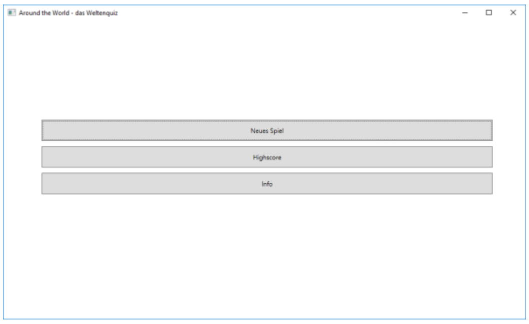
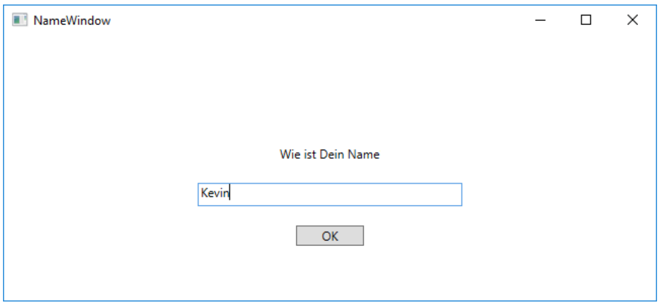
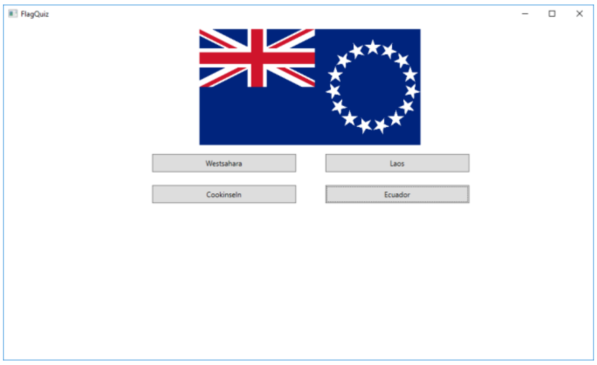
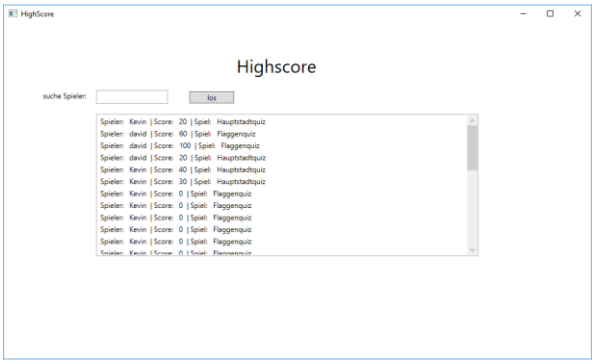
Flag Quiz Game with MySQL Database (in c#)

Database Model, Classes and Objects, Buttons







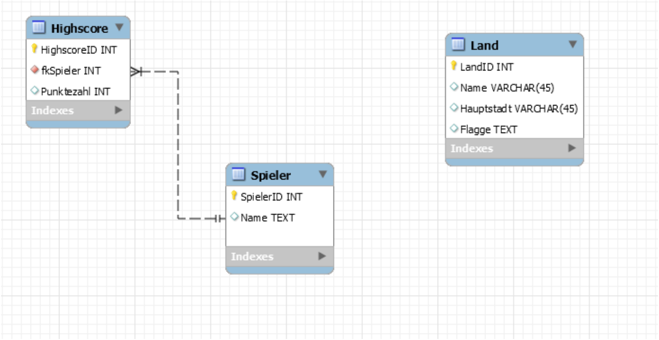


```
Spieler:  Kevin  | Score:  20  | Spiel:  Hauptstadtquiz
Spieler:  david  | Score:  60  | Spiel:  Flaggenquiz
Spieler:  david  | Score:  100  | Spiel:  Flaggenquiz
Spieler:  david  | Score:  20  | Spiel:  Hauptstadtquiz
Spieler:  Kevin  | Score:  40  | Spiel:  Hauptstadtquiz
Spieler:  Kevin  | Score:  30  | Spiel:  Hauptstadtquiz
Spieler:  Kevin  | Score:  0  | Spiel:  Flaggenquiz
Spieler:  Kevin  | Score:  0  | Spiel:  Flaggenquiz
Spieler:  Kevin  | Score:  0  | Spiel:  Flaggenquiz
Spieler:  Kevin  | Score:  0  | Spiel:  Flaggenquiz
Spieler:  Kevin  | Score:  0  | Spiel:  Flaggenquiz
Spieler:  Kevin  | Score:  0  | Spiel:  Flaggenquiz
```

Basically it is a game where the user can choose if he wants to guess flags while the name of the country is shown, if he wants to guess the other way around or if he wants to guess the capitals of the shown countries. The user can type in his name at the beginning of the game and his score gets saved in the database.

