Grade 12 PAT 2025 - INFO

Topic

TRANSPORTATION MANAGEMENT SYSTEM

Transportation plays a crucial role in a country's economy and the daily lives of its citizens.

The different modes of transportation, routes, logistics, freight, delivery services, infrastructure, safety rules and regulations are valuable to a country in enabling the planning and maintenance of transportation management systems that are vital towards economic growth.

Projects in the scenario above could include the following topics relating to transportation:

1. **Route/Trip planning:**

   A system that can be used to:

   - Plan/Compile the best/most economical route from the point of departure to the required destination. The system could be used to assist employees of delivery services, tourists and other travellers in the public and private sectors.
   - Plan routes/trips which include different modes of transportation to travel from one place to another

2. **Delivery/Courier services:**

   System to assist with the planning, administration and logistics of:

   - Private and public courier services, such as the 'Post Office', 'Aramex', 'Fastway', etc. to deliver parcels
   - The transportation of goods in bulk using the rail network
   - The rapid transit of high-value cargo/high-profile visitors over long distances using aeroplanes and/or helicopters
   - The shipping of goods for global trade/moving large volumes of goods via container ships, tankers and bulk carriers

3. **Infrastructure and fleet management:**

   System to:

   - Keep track/administer the state and condition of vehicles/trains/aircraft/ships, etc. at a company/school/municipality, etc.
   - Compile and update the maintenance schedule of vehicles/trains/aircraft/ships, etc. at a company/school/municipality
   - Compile and update the duty/trip schedule of vehicles/trains/aircraft/ships, etc. at a company/school/municipality, etc.

4. **Licencing of vehicles:**

   System to manage the licensing of drivers and vehicles

5. **Duty roster system:**

   A system to compile a working schedule for employees in the transportation industry/section of a business/public sector. Employees can include drivers/pilots/sailors/train conductors/taxi drivers, etc.

6. **Environmental impact:**

   System to provide information about:

   - Options to promote sustainable practices/options in transportation
   - Companies and techniques that can be used to counteract pollution and other negative aspects related to the use of the different modes of transportation

7. **Economic influence:**

   System to assist with processing information related to transportation in a company, such as income, expenditure, profit, loss, etc.

8. **Accessibility services:**

   System to provide/process/implement information on:

   - Accessibility services in transportation to ensure that people with disabilities can travel safely and comfortably when using public transport
   - Wheelchair-accessible route planning and station facilities
   - Paratransit services, such as door-to-door specialised services/on-demand services that allow flexibility based on passenger disabilities
   - Audio-visual announcements and assistance for passengers with disabilities
   - Vehicle modification systems, wheelchair lift systems and systems to control adaptive equipment for drivers and passengers with disabilities

9. **Safety and security systems:**

   System to manage and process:

   - CCTV (closed circuit television) surveillance and incident reporting
   - Emergency response coordination and communication tools

10. **Customer service in transportation:** System to manage and process:

    - Customer satisfaction surveys and feedback collection of Complaints, responses and resolutions

Choose an application related to transportation management services and do research on the information system requirements.

You are not limited to the list of ideas above, but you need to keep within the overall theme of transportation management services. Note that you need to choose data and functionalities (services) in such a way as to develop a well-rounded application related to the topic.

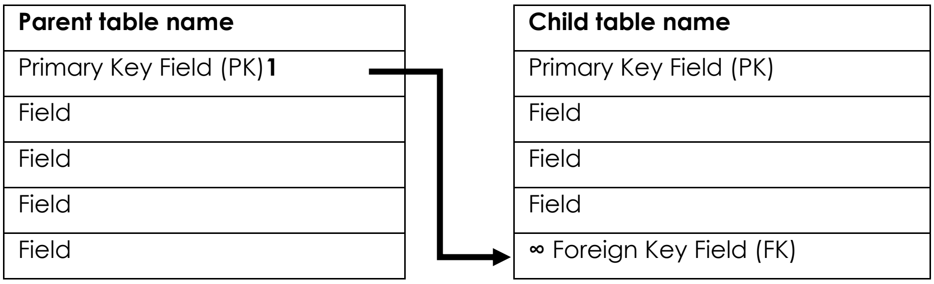
NOTE: Your final program must comprise **ONE** single project with logically related parts.

