Grade 10 PAT 2023 - INFO

Topic

Entrepreneurial Management System/Application

In the current economic environment, it is important for our citizens to think of ways to support themselves. Entrepreneurial initiatives will be necessary to ensure a steady economic growth. Many small businesses are using manual systems to manage their businesses.

This project allows you to create a business management system using Delphi. Use good programming principals throughout. Make use of the following pre-scripts:

The use PageControl with **at least** THREE tabsheets (more tabsheets are allowed)

- Tabsheets must be used in a way that it will add to the usability of the system. For example:
  - Form / tabsheet 1 – welcome and login as user/admin
  - Form / tabsheet 2 – buying/taking stock
  - Form / tabsheet 3 – print a report/receipt

This program is targeted to assist a small business user.

- The management system could be based on an existing system or it could be a new idea.
- The system can be for any small business.
- The complexity of the topics / themes are not awarded more marks in this project.

Examples of different types of business management systems (but not limited to):

- Tuckshop
  - Calculate input costs (price of purchasing stock at cost)
  - Stock take
  - Compiling a receipt, calculate VAT and change
  - Compile a list with unique product code
  - Calculate the rent for the building per square meter
- School clothing store
  - Stock take
  - Compiling a receipt, calculate VAT and change
  - Compile a list with unique product code
  - Print price stickers
  - Order new stock, possible discount option
- Spaza shop
  - Calculate selling price of products
  - Calculate change
  - Calculate the ratio of a recipe if it is double or 3 times
- Delivery service
  - Calculate delivery fee
  - Calculate distance
  - Calculate the time it will take

- Online Shop for home-made items
    - Calculate material costs
    - Calculate how much time it takes to create an item
    - Calculate packaging equipment and marketing (Website/Facebook)
    - Calculate profit
- Barber / Beauty Shop
    - Calculate costs of products used per person, according to what they want
    - Calculate running costs (Water, rent, employee-fees)
    - Calculate profit
- Repair service (Car, computer, etc.)
    - Calculate costs of products used per person, according to what they required
    - Calculate running costs (Water, rent, employee-fees)
    - Calculate profit
- Pet/House sitting Service
    - Calculate hourly rate for which type of pet / house
    - What will you need to make sure the job is well done.
    - Marketing costs must be calculated
    - Which pets needs what attention, type of food, etc.

Your final program must comprise of ***ONE*** single, logically related piece of software. Projects that consist of two or more unrelated programs will only obtain marks for ***ONE*** of the parts since only ***ONE*** of the programs will be regarded as the actual project.

Phases and Tasks

Phase	Task No.	Task Description	Due-date
1	Task 0	Problem definition and research	18 Sep 2023
1	Task 1	Task definition and user story	10 Oct 2023
1	Task 2	Acceptance Test	10 Oct 2023
1	Task 3	Navigation / flow between screens	10 Oct 2023
1	Task 4	Design screens (GUI Design)	28 Sep 2023
1	Task 5	IPO Tables and data validation	28 Sep 2023
2	Task 6	Create Delphi TWO screens	28 Sep 2023
2	Task 7	HCI Principles for TWO screens	23 Oct 2023
2	Task 8	Develop the code	23 Oct 2023
2	Task 9	Testing and data validation	23 Oct 2023
2	Task 10	Documentation	23 Oct 2023

Task 0 is a planning and preparation task that does not carry any marks (rough idea on paper)

Topic	What type of entrepreneurial management system was selected by you
Purpose of program	Describe the purpose of your program – why does the small business owner need your program.
Possible solution	What will the program do to meet the identified needs? Describe how your program will work. Include a description of each of your tab sheets and how the user will interact with your program.
Scope	Explain what limitations your program may have. (What your program will not do)

PAT Minimum requirements

- □ Add a PageControl with at least three tabs.
- □ Tab sheet 1: Create a welcome page. You need two types of users and depending on who they are it will take them to the next tab sheet.
- □ Tab sheet 2: One user's actions will be done here.
- □ Tab sheet 3: Another user's actions will be performed here.
- □ Make use of colour throughout the program.
- □ Use proper variable names like sName, iNumber etc.
- □ At least 4 of the following data types overall in your program:
    - Integer
    - String
    - Char
    - TDate
    - Real
    - Boolean
- □ At least 1 built-in functions (Math functions or String functions – Lists are on Classroom). Type casting functions (strtofloat, inttostr etc,) will not earn any marks for this section but should be used appropriately throughout the program.
- □ At least one nested If-Statement
- □ At least one us of the IN operator
- □ At least one user-defined function (Will be explained in **term 3**)
- □ Global variables used **appropriately**
- □ Comments in each event (Button / FormActivate / user-defined function etc.) explaining what that event will be doing
- □ Use indentation and open spaces appropriately (programming techniques)
- □ Output should be formatted appropriately and clear with proper labels, headings and tab stops where appropriate.
- □ Each tab sheet must have a Help and Reset button. The help must explain to the user how to use that tab.
- □ Depending on the tab sheet they should have the following buttons:
    - Next (Not on last tab)
    - Back (Not on home tab)
    - Home (Not on home tab)

