

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт информационных технологий и анализа данных

наименование института

Допускаю к защите

Руководитель

подпись

В.А. Харахинов

И.О. Фамилия

Разработка клиента по технологии MVC

наименование темы

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту по дисциплине

Управление данными

1.008.00.00 ПЗ

обозначение документа

Выполнил студент

ИСТ6-19-2

шифр группы

подпись

Ю.В. Комогорцева

И.О. Фамилия

Нормоконтроль

подпись

В.А. Харахинов

И.О. Фамилия

Курсовой проект защищен с оценкой

Иркутск 2021 г.

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ЗАДАНИЕ
НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ**

По курсу Управление данными

Студенту Комогорцевой Ю.В.
(фамилия, инициалы)

Тема проекта Разработка клиента по технологии MVC

Исходные данные Разработать веб-приложение для салонов красоты, предназначенное для автоматизации стратегий взаимодействия с клиентами, путём сохранения информации о клиентах и истории взаимоотношений с ним.

Рекомендуемая литература

1) Сосинская С.С. «Разработка клиента по технологии MVC»: Методические указания по выполнению курсового проекта» Электронный каталог кафедры вычислительной техники.

Графическая часть на _____ листах.

Дата выдачи задания « 07 » октября 2021 г.

Задание получил _____ Ю.В. Комогорцева
подпись И.О. Фамилия

Дата представления проекта руководителю « 14 » декабря 2021 г.

Руководитель курсового проектирования _____ В.А. Харахинов
подпись И.О. Фамилия

Содержание

Введение	4
Индивидуальный вариант	5
1 Структура базы данных	6
1.1 Метод «Объект-Связь»	6
1.2 DBDesigner	6
2 Структурная схема функционирования моделей, контроллеров и представлений	7
3 Модель данных	8
4 Контроллеры	11
4.1 ClientsController	12
4.2 WorksController	16
4.3 StatisticsServicesController	20
5 Представления	22
5.1 Клиенты	22
5.1.1 Index	22
5.1.2 Edit	25
5.1.3 Details	27
5.1.4 Delete	30
5.1.5 Create	32
5.2 Работы	34
5.2.1 Index	34
5.2.2 Edit	37
5.2.3 Details	40
5.2.4 Delete	42
5.2.5 Create	44
5.3 Выходной документ со статистикой по услугам	47
5.3.1 Index	47
5.3.2 Details	48
6 Контроль	50
6.1 Services	50
6.2 Clients	51
6.3 Employee	52
6.4 Record	53
6.5 Work	53
7 Таблица тестов	55
8 Результаты тестирования	62
Заключение	75
Список использованных источников	76

Введение

Целью курсового проекта является закрепление теоретических знаний полученных в процессе изучения курса «Управление данными» и развитие практических навыков работы с базами данных в ходе решения конкретной практической задачи – разработки веб-приложения с использованием паттерна MVC (model - view - controller), а также развитие способностей к обоснованному принятию самостоятельных решений в ходе проектной деятельности и получение базового опыта разработки программного обеспечения.

В рамках курсового проекта решается задача создания веб-приложения — это информационная система для автоматизированного взаимодействия с клиентами и с возможностью хранить и редактировать данные об услугах, клиентах, сотрудниках, выполненных работах и записях. Вся информация хранится в локальной базе данных.

Индивидуальный вариант

Имеются клиенты (фамилия, имя, отчество, день рождения, почта, телефон, комментарий), сотрудники (фамилия, имя, отчество, день рождения, должность, телефон, комментарий), услуги (название, стоимость, комментарий), записи (услуга, клиент, сотрудник, дата, время, комментарий) и выполненные работы (услуга, клиент, сотрудник, дата, время, комментарий, доход). Запись необходима для хранения информации о том, на какую услугу, к какому сотруднику, во сколько, когда и какой клиент придет в салон красоты. Выполненные работы хранят информацию о том, какая услуга, кому, кем, когда, во сколько и за какую стоимость была выполнена.

Выходные документы:

- выдать список услуг с количеством выполненных работ за определенный год;
- выдать список сотрудников, содержащих в имени определенную букву, с указанием максимального и минимального возраста клиентов этих сотрудников, а также с количеством клиентов сотрудников.

1 Структура базы данных

1.1 Метод «Объект-Связь»



Рисунок 1 – ER-диаграмма

Исходя из заданного индивидуального варианта была разработана ER-диаграмма. Объекты «Работа» и «Запись» обязательно содержат такие объекты как «Услуга», «Сотрудник» и «Клиент».

1.2 DBDesigner

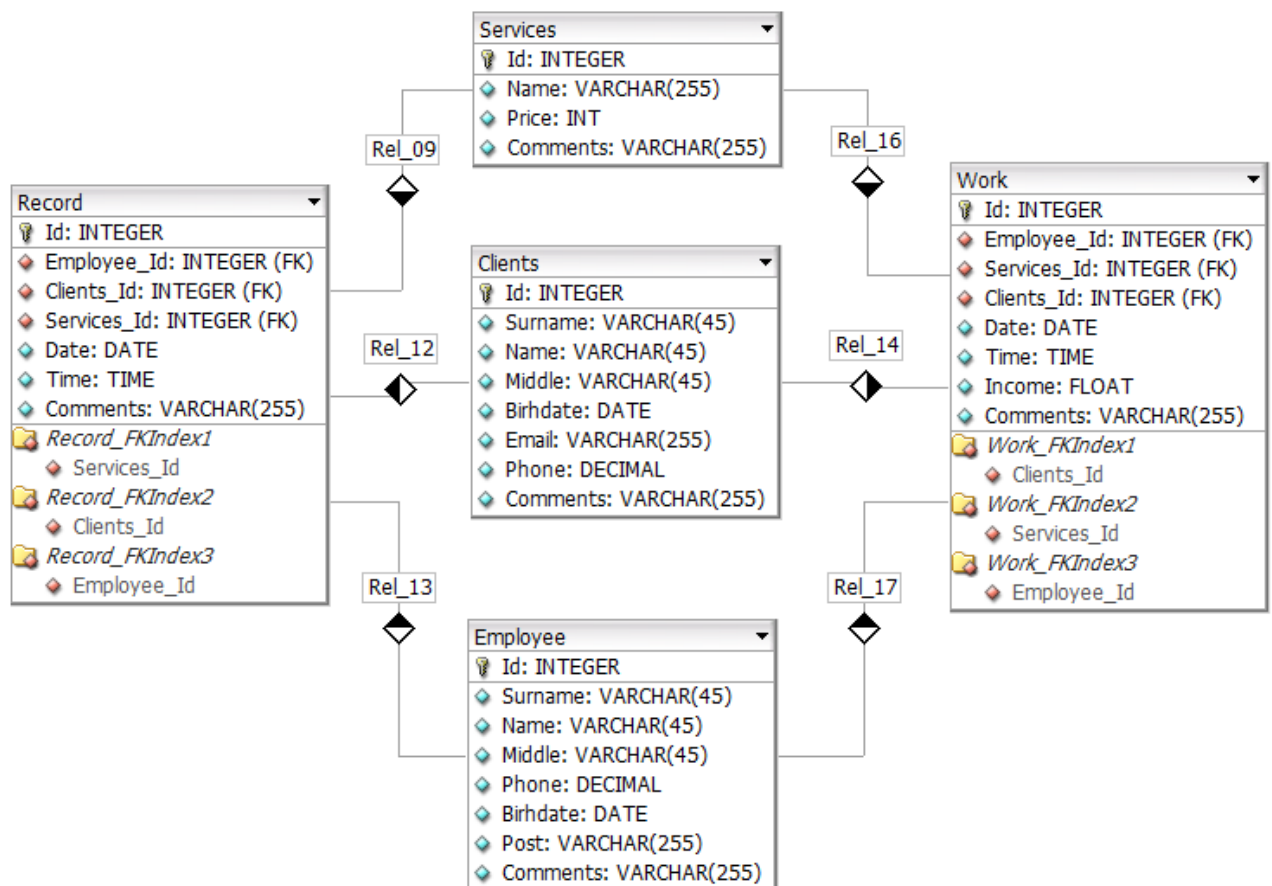


Рисунок 2 – Логическая модель БД «BeautyShop»

2 Структурная схема функционирования моделей, контроллеров и представлений

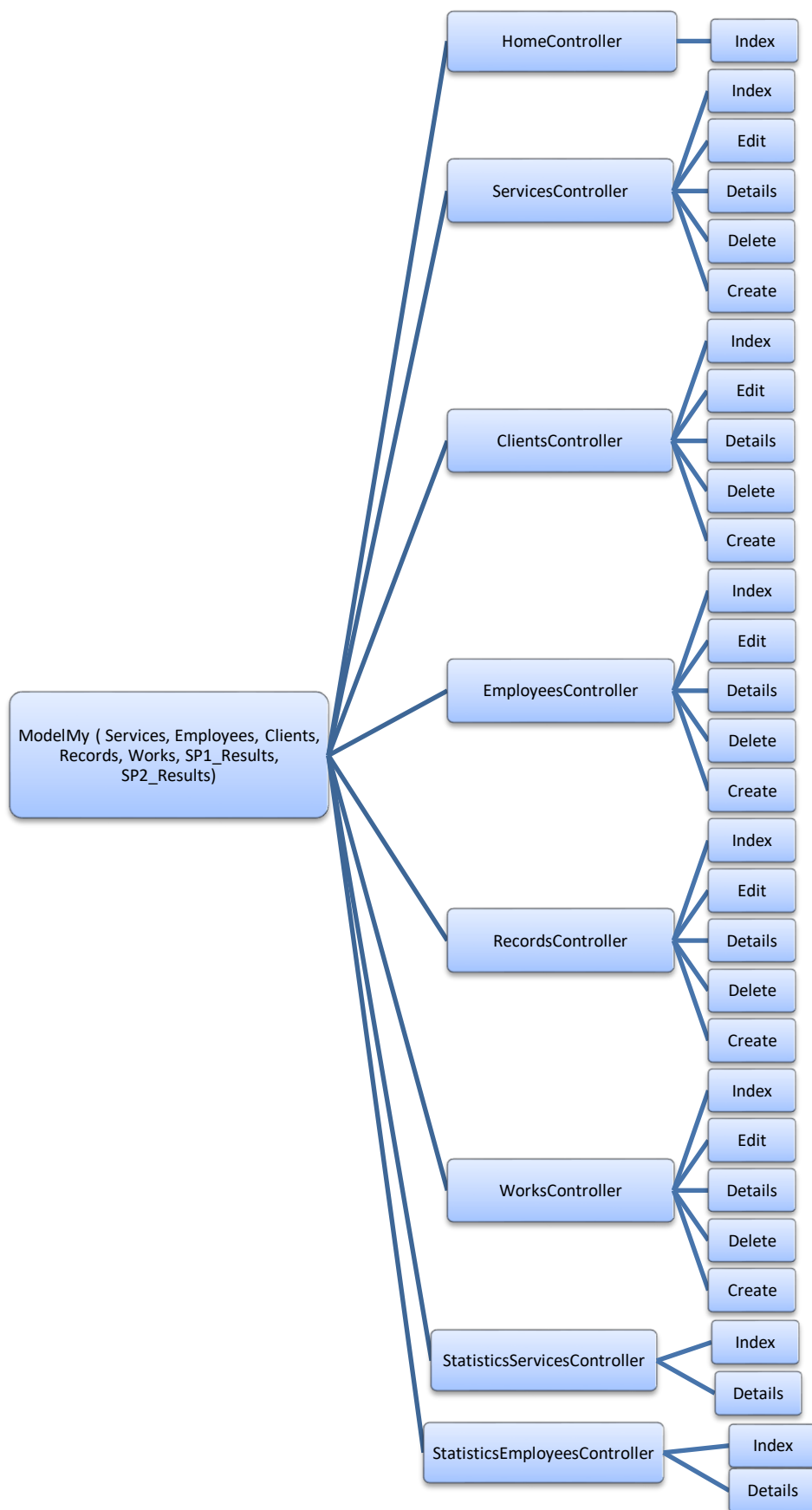


Рисунок 3 – Схема функционирования модели, контроллеров и представлений

3 Модель данных

Все сущности в приложении принято выделять в отдельные модели. Данные моделей хранятся в базе данных. Для взаимодействия с базой данных, использовался фреймворк Entity Framework.

Ниже описан процесс добавления модели с помощью визуального конструктора:

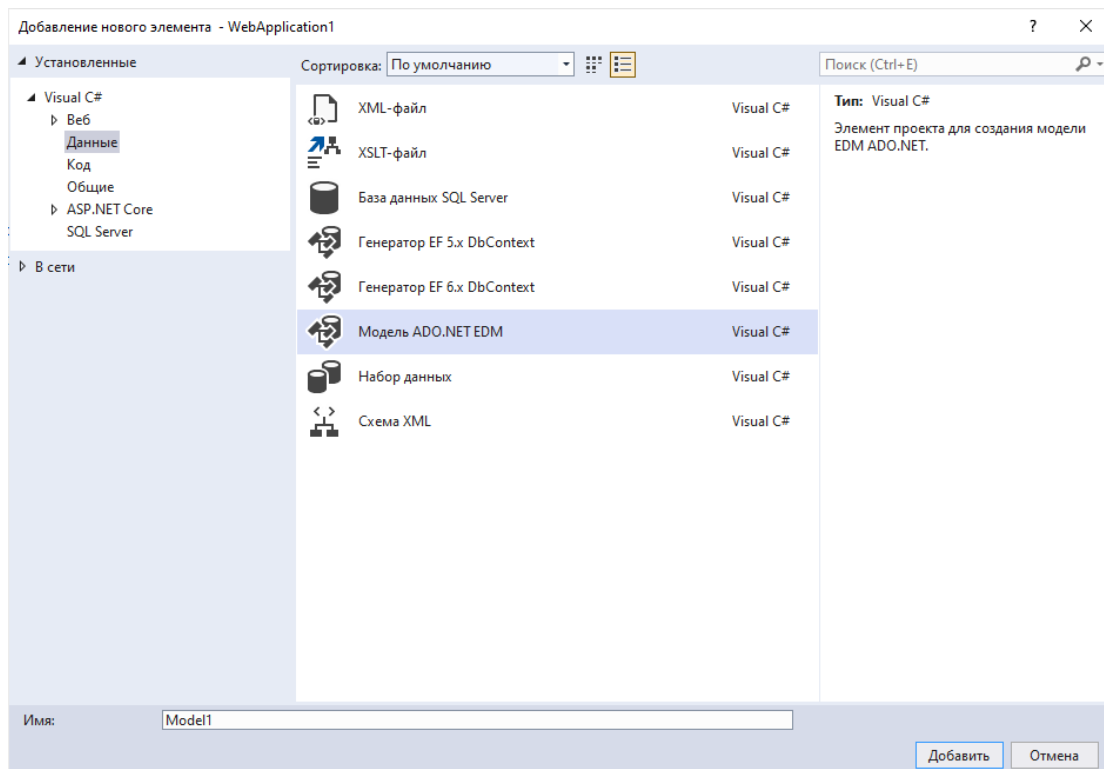


Рисунок 4 – Выбор элемента модели

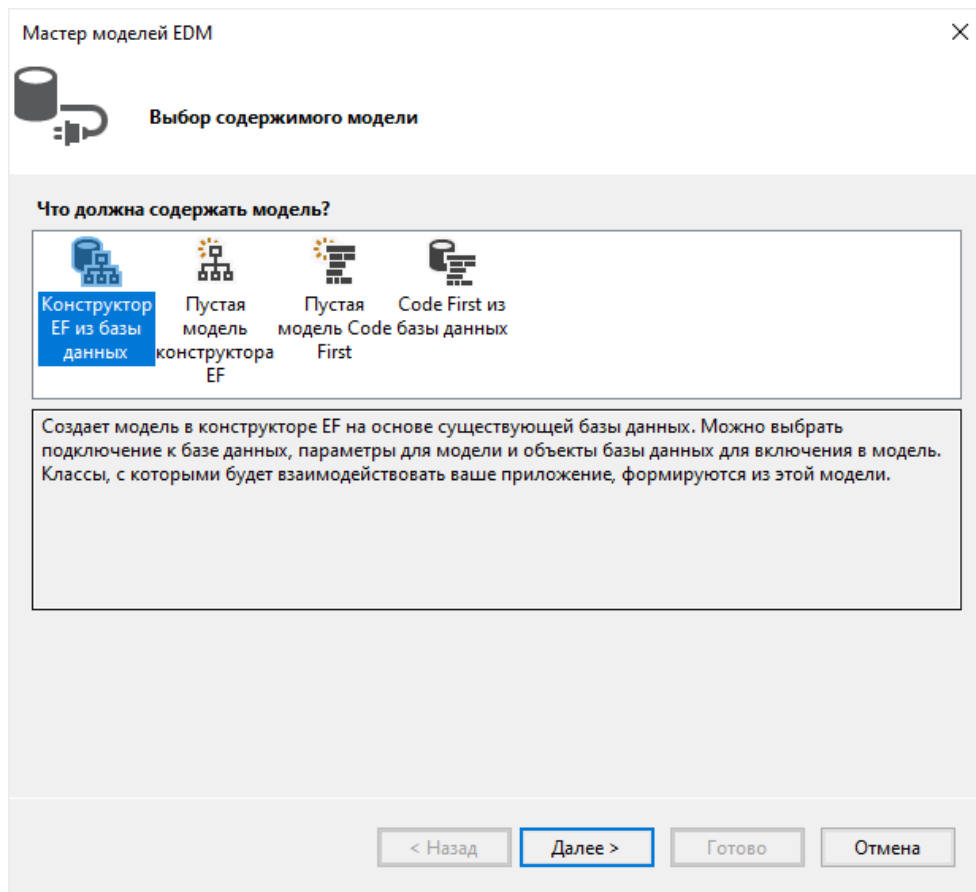


Рисунок 5 – Выбор содержимого модели (из существующей БД)

