```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ProductPerformance
{
    public class Lists
    {
        //-- Product Types - no filter required
        public static List<string> GetProductTypes()
        {

            //- Return the method
            return ProductCatalogueDB.GetProductTypes();
        }

        //-- Products - filter by ProductType
        public static List<string> GetProductsByType(string productType)
        {
            //-- return the method
            return ProductCatalogueDB.ProductsByType(productType);
        }

        #region Short Person Data from Database

        /// <summary>
        /// Short Data of Persons
        /// </summary>
        /// <returns>List of stringsreturns>
        public static List<string> ShortDataAllPersonsInDb()
        {
            //-- Create list to hole items to be returned
            List<string> tempList = new List<string>();

            //-- Loop through data in the text file
            using (StreamReader reader = new StreamReader(@"D:\persons.txt"))
            {
                //-- loop while data in reader
                while (true)
                {
                    //-- Get data in the line
                    string line = reader.ReadLine();
                    //-- drop if no line data
                    if (line == null)
                    {
                        break;
                    }

                    //-- split the line by comma separation
                    string[] fields = line.Split(',');
                    //-- initialise person object
                    Person person = new Person()
                    {
                        Forename = fields[0],
                        Surname = fields[1],
                        Postcode = fields[4]
                    };
                    //- get short data as string
                    string personShort = person.PersonShortData();
                    //-- Add tto list
                    tempList.Add(personShort);
                }

                //-- Return the method
                return tempList;
            }
        }
        #endregion

        #region String is found in list

        /// <summary>
        /// Check if a string is already in a list
        /// </summary>
        /// <param name="listToSearch"></param>
        /// <param name="stringToFind"></param>
        /// <returns>Boolean</returns>
        public static bool StringFound(List<string> listToSearch, string stringToFind)
        {
            //-- tracker variable
            bool stringFound = false;
            //-- Loop through all
            foreach (string value in listToSearch)
            {
                if (value == stringToFind)
                {
                    stringFound = true;
                    return stringFound;
                }
            }

            //-- return true or false: in list, or not
            return stringFound;
        }

        #endregion

        //----------------------------------------------------------------------
        //-- Static lists follow: Items do not change
        //-- These are what are called (hard-code) literals
        //-- Not best practice but quicker
        //-- ------------------------------------------------------------------

        public static List<string> ProductSizes()
        {
            List<string> sizes = new List<string>();

            sizes.Add("250ml");
            sizes.Add("500ml");
            sizes.Add("750ml");
            sizes.Add("1 Litre");
            sizes.Add("Other");
            return sizes;
        }

        //-- Observation 1 is mandatory so 'none' is not an option
        public static List<string> ObservationOneOptions()
        {
            //-- Working list for string to be returned
            List<string> tempList = new List<string>
            {
                "Very poor",
                "Poor",
                "Okay",
                "Quite good",
                "Good",
                "Works well",
                "Works very well",
                "Really good",
                "Excellent",
                "Other"
            };

            //- Return the method
            return tempList;
        }

        //-- Other observations list
        public static List<string> ObservationOtherOptions()
        {
            //-- Working list for string to be returned
            List<string> tempList = new List<string>
            {
                "None",
                "Very poor",
                "Poor",
                "Okay",
                "Quite good",
                "Good",
                "Works well",
                "Works very well",
                "Really good",
                "Excellent",
                "Other"
            };

            //- Return the method
            return tempList;
        }
    }
}
```