НП „Обучение за
ИТ кариера“
Име на проекта:
Farm Management
Екип: Деница Радичева, Дияна Маринова, Екатерина Радева
Група 08
Хасково 24.03.2024г.

1.Цели

Разработване на софтуер, който изгражда база данни за мониторинг на наличността на фермерски животни и продукти. Целта на софтуера е да улесни проследяването на измененията в наличността за потребителя и да предостави информацията по удобен и лесен за разбиране начин.

2.Разпределение на ролите

1. Дияна:
   - Проектиране и разработка на графичен потребителски интерфейс
   - Имплементиране на база данни
2. Деница:
   - Създаване на бази данни
   - Изготвяне на заявки
3. Екатерина:
   - Създаване на NUnit тестове за сверяване на получените данните

3.Основни етапи в реализирането на проекта

- Разработка на потребителски интерфейс
- Създаване и имплементиране на база данни
- Осъществяване на връзка между базата данни и интерфейса на програмата
- Разработка на тестове и имплементирането им
- Тестване и дебъгване на програмата

4.Реализация

Логически алгоритми:

1. Метод GetAnimals:

- Този метод извлича броя на различните видове животни от таблицата Animals в базата данни.
- Той отваря връзка с базата данни, изпълнява SQL заявка, за да групира животните по вид и да ги преброи, записва резултатите в речник.
- Накрая затваря връзката с базата данни и връща речника, съдържащ броя на видовете животни.

```
public static Dictionary<string, int> GetAnimals()
{
    SqlConnection connection;
    connection = new SqlConnection(connectionString);
    connection.Open();

    string query = "SELECT animal_type, COUNT(animal_type) FROM Animals GROUP BY animal_type;";
    SqlCommand command = new SqlCommand(query, connection);
    SqlDataReader reader = command.ExecuteReader();

    Dictionary<string, int> counts = new Dictionary<string, int>();
    while (reader.Read())
    {
        string animalType = reader.GetString(0);
        int count = reader.GetInt32(1);
        counts.Add(animalType, count);
    }

    connection.Close();
    return counts;
}
```

2. Метод AddAnimalToBaseTable:
   - Този метод добавя ново животно в таблицата Animals в базата данни.
   - Той отваря връзка с базата данни, изпълнява SQL заявка за добавяне на нов вид животно, извлича генерираното ID на добавения ред, присвоява го на свойството Id и след това затваря връзката.

```
protected void AddAnimalToBaseTable()
{
    SqlConnection connection;
    connection = new SqlConnection(connectionString);
    connection.Open();

    string query = "INSERT INTO Animals (animal_type) VALUES (@animalType);" +
        " SELECT CAST(scope_identity() AS int);";
    SqlCommand command = new SqlCommand(query, connection);
    command.Parameters.AddWithValue("@animalType", this.Type);

    int modified = (int)command.ExecuteScalar();
    Id = modified;

    connection.Close();
}
```

3. Метод DeleteAnimalFromBaseTable:
   - Този метод изтрива животно от таблицата Animals в базата данни.
   - Той отваря връзка с базата данни, изпълнява SQL заявка за изтриване на животното със зададения вид (най-скоро добавеното), и след това затваря връзката.

```
protected void DeleteAnimalFromBaseTable()
{
    SqlConnection connection;
    connection = new SqlConnection(connectionString);

    command = new SqlCommand("DELETE FROM Animals WHERE Id = (SELECT TOP (1) Id FROM Animals " +
        "WHERE animal_type = @animalType ORDER BY Id DESC);", connection);
    connection.Open();

    command.Parameters.AddWithValue("@animalType", this.Type);
    command.ExecuteNonQuery();

    connection.Close();
}
```

4. Метод GetProducts:
   - Този метод извлича броя на различните видове продукти от таблицата Products в базата данни.

- Той следва подобен процес като GetAnimals, отваря връзка с базата данни, изпълнява SQL заявка, за да групира продуктите по вид и да ги преброи, записва резултатите в речник и затваря връзката с базата данни.

```
public static Dictionary<string, int> GetProducts()
{
    SqlConnection connection;
    connection = new SqlConnection(connectionString);
    connection.Open();

    string query = "SELECT product, COUNT(product) FROM Products GROUP BY product;";

    SqlCommand command = new SqlCommand(query, connection);
    SqlDataReader reader = command.ExecuteReader();

    Dictionary<string, int> counts = new Dictionary<string, int>();
    while (reader.Read())
    {
        string product = reader.GetString(0);
        int quantity = reader.GetInt32(1);
        counts.Add(product, quantity);
    }
```

5. Метод AddProductToBaseTable:
   - Този метод добавя нов продукт в таблицата Products в базата данни.
   - Той следва подобен процес като AddAnimalToBaseTable, отваря връзка с базата данни, изпълнява SQL заявка за добавяне на нов продукт, извлича генерираното ID на добавения ред, присвоява го на свойството Id и след това затваря връзката.

```
protected void AddProductToBaseTable()
{
    SqlConnection connection;
    connection = new SqlConnection(connectionString);
    connection.Open();

    string query = "INSERT INTO Products (product) VALUES (@product); " +
        "SELECT CAST(scope_identity() AS int);";
    SqlCommand command = new SqlCommand(query, connection);
    command.Parameters.AddWithValue("@product", this.Product);

    int modified = (int)command.ExecuteScalar();
    Id = modified;

    connection.Close();
}
```

6. Метод DeleteProductFromBaseTable:

- Този метод изтрива продукт от таблицата Products в базата данни.
- Той следва подобен процес като DeleteAnimalFromBaseTable, отваря връзка с базата данни, изпълнява SQL заявка за изтриване на продукта със зададения вид (най-скоро добавеният), и след това затваря връзката.

```
protected void DeleteProductFromBaseTable()
{
    SqlConnection connection;
    connection = new SqlConnection(connectionString);

    command = new SqlCommand("DELETE FROM Products WHERE No = (SELECT TOP (1) No FROM " +
        "Products WHERE product = @product ORDER BY No DESC);", connection);
    connection.Open();

    command.Parameters.AddWithValue("@product", this.Product);
    command.ExecuteNonQuery();

    connection.Close();
}
```

Използвани технологии:

- Visual studio2022
- SQL Server Management Studio 19
- Microsoft.NET.Test.Sdk
- NUnit
- NUnit.Analyzers
- NUnit3TestAdapter
- System.Data.SqlClient

5.Развитие и нововъведения

Планирана е разработката на по-голям каталог за избор на животни и продукти както и добавяне на допълнителни функции полезни за потребителя.

6.Заключение

В заключение, проектът за разработване на софтуер за управление на ферма има за цел да улесни работата на фермерите и да оптимизира производствения процес. Чрез използването на съвременни технологии като Visual Studio 2022 и SQL Server, създадохме функционален софтуер, който да обработва данни за наличността на животни и продукти на фермата.

Логическите алгоритми, реализирани в програмата, са основа за ефективното и точно управление на данните за животните и продуктите. Те гарантират правилната обработка и визуализация на информацията за наличността и тяхното управление.

В бъдеще планирано е разширяване на функционалността на софтуера, като се добавят нови възможности и подобрения, които да увеличат ефективността на фермерските операции и да улеснят работата на потребителите.

7.Използвана литература

- https://stackoverflow.com/
- https://github.com/
- https://www.codecademy.com/
- https://www.pluralsight.com/
- https://docs.microsoft.com/en-us/dotnet/csharp/
- https://docs.microsoft.com/en-us/sql/sql-server/
- https://docs.nunit.org/