details to the system

**Adding details of new product in database with the help of unique id**

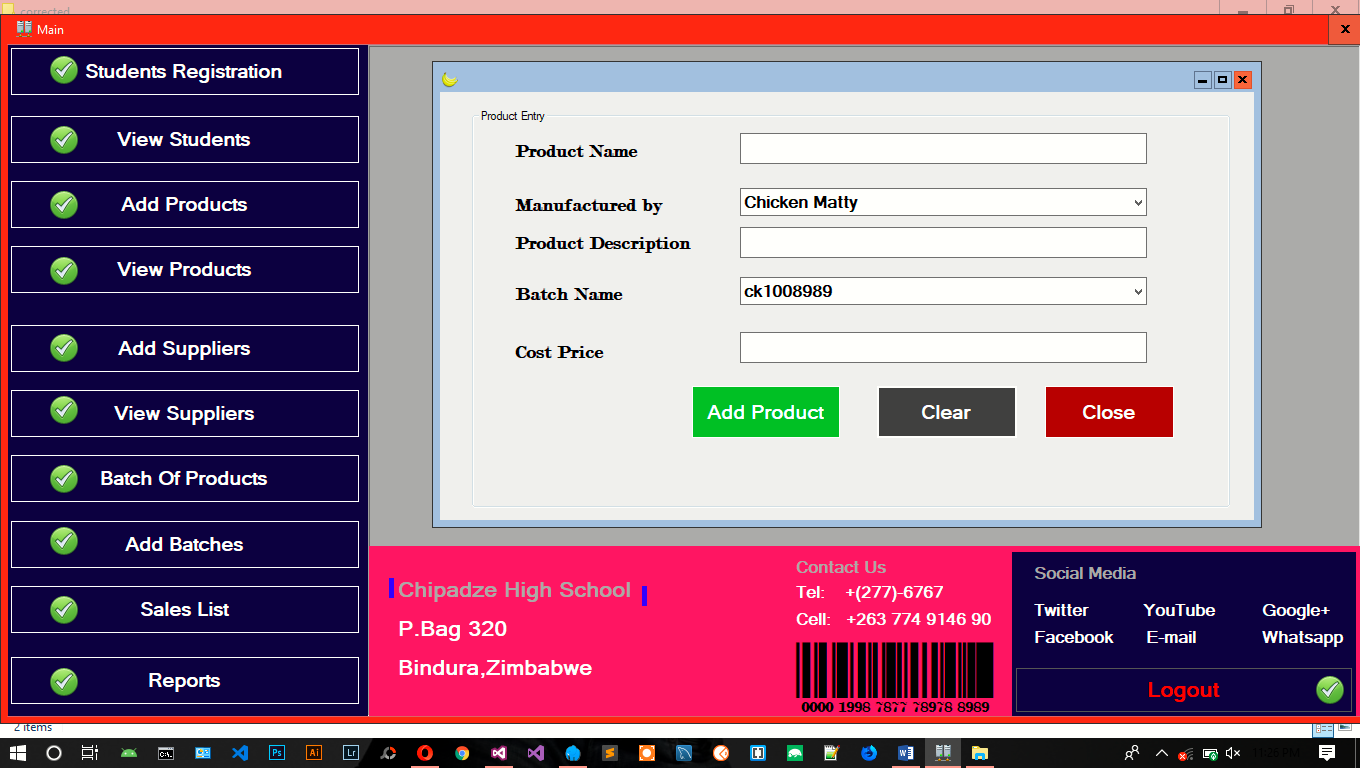


Figure 22 Adding details of new book in database

## 4.5. Conclusion

This chapter presented the code for the system for various components and functionalities. The chapter also presented the unit and the system tests and the test results that are performed by the system. The next chapter focuses on the implementation of the system.

# Chapter 5: Implementation

## 5.1 Introduction

This chapter will outline the implementation / deployment of the tuck-shop management system at Chipadze High School. Details of the conversion plan shall also be discussed. The user manual and user training plan will also be given

## 5.2 Conversion Plan

System changeover is concerned with the smooth shift from one way of doing things to another and the mitigation of disruption to business activities during the changeover (Banerjee, 2017). It entails changing from the old system to the new one. Chipadze high school tuckshop needs to change from the manual paper work and log product system to the new automated tuck shop management system. There are several methods which a systems analyst can choose from depending on which one best suit the organisation.