Phase 1 - Final Due date: 25 April 2025

Designing the Database

Design/plan your database on paper and bring it to class. (you can draw it or design it in Word). This planning should include the PK's and FK of a relational database with **at least TWO tables linked with a one-to-many-relationship** – see detailed requirements below.

Example of the tables:



Parent table name
Primary Key Field (PK)**1**
Field
Field
Field
Field

Child table name
Primary Key Field (PK)
Field
Field
Field
∞ Foreign Key Field (FK)

Once your database has been approved and feedback was given, you need to create and complete it in Microsoft Access. Ensure there is a copy of the database is in your Phase 1 folder. Give the database a descriptive name.

- □ Save the database as a .mdb and not an .accdb
- □ The name of all your tables should start with **tbl**
- □ Use **at least one of each** of the following data types across the two tables for processing:
  - o Short text
  - o Number / Currency
  - o Yes/No
  - o Date/Time
- □ Change the size of all the text data type fields to be less than the default (E.g. 255 for Short Text).
- □ If you are adding a name and surname it should be in two separate fields.
- □ If you chose a number data type and the value is an integer, change the format to integer as the default is double.
- □ Each table needs a **primary key**.
- □ You need at least **5 fields** in each table, but no more than 8.
- □ You need at least **10 records** in each table.
- □ Ensure that each foreign key in the child table has a parent.
- □ If the parent primary key is an AutoNumber the foreign key in the child table must be a long integer.
- □ Ensure that you link your two tables with a 1 – ∞ relationship (enforce referential integrity must be applied).
- □ Do not use the following data types in Access:
  - o Long text
  - o OLE object
  - o Hyperlink
  - o Long integer except for only one field if needed as foreign key.

MS Word document - Requirements and examples

Use the template provided: **(Rename it with your own Name and Surname !!)**

Built-in headings were used to create an automatic table of contents. You can create a table of contents using the headings. Please submit your final document as a PDF-document.

Use the **headings** below in your PAT and complete it as described. The scenario and scope are often easier to do after you completed the other sections in Phase 1, but it has to be the first part of your document. We will complete the document in a different order as it appears in the marksheet, but at the end your document should follow the order of the marksheet

1A. SCENARIO / SCOPE (Task 1A): (Describe your scenario in 200 words by completing each heading below – a total of **200** words for all four subheadings)

- **Topic** (Describe which topic you chose from the examples of your PAT document or your own. You have to stick to a topic that relates to **Transportation Management**)
- **Purpose of program** (Describe the purpose of your program – why does the company need your program)
- **Possible solution** (What will the program do to meet this need. Describe how your program will work. Include a description of each of your three forms / tab sheets and how the program will interact with your database.)
- **Scope** (Explain what your program cannot to)

1B. USER REQUIREMENTS (Task 1B): (Complete the given table in detail to clearly describe the users that will use your program – you need **at least TWO users**)

- **Users** (Name each user e.g. Customer or Administrator and give a short description of who the user is in the greater scenario of your topic.)
- **Role, Activity and Limitations of the users (**Complete the table)

Role	What role will they perform in the program, e.g. *change existing data according to customers' input*
Activity	What will they be able to do, e.g. *press a button and see all the transactions / bookings*
Limitations	What can't they do, e.g. *can't view reports if the user is not an administrator*

Example:

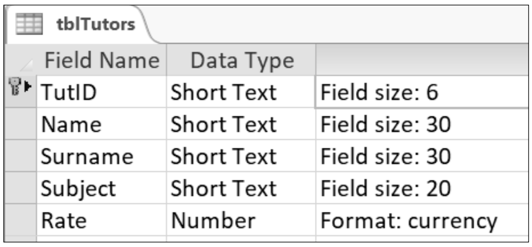
User	Role	Activity	Limitation
Clients	Browsing and ordering the products offered by the store.	They can login, search for specific items or categories, add items to their cart, remove items from their cart and checkout via online payment.	They cannot add new products or change any of the product details.

