МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

«Южно-Уральский государственный университет»

(национальный исследовательский университет)

Факультет «Экономика и управление»

Кафедра «Информатика»

РАБОТА ПРОВЕРЕНА

Рецензент, _____

«19» июня 2016 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.т.н., с.н.с.

_____ /Б.М. Суховилов /

«19» июня 2016 г.

Реализация Web-приложения для автоматизации работы санатория

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ–230700.62.2016.356.ВКР

Руководитель, к.т.н., доцент

_____ / В.А. Конов /

«19» июня 2016 г.

Автор

студент группы ЭиУ-457

_____ / О.О. Барышев /

«18» июня 2016 г.

Нормоконтролер, доцент

_____ / Е.А. Конова /

«18» июня 2016 г.

Челябинск 2016

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Южно-Уральский государственный университет»

(национальный исследовательский университет)

Факультет «Экономика и управление»

Кафедра «Информатика»

Направление 230700.62 «Прикладная информатика»

УТВЕРЖДАЮ:

Зав.кафедрой

_____ / **Б.М. Суховилов/**

«26» апреля 2016г.

З А Д А Н И Е

на выпускную квалификационную работу студента

Барышева Олега Олеговича

1 Тема проекта Реализация Web-приложения для автоматизации работы санатория.

Утверждена приказом по университету от 15 апреля 2016 г. № 661

2 Срок сдачи студентом законченного проекта 07.06.2015

3 Исходные данные к проекту: перечень должностей организации, список подразделений, штатное расписание, список должностных обязанностей

4 Перечень вопросов, подлежащих разработке: анализ предметной области функционирования будущей системы, построение алгоритмов работы приложения, конвертация базы данных, проектирование интерфейса веб-страниц, программи-

рование основного функционала веб-приложения, тестирование программного продукта.

5 Иллюстративный материал (плакаты, альбомы, раздаточный материал, макеты, электронные носители и др.)

Архитектура приложения, схема базы данных, страница должностные обязанности, ввод и корректировка должностных обязанностей, рассылка

Общее количество иллюстраций:

6 Дата выдачи задания: 15.03.2016

Руководитель Конов Вячеслав Анатольевич

Задание принял к исполнению Барышев Олег Олегович

КАЛЕНДАРНЫЙ ПЛАН

№ п/п	Наименование этапов выпускной квалификационной работы	Срок выполнения этапов работы	Отметка руководителя о выполнении
1	Анализ существующих систем.	24.02.2016	В.Хам
2	Описание структуры программы. Контекстная диаграмма и диаграмма первого уровня	9.03.2016	В.Хам
3	Разработка структуры БД	30.03.2016	В.Хам
4	Разработка интерфейса	11.05.2016	В.Хам
5	Реализация проекта	25.05.2016	В.Хам
6	Тестирование и пример работы программы	30.05.2016	В.Хам
7	Оформление пояснительной записки	14.06.2016	В.Хам

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Факультет «Экономика и управление»
Кафедра «Информатика»

РАБОТА ПРОВЕРЕНА
Рецензент, _____

«_____» _____2016 г.

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой, д.т.н.
_____/Б.М. Суховилов /
«____» _____2016 г.

Реализация Web-приложения для автоматизации работы санатория

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ–230700.62.2016.356.ВКР

Руководитель, к.т.н., доцент
_____/В.А. Конов /
«____» _____2016 г.
Автор
студент группы ЭиУ-457
_____/О.О. Барышев /
«____» _____2016 г.
Нормоконтролер, доцент
_____/Е.А. Конова /
«____» _____2016 г.

Челябинск 2016

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Факультет «Экономика и управление»
Кафедра «Информатика»
Направление 230700.62 «Прикладная информатика»

УТВЕРЖДАЮ:
Зав.кафедрой
____Б.М.Суховилов/
«__»_____2016г.

З А Д А Н И Е

на выпускную квалификационную работу студента

Барышева Олега Олеговича

1 Тема проекта Реализация Web-приложения для автоматизации работы санатория.

Утверждена приказом по университету от 15 апреля 2016 г. № 661

2 Срок сдачи студентом законченного проекта 07.06.2015
3 Исходные данные к проекту: перечень должностей организации, список подразделений, штатное расписание, список должностных обязанностей
4 Перечень вопросов, подлежащих разработке: анализ предметной области функционирования будущей системы, построение алгоритмов работы приложения, конвертация базы данных, проектирование интерфейса веб-страниц, программирование основного функционала веб-приложения, тестирование программного продукта.
5 Иллюстративный материал (плакаты, альбомы, раздаточный материал, макеты, электронные носители и др.)
Архитектура приложения, схема базы данных, страница должностные обязанности, ввод и корректировка должностных обязанностей, рассылка

Общее количество иллюстраций:
6 Дата выдачи задания: 15.03.2016

Руководитель Конов Вячеслав Анатольевич

Задание принял к исполнению Барышев Олег Олегович

КАЛЕНДАРНЫЙ ПЛАН

№ п/п	Наименование этапов выпускной квалификационной работы	Срок выполнения этапов работы	Отметка руководителя о выполнении
1	Анализ существующих систем.		
2	Описание структуры программы. Контекстная диаграмма и диаграмма первого уровня		
3	Разработка структуры БД		
4	Разработка интерфейса		
5	Реализация проекта		
6	Тестирование и пример работы программы		
7	Оформление пояснительной записки		

АННОТАЦИЯ

Барышев О.О. Реализация Web-приложения для автоматизации работы санатория – Челябинск: ЮУрГУ, ЭиУ-457, 2016. – 63 стр., библиогр. список – 4 наим. 5 л. плакатов ф. А1.

Выпускная квалификационная работа выполнена с целью разработки Web-приложения, предоставляющего возможность работы с базой данных санатория, автоматизирующего учет клиентов, проживания клиентов, оказание медицинских услуг, учет врачей, номеров.

В работе обоснована актуальность выбранной темы, сформулирована цель и задачи, проанализированы аналогичные существующие Web-приложения.

Разработано Web-приложение для автоматизации работы санатория, которое успешно отлажено и протестировано.

ОГЛАВЛЕНИЕ

КАЛЕНДАРНЫЙ ПЛАН 7
АННОТАЦИЯ 8
ОГЛАВЛЕНИЕ 9
ВВЕДЕНИЕ 10
1. Постановка задачи 11
1.1 Анализ существующих систем 11
1.2 Выбор инструментария для выполнения работы 13
Выводы по разделу 16
2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА 17
2.1 Создание модели информационной системы 17
2.2 Проектирование и применение базы данных 20
2.3 Описание таблиц базы данных 21
Выводы по разделу 25
3 Описание работы приложения 26
3.1 Работа в режиме обычного пользователя 26
3.2 Работа в режиме администратора 31
Выводы по разделу. 42
Приложение А Код страницы «Авторизация» 43
Приложение Б Код страницы «Фотогалерея» 45
Приложение В Код страницы «Отчет: Специализация врачей в санатории» 46
Приложение Г Код страницы «Отчет: Проживание клиентов» 47
Приложение Д Код страницы «Добавление сотрудника» 49
Приложение Е Код страницы «Добавление заявки» 51
Приложение Ж Код страницы «Добавление номера» 54
Приложение З Код страницы «Добавление услуги» 56
ЗАКЛЮЧЕНИЕ 58
БИБЛИОГРАФИЧЕСКИЙ СПИСОК 59

ВВЕДЕНИЕ

Основой для учета, контроля и планирования служат всевозможные картотеки, регистрационные журналы, списки и т.д. Они постепенно накапливаются и обновляются. При большом объеме информации поиск и обобщение необходимых сведений, осуществляемых вручную, представляют собой довольно трудоемкий процесс. Поэтому и возникла необходимость создания баз данных.

Без баз данных сегодня невозможно представить работу большинства финансовых, промышленных, торговых и прочих организаций. Они позволяют структурировать, хранить и извлекать информацию оптимальным для пользователя способом.

База данных (БД) – это компьютеризованная система, предназначенная для хранения информации и предоставления ее по требованию.

Задача базы данных состоит в хранении всех представляющих интерес данных в одном или нескольких местах, причем таким способом, который заведомо исключает ненужную избыточность. Создание баз данных преследует две основные цели: понизить избыточность данных и повысить их надежность.

Целью данной дипломной работы было проектирование и организация базы данных санатория и сопровождение ее программным продуктом, Web-приложением.

1. ПОСТАНОВКА ЗАДАЧИ

Используя возможности MS SQL Server, реализовать базу данных, хранящую в себе информацию о клиентах санатория, врачах, услугах, номерах, меню.

Разработать клиент-серверное Web-приложение в среде MS Visual Studio, используя базу данных SQL Server.

Основные требования к приложению:

- пользователь (Администратор) может вносить, изменять и удалять данные;
- обычный пользователь может работать с приложением только в режиме просмотра, может оставлять заявку на бронирование онлайн.

1.1 Анализ существующих систем

Анализ существующих систем приведен в таблице 1.

Таблица 1 – Анализ существующих систем

Санаторий/ Свойства	**Мой санаторий**	**Санаторий «Урал»**	**Санаторий «Таныр»**	**Санаторий «Дальняя дача»**
На сайте имеется кнопка авторизации для администраторов	+	—	—	—
Разграничение прав доступа	+	—	—	—
Возможность просмотра карты. Как добраться?	+	+	+	+
Описание номера и фото номера	+	Имеется только общая галерея.	+	+

Лечение + информация о сотрудниках	ЕСТЬ перечень услуг. Имеется информация о врачах + фото врача.	ЕСТЬ перечень услуг. НЕТ информации о врачах.	ЕСТЬ перечень услуг. НЕТ фото врача.	ЕСТЬ перечень услуг. Имеется информация о врачах + фото врача.
Бронирование на сайте	Есть возможность оставить заявку на бронирование. Данные сразу сохраняются в БД. Не придется вводить их заново.	Есть возможность оставить заявку на бронирование	Есть возможность оставить заявку на бронирование	При бронировании не указывается дата заезда, вид путевки, количество людей.
Фотогалерея	+	+	–	+
Положение строки меню	Горизонтальное.	Горизонтальное.	Горизонтальное.	Вертикальное.
Формирование отчетов.	+	+	+	+
Учет сотрудников	+	–	+	+
Учет базы врачей	+	+	+	+
Учет базы услуг	+	+	+	+
Учет номеров	+	+	+	+
Возможность работы с фотогалерей (Добавить/Удалить)	+	+	+	+

Как следует из таблицы, разрабатываемое Web-приложение отличается от других.

Имеет удобный и простой интерфейс для администратора, где он может работать с данными. Вести учет сотрудников, учет всех услуг, оказываемых в санатории, учет номеров.

Имеет более удобный и интуитивно понятный интерфейс для пользователя в отличие от других систем. Когда пользователь будет

бронировать номер на сайте, то он заполнит соответствующую форму, и эти данные уже будут храниться в базе данных. Таким образом, когда администратор захочет связаться с пользователем для уточнения заявки, то он уже будет иметь представление, что хочет пользователь и не придется вторично переспрашивать его.

1.2 Выбор инструментария для выполнения работы

Для разработки сайта используются технологии: HTML, CSS, XML, ASP.Net MVC, ADO.NET Entity Framework.

HTML – стандартный язык разметки документов во Всемирной паутине [1]. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами, полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

Язык HTML является приложением SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO 8879.

Язык XHTML является более строгим вариантом HTML, он следует всем ограничениям XML и, фактически, XHTML можно воспринимать как приложение языка XML к области разметки гипертекста.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием шифрования.

Все документы, содержащие разметку на языке HTML, имеют разрешением .html или .htm.

Для просмотра этих документов используются специальные приложения – браузеры.

Наиболее популярные браузеры на сегодняшний день:

- Google Chrome;
- Safari 9.1;

- Yandex Browser;
- Mozilla Firefox;
- Opera;
- Internet Explorer 11.

CSS – формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Преимущественно используется как средство описания, оформления внешнего вида Web-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL.

CSS используется создателями Web-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы от описания внешнего вида этой веб-страницы. Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

XML – рекомендованный Консорциумом Всемирной паутины (W3C) язык разметки. Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров. XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов программами и одновременно удобный для чтения и создания документов человеком, с подчёркиванием нацеленности на использование в Интернете. Язык называется расширяемым, поскольку он не фиксирует

разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка. Сочетание простого формального синтаксиса, удобства для человека, расширяемости, а также базирование на кодировках Юникод для представления содержания документов привело к широкому использованию как собственно XML, так и множества производных специализированных языков на базе XML в самых разнообразных программных средствах.

ASP.NET MVC Framework – фреймворк для создания веб-приложений, который реализует шаблон Model-view-controller.

Платформа ASP.NET MVC [2] базируется на взаимодействии трех компонентов: контроллера, модели и представления. Контроллер принимает запросы, обрабатывает пользовательский ввод, взаимодействует с моделью и представлением и возвращает пользователю результат обработки запроса.

Модель представляет слой, описывающий логику организации данных в приложении. Представление получает данные из контроллера и генерирует элементы пользовательского интерфейса для отображения информации.

ADO.NET Entity Framework(EF) – объектно-ориентированная технология доступа к данным, является object-relational mapping (ORM) решением для .NET Framework от Microsoft. Предоставляет возможность взаимодействия с объектами как посредством LINQ в виде LINQ to Entities, так и с использованием Entity SQL. Для облегчения построения web-решений используется как ADO.NET Data Services (Astoria), так и связка из Windows Communication Foundation и Windows Presentation Foundation, позволяющая строить многоуровневые приложения, реализуя один из шаблонов проектирования MVC, MVP или MVVM.