Direct changeover, also referred to as immediate replacement, tends to be the least favourite of the changeover techniques. In a direct changeover, the entire system is replaced in an instant. Basically, as soon as the new system is powered up, the old system is shut down. (Robertson, 2014) This type of changeover carries the most risk because; if something goes wrong, reverting back to the old system usually is impossible. Using the direct changeover technique tends to work best in situations where a system failure isn't critical enough to result in a disaster for the company.

For this project, the Chipadze high tuckshop system will use the parallel conversion plan in which the manual system will continue to be used in parallel with the new automated system. All of the data and information which are used as input into the old system may be used as input into the new system (Banerjee, 2017). Results of the new system is verified with old system and also verify the check list of products which are prepared previously. Ultimately, the old system or module is stopped working only when the new module works successfully. The advantages of using the parallel changeover include, its potential for continued smooth operations during the changeover period. The new system can be run alongside the old system for weeks or months, allowing the discovery of any problems in the new system without exposing the business to a shutdown when problems occur. This feature is especially useful because we are adopting a new untested system that may require unforeseen specialization and optimization after implementation.

## 5.3 user manual

a) Installing the software

Insert the disk that came with the packaging. Open the folder entitled Chipindura High library installation. Double click on the set up so that the system is set up in your computer

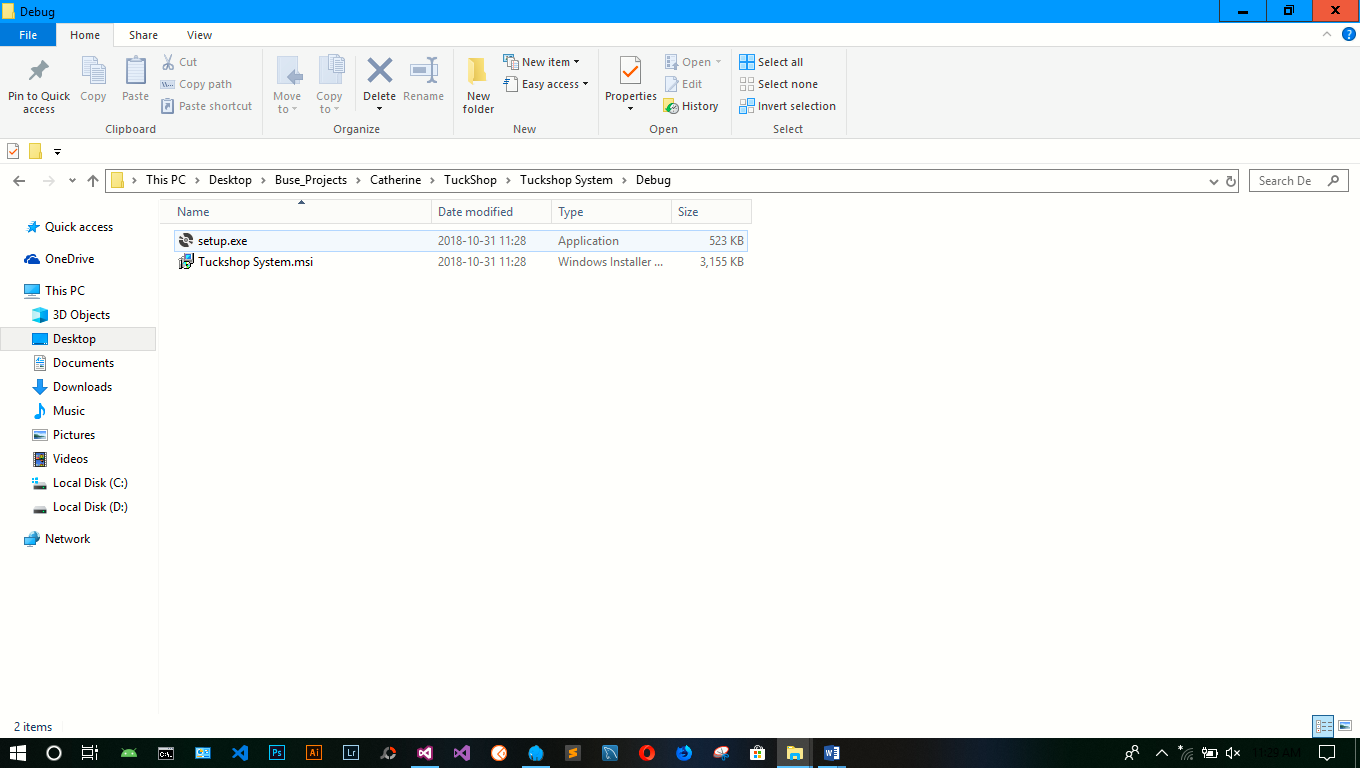


Figure 23 installing the software

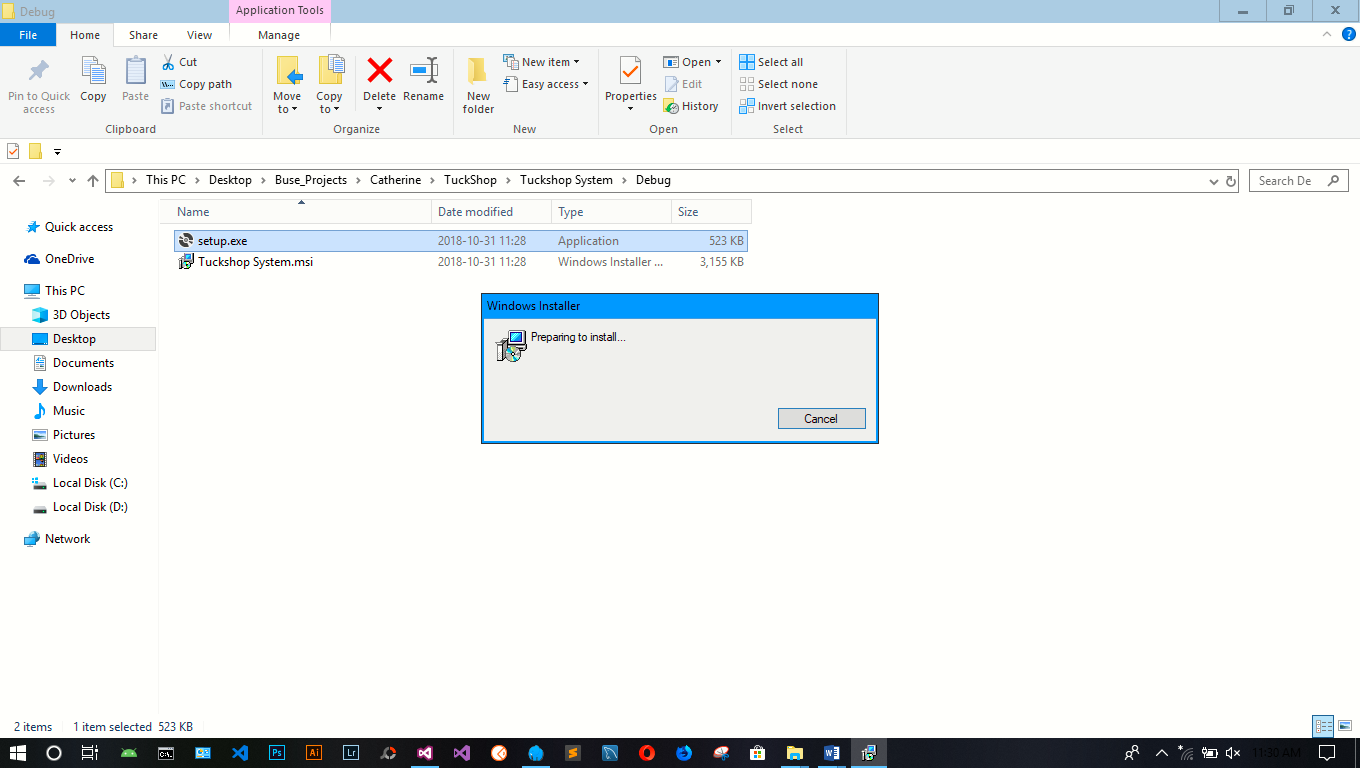


Figure 24 System is now installing

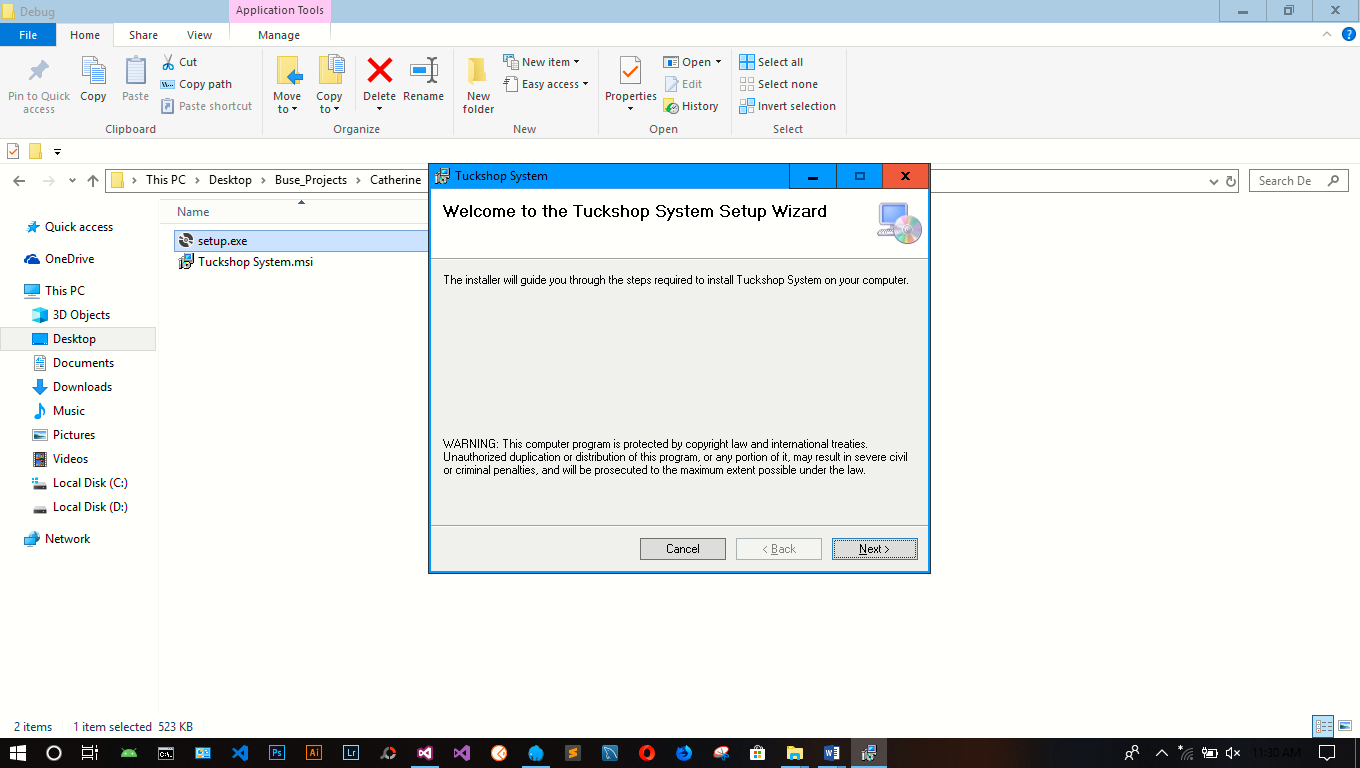


Figure 25 Choose next

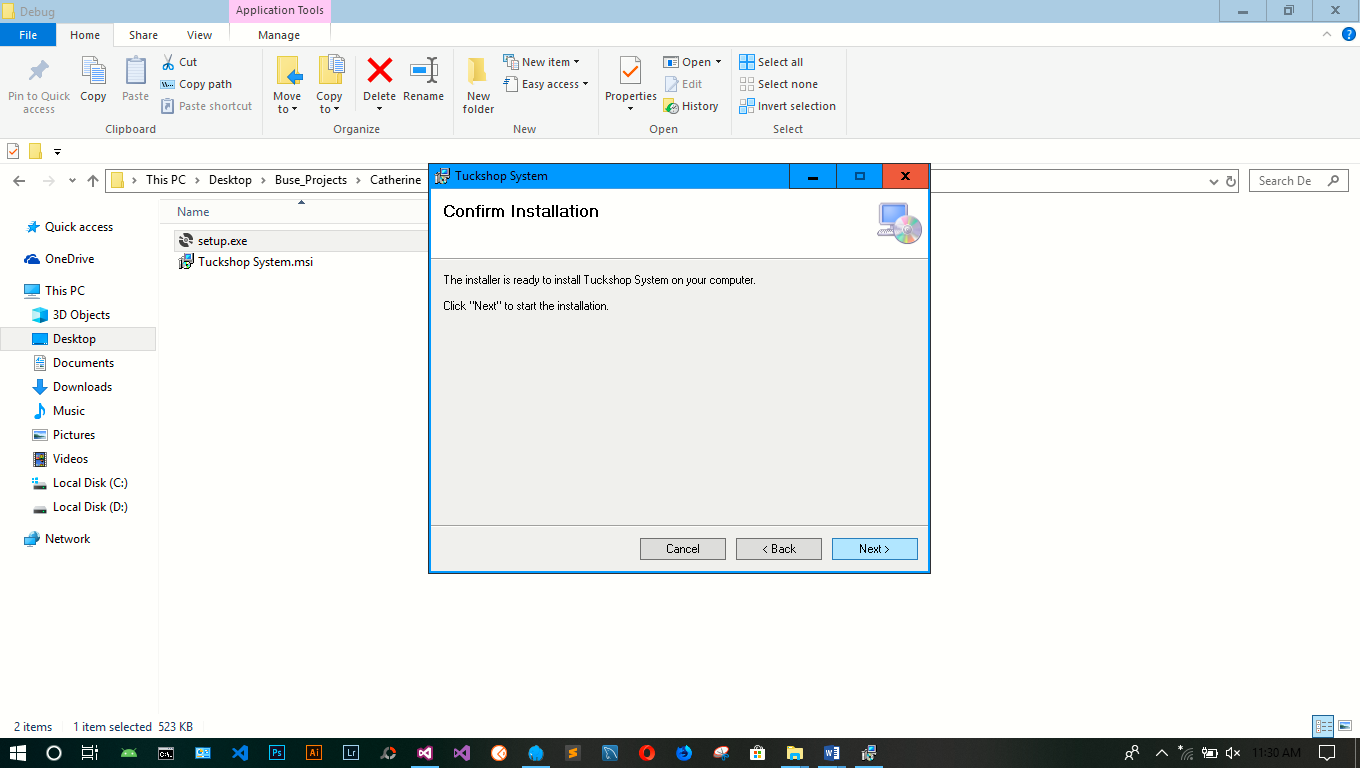


Figure 26 Installation completed successfully



Click This Icon