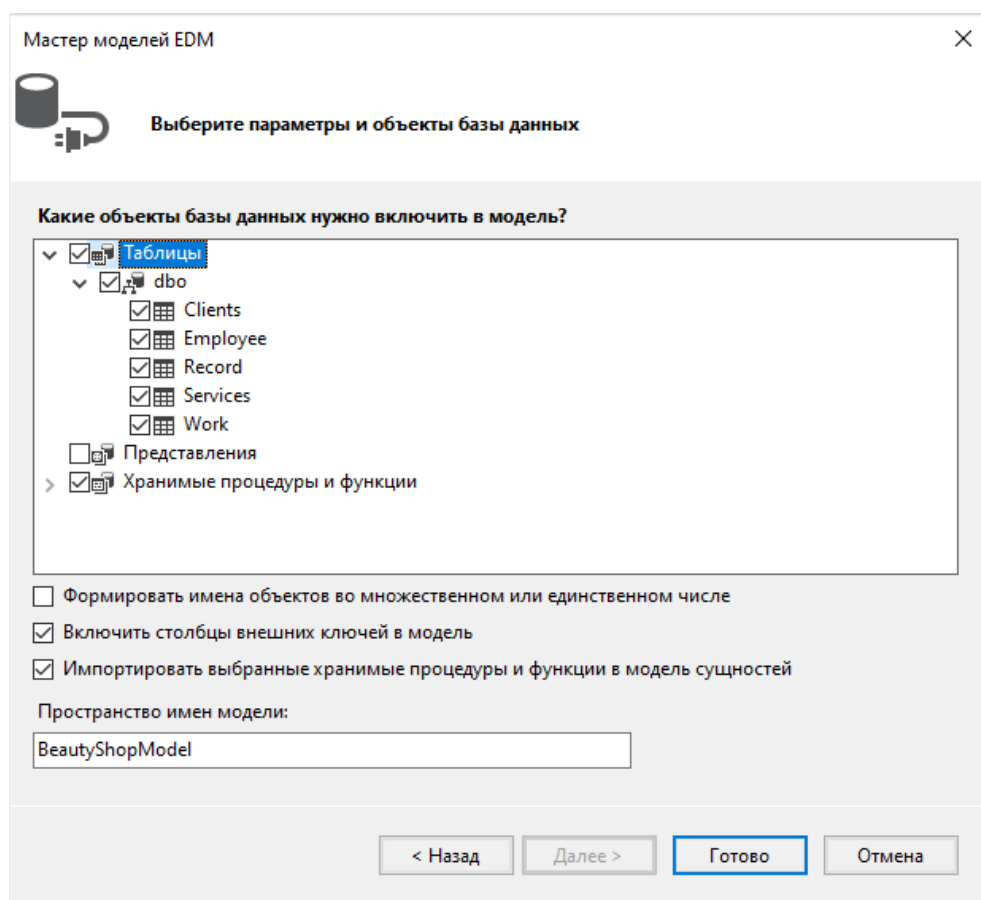


Рисунок 6 – Выбор необходимых объектов

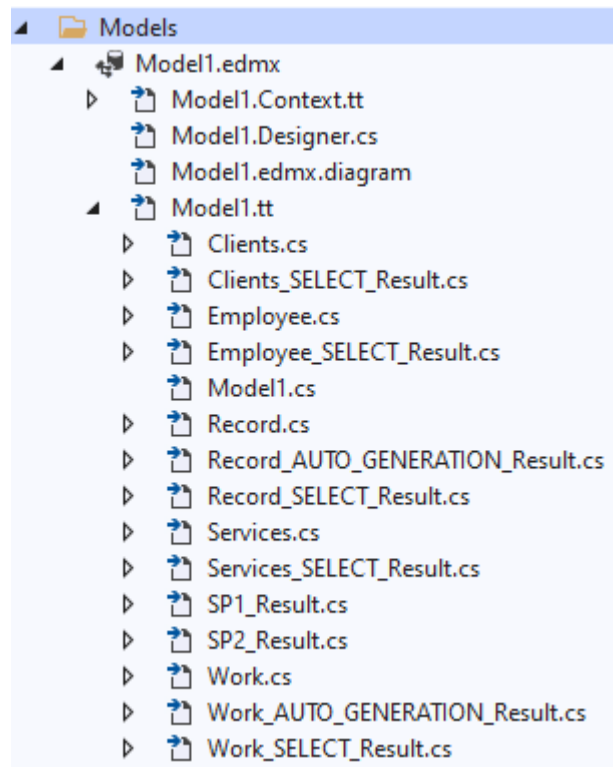


Рисунок 7 – Модель в окне «Обозреватель решений»

4 Контроллеры

Контроллер является центральным компонентом в архитектуре MVC. Контроллер получает ввод пользователя, обрабатывает его и посылает обратно результат обработки, например, в виде представления.

Рассмотрим некоторые контроллеры, а начнем с создания контроллера «ClientsController» конструктором.

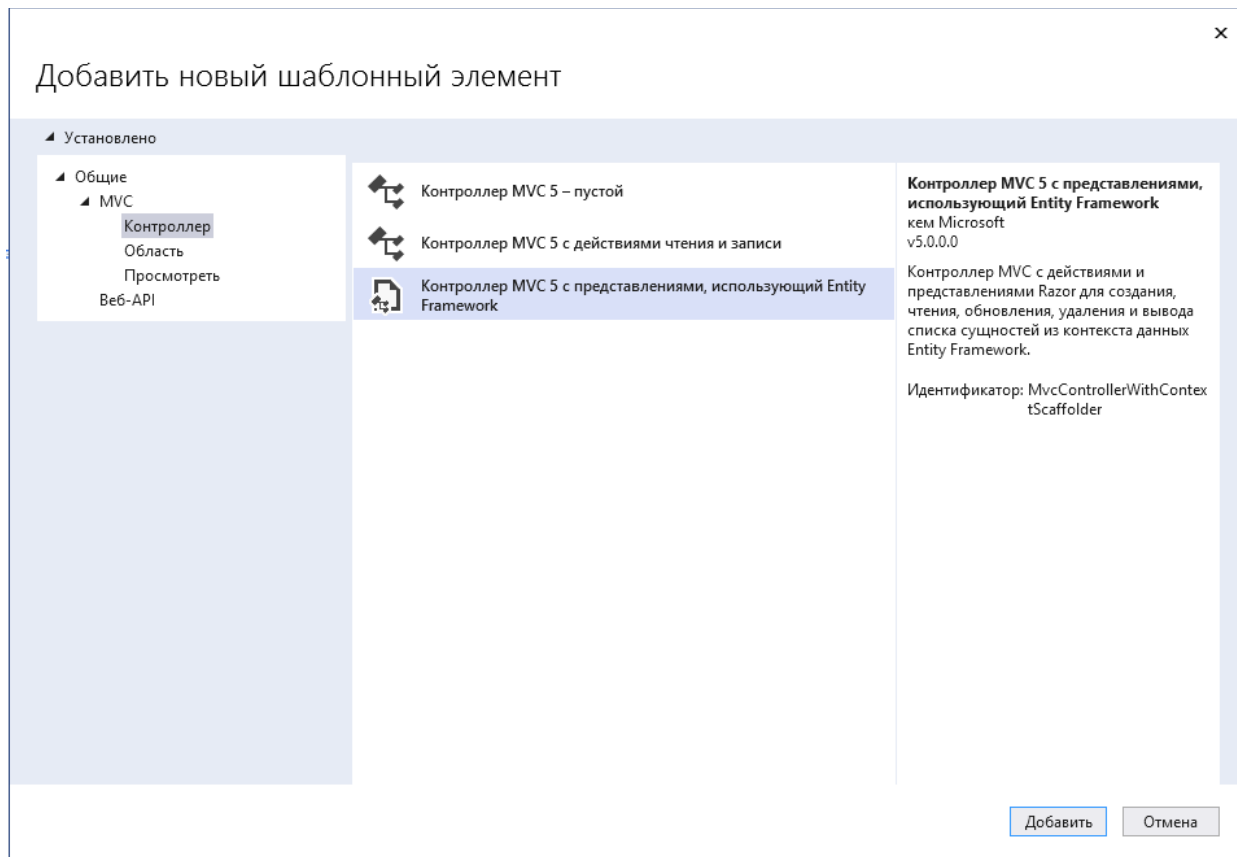


Рисунок 8 – Создание контроллера

Рисунок 9 – Добавление контроллера

Методы действий (action methods) представляют такие методы контроллера, которые обрабатывают запросы по определенному URL. В данном случае при автоматическом создании контроллера, создаются и методы действий:

- Index – метод для вывода данных всей таблицы
- Details – метод для вывода данных одной записи из таблицы
- Create – метод для создания новой записи
- Edit – метод для редактирования данных одной записи из таблицы
- Delete – метод для удаления данных одной записи из таблицы

После автоматического создания методы контроллеров были отредактированы. Далее приводятся 3 примера контроллеров: ClientsController(таблица без FK), WorksController(таблица с FK), StatisticsServicesController(выходной документ).

4.1 ClientsController

```
using PagedList;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using Web1.Models;

namespace Web1.Controllers
{
    public class ClientsController : Controller
```

```

    {
        private BeautyShopEntities1 db = new BeautyShopEntities1();

        // GET: Clients
        public ActionResult Index(string sortOrder, string currentFilter, string
searchString, int? page)
        {
            ViewBag.CurrentSort = sortOrder;
            ViewBag.SurnameSortParm = String.IsNullOrEmpty(sortOrder) ? "Sur-
name_desc" : "";
            ViewBag.NameSortParm = sortOrder == "name" ? "name_desc" : "name";
            ViewBag.MiddleSortParm = sortOrder == "middle" ? "middle_desc" : "mid-
dle";
            ViewBag.PhoneSortParm = sortOrder == "phone" ? "phone_desc" : "phone";
            ViewBag.BirhdateSortParm = sortOrder == "birhdate" ? "birhdate_desc"
: "birhdate";
            ViewBag.PostSortParm = sortOrder == "email" ? "email_desc" : "email";
            ViewBag.CommentSortParm = sortOrder == "comment" ? "comment_desc" :
"comment";
            if (searchString != null)
            {
                page = 1;
            }
            else
            {
                searchString = currentFilter;
            }

            ViewBag.CurrentFilter = searchString;

            var clients = from s in db.Clients
                           select s;
            if (!String.IsNullOrEmpty(searchString))
            {
                clients = clients.Where(s => s.Name.Contains(searchString)
|| s.Surname.Contains(searchString)
|| s.Middle.Contains(searchString));
            }
            switch (sortOrder)
            {
                case "Surname_desc":
                    clients = clients.OrderByDescending(s => s.Surname);
                    break;
                case "name":
                    clients = clients.OrderBy(s => s.Name);
                    break;
                case "name_desc":
                    clients = clients.OrderByDescending(s => s.Name);
                    break;
                case "middle":
                    clients = clients.OrderBy(s => s.Middle);
                    break;
                case "middle_desc":
                    clients = clients.OrderByDescending(s => s.Middle);
                    break;
                case "phone":
                    clients = clients.OrderBy(s => s.Phone);
                    break;
                case "phone_desc":
                    clients = clients.OrderByDescending(s => s.Phone);
                    break;
                case "email":
                    clients = clients.OrderBy(s => s.Email);

```

```

        break;
    case "email_desc":
        clients = clients.OrderByDescending(s => s.Email);
        break;
    case "comment":
        clients = clients.OrderBy(s => s.Comment);
        break;
    case "comment_desc":
        clients = clients.OrderByDescending(s => s.Comment);
        break;
    case "birhdate":
        clients = clients.OrderBy(s => s.Birhdate);
        break;
    case "birhdate_desc":
        clients = clients.OrderByDescending(s => s.Birhdate);
        break;
    default: // Name ascending
        clients = clients.OrderBy(s => s.Surname);
        break;
    }
    int pageSize = 10;
    int pageNumber = (page ?? 1);
    return View(clients.ToPagedList(pageNumber, pageSize));
}
// GET: Clients/Details/5
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Clients clients = db.Clients.Find(id);
    if (clients == null)
    {
        return HttpNotFound();
    }
    return View(clients);
}

// GET: Clients/Create
public ActionResult Create()
{
    return View();
}

// POST: Clients/Create
// Чтобы защититься от атак чрезмерной передачи данных, включите
определенные свойства, для которых следует установить привязку.
// Дополнительные сведения см. в статье https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Id,Surname,Name,Middle,Birhdate,Email,Phone,Comment")] Clients clients)
{
    if (ModelState.IsValid)
    {
        try
        {
            var e = db.Clients.Where(s => s.Phone.Equals(((decimal)clients.Phone)));
            if (e.Any())
            {

```

```

        ModelState.AddModelError("", "Клиент с таким номером
телефона уже существует.");
    }
    else
    {
        db.Clients.Add(clients);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
}
catch
{
    ModelState.AddModelError("", "Клиент уже существует.");
}

return View(clients);
}

// GET: Clients/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Clients clients = db.Clients.Find(id);
    if (clients == null)
    {
        return HttpNotFound();
    }
    return View(clients);
}

// POST: Clients/Edit/5
// Чтобы защититься от атак чрезмерной передачи данных, включите
определенные свойства, для которых следует установить привязку.
// Дополнительные сведения см. в статье https://go.mi-
crosoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "Id,Surname,Name,Middle,Bir-
hdate,Email,Phone,Comment")] Clients clients)
{
    if (ModelState.IsValid)
    {
        try
        {
            db.Entry(clients).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        catch
        {
            ModelState.AddModelError("", "Клиент уже существует.");
        }
    }
    return View(clients);
}

// GET: Clients/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)

```

```

        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Clients clients = db.Clients.Find(id);
        if (clients == null)
        {
            return HttpNotFound();
        }
        return View(clients);
    }

    // POST: Clients/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Clients clients = db.Clients.Find(id);
        db.Clients.Remove(clients);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

4.2 WorksController

```

using PagedList;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using Web1.Models;

namespace Web1.Views
{
    public class WorksController : Controller
    {
        private BeautyShopEntities1 db = new BeautyShopEntities1();

        // GET: Works
        public ActionResult Index(string sortOrder, string currentFilter, int?
page)
        {
            ViewBag.CurrentSort = sortOrder;
            ViewBag.DateSortParm = String.IsNullOrEmpty(sortOrder) ? "Date_desc"
: "";
            ViewBag.TimeSortParm = sortOrder == "time" ? "time_desc" : "time";
            ViewBag.ClientsSortParm = sortOrder == "clients" ? "clients_desc" :
"clients";
            ViewBag.IncomeSortParm = sortOrder == "income" ? "income_desc" : "in-
come";

```



```

        ViewBag.EmployeeSortParm = sortOrder == "employee" ? "employee_desc"
: "employee";
        ViewBag.ServicesSortParm = sortOrder == "services" ? "services_desc"
: "services";

        var work = from s in db.Work
                    select s;
        switch (sortOrder)
        {
            case "Date_desc":
                work = work.OrderByDescending(s => s.Date);
                break;
            case "time":
                work = work.OrderBy(s => s.Time);
                break;
            case "time_desc":
                work = work.OrderByDescending(s => s.Time);
                break;
            case "income":
                work = work.OrderBy(s => s.Income);
                break;
            case "income_desc":
                work = work.OrderByDescending(s => s.Income);
                break;
            case "clients":
                work = work.OrderBy(s => s.Id_Clients);
                break;
            case "clients_desc":
                work = work.OrderByDescending(s => s.Id_Clients);
                break;
            case "employee":
                work = work.OrderBy(s => s.Id_Employee);
                break;
            case "employee_desc":
                work = work.OrderByDescending(s => s.Id_Employee);
                break;
            case "services":
                work = work.OrderBy(s => s.Id_Services);
                break;
            case "services_desc":
                work = work.OrderByDescending(s => s.Id_Services);
                break;
            default: // Name ascending
                work = work.OrderBy(s => s.Date);
                break;
        }
        int pageSize = 10;
        int pageNumber = (page ?? 1);
        return View(work.ToPagedList(pageNumber, pageSize));
    }

    // GET: Works/Details/5
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Work work = db.Work.Find(id);
        if (work == null)
        {
            return HttpNotFound();
        }
    }

```

```

        return View(work);
    }

    // GET: Works/Create
    public ActionResult Create()
    {
        ViewBag.Id_Employee = new SelectList(db.Employee, "Id", "FIO");
        ViewBag.Id_Services = new SelectList(db.Services, "Id", "Concat");
        return View();
    }