Выводы по разделу

Предъявлены требования к функциональности приложения, выполнен анализ аналогичных программных продуктов, на основе которого принято решение о разработке собственного приложения.

Выбраны технологии разработки: HTML, CSS, XML, ASP.Net MVC, ADO.NET Entity Framework.

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

2.1 Создание модели информационной системы

Контекстная диаграмма первого уровня приведена на рисунке 2.1

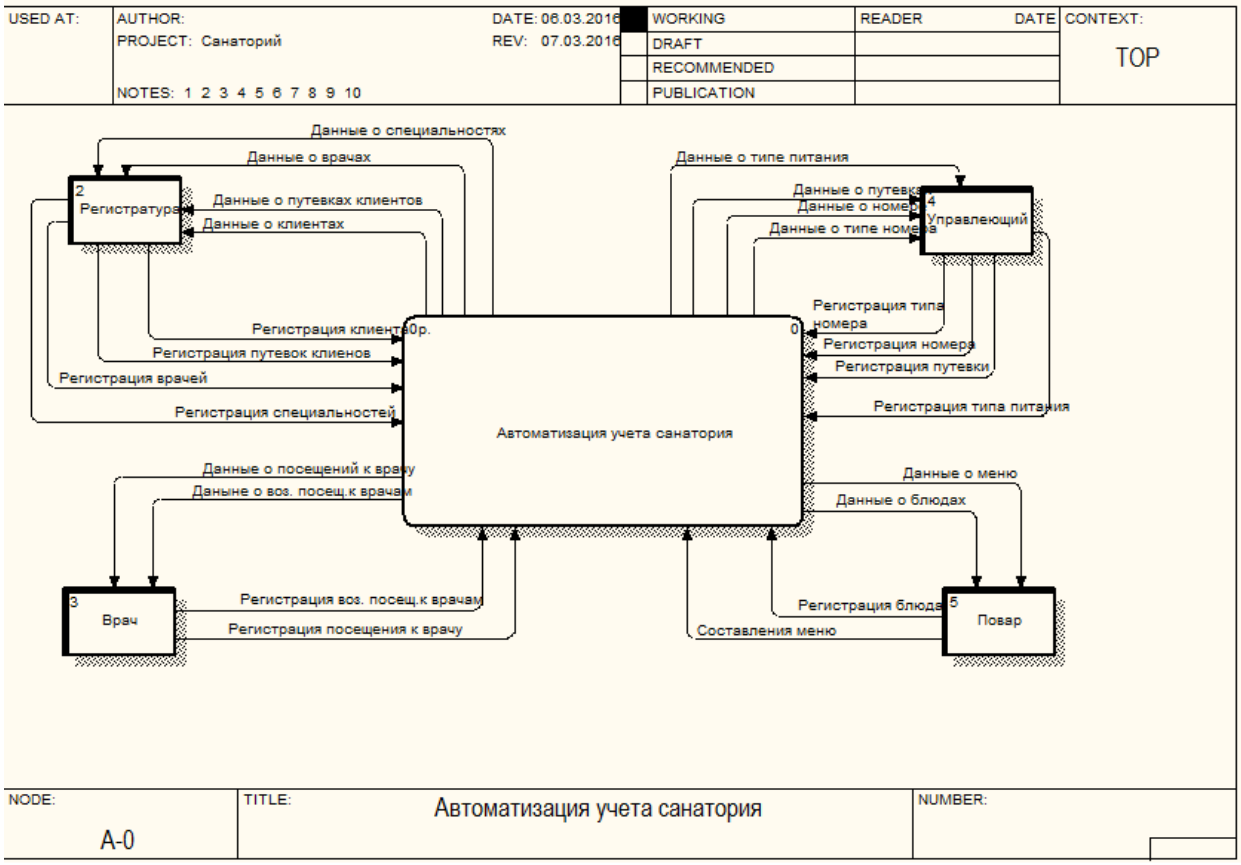

Рисунок. 2.1 – «Контекстная диаграмма IDEF0. Санаторий»

Для автоматизации работы санатория, используется четыре роли(Регистратура/Управляющий/Врач/Повар)

Контекстная диаграмма второго уровня приведена на рисунке 2.2

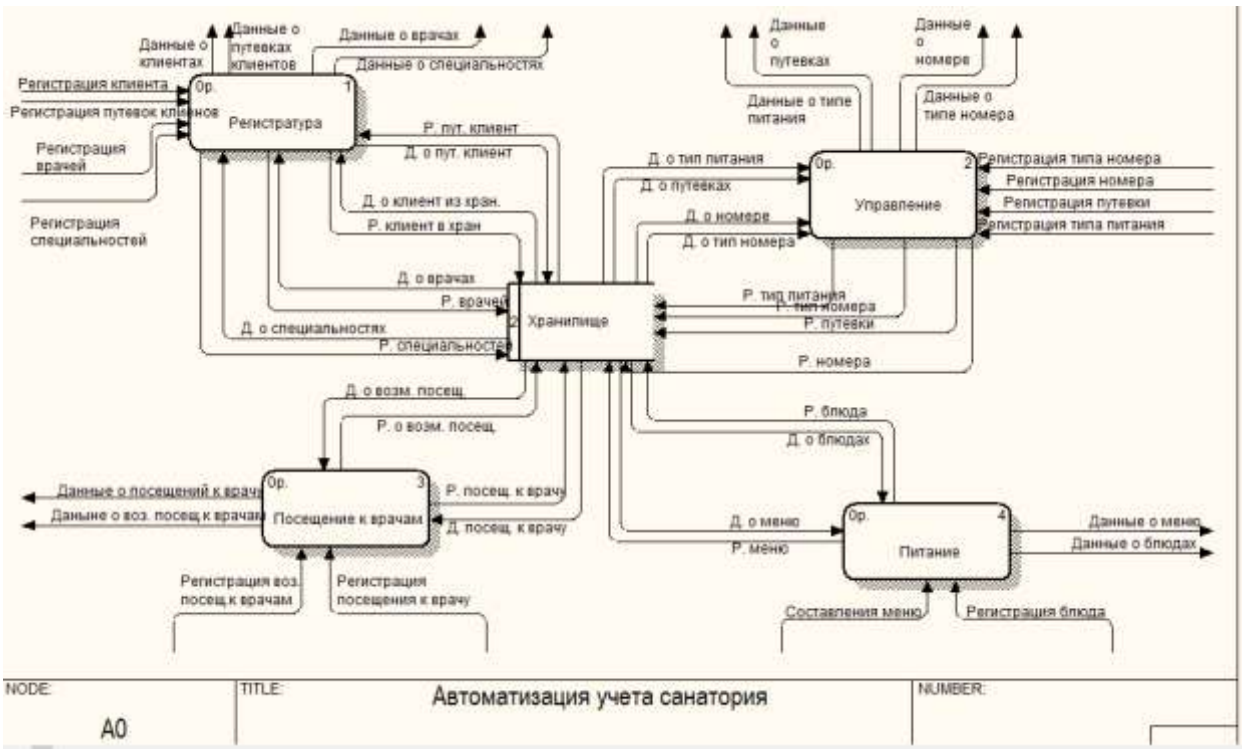

Рисунок. 2.2 – «Контекстная диаграмма IDEF0. Санаторий»

Для автоматизации работы санатория, используется четыре роли(Регистратура/Управление/Посещение к врачам/Питание)

Контекстная диаграмма третьего уровня приведена на рисунке 2.3

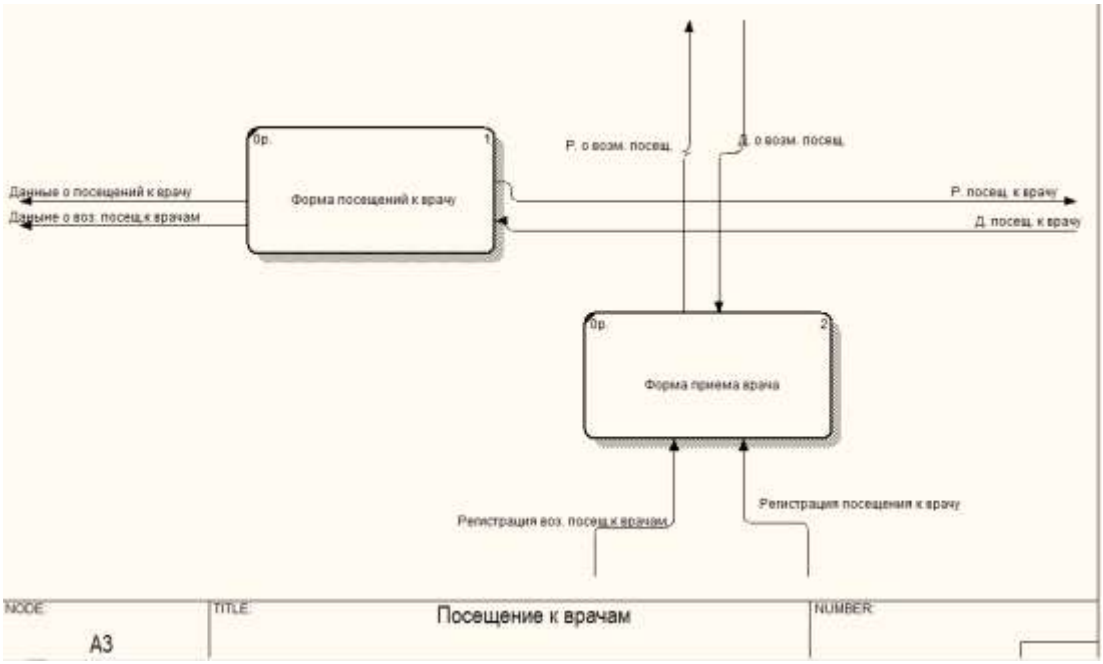

Рисунок. 2.3 – «Контекстная диаграмма IDEF0. Посещение к врачам»

Показан процесс регистрации на прием к врачу.

Контекстная диаграмма IDEF0. «Блюда» приведена на рисунке 2.4

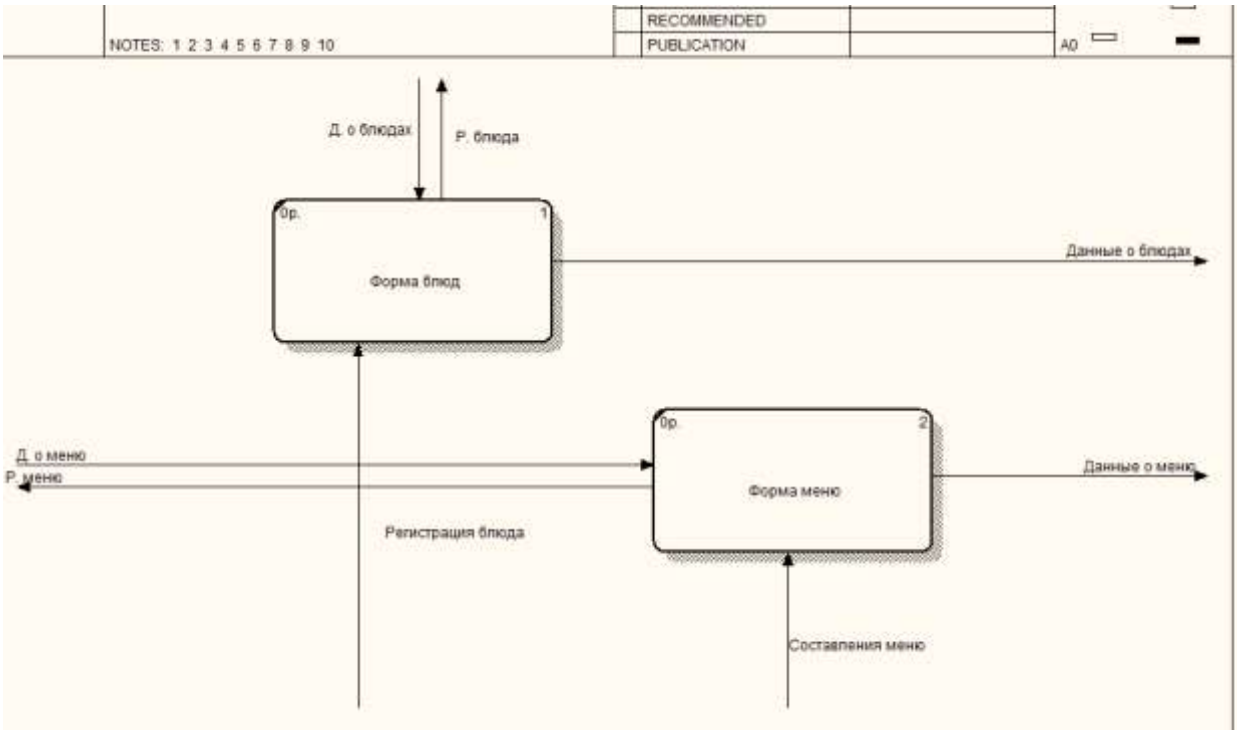

Рисунок. 2.4 – «Контекстная диаграмма IDEF0. Блюда»

Показан процесс составления блюд. Получение информации о блюдах.

