INFORMATION TECHNOLOGY

GRADE 12

PAT

**TASK 1 A : DEFINE THE TASK**

The topic of this year’s IT PAT for all the matrics is : “SAVE OUR PLANET”.

Our task is to create a Delphi Application that will spread awareness and minimise the harmful affect that we have on our planet to ultimately help save the Earth. This Application should use a database with at least 2 tables with minimum 5 fields, text files, OOP skills and should have a graphical user interface. The application should also have a minimum of 2 users.

Luckily I have a solution! My application will be an online store where users can buy a huge variety of different environmentally friendly, energy-effecient, reusable and recyclable products. By creating this program I will help contribute to saving our planet when I sell the products. People who buy the products will for example use less elictricity wich is made mostly by burning coal wich produces a lot of CO2 gasses and causes Global Warming. Or when people buy reuseable and recycleable products, there will be much less waste that ends up in our oceans. When they do throw away their products, it won’t have a negative impact because it is environmentally friendly.

To encourage buyers to buy environmentally friendly products, products will have an eco-friendly point between 1 and 5, when buyers have more than 20 eco friendly points they will receive 15% of off their next order. Users can choose wich products they want and then add it to their cart until they are ready do buy it. A manager will add and remove products ass well as their different attributes such as price, name, quantity and eco-friendly rating.

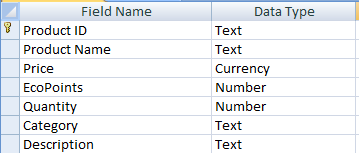
My application will use two different database tables, one for buyers and another for the products. It will also have two users : a buyer and a manager. Lastly, I will use all of my OOP skills and knowledge of text files to make the online store as effecient as possible.

**TASK 1 B : USER REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| **ROLE** | **ACTIVITY** | **LIMITATIONS** |
| Buyer | Browse the shop | Cannot add or remove any items on the shop. |
|  | Add and remove items to their cart | Can’t buy items without a valid payment method. |
|  | Get discount after they got 20 eco friendly points | Cannot decide how much discount they get, (They only get 15% discount when they have 20 eco friendly points). Eg. If they only have 10 points they can’t get half of the discount. |
|  | View eco friendly points that they get from buying a product, the total points they will get when buying everything in their cart, total eco friendly points on their account. | They can’t manually add eco friendly points to their account, they have to buy an item. |
|  | Return or exchange items | Cannot return items 7 days or later after purchase.  Cannot exchange items 30 days or later after purchase.  Canot return or exchange items without a valid reason.  Looses the eco friendly points on item. |
|  | Filter products according to eco friendly points, price, supplier etc. | Cannot add their own filters, they have to choose from a variety of choices. |
|  | Create edit or delete their account | Cannot edit password if they don’t give previous password OTP. |
| Manager | Add and remove items | Can only remove items when every unit of the item is sold. |
|  | Edit item’s price, name, eco friendly points, supplier. | Manager cannot edit the unique identification code of the products. |
|  | View sales, returns and exchanges. | Graphs and statistics is not included yet. |
|  | View changes made on the database by other managers. | Cannot undo changes, you must go and edit the error on the manager form. |