Before you start

- Rename the folder PAT2023_SurnameName and the replace the words Surname Name with your own (Start each name with a capital – note the surname must be given first)
- Rename TWO subfolders as follows:
    - PAT2023_Fase1_SurnameName where SurnameName is replaced with your own. Also rename the documents inside the folder.
    - PAT2023_Fase2_SurnameName where SurnameName is replaced with your own. Also rename the document inside the folder.
- The PAT Phase 1 word document should be saved in you Phase 1 folder on the network at all times.
- Rename the mark sheet in MS Excel to read PAT2023_SurnameName and replace SurnameName with you own.

Phase 1 Due date: 10 October 2023

MS Word document - Requirements and examples

Phase 1 will be completed in a Microsoft Word document **but not in the order of the headings of the template provided**. Built-in headings were used to create an automatic table of contents.

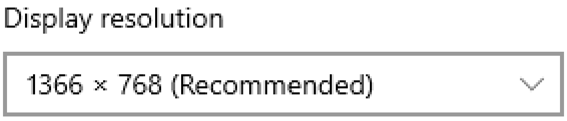
GUI Design (Task 4A)

The best is to create and design your interface in Delphi and save it in the phase 2 folder. You may make changes to your design in phase 2 once you start coding. However, you may **not** change your chosen scenario from phase 1 in phase 2. If you wish to do so, you need to redo phase 1.

Once you have designed your program, make screenshots of tabsheet 2 and 3 and place them in your document. Use the snipping tool or snip and sketch on your computer to create them. Your program should be running when you create the screenshots.

The TWO interfaces for the second and third tab sheets needs to be pasted under the main heading **GUI Design** in the **Fase 1** Word document.

□ If you have a large screen at home, change the resolution to



Display resolution

1366 × 768 (Recommended)

so that your program still fits on a school computer. Right click on your desktop > Display Settings
Also make sure to test your program at school to ensure that it first on the screens.

You are required to have **at least THREE** forms/tab sheet on a page control (you may have more)

□ Welcome / Login page
□ User 1 (Client Page)
□ User 2 (Admin Page)

Make sure of the following:

□ Name of the form/tab sheet.
□ Your program must have a logical flow where input is at the top and processing buttons below or to the right of input. Output can be at the bottom or to the right of the screen after processing.
□ Labels should have clear instruction to the user as to what to do
□ Interface must fit your scenario: include images and use appropriate colours.
□ Include at least one of each of the bullets below somewhere in your program:
  - Edit
  - SpinEdit – all integer input must make use of a SpinEdit
  - RadioGroup / ComboBox
  - RichEdit
  - Label
  - Image
  - DateTimePicker / CheckBox

□ All objects must be renamed with the correct pre-fixes, including your Form, page control and each tab sheet.

