PRACTICAL ASSESSMENT TASK

PHASE 1

[Brian Daniel

Grade 12 IT Set 1

**Description**

Many blood donor clinics use outdated systems of storing information such as pen and paper or software designed many years ago. This results in blood samples being spoiled, recipients receiving the wrong type of blood and many other issues.

I intend to write a program for a clinic that would catalogue blood donor samples, keeping track of important information such as the type of blood, how long ago it was donated, the volume of blood donated. It would have interface for both adding new samples to a database and for using queries to select specific samples for reception, utilizing an algorithm to ensure that the maximum number of samples are used before expiring. You would also be able to view the total number of certain types of samples and would be able to print receipts of donations / reception of blood samples.

The program would first allow users to select their role: Donation manager, Donation marketer or Hospital-side blood manager. They would then be able to decide which functions they would like to access, choosing from adding/removing blood samples, printing donation/reception receipts, viewing available blood samples and notifying past donors of donation eligibility

Generally, this would be invaluable to N.G.O’s running blood donor clinics and would result in many more lives and time saved. It would increase logistical efficiency allowing for resources to be saved where they once would have been wasted on dated systems.

USER REQUIREMENTS

Users will need an interface with which to:

* add blood donor samples
* retrieve and check out blood donor samples based on certain criteria
* monitor blood donor samples (how many of which type, etc.)
* monitor blood donors so they can be notified when they will be eligible to donate again

|  |  |  |
| --- | --- | --- |
| **User Role/Who** | **Description** | **Limitations** |
| Donation Manager | * Add New blood samples * Remove expired/spoiled blood samples * Monitor the demand levels and availability of different types of blood | * May not request blood samples * May not contact past donors |
| Donation Marketer | * Keep track of past donors so they may be notified of donation eligibility * Monitor the demand levels and availability of different types of blood * Print receipts for blood donations | * May not add/remove blood samples |
| Hospital-side blood manager | * Request blood samples * Notify clinics of changes in demand * Print receipts for blood reception | * May not add blood samples * May not contact past donors |

SOFTWARE DIAGRAM

HOSPITAL-SIDE BLOOD MANAGER   
Controls the requests for blood from hospitals to treat patients. Also can view supply and demand of blood samples

MAIN MENU

Serves as central hub for all functions, you are required to select your user type.

DONATIONS MANAGER

Is responsible for maintaining and updating the blood sample database. Is also able to view supply and demand levels

DONATIONS MARKETER  
Is responsible for maintaining the level of donations in the bank, and thus has access to past donors, and demand levels

ADD NEW BLOOD SAMPLES

Lets the manager update the current supply of blood samples

BLOOD SAMPLE DATABASE

Contains information about all available blood samples, cataloguing information such as Blood type, volume, age, etc.

DONOR DATABASE

Contains information about past donors, so they may be contacted again and otherwise stores their information

VIEW BLOOD SAMPLES

Allows users to see statistics and samples available using SQL queries that are prewritten for easy use

REMOVING BLOOD SAMPLES

Allows users to remove blood samples, either because they are being used or they have expired/spoiled

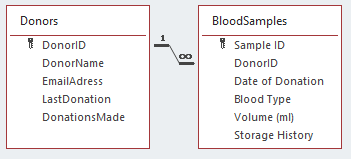
NOTIFYING PAST DONATORS OF ELIGILITY

This is done so the bank can maintain a high rate of donation, can be specified in relation to demand

DATABASE DESIGN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table name: BloodSamples** | | | | | | |
| Field Name | SampleID | DonorID | DonationDate | BloodType | Volume (ml) | StorageHistory |
| Data Type | String | String | Date | String | Integer | String |
| Example Data | 0312145491089004 | 0312145491089 | 31/12/2020 | O+ | 500 | [-15/-20]{10};[2/8]{5} |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table name: Donors** | | | | | |
| Field Name | DonorID (Primary) | Name | EmailAddress | LastDonation | DonationsMade |
| Data Type | String | String | String | Date | Integer |
| Example Data | 0312145491089 | Brian Daniel | Brian.daniel@bishops.org.za | 31/12/2020 | 4 |



DATA DICTIONARY

Classes

|  |  |  |  |
| --- | --- | --- | --- |
| TBloodSample Class | | | |
| fSampleID | | String | Unique sample Identifier |
| fDonorID | | String | Unique donor Identifier |
| fVolume | | Integer | Volume in ml of blood |
| fStorageHistory | | String | Information containing for how long and at wdAWDhat temperature samples were stored |
| fDonationDate | | TDateTime | The date of donation. Used to calculate age |
| fBloodType | | String | Identifies the type of blood |
| Constructor, Functions and Procedures | | | |
| Constructor | Create(SampleID, DonorID, StorageHistory, BloodType : String;  Volume : Integer;  DonationDate : TDateTime); | | |
| Function | GetSampIeID() : String; | | |
| Function | GetDonorID() : String; | | |
| Function | GetVolume() : integer; | | |
| Function | GetStorageHistory() : array of string; | | |
| Funciton | GetBloodType() : String | | |
| Procedure | SetAge(Age : TDateTime); | | |
| Procedure | SetEligibility(Eligibility : Boolean); | | |
| The purpose of this class is to help manage the data involved with each blood sample. It will serve as a useful format to store the data necessary to process each blood sample. Each sample would be created as an instance of the class, with the data being retrieved from the database. The samples could then be processed. | | | |

Text Files

Text files will be used as a form of output, specifically to generate receipts for blood donation / reception. They could then be printed by the user, allowing for easy one click generation of a practically useful receipt. It would contain information such as volume, Sample ID, donor ID, blood type, Sample age and more. It would be named according to the Sample ID. Additionally it would contain a short advertisement convincing people to donate blood.

