# https://www.hhs.co.za/images/logo.png

# 2023, PAT PHASE 1

|  |  |  |
| --- | --- | --- |
| **ADMIN NO**: [ ADMIN NO] |  | **NAME**: [FULL NAME] |

## SCENARIO/SCOPE

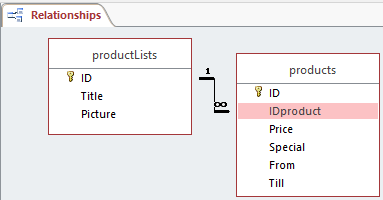
## USER REQUIREMENTS

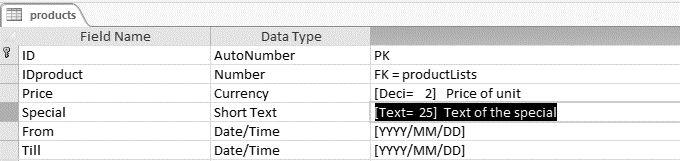
### Navigation/Description of Flow Diagram

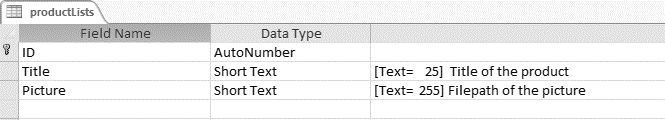
## DATA STRUCTURES

### Database Design

|  |  |
| --- | --- |
| * All fields relevant * Type and size of fields well-chosen | * Relational * Normalised |







## Class Description and Class Diagram

|  |  |
| --- | --- |
| * Attributes * Methods * Scope of methods | * Method type * Return types * Parameters |

|  |
| --- |
| **Class Name**: |
| **Attributes / Properties/ Members**  -  **Public ( + )**  **Private ( - )** |
| **Methods**   |  |  |  |  | | --- | --- | --- | --- | | Type | Method Name | Parameters | Return | | C | Create |  |  | | P | set |  | None | | F | get |  |  |   **Public ( P )**  **Private ( F )** |

|  |
| --- |
| **Class Name**: TValidator |
| **Attributes / Properties/ Members**   * error\_message: String   **Public ( + )**  **Private ( - )** |
| **Methods**   |  |  |  |  | | --- | --- | --- | --- | | Type | Method Name | Parameters | Return | | C | Create Validator |  |  | | P | clearErrorMessages | None | None | | F | getErrorMessages | None | String | | F | isValidEmail | email: String | Boolean | | F | isValidRange | minOf, maxOf : Integer | Boolean | | F | isValidFutureDate | dateOf : Date | Boolean |  |  |  |  |  | | --- | --- | --- | --- | | P | setErrorMessage | validator, Text : String | None |   **Public ( P )**  **Private ( F )** |

Text file(s) and Array / Advanced Programming Concepts**Text file(s)**: SQL statements will be stored as text in selectable Text-files.

The SQL Select Statements will follow the standardized format of:

|  |
| --- |
| **SELECT**  <Fields>  **FROM** <Table>  **WHERE** <Condition>  **GROUP BY** <Aggregate Fields>  **HAVING**  <Aggregate Field Condition>  **ORDER BY** <Fields> |

An Example is shown below:

|  |
| --- |
| <<PASTE YOUR EXAMPLE>> |

The following text-files will also be used:

* Login Text-files per each user
* Hint – language file
* Error Messages – language file
* CSV Data Files for the database to import / export data to and from
* Settings file = file paths/database connection/user settings

**Arrays**:

*Hint and Error messages will be stored into an array structure in the “arrays.pas” unit file. This will be used with all the hints and errors, to allow language changes and text messages to be updated easier throughout the program consistently*.

**Field Names and Tables used into the Delphi Data Dictionary Structure for Analytics**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table** | “product” | “product” | “product” |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | “Price” | “Special” | “From” | “Till” |  |  |  |  |

## GUI DESIGN

|  |  |
| --- | --- |
| * Design fits to program's intended use * Appropriate components * Ease of use, logical flow | * Clearly marked navigation * Friendly dialogue/Help |
| <Insert Image> | |

|  |  |  |  |
| --- | --- | --- | --- |
| Component Name | Data Type | Format | Data Source |
|  |  |  |  |
|  |  |  |  |

## IPO – Software Design Tool

**Interface #1: Subscription**

|  |  |  |
| --- | --- | --- |
| INPUT | PROCESSING | OUTPUT |
|  | |  |  | | --- | --- | | WHAT | HOW | |  |  | |  |  | |  |  | |  |  | | INPUT VALIDATION | | | No Empty Strings |  | | DoB not in future |  | | Email contains @ . |  | | Mobile is 10 digits |  | |  |

**Interface #2:**

|  |  |  |
| --- | --- | --- |
| INPUT | PROCESSING | OUTPUT |
|  | |  |  | | --- | --- | | WHAT | HOW | |  |  | |  |  | |  |  | |  |  | | INPUT VALIDATION | | |  |  | |  |  | |  |  | |  |  | |  |

### Interface #1:

|  |  |
| --- | --- |
| * Design fits to program's intended use * Appropriate components * Ease of use, logical flow | * Clearly marked navigation * Friendly dialogue/Help |
| <Insert Image> | |

|  |  |  |  |
| --- | --- | --- | --- |
| Component Name | Data Type | Format | Data Source |
|  |  |  |  |
|  |  |  |  |

### Interface #2:

|  |  |
| --- | --- |
| * Design fits to program's intended use * Appropriate components * Ease of use, logical flow | * Clearly marked navigation * Friendly dialogue/Help |
| <Insert Image> | |

|  |  |  |  |
| --- | --- | --- | --- |
| Component Name | Data Type | Format | Data Source |
|  |  |  |  |
|  |  |  |  |