    // POST: Works/Create
    // Чтобы защититься от атак чрезмерной передачи данных, включите
определенные свойства, для которых следует установить привязку.
    // Дополнительные сведения см. в статье https://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include = "Id,Id_Services,Id_Employee,Date,Time,Income,Comment")] Work work)
    {
        var clients = from s in db.Clients select s;
        var Name = Request["Clients.Name"].ToString();
        var Surname = Request["Clients.Surname"].ToString();
        var Middle = Request["Clients.Middle"].ToString();
        var c = clients.Where(s => s.Name.Contains(Name)
            & s.Surname.Contains(Surname)
            & s.Middle.Contains(Middle)).FirstOrDefault();

        try
        {
            if (c != null)
            {
                if (ModelState.IsValid)
                {
                    work.Id_Clients = c.Id;
                    db.Work.Add(work);
                    db.SaveChanges();
                    return RedirectToAction("Index");
                }
            }
            else {
                ModelState.AddModelError("", "Клиента не существует");
            }
        }
        catch
        {
            ModelState.AddModelError("", "Работа уже существует.");
        }

        ViewBag.Id_Employee = new SelectList(db.Employee, "Id", "FIO",
work.Id_Employee);
        ViewBag.Id_Services = new SelectList(db.Services, "Id", "Concat",
work.Id_Services);
        return View(work);
    }

    // GET: Works/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
    }

```

```

        Work work = db.Work.Find(id);
        if (work == null)
        {
            return HttpNotFound();
        }

        ViewBag.Id_Employee = new SelectList(db.Employee, "Id", "FIO",
work.Id_Employee);
        ViewBag.Id_Services = new SelectList(db.Services, "Id", "Concat",
work.Id_Services);
        return View(work);
    }

    // POST: Works/Edit/5
    // Чтобы защититься от атак чрезмерной передачи данных, включите
определенные свойства, для которых следует установить привязку.
    // Дополнительные сведения см. в статье https://go.mi-
crosoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include = "Id,Id_Services,Id_Em-
ployee,Id_Clients,Date,Time,Income,Comment")] Work work)
    {
        var clients = from s in db.Clients select s;
        var Name = Request["Clients.Name"].ToString();
        var Surname = Request["Clients.Surname"].ToString();
        var Middle = Request["Clients.Middle"].ToString();
        var c = clients.Where(s => s.Name.Contains(Name)
            & s.Surname.Contains(Surname)
            & s.Middle.Contains(Middle)).FirstOrDefault();

        try
        {
            if (c != null)
            {
                if (ModelState.IsValid)
                {
                    work.Id_Clients = c.Id;
                    db.Entry(work).State = EntityState.Modified;
                    db.SaveChanges();
                    return RedirectToAction("Index");
                }
            }
            else
            {
                ModelState.AddModelError("", "Клиента не существует");
            }
        }
        catch
        {
            ModelState.AddModelError("", "Работа уже существует.");
        }

        ViewBag.Id_Employee = new SelectList(db.Employee, "Id", "FIO",
work.Id_Employee);
        ViewBag.Id_Services = new SelectList(db.Services, "Id", "Concat",
work.Id_Services);
        return View(work);
    }

    // GET: Works/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {

```

```

        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Work work = db.Work.Find(id);
    if (work == null)
    {
        return HttpNotFound();
    }
    return View(work);
}

// POST: Works/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Work work = db.Work.Find(id);
    db.Work.Remove(work);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

4.3 StatisticsServicesController

Особенностью выходного документа является отсутствие возможности редактировать, удалять и создавать новые записи. Это связано с тем, что данные для этого контроллера — это статистика, собираемая из других таблиц.

```

using PagedList;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Web1.Models;

namespace Web1.Controllers
{
    public class StatisticsServicesController : Controller
    {
        private BeautyShopEntities1 db = new BeautyShopEntities1();

        public ActionResult Index()
        {
            return View();
        }

        // GET: Statistics
        public ActionResult Details(string currentFilter, string searchString,
int? page)
        {

```

```

        if (searchString != null)
        {
            page = 1;
        }
        else
        {
            searchString = currentFilter;
        }
        String d = null;
        try {
            Convert.ToInt32(searchString);
            d = searchString;
        }
        catch {
            ModelState.AddModelError("", "Вы ввели не число");
            TempData["msg"] = "1";
            return View();
        }
        ViewBag.CurrentFilter = searchString;
        if ((String.IsNullOrEmpty(searchString) && page == 1) || (Convert.ToInt32(searchString) > 2050) || (Convert.ToInt32(searchString) < 1900))
        {
            TempData["msg"] = "1";
            return View();
        }
        else
        {
            TempData["msg"] = "";
        }

        object[] parameters = {
            new SqlParameter("@date1", SqlDbType.VarChar) {Value=d}
        };
        db.Database.CommandTimeout = 360;
        IEnumerable<SP1_Result> query = db.Database.SqlQuery<SP1_Result>("SP1
@date1", parameters).ToList();
        int pageSize = 10;
        int pageNumber = (page ?? 1);
        return View(query.ToPagedList(pageNumber, pageSize));
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}
}

```

5 Представления

Хотя работа приложения MVC управляется главным образом контроллерами, но непосредственно пользователю приложение доступно в виде представления, которое и формирует внешний вид приложения.

При создании контроллера, автоматически создаются представления Index, Edit, Details, Delete, Create. В ходе разработки проекта представления были изменены, а именно дизайн, верстка и формы для ввода.

Ниже приведены рисунки и листинг кода представлений для каждого из контроллеров раздела 4.

5.1 Клиенты

5.1.1 Index































Фамилия	Имя	Отчество	Дата рождения	Телефон	Почта	Комментарий
Абакумов	Азат	Зиганшинович	01.12.1978	79498006478	UpM5@rambler.ru	  
Абакумов	Шалеа	Егграфович	13.06.1995	79662491480	3NOCi_s@gmail.com	  
Абакумов	Муртаза	Сафронович	23.08.1982	79133148159	gGYISu@yandex.ru	  
Абакумов	Курбан	Гарифуллович	11.12.2003	79395626206	q0RW.h_KYGE4@list.ru	  
Абакумов	Станислав	Фатеевич	20.05.1973	79688563010	hbkCS7Soaf2Fs@yandex.ru	  
Абакумов	Ахметия	Шафигуллович	05.03.2005	79469922722	yDAotA@mail.ru	  
Абакумов	Иосиф	Пиманович	09.09.1981	79955568894	Pzutienyo@yandex.ru	  
Абакумов	Мубарак	Миннихметович	08.05.2003	79713496876	ITIm@yandex.ru	  
Абакумов	Афанасий	Саликович	30.06.2006	79711046589	pO8Nti5ti@yandex.ru	  
Абакумов	Филарет	Габитович	02.02.1981	79149704001	wIda3vehKIh5Kh@rambler.ru	  

Рисунок 10 – Index

```
@model PagedList.IPagedList<Web1.Models.Clients>
@using PagedList.Mvc;
<link href="~/Content/PagedList.css" rel="stylesheet" type="text/css" />
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Клиенты</h2>
<div class="row">
    <div class="col-auto">
        <a class="btn btn-outline-primary" href="@Url.Action("Create", "Clients")">
            Добавить
        </a>
    </div>
    <div class="col-auto">
        <form asp-action="Index" method="get">
            <div class="form-actions no-color row">
                <div class="input-group mb-3">
                    <input type="text" id="SearchString" name="SearchString"
                    class="form-control" placeholder="Фамилия/Имя/Отчество/Должность" aria-label="Введите ФИО">
                    <input type="submit" value="Поиск" class="btn btn-outline-primary" />
                </div>
            </div>
        </form>
    </div>
</div>
```

```

        </form>
    </div>
    <div class="col-auto">
        @Html.ActionLink("Вернуться к полному списку", "Index", new { SizePage =
@Model.PageSize }, new { @class = "btn btn-outline-primary" })
    </div>
</div>
<table class="table">
    <tr>
        <th>
            <a class="btn btn-light" href="@Url.Action("Index", "Clients", new {
sortOrder = ViewBag.SurnameSortParm, currentFilter = ViewBag.CurrentFilter
})">Фамилия</a>
        </th>
        <th>
            <a class="btn btn-light" href="@Url.Action("Index", "Clients", new {
sortOrder = ViewBag.NameSortParm, currentFilter = ViewBag.CurrentFilter
})">Имя</a>
        </th>
        <th>
            <a class="btn btn-light" href="@Url.Action("Index", "Clients", new {
sortOrder = ViewBag.MiddleSortParm, currentFilter = ViewBag.CurrentFilter
})">Отчество</a>
        </th>
        <th>
            <a class="btn btn-light" href="@Url.Action("Index", "Clients", new {
sortOrder = ViewBag.BirhdateSortParm, currentFilter = ViewBag.CurrentFilter
})">Дата рождения</a>
        </th>
        <th>
            <a class="btn btn-light" href="@Url.Action("Index", "Clients", new {
sortOrder = ViewBag.PhoneSortParm, currentFilter = ViewBag.CurrentFilter
})">Телефон</a>
        </th>
        <th>
            <a class="btn btn-light" href="@Url.Action("Index", "Clients", new {
sortOrder = ViewBag.EmailSortParm, currentFilter = ViewBag.CurrentFilter
})">Почта</a>
        </th>
        <th>
            <a class="btn btn-light" href="@Url.Action("Index", "Clients", new {
sortOrder = ViewBag.CommentSortParm, currentFilter = ViewBag.CurrentFilter
})">Комментарий</a>
        </th>
    </th></th>
    </tr>

    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Surname)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Name)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Middle)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Birhdate)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Phone)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Email)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Comment)
            </td>
        </tr>
    }
</table>

```


5.1.2 Edit

[Главная](#)
[Услуги](#)
[Сотрудники](#)
[Клиенты](#)
[Работы](#)
[Записи](#)
[Статистика по услугам](#)
[Статистика по сотрудникам](#)

Редактирование

Клиент

Фамилия

Имя

Отчество

День рождения

Email

Телефон

Комментарий

Сохранить

[Вернуться к списку](#)

Рисунок 11 – Изменение

```
@model Web1.Models.Clients

@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Редактирование</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />
        <h4>Клиент</h4>

        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)

        <div class="form-group">
            @Html.LabelFor(model => model.Surname, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Surname, new { htmlAttributes =
new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Surname, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Name, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Name, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Name, "", new { @class
= "text-danger" })
            </div>
        </div>
    </div>
```

```

</div>

<div class="form-group">
    @Html.LabelFor(model => model.Middle, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Middle, new { htmlAttributes = new
{ @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Middle, "", new {
@class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Birhdate, htmlAttributes: new { @class
= "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Birhdate, new { htmlAttributes =
new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Birhdate, "", new {
@class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Email, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Email, new { htmlAttributes = new
{ @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Email, "", new { @class
= "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Phone, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Phone, new { htmlAttributes = new
{ @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Phone, "", new { @class
= "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Comment, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Comment, new { htmlAttributes =
new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Comment, "", new {
@class = "text-danger" })
    </div>
</div>
</div>
<br />
<div class="row">
    <div class="col-auto md-6">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Сохранить" class="btn btn-success" />
        </div>
    </div>
</div>

```

```

        <div class="col-auto md-6">
            <a class="btn btn-outline-primary" href="@Url.Action("Index", "Employees", null)">
                <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
                    fill="currentColor" class="bi bi-box-arrow-in-left" viewBox="0 0 16 16">
                    <path fill-rule="evenodd" d="M10 3.5a.5.5 0 0 0-.5-.5h-8a.5.5
                    0 0 0-.5.5v9a.5.5 0 0 0 .5.5h8a.5.5 0 0 0 .5-.5v-2a.5.5 0 0 1 1 1 0v2A1.5 1.5 0 0
                    1 9.5 14h-8A1.5 1.5 0 0 1 0 12.5v-9A1.5 1.5 0 0 1 1.5 2h8A1.5 1.5 0 0 1 11
                    3.5v2a.5.5 0 0 1-1 1 0v-2z" />
                    <path fill-rule="evenodd" d="M4.146 8.354a.5.5 0 0 1 0-.708l3-.708
                    3a.5.5 0 1 1 .708.708L5.707 7.5H14.5a.5.5 0 0 1 0 1H5.707l2.147 2.146a.5.5 0 0 1-.
                    .708.708l-3-3z" />
                </svg>
                Вернуться к списку
            </a>
        </div>
    </div>
}

```

5.1.3 Details

<ul style="list-style-type: none"> Главная Услуги Сотрудники Клиенты Работы Записи Статистика по услугам Статистика по сотрудникам 	<h2>Информация</h2> <hr/> <p>Клиент</p> <p>Фамилия Комогорцева</p> <p>Имя Юлия</p> <p>Отчество Владимировна</p> <p>День рождения 20.11.2001</p> <p>Email komog-y-v@yandex.ru</p> <p>Телефон 79144767878</p> <div> ✎ Редактировать 🗑 Удалить 🏠 Вернуться к списку </div>
---	--

Рисунок 12 – Просмотр

```

@model Web1.Models.Clients

@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Информация</h2>
<div>
    <hr />
    <h4>Клиент</h4>
    <dl class="dl-horizontal">
        <dt>
            @Html.LabelFor(model => model.Surname, htmlAttributes: new {
                @class = "control-label col-md-2" })
        </dt>
        <dd class="mb-2">
            @Html.DisplayFor(model => model.Surname)
        </dd>
    </dl>

```

```

        <dt>
            @Html.LabelFor(model => model.Name, htmlAttributes: new {
@class = "control-label col-md-2" })
        </dt>

        <dd class="mb-2">
            @Html.DisplayFor(model => model.Name)
        </dd>

        <dt>
            @Html.LabelFor(model => model.Middle, htmlAttributes: new {
@class = "control-label col-md-2" })
        </dt>

        <dd class="mb-2">
            @Html.DisplayFor(model => model.Middle)
        </dd>

        <dt>
            @Html.LabelFor(model => model.Birhdate, htmlAttributes: new {
@class = "control-label col-md-2" })
        </dt>

        <dd class="mb-2">
            @Html.DisplayFor(model => model.Birhdate)
        </dd>

        <dt>
            @Html.LabelFor(model => model.Email, htmlAttributes: new {
@class = "control-label col-md-2" })
        </dt>

        <dd class="mb-2">
            @Html.DisplayFor(model => model.Email)
        </dd>

        <dt>
            @Html.LabelFor(model => model.Phone, htmlAttributes: new {
@class = "control-label col-md-2" })
        </dt>

        <dd class="mb-2">
            @Html.DisplayFor(model => model.Phone)
        </dd>

        <dt>
            @Html.DisplayFor(model => model.Comment)
        </dt>

        <dd class="mb-2">
            @Html.DisplayFor(model => model.Comment)
        </dd>
    </dl>
</div>
<div class="row">

    <div class="col-auto">
        <a class="btn btn-outline-warning" href="@Url.Action("Edit", "Cli-
ents", new { id = Model.Id })">
            <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor" class="bi bi-pencil" viewBox="0 0 16 16">

```

```

        <path d="M12.146.146a.5.5 0 0 1 .708 0l3 3a.5.5 0 0 1 0
.708l-10 10a.5.5 0 0 1-.168.11l-5 2a.5.5 0 0 1-.65-.65l2-5a.5.5 0 0 1 .11-.168l10-
10zM11.207 2.5 13.5 4.793 14.793 3.5 12.5 1.207 11.207 2.5zm1.586 3L10.5 3.207 4
9.707V10h.5a.5.5 0 0 1 .5.5v.5h.5a.5.5 0 0 1 .5.5v.5h.293l6.5-6.5zm-9.761 5.175-
.106.106-1.528 3.821 3.821-1.528.106-.106A.5.5 0 0 1 5 12.5V12h-.5a.5.5 0 0 1-.5-
.5V11h-.5a.5.5 0 0 1-.468-.325z" />
    </svg>
    Редактировать
</a>
</div>
<div class="col-auto">
    <a class="btn btn-outline-danger" href="@Url.Action("Delete", "Cli-
ents", new { id = Model.Id })">
        <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor" class="bi bi-trash" viewBox="0 0 16 16">
            <path d="M5.5 5.5A.5.5 0 0 1 6 6v6a.5.5 0 0 1-1 0V6a.5.5 0
0 1 .5-.5zm2.5 0a.5.5 0 0 1 .5.5v6a.5.5 0 0 1-1 0V6a.5.5 0 0 1 .5-.5zm3 .5a.5.5
0 0 0-1 0v6a.5.5 0 0 0 1 0V6z" />
            <path fill-rule="evenodd" d="M14.5 3a1 1 0 0 1-1 1H13v9a2 2
0 0 1-2 2H5a2 2 0 0 1-2 2V4h-.5a1 1 0 0 1-1-1V2a1 1 0 0 1 1-1H6a1 1 0 0 1 1-1h2a1
1 0 0 1 1 1h3.5a1 1 0 0 1 1 1v1zM4.118 4 4 4.059V13a1 1 0 0 0 1 1h6a1 1 0 0 0 1-
1V4.059L11.882 4H4.118zM2.5 3V2h11v1h-11z" />
        </svg>
        Удалить
    </a>
</div>
<div class="col-auto">
    <a class="btn btn-outline-primary" href="@Url.Action("Index",
"Clients",null)">
        <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor" class="bi bi-box-arrow-in-left" viewBox="0 0 16 16">
            <path fill-rule="evenodd" d="M10 3.5a.5.5 0 0 0-.5-.5h-
8a.5.5 0 0 0-.5.5v9a.5.5 0 0 0 .5.5h8a.5.5 0 0 0 .5-.5v-2a.5.5 0 0 1 1 0v2A1.5
1.5 0 0 1 9.5 14h-8A1.5 1.5 0 0 1 0 12.5v-9A1.5 1.5 0 0 1 1.5 2h8A1.5 1.5 0 0 1
11 3.5v2a.5.5 0 0 1-1 0v-2z" />
            <path fill-rule="evenodd" d="M4.146 8.354a.5.5 0 0 1 0-
.708l3-3a.5.5 0 1 1 .708.708L5.707 7.5H14.5a.5.5 0 0 1 0 1H5.707l2.147 2.146a.5.5
0 0 1-.708.708l-3-3z" />
        </svg>
        Вернуться к списку
    </a>
</div>
</div>

```

5.1.4 Delete

[Главная](#)
[Услуги](#)
[Сотрудники](#)
[Клиенты](#)
[Работы](#)
[Записи](#)
[Статистика по услугам](#)
[Статистика по сотрудникам](#)

Удаление

Клиент
Фамилия
Комогорцева
Имя
Юлия
Отчество
Владимировна
День рождения
20.11.2001
Email
komog-y-v@yandex.ru
Телефон
79144767878
Комментарий

Вы уверены что хотите удалить этого клиента?

