**МИНОБРНАУКИ РОССИИ**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

Институт среднего специального образования

УТВЕРЖДЕНО   
предметной комиссией  
 «Программное обеспечение»  
   
Председатель ПЦК   
\_\_\_\_\_\_\_\_\_\_\_\_

**ЗАДАНИЕ на курсовое проектирование**

по модулю ПМ.03 «Участие в интеграции программных модулей»

Специальность 09.02.03 «Программирование в компьютерных системах»

Студенту группы з42928/1

Тема курсового проекта: «Информационная система “Информационный бизнес в Ставрополе”»

**Индивидуальное задание**

1. *Построить Use Case Diagram*
2. *Построить диаграммы этапа проектирования*
3. *Спроектировать схему данных*
4. *Реализовать базу данных SQL Server в Visual Studio*
5. *Создать проект в VS: веб-приложение ASP.NET MVC*
6. *Определить примерный интерфейс приложения*
7. *Реализовать процесс авторизации и регистрации*
8. *Разработать тест-кейсы*

Дата выдачи задания 05.10.2020 Зав. отделением \_\_\_\_\_\_\_\_\_Симонова О.М.

Срок окончания 23.10.2020 Преподаватель \_\_\_\_\_\_\_\_\_\_ Зернова Е. Н.

Студент \_\_\_\_\_\_\_\_\_\_\_

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**автономное образовательное учреждение высшего образования**

**«Санкт-Петербургский государственный политехнический университет Петра Великого»**

**(ФГАОУ ВО «СПбПУ»)**

**Институт среднего профессионального образования**

**КУРСОВАЯ РАБОТА**

Программирование в компьютерных системах

специальность

отделение Заочное

По дисциплине: Технология разработки программного обеспечения

Тема Информационная система “Информационный бизнес в Ставрополе”\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Исполнитель: студент(ка) \_\_\_\_IV\_\_\_ курса группы \_\_з42928/1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Матвеенко Дмитрий Владимирович

(фамилия, имя, отчество)

Руководитель \_\_\_преподаватель ИСПО\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ученная степень, звание)

\_\_\_\_\_\_\_Зернова Елена Николаевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

Курсовая работа допущена к защите

Защитил(а) курсовую работу с оценкой

«23» октября 2020

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2020

«23» октября 2020

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc54970827)

[1 Постановка задачи 4](#_Toc54970828)

[1.1 Анкетирование 4](#_Toc54970829)

[1.2 Спецификация требований 5](#_Toc54970830)

[1.2.1 Требования к функциональным характеристикам 5](#_Toc54970831)

[1.2.2 Требования к надёжности 5](#_Toc54970832)

[1.2.3 Требования к составу и параметрам технических средств 5](#_Toc54970833)

[1.2.4 Требования к программной совместимости 6](#_Toc54970834)

[1.2.5 Требования к маркировке и упаковке 6](#_Toc54970835)

[1.2.6 Требования к транспортировке и хранению 6](#_Toc54970836)

[1.2.7 Специальные требования 6](#_Toc54970837)

[1.3 Опорные точки 6](#_Toc54970838)

[2 Проектирование 9](#_Toc54970839)

[2.1 Диаграммы 9](#_Toc54970840)

[2.2 Построение схемы базы данных 10](#_Toc54970841)

[3 Разработка 12](#_Toc54970842)

[3.1 Модель данных 12](#_Toc54970843)

[3.2 Описание программы 12](#_Toc54970844)

[3.2.1 Вход 12](#_Toc54970845)

[3.2.2 Список организаций 13](#_Toc54970846)

[3.2.3 Добавление услуг для организации 14](#_Toc54970847)

[3.2.4 Редактирование услуги для организации 15](#_Toc54970848)

[3.2.6 Удаление услуги организации 16](#_Toc54970849)

[4 Тестирование 17](#_Toc54970850)

[4.1 Unit test 17](#_Toc54970851)

[4.2 Тест-кейсы 17](#_Toc54970852)

[ЗАКЛЮЧЕНИЕ 19](#_Toc54970853)

[Приложение А 20](#_Toc54970854)

ВВЕДЕНИЕ

Актуальность темы исследования заключается упрощении работы производства, а также последующей его расширяемости.

Для написания пользовательского приложения в данной работе был использован высокоуровневый язык программирования C# совместно с технологией ASP .NET Core. Исходя из цели курсовой работы, были поставлены следующие задачи:

* получить представление о главных принципах объектно-ориентированного программирования и изучить основы языка C#;
* разработать пользовательский интерфейс, а также описать структуру и принцип работы разрабатываемого приложения;
* выполнить реализацию программы на языке, а также ее тестирование.

Предметом курсовой работы является информационная система для библиотеки «Информационный бизнес в Ставрополе».

Объектом исследования является практическое применение объектно-ориентированного подхода к программированию для разработки автоматизированной информационной системы.

1 Постановка задачи

1.1 Анкетирование

Изначально необходимо произвести анкетирование и выявить заинтересованные лица.

Заинтересованное лицо – это все те, кто имеет прямое или косвенное отношение к процессу, автоматизация которого производится.

Заинтересованными лицами в этом проекте являются служащие компании:

1. администратор
2. библиотекарь
3. информационное агентство

В таблицах 1, 2 и 3 представлено анкетирование с каждым заинтересованным лицом.

