



BuzzWire

by Stefanie Garises 12 HIGSCE

Index

1. Introduction	Page 3
2. Problem	Page 3
3. Objectives: General Terms	Page 4
Computer Related Terms	Page 5
4. Flow Charts	Page 7
5. Description of Existing Solution	Page 10
6. Questionnaire	Page 11
7. Evaluation of Existing Solution	Page 12
8. Description of Other Possible Solution	Page 13
9. Action Plan	Page 15
10. Modules: Database	Page 16
11. Modules: Program	Page 19
12. Hardware Requirements	Page 23
13. Software Requirements	Page 25
14. System Flowcharts: Searching for member details	Page 26
Customer makes a purchase	Page 27
Add new member	Page 28
Calculate total	Page 29
Calculate Amount_Owed	Page 29
15. Design Method of Solution	Page 30
16. Testing	Page 31
17. Technical Documentation	Page 35
18. User Documentation	Page 39
19. Evaluation	Page 44
20. Opportunities for Development	Page 45
21. Program Code	Page 46

Introduction

I have been assigned to **Buzzwire**. It is a shop which specialises in the sale of gadgets that are used in everyday life. It was established on the 01 January 2010 in Windhoek, Namibia. Specifically gadgets like cell phones, laptops, printers, etc. this store was established for people who cannot afford the average gadgets which are available in the town. It has different departments such as Laptops, Phones, Routers, etc. The store has two employees and recently purchased a new computer.

Problems

- The employees do not use the computer often because they are so used to the old system
- The computer does not have the software necessary to perform the necessary functions in the store
- The employees do not know how to use the computer
- Documents are often lost when using the old filing system
- A lot of money is spent on paper
- Because transactions are so frequent, the employees cannot sort the documents fast enough
- The files look boring and disorganised which makes it difficult to find documents in them
- It is very time consuming to sort the files
- The stock is unprotected and some of it always goes missing
- Can only reach local customers
- Limited advertising due to cost

Objectives

General Terms

- Computerise system
- Employ people to improve customer care and lighten the work load
- To speed up the process of entering, editing, adding and removing data
- Allowing entries into database from different work stations
- To have a neater system
- To attract more customers
- To attract more suppliers
- To advertise more effectively (e.g. Print media (Newspapers, Posters, Hand-outs, etc.) Electronic media (website, TV Ads, Radio, etc.) Other media (Billboards, Mobile solutions, etc.)
- to have security cameras for the safety of the customers and my stock
- to have random audits financial as well as system audits to prevent fraud
- to create a clean and healthy working space for the staff
- to create competitions for visibility purposes
- to conduct interviews to find out customer preferences and design tailor made products suiting the market demands
- to attract more customers by adding value to products (e.g. 2 for 1 sale)
- to install a multi access system so that employees can access all data simultaneously
- to install stock control software
- to save resources/Being cost effective
- to save time/allow computer to do small time consuming tasks
- to ensure steady business growth and continuity
- to allow online backups (e.g. Cloud)
- to be on par with current trends

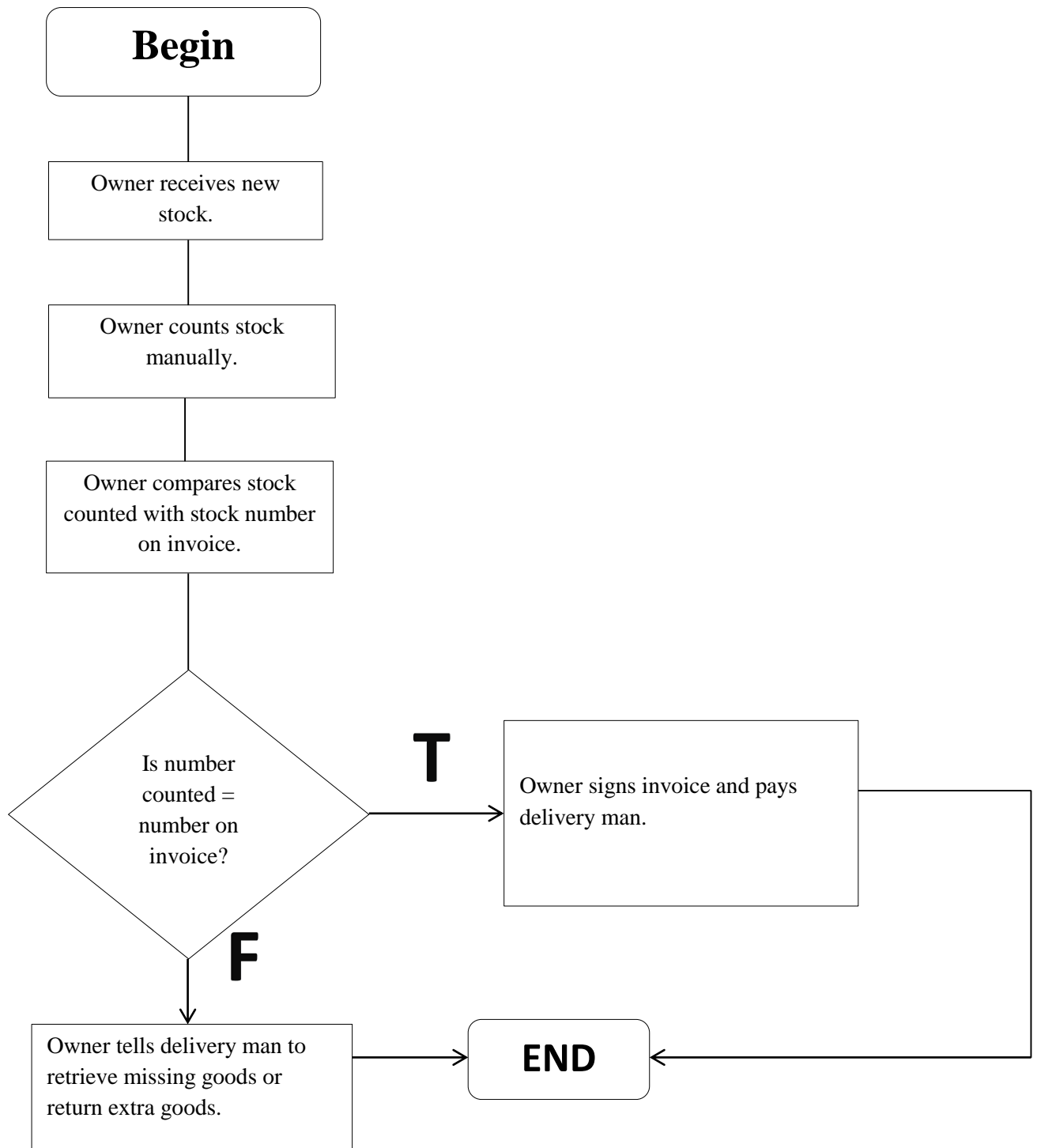
Computer Related Terms

- To create a relational database so that all the data is easily linked and found
- The database will contain the following tables:
 - Members
 - Suppliers
 - Stock
 - Transaction File
- To create fields for these tables
- The stock table should contain these fields:
 - Item_Type
 - Item_Code
 - Quantity
 - Cost Price
 - Selling Price
 - Total
- The members table should contain these fields:
 - MemID
 - Surname
 - Name
 - Gender
 - DOB
 - CellNo
 - Address
 - Email
 - AccNo
- The suppliers table should contain these fields:
 - SuppID
 - Name
 - Tel_No
 - Fax_No
 - Email
 - Physical Address
 - Postal Address
 - Acc_No
- The Transaction File table should contain these fields:
 - Date
 - Item Type
 - Item Code
 - Price
 - Quantity

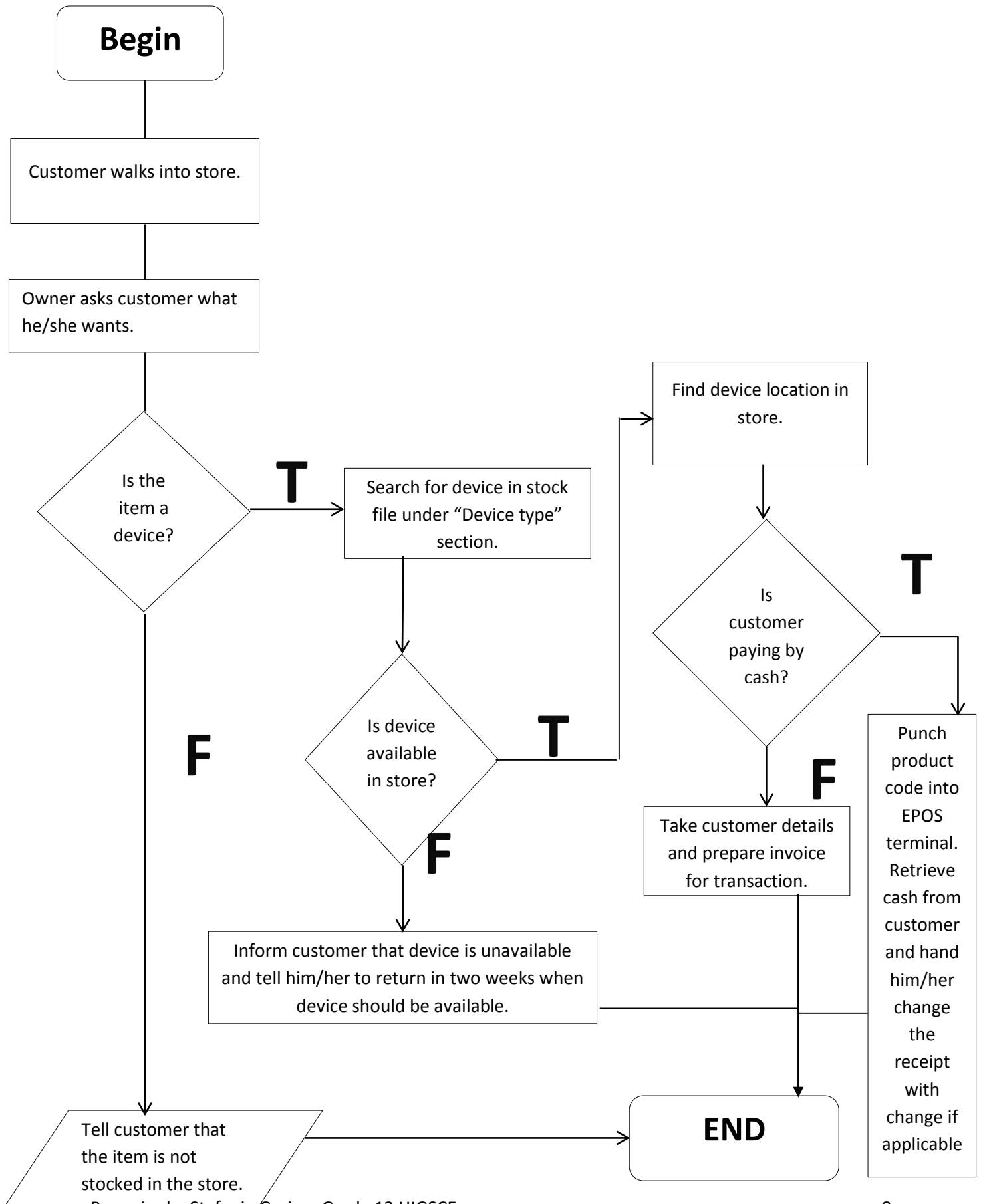
- Total
 - Amount Received
 - Change
 - Receipt No
- to design a program which is completely computerised and efficient
- to design a user-friendly interface for the employees to use
- to design a program that fully links all the tables in the database to the program
- to design the program so that it is stimulating to the eye
- to design a program that collects data on members, suppliers and stock
- the program should be able to prepare custom invoices and receipts for the customers
- to design queries to find data
- to edit, add, remove and save data
- to make the system run smoothly
- to design custom membership forms for Buzzwire in a programming software
- to search the database
- to create sales records
- to enable real time/near real time actions