2. DATABASE DESIGN (Task 2) :

Place a screenshot of each of your tables here. Two screenshots per table, of the design view and one showing the data. Also take a screenshot of your relationship and paste it here.

- **Indicate the field format and size in the description column in Design View!!**

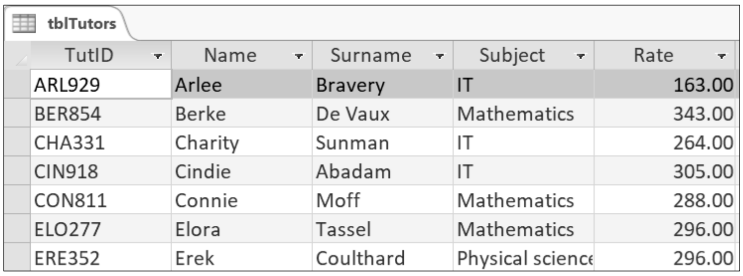
Example of design view:



tblTutors

Field Name	Data Type	
TutID	Short Text	Field size: 6
Name	Short Text	Field size: 30
Surname	Short Text	Field size: 30
Subject	Short Text	Field size: 20
Rate	Number	Format: currency

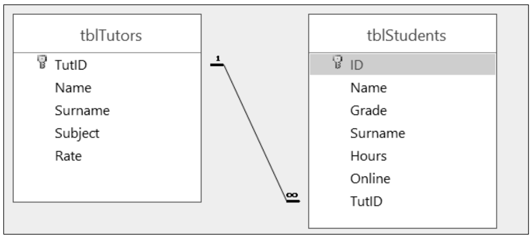
Example of datasheet view:



tblTutors

TutID	Name	Surname	Subject	Rate
ARL929	Arlee	Bravery	IT	163.00
BER854	Berke	De Vaux	Mathematics	343.00
CHA331	Charity	Sunman	IT	264.00
CIN918	Cindie	Abadam	IT	305.00
CON811	Connie	Moff	Mathematics	288.00
ELO277	Elora	Tassel	Mathematics	296.00
ERE352	Erek	Coulthard	Physical science	296.00

Example of relationship:



tblTutors
TutID
Name
Surname
Subject
Rate

tblStudents
ID
Name
Grade
Surname
Hours
Online
TutID

3. DATA DICTIONARY (Task 3)

3.1 Class description AND class diagram (Task 3A):

Clearly explain the purpose of your class in your program and **how it will be used to add value** to your program.

Class diagram: Follow the steps below to create a class diagram:

- Place the class name at the top (TBooks)
- List the fields / attributes with data types. All fields should be Private
- List the methods:
  - Those that are public needs + and private –
  - You will not add Procedure or Function before the method name, but rather indicate that the method is a function indicating it returns a data type at the end.
  - You will add the word **Constructor** to your constructor method
  - All parameters need a data type

- Methods that will always be public (+)
    - Constructor
    - Mutators (Set Procedures)
    - Accessors (Get Functions)
    - ToString (Function with string data type)

Example of an UML Class Diagram:

Tbooks
- fTitle, fISBN : string; - fSold : integer; - fCost, fSelling : real; - feBook : boolean;

+ Constructor Create(pTitle, pISBN : string; pCost: Real; peBook : boolean);
- CalculateProfit : real;
+ SellingPrice(pPercentage : real);
+ GetSellingPrice : real;
+ SetBooksSold(pNum : integer);
+ ToString : string;

3.2 TEXT FILE (Task 3B): You have to use a text file. Give the name of the text file.

If there was extra fields/an extra table needed for your database that did not fit into your database or that is not part of your main functionality, use these fields / table as ideas for your textfile, for example if you have a type field in one of your tables that a user can select from a combox, you can keep these types stored in textfiles and arrays (Lookup fields).