Other Data Structures

**Arrays and Tuples**

Arrays would be used throughout implementation, although a specific example would be in processing storage period data. Each element would represent one unique storage period, containing a tuple with the first element being the length of storage, and the second being the temperature at which it was stored.

An example would be converting ' [-17.5]{10};[5]{5};[12.5]{1}' into: [ (-17.5, 10), (5, 5), (12.5, 1) ]

(This data represents first storage period at -17.5 degrees for 10 days, second at 5 degrees for 5 days, third at 12.5 degrees for 1 day.)

GUI DESIGN

**PAT INSTRUCTION**

Design a GUI that considers good human-computer-interface (HCI) principles, that prevents errors occurring from invalid input and that minimises the amount of information a user must input.

Use HCI design principles and design a GUI that considers the following:

• The user – type and context

• User requirements/needs, usability

• Dialogue – must be relevant, simple and clear

• Icon usage and presentation – well selected and relevant, well placed with a clear purpose

• Colour – use and combination of colour

• Feedback – neat, clear and well presented

• Helpful error messages

• Exits – clearly marked, placed correctly

• Shortcuts

• Flow of information on the screen – top to bottom and left to right

• Sensible usage of space on the screen

Provide sample(s) of planned data capture and data entry designs (prototype screen dumps may be used but must be annotated) and of planned valid output designs.

Show the GUI design following HCI principles of interface(s), excluding introductory screens.

**NG’s TRANSLATION/NOTES:**

* I have created a table on the next page. You must complete the table for each form/tab/GUI that you have in your software.
* Copy and paste the table below one another for however many times you need it.
* Remember to be consistent across your various forms.
* Don’t forget to clearly indicate how you will be providing help, navigation (closing the form, going back to main menu etc.) as you get marks for this.
* Once again, check the rubric to make sure that you cover everything!

**GUI TEMPLATE**

|  |  |
| --- | --- |
| **Design:** | |
| **Purpose/Use:** |  |
| **Components:** |  |
| **Data Validation:** |  |
| **Additional Info:**  **(Optional)** |  |

IPO

**NG’s TRANSLATION/NOTES:**

I decided not to copy the instructions for the PAT document as it is very lengthy. Make sure you read it first before proceeding. Also look at the rubric for this section. The IPO counts a whopping 20 marks out of the possible 48 marks you can get for Phase 1. Make sure you do this section properly! **CHECK YOUR RUBRIC TOO!**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task:** | **Input** | | | | | | **Processing** | | **Output** | | |
| **General** | | | **Validation** | | |
| **Source (GUI Component)** | **Data Type** | **Format** | **What** | **Method** | **Feedback** | **What** | **How** | **What** | **Format** | **Object** |
| User Registration | Keyboard (edtName) | String | Sentence Case | Numbers/  Special Characters | Loop through and check each char. | “Error! Your name cannot contain….” | Store in variable | sName := edtName.Text | Confirmation Message: “Your details have been stored successfully” | Text | Dialog Box |
| Keyboard (edtSurname) | String | Sentence Case | Numbers/  Special Characters | Loop through and check each char. | “Error! Your surname cannot contain….” | Store in variable | sSurname := edtSurname.Text |
| Mouse (dtpDOB) | TDate | Regional Settings  (dd/mm/yyyy) | None  (built-in) | N/A | N/A | Store in variable | sDOB := DateToStr(dtpDOB.Date) |
| Keyboard (edtCell) | String | Digits without any spaces in between e.g.  0789419712 | Has 10 Chars | if(length(edtCell) <10) then…. | “Your cell phone number must have 10 chars” | Store in variable | sCell := edtCell.Text |
| Calculate Total Price | Mouse (spnQty) | integer | Numeric | Range (no more than 5 items) | MaxValue Property | “Error! You cannot have more than 5 items!” | Calculate Price | quantity \* price (constant) = total price | “Your total comes to <price>” | Text  Currency | Price Label |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task:** | **Input** | | | | | | **Processing** | | **Output** | | |
| **General** | | | **Validation** | | |
| **Source (GUI Component)** | **Data Type** | **Format** | **What** | **Method** | **Feedback** | **What** | **How** | **What** | **Format** | **Object** |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

Annexure A – Declaration of Authenticity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Learner's name** |  | **ID Number** | |  |
| **Grade** | 12 | **Year** | | 2020 |
| **Subject** | Information Technology | | | |
| Practical Assessment Task (PAT) | | **Teacher** | | Mr GHI Nieuwoudt |
| I hereby declare that the contents of this assessment task are my own original work (except for items listed below or where there is clear acknowledgement and appropriate reference to the work of others) and have not been plagiarised, copied from someone else or previously submitted for assessment by anyone.  **List of assistance received:** | | | | |
| Nature of assistance | | | Person who provided assistance | |
|  | | |  | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_ / \_\_\_ / 2020  SIGNATURE OF LEARNER DATE | | | | |