Таблица 1 – Анкетирование администратора

|  |  |
| --- | --- |
| **Вопрос** | **Ответ** |
| Ваша должность? | Администратор |
| Каковы Ваши основные обязанности? | Поддержка работы базы знаний и компьютерного оборудования в компании. |
| Какие функции необходимы для комфортного использования АИС? | 1. Разграничение прав доступа 2. Предоставление информации о пользователях системы |

Таблица 2 – Анкетирование библиотекаря

|  |  |
| --- | --- |
| **Вопрос** | **Ответ** |
| Ваша должность? | Библиотекарь |
| Каковы Ваши основные обязанности? | Мониторинг базы знаний и актуализация информации |
| Какие функции необходимы для комфортного использования АИС? | Возможность добавлять, обновлять, удалять данные информационных агентств |

Таблица 3 – Анкетирование информационного агентства

|  |  |
| --- | --- |
| **Вопрос** | **Ответ** |
| Ваша должность? | Системный администратор информационного агентства |
| Каковы Ваши основные обязанности? | Актуализация информации об услугах, предоставляемых организацией |
| Какие функции необходимы для комфортного использования АИС? | 1. Возможность редактировать информацию об организации 2. Добавлять, обновлять, удалять услуги, которые предоставляет организация |

1.2 Спецификация требований

1.2.1 Требования к функциональным характеристикам

1. Разделение прав доступа
2. Система должна поддерживать возможность заполнения следующих данных об организациях:

* Информация об организации
* Список услуг (продуктов)

1. Проверка корректности ввода
2. Сохранение данных в базу данных

1.2.2 Требования к надёжности

* использование лицензированного программного обеспечения;

1.2.3 Требования к составу и параметрам технических средств

Для полноценного функционирования данной системы необходимо наличие следующих требований:

* операционной системы Windows Vista, 7, 8, 8.1, 10 (32/64-bit).
* процессор Intel Celeron J1800 @ 2.41 Ггц;
* оперативная память: 1 GB ОЗУ;
* место на диске от 512 МБ.

1.2.4 Требования к программной совместимости

Для полноценного функционирования данной системы необходимо наличие операционной системы Windows Vista, 7, 8, 8.1, 10. С разрядностью 32 или 64. Язык интерфейса – русский.

1.2.5 Требования к маркировке и упаковке

Требования к маркировке не предъявляются

1.2.6 Требования к транспортировке и хранению

Хранение должно производиться на специальном сервере с бесперебойной подачей электроэнергии и доступом в сеть.

1.2.7 Специальные требования

1. Доступность программы каждому пользователю

(кроссплатформенность и совместимость с каждым устройством).

1. Четкость в использовании и понимании системы (чтобы программа была понятная каждому пользователю любого возраста и любого социального класса).
2. Каждый пользователь, обладающий стандартными знаниями работы с ПК, должен освоить работу с программой за 2 дня.

1.3 Опорные точки

Для разработки системы следует провести анализ опорных точек и на основе их определить иерархию точек зрения. Диаграмма идентификации точек зрения представлена на рисунке 1.

****

Рисунок 1 - Диаграмма идентификации точек зрения

На основе диаграммы была построена таблица сервисов, соотнесённых с точками зрения. Данные представлены на таблице 4. А также схема иерархии точек зрения.

Таблица 4 - Сервисы, соотнесенные с точками зрения

|  |  |  |
| --- | --- | --- |
| **Инф. агентство** | **Библиотекарь** | **Администратор** |
| Поиск услуг инф. агентства | Поиск услуг инф. агентства |  |
| Добавление услуг инф. агентству | Добавление услуг инф. агентству |  |
| Удаление услуг инф. агентства | Удаление услуг инф. агентства |  |
| Обновление услуг инф. агентства | Обновление услуг инф. агентства |  |
| Редактировать информацию об инф. агентстве | Редактировать информацию об инф. агентстве |  |
|  | Добавить инф. агентство |  |
|  | Просмотр инф. агентств |  |
|  |  | Просмотр списка пользователей |