Text file:

- **Purpose**: **Describe** here what you will be using it for in your program. Include how/when data will be written to the text file during the running of your program.
- **Format:** Include the format of the text file. For Example: <Name>#<Surname>#<DOB>. You may not put all your data from your database in your text file. You can use one field in the file, to link the primary key of a table, to be able to extract data from the text file, according to data from your database. Or you can use your text file to populate data in a combobox for a field to be saved in your database.
- **Extract of data:** Place three lines as an example of your text file here.

Example:

Name of text file	Format/Extract of data	Purpose
txtTransactions	Format: <Date>#<Customer name>#<Customer surname>#<Amount paid> Example: 2023/02/16#Jason#Smith#1055	Each time the transaction saved, the details of the transaction will be written to the textfile.

3.3 ARRAYS (Task 3C): You have to use a parallel array i.e. at least two arrays that run parallel.

- **Purpose/Description:** Clearly explain how your arrays will be used in your program.

  It needs to involve **significant** processing. Include a list of what you will do explaining at least 2 of the below in **detail**:

  - Search with a while loop to find one / many
  - Calculations
  - Highest, lowest or average
  - Sort
  - Add to arrays
  - Delete from the arrays

- **Where/how it will be used:** Clearly explain how your array will be populate i.e. how and where in your program it will receive values. It could be the data you read from your text file. (Recommended)

Example:

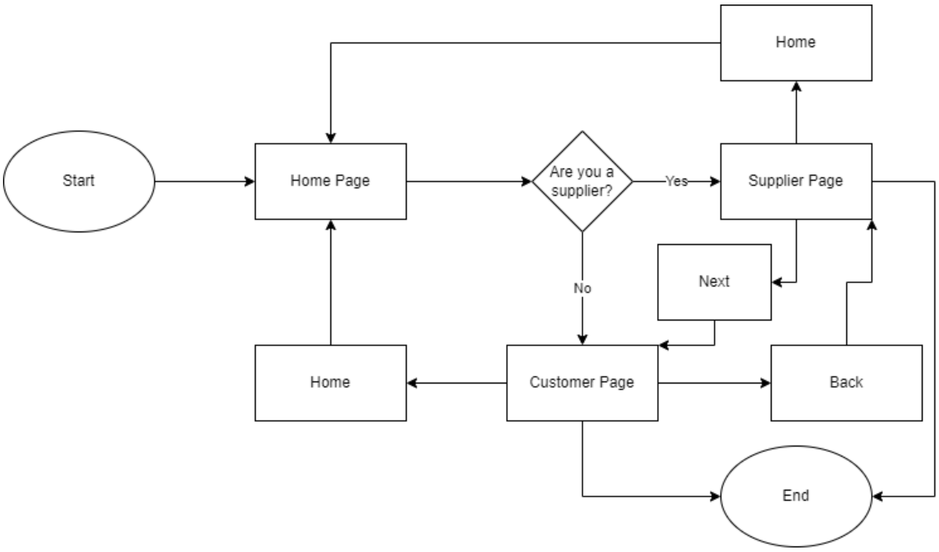
Name of array	Datatype	Purpose/Description	Where/how will it be used
arrEmail	String	Stores the email addresses of all the users stored in the applicable textfile	It will be used when a customer is logging in to see if their input matches their email address stored in the text file

4 NAVIGATION (Task 4A):

Go to https://www.draw.io/ and export your diagram as 'n picture file to be added to your document.

- Your program will need **at least three tabsheets** 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.
- Indicate the sequence and steps they will use to navigate through your program.
- Remember to also include how the user can go back to previous screens.

Example of Navigation:



4.1 GUI DESIGN (Task 4B)

You are required to have **at least 3 tabsheets** on a page control. **Do NOT use separate Delphi Forms**. For Example:

- Welcome form that will then take the user to the appropriate forms.
- 1 or more Admin Form(s)
- 1 or more Client Form(s)

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 ALL your tabsheets and place them in your document.