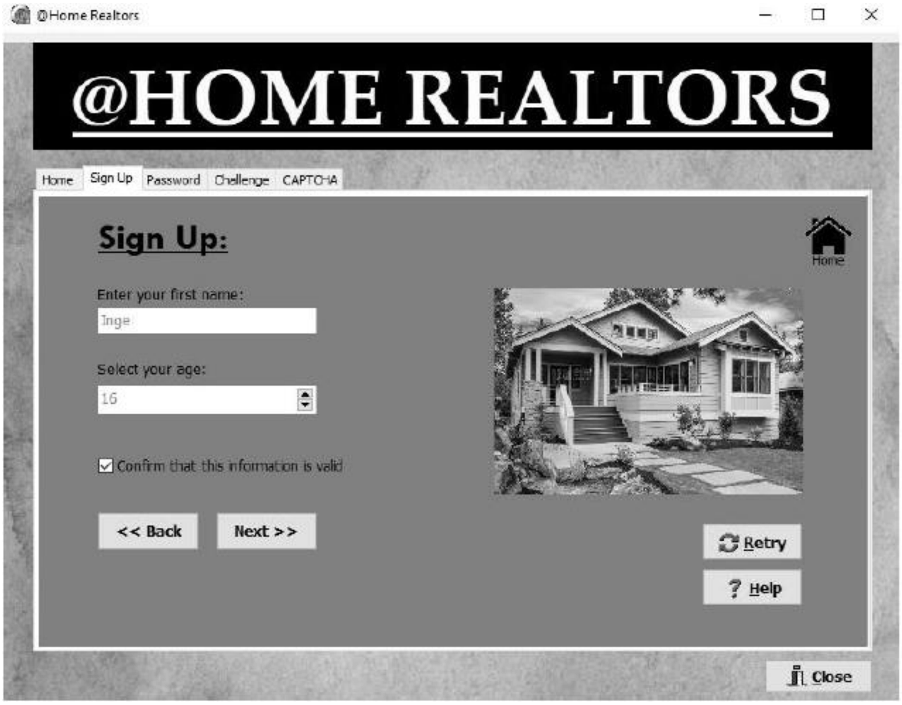
□ No names of object should be visible to the user and the appropriate property of the object must be changed to display appropriate messages to the user.
□ Default values for all edit's i.e. change the text property.
□ Group objects together by adding a **GroupBox** first and then adding objects to the GroupBox. Change the caption of the GroupBox.

Each form/tab sheet needs to have the following buttons/menus:

□ Next
□ Home – except on the home screen
□ Back - except on the home screen
□ Reset – on any screen where the user needs to enter data
□ Help – In Phase 2 you will write code to display information to assist the user in using the tab sheet / form. **(This earns you marks for Program Documentation in Phase 2)**
□ **Every edit needs an example of data that would be entered by the user** in the text property. This not only helps the user, but also makes testing and marking your program easier.
□ Only one screenshot per line of only the tab sheet, not the whole Delphi interface.

Note that the program does not need any functionality yet.

Example of a screenshot



Sign Up:

Enter your first name:
Inge

Select your age:
16

☑ Confirm that this information is valid

IPO (Task 5) – Input, Output, Validation and Processing

You need to insert TWO screenshots: 1 screenshot for the second tabsheet for user 1 and a second for the third tabsheet for user 2. Below each screen shot you need to complete the input, output and validation tables **for each screen** (**TWO** tabsheets) as described below:

INPUT:

- ☐ NB! This table is for **input** only! The GUI component will **NOT** be a button, label, ShowMessage or RichEdit. The **ONLY** exception would be for random (computer generated) input making use of randomrange. The object in this case would be the button since it needs to be clicked to store the input.
- ☐ Ensure the most appropriate components are used on your tab sheets.
- ☐ At least THREE data types in total must be covered in this table and the next input table for the next tab sheet.
- ☐ Add to this table to include ALL your input for this tab sheet.

Example of input:

Tabsheet 1: Name of tabsheet			
Variable name	**Data type**	**Format (size, M/F for gender, yyyy/mm/dd)**	**GUI Component used**
sName	String	String with no more than 20 characters	Edit box
cGender	Char	Use a radiogroup to store M/F when user clicks on Male / Female option	Radio group

OUTPUT:

You need to describe **all** the output you will display on this tab sheet. The < > indicate where a variable would be displayed on an option between two values. Add as many rows as you need but include **at least TWO** different components.

Create a table as below to describe **each** form's (tabsheet) output.

Example of output for a form/tabsheet:

Tabsheet 1 (Name of tab sheet)		
Output	**Format (Type, size)**	**Output Component**
Cost	You owe: <Cost as a currency with two decimal places>	Label
Correct or incorrect answer	Your answer is: <Correct / Incorrect>	Label
Name, Surname and Total Cost	In neat columns with headings Name, Surname and cost. Cost will be displayed in Rand currency	RichEdit

INPUT VALIDATION:

Complete the table below for data validation. Follow the instructions below. This can be for one or both of your screens (tabsheets) above.

For example:

- □ Validate if an edit box has been left empty (presence check)
- □ Test if a user selected something from the ComboBox / Radiogroup (presence check)
- □ Choose THREE of the following:
    - o Test if an integer number is in a certain range (range check)
    - o Test the length of a string entered or for a special character in a string(string data type)
    - o Use TryStrToFloat for real input (real data type)

Example of Data Input validation (**Include validation for one / both of your tabsheets in this table**):

What will be validated	Associated error message
I will test if the user entered an email address	'Please enter your email address'
I will test if the age entered is smaller than 18	'Please enter an age of younger than 18'

PROCESSING:

You need to **list** and write **algorithms** for any **TWO** places where **significant** processing will be done.