Database:

The members of the database look like that:

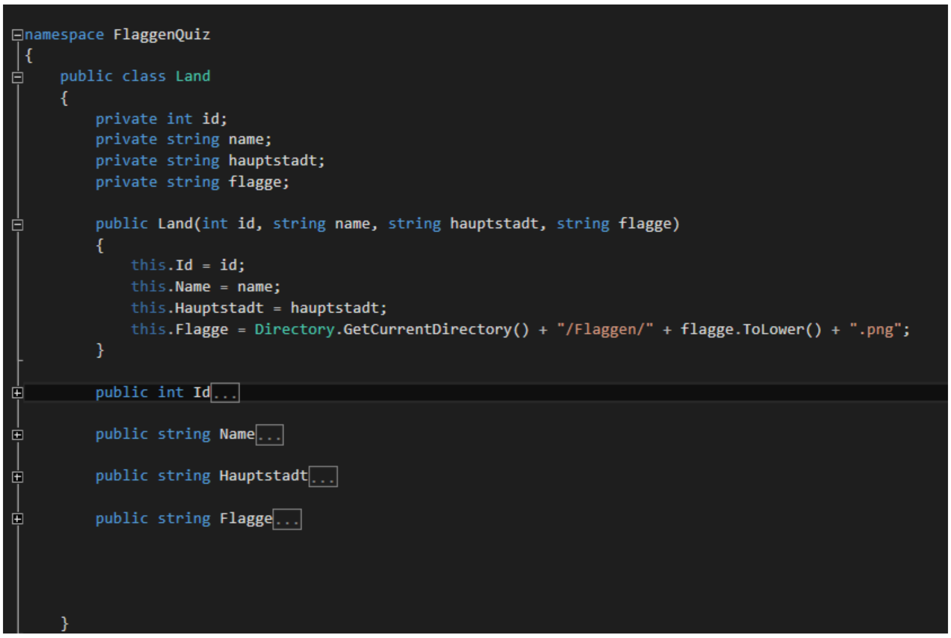


Notice that the Highscore-table uses the player as a foreign key so you can later sort the score by name.

Classes and Objects:

The classes in the program have the same attributes as the members in the database.

As an example:



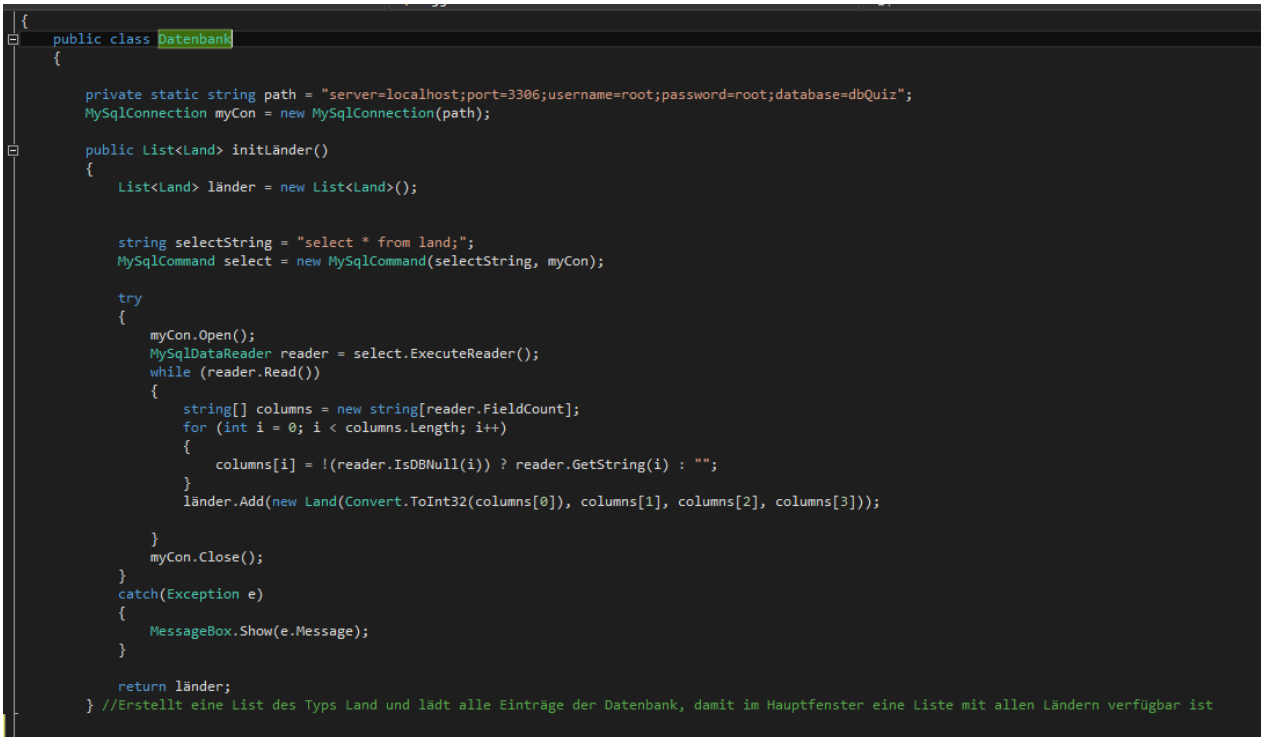
```
namespace FlaggenQuiz
{
    public class Land
    {
        private int id;
        private string name;
        private string hauptstadt;
        private string flagge;

        public Land(int id, string name, string hauptstadt, string flagge)
        {
            this.Id = id;
            this.Name = name;
            this.Hauptstadt = hauptstadt;
            this.Flagge = Directory.GetCurrentDirectory() + "/Flaggen/" + flagge.ToLower() + ".png";
        }

        public int Id...

        public string Name...

        public string Hauptstadt...

        public string Flagge...



    }
```