Make sure of the following:

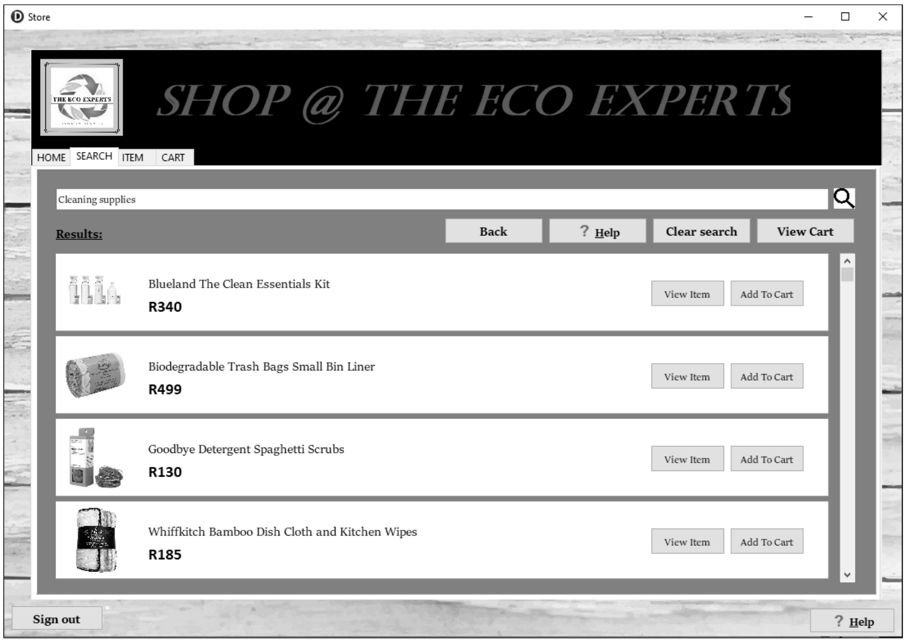
- □ Name of the form/tab sheet.
- □ Only one screenshot of only the form in runtime, not the whole Delphi interface.
- □ Your program must have a logical flow where input is at the top and processing buttons below or to the right of input. Output can be at the bottom or to the right of the screen after processing.
- □ Pick appropriate components like ComboBoxes for the user to select from when only certain options are available to select from like the type of home.
- □ Use CheckBoxes for input into Yes/No fields.
- □ Use a DateTimePicker for input for a Date/Time data type field.
- □ SpinEdits for integer values.
- □ **Do not use InputBoxes unless there are no set values to choose from**.
- □ Remember to rename your form and all your other objects using correct prefixes for the components i.e. edt, red, cmb, rgp, frm. There are marks allocated for this in phase 2.
- □ Change the captions, text etc. properties of the object so that it does not contain the name of the object.
- □ Remember to change the caption of each form/tabsheet so that it does not contain the name of the form any longer.
- □ Make use of colours and pictures that fit your scenario.
- □ Use labels to ensure that the user will not exactly what to do with that component.
- □ **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.

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

- □ Next
- □ Home – except on the home screen
- □ Back - except on the home screen
- □ Help – In Phase 2 you will write code to display information to assist the user in using the tabsheet. **(This earns you marks for Program Documentation in Phase 2)**

Note that the program does not need any functionality yet.

Example:



Cleaning supplies

Results:

Blueland The Clean Essentials Kit
R340

Biodegradable Trash Bags Small Bin Liner
R499

Goodbye Detergent Spaghetti Scrubs
R130

Whiffkitch Bamboo Dish Cloth and Kitchen Wipes
R185

5 SOFTWARE DESIGN TOOL – IPO (Task 5)

5.1 DATA INPUT

You need to describe **all** the input of **at least TWO** of your interfaces with **SIGNIFICANT** input.