Flow Charts

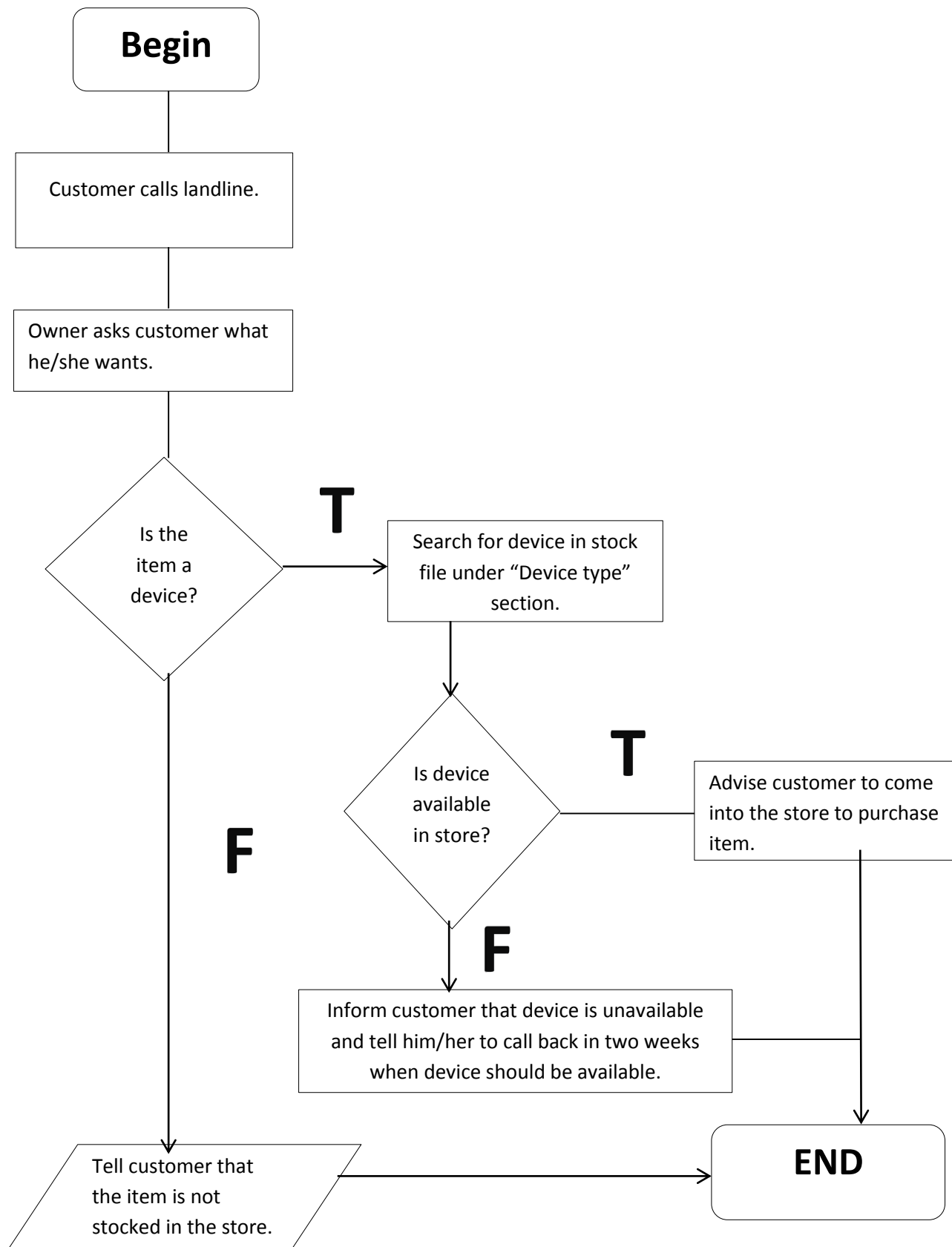
1. Stock Retrieval



2. Cash/Credit Sale



3. Telephone Call



Description of existing solution

The existing solution is that the employees file and sort the documents received and created by business. They sort the documents into chronological order to their best ability. There are three files. There is a file for members, a file for transactions and a file for stock. There is a separate set of files for the accounting side of business. The stock file is divided into sections. These sections are the names of the suppliers of the business as well as the product types in general. The product type sections are seven in total. These are the "Mobile", "Printer", "Laptop", "PC", "Smart watch", "Gaming" and "Router" sections. It is a problem to sort these files because the documents come in very frequently and they cannot match that speed. This causes disorganisation. This causes the employees stress. This leads to substandard work. This is bad because it decreases the amount of customers frequenting the store. As a result, fewer sales are made and the business finds it difficult to cover its costs. Another problem is that some documents get lost. The existing solution is that the employees then search for the lost document until they find it. This is obviously a frustrating task. Doing these tasks manually allows frequent errors. The business can only reach local customers due to lack of advertising. The lack of advertising is due to the lack of money to cover the advertising costs. At this stage the business can only afford to place a small advertisement in the local newspaper.

Proof of fact finding

- The store handed out 100 questionnaires of which only 40 returned and only 35 returned fully completed.
- An observation, of how the business works, was conducted over a period of 2 weeks in the beginning of January.
- Several interviews with the owner of BuzzWire and the general public were conducted.
- All existing paper work was studied over 4 weeks in January.

Questionnaire



This questionnaire is meant for professional purposes only. The recipient of this questionnaire does not have to state his/her identity. Tick the box which suits you.

1. Have you seen the newspaper ad for Buzzwire?

Yes ☐ No ☐

2. Have you ever purchased an item in the Buzzwire store?

Yes ☐ No ☐

3. What do you think about the quality of Buzzwire items?

Excellent	<input type="checkbox"/>
Good	<input type="checkbox"/>
Quite good	<input type="checkbox"/>
Poor	<input type="checkbox"/>
Bad	<input type="checkbox"/>

4. How often do you purchase items from Buzzwire?

Daily ☐ Weekly ☐ Monthly ☐ Yearly ☐

Thank you for answering this questionnaire.

Have a nice day.

Evaluation of existing solution

Advantages

- The employees know exactly what is going on in the business and they can easily refer back to any document
- It encourages the employees to offer proper customer/after sale service
- In the event that there is no electricity they will still be able to function properly and no data will be lost
- The lack of computers allows more employment for people
- In the event that there is an error the employees can correct it him/herself. A computer may not notice every error let alone be able to fix it.

Disadvantages

- It is paper intensive
- The filing and sorting is boring and repetitive
- If an employee is absent for any apparent reason the work assigned to that person will either become the responsibility of the other employee or it simply won't be done.
- There may not be a customer everyday
- Employees may lose some documents making it a difficult task to reconcile documents in order to make statements for customers.
- Sorting and filing wastes valuable time which could be spent doing something more productive.
- There is no way to protect the stock so that it doesn't go missing
- The form and scale of advertising is too weak to attract more/international customers

Improvements

1. The store can hire an employee to arrange documents into alphabetical and chronological order and complete the periodic tasks (e.g. Stock taking, etc.) while the other focuses on the more admin orientated tasks (e.g. Dealing with invoices, etc.) and the third employee can deal with customer service.
2. The store can collect data on a larger scale by sending out more questionnaires asking relevant questions and conducting interviews with people who have an interest in the store and the products sold therein.
3. The store can have CCTV cameras installed to monitor the employees and protect the stock and the customers when they are in the store.

Description of other possible solutions

A: The business can hire several employees to assist them by doing the boring and repetitive tasks.

Advantages

- Customers still get attention which helps increase sales.
- The work will be less stressful for the employees creating a productive atmosphere in the store.

Disadvantages

- It will be expensive to hire more employees
- The salaries for these employees might be drawn from a part of the store which really needs the money.
- Sales and customer frequency may not necessarily increase
- It still doesn't solve the fact that paper is wasted and stock goes missing

B: The store can create a turbo Pascal program.

Advantages

- Does not utilise all the available space in the computer system
- There are many hardware requirements
- It will be cheaper for the store

Disadvantages

- It will have no effect on the amount of customers frequenting the store
- It will have limited graphics and graphics make the system easier for any employee to use
- The program cannot be linked to the internet and the website will be boring

C: Computerise the store by using a database and Visual Basics 6.0

Advantages

- Has a lot of graphics and will be easier for employees to use
- Documents will not be lost because they will all be computerised
- Time will not be wasted sorting and filing documents
- The website will be used to advertise the store and allow customers in other towns/countries notice the store and purchase items using the online store.

Disadvantages

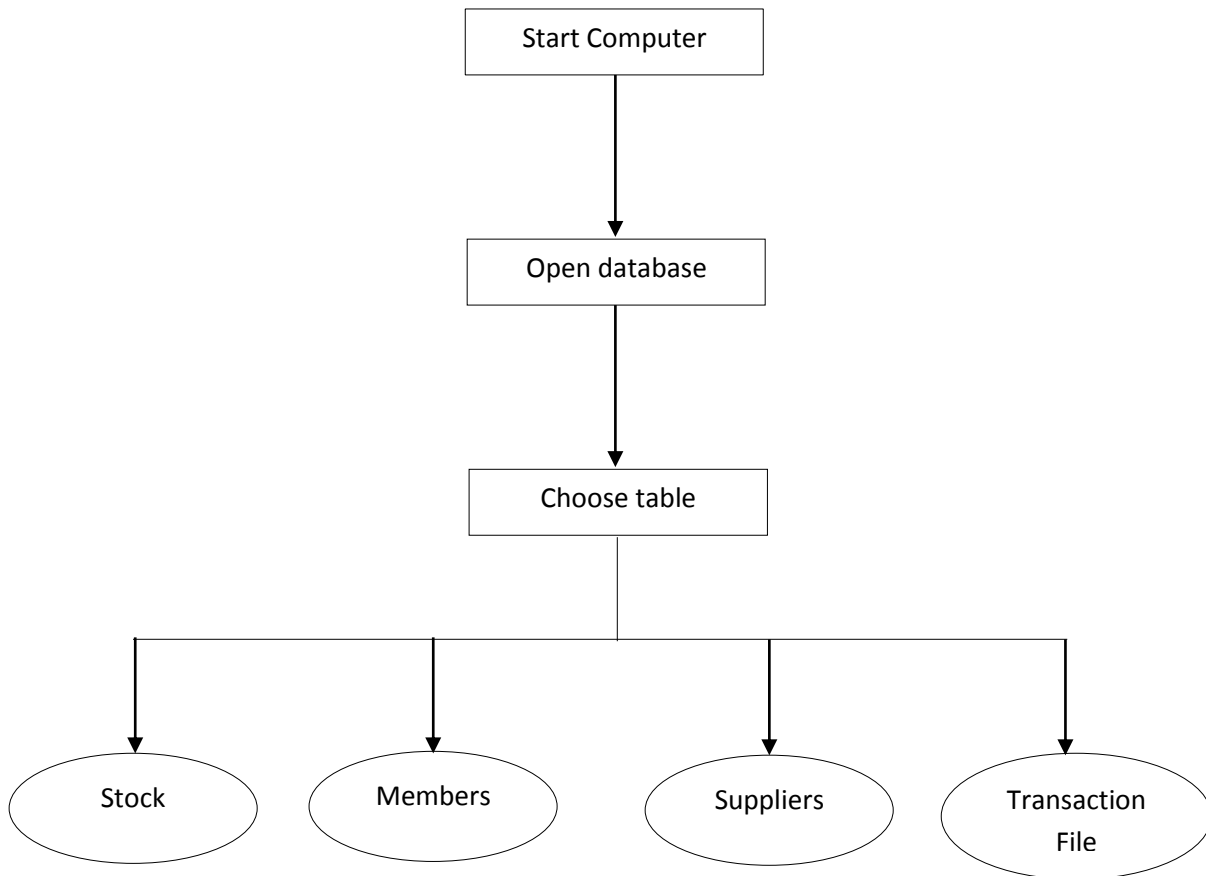
- Data will be lost if there is an electrical outage
- A hacker can gain access to the stores records and change them or view confidential data
- There will be a lot of hardware requirements
- A virus can penetrate the system and destroy the data

I choose solution C.