You are now busy planning processing so note the following:

- This should **NOT** include validation processing.
- Only input and output will score NO marks.

Explain what it will do and the add the algorithm. In the algorithm ensure that it is clear where each variable gets a value.

TIP: To insert ← in Word type < (smaller than key) followed by 2 – directly next to each other and word will create an back arrow. Or you may use a = to assign a value to a variable.

Example:

What processing will be done	Algorithms, formulas, etc.
Describe:	Algorithms:

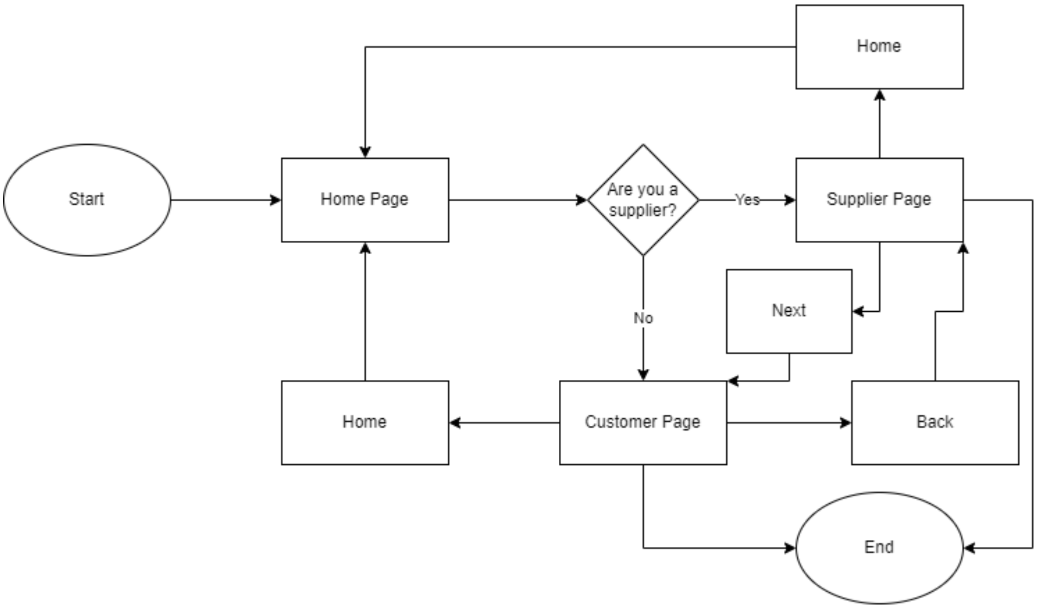
Determine whether the user is under the age of 18 and if so give a discount of 20%	rCost ← input from user (or rCost = input from user)
	iAge ← input from user
	
	if iage < 18
	  rCost = rCost – (rCost x 20/100)
	
	display rCost

2nd process	

Navigation / flow between screens & GUI design (Task 3):

- Your program will need at least three forms/tabs on a page control. Draw a diagram to indicate how your program will be used starting at the home screen and then showing with arrows where the user will go, depending on what they click on.
- Your aim is to show how the navigation buttons, Home, Next, Back will work
- Do **NOT** include the help and reset buttons in your flowchart.
- Go to https://www.draw.io to create your flowchart and export your diagram as 'n picture file to be added to your document.

Example of Navigation:



Task Definition and user story (Task 1)

Task Definition (Task 1A)

Describe your scenario in at least **150** words by completing each heading below: (A total of 150 words for all four subheadings)

- **Topic -** Describe which topic you chose from the examples or your own.
- **Purpose of program -** Describe why the "company" need your program.
- **Possible solution -** What will the program do to meet these needs above. Describe how your program will work. Include a description of EACH of your tabsheets and how the user will interact with your program. Imagine you are writing the "About this game" section for the Google Play store. Here is an example from the play store:
- **Scope -** Explain what your program cannot do i.e. your program's limitations.

User Story (Task 1B)

Complete the given table in detail to clearly describe the users that will use your program – you need at least **TWO** users. They must use the program differently.

- **User Stories (**You need to be able to answer the following questions for each user)

WHO	As a … (user/role)
WHAT	I want to … (program feature/action)
WHY	So that … (value/reason)

Examples: (Each user must perform **FOUR** tasks i.e. you will have 8 rows in this table)

WHO	WHAT	WHY
As a … (user / role)	I want to … (action)	So that … (reason)
Owner	Add new products	Clients can buy new products
Client	Select a clothing item	I can buy new clothes online
Client	Register	To order food online
Receptionist	Select type of service client requires at hair salon	I can make a client booking

Acceptance Test (Task 2)

List at least **FOUR** functions the program will perform. Make sure that the list includes processing that will need to be done before the user views the data.