Удалить

Возврат к списку

Рисунок 13 – Удаление

```
@model Web1.Models.Clients

@{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Удаление</h2>

<div>
    <hr />
    <h4>Клиент</h4>
    <dl class="dl-horizontal">
        <dt>
            @Html.LabelFor(model => model.Surname, htmlAttributes: new { @class =
"control-label col-md-2" })
        </dt>
        <dd>
            @Html.DisplayFor(model => model.Surname)
        </dd>
        <dt>
            @Html.LabelFor(model => model.Name, htmlAttributes: new { @class =
"control-label col-md-2" })
        </dt>
        <dd>
            @Html.DisplayFor(model => model.Name)
        </dd>
        <dt>
            @Html.LabelFor(model => model.Middle, htmlAttributes: new { @class =
"control-label col-md-2" })
        </dt>
```

```

<dd>
    @Html.DisplayFor(model => model.Middle)
</dd>

<dt>
    @Html.LabelFor(model => model.Birhdate, htmlAttributes: new { @class
= "control-label col-md-2" })
</dt>

<dd>
    @Html.DisplayFor(model => model.Birhdate)
</dd>

<dt>
    @Html.LabelFor(model => model.Email, htmlAttributes: new { @class =
"control-label col-md-2" })
</dt>

<dd>
    @Html.DisplayFor(model => model.Email)
</dd>

<dt>
    @Html.LabelFor(model => model.Phone, htmlAttributes: new { @class =
"control-label col-md-2" })
</dt>

<dd>
    @Html.DisplayFor(model => model.Phone)
</dd>

<dt>
    @Html.LabelFor(model => model.Comment, htmlAttributes: new { @class =
"control-label col-md-2" })
</dt>

<dd>
    @Html.DisplayFor(model => model.Comment)
</dd>
</dl>
<br />
<h3>Вы уверены что хотите удалить этого клиента?</h3>
<br />
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-actions no-color row">
        <div class="col-auto">
            <input type="submit" value="Удалить" class="btn btn-outline-
danger" />
        </div>
        <div class="col-auto">
            <a class="btn btn-outline-primary " href="@Url.Action("In-
dex", "Clients",null)">
                <svg      xmlns="http://www.w3.org/2000/svg"      width="16"
height="16" fill="currentColor" class="bi bi-box-arrow-in-left" viewBox="0 0 16
16">
                    <path fill-rule="evenodd" d="M10 3.5a.5.5 0 0 0-.5-
.5h-8a.5.5 0 0 0-.5.5v9a.5.5 0 0 0 .5.5h8a.5.5 0 0 0 .5-.5v-2a.5.5 0 0 1 1 0v2A1.5
1.5 0 0 1 9.5 14h-8A1.5 1.5 0 0 1 0 12.5v-9A1.5 1.5 0 0 1 1.5 2h8A1.5 1.5 0 0 1
11 3.5v2a.5.5 0 0 1-1 0v-2z" />

```

```

<path fill-rule="evenodd" d="M4.146 8.354a.5.5 0 0 1
0-.708l3-3a.5.5 0 1 1 .708.708L5.707 7.5H14.5a.5.5 0 0 1 0 1H5.707l2.147 2.146a.5.5
0 0 1-.708.708l-3-3z" />
</svg>
Вернуться к списку
</a>
</div>
</div>
}
</div>

```

5.1.5 Create

Рисунок 14 – Добавление

```

@model Web1.Models.Clients

@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Создание</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">

        <hr />
        <h4>Клиент</h4>
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Surname, htmlAttributes: new { @class =
"control-label col-md-2" })

```



```

        <div class="col-md-10">
            @Html.EditorFor(model => model.Surname, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Surname, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "con-
trol-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Name, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Name, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Middle, htmlAttributes: new { @class = "con-
trol-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Middle, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Middle, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Birhdate, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Birhdate, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Birhdate, "", new { @class
= "text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Email, htmlAttributes: new { @class = "con-
trol-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Email, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Email, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Phone, htmlAttributes: new { @class = "con-
trol-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Phone, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Phone, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">

```

```

        @Html.LabelFor(model => model.Comment, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Comment, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Comment, "", new { @class =
"text-danger" })
        </div>
    </div>

    <br />
    <div class="row">
        <div class="col-auto md-6">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Сохранить" class="btn btn-success" />
            </div>
        </div>
        <div class="col-auto md-6">
            <a class="btn btn-outline-primary" href="@Url.Action("Index", "Cli-
ents", null)">
                <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor" class="bi bi-box-arrow-in-left" viewBox="0 0 16 16">
                    <path fill-rule="evenodd" d="M10 3.5a.5.5 0 0 0-.5-.5h-8a.5.5
0 0 0-.5.5v9a.5.5 0 0 0 .5.5h8a.5.5 0 0 0 .5-.5v-2a.5.5 0 0 1 1 1 0v2A1.5 1.5 0 0
1 9.5 14h-8A1.5 1.5 0 0 1 0 12.5v-9A1.5 1.5 0 0 1 1.5 2h8A1.5 1.5 0 0 1 11
3.5v2a.5.5 0 0 1-1 1 0v-2z" />
                    <path fill-rule="evenodd" d="M4.146 8.354a.5.5 0 0 1 0-.708l3-
3a.5.5 0 1 1 .708.708l5.707 7.5H14.5a.5.5 0 0 1 0 1H5.707l2.147 2.146a.5.5 0 0 1-
.708.708l-3-3z" />
                </svg>
                Вернуться к списку
            </a>
        </div>
    </div>
</div>
}

```

5.2 Работы

5.2.1 Index

Главная

Услуги

Сотрудники

Клиенты

Работы

Записи

Статистика по услугам

Статистика по сотрудникам

Работы

Добавить

Вернуться к полному списку

Дата	Время	Доход	Клиент	Сотрудник	Услуга	
02.10.2012	17:08:02.4133333	100	Федосеева Матрена	Минаева Гульсум	Скрабирование бровей	<div><div></div><div></div><div></div></div>
02.10.2012	01:50:18.8166667	700	Симонова Феодания	Трегубов Габдулла	Коррекция нитью/воском + окрашивание бровей (зна или краской)	<div><div></div><div></div><div></div></div>
02.10.2012	14:01:01.3100000	500	Малышев Кутдус	Мамонов Ион	Дизайн Ombre	<div><div></div><div></div><div></div></div>
02.10.2012	20:05:26.2533333	170	Гончарова Аида	Бердников Асгат	Наращивание волос Южно-русские волосы (цена за прядь с учетом стоимости работы) 50 см	<div><div></div><div></div><div></div></div>
02.10.2012	18:07:44.9700000	2000	Савельева Алиса	Тетерин Хачатур	Сложная чистка лица (при осложнениях, обширных акне и воспалениях) (техника на выбор)	<div><div></div><div></div><div></div></div>
02.10.2012	13:52:54.2433333	1500	Чернышова Инесса	Фомина Капитолина	Окрашивание короткие волосы	<div><div></div><div></div><div></div></div>
02.10.2012	04:59:43.6333333	2500	Храмов Адольф	Кучеров Елизар	Безинъекционная мезотерапия "Эффект тонального крема BB-glow"	<div><div></div><div></div><div></div></div>
02.10.2012	04:08:53.8833333	1500	Алиетшин Шавкат	Севостьянов Варфоломей	Массаж лица лимфодренажный	<div><div></div><div></div><div></div></div>
02.10.2012	16:25:45.7433333	500	Антропов Аскольд	Игнатьева Аргетина	Эпиляция(сахар/воск) Руки до локтя	<div><div></div><div></div><div></div></div>
02.10.2012	05:53:03.9600000	1400	Некрасова Изабелла	Денисов Эльмар	Педикюр женский аппаратный	<div><div></div><div></div><div></div></div>

1

2

3

4

5

6

7

8

9

10

...

»

Страница: 1 из 146428

Рисунок 15 – Index

```

@model PagedList.IPagedList<Web1.Models.Work>
using PagedList.Mvc;
<link href="~/Content/PagedList.css" rel="stylesheet" type="text/css" />
@{

```

```

    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Работы</h2>

<form asp-action="Index" method="get">
    <div class="row">
        <div class="col-auto">
            <a class="btn btn-outline-primary" href="@Url.Action("Create",
"Works")">
                Добавить
            </a>
        </div>
        <div class="col-auto">
            @Html.ActionLink("Вернуться к полному списку", "Index", new { SizePage
= @Model.PageSize }, new { @class = "btn btn-outline-primary" })
        </div>
    </div>
    <table class="table">
        <tr>
            <th>
                <a class="btn btn-light" href="@Url.Action("Index", "Works", new
{ sortOrder = ViewBag.DateSortParm, currentFilter = ViewBag.CurrentFilter
})">Дата</a>
            </th>
            <th>
                <a class="btn btn-light" href="@Url.Action("Index", "Works", new
{ sortOrder = ViewBag.TimeSortParm, currentFilter = ViewBag.CurrentFilter
})">Время</a>
            </th>
            <th>
                <a class="btn btn-light" href="@Url.Action("Index", "Works", new
{ sortOrder = ViewBag.IncomeSortParm, currentFilter = ViewBag.CurrentFilter
})">Доход</a>
            </th>
            <th>
                <a class="btn btn-light" href="@Url.Action("Index", "Works", new
{ sortOrder = ViewBag.ClientsSortParm, currentFilter = ViewBag.CurrentFilter
})">Клиент</a>
            </th>
            <th>
                <a class="btn btn-light" href="@Url.Action("Index", "Works", new
{ sortOrder = ViewBag.EmployeeSortParm, currentFilter = ViewBag.CurrentFilter
})">Сотрудник</a>
            </th>
            <th>
                <a class="btn btn-light" href="@Url.Action("Index", "Works", new
{ sortOrder = ViewBag.ServicesSortParm, currentFilter = ViewBag.CurrentFilter
})">Услуга</a>
            </th>
        </tr>

        @foreach (var item in Model)
        {
            <tr>
                <td>
                    @Html.ValueFor(modelItem => item.Date, "{0:dd/MM/yyyy}")
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Time)
                </td>
                <td>

```

```

        @Html.DisplayFor(modelItem => item.Income)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Clients.Surname)
    @Html.DisplayFor(modelItem => item.Clients.Name)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Employee.Surname)
    @Html.DisplayFor(modelItem => item.Employee.Name)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Services.Name)
    </td>
    <td class="col-2">
        <a class="btn btn-success" href="@Url.Action("Details",
"Works", new { id = item.Id })">
            <svg xmlns="http://www.w3.org/2000/svg" width="16"
height="16" fill="currentColor" class="bi bi-info-circle" viewBox="0 0 16 16">
                <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8
0 1 0 8 0a8 8 0 0 0 0 16z" />
                <path d="m8.93 6.588-2.29.287-.
.082.38.45.083c.294.07.352.176.288.469l-.738
3.468c-.194.897.105 1.319.808
1.319.545 0 1.178-.252 1.465-.598l.088-.416c-.2.176-.492.246-.686.246-.275
0-.375-.193-.304-.533L8.93 6.588z" />
            </svg>
        </a>
        <a class="btn btn-warning" href="@Url.Action("Edit", "Works",
new { id = item.Id })">
            <svg xmlns="http://www.w3.org/2000/svg" width="16"
height="16" fill="currentColor" class="bi bi-pencil" viewBox="0 0 16 16">
                <path d="M12.146.146a.5.5 0 0 1 .708 0l3 3a.5.5 0 0 1
0 .708l-10 10a.5.5 0 0 1-.168.111l-5 2a.5.5 0 0 1-.65-.65l2-5a.5.5 0 0 1
.11-.168l10-10zM11.207 2.5 13.5 4.793 14.793 3.5 12.5 1.207 11.207 2.5zm1.586
3L10.5 3.207 4 9.707V10h.5a.5.5 0 0 1 .5.5v.5h.5a.5.5 0 0 1 .5.5v.5h.293l6.5-6.5zm-9.761
5.175-.106.106-1.528 3.821 3.821-1.528.106-.106A.5.5 0 0 1 5 12.5V12h-.5a.5.5 0 0
1-.5-.5V11h-.5a.5.5 0 0 1-.468-.325z" />
            </svg>
        </a>
        <a class="btn btn-danger" href="@Url.Action("Delete",
"Works", new { id = item.Id })">
            <svg xmlns="http://www.w3.org/2000/svg" width="16"
height="16" fill="currentColor" class="bi bi-trash" viewBox="0 0 16 16">
                <path d="M5.5 5.5A.5.5 0 0 1 6 6v6a.5.5 0 0 1 1-1
0V6a.5.5 0 0 1 .5-.5zm2.5 0a.5.5 0 0 1 .5.5v6a.5.5 0 0 1 1-1 0V6a.5.5 0 0 1
.5-.5zm3 .5a.5.5 0 0 0-1 0v6a.5.5 0 0 1 0V6z" />
                <path fill-rule="evenodd" d="M14.5 3a1 1 0 0 1 1-1
1H13v9a2 2 0 0 1-2 2H5a2 2 0 0 1-2 2V4h-.5a1 1 0 0 1-1-1V2a1 1 0 0 1 1-1H6a1 1 0
0 1 1-1h2a1 1 0 0 1 1 1h3.5a1 1 0 0 1 1 1v1zM4.118 4 4.059V13a1 1 0 0 0 1 1h6a1
1 0 0 0 1-1V4.059L11.882 4H4.118z" />
            </svg>
        </a>
    </td>
</tr>
}

</table>
<div class="row">
    <div class="col-auto">
        @Html.PagedListPager(Model, page => Url.Action("Index", new { page,
SizePage = @Model.PageSize, sortOrder = ViewBag.CurrentSort, currentFilter = View-
Bag.CurrentFilter }))

    </div>

```

```

        <div class="col-auto align-self-center">
            <span class="align-middle">Страница: @(Model.PageCount < Model.Page-
Number ? 0 : Model.PageNumber) из @Model.PageCount</span>
        </div>
    </div>
</form>

```

5.2.2 Edit

Рисунок 16 – Изменение

```

@model Web1.Models.Work

@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Редактирование</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />
        <h4>Работа</h4>

        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)

        <div class="form-group">
            @Html.LabelFor(model => model.Id_Services, null, htmlAttributes: new {
@class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("Id_Services", null, htmlAttributes: new { @class
= "form-control" })
                @Html.ValidationMessageFor(model => model.Id_Services, "", new {
@class = "text-danger" })
            </div>

```

```

</div>

<div class="form-group">
    @Html.LabelFor(model => model.Id_Employee, null, htmlAttributes: new {
@class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("Id_Employee", null, htmlAttributes: new { @class
= "form-control" })
        @Html.ValidationMessageFor(model => model.Id_Employee, "", new {
@class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Id_Clients, null, htmlAttributes: new {
@class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.LabelFor(model => model.Clients.Surname, null, htmlAttributes:
new { @class = "control-label col-md-2" })
        @Html.EditorFor(model => model.Clients.Surname, new { htmlAttributes
= new { @class = "form-control " } })
        @Html.LabelFor(model => model.Clients.Name, null, htmlAttributes: new
{ @class = "control-label col-md-2" })
        @Html.EditorFor(model => model.Clients.Name, new { htmlAttributes =
new { @class = "form-control " } })
        @Html.LabelFor(model => model.Clients.Middle, null, htmlAttributes:
new { @class = "control-label col-md-2" })
        @Html.EditorFor(model => model.Clients.Middle, new { htmlAttributes =
new { @class = "form-control " } })
        @Html.ValidationMessageFor(model => model.Id_Clients, "", new {
@class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Date, htmlAttributes: new { @class = "con-
trol-label col-md-1" })
    @Html.DisplayFor(model => model.Date)
    <div class="col-md-10">
        @Html.EditorFor(model => model.Date, new { htmlAttributes = new {
@class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Date, "", new { @class =
"text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Time, htmlAttributes: new { @class = "con-
trol-label col-md-1" })
    @Html.DisplayFor(model => model.Time, new { htmlAttributes = new { @class
= "form-control col-md-auto" } })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Time, new { htmlAttributes = new {
@class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Time, "", new { @class =
"text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Income, htmlAttributes: new { @class = "con-
trol-label col-md-2" })
    <div class="col-md-10">