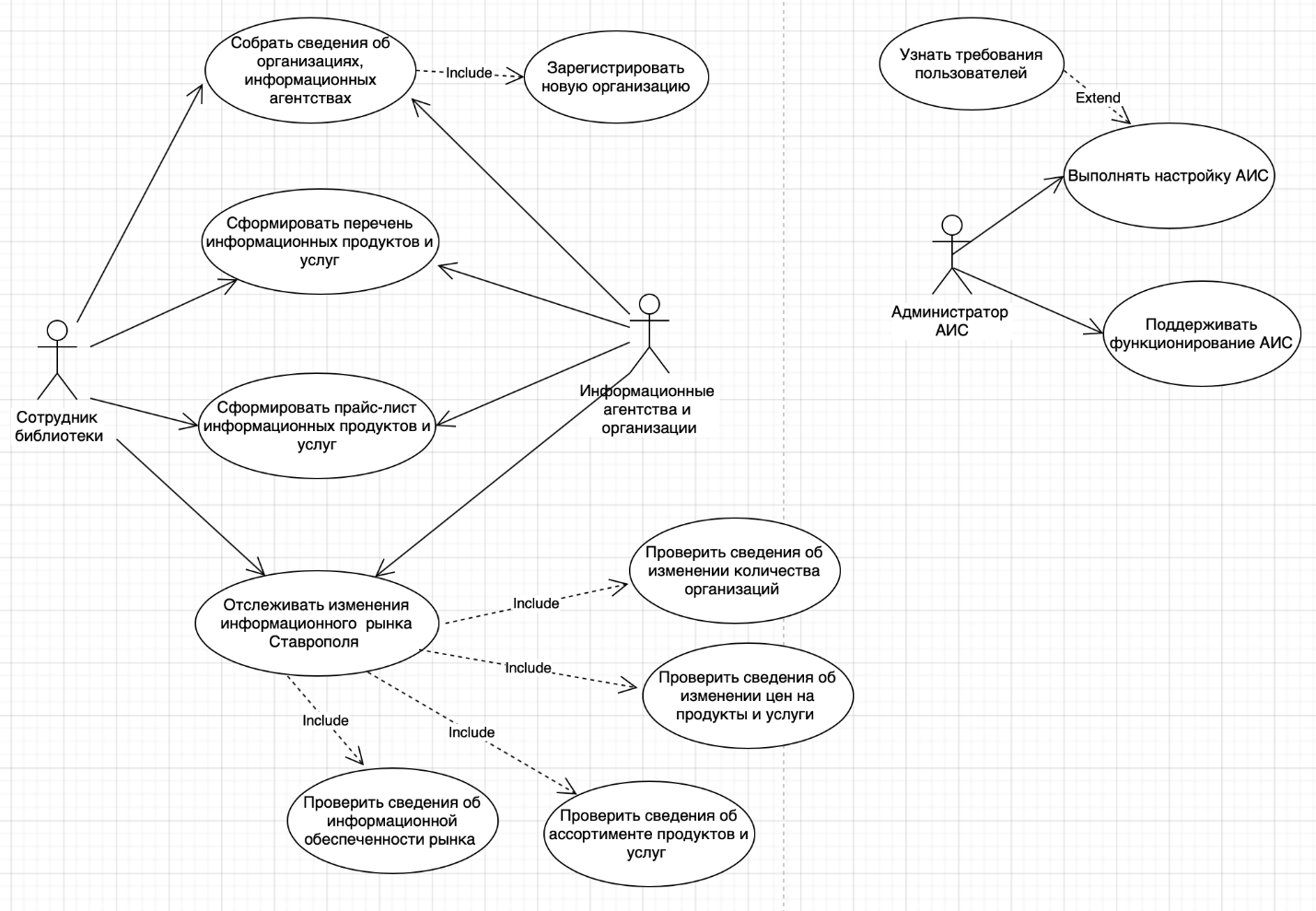


Рисунок 3 – Диаграмма вариантов использования

2 Проектирование

2.1 Диаграммы

Первым этапом проектирования информационной системы библиотеки является создание диаграмм. Были разработаны 3 вида диаграмм – Активностей, Последовательностей, Состояния. Они, соответственно, представлены на рисунках 4, 5.