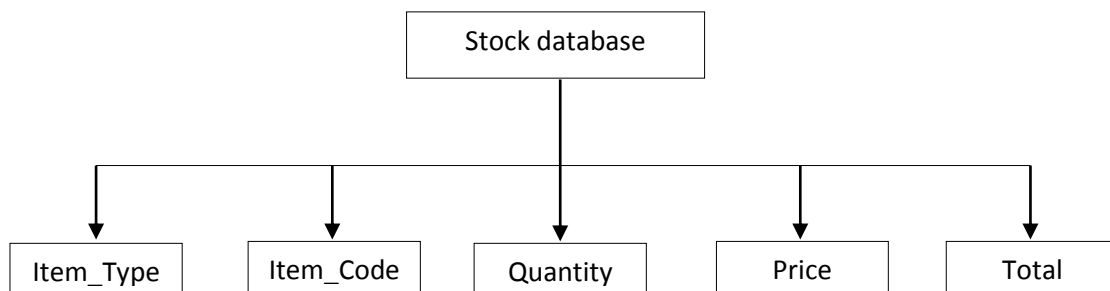
Action Plan

Task	Description	Time	Done	Real Time
Feasibility study	Terms of reference: objectives	4 weeks		
	General objectives	2 weeks		
	Computer related objectives	2 weeks		
Analysis	Collecting data through observation, questionnaires, interviews and looking at existing files.	4 weeks		
	Observation	2 weeks		
	Questionnaires	2 weeks		
	Interviews	2 weeks		
	Looking at existing files	4 weeks		
Design	Design input forms, screen layouts, reports, flow charts, validation rules, etc.	2 months		
	Data input forms	2 weeks		
	Screen layouts	2 weeks		
	Output forms	2 weeks		
	Flow charts	2 weeks		
	Validation rules	2 weeks		
	Select appropriate verification methods	2 weeks		
	File structures and tables	2 weeks		
	Hardware and Software requirements	4 weeks		
	Produce algorithms and program flowcharts	4 weeks		
	Test strategy/plan	1 week		
Implementation	Implement the new system.	6 months		
	Design user documentation and technical documentation	4 weeks		
	install new hardware and/or software	1 month		
	full system test	3 months		
	Staff training	1 month		
	Transfer paper files to new system	1 month		
	Change over to new system	2 weeks		
System Maintenance and Evaluation	Maintaining and evaluating the system.	2 months and 3 weeks		
	Compare final solution original requirement	1 week		
	Identify any limitations in the system	1 month		
	Identify any necessary improvements that need to be made	2 weeks		
	Evaluate the user's responses to using the new system	2 weeks		
	Compare test results from new system with result from the old system	2 weeks		
	Compare performance of new system with performance of old system	2 weeks		
	Update hardware as new items come on the market	N/A		N/A
	Update software	N/A		N/A

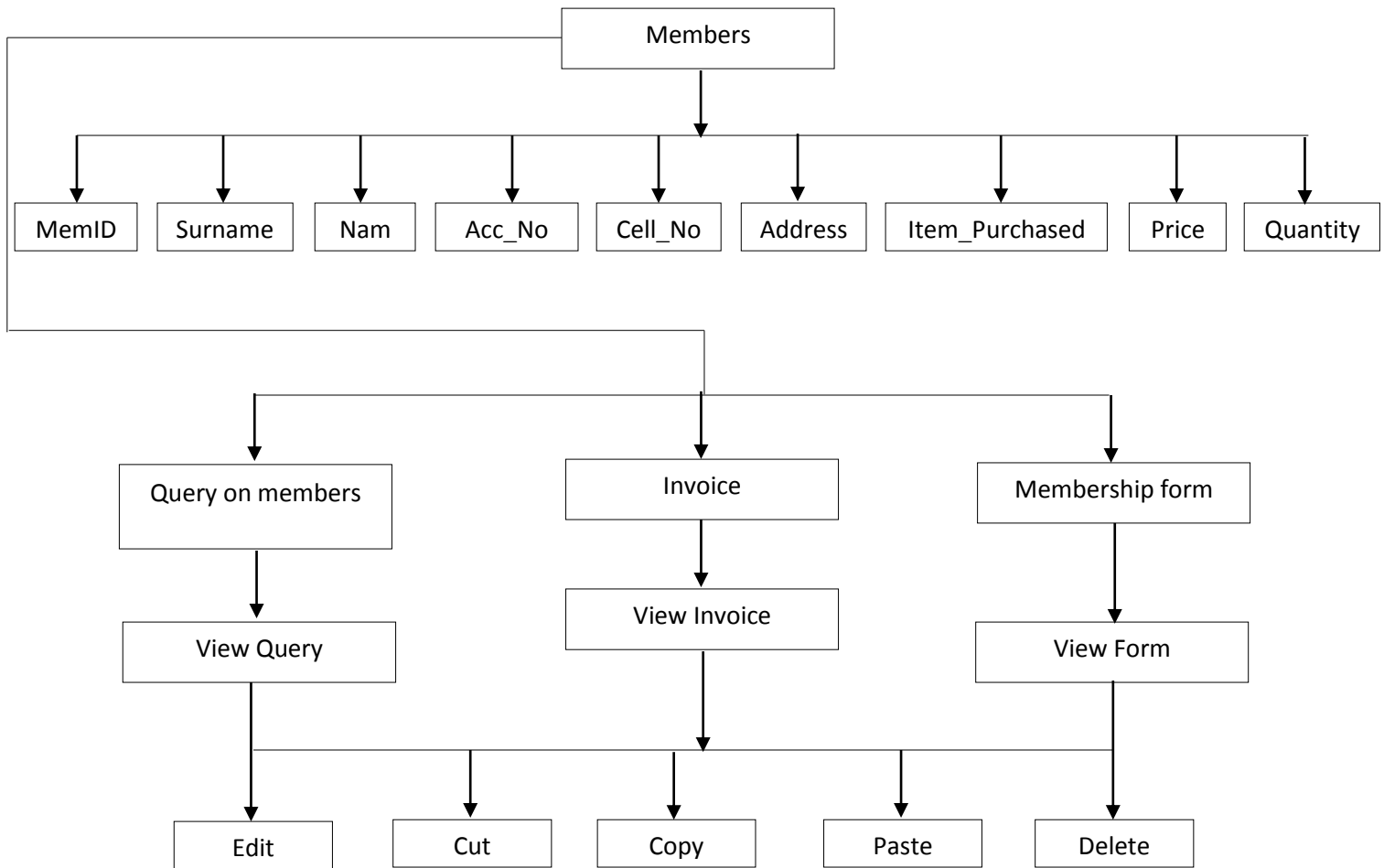
Modules: Database



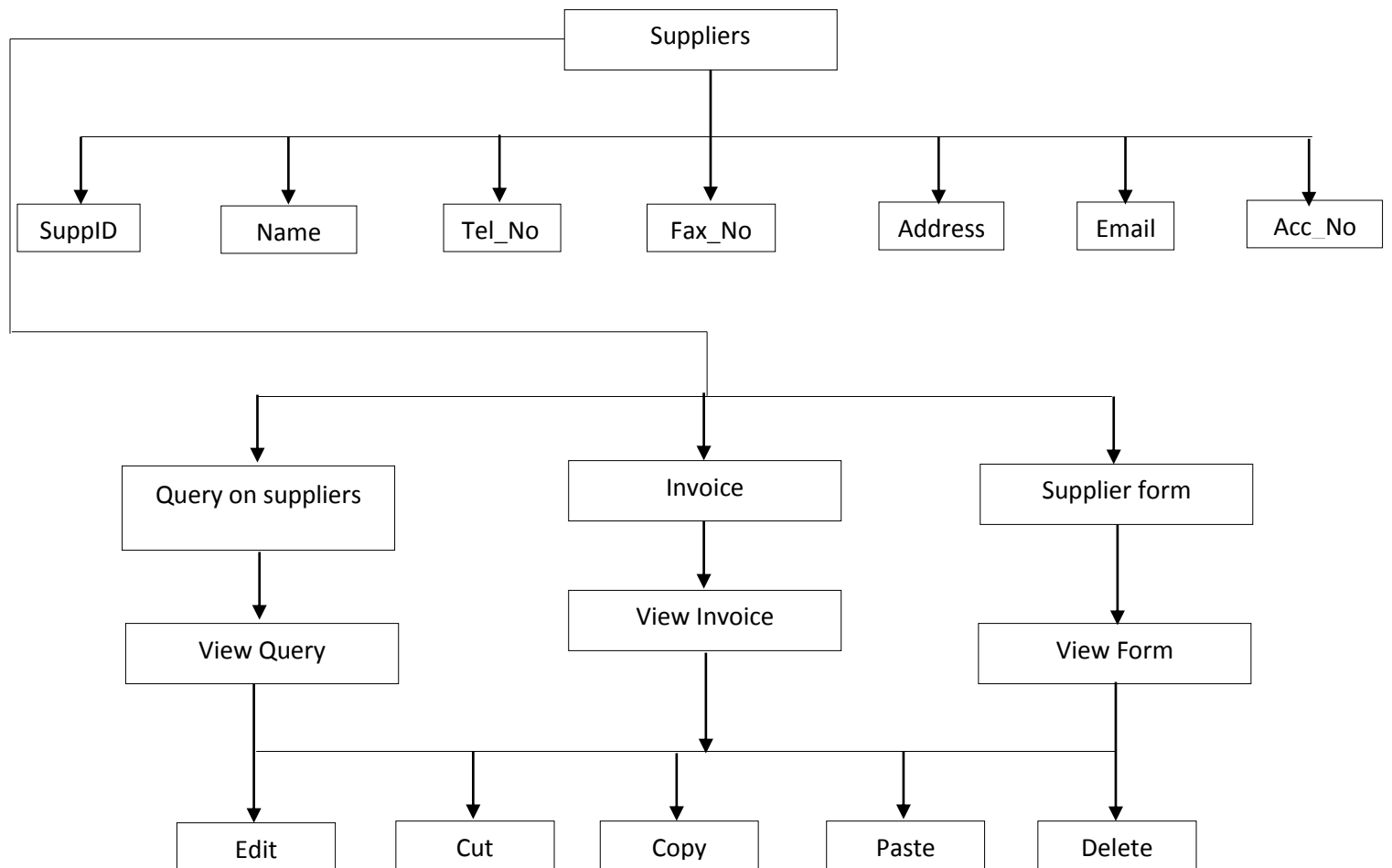
Stock:



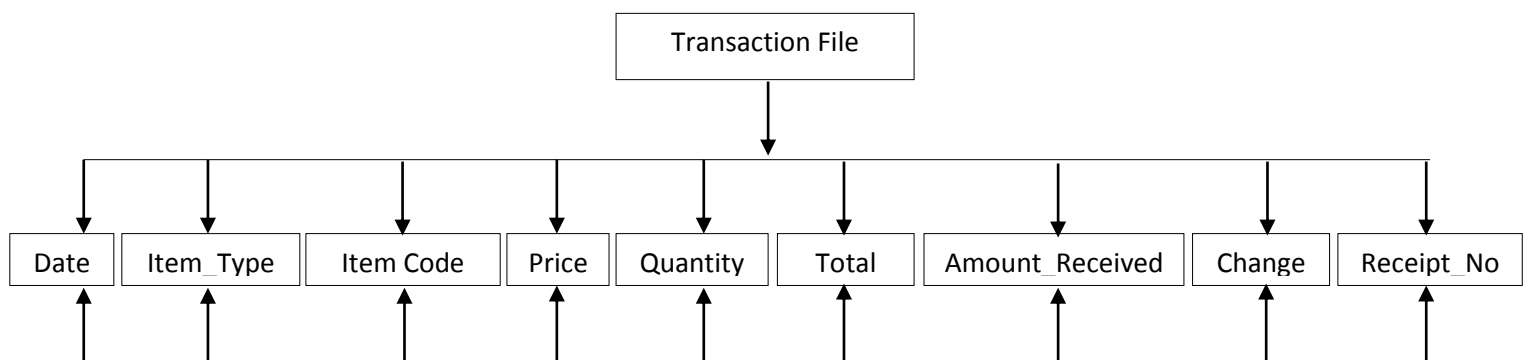
Members:



Suppliers:

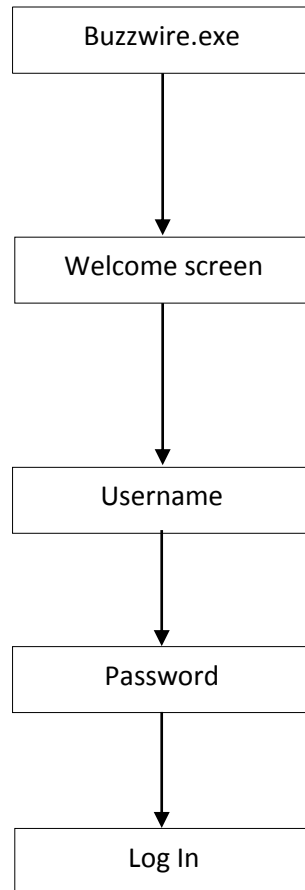


Transaction File:

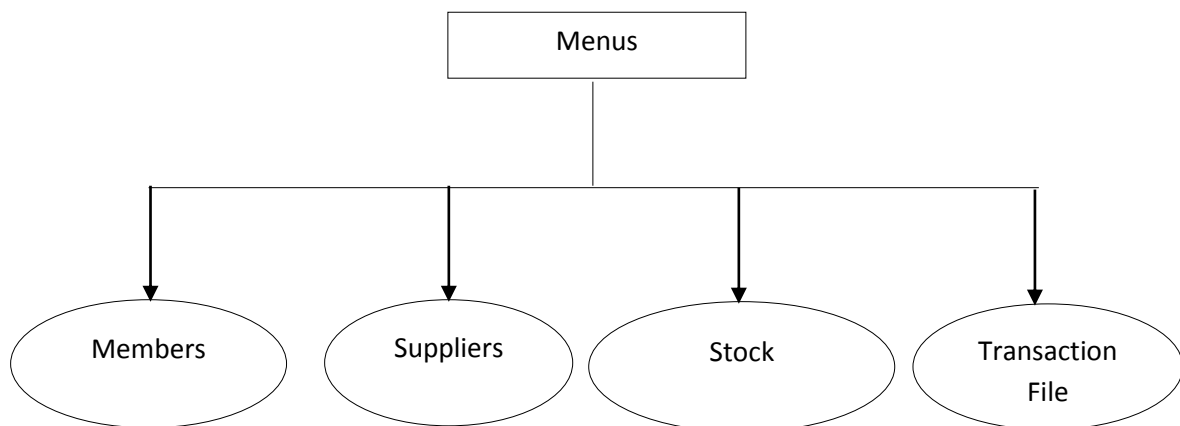


Modules: Program

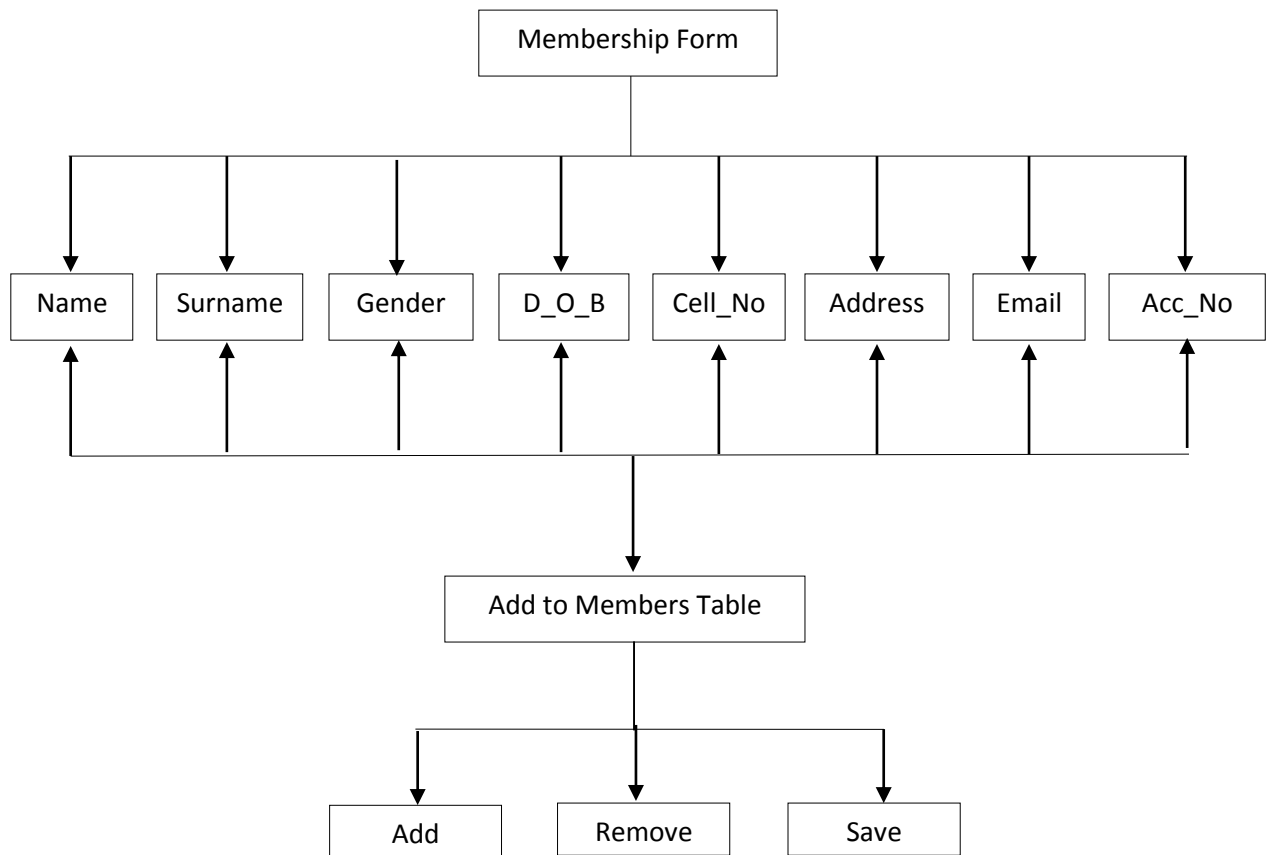
Log In:



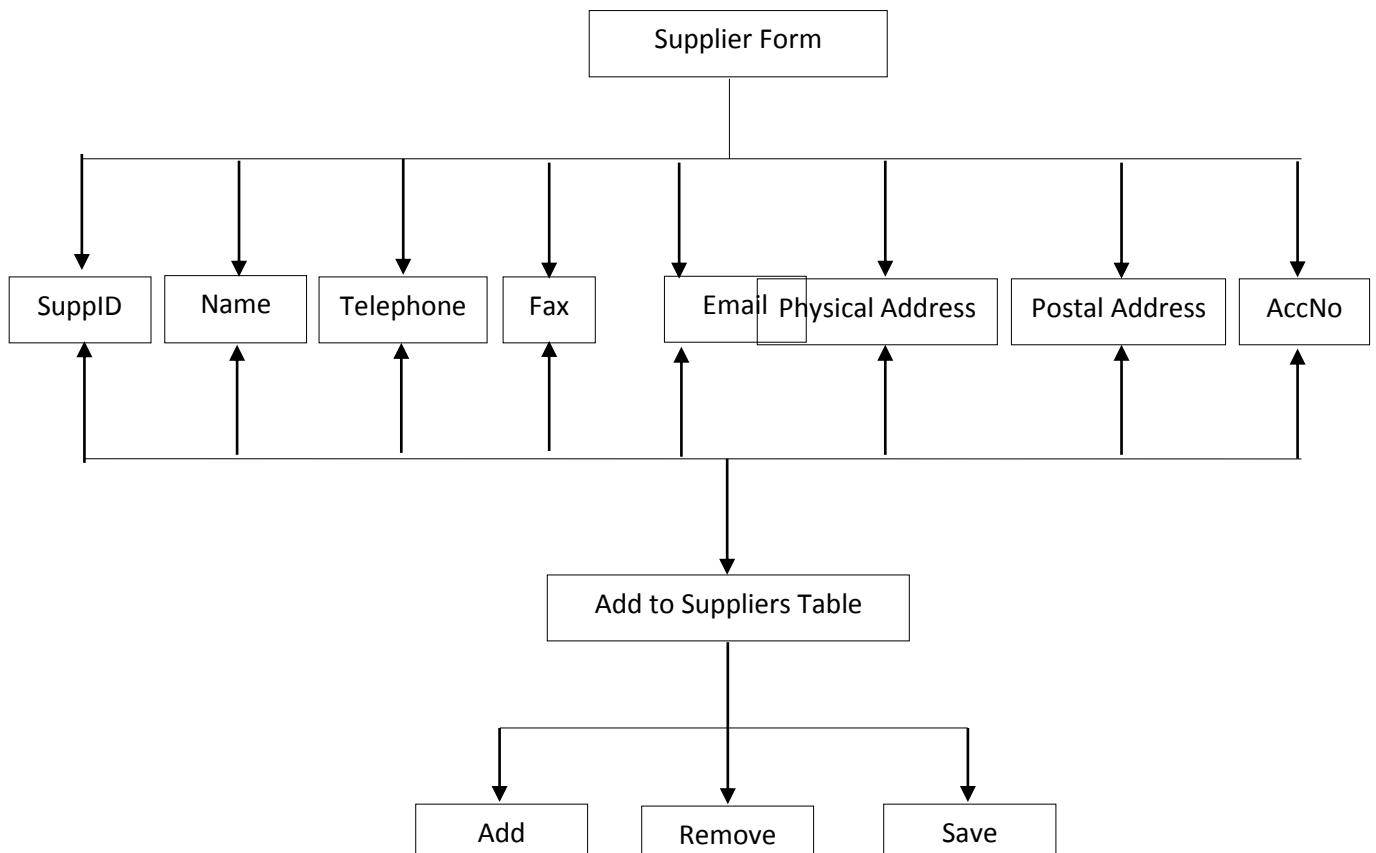
Menus:



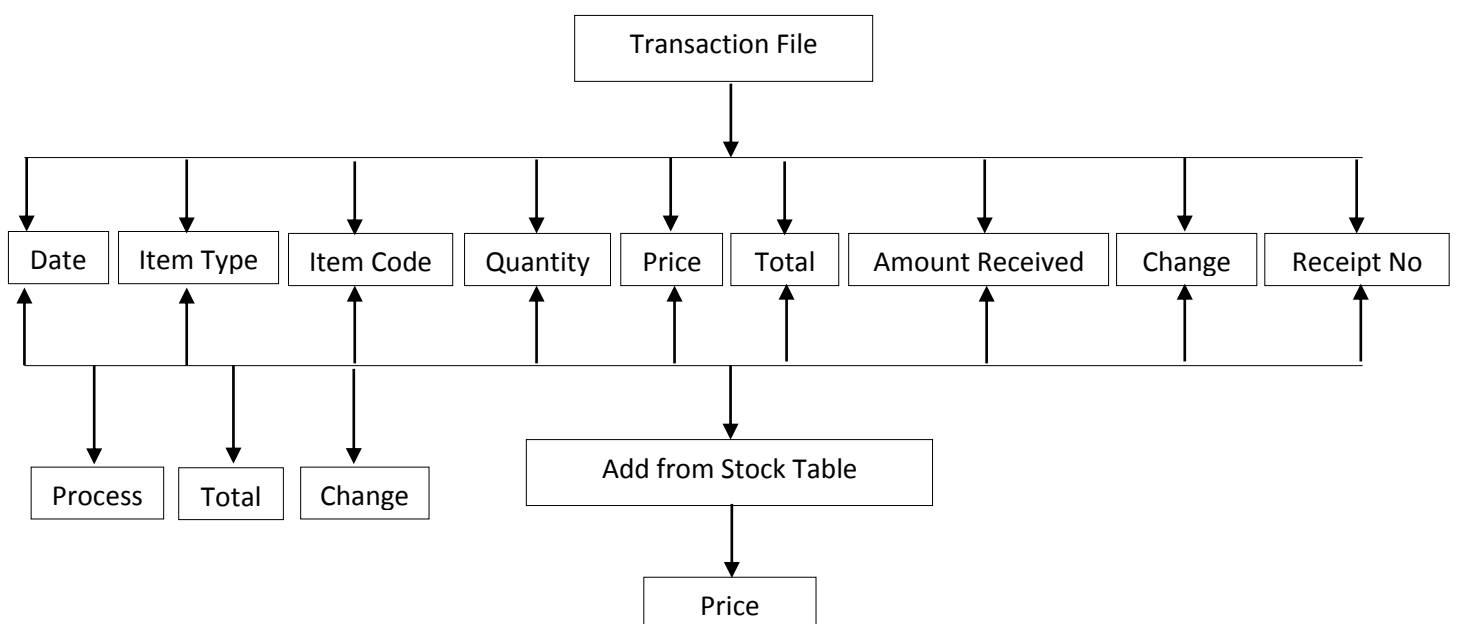
Membership Form:



Supplier Form:



Transaction File:



Hardware Requirements

- Monitor
 - To display the icons which are available for use
 - To display the design of the membership forms
 - To display supplier detail forms
 - To display stock control forms
 - To display the details of cash/credit transactions
 - To display the software of the new system
 - To display the pictures sent from the CCTV cameras and monitor them
- Keyboard
 - To input instructions e.g. leave a space between paragraphs in reports.
 - To input e.g. details of members, suppliers, transactions
 - To input e.g. the number of stock
- Printer
 - To print invoices for debtors
 - To print membership forms
 - To print company/customer notices
 - To print financial reports
 - To print receipts
- A LAN router
 - To have a network on which to store data which needs to be accessed frequently by employees
 - To have an internet connection in the store for research and communication reasons
- Server
 - To save data centrally
 - To do regular system backups of all the data thereon
 - To have the software available centrally for multiple users
 - To connect to the LAN network in the store where all software and data is kept centrally
- 2.7 GHz Processor
 - To process transactions faster
 - To reduce the 'freezing of computers'
 - To process queries faster
 - To load software faster
 - To boot the system faster
- 4 GB RAM
 - To make processing faster by providing more space on the temporary memory
 - To process more instructions at a time
- CCTV Cameras
 - To protect stock on hand by recording what happens in store

- To protect customers when they are in store
 - To monitor employees while on premises to prevent them from being fraudulent
- Landline
 - To allow the customers to call and make enquiries through the phone
 - To be in touch with customers who are out of reach
- Swipe Machine
 - To support electronic fund transfers in case customers have an insufficient amount of cash on hand when they want to buy an item.
- Access Cards with magnetic strips
 - To have staff rooms which are only accessible by authorised personnel/card holders
 - To restrict customers from certain areas in the store

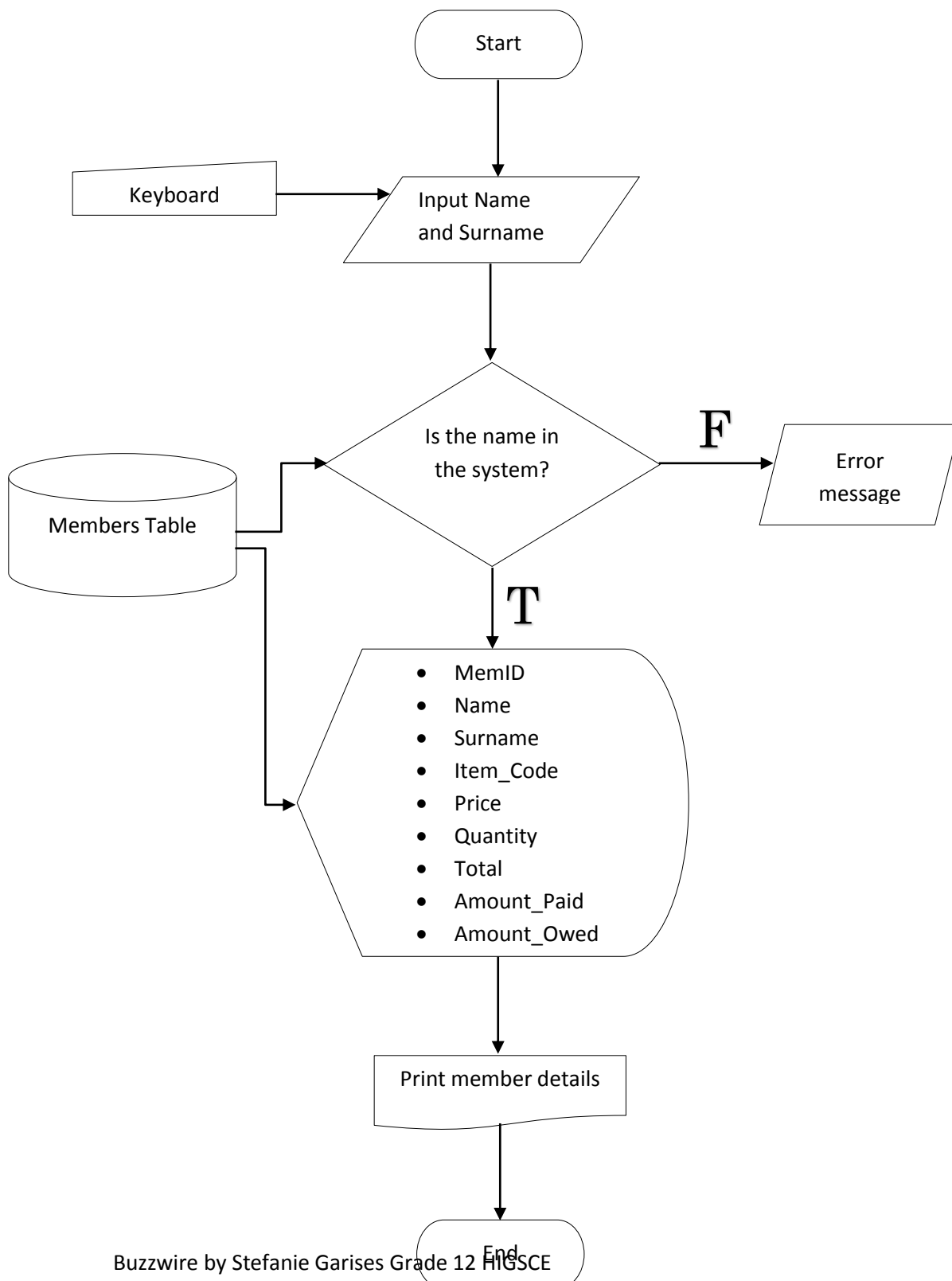
Software Requirements

- Anti-Virus
 - To protect system as well as data against viruses and hackers
 - To protect system from spam mail
- Word processor
 - To type letters/notices about the store or their debts to customers
 - To draw flow charts
 -
- Database
 - To create databases for stock, debtors, suppliers and EPOS
 - To sort/filter/edit data
 - To create relationships between tables
 - To design queries
 - To design expense reports
 - To amend data
 - To search for data
 - To add records to database
 - To link MS Access to Visual Studio
 - To create sales records
 - To create relationships
 - To keep track of all the stock received by the business
 - To keep track of the quantity of stock
 - To keep track of the prices of the stock
 - To reduce the chances of losing/misplacing stock
 -
- Spreadsheet
 - To design Gantt chart which shows the amount of time set out for every task involved in the development, installing and testing of the new system.
 - To record financial data electronically
 - To be able to add/remove/edit data
 - To do complex calculations with the options available
- Email Address
 - To redirect some of the traffic from the landline.
 - To be in contact with customers who are out of reach.
- Calculator
 - To calculate the stock
 - To calculate the change for customers
 - To calculate the total price for a customer's purchases
 - To calculate the total amount owing to a supplier
- Programming language
 - To design the membership forms
 - To design the next, add, clear and end buttons for the system

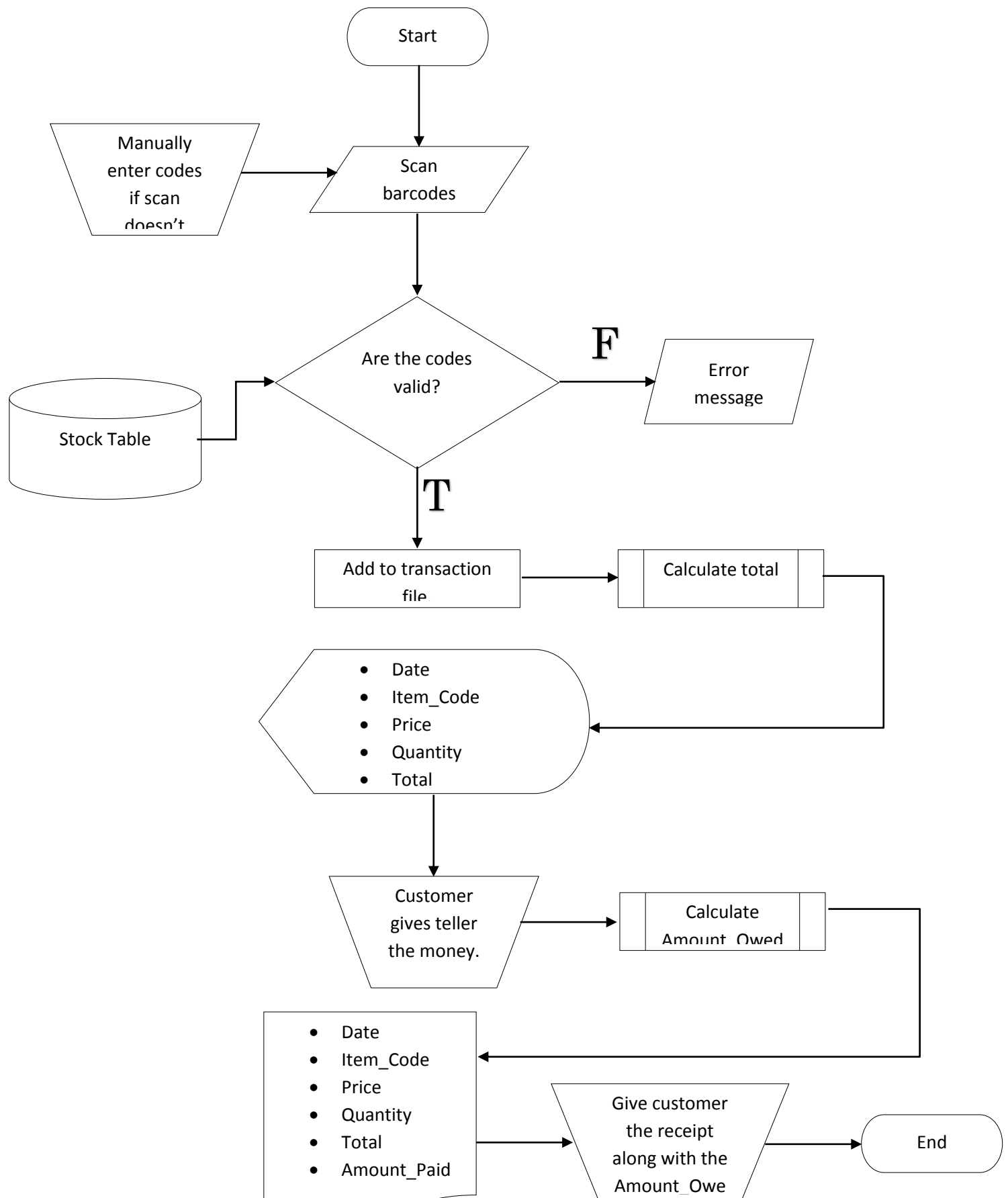
- To design the supplier forms
 - To design the transaction file for the POS
 - To design the stock control form
- Ram
 - To make processing faster by providing more space on the temporary memory
 - To process more instructions at a time
- Website
 - To display the businesses products
 - To display specials
 - To display advertisements for the business
 - To allow online shopping
 - To attract more customers
 - To defer some traffic from the landline as well as the email