Контекстная диаграмма IDEF0. «Управление» приведена на рисунке 2.5

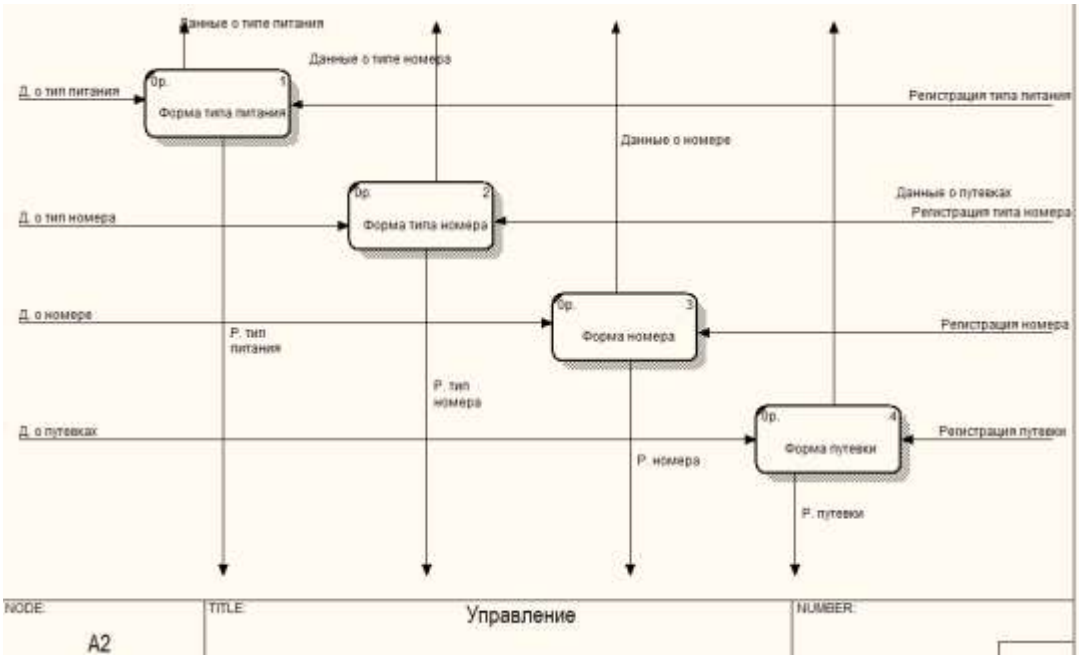

Рисунок. 2.5 – «Контекстная диаграмма IDEF0. Управление»

Контекстная диаграмма IDEF0. «Регистратура» приведена на рисунке 2.6

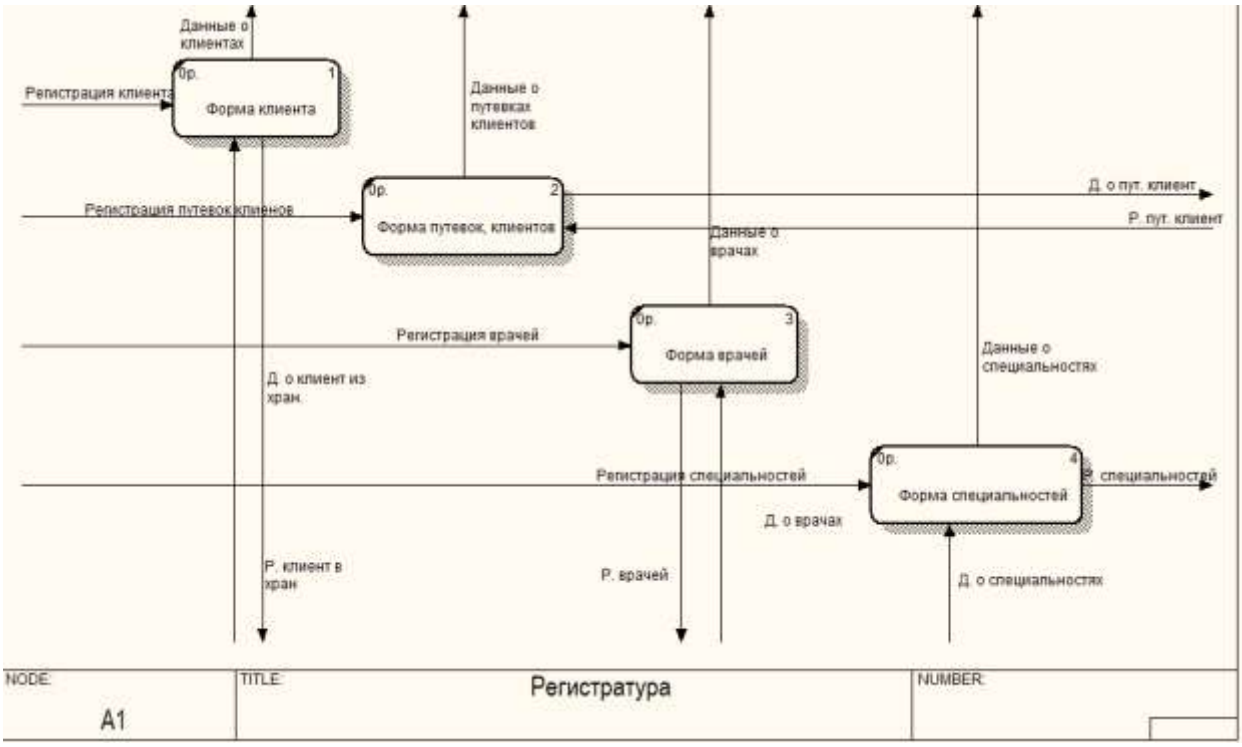

Рисунок. 2.6 – «Контекстная диаграмма IDEF0. Регистратура»

Используется для заполнения данных.

2.2 Проектирование и применение базы данных

Для хранения данных, используемых в приложении, применяется база данных SQL Server. Схема данных приведена на рисунке 2.7

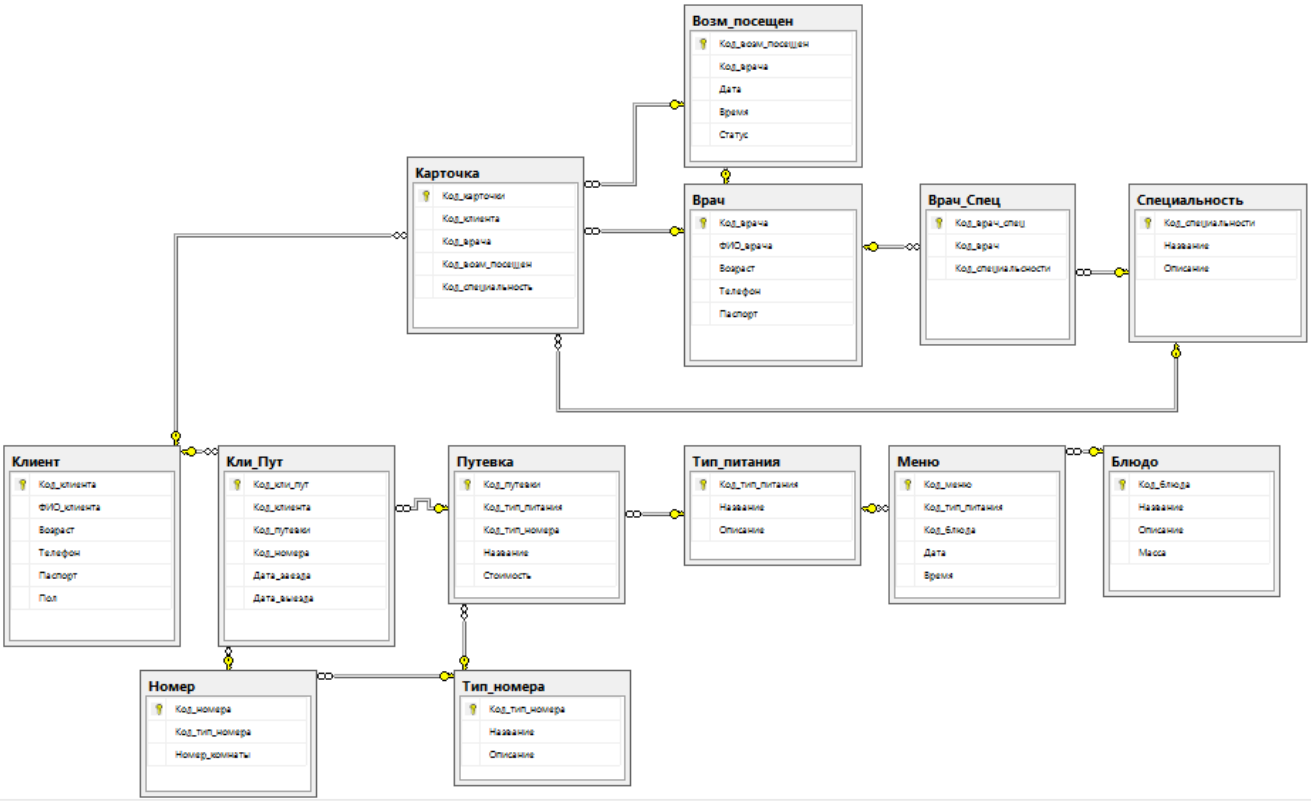

Рисунок. 2.7 – Схема данных

Описана схема данных для автоматизации работы санатория.

2.3 Описание таблиц базы данных

Для решения поставленной задачи созданы и использованы следующие таблицы:

- возможные посещения;
- карточка клиента;
- врач;
- специальность врача;
- клиент;
- покупка путевки;
- путевки;
- тип питания;
- меню;
- блюда;

- номер;
- тип номера.

Далее описаны структура и тип хранящихся данных каждой таблицы.

1. Таблица «Возможные посещения»

Структура таблицы «Возможные посещения» приведена на рисунке 2.8.

Имя столбца	Тип данных
Код_возм_посещен	int
Код_врача	int
Дата	date
Время	nvarchar(5)
Статус	int

Рисунок. 2.8 – таблица «Возможные посещения»

Хранение информации о возможных посещениях.

2. Структура таблицы «Карточка клиента» приведена на рисунке 2.9.

Имя столбца	Тип данных
Код_карточки	int
Код_клиента	int
Код_врача	int
Код_возм_посещен	int
Код_специальность	int

Рисунок. 2.9 – таблица «Карточка клиента»

Хранение информации о медицинских услугах клиента.

3. Структура таблицы «Врач» приведена на рисунке 2.10.

Имя столбца	Тип данных
Код_врача	int
ФИО_врача	nvarchar(50)
Возраст	date
Телефон	nvarchar(50)
Паспорт	nvarchar(50)

Рисунок. 2.10 – таблица «Врач»

Хранение информации о врачах.

4. Структура таблицы «Специальность врача» приведена на рисунке 2.11.

Имя столбца	Тип данных
Код_специальности	int
Название	nvarchar(50)
Описание	nvarchar(MAX)

Рисунок. 2.11 – таблица «Специальность врача»

Хранение информации о специальностях врачей.

5. Структура таблицы «Клиент» приведена на рисунке 2.12.

Имя столбца	Тип данных
Код_клиента	int
ФИО_клиента	nvarchar(50)
Возраст	date
Телефон	nvarchar(50)
Паспорт	nvarchar(50)
Пол	nvarchar(10)

Рисунок. 2.12 – таблица «Клиент»

Хранение информации о клиентах.

6. Структура таблицы «Покупка путевки» приведена на рисунке 2.13.

Имя столбца	Тип данных
Код_кли_пут	int
Код_клиента	int
Код_путевки	int
Код_номера	int
Дата_заезда	date
Дата_выезда	date

Рисунок. 2.13 – таблица «Покупка путевки»

Хранение информации о покупке путевки.

7. Структура таблицы «Путевки» приведена на рисунке 2.14.

Имя столбца	Тип данных
Код_путевки	int
Код_тип_питания	int
Код_тип_номера	int
Название	nvarchar(50)
Стоимость	money

Рисунок. 2.14 – таблица «Путевки»

Хранение информации о путевках.

8. Структура таблицы «Тип питания» приведена на рисунке 2.15.

Имя столбца	Тип данных
Код_тип_питания	int
Название	nvarchar(50)
Описание	nvarchar(50)

Рисунок. 2.15 – таблица «Тип питания»

Хранение информации о типе питания.

9. Структура таблицы «Меню» приведена на рисунке 2.16.

Имя столбца	Тип данных
Код_меню	int
Код_тип_питания	int
Код_блюда	int
Дата	nvarchar(50)
Время	nvarchar(50)

Рисунок. 2.16 – таблица «Меню»

Хранение информации о меню.

10. Структура таблицы «Блюдо» приведена на рисунке 2.17.

Имя столбца	Тип данных
Код_блюда	int
Название	nvarchar(50)
Описание	nvarchar(50)
Масса	nvarchar(50)

Рисунок. 2.17 – таблица «Блюдо»

Хранение информации о блюдах.

11. Структура таблицы «Номер» приведена на рисунке 2.18.

	Имя столбца	Тип данных
🔑	Код_номера	int
	Код_тип_номера	int
	Номер_комнаты	nvarchar(50)

Рисунок. 2.18 – таблица «Номер»

Хранение информации о номерах.

12. Структура таблицы «Тип номера» приведена на рисунке 2.19.

	Имя столбца	Тип данных
🔑	Код_тип_номера	int
	Название	nvarchar(50)
	Описание	nvarchar(50)

Рисунок. 2.19 – таблица «Тип номера»

Хранение информации о типах номеров.