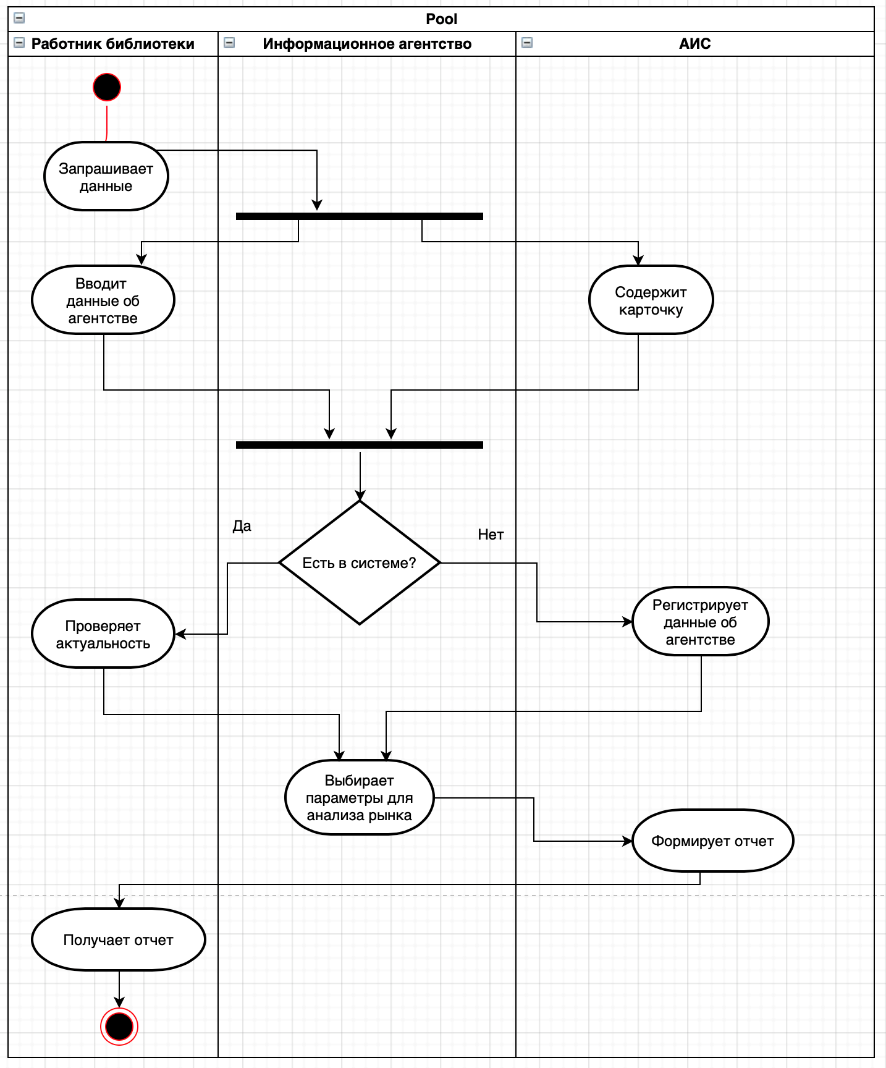


Рисунок 4 – Диаграмма активностей

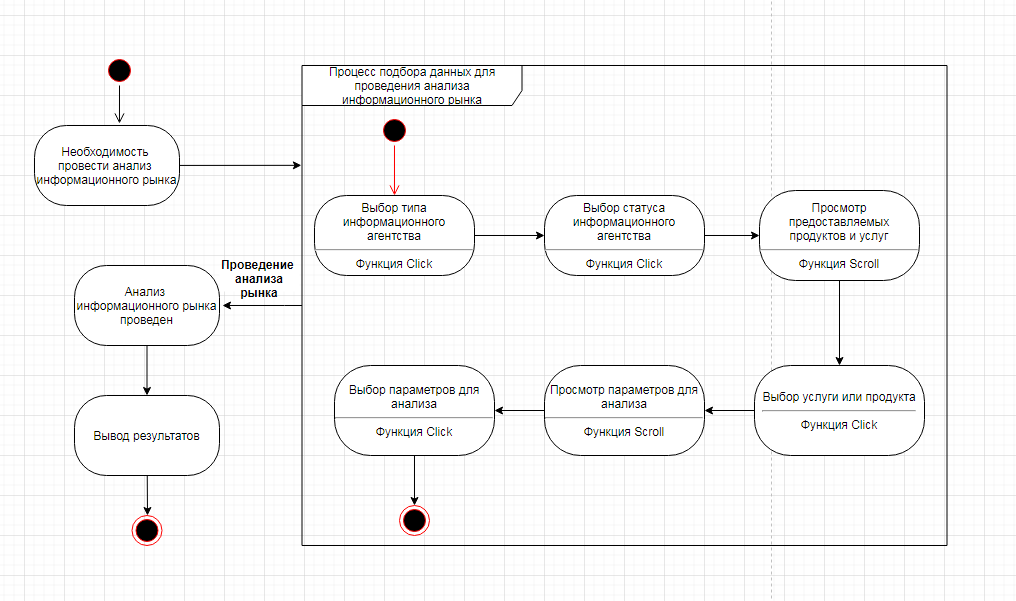


Рисунок 5 – Диаграмма последовательностей

2.2 Построение схемы базы данных

Для функционирования информационной системы и для хранения информации была разработана база данных, на рисунке 7 представлена схема базы данных с описанием ее таблиц.



Рисунок 7 – Схема базы данных

Таблица 5 – Описание к таблице «Organizations»

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Комментарий |
| Id | int | Id организации |
| InfoId | int | Id информации об организации |
| UserId | int | Id пользователя |

Таблица 6 – Описание к таблице «OrganizationInfo»

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Комментарий |
| Id | int | Id информации об организации |
| Name | text | Имя организации |
| FullAdress | text | Адрес организации |
| PhoneNumber | text | Номер телефона организации |

Таблица 7 – Описание к таблице «Products»

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Комментарий |
| Id | int | Id услуги/продукта |
| Name | text | Название |
| Price | text | Цена |
| OrganizationId | int | Id организации |

3 Разработка

3.1 Модель данных

На основе построенной схемы базы данных была создана EDM-модель данных. Модель представлена на рисунке 7. Генерация модели осуществлена при помощи Entity Framework и подхода Code First.

3.2 Описание программы

3.2.1 Вход

Страница входа в систему представлена на рисунке 9.

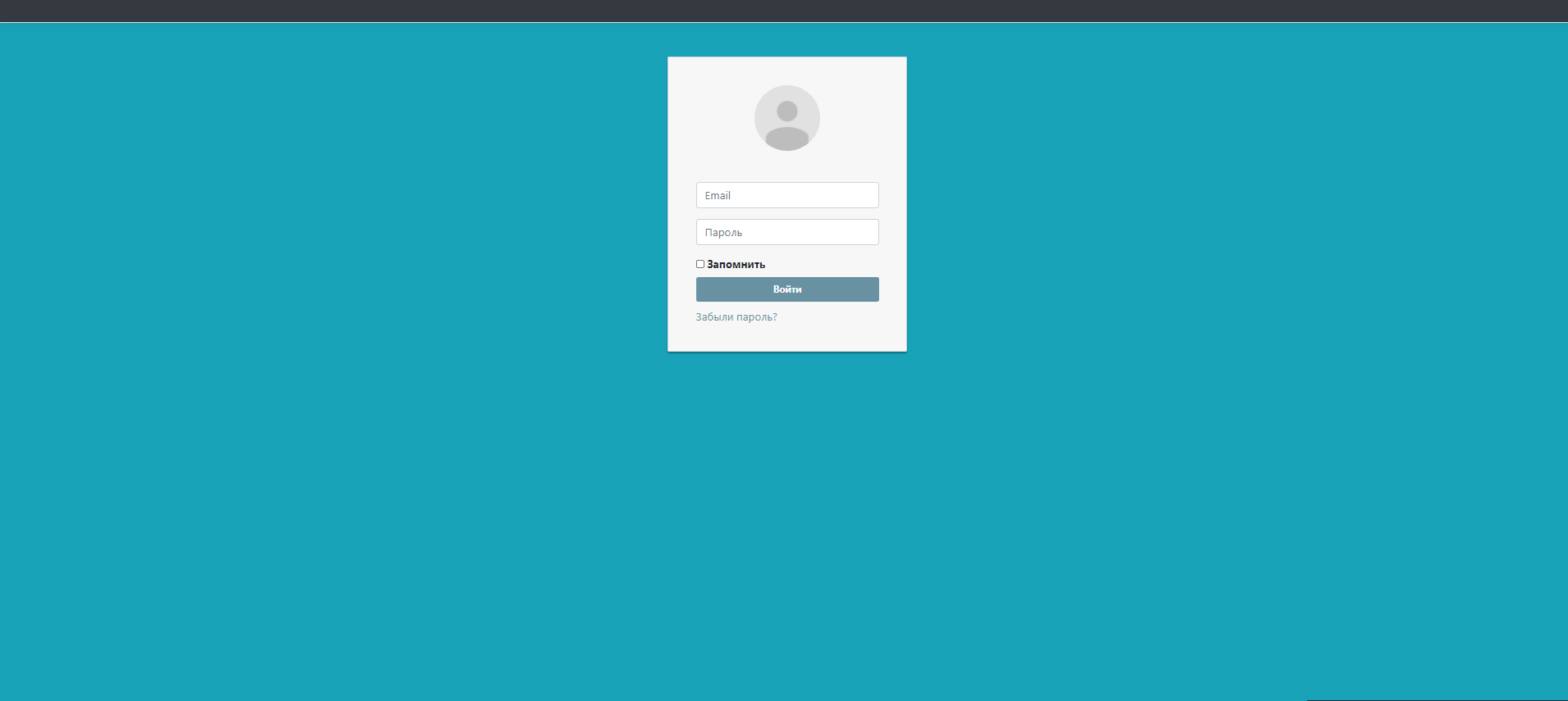


Рисунок 9 – Вход

3.2.2 Список организаций

Страница со списком организаций представлена на рисунке 10.