System flowcharts

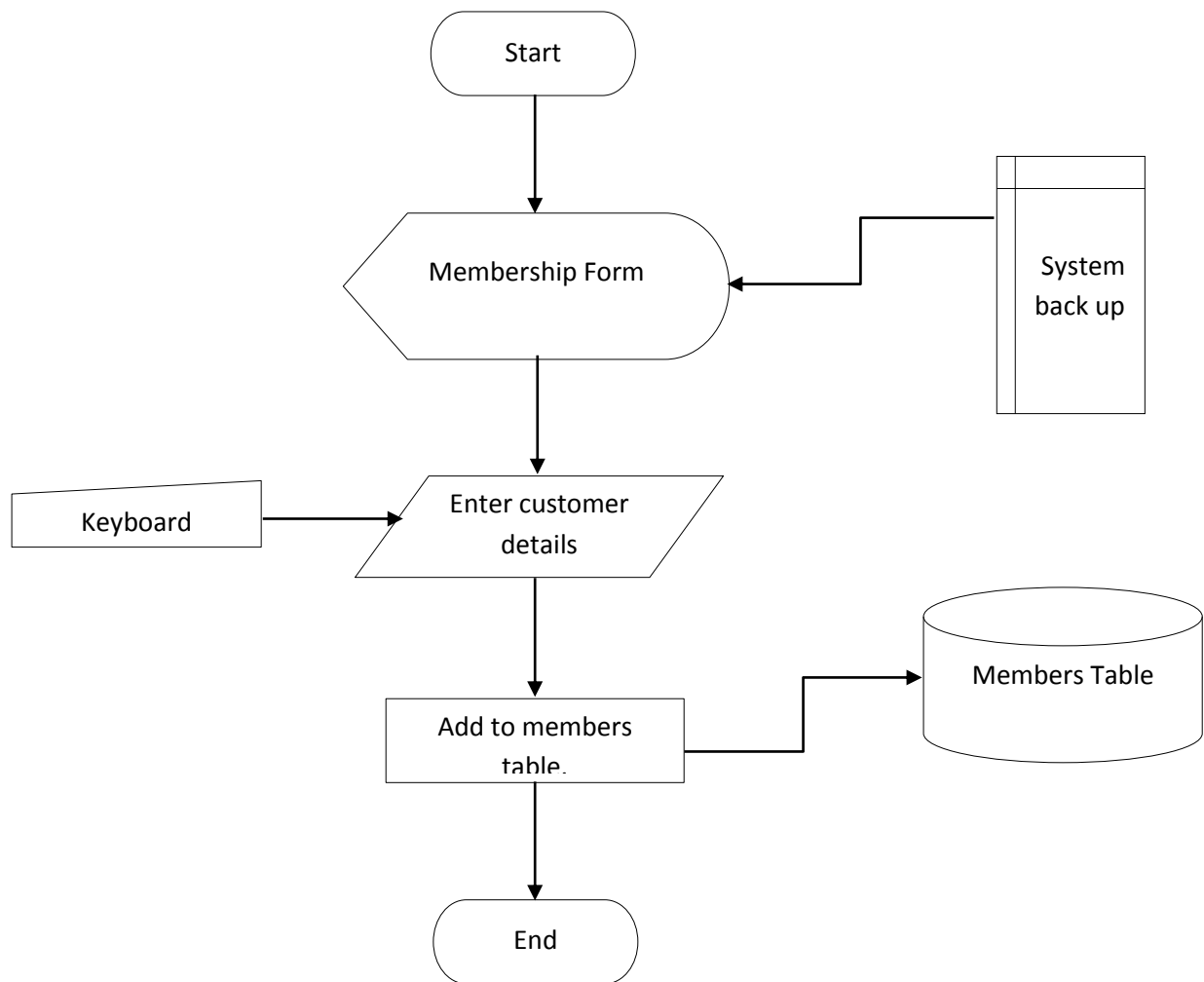
1. Searching for Member Details



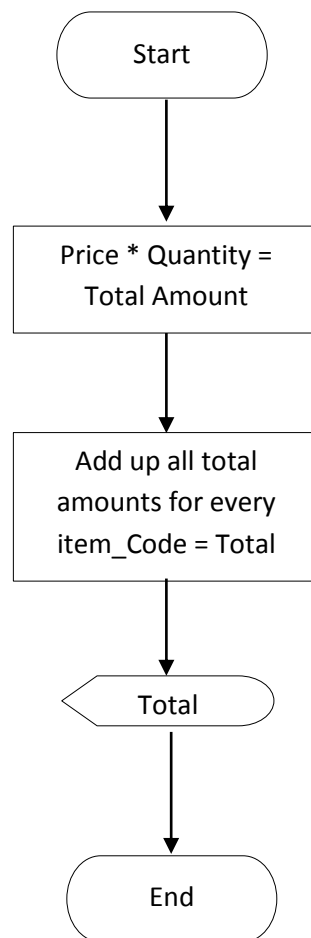
2. Customer makes a purchase



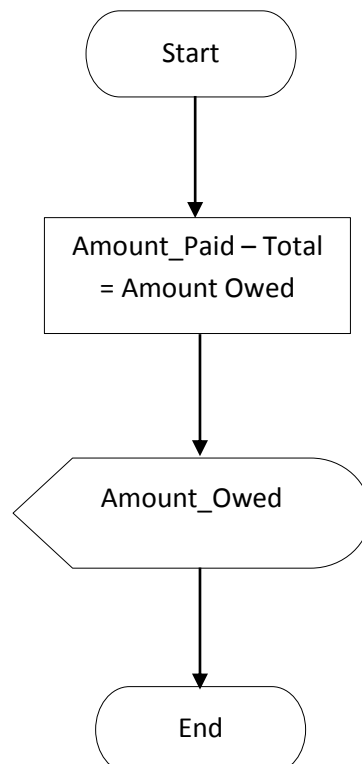
3. Add new member



4. Calculate total



5. Calculate Amount_Owed



Design Method of Solution

Although the program is far from perfect, it's a large improvement from the old system. Majority of the objectives were achieved.

Advantages

- The program is user friendly to employees as well as online customers. Anybody in any part of the world can access the website.
- The database keeps data on all customers, suppliers, stock and customers
- Masses of paper are saved.
- It is time saving.

Disadvantages

- The store could not afford CCTV cameras so the stock and customers remain unprotected.
- The security of the system is weak. Anyone can hack it.
- The entire system will stop and nothing will work if there is no electricity.

TESTING

Test strategy

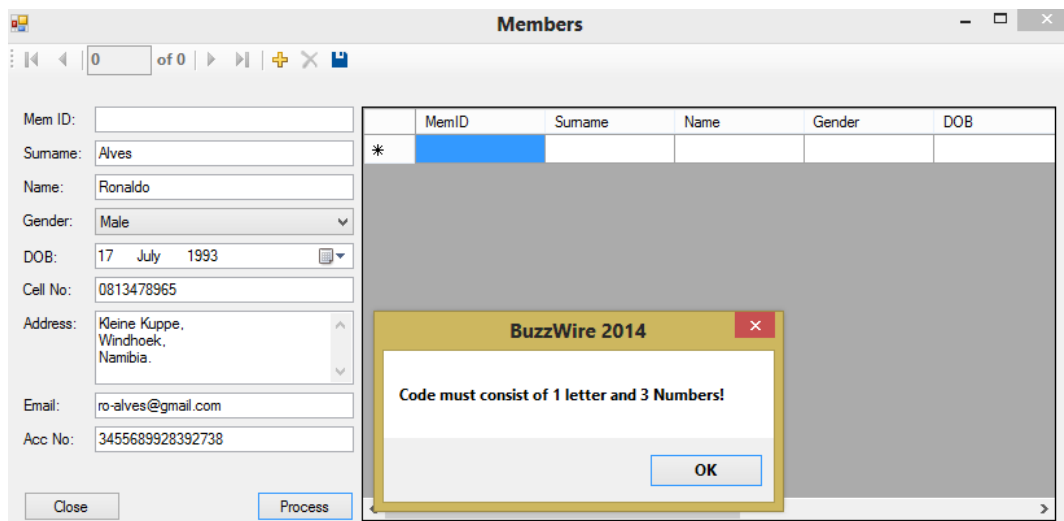
Normal, Abnormal and Extreme data will be used to test the accuracy of the validation rules set in the program.

Expected results

- Error messages when incorrect data is entered into certain fields.
- Data must show up in the table once add icon is clicked if the data is accepted.
- Whether normal data is accepted as expected.
- Whether extreme data is accepted as expected.

1. Format Check: Only input consisting of the letter “M” and 3 digits following it can be allowed.

- a. Abnormal data




This is what happens when incorrect data is entered into the MemID field.

b. Normal data

Mem ID: M001
 Surname: Alves
 Name: Ronaldo
 Gender: Male
 DOB: 17 July 1993
 Cell No: 0813478965
 Address: Kleine Kuppe, Windhoek, Namibia.
 Email: ro-alves@gmail.com
 Acc No: 3455689928392738

	MemID	Surname	Name	Gender	DOB
*					

When normal data is added there are no error messages and data can easily be added to the table when the add icon  is clicked. See below.

	MemID	Surname	Name	Gender	DOB
	M001	Alves	Ronaldo	Male	17/07/1993
*					

c. Extreme data

Mem ID: M999
 Surname: Alves
 Name: Ronaldo
 Gender: Male
 DOB: 17 July 1993
 Cell No: 0813478965
 Address: Kleine Kuppe, Windhoek, Namibia.
 Email: ro-alves@gmail.com
 Acc No: 3455689928392738

	MemID	Surname	Name	Gender	DOB
*					

As you may notice, the most extreme data (M999) is accepted.