Выводы по разделу

Разработаны контекстные диаграммы первого, второго, третьего уровней в среде BPWin. Разработана схема данных в среде MS SQL Server. Описаны таблицы базы данных.

3 ОПИСАНИЕ РАБОТЫ ПРИЛОЖЕНИЯ

3.1 Работа в режиме обычного пользователя

При входе на сайт как обычный пользователь, доступны следующие поля меню: «Главная», «Фотогалерея», «Номера», «Бронирование», «Лечение», «Персонал», «Контакты».

Главная страница

Главная страница Web-приложения представлена на рисунке 3.1.

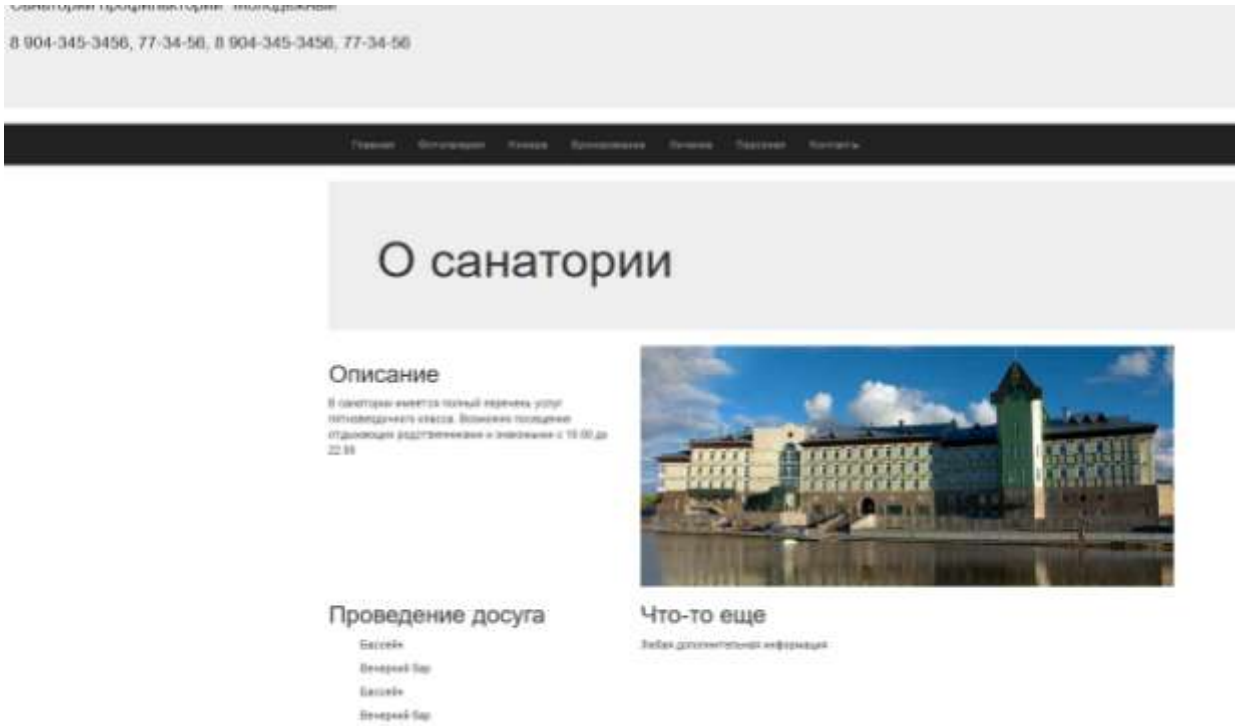

Рисунок. 3.1 – «Главная страница»

На главной странице можно просмотреть общую информацию о санатории.

Страница Web-приложения «Бронирование» представлена на рисунке 3.2.

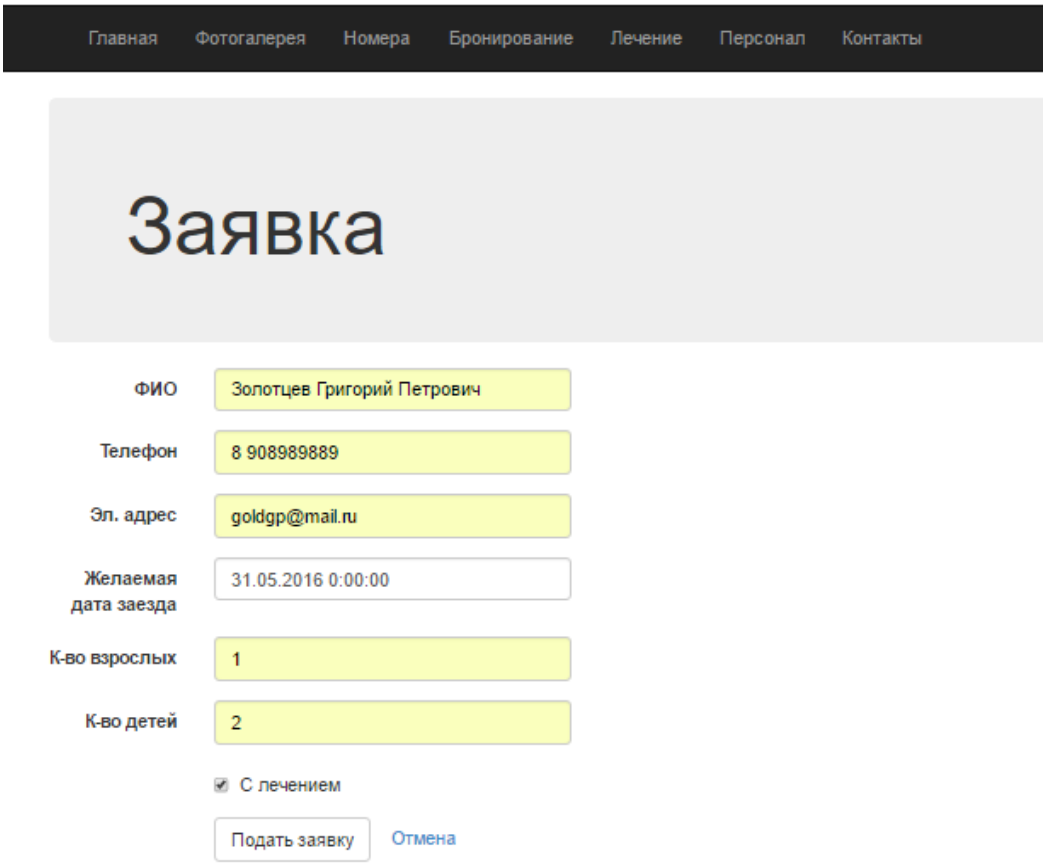

Рисунок. 3.2 – «Бронирование»

При переходе на панели меню в пункт «Бронирования», будет предложено заполнить следующие поля: Фамилия/Имя/Отчество, телефон, электронный адрес, предполагаемую дату заезда, и выбор самой путевки(с лечением/без лечения).

Страница «Принятие заявки» представлена на рисунке 3.3.

Фотогалерея Номера Бронирования Лечение Персонал Отчеты

Ваша заявка под номером 12 принята

ФИО	Золотцев Григорий Петрович
Телефон	8 908989889
Эл. адрес	goldgp@mail.ru
К-во взрослых	1
К-во детей	2
Желаемая дата заезда	31.05.2016 0:00:00
С лечением	☑

На главный экран | Добавить новую |

Отменить заявку

© 2016

Рисунок. 3.3 – «Принятие заявки»

Принятие заявки к обработке.

Страница Web-приложения «Фотогалерея» представлена на рисунке 3.4.

Главная Фотогалерея Номера Бронирование Лечение Персонал Контакты

Фотогалерея

© 2016

Рисунок. 3.4 – «Фотогалерея»

При переходе на пункт меню «Фотогалерея», будет представлена фотогалерея санатория.

Страница Web-приложения «Номера» представлена на рисунке 3.5.

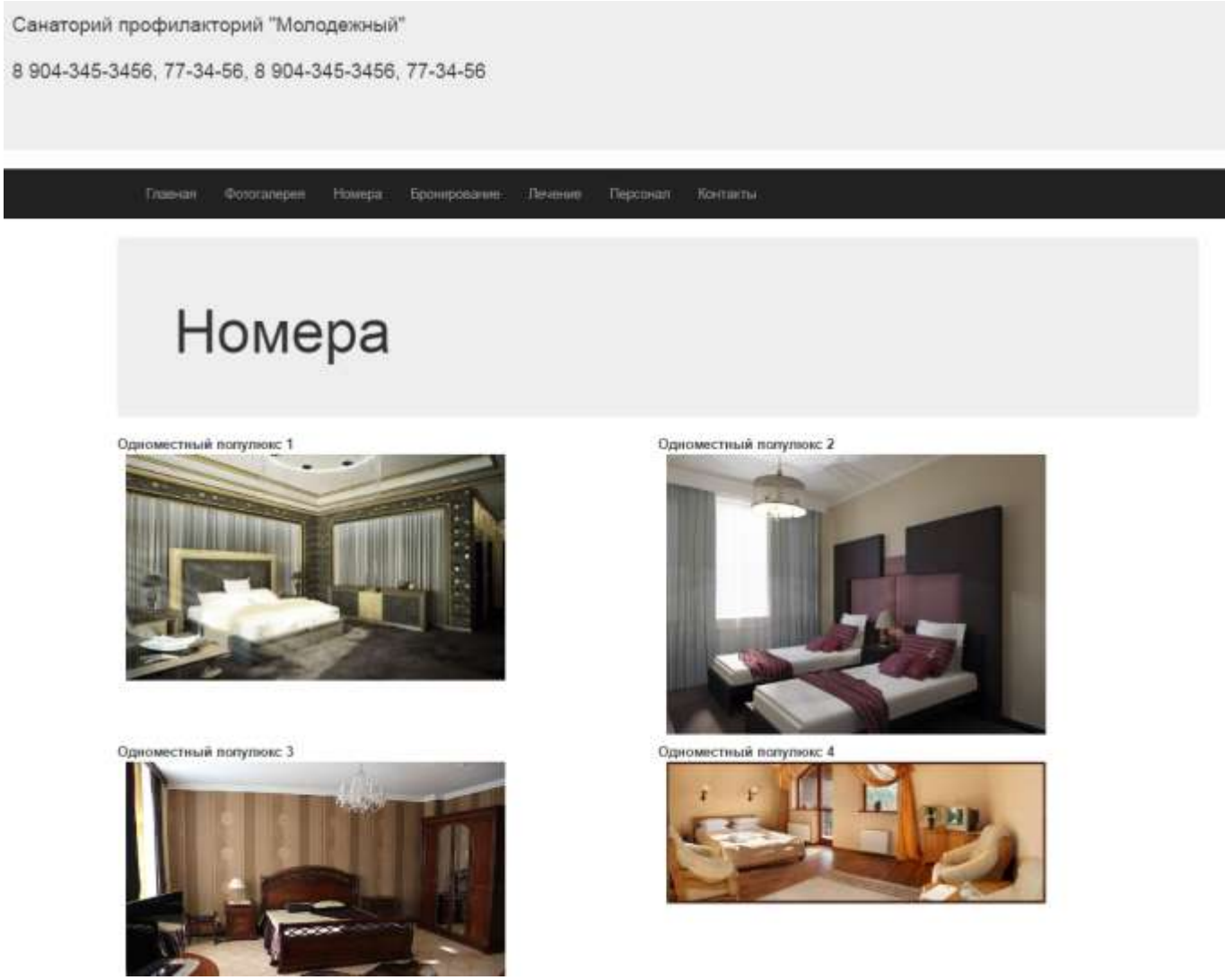

Рисунок. 3.5 – «Номера»

При переходе на пункт меню «Номера», будет представлено описание номера и его фотография.

Страница Web-приложения «Лечение» представлена на рисунке 3.6.

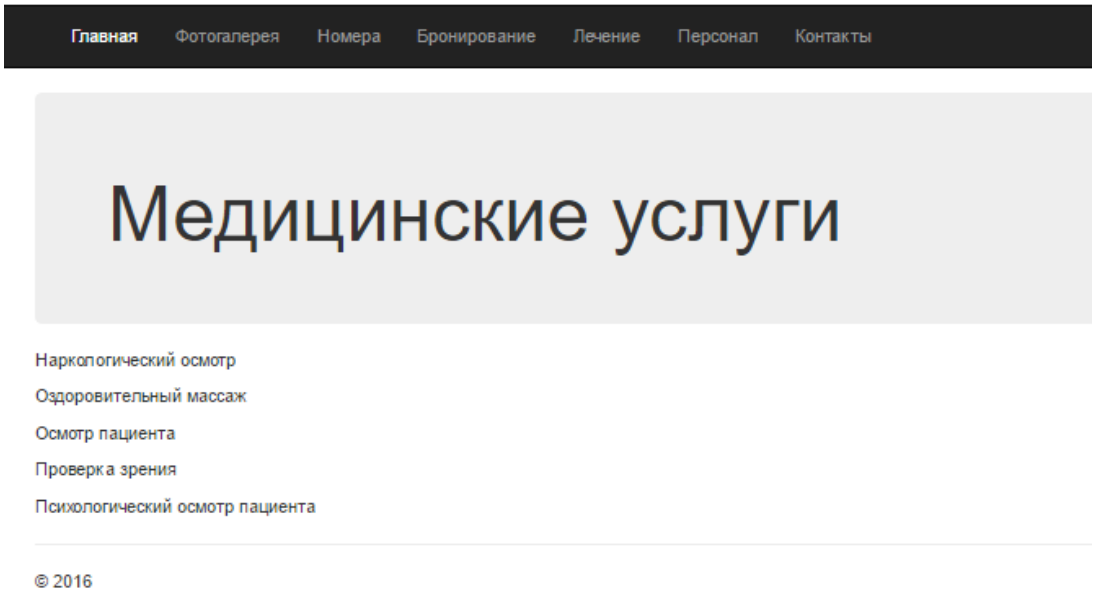

Рисунок. 3.6 – «Лечение»

При переходе на пункт меню «Лечение», будет представлен список всех медицинских услуг, которые оказывают в санатории.