```

```

        @Html.EditorFor(model => model.Income, new { htmlAttributes = new {
@class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Income, "", new { @class =
"text-danger" })
    </div>
</div>

    <div class="form-group">
        @Html.LabelFor(model => model.Comment, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Comment, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Comment, "", new { @class =
"text-danger" })
        </div>
    </div>

    <br />
    <div class="row">
        <div class="col-auto md-6">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Сохранить" class="btn btn-success" />
            </div>
        </div>
        <div class="col-auto md-6">
            <a class="btn btn-outline-primary" href="@Url.Action("Index",
"Works", null)">
                <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor" class="bi bi-box-arrow-in-left" viewBox="0 0 16 16">
                    <path fill-rule="evenodd" d="M10 3.5a.5.5 0 0 0-.5-.5h-8a.5.5
0 0 0-.5.5v9a.5.5 0 0 0 .5.5h8a.5.5 0 0 0 .5-.5v-2a.5.5 0 0 1 1 1 0v2A1.5 1.5 0 0
1 9.5 14h-8A1.5 1.5 0 0 1 0 12.5v-9A1.5 1.5 0 0 1 1.5 2h8A1.5 1.5 0 0 1 11
3.5v2a.5.5 0 0 1-1 1 0v-2z" />
                    <path fill-rule="evenodd" d="M4.146 8.354a.5.5 0 0 1 0-.708l3-
3a.5.5 0 1 1 .708.708L5.707 7.5H14.5a.5.5 0 0 1 0 1H5.707l2.147 2.146a.5.5 0 0 1-
.708.708l-3-3z" />
                </svg>
                Вернуться к списку
            </a>
        </div>
    </div>
</div>
}

```

5.2.3 Details

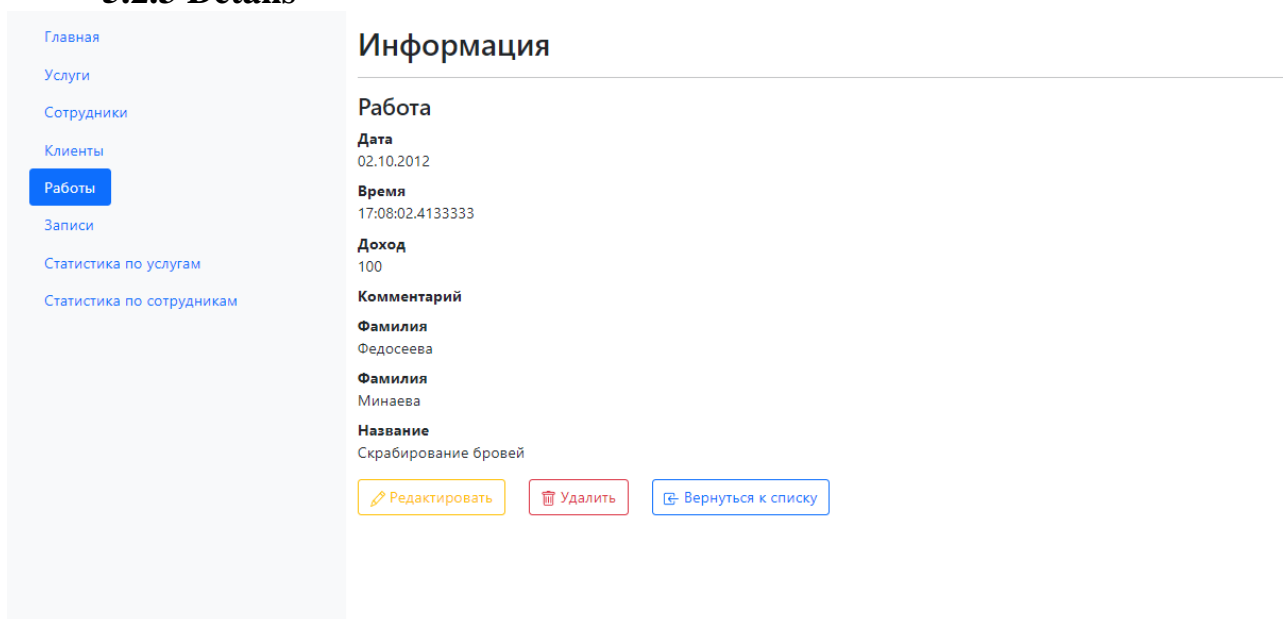


Рисунок 17 – Просмотр

```
@model Web1.Models.Work

@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Информация</h2>

<div>
    <hr />
    <h4>Работа</h4>

    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Date)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Date)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Time)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Time)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Income)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Income)
        </dd>

        <dt>
```



```

        @Html.DisplayNameFor(model => model.Comment)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Comment)
    </dd>

    <dt>
        @Html.DisplayNameFor(model => model.Clients.Surname)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Clients.Surname)
    </dd>

    <dt>
        @Html.DisplayNameFor(model => model.Employee.Surname)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Employee.Surname)
    </dd>

    <dt>
        @Html.DisplayNameFor(model => model.Services.Name)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Services.Name)
    </dd>

</dl>
</div>
<div class="row">

    <div class="col-auto">
        <a class="btn btn-outline-warning" href="@Url.Action("Edit",
"Works", new { id = Model.Id })">
            <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor" class="bi bi-pencil" viewBox="0 0 16 16">
                <path d="M12.146 146a.5.5 0 0 1 .708 0l3 3a.5.5 0 0 1 0
.708l-10 10a.5.5 0 0 1-.168 1.11l-5 2a.5.5 0 0 1-.65-.65l2-5a.5.5 0 0 1 .11-.168l10-
10zM11.207 2.5 13.5 4.793 14.793 3.5 12.5 1.207 11.207 2.5zm1.586 3L10.5 3.207 4
9.707V10h.5a.5.5 0 0 1 .5.5v.5h.5a.5.5 0 0 1 .5.5v.5h.293l6.5-6.5zm-9.761 5.175-
.106.106-1.528 3.821 3.821-1.528.106-.106A.5.5 0 0 1 5 12.5V12h-.5a.5.5 0 0 1-.5-
.5V11h-.5a.5.5 0 0 1-.468-.325z" />
            </svg>
            Редактировать
        </a>
    </div>

    <div class="col-auto">
        <a class="btn btn-outline-danger" href="@Url.Action("Delete",
"Works", new { id = Model.Id })">
            <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor" class="bi bi-trash" viewBox="0 0 16 16">
                <path d="M5.5 5.5A.5.5 0 0 1 6 6v6a.5.5 0 0 1-1 1 0V6a.5.5 0 0 1 .5-.5zm2.5 0a.5.5 0 0 1 .5.5v6a.5.5 0 0 1-1 1 0V6a.5.5 0 0 1 .5-.5zm3 .5a.5.5
0 0 0-1 0v6a.5.5 0 0 0 1 0V6z" />
                <path fill-rule="evenodd" d="M14.5 3a1 1 0 0 1-1 1H13v9a2 2
0 0 1-2 2H5a2 2 0 0 1-2 2V4h-.5a1 1 0 0 1-1 1V2a1 1 0 0 1 1-1H6a1 1 0 0 1 1-1h2a1
1 0 0 1 1 1h3.5a1 1 0 0 1 1 1v1zM4.118 4 4 4.059V13a1 1 0 0 0 1 1h6a1 1 0 0 0 1-
1V4.059L11.882 4H4.118zM2.5 3V2h11v1h-11z" />
            </svg>
            Удалить
        </a>
    </div>
</div>

```

```

        </a>
    </div>
    <div class="col-auto">
        <a class="btn btn-outline-primary " href="@Url.Action("Index",
"Works",null)">
            <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor" class="bi bi-box-arrow-in-left" viewBox="0 0 16 16">
                <path fill-rule="evenodd" d="M10 3.5a.5.5 0 0 0-.5-.5h-
8a.5.5 0 0 0-.5.5v9a.5.5 0 0 0 .5.5h8a.5.5 0 0 0 .5-.5v-2a.5.5 0 0 1 1 0v2A1.5
1.5 0 0 1 9.5 14h-8A1.5 1.5 0 0 1 0 12.5v-9A1.5 1.5 0 0 1 1.5 2h8A1.5 1.5 0 0 1
11 3.5v2a.5.5 0 0 1-1 0v-2z" />
                <path fill-rule="evenodd" d="M4.146 8.354a.5.5 0 0 1 0-
.708l3-3a.5.5 0 1 1 .708.708L5.707 7.5H14.5a.5.5 0 0 1 0 1H5.707l2.147 2.146a.5.5
0 0 1-.708.708l-3-3z" />
            </svg>
            Вернуться к списку
        </a>
    </div>
</div>

```

5.2.4 Delete

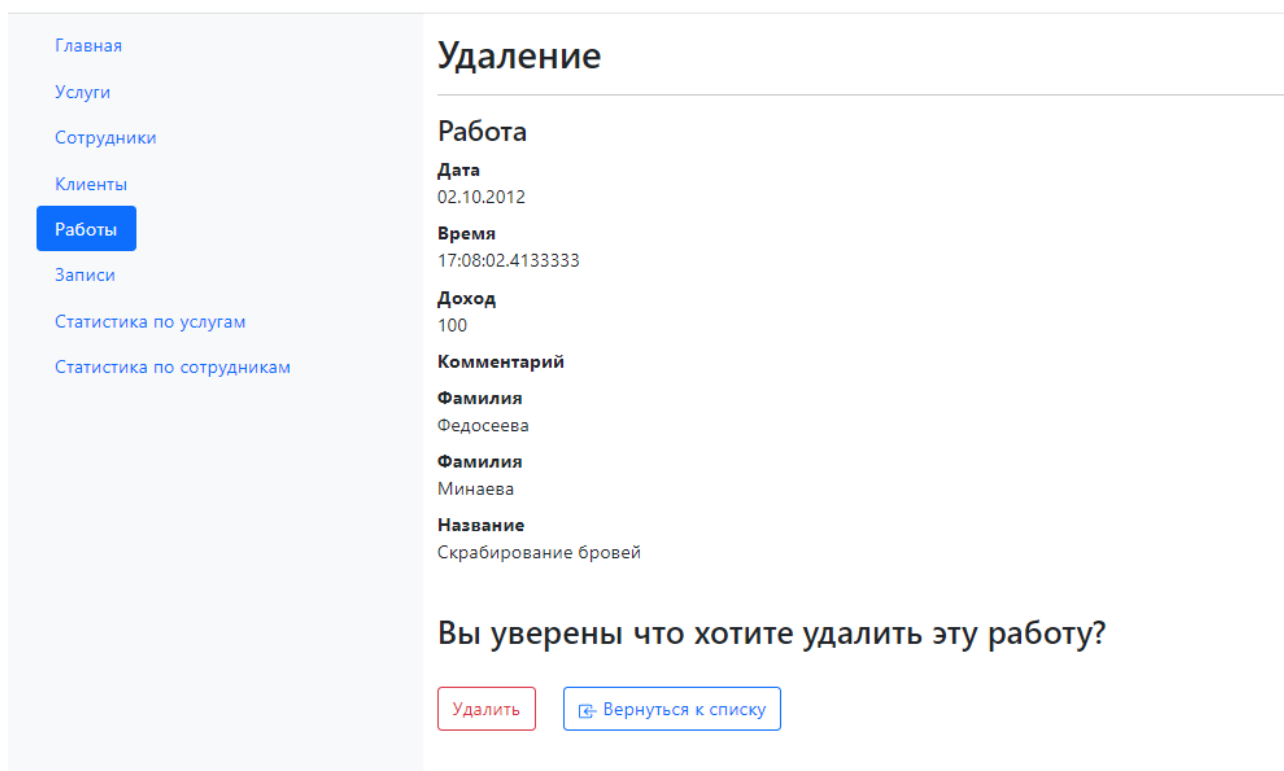


Рисунок 18 – Удаление

```

@model Web1.Models.Work

@{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Удаление</h2>

<div>
    <hr />
    <h4>Работа</h4>
    <dl class="dl-horizontal">
        <dt>

```

```

        @Html.DisplayNameFor(model => model.Date)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Date)
    </dd>

    <dt>
        @Html.DisplayNameFor(model => model.Time)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Time)
    </dd>

    <dt>
        @Html.DisplayNameFor(model => model.Income)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Income)
    </dd>

    <dt>
        @Html.DisplayNameFor(model => model.Comment)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Comment)
    </dd>

    <dt>
        @Html.DisplayNameFor(model => model.Clients.Surname)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Clients.Surname)
    </dd>

    <dt>
        @Html.DisplayNameFor(model => model.Employee.Surname)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Employee.Surname)
    </dd>

    <dt>
        @Html.DisplayNameFor(model => model.Services.Name)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Services.Name)
    </dd>
</dl>
<br />
<h3>Вы уверены что хотите удалить эту работу?</h3>
<br />
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-actions no-color row">

```

```

        <div class="col-auto">
            <input type="submit" value="Удалить" class="btn btn-outline-dan-
ger" />
        </div>
        <div class="col-auto">
            <a class="btn btn-outline-primary " href="@Url.Action("Index",
"Works",null)">
                <svg          xmlns="http://www.w3.org/2000/svg"          width="16"
height="16" fill="currentColor" class="bi bi-box-arrow-in-left" viewBox="0 0 16
16">
                    <path fill-rule="evenodd" d="M10 3.5a.5.5 0 0 0-.5-.5h-
8a.5.5 0 0 0-.5.5v9a.5.5 0 0 0 .5.5h8a.5.5 0 0 0 .5-.5v-2a.5.5 0 0 1 1 1 0v2A1.5
1.5 0 0 1 9.5 14h-8A1.5 1.5 0 0 1 0 12.5v-9A1.5 1.5 0 0 1 1.5 2h8A1.5 1.5 0 0 1
11 3.5v2a.5.5 0 0 1-1 1 0v-2z" />
                    <path fill-rule="evenodd" d="M4.146 8.354a.5.5 0 0 1 0-
.708l3-3a.5.5 0 1 1 .708.708L5.707 7.5H14.5a.5.5 0 0 1 0 1H5.707l2.147 2.146a.5.5
0 0 1-.708.708l-3-3z" />
                </svg>
                Вернуться к списку
            </a>
        </div>
    </div>
}
</div>

```

5.2.5 Create

Рисунок 19 – Добавление

```

@model Web1.Models.Work
@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Создание</h2>

@using (Html.BeginForm())
{

```

```

@Html.AntiForgeryToken()

<div class="form-horizontal row">
    <hr />
    <h4>Paĝora</h4>

    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    <div class="form-group col-8">
        @Html.LabelFor(model => model.Id_Services, null, htmlAttributes: new {
@class = "control-label col-md-2" })
        <div class="col-8">
            @Html.DropDownList("Id_Services", null, htmlAttributes: new { @class
= "form-control col-8" })
            @Html.ValidationMessageFor(model => model.Id_Services, "", new {
@class = "text-danger" })
        </div>
    </div>

    <div class="form-group col-8">
        @Html.LabelFor(model => model.Id_Employee, null, htmlAttributes: new {
@class = "control-label col-md-2" })
        <div class="col-8">
            @Html.DropDownList("Id_Employee", null, htmlAttributes: new { @class
= "form-control col-8" })
            @Html.ValidationMessageFor(model => model.Id_Employee, "", new {
@class = "text-danger" })
        </div>
    </div>

    <div class="form-group col-8">
        @Html.LabelFor(model => model.Id_Clients, null, htmlAttributes: new {
@class = "control-label col-md-2" })
        <div class="col-8">
            @Html.LabelFor(model => model.Clients.Surname, null, htmlAttributes:
new { @class = "control-label col-md-2" })
            @Html.EditorFor(model => model.Clients.Surname, new { htmlAttributes
= new { @class = "form-control col-8" } })
            @Html.LabelFor(model => model.Clients.Name, null, htmlAttributes: new
{ @class = "control-label col-md-2" })
            @Html.EditorFor(model => model.Clients.Name, new { htmlAttributes =
new { @class = "form-control col-8" } })
            @Html.LabelFor(model => model.Clients.Middle, null, htmlAttributes:
new { @class = "control-label col-md-2" })
            @Html.EditorFor(model => model.Clients.Middle, new { htmlAttributes =
new { @class = "form-control col-8" } })
            @Html.ValidationMessageFor(model => model.Id_Clients, "", new {
@class = "text-danger" })
        </div>
    </div>

    <div class="form-group col-8">
        @Html.LabelFor(model => model.Date, htmlAttributes: new { @class = "con-
trol-label col-md-2" })
        <div class="col-8">
            @Html.EditorFor(model => model.Date, new { htmlAttributes = new {
@class = "form-control col-8" } })
            @Html.ValidationMessageFor(model => model.Date, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Time, htmlAttributes: new { @class = "con-
trol-label col-md-2" })

