Код Unit Tests:

using Microsoft.VisualStudio.TestTools.UnitTesting;

using DatabaseController;

using System;

namespace DatabaseControllerTests

{

[TestClass]

public class TestDatabase

{

private string sqlConnectionString = "Server=MANGUST17;Database=praktika;Trusted\_Connection=True;";

private Database db;

[TestInitialize]

public void Setup()

{

db = Database.GetInstance(sqlConnectionString);

}

[TestMethod]

public void TestExecute()

{

string query = "INSERT INTO Test (login, password) VALUES (@login, @password)";

Parameter[] parameters = {

new Parameter("login", "Matvey"),

new Parameter("password", "Leyko")

};

int rows = db.Execute(query, parameters);

Assert.AreEqual(1, rows, "Должна быть затронута 1 строка.");

}

[TestMethod]

public void TestGetScalar()

{

string query = "SELECT COUNT(\*) FROM Test";

object result = db.GetScalar(query);

Assert.IsNotNull(result, "Результат не должен быть null");

}

[TestMethod]

public void TestGetRowsData()

{

string query = "SELECT id, login FROM Test";

object[][] result = db.GetRowsData(query);

Assert.IsTrue(result.Length > 0, "Должны быть возвращены строки из таблицы.");

}

[TestMethod]

public void TestUpdateRecord()

{

string query = "UPDATE Test SET login = @newLogin WHERE login = @oldLogin";

Parameter[] parameters = {

new Parameter("newLogin", "UpdatedMatvey"),

new Parameter("oldLogin", "Matvey")

};

int rows = db.Execute(query, parameters);

Assert.AreEqual(1, rows, "Должна быть затронута 1 строка.");

}

[TestMethod]

public void TestDeleteRecord()

{

string insertQuery = "INSERT INTO Test (login, password) VALUES (@login, @password)";

Parameter[] parameters = {

new Parameter("login", "ToDelete"),

new Parameter("password", "password")

};

db.Execute(insertQuery, parameters);

string deleteQuery = "DELETE FROM Test WHERE login = @login";

Parameter[] parameterss = { new Parameter("login", "ToDelete") };

int rows = db.Execute(deleteQuery, parameterss);

Assert.AreEqual(1, rows, "Должна быть удалена 1 строка.");

}

}

}

Аннотация теста:

|  |  |
| --- | --- |
| **Название проекта** | DatabaseControllerTests |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Матвей Л.А |
| **Дата(ы) теста** | 06.11.2024 |

Расшифровка тестовых информационных полей:

**Тестовый пример #1: Вставка записи в базу данных**

|  |  |
| --- | --- |
| Поле | Описание |
| Тестовый пример #1 | TC\_DB\_1 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Проверка создания новой заявки |
| Краткое изложение теста | Тест проверяет успешное создание новой заявки в таблицу |
| Этапы теста | 1. Заполнить поля id заявки, описание проблемы, тип и модель техники.  2.Нажать на кнопку добавить заявку. |
| Тестовые данные | id заявки = 3, описание проблемы=не жарит, тип техники=тостер, модель техники= тост |
| Ожидаемый результат | Вывелся messageBox с сообщении об успехе добавления, вставлена одна запись в таблицу |
| Фактический | Соответствует ожидаемому |
| Статус | Успех |
| Предварительное условие | Пользователь авторизовался в системе под ролью заказчика |
| Постусловие | Запись была добавлена в таблицу |
| Примечания/комментарии | - |

**Тестовый пример #2: Получение строк данных из таблицы Test**

|  |  |
| --- | --- |
| Поле | Описание |
| Тестовый пример #1 | TC\_DB\_1\_Negative |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Проверка отклонения создания новой заявки с некорректными данными |
| Краткое изложение теста | Тест проверяет, что заявка не будет создана, если введены некорректные данные, например, пустые поля. |
| Этапы теста | 1. Оставить поля ID заявки, описание проблемы, тип и модель техники пустыми. 2. Нажать на кнопку "Добавить заявку". |
| Тестовые данные | id заявки = (пусто), описание проблемы = (пусто), тип техники = (пусто), модель техники = (пусто) |
| Ожидаемый результат | Выводится messageBox с сообщением об ошибке (например, "Заполните все обязательные поля"). Данные не добавляются в таблицу. |
| Фактический | Соответствует ожидаемому |
| Статус | Успех |
| Предварительное условие | Пользователь авторизован в системе под ролью заказчика |
| Постусловие | Запись не была добавлена в таблицу |
| Примечания/комментарии | - |

**Тестовый пример #3: Проверка выполнения фильтрации записей**

|  |  |
| --- | --- |
| Поле | Описание |
| Тестовый пример #1 | TC\_DB\_3 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Выполнение фильтрации данных по нажатию кнопки пользователем. |
| Краткое изложение теста | Тест проверяет корректность выполнения фильтрации данных. |
| Этапы теста | 1. Нажать на кнопку для фильтрации по не зарегистрированным заявкам. |
| Тестовые данные |  |
| Ожидаемый результат | В таблице остались, лишь заявки со статусом = “Не зарегистрирована” |
| Фактический | Соответствует ожидаемому |
| Предварительное условие | Пользователь должен быть авторизован под ролью оператора |
| Постусловие |  |
| Примечания/комментарии | - |

**Тестовый пример #4: Обновление данных**

|  |  |
| --- | --- |
| Поле | Описание |
| Тестовый пример #1 | TC\_DB\_4 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Проверка обновления данных пользователем |
| Краткое изложение теста | Тест проверяет успешное обновление данных на основе введённых пользователем id заявки и ее статусом. |
| Этапы теста | 1. Ввести id заявки и выбрать ее новый статус. |
| Тестовые данные | id заявки=1,статус заявки = “Назначен мастер” |
| Ожидаемый результат | Строка с вписанным пользователем id изменила статус. |
| Фактический | Соответствует ожидаемому |
| Предварительное условие | Пользователь должен быть авторизован под ролью оператора, зааявка с заданным ID должна существовать в таблице |
| Постусловие | Обновленный статус заявки должен быть сохранен в таблице. |
| Примечания/комментарии | - |

**Тестовый пример #5: Удаление записи**

|  |  |
| --- | --- |
| Поле | Описание |
| Тестовый пример #1 | TC\_DB\_5 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Проверка удаления записи из таблицы |
| Краткое изложение теста | Тест проверяет успешное удаление записи id которой написал пользователь |
| Этапы теста | 1. Ввести id заявки и нажать на кнопку удалить запись. |
| Тестовые данные | id заявки = 2 |
| Ожидаемый результат | Заявка с id = 2 была удалена из таблицы |
| Фактический | Соответствует ожидаемому |
| Предварительное условие | Пользователь должен быть авторизован, как оператор. Заявка с заданным ID должна существовать в таблице. |
| Постусловие | Заявка с указанным ID больше не отображается в таблице. |
| Примечания/комментарии | - |