Страница Web-приложения «Персонал» представлена на рисунке 3.7.

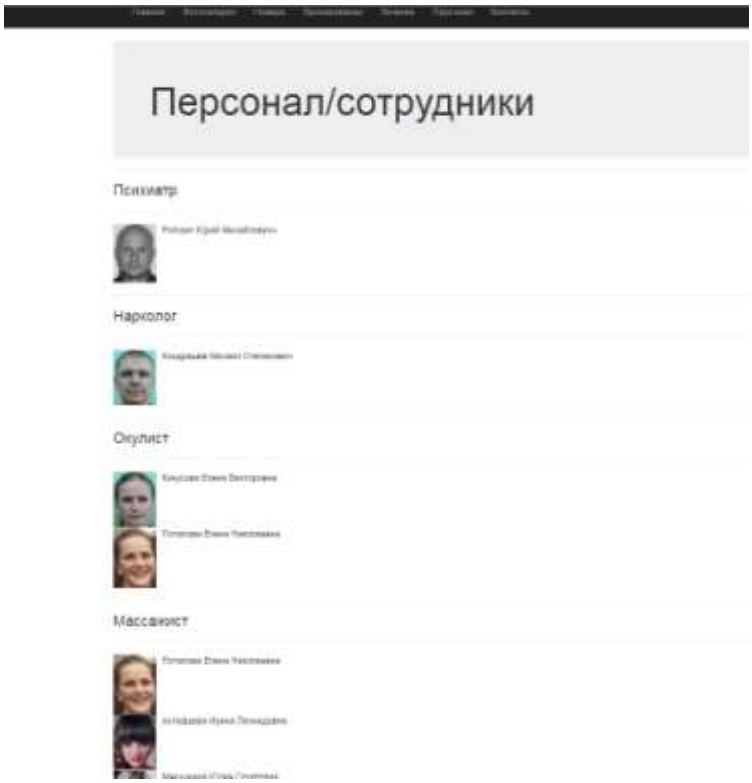

Рисунок. 3.7 – «Персонал/Сотрудники»

При переходе на пункт меню «Персонал», будет представлена информация о сотрудниках, их специальность, фотография врача.

Страница Web-приложения «Контакты» представлена на рисунке 3.8.

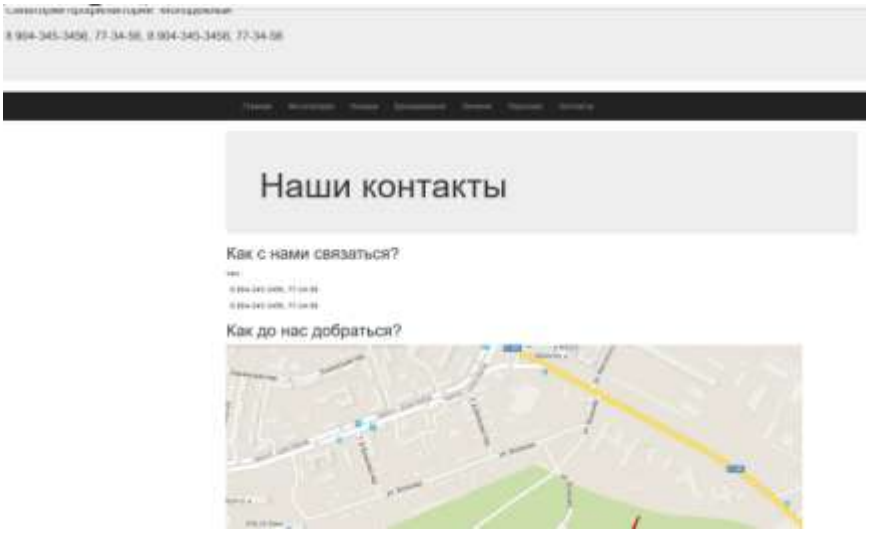

Рисунок. 3.8 – «Контакты»

При переходе на пункт меню «Контакты», будет представлена информация о местоположении санатория, контактный телефон.

3.2 Работа в режиме администратора

Для работы в режиме Администратора, требуется авторизация на сайте. В режиме Администратора нужно авторизоваться.

На рисунке 3.9 представлен процесс авторизации на сайте.

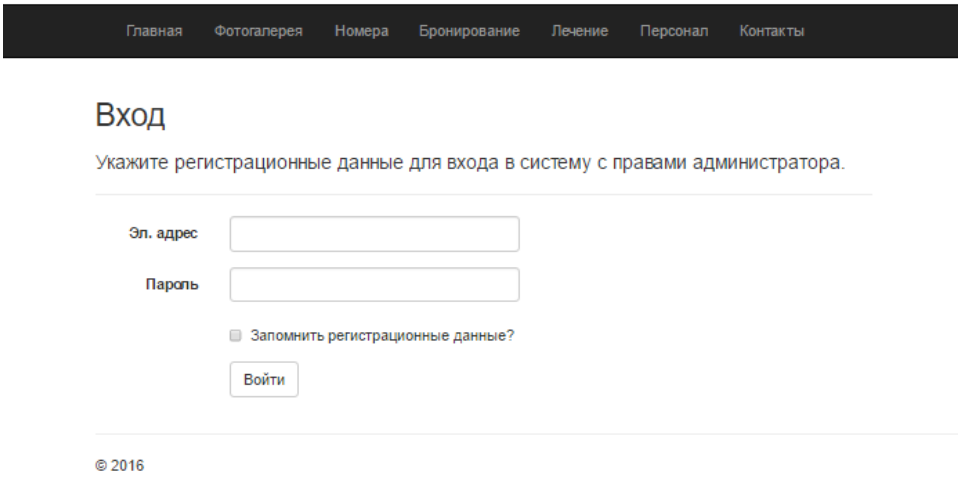

Рисунок. 3.9 – «Авторизация»

Чтобы авторизоваться на сайте, необходимо ввести электронную почту и пароль. При необходимости можно запомнить регистрационные данные. При нажатии кнопки «Войти» мы войдем в систему от имени Администратора.

После авторизации меню будет иметь другие поля выбора, такие как: «Фотогалерея», «Номера», «Бронирование», «Лечение», «Персонал», «Отчеты».

Страница «Фотогалерея» для работы от имени Администратора представлена на рисунке 3.10.

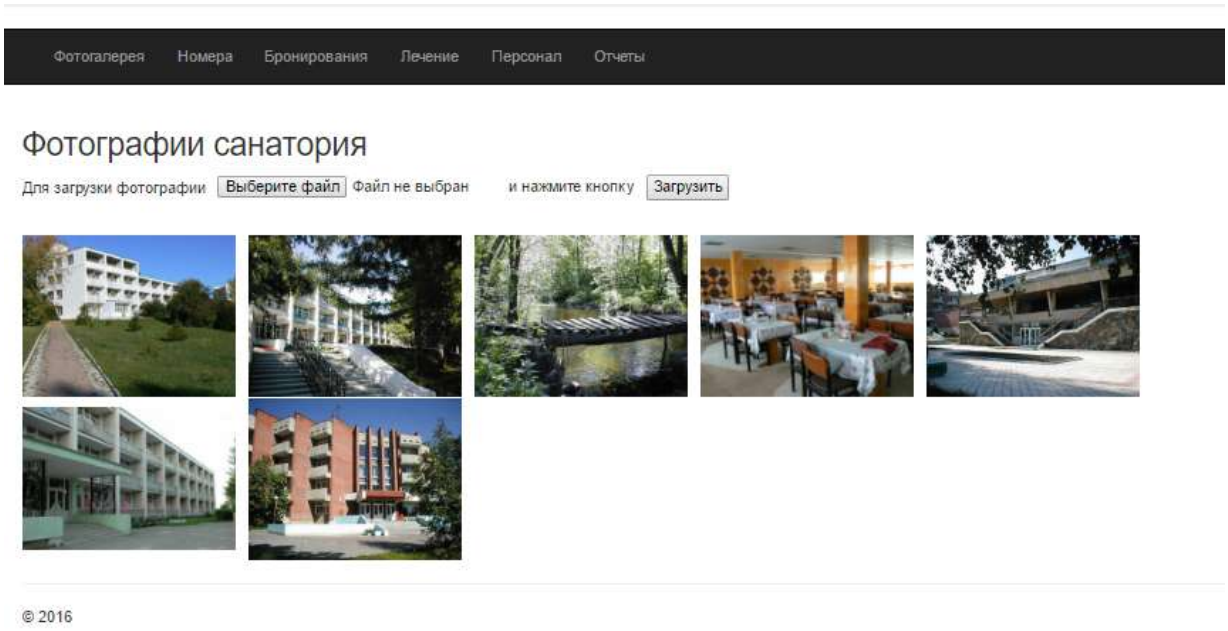

Рисунок. 3.10 – «Фотогалерея»

При переходе на пункт меню «Фотогалерея», будет представлена фотогалерея санатория. Здесь мы имеем возможность работы с фотогалереей. Можем добавлять новые фотографии, а также удалять их.

Для того, чтобы увидеть все наши номера, нужно выбрать на панели меню пункт «Номера».

Здесь мы имеем возможность работы с номерами. Мы можем изменить номер, просмотреть номер, удалить номер.

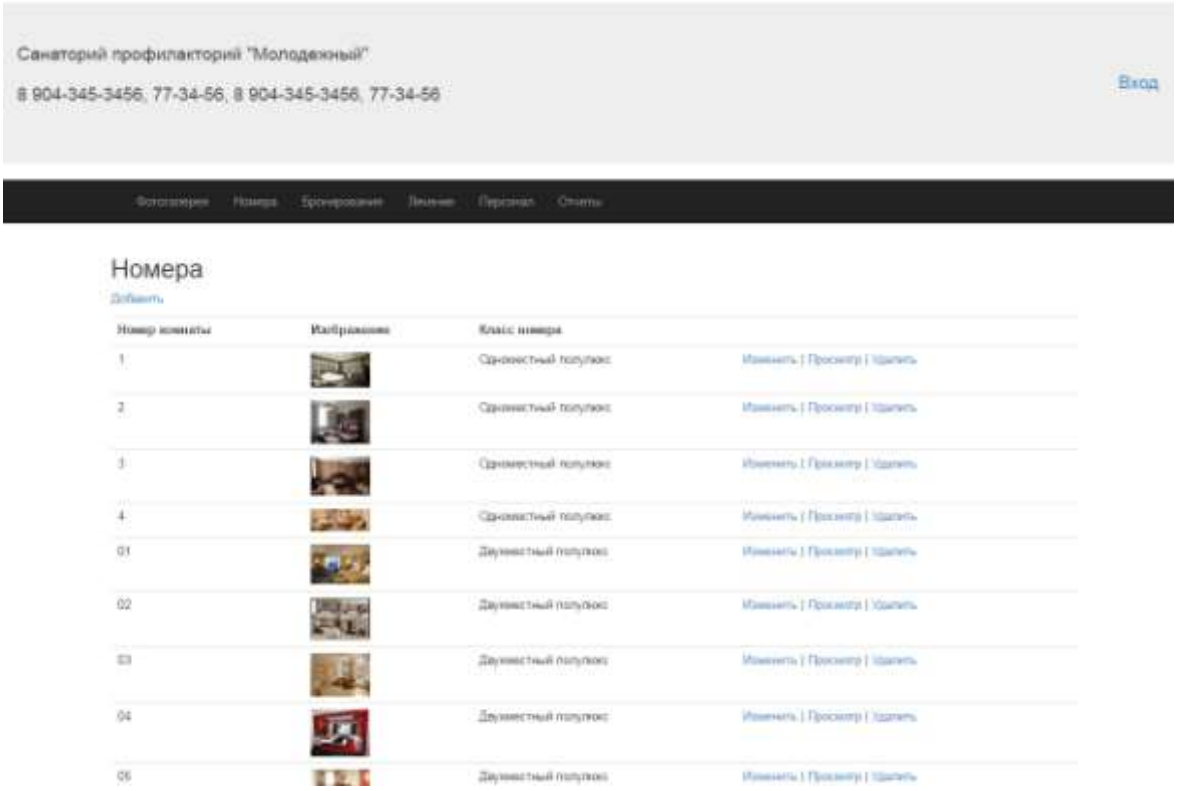

Рисунок. 3.11 – «Номера»

При переходе на пункт меню «Номера», будет представлено описание номера и фотография. Имеем возможность работы с номерами. Можно изменить номер, просмотреть номер, удалить номер.