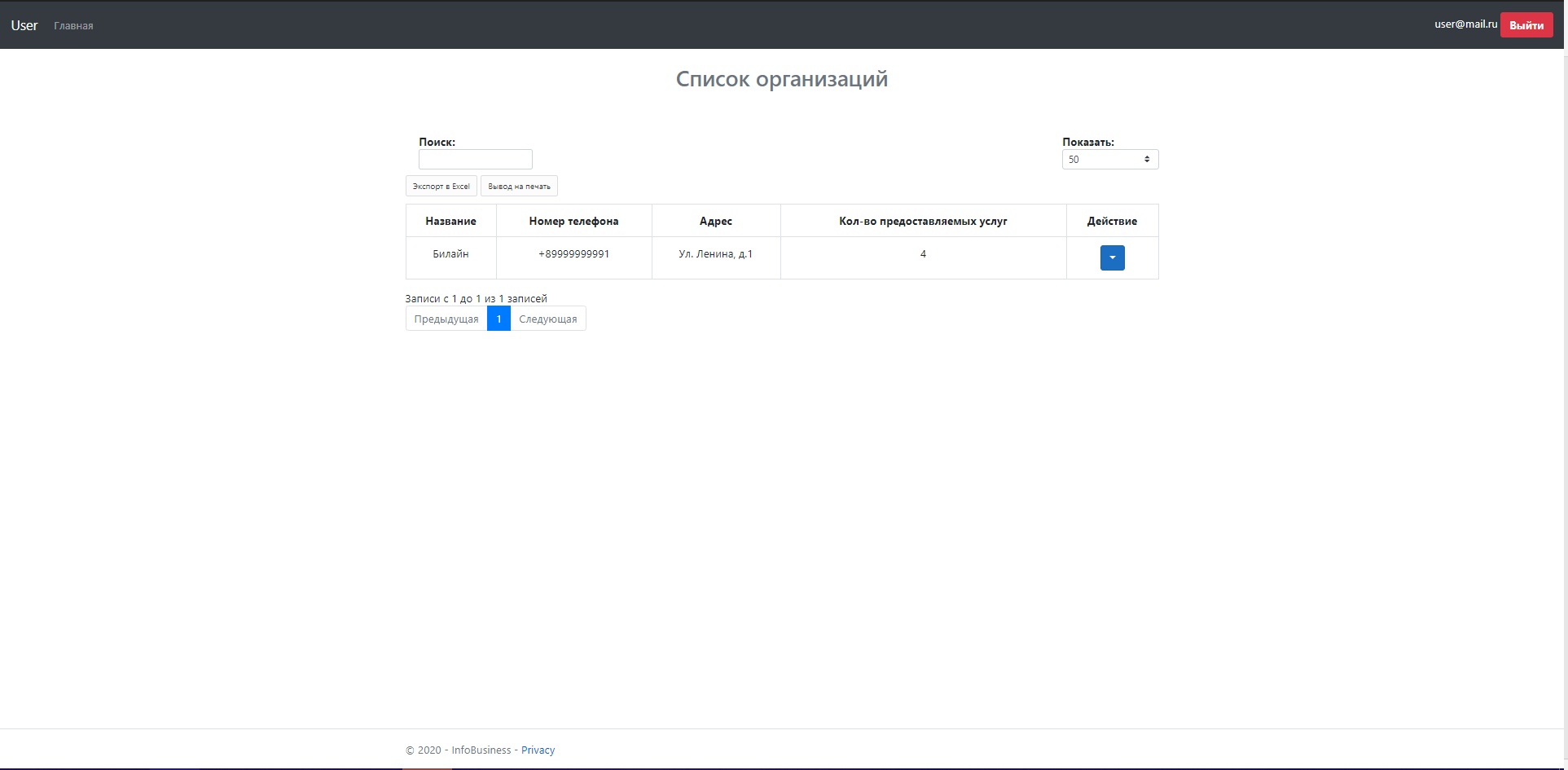


Рисунок 10 – Список организаций

Для редактирования организации необходимо напротив требуемой организации перейти на кнопку «действие» -> «Информация об организации» Окно редактирования организации представлено на рисунке 11.