**TASK 2 : DATABASE DESIGN**

tblProducts: Design View

****

Product ID Field size: 10

Product Name Field size: 35

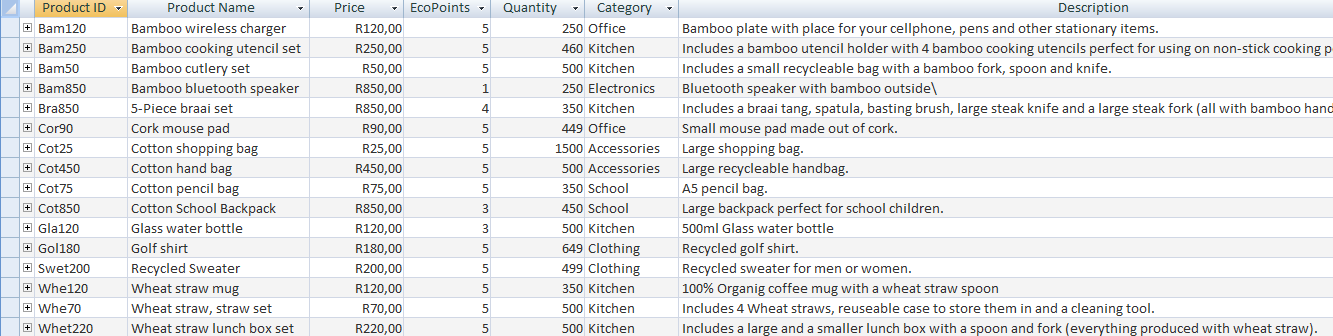
Price Field size: Currency

EcoPoints Field size: single

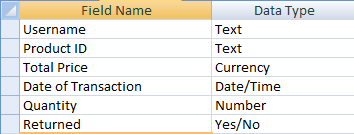
Quantity Field size: Single

Category Field size: 25

Description Field size: 255

tblProducts: Datasheet view

tblSales: Design View

****

Username Field size: 25

Product ID Field size: 10

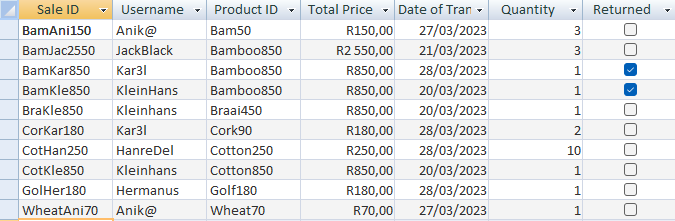
Total Price Field size: Currency

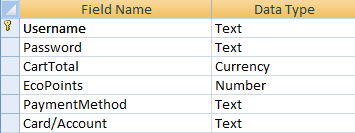
Date of Transaction Field size: Long Date

Quantity Field size: Integer

Returned : Yes/No

tblSales: Datasheet view

****

tblBuyers: Design View

Username Field size: 25

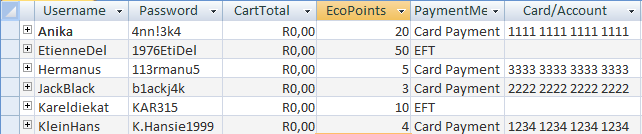
Password Field size: 25

CartTotal Field size: Currency

EcoPoints Field size: Single

PaymentMethodField size: 25

Card/Account Field size: 25

tblBuyers: Datasheet view

**TASK 3 A : CLASSES AND OBJECTS**

clsUser is a class that I am going to use to store all the information of users that my applications are going to use. For this spesific application I am just going to have one object: Buyer. The buyer has 4 attributes/pieces of data that the application has to store: Username, Password, Cart total and the ammount of eco points that the user has.

|  |  |  |  |
| --- | --- | --- | --- |
| **METHOD** | **METHOD TYPE** | **RETURN TYPE** | **PARAMETERS** |
| Create | Mutator | None | sUsername, sPassword, sAccount, iEcoPoints |
| GetUsername | Accessor method | String | None |
| GetPassword | Accessor method | String | None |
| GetCartTotal | Accessor method | Real/Currency | None |
| GetEcoPoints | Accessor method | Integer | None |
| ToStr | Auxiliry | String | None |
| CalculateDiscount | General Auxiliary | Real | None |
| CalculateNewTotal | General Auxiliary | Real | None |

SCOPE OF METHODS:

SetBuyer method, is going to be used to give my object the necessary attributes. All the “get-” functions is used to get the attribute from my object so that it can be displayed alongside the labels in “frmBuyer” on the account page. ToStr is going to be used to change the methods’ return types to string and format the output. The two “calculate-” fields are going to be used to calculate the ammount discount that a buyer gets because of having ecopoints and also what the new total is going to be after subtracting the discount from the total.

**TASK 3 B : TEXT FILES AND ARRAYS**

In my application text files shall be used as input for a “help button”. When users doesn’t understand how to use the program they can click the button and there will be a text file loaded into a rich-edit wich will then explain how the points system works, and how to buy or search products products.

Text files could also be used to log all the interactions that users or the mannager had with the program. When the manager changes anything on the database it will be logged on the text file , the text file shall display the time and date that the interaction was done and it will describe what was changed on the database.