Редактирование номера представлено на рисунке 3.12

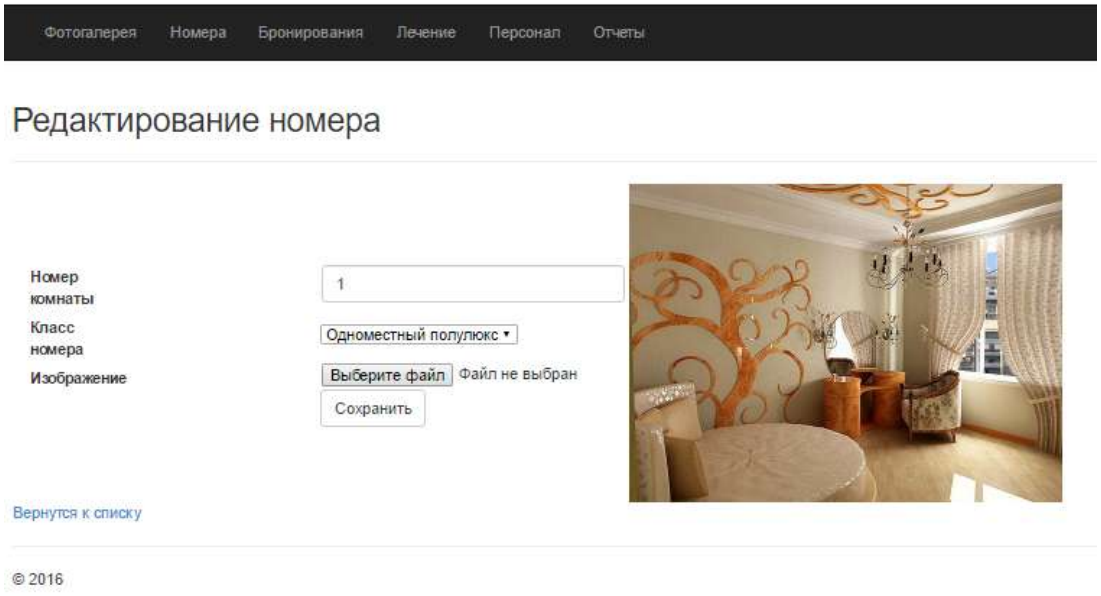

Рисунок. 3.12 – «Редактирование номера»

Здесь можно изменить информацию о номере.

Просмотр информации о номере отображен на рисунке 3.13

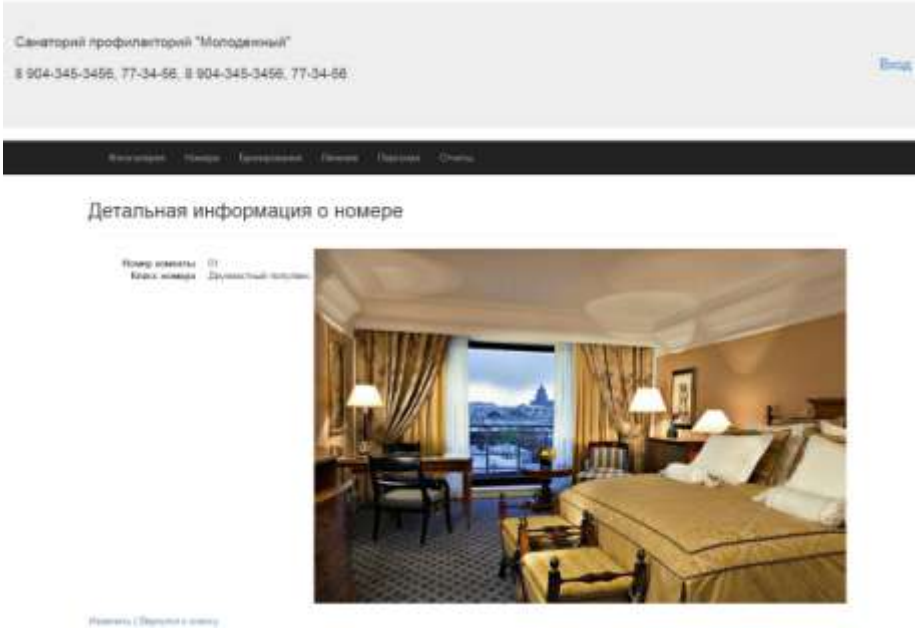

Рисунок. 3.13 – «Просмотр информации о номере»

Имеем возможность просмотреть основную информацию о номере.

Добавление нового номера показано на рисунке 3.14

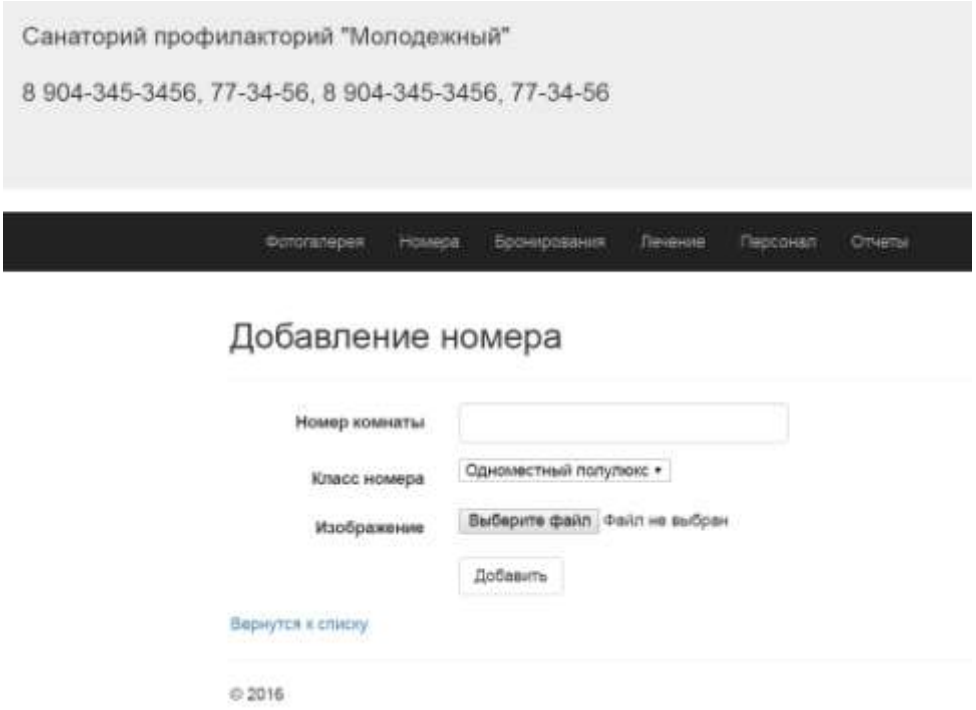

Рисунок. 3.14 – «Добавление номера»

Здесь можно добавить новый номер и информацию о нем.

Бронирование номеров показано на рисунке 3.15

Фотогалерея Номера Бронирования Лечение Персонал Отчеты

Бронирования

Добавить новое бронирование

ФИО	Телефон	Эл. адрес	Желаемая дата заезда	С лечением	Заявка подтверждена	
Таблеткин Федор Иванович	8 3453453445	Tabletka@mail.ru	01.01.2016 0:00:00	☑	☐	Изменить | Просмотр | Удалить
Огурцов Николай Васильевич	8 9054535353	onv@rambler.ru	02.04.2015 0:00:00	☐	☐	Изменить | Просмотр | Удалить
Васильев Петр Ильич	8 9052355553	vasilev@mail.ru	02.01.2016 0:00:00	☑	☐	Изменить | Просмотр | Удалить
Костюк Дмитрий Анатольевич	8 9032525257	kostyk@mail.ru	03.04.2016 0:00:00	☑	☐	Изменить | Просмотр | Удалить
Иванов Игорь Пертович	8 9005464664	sday2017@mail.ru	30.03.2016 0:00:00	☑	☐	Изменить | Просмотр | Удалить
Давыдов Василий Петрович	8 8903409575	davvaset@gmail.com	04.12.2016 0:00:00	☐	☑	Изменить | Просмотр | Удалить
Советов Илья Геннадьевич	8 9315813670	covetov@yandex.ru	01.01.2016 0:00:00	☐	☑	Изменить | Просмотр | Удалить
Золотцев Григорий Петрович	8 9089898890	goldgp@mail.ru	31.05.2016 0:00:00	☑	☐	Изменить | Просмотр | Удалить

Рисунок. 3.15 – «Бронирования»

Чтобы увидеть список всех поступивших заявок, необходимо выбрать пункт меню «Бронирования».

Редактирование бронирования показано на рисунке 3.16

Фотогалерея Номера Бронирования Лечение Персонал Отчеты

Редактирование бронирования №12

ФИО: Золотцев Григорий Петрович

Телефон: 8 9089898890

Эл. адрес: goldgp@mail.ru

Желаемая дата заезда: 31.05.2016 0:00:00

С лечением ☑

Заявка подтверждена ☐

Сохранить

Вернутся к списку

© 2016

Рисунок. 3.16 – «Редактирование бронирования»

Имеем возможность работы с заявками, возможность подтвердить заявку. Можно изменить заявку.

Просмотр заявки показан на рисунке 3.17

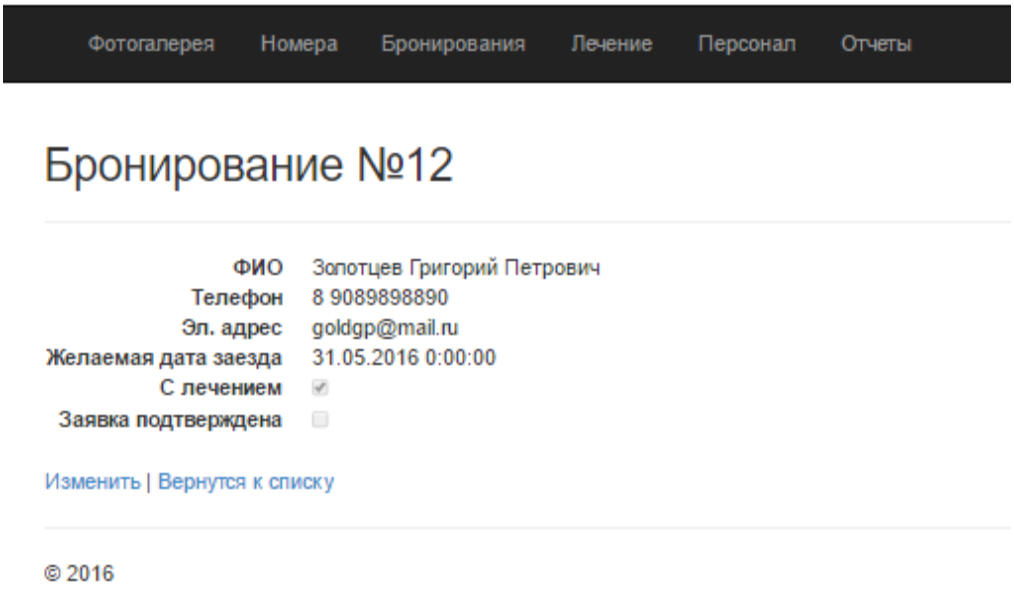

Рисунок. 3.17 – «Просмотр бронирования»

Имеем возможность просмотр заявки.

Удаление заявки показано на рисунке 3.18.

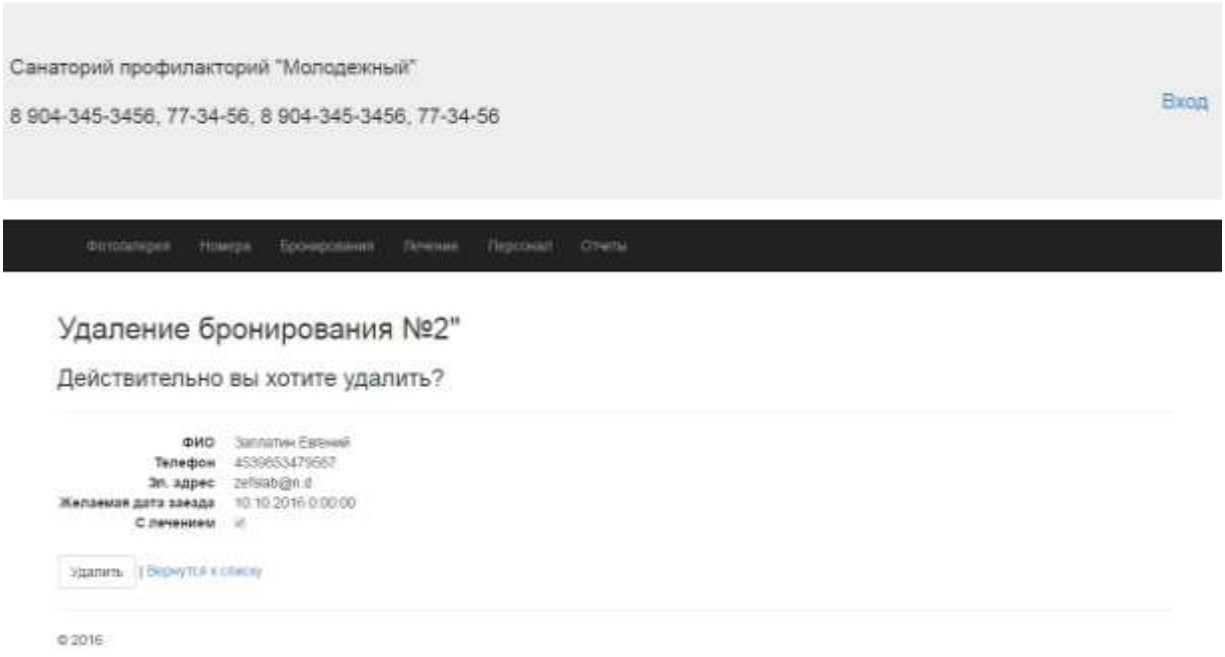

Рисунок. 3.18 – «Удаление бронирования»

Имеем возможность удаления заявки.