```

```

        <div class="col-8">
            @Html.EditorFor(model => model.Time, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Time, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Income, htmlAttributes: new { @class = "con-
trol-label col-md-2" })
        <div class="col-8">
            @Html.EditorFor(model => model.Income, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Income, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Comment, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-8">
            @Html.EditorFor(model => model.Comment, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Comment, "", new { @class =
"text-danger" })
        </div>
    </div>

    <br />
    <div class="row">
        <div class="col-auto md-6">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Сохранить" class="btn btn-success" />
            </div>
        </div>
        <div class="col-auto md-6">
            <a class="btn btn-outline-primary" href="@Url.Action("Index",
"Works",null)">
                <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor" class="bi bi-box-arrow-in-left" viewBox="0 0 16 16">
                    <path fill-rule="evenodd" d="M10 3.5a.5.5 0 0 0-.5-.5h-8a.5.5
0 0 0-.5.5v9a.5.5 0 0 0 .5.5h8a.5.5 0 0 0 .5-.5v-2a.5.5 0 0 1 1 1 0v2A1.5 1.5 0 0
1 9.5 14h-8A1.5 1.5 0 0 1 0 12.5v-9A1.5 1.5 0 0 1 1.5 2h8A1.5 1.5 0 0 1 11
3.5v2a.5.5 0 0 1-1 1 0v-2z" />
                    <path fill-rule="evenodd" d="M4.146 8.354a.5.5 0 0 1 0-.708l3-
3a.5.5 0 1 1 .708.708L5.707 7.5H14.5a.5.5 0 0 1 0 1H5.707l2.147 2.146a.5.5 0 0 1-
.708.708l-3-3z" />
                </svg>
                Вернуться к списку
            </a>
        </div>
    </div>
</div>
}

```

5.3 Выходной документ со статистикой по услугам

5.3.1 Index

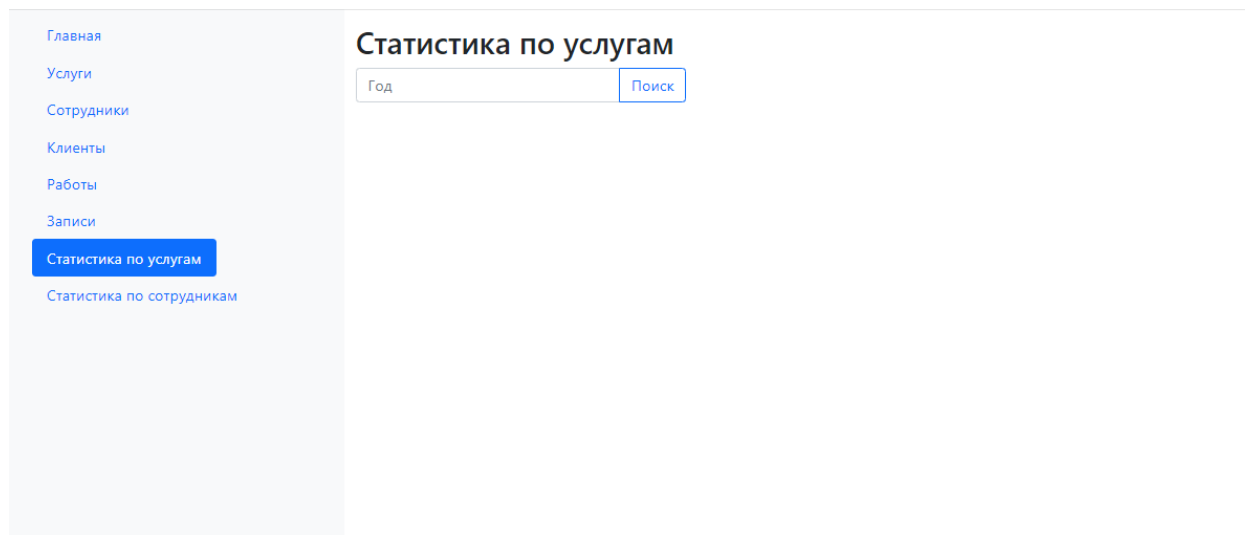


Рисунок 20 – Index

```
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<div class="row">
    <h2>Статистика по услугам</h2>

    @using (Html.BeginForm("Details", "StatisticsServices", FormMethod.Get))
    {
        <div class="col-auto">
            <form asp-action="Index" method="get">
                <div class="form-actions no-color row">
                    <div class="input-group mb-3">
                        <input type="text" id="SearchString" name="SearchString"
class="form-control" placeholder="Год" aria-label="Введите год">
                        <input type="submit" value="Поиск" class="btn btn-out-
line-primary" />
                    </div>
                </div>
            </form>
        </div>
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    }
</div>
```

5.3.2 Details

Главная	Статистика по услугам
Услуги	
Сотрудники	
Клиенты	
Работы	
Записи	
Статистика по услугам	
Статистика по сотрудникам	

Статистика по услугам

Год

Название услуги	Количество
Наращивание ресниц Эффект "Лучики/Кили/Кайли" (доплата)	1659
Массаж 60 минут (на выбор)	1642
Окрашивание бровей при ламинировании	1642
Лазерная эпиляция Классическое бикини	1638
Счастье для бровей	1637
Удаление татуировки 10-15 см	1627
Маникюр с покрытием VinyLux/ OPI	1626
Укладка на брашинг/стайлер длина ниже лопаток	1624
Наращивание волос Южно-русские волосы (цена за прядь с учетом стоимости работы) 30 см	1623
Эпиляция(сахар/воск) Руки до локтя	1623

12345678910...»

Страница: 1 из 14

Рисунок 21 – Просмотр

```
@model PagedList.IPagedList<Web1.Models.SP1_Result>
@using PagedList.Mvc;
<link href="~/Content/PagedList.css" rel="stylesheet" type="text/css" />

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Статистика по услугам</h2>

@using (Html.BeginForm("Details", "StatisticsServices", FormMethod.Get))
{
    <div class="col-auto">
        <form asp-action="Index" method="get">
            <div class="form-actions no-color row">
                <div class="input-group mb-3">
                    <input type="text" id="SearchString"
name="SearchString" class="form-control" placeholder="Год" aria-label="Введите
год">
                    <input type="submit" value="Поиск" class="btn btn-out-
line-primary" />
                </div>
            </div>
        </form>
    </div>
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
}

@if (!TempData["msg"].Equals(""))
{
    <h1>Результатов не найдено </h1>
}

@if (TempData["msg"].Equals(""))
{
    <table class="table">
        <tr>
            <th>
                Название услуги
            </th>
            <th>
                Количество
            </th>
        </tr>
```



```

        @foreach (var item in Model)
        {
            <tr>
                <td> @Html.DisplayFor(modelItem => item.Name)</td>
                <td> @Html.DisplayFor(modelItem => item.Column1)</td>
            </tr>
        }
    </table>
    <div class="row">
        <div class="col-auto">
            <form asp-action="Index" method="get">
                @Html.PagedListPager(Model, page => Url.Action("Details",
new { page, sortOrder = ViewBag.CurrentSort, currentFilter = ViewBag.CurrentFil-
ter, currentFilter1 = ViewBag.CurrentFilter1 })))
            </form>
        </div>
        <div class="col-auto align-self-center">
            <span class="align-middle">Страница:      @ (Model.PageCount  <
Model.PageNumber ? 0 : Model.PageNumber)  из  @Model.PageCount</span>
        </div>
    </div>
}

```

6 Контроль

Большую роль в приложении играет валидация модели или проверка вводимых данных на корректность. Фреймворк .NET предлагает удобный функционал для валидации в виде атрибутов из пространства имен `System.ComponentModel.DataAnnotations`.

Все правила валидации модели в `System.ComponentModel.DataAnnotations` определяются в виде атрибутов. В данном случае были использованы атрибуты: `Required`, `Display`, `DataType`, `Range`, `UIHint` и `RegularExpression`.

6.1 Services

```
namespace Web1.Models
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel.DataAnnotations;
    using System.Web.Mvc;

    public partial class Services
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
        public Services()
        {
            this.Record = new HashSet<Record>();
            this.Work = new HashSet<Work>();
        }

        public int Id { get; set; }

        [Display(Name = "Название")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        public string Name { get; set; }

        [Display(Name = "Стоимость")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        [RegularExpression(@"[z0-9]{0,10}", ErrorMessage = "Некорректная стоимость")]
        public int Price { get; set; }

        [DataType(DataType.MultilineText)]
        [Display(Name = "Комментарий")]
        public string Comment { get; set; }

        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
        public virtual ICollection<Record> Record { get; set; }
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
        public virtual ICollection<Work> Work { get; set; }
        public string Concat
        {
            get
            {
                return string.Format("{0} {1} ", Name, Price);
            }
            set { }
        }
    }
}
```

```

    }
}

```

6.2 Clients

```

namespace Web1.Models
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel.DataAnnotations;
    using Xunit;

    public partial class Clients
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
        public Clients()
        {
            this.Email = "-";
            this.Record = new HashSet<Record>();
            this.Work = new HashSet<Work>();
        }

        public int Id { get; set; }

        [Display(Name = "Фамилия")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        public string Surname { get; set; }

        [Display(Name = "Имя")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        public string Name { get; set; }

        [Display(Name = "Отчество")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        public string Middle { get; set; }

        [Display(Name = "День рождение")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        [DataType(DataType.Date)]
        [Range(typeof(DateTime), "01.01.1940", "01.01.2008",
            ErrorMessage = "Значение для {0} должно быть между {1} и {2}")]
        public Nullable<System.DateTime> Birhdate { get; set; }

        [RegularExpression(@"[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}", ErrorMessage = "Некорректный адрес")]
        [DataType(DataType.EmailAddress)]
        public string Email { get; set; }

        [Display(Name = "Телефон")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        [DataType(DataType.PhoneNumber)]
        [UIHint("String")]
        [RegularExpression(@"^\+?[78] [-\(\)?\d{3}\) ?-?\d{3}-?\d{2}-?\d{2}$", ErrorMessage = "Некорректный номер телефона")]
        public decimal Phone { get; set; }

        [Display(Name = "Комментарий")]
        [DataType(DataType.MultilineText)]
        public string Comment { get; set; }
    }
}

```

```

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
public virtual ICollection<Record> Record { get; set; }
[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
public virtual ICollection<Work> Work { get; set; }
}
}

```

6.3 Employee

```

namespace Web1.Models
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel.DataAnnotations;

    public partial class Employee
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
        public Employee()
        {
            this.Record = new HashSet<Record>();
            this.Work = new HashSet<Work>();
        }

        public int Id { get; set; }

        [Display(Name = "Фамилия")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        public string Surname { get; set; }

        [Display(Name = "Имя")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        public string Name { get; set; }

        [Display(Name = "Отчество")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        public string Middle { get; set; }

        [Display(Name = "Телефон")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        [DataType(DataType.PhoneNumber)]
        [UIHint("String")]
        [RegularExpression(@"^\+?[78] [-\ (\)?\d{3}\) ?-?\d{3}-?\d{2}-?\d{2}$", ErrorMessage = "Некорректный номер телефона")]
        public decimal Phone { get; set; }

        [Display(Name = "День рождение")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        [DataType(DataType.Date)]
        [Range(typeof(DateTime), "01.01.1940", "01.01.2008", ErrorMessage = "Значение для {0} должно быть между {1} и {2}")]
        public Nullable<System.DateTime> Birthdate { get; set; }

        [Display(Name = "Должность")]
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        public string Post { get; set; }

        [Display(Name = "Комментарий")]
        [DataType(DataType.MultilineText)]
        public string Comment { get; set; }
    }
}

```

```

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
"CA2227:CollectionPropertiesShouldBeReadOnly")]
public virtual ICollection<Record> Record { get; set; }
[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
"CA2227:CollectionPropertiesShouldBeReadOnly")]
public virtual ICollection<Work> Work { get; set; }

public string FIO
{
    get
    {
        return string.Format("{0} {1} {2} {3}", Surname, Name, Middle, Post);
    }
    set { }
}
}
}

```

6.4 Record

```

namespace Web1.Models
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel.DataAnnotations;

    public partial class Record
    {
        public int Id { get; set; }
        [Required(ErrorMessage = "Поле должно быть заполнено")]
        [Display(Name = "Услуга")]
        public int Id_Services { get; set; }

        [Required(ErrorMessage = "Поле должно быть заполнено")]
        [Display(Name = "Сотрудник")]
        public int Id_Employee { get; set; }

        [Required(ErrorMessage = "Поле должно быть заполнено")]
        [Display(Name = "Клиент")]
        public int Id_Clients { get; set; }

        [Required(ErrorMessage = "Поле должно быть заполнено")]
        [Display(Name = "Дата")]
        [DataType(DataType.Date)]
        public System.DateTime Date { get; set; }

        [Required(ErrorMessage = "Поле должно быть заполнено")]
        [Display(Name = "Время")]
        [DataType(DataType.Time)]
        public System.TimeSpan Time { get; set; }

        [Display(Name = "Комментарий")]
        public string Comment { get; set; }

        public virtual Clients Clients { get; set; }
        public virtual Employee Employee { get; set; }
        public virtual Services Services { get; set; }
    }
}

```

6.5 Work

```

namespace Web1.Models
{

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

public partial class Work
{
    public int Id { get; set; }

    [Required(ErrorMessage = "Поле должно быть заполнено")]
    [Display(Name = "Услуга")]
    public int Id_Services { get; set; }

    [Required(ErrorMessage = "Поле должно быть заполнено")]
    [Display(Name = "Сотрудник")]
    public int Id_Employee { get; set; }

    [Required(ErrorMessage = "Поле должно быть заполнено")]
    [Display(Name = "Клиент")]
    public int Id_Clients { get; set; }

    [Required(ErrorMessage = "Поле должно быть заполнено")]
    [Display(Name = "Дата")]
    [DataType(DataType.Date)]
    public System.DateTime Date { get; set; }

    [Required(ErrorMessage = "Поле должно быть заполнено")]
    [Display(Name = "Время")]
    [DataType(DataType.Time)]
    public System.TimeSpan Time { get; set; }

    [Required(ErrorMessage = "Поле должно быть заполнено")]
    [Display(Name = "Доход")]
    [RegularExpression(@"[z0-9]{0,10}", ErrorMessage = "Некорректная
стоимость")]
    public double Income { get; set; }

    [Display(Name = "Комментарий")]
    public string Comment { get; set; }

    public virtual Clients Clients { get; set; }
    public virtual Employee Employee { get; set; }
    public virtual Services Services { get; set; }
}
}

```

7 Таблица тестов

Результаты тестирования работоспособности реализованного приложения приведены в таблице 1.

Таблица 1 – Тесты

Номер теста	Название таблицы	Название теста	Входные данные	Выходные данные
1	Services	Добавление	Название: Анестезия1 Стоимость: 300 Комментарий:	Данные добавлены (см. рисунок 22)
2		Редактирование	Замена записи: Название: Анестезия1 Стоимость:300 Комментарий: На: Название: Анестезия1234 Стоимость:10 Комментарий:	Данные изменены (см. рисунок 23)
3		Просмотр	Просмотр записи: Название: Анестезия1234 Стоимость:10 Комментарий:	Просмотр данных (см. рисунок 24)
4		Удаление	Удаление записи: Название: Анестезия1234 Стоимость: 10 Комментарий:	Данные удалены (см. рисунок 25)
5	Employee	Добавление	Фамилия: aaa Имя: aaa Отчество: aaa Телефон: 79246284302 Дата рождения: 20.11.2001 Должность: aaa Комментарий:	Данные добавлены (см. рисунок 26)

Продолжение таблицы 1

Номер теста	Название таблицы	Название теста	Входные данные	Выходные данные
6	Employee	Редактирование	Замена записи: Фамилия: aaa Имя: aaa Отчество: aaa Телефон: 79246284302 Дата рождения: 20.11.2001 Должность: aaa Комментарий: На: Фамилия: aaa1 Имя: aaa1 Отчество: aaa1 Телефон: 79246284306 Дата рождения: 20.11.2001 Должность: aaa1 Комментарий:	Данные изменены (см. рисунок 27)
7		Просмотр	Просмотр записи: Фамилия: aaa1 Имя: aaa1 Отчество: aaa1 Телефон: 79246284306 Дата рождения: 20.11.2001 Должность: aaa1 Комментарий:	Просмотр данных (см. рисунок 28)
8		Удаление	Удаление записи: Фамилия: aaa1 Имя: aaa1 Отчество: aaa1 Телефон: 79246284306 Дата рождения: 20.11.2001 Должность: aaa1 Комментарий:	Данные удалены (см. рисунок 29)

Продолжение таблицы 1

Номер теста	Название таблицы	Название теста	Входные данные	Выходные данные
9	Clients	Добавление	Фамилия: aaa Имя: aaa Отчество: aaa Телефон: 79246284301 Дата рождения: 20.11.2001 Почта: komog-y-v@yandex.ru Комментарий:	Данные добавлены (см. рисунок 30)
10		Редактирование	Замена записи: Фамилия: aaa Имя: aaa Отчество: aaa Телефон: 79246284301 Дата рождения: 20.11.2001 Почта: komog-y-v@yandex.ru Комментарий: На: Фамилия: aaa1 Имя: aaa1 Отчество: aaa1 Телефон: 79246284323 Дата рождения: 21.11.2001 Почта: komog-y-v1@yandex.ru Комментарий:	Данные изменены (см. рисунок 31)
11		Просмотр	Просмотр записи: Фамилия: aaa Имя: aaa Отчество: aaa Телефон: 79246284301 Дата рождения: 20.11.2001	Просмотр данных (см. рисунок 32)