The pictures of the several flags will be in the project folder and are titled with the ISO standard country shortcuts. Since the database entries are titled the same it is very simple to import the flag later to later to instantiate them as objects.

For this program I worked with **Windows Presentation Forms** and here it is much easier to work with fully qualified path names, so I decided to make use of the GetCurrentDirectory – function of the IO-Namespace of C#.

A database-class in the program imports and exports the data.

Example:

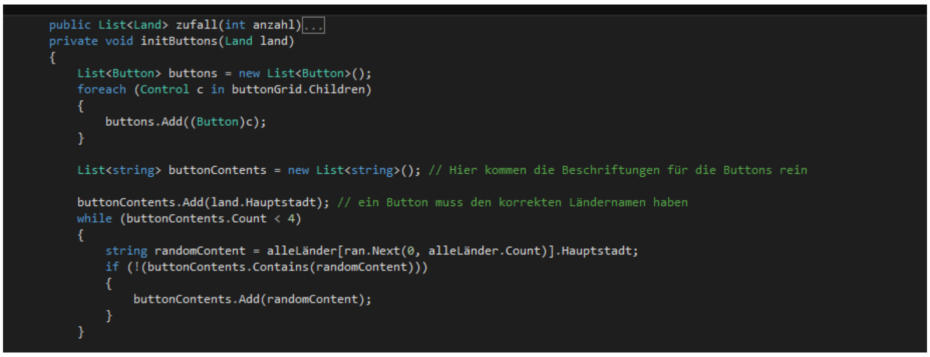


```
{
    public class Datenbank
    {

        private static string path = "server=localhost;port=3306;username=root;password=root;database=dbQuiz";
        MySqlConnection myCon = new MySqlConnection(path);

        public List<Land> initLänder()
        {
            List<Land> länder = new List<Land>();


            string selectString = "select * from land;";
            MySqlCommand select = new MySqlCommand(selectString, myCon);

            try
            {
                myCon.Open();
                MySqlDataReader reader = select.ExecuteReader();
                while (reader.Read())
                {
                    string[] columns = new string[reader.FieldCount];
                    for (int i = 0; i < columns.Length; i++)
                    {
                        columns[i] = !(reader.IsDBNull(i)) ? reader.GetString(i) : "";
                    }
                    länder.Add(new Land(Convert.ToInt32(columns[0]), columns[1], columns[2], columns[3]));

                }
                myCon.Close();
            }
            catch(Exception e)
            {
                MessageBox.Show(e.Message);
            }

            return länder;
        } //Erstellt eine List des Typs Land und lädt alle Einträge der Datenbank, damit im Hauptfenster eine Liste mit allen Ländern verfügbar ist
```

This function is used to import a list of all countries from the database to the program to make them accessible during the program procedure.

Buttons:

During the game the buttons contents and the answers are generated randomly (well, the right answer should be included)



```
        public List<Land> zufall(int anzahl)...
        private void initButtons(Land land)
        {
            List<Button> buttons = new List<Button>();
            foreach (Control c in buttonGrid.Children)
            {
                buttons.Add((Button)c);
            }

            List<string> buttonContents = new List<string>(); // Hier kommen die Beschriftungen für die Buttons rein

            buttonContents.Add(land.Hauptstadt); // ein Button muss den korrekten Ländernamen haben
            while (buttonContents.Count < 4)
            {
                string randomContent = alleLänder[ran.Next(0, alleLänder.Count)].Hauptstadt;
                if (!(buttonContents.Contains(randomContent)))
                {
                    buttonContents.Add(randomContent);
                }
            }
```