2. Range Check: Only dates before 31/12/1996 may be entered in order to ensure that only person of age 18 and older may become members.

a. Abnormal data

The screenshot shows a web application window titled "Members". On the left is a form with the following fields: Mem ID (M001), Surname (Alves), Name (Ronaldo), Gender (Male), DOB (17 July 2015), Cell No (0813478965), Address (Kleine Kuppe, Windhoek, Namibia), Email (ro-alves@gmail.com), and Acc No (3455689928392738). At the bottom of the form are "Close" and "Process" buttons. On the right is a table with columns: MemID, Surname, Name, Gender, and DOB. The first row is highlighted in blue and contains an asterisk (*). A yellow error dialog box titled "BuzzWire 2014" is overlaid on the table, displaying the message "Must be 18 years or older!" and an "OK" button.

This is what happens when incorrect data is entered into the Date field.

b. Normal data

The screenshot shows the same "Members" web application window. The form fields are identical to the previous one, but the DOB is now "17 July 1993". The error dialog box is no longer present. The table on the right remains the same, with the first row highlighted in blue and containing an asterisk (*).

c. Extreme data

The screenshot shows a web application window titled "Members". On the left is a form with the following fields: Mem ID (M001), Surname (Alves), Name (Ronaldo), Gender (Male), DOB (17 July 1993), Cell No (0813478965), Address (Kleine Kuppe, Windhoek, Namibia), Email (ro-alves@gmail.com), and Acc No (3455689928392738). At the bottom of the form are "Close" and "Process" buttons. On the right is a table with columns: MemID, Surname, Name, Gender, and DOB. The first row contains an asterisk (*) in the MemID column, and the rest of the table area is greyed out.

MemID	Surname	Name	Gender	DOB
*				

The findings of my test results are that everything works as expected. All the fields which were mentioned in the computer related objectives, previously mentioned, have been programmed as desired. All three types of data (Abnormal, Normal and Extreme) were tested on two different types of validation rules for two different types of fields. They tested as expected.



Technical Documentation

Index

A. Welcome	Page 34
B. Validation Rules	Page 35
1. Range check	Page 37
2. Format check	Page 38

- Program used – Visual Studio Professional 2013

a. **Welcome:**



- All Buttons are called “btn”, e.g. btnClose.
- All forms have their original Design Names, e.g. Form1.
- All Text Boxes are called TextBox, e.g. MembersTextBox.

b. **Validation Rules used:**

1. Range Check
2. Format Check

1. **Range Check:**

Used in Members Form

```
Private Sub btnProcess_Click(sender As Object, e As EventArgs) Handles
btnProcess.Click
    If Not DOBDateTimePicker.Text <= 31 / 12 / 1996 Then
        MsgBox("Must be 18 years or older!")
    End If
```

And used in Transaction File

```
Private Sub btnProcess_Click(sender As Object, e As EventArgs) Handles
btnProcess.Click
    If Not DateDateTimePicker.Text = Date.Today Then
        MsgBox("Must enter current date!")
    End If
```

2. **Format Check:** used in the following fields:
- Item Code
 - Price
 - Quantity
 - Amount Received
 - AccNo
 - CellNo

e.g. If Not FaxTextBox.Text = "#####" Then
 MsgBox("Fax number must consist of 6 digits!")
 End If



User Documentation

Index

Log In Page 38

1. Start Computer
2. Open Buzzwire.exe
3. Log In Form

Use Form Page 39

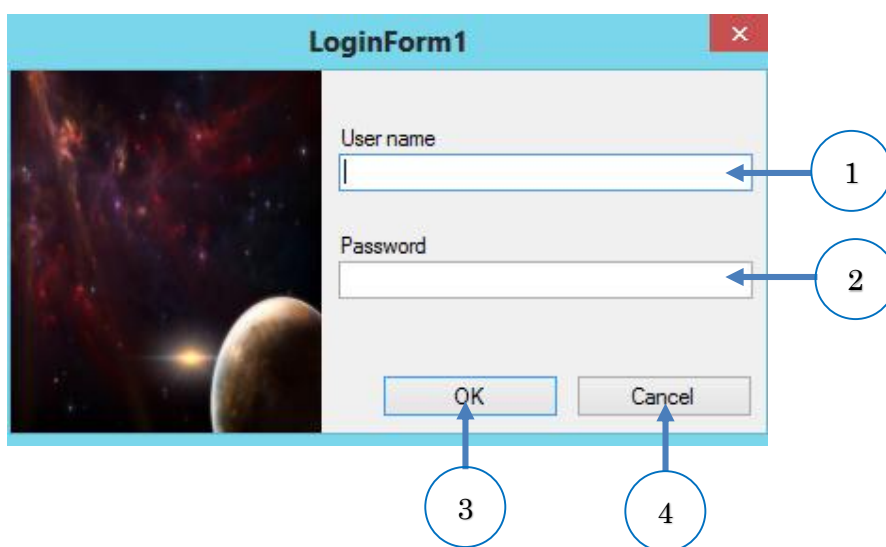
1. Menu
2. Buttons and Icons
3. All Forms
4. Transaction File

Log In:

1. Start Computer
2. Open BuzzWire.exe which is on the desktop
3. Click "Next"

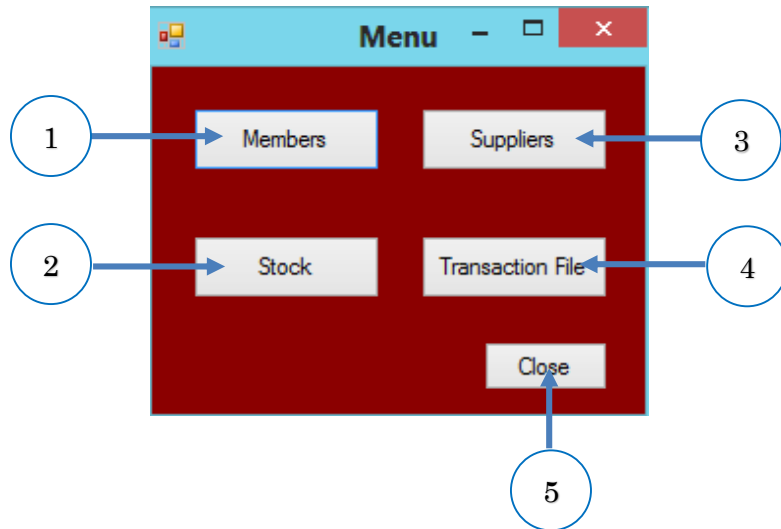


4. 1. Input Username.
2. Input Password.
3. Press OK to continue.
4. Press Cancel if you no longer wish to continue.

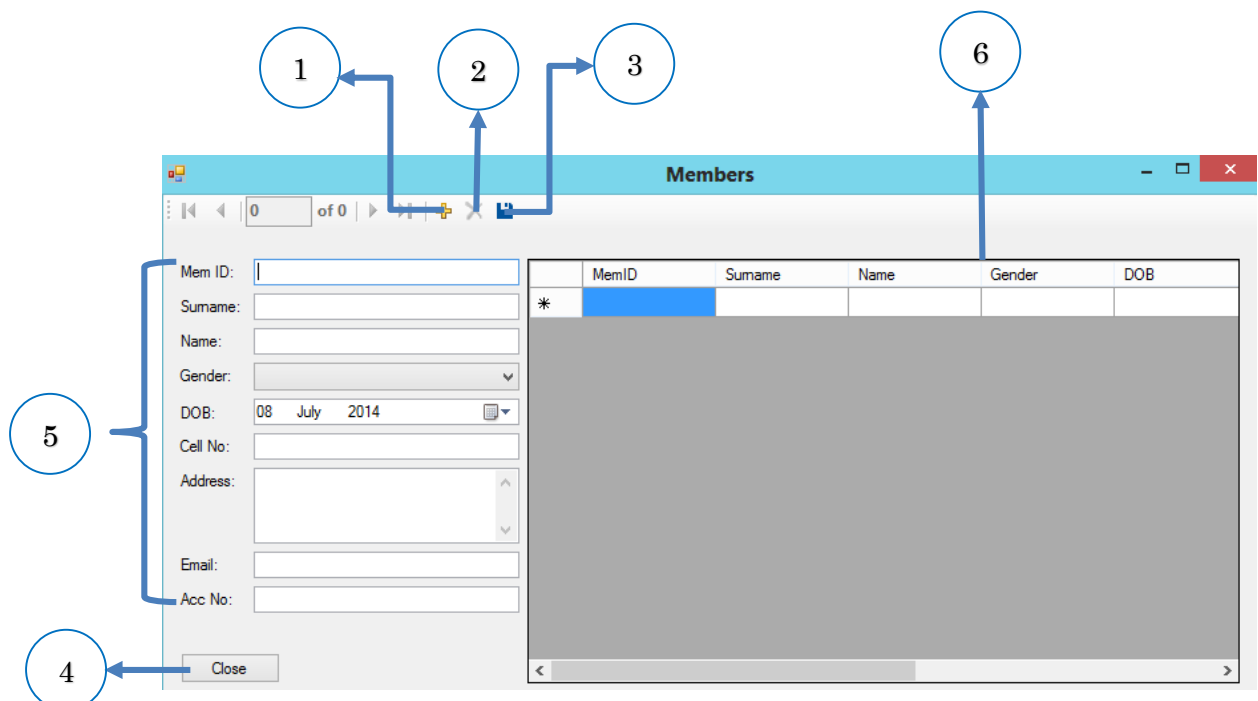


Use a form:

1. Click a button to view the input form and Information of that table.
 1. **Members' button** – Opens the Members table and input form.
 2. **Stock button** – Opens the Stock table and input form.
 3. **Suppliers' button** – Opens the Suppliers table and input form.
 4. **Transaction File button** – Opens the Transaction File table and input form.



2. Use buttons/icons to add, remove and save data.
 1. **Add icon** – to add data from the **Form (5)** into the **Table (6)**.
 2. **Delete icon** – to remove data from the **Table (6)**.
 3. **Save icon** – to save data to the **Table (6)**.
 4. **Close button** – to close page.



3. Do this with all forms connected to the buttons on the Menu.
4. Transaction File.

The screenshot shows a software window titled "Transaction File". It contains a form with the following elements:

- 1** points to the **Date:** field, which displays "12 June 2014".
- 2** points to the **Item Type:** list box, which contains a scrollable list of items: Printer, Laptop, PC, Smart Watch, Gaming, Router, and Mobile.
- 3** points to the **Process** button.
- 4** points to the **Total** button.
- 5** points to the **Change** button.
- 6** points to the **Receipt No:** field.
- 7** points to the **Close** button.

Other fields include **Item Code:** (text box), **Price:** (0.0000), **Quantity:** (0), **Total:** (0.0000), **Amount Received:** (0.0000), and **Change:** (0.0000). A table on the right side of the window displays transaction data:

	Date	Item Type	Item Code	Price	Quantity
▶	12/06/2014			0.0000	0
*					

1. Date is automatically set to the current date.
2. **List Box** – Click on any item in this box and the Price will come up automatically.
3. **Process button** – takes item code into consideration and automatically inputs the price of that item.
4. **Total button** – Calculates the total amount based on the **Price** and **Quantity** of the items.
5. **Change button** – Calculates the amount of change required for that specific transaction based on the **Total** and **Amount Received**.
6. Receipt No is loaded automatically.
7. **Close button** – Closes the page.

Evaluation

The program which I prepared had a few glitches to say the least. It fulfilled many of the objectives previously stated, both general and computer related ones.