			Почта: komog-y-v@yandex.ru Комментарий:	
--	--	--	--	--

Продолжение таблицы 1

Номер теста	Название таблицы	Название теста	Входные данные	Выходные данные
12	Clients	Удаление	Удаление записи: Фамилия: aaa1 Имя: aaa1 Отчество: aaa1 Телефон: 79246284323 Дата рождения: 21.11.2001 Почта: komog-y-v1@yandex.ru Комментарий:	Данные удалены (см. рисунок 33)
13	Record	Добавление	Услуга: Снятие ресниц Сотрудник: Журавлева Ульяна Самсоновна Клиент: Комогорцева Юлия Владимировна Дата: 18.12.2021 Время: 17:34 Комментарий:	Данные добавлены (см. рисунок 34)
14		Редактирование	Замена записи: Услуга: Снятие ресниц Сотрудник: Журавлева Ульяна Самсоновна Клиент: Комогорцева Юлия Владимировна Дата: 18.12.2021 Время: 17:34 Комментарий: На: Услуга: Наращивание ресниц 3D Сотрудник: Матвеева Ирина Маркиановна	Данные изменены (см. рисунок 35)

			Клиент: Комогорцева Юлия Владимировна Дата: 19.01.2001 Время: 18:00 Комментарий:	
--	--	--	--	--

Продолжение таблицы 1

Номер теста	Название таблицы	Название теста	Входные данные	Выходные данные
15	Record	Просмотр	Просмотр записи: Услуга: Снятие ресниц Сотрудник: Журавлева Ульяна Самсоновна Клиент: Комогорцева Юлия Владимировна Дата: 18.12.2021 Время: 17:34 Комментарий:	Просмотр данных (см. рисунок 36)
16		Удаление	Удаление записи: Услуга: Снятие ресниц Сотрудник: Журавлева Ульяна Самсоновна Клиент: Комогорцева Юлия Владимировна Дата: 18.12.2021 Время: 17:34 Комментарий:	Данные удалены (см. рисунок 37)
17	Work	Добавление	Услуга: Снятие ресниц Сотрудник: Журавлева Ульяна Самсоновна Клиент: Комогорцева Юлия Владимировна Дата: 31.12.2021 Время: 10:30 Доход: 700 Комментарий:	Данные добавлены (см. рисунок 38)
18		Просмотр	Просмотр записи: Услуга: Снятие ресниц	Просмотр данных (см. рисунок 39)

			Сотрудник: Журавлева Ульяна Самсоновна Клиент: Комогорцева Юлия Владимировна Дата: 31.12.2021 Время: 10:30 Доход: 700 Комментарий:	
--	--	--	---	--

Продолжение таблицы 1

Номер теста	Название таблицы	Название теста	Входные данные	Выходные данные
19	Work	Редактирование	Замена записи: Услуга: Снятие ресниц Сотрудник: Журавлева Ульяна Самсоновна Клиент: Комогорцева Юлия Владимировна Дата: 31.12.2021 Время: 10:30 Доход: 700 Комментарий: На: Услуга: Скрабирование бровей Сотрудник: Матвеева Ирина Маркияновна Клиент: Комогорцева Юлия Владимировна Дата: 01.01.2022 Время: 10:00 Доход: 150 Комментарий:	Данные изменены (см. рисунок 40)
20		Удаление	Удаление записи: Услуга: Скрабирование бровей Сотрудник: Матвеева Ирина Маркияновна Клиент: Комогорцева Юлия Владимировна Дата: 01.01.2022 Время: 10:00 Доход: 150	Данные удалены (см. рисунок 41)

			Комментарий:	
--	--	--	--------------	--

Продолжение таблицы 1

Номер теста	Название таблицы	Название теста	Входные данные	Выходные данные
21	Запрос статистики по услугам	Запрос	Год: 2020	Перечень выходных данных (см. рисунок 42)
22	Запрос статистики по сотрудникам	Запрос	Буква: б	Перечень выходных данных (см. рисунок 43)
23	Контроль данных	Контроль данных		Перечень используемых контролей (см. рисунок 44)

8 Результаты тестирования

Результат теста №1

Создание

Услуга

Название

Анестезия1

Стоимость

300

Комментарий

Сохранить

Вернуться к списку

Услуги

Добавить

Название/Стоимость

Поиск

Вернуться к полному списку

Название	Стоимость	Комментарий	
Анестезия	503		<div><div></div><div></div><div></div></div>
Анестезия1	300		<div><div></div><div></div><div></div></div>
Афрокудри	3000		<div><div></div><div></div><div></div></div>
Безинъекционная мезотерапия "Эффект тонального крема BB-glow"	2500		<div><div></div><div></div><div></div></div>
Ботокс волос японской косметикой "Honna Tokio"	5500		<div><div></div><div></div><div></div></div>
Вечерняя укладка	3000		<div><div></div><div></div><div></div></div>
Всесезонный супер-пилинг "Идеальная кожа после первой процедуры"	3000		<div><div></div><div></div><div></div></div>
Декопирование волос (выход из темного в блонд)	6000		<div><div></div><div></div><div></div></div>
Дизайн Ombre	500		<div><div></div><div></div><div></div></div>
Дизайн Френч	400		<div><div></div><div></div><div></div></div>

1

2

3

4

5

6

7

8

9

10

...

»

»»

Страница: 1 из 14

Рисунок 22 – Добавление «Services»

Результат теста №2

Редактирование

Услуга

Название

Стоимость

Комментарий

Сохранить

[Вернуться к списку](#)



Редактирование

Услуга

Название

Стоимость

Комментарий

Сохранить

[Вернуться к списку](#)



Услуги

[Добавить](#)

[Название/Стоимость](#)

[Поиск](#)

[Вернуться к полному списку](#)

Название	Стоимость	Комментарий	
Анестезия	503		i ✎ ✖
Анестезия1234	10		i ✎ ✖
Афрокудри	3000		i ✎ ✖
Безинъекционная мезотерапия "Эффект тонального крема BB-glow"	2500		i ✎ ✖
Ботокс волос японской косметикой "Homna Tokio"	5500		i ✎ ✖
Вечерняя укладка	3000		i ✎ ✖
Всесезонный супер-пилинг "Идеальная кожа после первой процедуры"	3000		i ✎ ✖
Декопирование волос (выход из темного в блонд)	6000		i ✎ ✖
Дизайн Ombre	500		i ✎ ✖
Дизайн Френч	400		i ✎ ✖

1 2 3 4 5 6 7 8 9 10 ... » »»

Страница: 1 из 14

Рисунок 23 – Изменение «Services»

Результат теста №3

Услуги

Добавить	<input type="text" value="Название/Стоимость"/>	Поиск	Вернуться к полному списку
Название	Стоимость	Комментарий	
Анестезия	503		i ✎ ✖
Анестезия1234	10		i ✎ ✖
Афрокудри	3000		i ✎ ✖
Безинъекционная мезотерапия "Эффект тонального крема BB-glow"	2500		i ✎ ✖
Ботокс волос японской косметикой "Homna Tokio"	5500		i ✎ ✖
Вечерняя укладка	3000		i ✎ ✖
Всесезонный супер-пилинг "Идеальная кожа после первой процедуры"	3000		i ✎ ✖
Декопирование волос (выход из темного в блонд)	6000		i ✎ ✖
Дизайн Ombre	500		i ✎ ✖
Дизайн Френч	400		i ✎ ✖

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) ... [»](#) [»»](#)

Страница: 1 из 14



Информация

Услуга

Название

Анестезия1234

Стоимость

10 р.

Комментарий[✎ Редактировать](#)[✖ Удалить](#)[↶ Вернуться к списку](#)

Рисунок 24 –Просмотр «Services»

Результат теста №4

Удаление

Услуга

Название

Анестезия1234

Стоимость

10 р.

Комментарий

Вы уверены что хотите удалить эту услугу?

Удалить

Вернуться к списку



Услуги

Добавить

Название/Стоимость

Поиск

Вернуться к полному списку













Название	Стоимость	Комментарий
Анестезия	503	  
Афрокудри	3000	  
Безинъекционная мезотерапия "Эффект тонального крема BB-glow"	2500	  
Ботокс волос японской косметикой "Homna Tokio"	5500	  

Рисунок 25 – Удаление «Services»

Результат теста №5

Создание

Сотрудник

Фамилия

aaa

Имя

aaa

Отчество

aaa

Телефон

+79246284302

День рождения

20.11.2001

Должность

aaa

Комментарий

Сохранить

Вернуться к списку

































Сотрудники

Добавить

Фамилия/Имя/Отчество/И

Поиск

Вернуться к полному списку

Фамилия	Имя	Отчество	Телефон	Дата рождения	Должность	Комментарий
aaa	aaa	aaa	79246284302	20.11.2001	aaa	  
Абакумов	Гайфулла	Хабибович	79118254862	14.07.1992	Аппаратный косметолог	  
Абдуллин	Шамиль	Абдуллаевич	79304077485	06.06.1973	Стилист-парикмахер	  
Абдуллин	Наум	Миннегалиевич	79817675328	28.01.1982	Косметолог	  
Абрамов	Зарифулла	Ананьевич	79173632090	24.07.1994	Косметолог-эстетист	  
Абрамов	Минулла	Мавлютович	79700782161	19.07.1991	Стилист-имиджмейкер	  
Абрамов	Али	Геннадиевич	79460642599	27.03.1998	Ассистент Стилиста-имиджмейкера	  
Абрамов	Азарий	Панфилович	79048703623	14.05.1973	Парикмахер-колорист	  
Абрамова	Ирма	Валерьевна	79654335071	21.07.1982	Стилист-имиджмейкер	  
Абрамова	Гертруда	Валерьевна	79484874913	20.01.1971	Мастер по наращиванию ресниц	  

1 2 3 4 5 6 7 8 9 10 ... > >>

Страница: 1 из 251

Рисунок 26 – Добавление «Employee»

Результат теста №6

Редактирование

Сотрудник

Фамилия
aaa1

Имя
aaa1

Отчество
aaa1

Телефон
79246284306

День рождения
20.11.2001

Должность
aaa1

Комментарий

[Сохранить](#) [Вернуться к списку](#)

Сотрудники

[Добавить](#) [Поиск](#) [Вернуться к полному списку](#)

Фамилия	Имя	Отчество	Телефон	Дата рождения	Должность	Комментарий
aaa1	aaa1	aaa1	79246284306	20.11.2001	aaa1	
Абакумов	Гайфулла	Хабирович	79118254862	14.07.1992	Аппаратный косметолог	
Абдуллин	Шамиль	Абдуллаевич	79304077485	06.06.1973	Стилист-парикмахер	
Абдуллин	Наум	Миннегалиевич	79817675328	28.01.1982	Косметолог	

Рисунок 27 – Редактирование «Employee»

Результат теста №7

Информация

Сотрудник

Фамилия
aaa1

Имя
aaa1

Отчество
aaa1

Телефон
79246284306

День рождения
20.11.2001

Должность
aaa1

Комментарий

[Редактировать](#) [Удалить](#) [Вернуться к списку](#)

Сотрудники

[Добавить](#) [Поиск](#) [Вернуться к полному списку](#)

Фамилия	Имя	Отчество	Телефон	Дата рождения	Должность	Комментарий
aaa1	aaa1	aaa1	79246284306	20.11.2001	aaa1	
Абакумов	Гайфулла	Хабирович	79118254862	14.07.1992	Аппаратный косметолог	
Абдуллин	Шамиль	Абдуллаевич	79304077485	06.06.1973	Стилист-парикмахер	
Абдуллин	Наум	Миннегалиевич	79817675328	28.01.1982	Косметолог	

Рисунок 28 – Просмотр «Employee»

Результат теста №8

Удаление

Сотрудник

Фамилия

aaa1

Имя

aaa1

Отчество

aaa1

Телефон

79246284306

День рождения

20.11.2001

Должность

aaa1

Комментарий

Вы уверены что хотите удалить этого сотрудника?

Удалить

[Вернуться к списку](#)



Сотрудники

[Добавить](#)

Фамилия/Имя/Отчество//

[Поиск](#)

[Вернуться к полному списку](#)

Фамилия	Имя	Отчество	Телефон	Дата рождения	Должность	Комментарий
Абакумов	Гайфулла	Хабибович	79118254862	14.07.1992	Аппаратный косметолог	И П У
Абдуллин	Шамиль	Абдуллаевич	79304077485	06.06.1973	Стилист-парикмахер	И П У
Абдуллин	Наум	Миннегалиевич	79817675328	28.01.1982	Косметолог	И П У
Абрамов	Зарифулла	Ананьевич	79173632090	24.07.1994	Косметолог-эстетист	И П У

Рисунок 29 – Удаление «Employee»

Результат теста №9

Создание

Клиент

Фамилия

aaa

Имя

aaa

Отчество

aaa

День рождения

20.11.2001

Email

komog-y-v@yandex.ru

Телефон

+79246284301

Комментарий

[Сохранить](#)

[Вернуться к списку](#)



Клиенты

[Добавить](#)

Фамилия/Имя/Отчество//

[Поиск](#)

[Вернуться к полному списку](#)

Фамилия	Имя	Отчество	Дата рождения	Телефон	Почта	Комментарий
aaa	aaa	aaa	20.11.2001	79246284301	komog-y-v@yandex.ru	И П У
Абакумов	Ахметия	Шафигуллович	05.03.2005	79469922722	yDAgtA@mail.ru	И П У
Абакумов	Иосиф	Пиманович	09.09.1981	79955568894	Pzutlxmvg@yandex.ru	И П У
Абакумов	Азат	Зиганшинович	01.12.1978	79498006478	UpMS@rambler.ru	И П У
Абакумов	Станислав	Фатеевич	20.05.1973	79688563010	hhKCR575qaF2Fs@yandex.ru	И П У
Абакумов	Шалва	Евграфович	13.06.1995	79662491480	3NO6j_s@gmail.com	И П У
Абакумов	Мубарак	Миннихметович	08.05.2003	79713496876	TjIm@yandex.ru	И П У
Абакумов	Муртаза	Сафронович	23.08.1982	79133148159	gGYtGu@yandex.ru	И П У
Абакумов	Курбан	Гарифуллович	11.12.2003	79395626206	s0RW.h_KYjGF4@list.ru	И П У
Абакумов	Рахим	Мухаметгалиевич	22.02.1990	79291130133	51p2@gmail.com	И П У

1 2 3 4 5 6 7 8 9 10 ... > >> Страница: 1 из 10001

Рисунок 30 – Добавление «Clients»

Результат теста №10

Редактирование

Клиент

Фамилия

Имя

Отчество

День рождения

Email

Телефон

Комментарий

[Сохранить](#) [Вернуться к списку](#)

Редактирование

Клиент

Фамилия

Имя

Отчество

День рождения

Email

Телефон

Комментарий

[Сохранить](#) [Вернуться к списку](#)

Клиенты

[Добавить](#) [Поиск](#) [Вернуться к полному списку](#)

Фамилия	Имя	Отчество	Дата рождения	Телефон	Почта	Комментарий
aaa1	aaa1	aaa1	21.11.2001	79246284323	komog-y-v1@yandex.ru	? ✎ ✖
Абакумов	Иосиф	Пиманович	09.09.1981	79955568894	PzutiXnvq@yandex.ru	? ✎ ✖
Абакумов	Шалва	Евграфович	13.06.1995	79662491480	3NO6j_s@gmail.com	? ✎ ✖

Рисунок 31 – Изменение «Clients»

Результат теста №11

Информация

Клиент

Фамилия
aaa

Имя
aaa

Отчество
aaa

День рождения
20.11.2001

Email
komog-y-v@yandex.ru

Телефон
79246284301

[✎ Редактировать](#) [✖ Удалить](#) [Вернуться к списку](#)

Клиенты

[Добавить](#) [Поиск](#) [Вернуться к полному списку](#)

Фамилия	Имя	Отчество	Дата рождения	Телефон	Почта	Комментарий
aaa	aaa	aaa	20.11.2001	79246284301	komog-y-v@yandex.ru	? ✎ ✖
Абакумов	Ахметия	Шафигулович	05.03.2005	79469922722	yDartA@mail.ru	? ✎ ✖
Абакумов	Иосиф	Пиманович	09.09.1981	79955568894	PzutiXnvq@yandex.ru	? ✎ ✖
Абакумов	Азат	Зиганшинович	01.12.1978	79498006478	UpMS@rambler.ru	? ✎ ✖
Абакумов	Станислав	Фатеевич	20.05.1973	79688563010	hhKCR57SqaE2Fs@yandex.ru	? ✎ ✖
Абакумов	Шалва	Евграфович	13.06.1995	79662491480	3NO6j_s@gmail.com	? ✎ ✖
Абакумов	Мубарак	Минияхметович	08.05.2003	79713496876	TiIm@yandex.ru	? ✎ ✖
Абакумов	Муртаза	Сафронович	23.08.1982	79133148159	gGYfGu@yandex.ru	? ✎ ✖
Абакумов	Курбан	Гарифуллович	11.12.2003	79395626206	s0BW.h.KYjG54@list.ru	? ✎ ✖
Абакумов	Рахим	Мухаметгалиевич	22.02.1990	79291130133	51p2@gmail.com	? ✎ ✖

1 2 3 4 5 6 7 8 9 10 ... » Страница: 1 из 10001

Рисунок 32 –Просмотр «Clients»

Результат теста №12

Удаление

Клиент

Фамилия

aaa1

Имя

aaa1

Отчество

aaa1

День рождения

21.11.2001

Email

komog-y-v1@yandex.ru

Телефон

79246284323

Комментарий

Вы уверены что хотите удалить этого клиента?