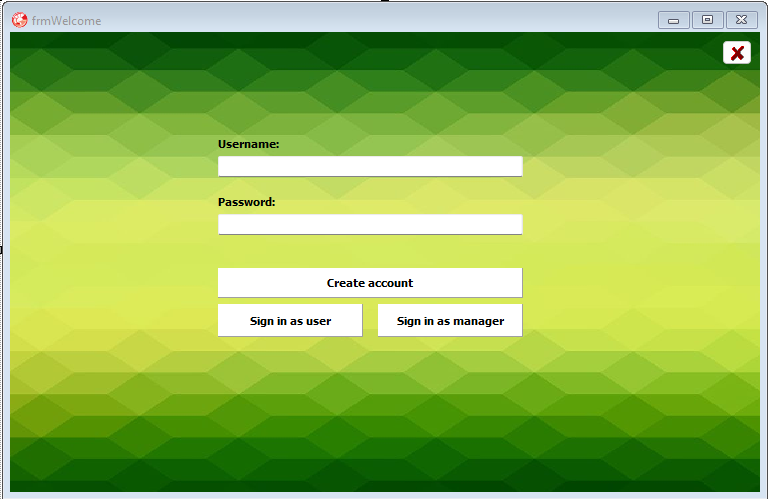
Thirdly text files could be used to show how many products was sold each day and how many products are currently in buyers’ carts.

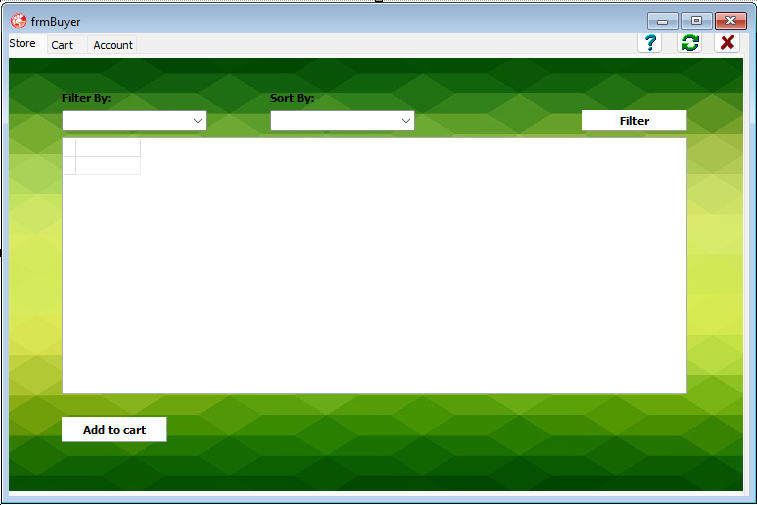
I could store the usernames and passwords of the users from tblUsers in 2 arrays when the program is activated, the mangers’ usernames and passwords will be stored in seperate constant arrays so that I don’t have to create a seperate table for administrators.

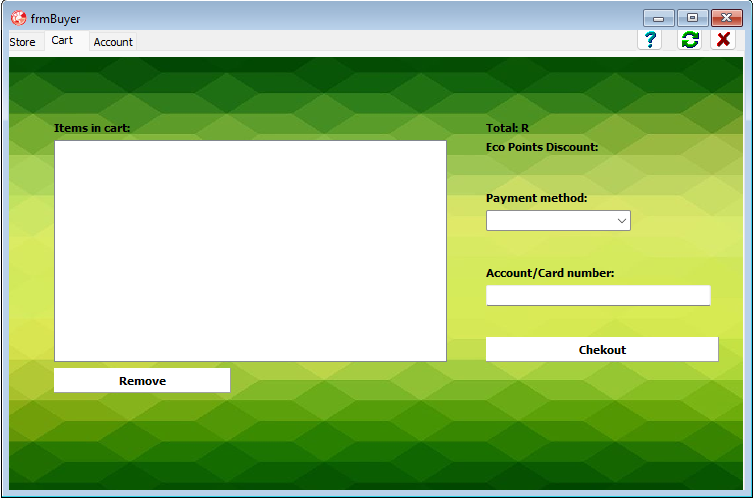
**TASK 4 A : NAVIGATION/DESCRIPTION OF FLOW DIAGRAM:**