Страница «Лечение» показана на рисунке 3.19

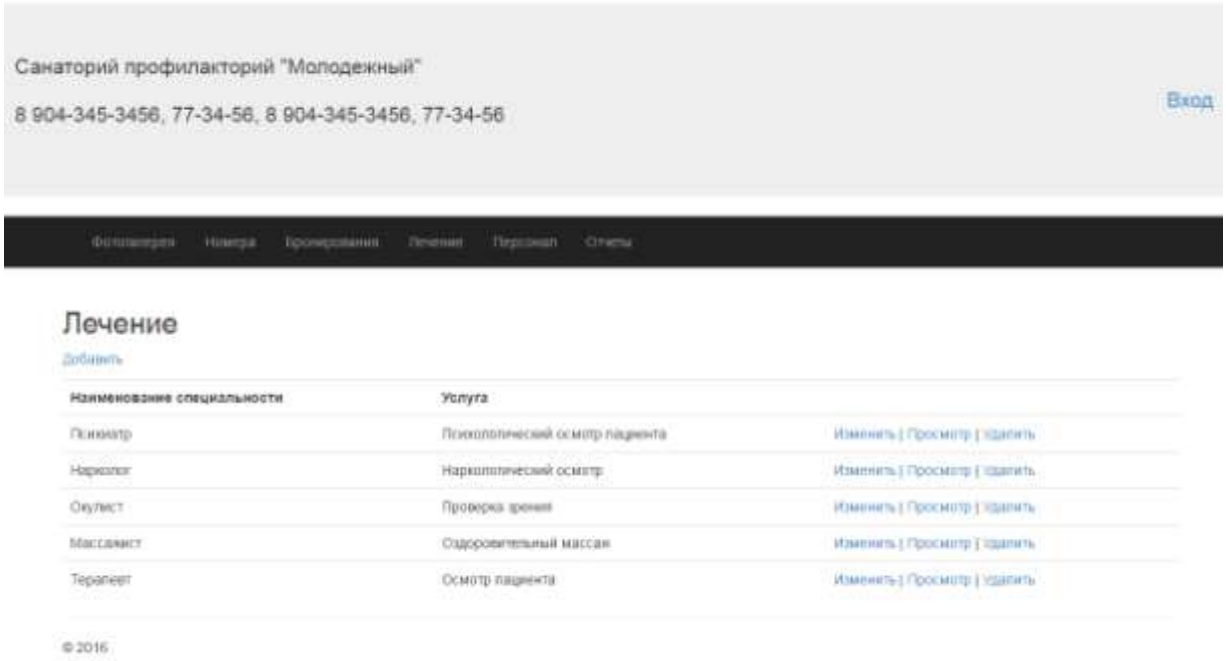

Рисунок. 3.19 – «Лечение»

Имеем возможность Изменить/Просмотреть/Удалить запись.

Страница «Изменение услуги» показана на рисунке 3.20.

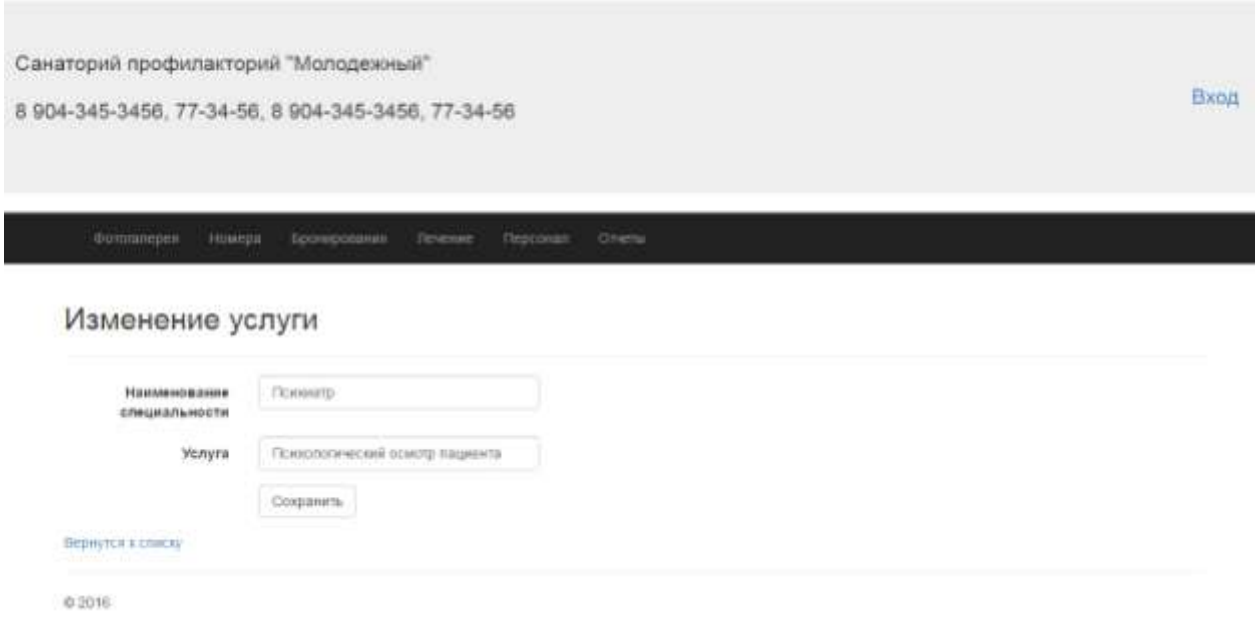

Рисунок. 3.20 – «Изменение услуги»

Имеем возможность изменения медицинских услуг.

Страница «Просмотр услуги» показана на рисунке 3.21.

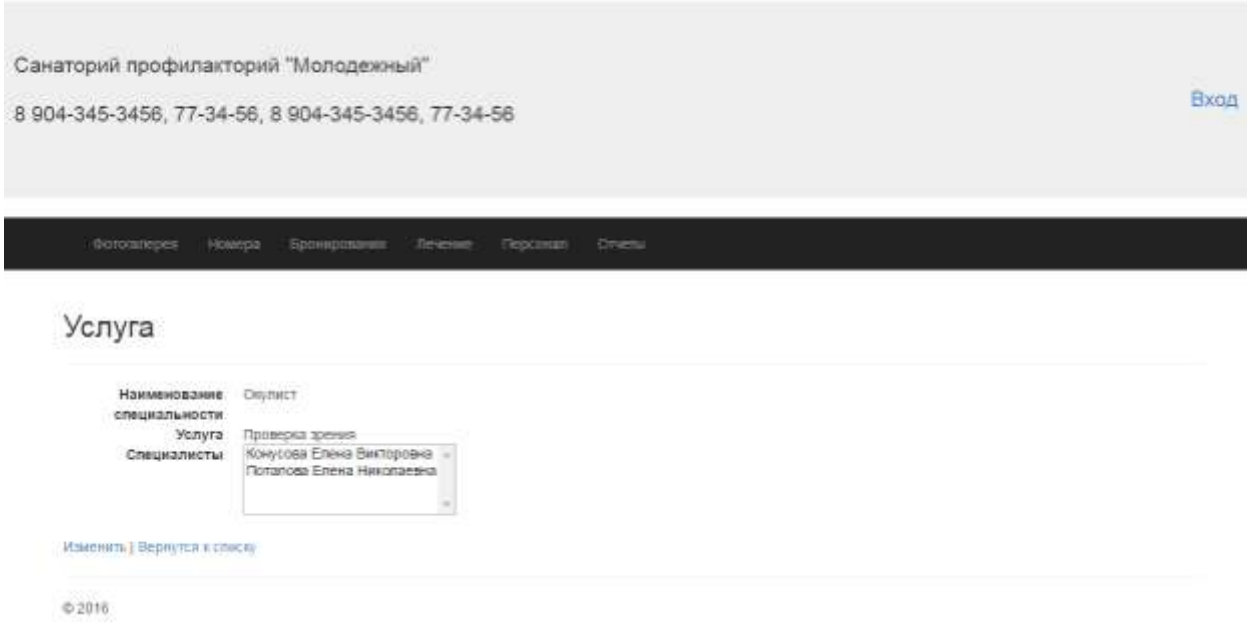

Рисунок. 3.21– «Просмотр услуги»

Имеем возможность просмотра медицинских услуг.

Страница «Добавление услуги» показана на рисунке 3.22.

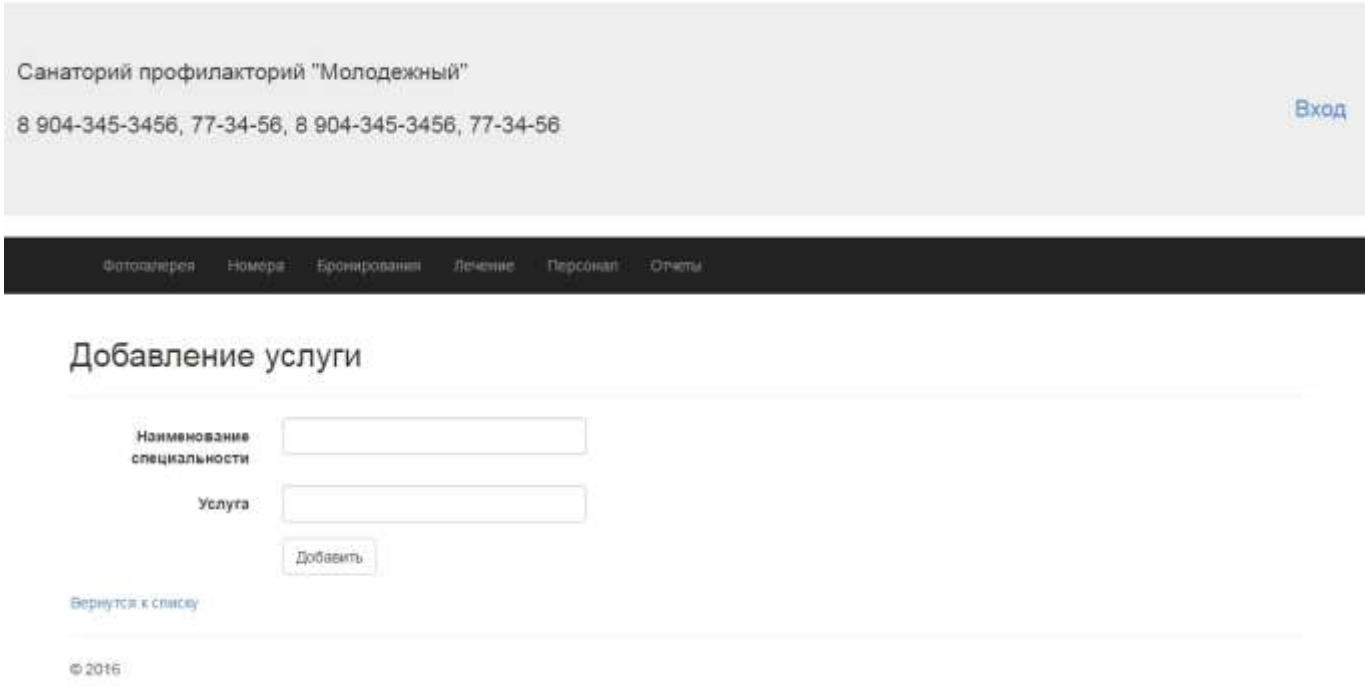

Рисунок. 3.22 – «Добавление услуги»

Имеем возможность добавления новых медицинских услуг.

Страница «Персонал» показана на рисунке 3.23.

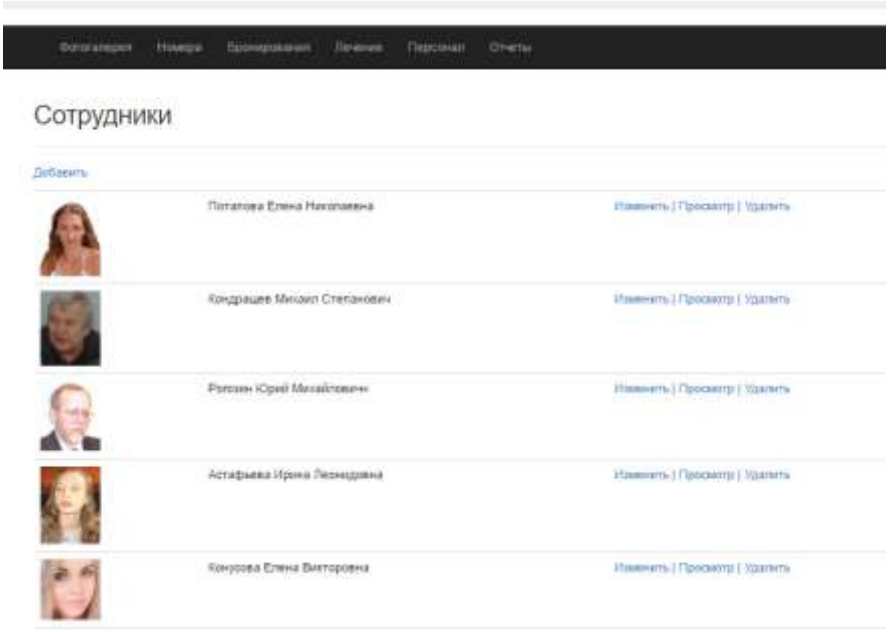

Рисунок. 3.23 – «Сотрудники/Персонал»

Имеем возможность Изменить/Просмотреть/Удалить информацию о сотрудниках санатория.