Удалить

[Вернуться к списку](#)



Клиенты

[Добавить](#)

Фамилия/Имя/Отчество//

[Поиск](#)

[Вернуться к полному списку](#)

Фамилия	Имя	Отчество	Дата рождения	Телефон	Почта	Комментарий
Абакумов	Иосиф	Пиманович	09.09.1981	79955568894	Pzut!xnvq@yandex.ru	И П У
Абакумов	Станислав	Фатеевич	20.05.1973	79688563010	hhKCR57SqaE2Fs@yandex.ru	И П У
Абакумов	Мубарак	Миннихметович	08.05.2003	79713496876	TiIm@yandex.ru	И П У

Рисунок 33 – Удаление «Clients»

Результат теста №13

Создание

Запись

Услуга

Снятие ресниц 300

Сотрудник

Журавлева Ульяна Самсоновна Ко

Клиент

Фамилия

Комогорцева

Имя

Юлия

Отчество

Владимировна

Дата

18.01.2023

Время

17:54

Комментарий

[Сохранить](#)

[Вернуться к списку](#)

Записи

[Добавить](#)

[Вернуться к полному списку](#)

Дата	Время	Клиент	Сотрудник	Услуга	
18.01.2023	17:54:00	Комогорцева Юлия	Журавлева Ульяна	Снятие ресниц	И П У
01.10.2022	00:04:01.4133333	Бураков Эрик	Бураков Оскар	Снятие ресниц	И П У
01.10.2022	19:25:01.8700000	Дорофеев Кондратий	Васильева Параскева	Наращивание волос Южно-русские волосы (цена за прядь с учетом стоимости работы) 30 см	И П У
01.10.2022	03:57:06.9133333	Петрушин Киям	Юрьев Мидхат	Наращивание волос Южно-русские волосы (цена за прядь с учетом стоимости работы) 30 см	И П У

Рисунок 34 – Добавление «Record»

Результат теста №14

Редактирование

Запись

Услуга
Снятие ресниц 300

Сотрудник
Журавлева Ульяна Самсоновна Ко

Клиент
Фамилия
Комогорцева

Имя
Юлия

Отчество
Владимировна

Дата
18.01.2023

Время
17:54:00

Комментарий

Сохранить

[Вернуться к списку](#)

Редактирование

Запись

Услуга
Наращивание ресниц 3D 1800

Сотрудник
Матвеева Ирина Маркияновна Ко

Клиент
Фамилия
Комогорцева

Имя
Юлия

Отчество
Владимировна

Дата
18.01.2023

Время
17:54:00

Комментарий

Сохранить

[Вернуться к списку](#)

Записи

[Добавить](#) [Вернуться к полному списку](#)













Дата	Время	Клиент	Сотрудник	Услуга	
19.01.2023	18:00:00	Комогорцева Юлия	Матвеева Ирина	Наращивание ресниц 3D	  
01.10.2022	00:04:01.4133333	Бураков Эрик	Бурлаков Оскар	Снятие ресниц	  
01.10.2022	19:25:01.8700000	Дорофеев Кондратий	Васильева Параскева	Наращивание волос Южно-русские волосы (цена за прядь с учетом стоимости работы) 30 см	  
01.10.2022	03:57:06.9133333	Петовшин Киям	Юрьев Мидхат	Наращивание волос Южно-русские волосы (цена за прядь с учетом стоимости работы) 30 см	  

Рисунок 35 – Изменение «Record»

Результат теста №15

Информация

Запись

Дата
18.01.2023

Время
17:54:00

Комментарий

Фамилия
Комогорцева

Фамилия
Журавлева

Название
Снятие ресниц

[Редактировать](#)

[Удалить](#)

[Вернуться к списку](#)

Записи

[Добавить](#) [Вернуться к полному списку](#)













Дата	Время	Клиент	Сотрудник	Услуга	
18.01.2023	17:54:00	Комогорцева Юлия	Журавлева Ульяна	Снятие ресниц	  
01.10.2022	00:04:01.4133333	Бураков Эрик	Бурлаков Оскар	Снятие ресниц	  
01.10.2022	19:25:01.8700000	Дорофеев Кондратий	Васильева Параскева	Наращивание волос Южно-русские волосы (цена за прядь с учетом стоимости работы) 30 см	  
01.10.2022	03:57:06.9133333	Петрушин Киям	Юрьев Мидхат	Наращивание волос Южно-русские волосы (цена за прядь с учетом стоимости работы) 30 см	  

Рисунок 36 – Просмотр «Record»

Результат теста №16

Удаление

Запись

Дата

19.01.2023

Время

18:00:00

Комментарий**Фамилия**

Комогорцева

Фамилия

Матвеева

Название

Наращивание ресниц 3D

Вы уверены что хотите удалить эту запись?

Удалить

[Вернуться к списку](#)



Записи

[Добавить](#)

[Вернуться к полному списку](#)










Дата	Время	Клиент	Сотрудник	Услуга	
01.10.2022	00:04:01.4133333	Бураков Эрик	Бурлаков Оскар	Снятие ресниц	  
01.10.2022	19:25:01.8700000	Дорофеев Кондратий	Васильева Параскева	Наращивание волос Южно-русские волосы (цена за прядь с учетом стоимости работы) 30 см	  
01.10.2022	03:57:06.9133333	Петрушин Киям	Юрьев Мидхат	Наращивание волос Южно-русские волосы (цена за прядь с учетом стоимости работы) 30 см	  

Рисунок 37 – Удаление «Record»

Результат теста №17

Создание

Работа

Услуга

Счастье для бровей 700

Сотрудник

Журавлева Ульяна Самсоновна Ко

Клиент**Фамилия**

Комогорцева

Имя

Юлия

Отчество

Владимировна

Дата

31.12.2021

Время

10:30

Доход

700

Комментарий

[Сохранить](#)































[Вернуться к списку](#)



Работы

[Добавить](#)

[Вернуться к полному списку](#)

Дата	Время	Доход	Клиент	Сотрудник	Услуга	
31.12.2021	10:30:00	700	Комогорцева Юлия	Журавлева Ульяна	Счастье для бровей	  
23.12.2021	15:20:00	1000	Комогорцева Юлия	Ушаков Ахат	Педикюр женский классический	  
17.12.2021	12:47:00	1000	Комогорцева Юлия	Ушаков Ахат	Педикюр женский классический	  
14.12.2021	11:03:00	1000	Сафиуллин Мартын	Ушаков Ахат	Педикюр женский классический	  
14.12.2021	23:50:00	1200	Комогорцева Юлия	Катков Камал	Стрижка короткие волосы	  
13.12.2021	10:29:00	1200	Комогорцева Юлия	Бородин Натан	Педикюр женский классический	  
09.12.2021	15:23:00	100	Сафиуллин Мартын	Ушаков Ахат	Педикюр женский классический	  
01.10.2021	18:36:49.0333333	800	Киреева Халида	Белякова Евгения	Лазерная эпиляция Классическое бикини	  
01.10.2021	22:43:53.2333333	2200	Горшков Сафрон	Бережной Адели	Эпиляция(сахар/воск) Комплекс 3 (бикини глубокое, ноги полностью, руки до локтя, подмышки)	  
01.10.2021	20:04:01.9800000	1000	Сорокин Гариф	Сахаров Шарип	Комплексный уход с использованием профессиональной косметики против угрей и прыщей "Антиакне"	  

1 2 3 4 5 6 7 8 9 10 ... » >>

Страница: 1 из 146428

Рисунок 38 – Добавление «Work»

Результат теста №18

Информация

Работа

Дата
31.12.2021

Время
10:30:00

Доход
700

Комментарий
Счастье для бровей

Фамилия
Журавлева

Название
Счастье для бровей

[✎ Редактировать](#)
[✖ Удалить](#)
[🔍 Вернуться к списку](#)

Работы

[Добавить](#)
[Вернуться к полному списку](#)

Дата	Время	Доход	Клиент	Сотрудник	Услуга	
31.12.2021	10:30:00	700	Комогорцева Юлия	Журавлева Ульяна	Счастье для бровей	🕒 ✎ ✖
23.12.2021	15:20:00	1000	Комогорцева Юлия	Ушаков Ахат	Педикюр женский классический	🕒 ✎ ✖
17.12.2021	12:47:00	1000	Комогорцева Юлия	Ушаков Ахат	Педикюр женский классический	🕒 ✎ ✖
14.12.2021	11:03:00	1000	Сафиуллин Мартын	Ушаков Ахат	Педикюр женский классический	🕒 ✎ ✖
14.12.2021	23:50:00	1200	Комогорцева Юлия	Катков Камал	Стрижка короткие волосы	🕒 ✎ ✖
13.12.2021	10:29:00	1200	Комогорцева Юлия	Бородин Натан	Педикюр женский классический	🕒 ✎ ✖
09.12.2021	15:23:00	100	Сафиуллин Мартын	Ушаков Ахат	Педикюр женский классический	🕒 ✎ ✖
01.10.2021	18:36:49.0333333	800	Киреева Халида	Белякова Евгения	Лазерная эпиляция Классическое бикини	🕒 ✎ ✖
01.10.2021	22:43:53.2333333	2200	Горшков Сафрон	Бережной Адели	Эпиляция(сахар/воск) Комплекс 3 (бикини глубокое, ноги полностью, руки до локтя, подмышки)	🕒 ✎ ✖
01.10.2021	20:04:01.9800000	1000	Сорокин Гариф	Сахаров Шарип	Комплексный уход с использованием профессиональной косметики против угрей и прыщей "Антиакне"	🕒 ✎ ✖

[1](#)
[2](#)
[3](#)
[4](#)
[5](#)
[6](#)
[7](#)
[8](#)
[9](#)
[10](#)
[...](#)
[»](#)

Страница: 1 из 146428

Рисунок 39 – Просмотр «Work»

Результат теста №19

Редактирование

Работа

Услуга
Скрабирование бровей 100

Сотрудник
Матвеева Ирина Маркиановна Ко

Клиент
Фамилия
Комогорцева

Имя
Юлия

Отчество
Владимировна

Дата
31.12.2021
01.01.2022

Время
10:30:00
10:00

Доход
150

Комментарий

[Сохранить](#)
[🔍 Вернуться к списку](#)

Работы

[Добавить](#)
[Вернуться к полному списку](#)

Дата	Время	Доход	Клиент	Сотрудник	Услуга	
01.01.2022	10:00:00	150	Комогорцева Юлия	Матвеева Ирина	Скрабирование бровей	🕒 ✎ ✖
23.12.2021	15:20:00	1000	Комогорцева Юлия	Ушаков Ахат	Педикюр женский классический	🕒 ✎ ✖
17.12.2021	12:47:00	1000	Комогорцева Юлия	Ушаков Ахат	Педикюр женский классический	🕒 ✎ ✖

Рисунок 40 – Изменение «Work»

Результат теста №20

Удаление

Работа

Дата

01.01.2022

Время

10:00:00

Доход

150

Комментарий**Фамилия**

Комогорцева

Фамилия

Матвеева

Название

Скрабирование бровей

Вы уверены что хотите удалить эту работу?

Удалить

[Вернуться к списку](#)



Работы

[Добавить](#)

[Вернуться к полному списку](#)

Дата	Время	Доход	Клиент	Сотрудник	Услуга	
23.12.2021	15:20:00	1000	Комогорцева Юлия	Ушаков Ахат	Педикюр женский классический	И П У
17.12.2021	12:47:00	1000	Комогорцева Юлия	Ушаков Ахат	Педикюр женский классический	И П У
14.12.2021	11:03:00	1000	Сафиуллин Мартын	Ушаков Ахат	Педикюр женский классический	И П У
14.12.2021	23:50:00	1200	Комогорцева Юлия	Катков Камал	Стрижка короткие волосы	И П У
13.12.2021	10:29:00	1200	Комогорцева Юлия	Бородин Натан	Педикюр женский классический	И П У

Рисунок 41 –Удаление «Work»

Результат теста №21

Статистика по услугам

2020

[Поиск](#)

Название услуги	Количество
Наращивание ресниц Эффект "Лучики/Ким/Кайли" (доплата)	1659
Массаж 60 минут (на выбор)	1642
Окрашивание бровей при ламинировании	1642
Лазерная эпиляция Классическое бикини	1638
Счастье для бровей	1637
Удаление татуировки 10-15 см	1627
Маникюр с покрытием Vinylux/ OPI	1626
Укладка на брашинг/стайлер длина ниже лопаток	1624
Наращивание волос Южно-русские волосы (цена за прядь с учетом стоимости работы) 30 см	1623
Эпиляция(сахар/воск) Руки до локтя	1623

1 2 3 4 5 6 7 8 9 10 ... » >>

Страница: 1 из 14

Рисунок 42 – Результат выполнения запроса

Результат теста №22

Статистика по сотрудникам

6

Поиск

Имя сотрудника	Максимальный возраст клиента	Минимальный возраст клиента	Количество записанных клиентов
Любовь	19	54	11
Асланбек	15	49	9
Габдульбар	24	50	8
Ибрагим	17	53	8
Амбарцум	15	52	8
Габдрахман	14	55	8
Глеб	17	55	8
Хабибрахман	26	48	7
Миннибай	17	43	7
Белла	30	48	7

12345678910...»»

Страница: 1 из 18

Рисунок 43 – Результат выполнения запроса

Результат теста №23

Создание

Услуга

Название

Поле должно быть заполнено

Стоимость

Поле должно быть заполнено

Комментарий

Сохранить

Возврат к списку

Статистика по услугам

Год

Поиск

Вы ввели не число

Результатов не найдено

Создание

Сотрудник

Фамилия

Поле должно быть заполнено

Имя

Поле должно быть заполнено

Отчество

Поле должно быть заполнено

Телефон

Поле должно быть заполнено

День рождения

Поле должно быть заполнено

Должность

Поле должно быть заполнено

Комментарий

Сохранить

Возврат к списку

Создание

Клиент

Фамилия

Поле должно быть заполнено

Имя

Поле должно быть заполнено

Отчество

Поле должно быть заполнено

День рождения

Поле должно быть заполнено

Email

Поле должно быть заполнено

Телефон

Поле должно быть заполнено

Комментарий

Сохранить

Возврат к списку

Создание

Работа

Услуга

Педикюр женский классический 1

Сотрудник

Ушаков Ахат Акиндинович Подолс

Клиент

Фамилия

Имя

Отчество

Дата

Поле должно быть заполнено

Время

Поле должно быть заполнено

Доход

Поле должно быть заполнено

Комментарий

Сохранить

Возврат к списку

Создание

Запись

Услуга

Педикюр женский классический 1

Сотрудник

Ушаков Ахат Акиндинович Подолс

Клиент

Фамилия

Имя

Отчество

Дата

Поле должно быть заполнено

Время

Поле должно быть заполнено

Комментарий

Сохранить

Возврат к списку

Рисунок 44 – Контроль данных

Заключение

В процессе разработки данного приложения были получены и закреплены следующие навыки:

1. Проектирования и применения паттерна MVC в реализации информационной системы
2. Проектирования пользовательского интерфейса и компоновки элементов взаимодействия с пользователем
3. Создания, проектирования и работы с локальной базы данных с помощью СУБД Microsoft SQL Server
4. Программирование веб-приложения

Были решены следующие задачи:

1. Спроектирована локальная база данных
2. Спроектирована структурная схема функционирования моделей, контроллеров и представлений
3. Развернута модель данных с помощью фреймворка Entity Framework
4. Разработано веб-приложение для автоматизированного взаимодействия с клиентами
5. Тестирование приложения
6. Составлена документация по всей проделанной работе

Исходный код приложения опубликован в публичном репозитории на сервисе GitHub [6].

Цель курсового проекта достигнута.

Список использованных источников

1. Стасышин В. М., Стасышина Т. Л. Базы данных: Технологии доступа 2-е изд., испр. и доп. Учебное пособие для академического бакалавриата, 2019. 164 с. ISBN: 978-5-534-09888-4 Серия: Профессиональное образование. URL: <https://biblio-online.ru/viewer/bazy-dannyhtehnologii-dostupa-415342#page/1>.
2. Голицына О. Л. Базы данных: учебное пособие для вузов по направлению 230700 "Прикладная информатика" / О. Л. Голицына, Н. В. Максимов, И. И. Попов, 2012. 399 с.
3. Сосинская С. С. Использование языка С# в различных информационных технологиях: учебник для студентов вузов, обучающихся по направлению "Информационные системы и технологии" / Сосинская, 2014. 367 с.
4. Шубина М.А. Управление данными: учебное пособие для студентов направлений подготовки 09.03.02 и 09.04.02 «Информационные системы и технологии» / Шубина М.А, 2016. 132 с. ISBN: 978-5-9239-0832-9. URL: https://e.lanbook.com/book/74029#book_name.
5. Сосинская С.С. «Разработка клиента по технологии MVC»: Методические указания по выполнению курсового проекта» Электронный каталог кафедры вычислительной техники (дата обращения: 07.12.2021).
6. Репозиторий с исходным кодом проекта // GitHub URL: https://github.com/KomogortsevaYulia/CourseWork_Beauty1.git (дата обращения: 10.12.2021).
7. Введение в ASP.NET MVC // Особенности платформы ASP.NET MVC URL: <https://metanit.com/sharp/mvc/1.1.php> (дата обращения: 01.11.2021).