Figure 27 Click Icon on the Desktop

## 5.4 User training plan

The author shall schedule a two day training workshop for all the users of the system which includes; the storekeeper and administration staff. The system users shall be trained on system data and way around the system.

# Chapter 6: Evaluation and conclusion

## 6.1 introduction

This chapter will evaluate the system and give the conclusion of the automated tuck-shop management system.

## 6.2 Evaluation

### 6.2.1 Achievements

The developer was able to create a system that is able to

* automatically search for products,
* update the products
* highlight products that are overdue
* keep unauthorised users out by use of password verification
* The system is user friendly as it allows interaction between the user and the system.
* Data security was achieved as the databases for products and borrowers are not accessed by unauthorised users.
* The system meets user expectations since there is minimum hardware requirements and costs to implement the system.

### 6.2.2 Limitations

Despite having made a number of achievements, the systems still have the following weaknesses;

* No barcode reading products
* The school administration has no control of the system, the shopkeeper can lock them out
* File encryption is not yet possible.
* The system also failed to maximise its security ability as access rights were not fully implemented into the system.

## 6.3 Opportunities for further development

The developer can add the following features to the system;

* Printing of receipts for penalty payments
* Have an administrator account that overrides the shopkeeper and can track every transaction within the system and print reports.

## 6.4 Conclusion

The research was completed and the researcher was able to create a tuckshop management system that is able to make accurate calculations on items purchased, costs and items sold, through the use of formulas, Validate input purchases, sales and order figures, Update stock levels in the stock details file when item are purchased or sold and function according to user expectations.

# References

**There are no sources in the current document.**