These are the computer related objectives which I successfully achieved:

1. Computerise system.
2. Speed up the process of entering, editing, adding and removing data.
3. Have a neater system.
4. Install stock control system.
5. Create a relational database so that all the data is easily linked and found.
6. Database containing a Members, Suppliers, Stock and Transaction File tables.
7. Stock table containing Item_Type, Item_Code, Quantity, Cost Price, Selling Price and Total fields.
8. Members table containing MemID, Surname, Name, Gender, DOB, CellNo, Address, Email and AccNo fields.
9. Suppliers table containing SuppID, Name, Tel_No, Fax_No, Email, Physical Address, Postal Address and Acc_No fields.
10. Transaction File table containing Date, Item_Type, Item_Code, Price, Quantity, Total, Amount Received, Change and Receipt No fields.
11. Design a user-friendly Interface for employees to use.
12. Design program so that it is stimulating to the eye.
13. Design a program which links all the tables in the database to the program.
14. Design a program which collects data on members, suppliers and stock.
15. Prepare invoices and receipts for the customers.
16. Design queries to find data.
17. Design membership forms.
18. Search database.
19. Create sales records.

This is proven in the testing as well as the user documentation because, it shows how the forms and tables look and work. I failed to fulfil the rest of the computer related objectives.

Opportunities for Development

- Add date field to suppliers table in order to record date on which supplier became a suppliers.
- Link transaction file to stock table in order to keep track of how much stock is in store at all times.
- Create an algorithm to make the program more secure so that it cannot be altered by any unauthorised person.
- Add a search option to the program similar to that in the database software in order to find data much faster.



Program Code

Index

1. Form 1: Welcome	Page 48
2. Log In Form	Page 48
3. Form 2: Menu	Page 49
4. Form 3: Members	Page 49
5. Form 4: Suppliers	Page 50
6. Form 5: Stock	Page 51
7. Form 6: Transaction File	Page 52

Form 1: Welcome

```
Public Class Form1
    Private Sub btnNext_Click(sender As Object, e As EventArgs) Handles btnNext.Click
        LoginForm1.Show()
    End Sub
    Private Sub btnClose_Click(sender As Object, e As EventArgs) Handles
btnClose.Click
        Me.Close()
    End Sub
End Class
```

Log In Form

```
Public Class LoginForm1

    ' TODO: Insert code to perform custom authentication using the provided username
and password
    ' (See http://go.microsoft.com/fwlink/?LinkId=35339).
    ' The custom principal can then be attached to the current thread's principal as
follows:
    '     My.User.CurrentPrincipal = CustomPrincipal
    ' where CustomPrincipal is the IPrincipal implementation used to perform
authentication.
    ' Subsequently, My.User will return identity information encapsulated in the
CustomPrincipal object
    ' such as the username, display name, etc.

    Private Sub OK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles OK.Click
        If UsernameTextBox.Text = "Admin" And PasswordTextBox.Text = "BuzzWire" Then
            MsgBox("You have successfully logged in!")
            Form2.Show()
            Me.Close()
        Else : MsgBox("Incorrect Username and/or Password!")
        End If
    End Sub

    Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Cancel.Click
        Me.Close()
    End Sub

    Private Sub LoginForm1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    End Sub
End Class
```


Form 2: Menu

```
Public Class Form2

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
btnMembers.Click
        Form3.Show()
    End Sub

    Private Sub btnClose_Click(sender As Object, e As EventArgs) Handles
btnClose.Click
        Me.Close()
        LoginForm1.Show()
    End Sub

    Private Sub btnSuppliers_Click(sender As Object, e As EventArgs) Handles
btnSuppliers.Click
        Form4.Show()
    End Sub

    Private Sub btnStock_Click(sender As Object, e As EventArgs) Handles
btnStock.Click
        Form5.Show()
    End Sub

    Private Sub btnTransactionFile_Click(sender As Object, e As EventArgs) Handles
btnTransactionFile.Click
        Form6.Show()
    End Sub
End Class
```

Form 3: Members

```
Public Class Form3

    Private Sub MembersBindingNavigatorSaveItem_Click(sender As Object, e As
EventArgs) Handles MembersBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.MembersBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.MembersDataSet)

    End Sub

    Private Sub Form3_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'MembersDataSet.Members' table.
You can move, or remove it, as needed.
        Me.MembersTableAdapter.Fill(Me.MembersDataSet.Members)
    End Sub

    Private Sub btnClose_Click(sender As Object, e As EventArgs) Handles
btnClose.Click
        Me.Close()
    End Sub

    Private Sub btnProcess_Click(sender As Object, e As EventArgs) Handles
btnProcess.Click
        If Not MemIDTextBox.Text = "M001" <> "M999" Then
            MsgBox("MemID must consist of the letter M and 3 didgits!")
        End If
    End Sub
End Class
```

```

        If Not DOBDateTimePicker.Text <= 31 / 12 / 1996 Then
            MsgBox("Must be 18 years or older!")
        End If
        If Not CellNoTextBox.Text = "#####" Then
            MsgBox("CellNo consists of 10 digits!")
        End If
        If Not EmailTextBox.Text = "*@*.*" Then
            MsgBox("Please use the proper format for an email address. e.g.
juicegrape@gmail.com")
        End If
        If Not AccNoTextBox.Text = "#####" Then
            MsgBox("AccNo consists of 16 digits!")
        End If
    End Sub
End Class

```

Form 4: Suppliers

```
Public Class Form4
```

```

    Private Sub SuppliersBindingNavigatorSaveItem_Click(sender As Object, e As
EventArgs) Handles SuppliersBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.SuppliersBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.SuppliersDataSet)

```

```
End Sub
```

```

    Private Sub Form4_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'SuppliersDataSet.Suppliers'
table. You can move, or remove it, as needed.
        Me.SuppliersTableAdapter.Fill(Me.SuppliersDataSet.Suppliers)

```

```
End Sub
```

```

    Private Sub btnClose_Click(sender As Object, e As EventArgs) Handles
btnClose.Click
        Me.Close()
    End Sub

```

```

    Private Sub btnProcess_Click(sender As Object, e As EventArgs) Handles
btnProcess.Click
        If Not Supp_IDTextBox.Text = "S001" <> "S999" Then
            MsgBox("Supp_ID must consist of the letter S and 3 didgits!")
        End If
        If Not TelephoneTextBox.Text = "#####" Then
            MsgBox("Telephone number must consist of 6 digits!")
        End If
        If Not FaxTextBox.Text = "#####" Then
            MsgBox("Fax number must consist of 6 digits!")
        End If
        If Not E_mailTextBox.Text = "*@*.*" Then
            MsgBox("Please insert proper format of an email address!")
        End If
    End Sub
End Class

```

Form 5: Stock

Public Class Form5

```
Private Sub StockBindingNavigatorSaveItem_Click(sender As Object, e As EventArgs)
Handles StockBindingNavigatorSaveItem.Click
    Me.Validate()
    Me.StockBindingSource.EndEdit()
    Me.TableAdapterManager.UpdateAll(Me.StockDataSet)
End Sub
```

```
Private Sub Form5_Load(sender As Object, e As EventArgs) Handles MyBase.Load
'TODO: This line of code loads data into the 'StockDataSet.Stock' table. You
can move, or remove it, as needed.
    Me.StockTableAdapter.Fill(Me.StockDataSet.Stock)
End Sub
```

```
Private Sub btnSellingPrice_Click(sender As Object, e As EventArgs)
    Selling_PriceTextBox.Text = QuantityTextBox.Text * Cost_PriceTextBox.Text
End Sub
```

```
Private Sub btnClose_Click(sender As Object, e As EventArgs) Handles
btnClose.Click
    Me.Close()
End Sub
```

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If Item_CodeTextBox.Text = 1000 <> 1999 Then
        Selling_PriceTextBox.Text = "200"
        Cost_PriceTextBox.Text = "300"
    ElseIf Item_CodeTextBox.Text = 2000 <> 2999 Then
        Selling_PriceTextBox.Text = "800"
        Cost_PriceTextBox.Text = "600"
    ElseIf Item_CodeTextBox.Text = 300 <> 3999 Then
        Selling_PriceTextBox.Text = "5000"
        Cost_PriceTextBox.Text = "3000"
    ElseIf Item_CodeTextBox.Text = 4000 <> 4999 Then
        Selling_PriceTextBox.Text = "3000"
        Cost_PriceTextBox.Text = "1500"
    ElseIf Item_CodeTextBox.Text = 5000 <> 5999 Then
        Selling_PriceTextBox.Text = "500"
        Cost_PriceTextBox.Text = "300"
    ElseIf Item_CodeTextBox.Text = 6000 <> 6999 Then
        Selling_PriceTextBox.Text = "4000"
        Cost_PriceTextBox.Text = "1000"
    ElseIf Item_CodeTextBox.Text = 7000 <> 7999 Then
        Selling_PriceTextBox.Text = "400"
        Cost_PriceTextBox.Text = "200"
    Else : MsgBox("Incorrect code!")
    End If
End Sub
End Class
```

Form 6: Transaction File

Public Class Form6

```
Private Sub Transaction_FileBindingNavigatorSaveItem_Click(sender As Object, e As EventArgs) Handles Transaction_FileBindingNavigatorSaveItem.Click
    Me.Validate()
    Me.Transaction_FileBindingSource.EndEdit()
    Me.TableAdapterManager.UpdateAll(Me.TransactionFileDataSet)
```

End Sub

```
Private Sub Form6_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'TODO: This line of code loads data into the 'StockDataSet.Stock' table. You can move, or remove it, as needed.
```

```
    Me.StockTableAdapter.Fill(Me.StockDataSet.Stock)
    'TODO: This line of code loads data into the 'TransactionFileDataSet.Transaction_File' table. You can move, or remove it, as needed.
```

```
Me.Transaction_FileTableAdapter.Fill(Me.TransactionFileDataSet.Transaction_File)
```

End Sub

```
Private Sub btnTotal_Click(sender As Object, e As EventArgs) Handles btnTotal.Click
```

```
    TotalTextBox.Text = QuantityTextBox.Text * PriceTextBox.Text
```

End Sub

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles btnChange.Click
```

```
    ChangeTextBox.Text = Amount_ReceivedTextBox.Text - TotalTextBox.Text
```

End Sub

```
Private Sub btnClose_Click(sender As Object, e As EventArgs) Handles btnClose.Click
```

```
    Me.Close()
```

End Sub

```
Private Sub btnProcess_Click(sender As Object, e As EventArgs) Handles btnProcess.Click
```

```
    If Not DateDateTimePicker.Text = Date.Today Then
```

```
        MsgBox("Must enter current date!")
```

```
    End If
```

```
    If Item_CodeTextBox.Text = 1000 <> 1999 Then
```

```
        PriceTextBox.Text = "200.00"
```

```
    ElseIf Item_CodeTextBox.Text = 2000 <> 2999 Then
```

```
        PriceTextBox.Text = "800"
```

```
    ElseIf Item_CodeTextBox.Text = 300 <> 3999 Then
```

```
        PriceTextBox.Text = "5000"
```

```
    ElseIf Item_CodeTextBox.Text = 4000 <> 4999 Then
```

```
        PriceTextBox.Text = "3000"
```

```
    ElseIf Item_CodeTextBox.Text = 5000 <> 5999 Then
```

```
        PriceTextBox.Text = "500"
```

```
    ElseIf Item_CodeTextBox.Text = 6000 <> 6999 Then
```

```
        PriceTextBox.Text = "4000"
```

```
    ElseIf Item_CodeTextBox.Text = 7000 <> 7999 Then
```

```
        PriceTextBox.Text = "400"
```

```
    Else : MsgBox("Incorrect code!")
```

```
    End If
```

End Sub

End Class