Example of input: **(NB – LOGIN SCREEN WILL EARN NO MARKS for Data Input)**

Tabsheet 1 (Name of tab sheet)				
Input	**Source**	**Data type**	**Format**	**Component used**
edtProductName	Keyboard	String	Less than 50 characters	TEdit
sedPrice	Keyboard/ mouseclick	Integer	Any positive integer	TSpinEdit

5.2 INPUT VALIDATION

You need to validate **at least 4 different data types** and **4 inputs from the user** that will include testing if input is empty and testing if they have selected an option from an object like a ComboBox. If you do your data type testing on user input, you will only need to validate 6 inputs. For example:

- Test if an edit box has been left empty (presence check)
- Test if a user selected something from the ComboBox / Radiogroup / ListBox
- Test the length of a string entered or for a special character in a string(string data type)
- Test if an integer number is in a certain range (integer data type & also range check)
- Test if the file exists (text file data type)
- Use TryStrToFloat for real input (real data type)
- Test if a date is earlier / later than a specified date (datetime data type)

Example of Data Input validation (**Use same form / tabsheet as for data input above**):

Input	Validation technique and description	Associated error message
EdtEmail	Presence check – check if user entered their email address	'Please enter your email address'

5.3 DATA PROCESSING

You need to **list** a total of **8 processes** to be done. You need to write **algorithms for 4** of these processes.

You are now busy planning processing so note the following:

- **Input**, **validation** and **output** should not be part of the algorithms or listed processes.
- **SQL processing will not earn any marks for the algorithm but can be part of the 4 processes that you need to list**.
- Ensure that the four algorithms you write has significant processing.
- Reading from the text files, string handling and your array manipulation should provide you with proper algorithms to write.

What processing will be done	Algorithms, formulas, etc.
List:	Algorithms:
Display all houses available when user did a search for suitable house.	
Determine whether a product has made no sales so it can be deleted	Loop while end of table Products has not been reached If tblProducts['AmountSold'] = 0 ComboBox Item ← tblProducts['ProductName'] End loop Delete ← user input from combobox Loop while end of table Products has not been reached If Uppercase(Delete) = Uppercase(tblProducts['ProductName']) Delete from tblProducts End loop

5.4 DATA OUTPUT

You need to describe **all** the output of **TWO** of your interfaces with **SIGNIFICANT** output. Create a table as below to describe each form's output. **NOTE: you may describe your output in the DB Grids only once.**

Example of output for a form/tabsheet:

Tabsheet 1 (Name of tab sheet)		
Output	**Format**	**Component**
Cost	Currency, with two decimal places	Label
Name, Surname and Total Cost	In neat columns with headings Name, Surname and Cost,displayed as currency with two decimals	Richedit

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 - Final Due date: 1 Augustus 2025

You will be required to hand in a single Delphi program that is linked to your Database.

Step 1: Copy your database from the phase 1 folder to the phase 2 folder.

Step 2: Make all the changes as indicated in phase 1 to the database in **PHASE 2.**

Step 3: Make all the changes to the interface as in indicated in phase 1.

Step 4: You will be assisted to link the database to the program if you kept to all the due dates.

Step 5: You need to have at least 10 records per table and 5 fields per table.

Step 6: It is a good idea to keep a copy of the Database in another folder, so that you can go back to the original state it was in later.

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.

Do not spend hours on logging in. This count no marks. Focus on your SQL and database manipulation (DB work makes up more than 30% of your PAT marks). Add your fancy things later.