Examples:

- A customer can view their outstanding balance.
- A administrator can add a product.
- A customer can search for available cars

Data Dictionary (Task 4B)

Complete the following under the Data Dictionary heading:

User defined methods: Complete the table by listing your self written function(s) and or procedure signatures, with an explanation of how they will be used.

Signature	Explanation
Function CalculateVAT(pCost:real):real;	Calculates and returns the VAT of the cost received as a parameter.
Procedure CreateHeadings;	Clears the RichEdit and create 3 headings with tabstops the name, surname and cost in the rich edit.

Now that you are done, go back to the marksheet in the official PAT document for phase 1 and mark your own phase 1.

Phase 2 Due date: 23 October 2023

You will be required to hand in a single Delphi program.
NB!! Make sure that your program is running at ALL times. A program that does not run due to syntax or runtime errors will lose lots of marks. Place the code in comments that is causing the error. Once an error is preventing your program from running do not continue to code. First fix that error and if you cannot find it bring it to school so that I can assist you.
Ensure that your program contains the following:

- □ Make the changes as indicated in your feedback of phase 1 and ensure that all is done correctly for the GUI as indicated under the heading "Minimum requirements for the Interface" of this document.
- □ Apply your planned validation and add to it if you did not score full marks in phase 1. Ensure you have proper feedback to the user as to what is expected from them.
- □ Use a variety of data types for your variables – include at least 4 different types.
- □ Use local and global variables appropriately
- □ Use the naming convention for variables throughout your program iNumber sName etc.
- □ All your objects should be renamed making use of the prefixes i.e. edt, red, btn, frm. Remember about your form and its caption.
- □ Use the most effective code to solve a problem i.e. use a nested-if or case-statement where appropriate.
- □ Ensure all Edits and InputBoxes have default values.
- □ Take time to make your output display properly in RichEdits and labels. Add label and column headings and ensure data is in neat columns. Leave lines open to improve readability. Remember use ffCurrency, where appropriate.
- □ It should be easy for the marker to get into your program for both users. Passwords for both should be made available on the form as default input.
- □ Use the sections in this document "PAT minimum requirements" to complete your phase 2.
- □ To go to a next tab sheet use this code: pgcMain.ActivePage := tsAdmin; Where pgc is the name of your page control and ts is the tab sheet you want to go to.
- □ In Form Activate add the code above to ensure that your program will always open on the home page when the user runs the program.
- □ To display a GIF, use an Image component and load the .gif file as usual. Then add this code where you want the gif to start moving.
  (imgCAT.Picture.Graphic as TGIFImage).Animate := True;
- □ **While you are coding make comments for sections of code in each event.** These comments are not the ones mentioned in this document but your own comments. For example: **//Calculating VAT on total cost** (This earns you marks for **Program Documentation**)
- □ Make sure there is a help button with necessary help provided for each form/tabsheet (This earns you marks for **Program Documentation**)

You could also teach yourself how to use text files to store data. But only do this once you have completed the rest.

Once you think you are done allow someone else to use your program. Do not tell them where to go or what to click on. Make a note of where you should make changes to your program to make it more user-friendly.

After you made these changes go to the marksheet of phase 2 in the official PAT document and mark your own phase 2.

FOR THE INTERVIEW:

□ Make sure you have tested your program with valid, extreme and invalid data
□ Demonstrate all functionality for the marker for all your different users
□ Make sure you can explain selected code
□ Create a ReadMe.txt text file with instructions for the marker on how to log into you program and supply username and password for each of your users.

ASSESSMENT SUMMARY:

Assessment Summary

Phase	Task	Focus	Maximum Mark	Mark Obtained	Mark Moderated
1	Task 1	Task definition and user story	8		
	Task 2	Acceptance test	4		
	Task 3	Navigation / flow between screens	4		
	Task 4	Design a screen	8		
	Task 5	IPO table and data validation	16		
2	Task 6	Create TWO Screens in Delphi	4		
	Task 7	HCI principles for TWO screens	4		
	Task 8	Develop the Code	20		
	Task 9	Testing and data validation	8		
	Task 10	Documentation	4		
3	General	Finale product and impression	20		
	Total		**100**		
	Final mark (100%)				