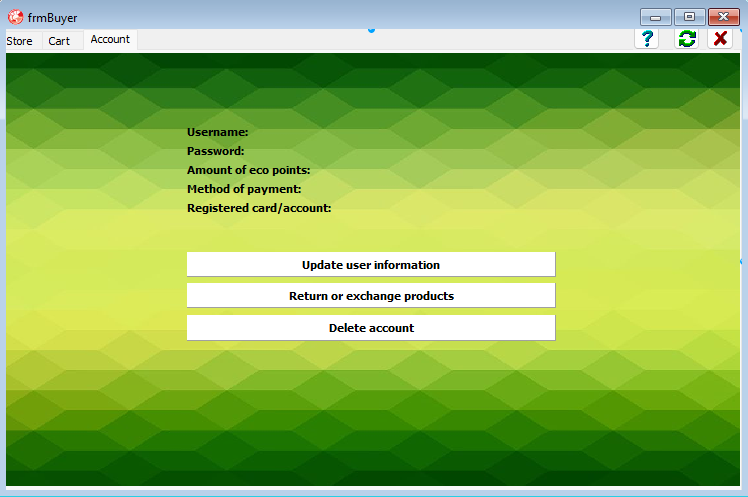
**TASK 4 B :GUI DESIGN**

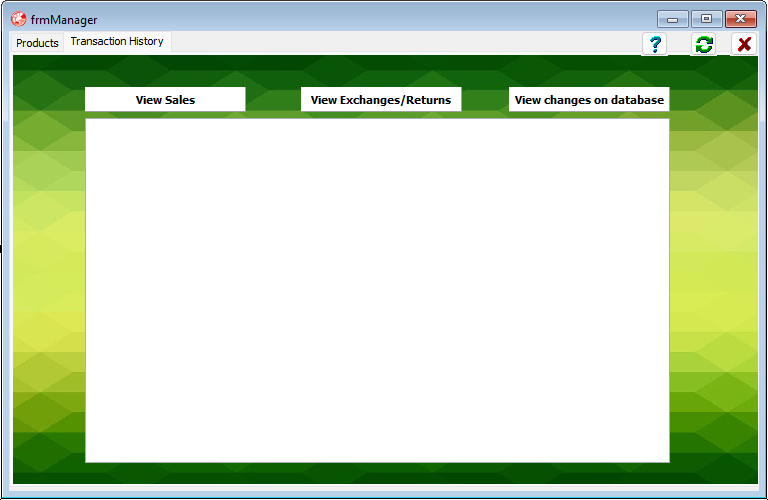
frmWelcome:

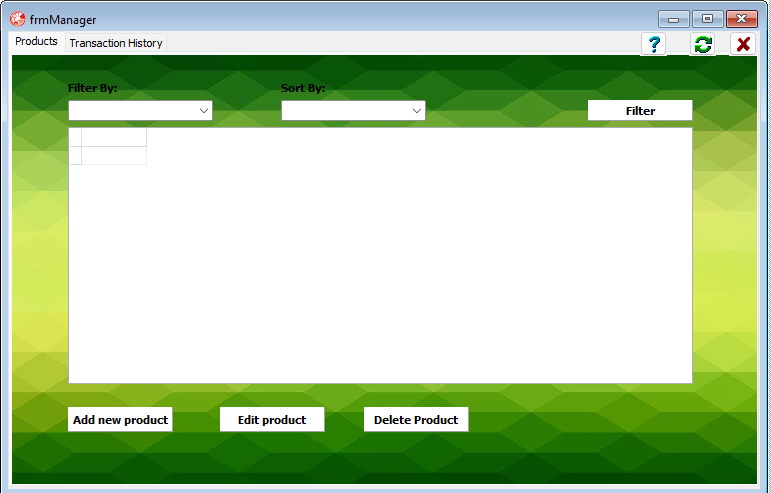


frmBuyer: Store

frmBuyer: Cart

frmBuyer: Account

frmManager: Products



frmManager: Transaction History

**TASK 5 : IPO – SOFTWARE DESIGN TOOL**

INPUT:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **INPUT** | **SOURCE OF INPUT** | **GUI COMPONENT** | **DATA TYPE** | **FORMAT** |
| **frmWelcom** |  |  |  |  |
| Username/Password | Keyboard | Edit box | String | No spesific format, every username and password will be unique. No spesific format, Password must be longer than 8 characters and must have a spesific character, number and a capital letter. |
| **frmBuyer** |  |  |  |  |
| Filter | Mouse | Combo box | Integer (Item index) | The name of the category wherin a product falls, starts with a Capital letter for example: Kitchen |
| Product Name | Database | Not applicable | String | Descriptive name, not in a spesific format. |
| Product ID | Database | Not applicable | String | Consists of the first 3 letters of the product name and its price (without R) put together for example: Bam120 |
| Price | Database | Not applicable | Currency | Price in Rand, with 2 decimal places for cents |
| Eco Points | Database | Not applicable | Integer | Just the number of the amount of eco points for example: 5  The number ranges from 1-5, no more, no less. |
| Quantity | Databse | Not applicable | Integer | Just the number of products that are available to sell for example: 550 |
| Category | Database | Not applicable | String | The name of the category wherin a product falls, starts with a Capital letter for example: Kitchen |
| Description | Database | Not applicable | String | No spesific format. |
| Payment method | Mouse | Combo box | String | No spesific format, user must choose between Credit/Debit card or to eft the payment |
| Card/Account number | Keyboard | Edit box | String | Just the number for examle: 1111 1111 1111 1111. Must contain 16 numbers. |
| Username/Password | Keyboard | Inputbox | String | No spesific format, Password must be longer than 8 characters and must have a spesific character, number and a capital letter. |
| Sale ID | Not applicable | Not applicable (Generated by the program after a user checked out) | string | Consists of the first three Characters of the product ID, the first three characters of the username that bought the product, and finally the price of the product for exmple:  BamAni150 |
| Total Price | Not applicable | Not applicable (calculated by the program after the user checked out) | Currency | Price in Rand, with 2 decimal places for cents |
| Date of transaction | Not applicable | Not applicable (Program retrieves current date from the computer when the transaction is made.) | DateTime | Long Date  For example: 13/03/2023 |
| Quantity | Keyboard | Inputbox (when user chooses products to put in cart) | Integer | Just the number of products that are available to sell for example: 550 |
| Returned | Mouse | Button | Yes/no (Boolean) | Not applicable |
| **frmManager** |  |  |  |  |
| Filter | Mouse | Combo box | String | The name of the category wherin a product falls, starts with a Capital letter for example: Kitchen |
| Product Name | Keyboard | Input box | String | Descriptive name, not in a spesific format. |
| Product ID | Keyboard | Input box | String | Consists of the first 3 letters of the product name and its price (without R) put together for example: Bam120 |
| Price | Keyboard | Inputbox | Currency | Price in Rand, with 2 decimal places for cents |
| Eco Points | Keyboard | Inputbox | Integer | Just the number of the amount of eco points for example: 5  The number ranges from 1-5, no more, no less. |
| Quantity | Keyboard | Inputbox | Integer | Just the number of products that are available to sell for example: 550 |
| Category | Keyboard | Inputbox | String | The name of the category wherin a product falls, starts with a Capital letter for example: Kitchen |
| Description | Keyboard | Inputbox | String | No spesific format. The text may not be longer than 255 characters |

INPUT VALIDATION:

|  |  |  |  |
| --- | --- | --- | --- |
| **INPUT** | **DATA TYPE** | **VALIDATION** | **ERROR MESSAGE** |
| Username/Password | String | Validate for NULL/empty field. | “Please fill in both your username and password.” |
| Payment method | string | Check if a payment method has been selected in the combobox. | “Please select a payment method before enterring your bank details.” |
| Price | Currency/Float | Check for a NULL field | “Please fill in the product’s price.” |
| Price | Currency/Float | Test the data type | “Please only fill in numbers and one coma if the price also contains cents.” |
| Eco Points | Integer | Test range | “Please only fill in a number between 1 and 5” |
| Returned | Boolean | Test wether the returned property is true when an item is being returned. | Not applicable.  My program automatically updates the “returned” property of an item when an user returns the item, this way I know that the data will never be incorrect. |
| Date Of Transaction | Date/Time | Test format of the date. | Not applicable  By retrieving the date of the computer when the transaction is being done, I can be sure that the format of the date is correct. |