Ensure that your program contains the following:

- □ Use a variety of data types for your variables
- □ 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 your text file. Place a **comment** **//text file**
- □ Use your parallel arrays. Place a **comment** **//arrays** where you do processing on arrays.
- □ Implement your OOP class. Place a **comment** **//OOP**
- □ Use a variety of sources for your input: i.e. keyboard, mouse, text file, array or database
- □ Use appropriate data types. For example: do not store a number as a string unless it is an ID or telephone number in which case it should be stored as a string.
- □ Allow the user to rather select an option from a ComboBox instead of letting them enter, where applicable.
- □ **Apply the validation you planned in phase 1. Comment** **//Validation** above this code
- □ **Add validation for ALL input before you insert data into the database.**
- □ If you **did not use an OOP class use of functions and procedures with parameter passing to score some marks.**
- □ Use good programming techniques
- □ Use the most effective code to solve problems i.e. use a case statement for integers and chars, use nested-if statements where appropriate, make your loop stop once you have found what you were looking for in a your arrays.
- □ **Add a dynamic object with an event linked to it that does something when clicked.** Comment next to your .Create and next to the event **//dynamic object**

☐ Take time to make your output display properly in RichEdits and labels. Add labels and column headings and ensure data is in neat columns. Leave lines open to improve readability. Remember use ffCurrency or “Currency” in SQL, where appropriate.

☐ **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: //Reading data from the text file into the parallel arrays. For SQL code use a comment similar to a question that would be asked in a SQL test. For example: //Finding the number of products in stock. **(This earns you marks for Program Documentation)**

☐ **Every Edit and InputBox needs default input**.

☐ **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.**

Database manupilation : Delphi Code (ADO) AND SQL:

You need to have at least one of each of the following:

☐ Your tables must start with **tbl**

☐ **ORDER BY** (Sort records in a table)

☐ Insert a new record (Recommendation : Use ADO for Insert, Update and Delete)

☐ Delete **(Child record recommended)**

☐ Update at least **TWO fields in one query** OR **TWO updates** changing different fields.

☐ Show ALL / selected fields in a record

☐ A **Where with a combination of AND/OR** in a single query / **GROUP BY with HAVING**. **(Complex selection query)**

☐ **TWO** queries **with calculations (aggregate functions)** like **min**, **max**, **count**, **sum** or **avg**.

☐ At least one SQL Select / ADO reading from both tables.

☐ A Where with a variable from user input

☐ Comments: NB Make comments where your DB manipulation is done ADO / SQL

Documentation (Notes/Comments):

You need to have the following:

☐ Comments for the marker - //Validation or OOP or Text file or Array or Dynamic Object. This will help the marker and moderator to find where you’ve implemented the required code. It will most definitely makes the marking process easier.

☐ Comments in your coding – This is notes for a programmer / marker (Every line of code does not need a comment BUT every EVENT / FUNCTION / PROCEDURE needs good explainatory comments.)

☐ Notes for the user : A readme.txt text file / Word document with simple notes on how to use your program. If there is sufficient help on each form, there is no need for an extra file / document.

Exceptional features:

Ideas to score 2 marks:

- Any animation where an object change when you hover the mouse over it.
- Gifs
- Playing a video.

Ideas to score 3 marks: (The code should be more complex than the 2 of out 4 marks)

- Creating animation with a loop or timer loading different files
- A simple graph
- TTaskDialog – A custom dialog in Delphi. This is non visual component, which shows full-featured dialog at run-time. This dialogs include hyperlinks.
  See link provided : http://delphiprogrammingdiary.blogspot.com/2019/04/ttaskdialog-custom-dialog-in-delphi.html?m=1

To score 4 out of 4 marks:

The code must NOT be in ou syllabus and it must be working 100%. It must be highly complex. You must be able to explain the code if you borrowed it from someone else.

- Creating an App for a phone
- Sending an OTP from your program

VERY IMPORTANT:

As you can see to score 4 out of 4 marks, is a LOT of work for 1 mark. First complete your entire PAT and go for a 2 or 3 out of 4. It is not worth the time you will put in to get the 4 marks. Do your best on the rest of your PAT.

Focus on your database work – The database work across Phase 1 and 2 accounts for a 3rd of your PAT marks. Focus on what is important. Do the fancy stuff later!

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.** Only now spend some time on the exceptional features below

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**

☐ Supply username and password for each of your users for the marker to log into you program.