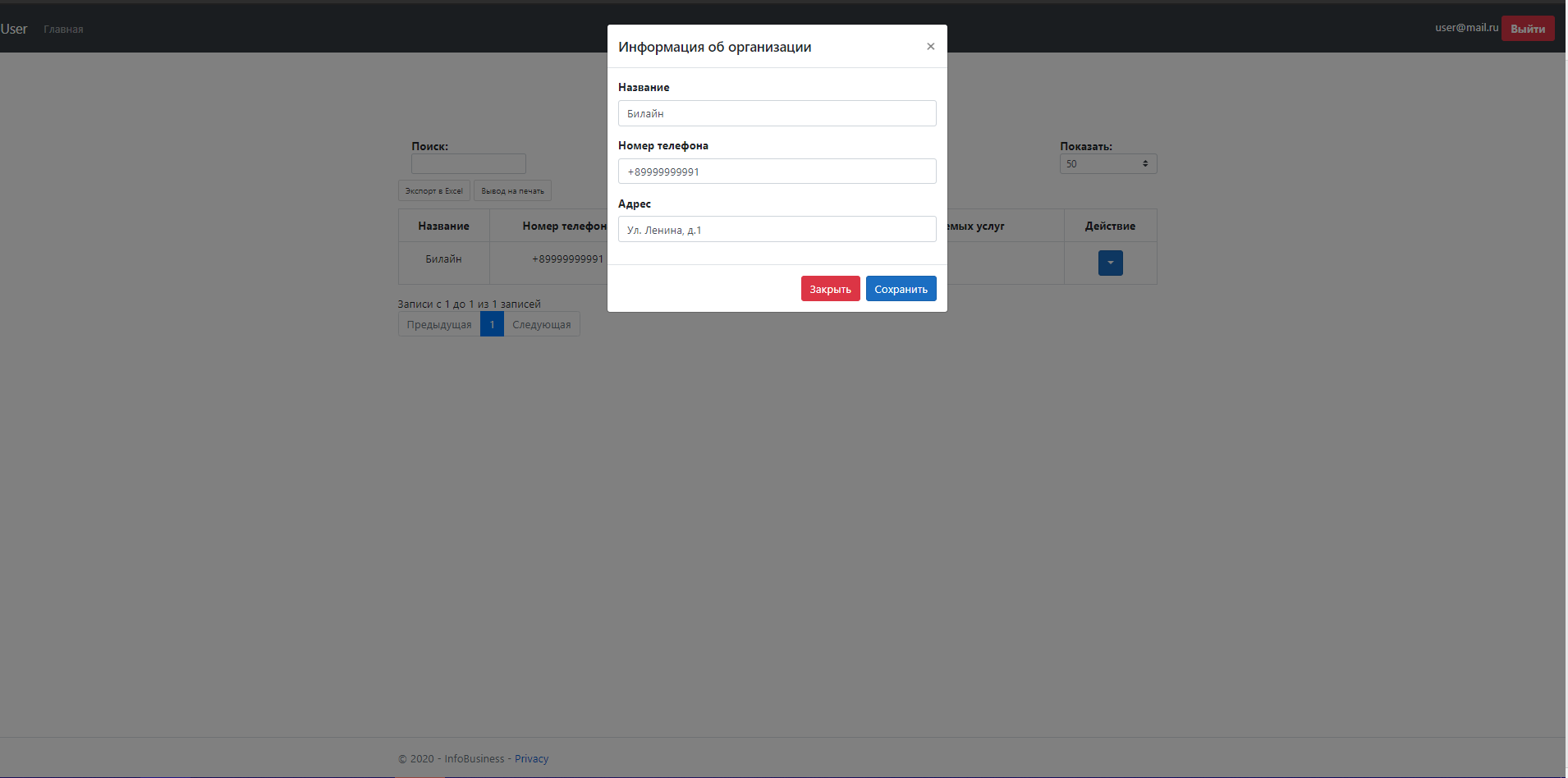


Рисунок 11 – Окно редактирования информации об организации

3.2.3 Добавление услуг для организации

Для добавления услуг необходимо выбрать организацию, куда необходимо добавить услугу и перейти по кнопке «действие» -> «Предоставляемые услуги». После чего отобразится страница со всеми услугами для выбранной организации [Рисунок 12].

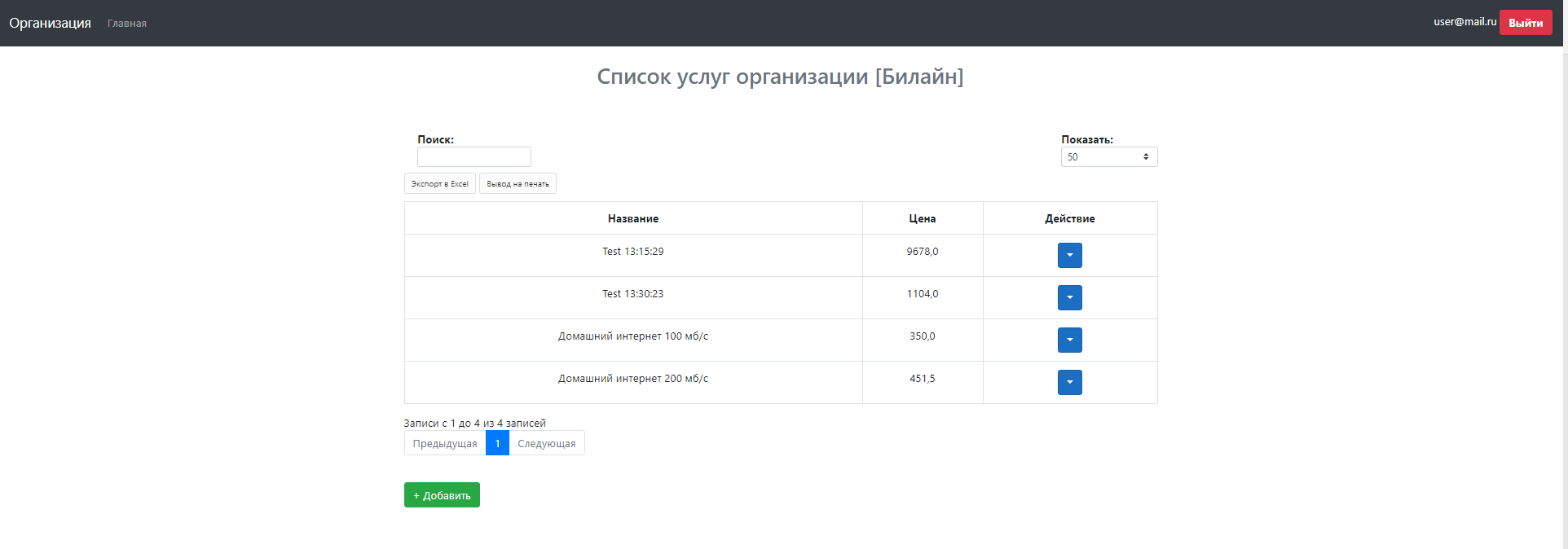


Рисунок 12 – Страница услуг организации

Далее необходимо нажать на кнопку «Добавить» и внести данные о новой услуге организации.

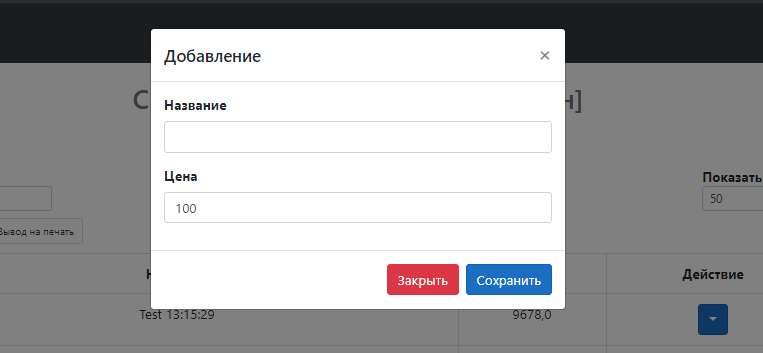


Рисунок 13 – Добавление услуги для организации

После заполнения всех данных необходимо нажать на кнопку «Сохранить», после чего услуга будет добавлена в список услуг организации.