DATA PROCESING:

|  |  |
| --- | --- |
| **PROCESSING THAT NEEDS TO BE DONE** | **HOW PROCESSING WILL BE DONE** |
| Check wether the username and password typed by the user is correct. | Firstly check if the username appears in the array of saved usernames and get it’s position(iPos). If the password typed by the user is not equal to the password in the position (iPos) in the array of saved passwords display a message: “The password is incorrect”, if the username is not found in the array of usernames display a message: “The username you typed does not exist, if you want to create a new account click on the button ‘create account’ . ” |
| Filter all the products in the table. | Firstly extract the caption of the item selected in the filter combo box and save it as “sFilter”.  Use the next part of a SQL statement:  SELECT\* FROM tblProducts WHERE Category= “sFilter” |
| Sort the products in the table. | Case cmb.Itemindex of  begin  0: qry.sql.add(SELECT\* FROM tblProducts WHERE Category= “sFilter” ORDER BY Price ASC);  1: qry.sql.add(SELECT\* FROM tblProducts WHERE Category= “sFilter” ORDER BY Price DESC);  2: qry.sql.add(SELECT\* FROM tblProducts WHERE Category= “sFilter” ORDER BY EcoPoints ASC);  3: qry.sql.add(SELECT\* FROM tblProducts WHERE Category= “sFilter” ORDER BY EcoPoints DESC);  End; |
| Calculate amount of discount on total. | If the user has 20 or more eco points on their account (Before buying the items in their cart) they get 15% off of their total that they have to pay.  Calculate the amount discount they get:  rDiscount:=15/100\*rTotal  Their total must now be updated:  rTotal:=rTotal-rDiscount |
| Exchange products. | Get the SaleID of the product the user wants to exchange from the user.  Change the exchanged field in the database to true.  Display message: “The product can now be exchanged, we will transfer the money in your account when you have dropped the product off at one of our depots” |
| Add new product as manager. | Ask the manager to type the Product name, price, eco points, quantity available, Category and description in 6 different input boxes.  Generate the product ID by putting the first 3 letters of it’s name together with the price.  Use Delphi code to add this new product on tblProducts. |
| Edit product as manager. | Ask the manager to type in the field’s name that he/she wants to change. Then ask for the new value. Use delphi code to change the old value in the field to the new value that the manager typed in. |
| Log changes made by Managers on the database, in a text file. | Everytime that any manager adds, deletes or edits products on tblProducts the date on the computer must be stored in a text file, together with the name of the manager, the action that he did (Add­-, Delete-, Edit- product), product ID of the product and if a product was edited the field that was changed, the old value and new value must also be saved. For example:  23/13/2023 TheBoss Edit Bam150 Price 120, 150. |

DATA OUTPUT:

|  |  |  |
| --- | --- | --- |
| **DATA TO OUTPUT** | **FORMAT OF OUTPUT** | **GUI COMPONENT** |
| All the products and their info. | Each field’s data in the format as spesified in the database. | DBGrid |
| All the items that an user has in their cart. | Item Name #9 Price #9 Eco Points | List Box |
| Total amount of all the products in user’s cart | ‘R’+rTotal | Label |
| Eco Points discount | ‘R’ +rDiscount | Label |
| New total, after the eco points discount have been subtracted from the original total. | ‘R’+rNewTotal | Label |
| Indication that the transaction was successful/unseccessful | The word “Transaction:” followed by “Successful” or “Unsuccessful” | Message box |
| Username | “Username: ”+sUsername | Label |
| Password | “Password: ”+sPassword | Label |
| Amount of Eco points | “Amount of Eco points: ”+iEcoPoints | Label |
| Method of payment | “Method of payment: ”+ sPaymentMethod | Label |
| Registered card/account | “Registered card/account: ”+sBankdetails | Label |
| Changes made on database | Date of the interaction with database +#9 + Manager’s username who interacted with database + #9 + Action that he/she did + #9+ The ProductID of the product that has been deleted/added/edited + #9 + Field name that has been changed + #9 + old value of the field + #9n + new value of the field. | Rich edit |
| Indication that change on the database has been successfull/ unsuccessful | “Product has been successfully ”+ Updated/ Added/ Deleted  Or  “Product has not been ”+ Updated/ Added/ Deleted | Message box |
| Indication that login was successfull or not | “Successfully logged in as ”+ Username  OR  “Login failed, please try again” | Message Box |