3.2.4 Редактирование услуги для организации

Для редактирования данных напротив необходимой услуги нажали на кнопку *«действие»-> «Редактирование»*

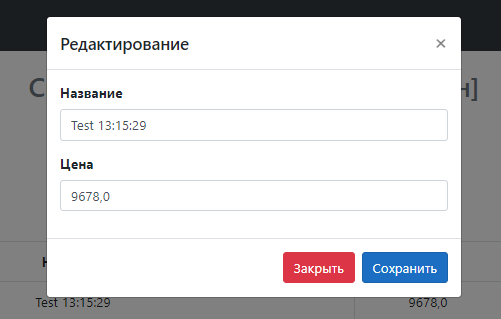


Рисунок 14 – Редактирование данных

3.2.6 Удаление услуги организации

Для удаления услуги напротив необходимой услуги нажали кнопку *«действие»-> «Удалить»*

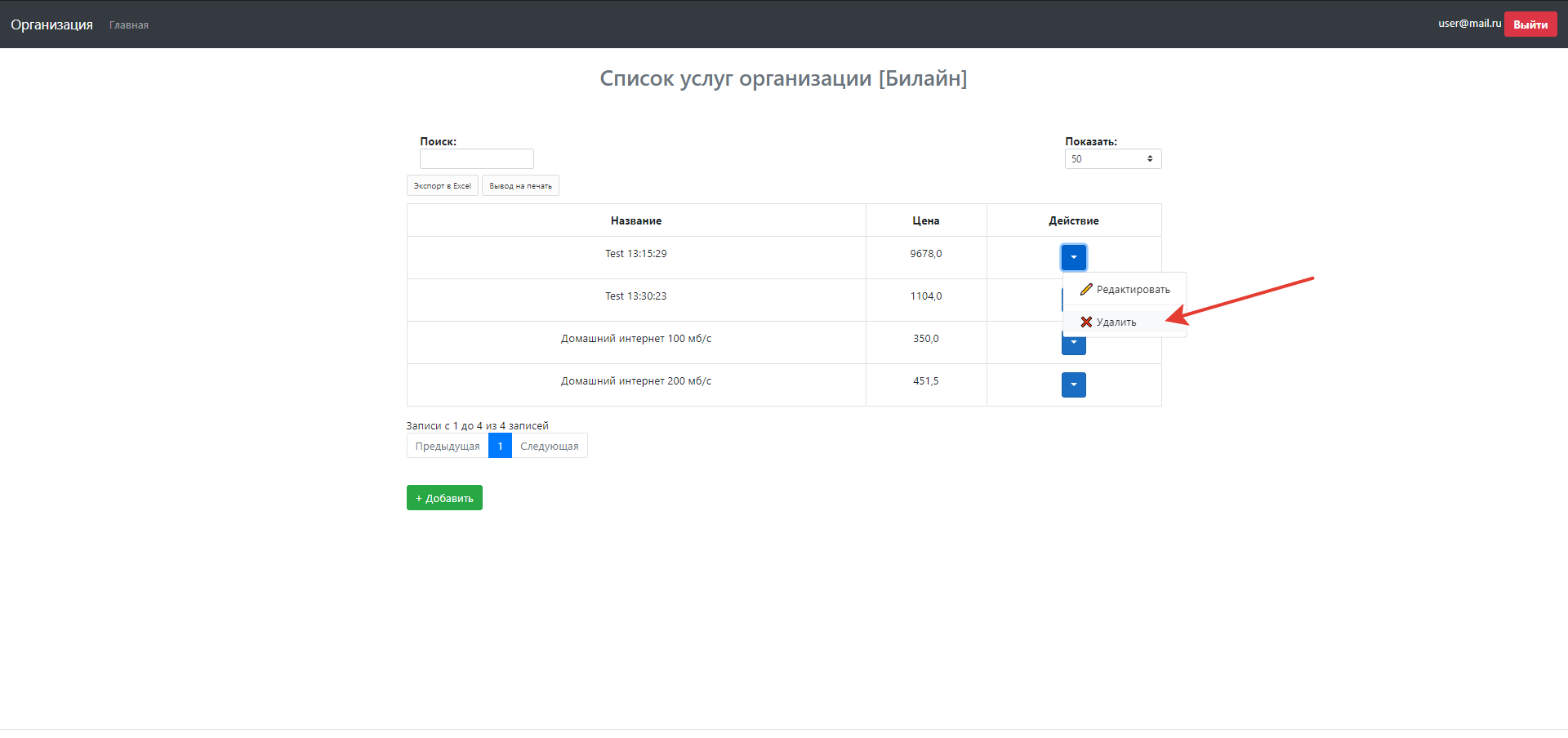


Рисунок 15 – Удаление данных

4 Тестирование

4.1 Unit test

Для отслеживания работоспособности приложения, при внесении нового функционала, необходимо каждый раз проверять работоспособность внесённых изменений. Для этого разработали Unit-тесты.

Обозреватель тестов представлен на рисунке 16. Программный код приложения Unit-тестов представлен в приложении А.

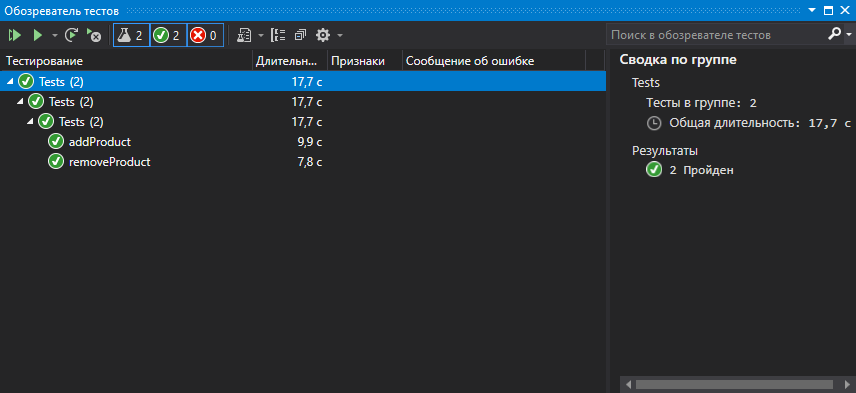


Рисунок 16 – Обозреватель тестов

4.2 Тест-кейсы

Тест-кейс — это профессиональная документация тестировщика, последовательность действий, направленная на проверку какого-либо функционала, описывающая как достичь фактического результата.  Тест-кейсы представлены в таблицах 11, 12.

Таблица 11 – Тест-кейс

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Добавление услуги/продукта для организации | | |
| **Функция:** | addProduct | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * **пройден** * **провален** * **заблокирован** |
| **Предусловие:** | |  | |
| * **Авторизоваться** * **Выбрать организацию** | | * Пользователь авторизовался * Перешли на страницу с услугами для организации |  |
| **Шаги теста:** | |  | |
| * **Нажать на кнопку «Добавить»** * Ввести название * Ввести цену | | * Появления окна с добавлением услуги |  |
| * Заполнить поля и нажать кнопку «Сохранить» | | * Закрытие окна, новая услуга добавилась в таблицу услуг |  |

Таблица 12 – Тест-кейс

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Удаление услуги/продукта для организации | | |
| **Функция:** | removeProduct | | |
| **Действие** | | **Ожидаемый результат** | **Результат:**   * **пройден** * **провален** |
| **Предусловие:** | |  | |
| * **Авторизоваться** * **Выбрать организацию** | | * Пользователь авторизовался * Перешли на страницу с услугами для организации |  |
| **Шаги теста:** | |  | |
| * Напротив удаляемой услуги нажать «действие»-> «Удалить» | | * Услуга удалилась из таблицы услуг |  |

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта была изучена технология «ASP .NET Core», работа сервера и базы данных SQL. Задачи были выполнены в полном объеме. Была спроектирована и разработана информационная система для библиотеки «Информационный бизнес в Ставрополе». Информационная система отвечает всем поставленным требованиям заказчика и предоставляет актуальную информацию необходимую для работников и клиентов библиотеки. Благодаря системе деятельность библиотеки становится более доступной и понятной для всех пользователей.

Приложение А

(справочное)

**Unit-test**

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

using OpenQA.Selenium.Support.UI;

using OpenQA.Selenium.Support.Extensions;

using System;

using System.Threading;

using System.Linq;

namespace Tests

{

public class Tests

{

IWebDriver driver;

WebDriverWait wait;

public string login = "user@mail.ru";

public string pass = "userUser1!";

[SetUp]

public void Setup()

{

var options = new ChromeOptions() { };

//options.AddArguments("--headless");

driver = new ChromeDriver();

driver.Navigate().GoToUrl("https://localhost:44304/");

wait = new WebDriverWait(driver, TimeSpan.FromSeconds(20));

}

[Test]

public void addProduct()

{

Login();

driver.Navigate().GoToUrl("https://localhost:44304/User/EditOrganizationView?id=1");

wait.Until(ExpectedConditions.ElementExists(By.ClassName("table")));

var elBeforeTest = driver.FindElements(By.CssSelector("[role='row']")).Count;

driver.ExecuteJavaScript("$(\".addProduct\")[0].click()");

wait.Until(ExpectedConditions.ElementExists(By.ClassName("modal")));

driver.ExecuteJavaScript($"$(\"[data-name = 'Name']\")[0].value = \"Test {DateTime.Now.ToString("HH:mm:ss")}\"");

driver.ExecuteJavaScript($"$(\"[data-name = 'Price']\")[0].value = \"{new Random().Next(10000)}\"");

driver.ExecuteJavaScript("$(\"[data-save = 'modal']\")[0].click()");

Thread.Sleep(2000);

wait.Until(ExpectedConditions.ElementExists(By.ClassName("table")));

var elAfterTest = driver.FindElements(By.CssSelector("[role='row']")).Count;

if (elAfterTest > elBeforeTest) //исходя и строк в таблице после теста

{

Assert.Pass();

}

else

{

Assert.Fail();

}

}

[Test]

public void removeProduct()

{

Login();

driver.Navigate().GoToUrl("https://localhost:44304/User/EditOrganizationView?id=1");

wait.Until(ExpectedConditions.ElementExists(By.ClassName("table")));

var elBeforeTest = driver.FindElements(By.CssSelector("[role='row']")).Count;

driver.FindElement(By.ClassName("dropdown-toggle")).Click();

driver.FindElements(By.CssSelector("[title='Удалить выбранную запись'")).FirstOrDefault().Click();

Thread.Sleep(2000);

wait.Until(ExpectedConditions.ElementExists(By.ClassName("table")));

var elAfterTest = driver.FindElements(By.CssSelector("[role='row']")).Count;

if (elAfterTest < elBeforeTest) //исходя и строк в таблице после теста

{

Assert.Pass();

}

else

{

Assert.Fail();

}

}

public void Login()

{

driver.Navigate().GoToUrl("https://localhost:44304/Account/Login");

driver.ExecuteJavaScript($"$(\"[type = 'email']\")[0].value = \"{login}\"");

driver.ExecuteJavaScript($"$(\"[type = 'password']\")[0].value = \"{pass}\"");

driver.ExecuteJavaScript("$(\"[type = 'submit']\")[0].click()");

wait = new WebDriverWait(driver, TimeSpan.FromSeconds(20));

}

}

}