Страница «Просмотр информации о сотруднике» показана на рисунке 3.24.

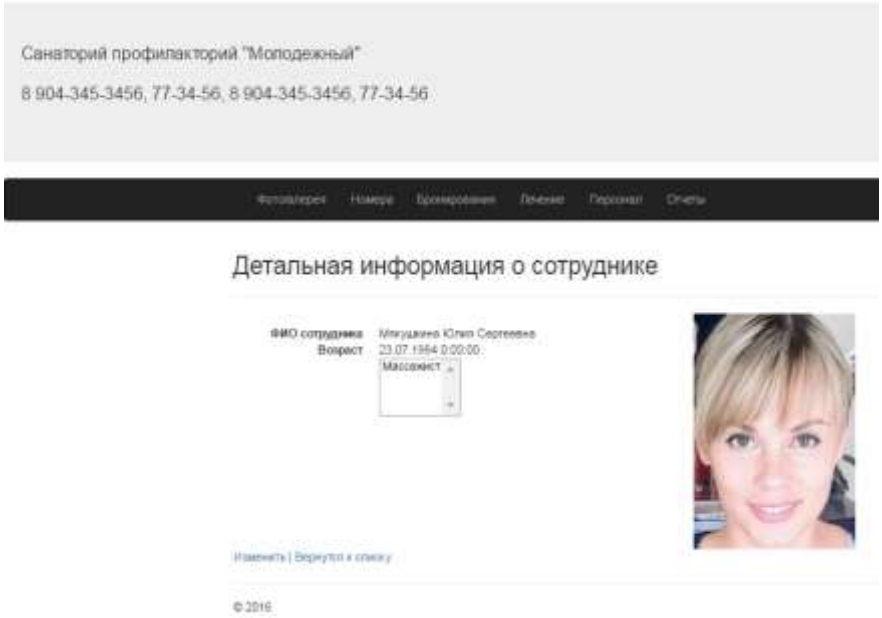

Рисунок. 4.24 – «Просмотр информации о сотруднике»

Имеем возможность просмотра информации о сотруднике.

Страница «Изменение информации о сотруднике» показана на рисунке 3.25.

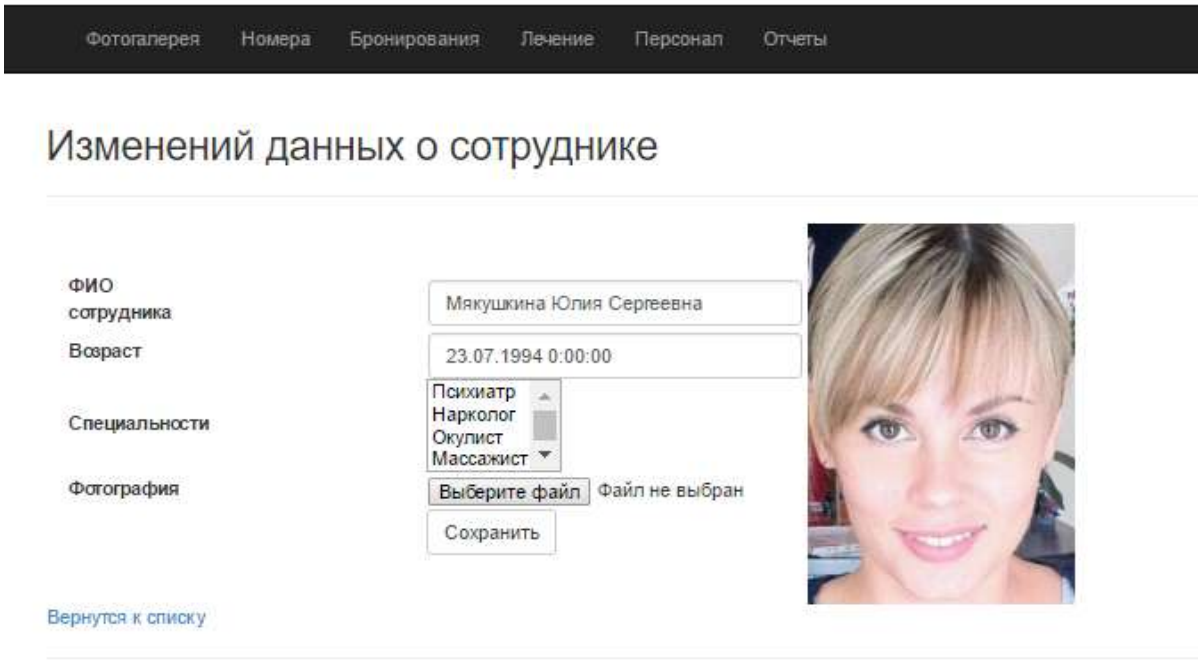

Рисунок. 3.25 – «Изменение информации о сотруднике»

Имеем возможность изменения информации для каждого сотрудника.

Страница «Удаление сотрудника» показана на рисунке 3.26.

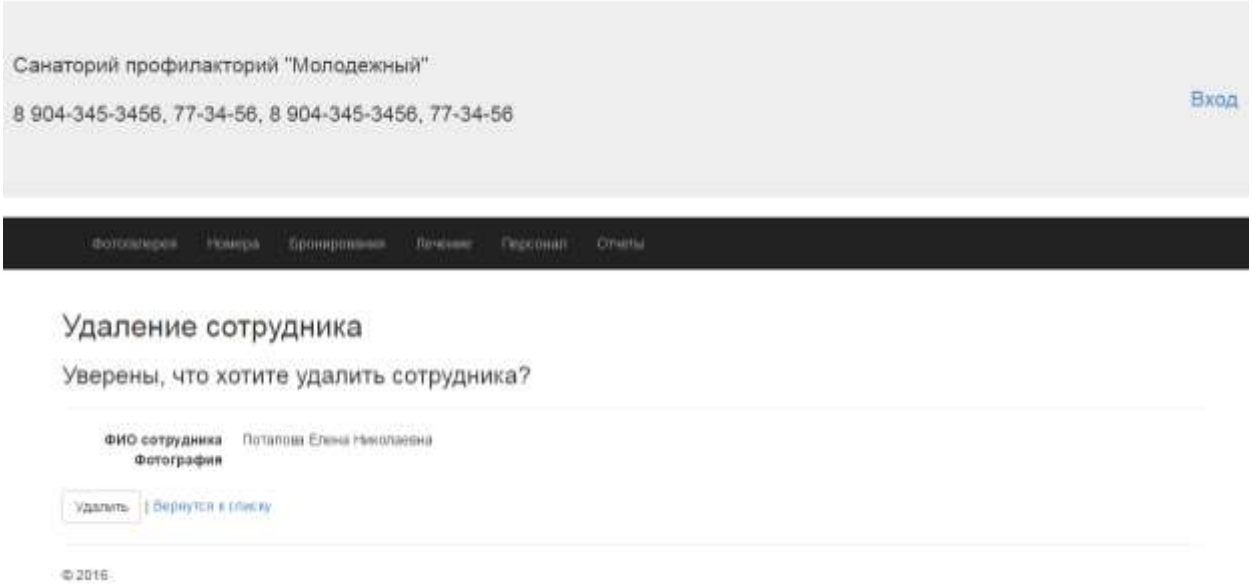

Рисунок. 3.26 – «Удаление сотрудника»

Имеем возможность удаления всей информации о сотруднике.

Страница «Добавление нового сотрудника» показана на рисунке 3.27.

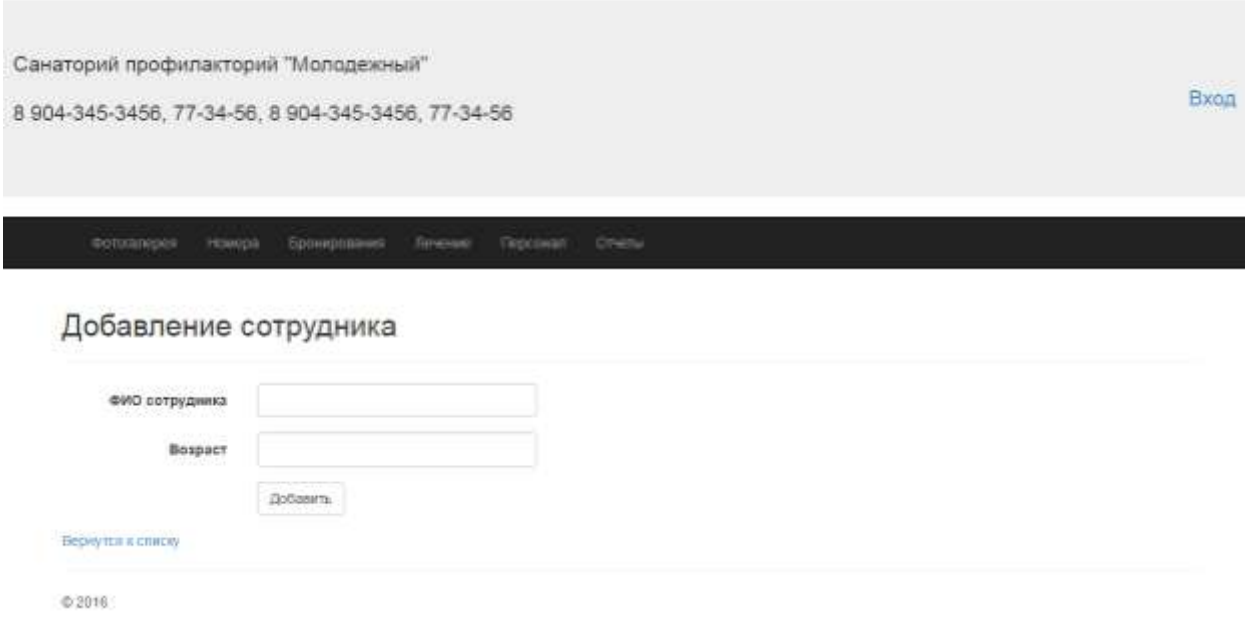

Рисунок. 3.27 – «Добавление нового сотрудника»

Имеем возможность добавления нового сотрудника.

Просмотр отчетов

Страница «Отчет по сотрудникам санатория» показан на рисунке 4.1.

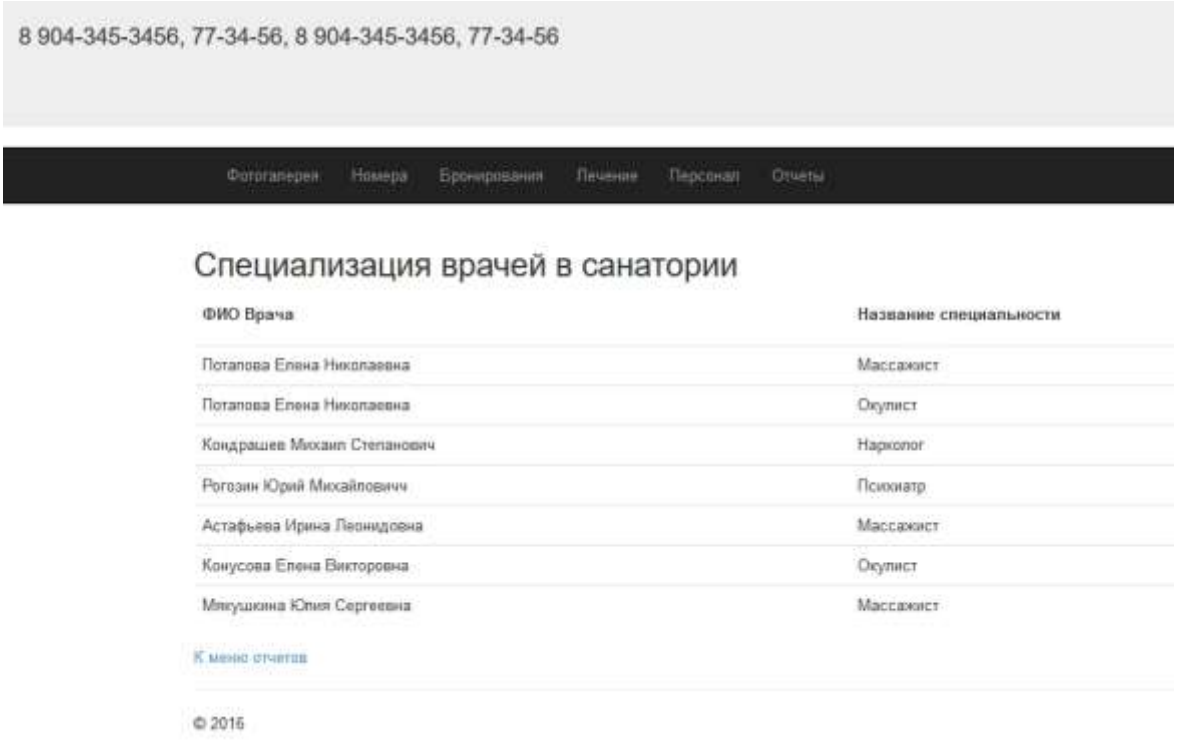

Рисунок. 4.1 – «Отчет по сотрудникам санатория»

Исходя из всех данных формируется отчет по сотрудникам санатория.

Страница «Отчет по клиентам» показана на рисунке 4.2.

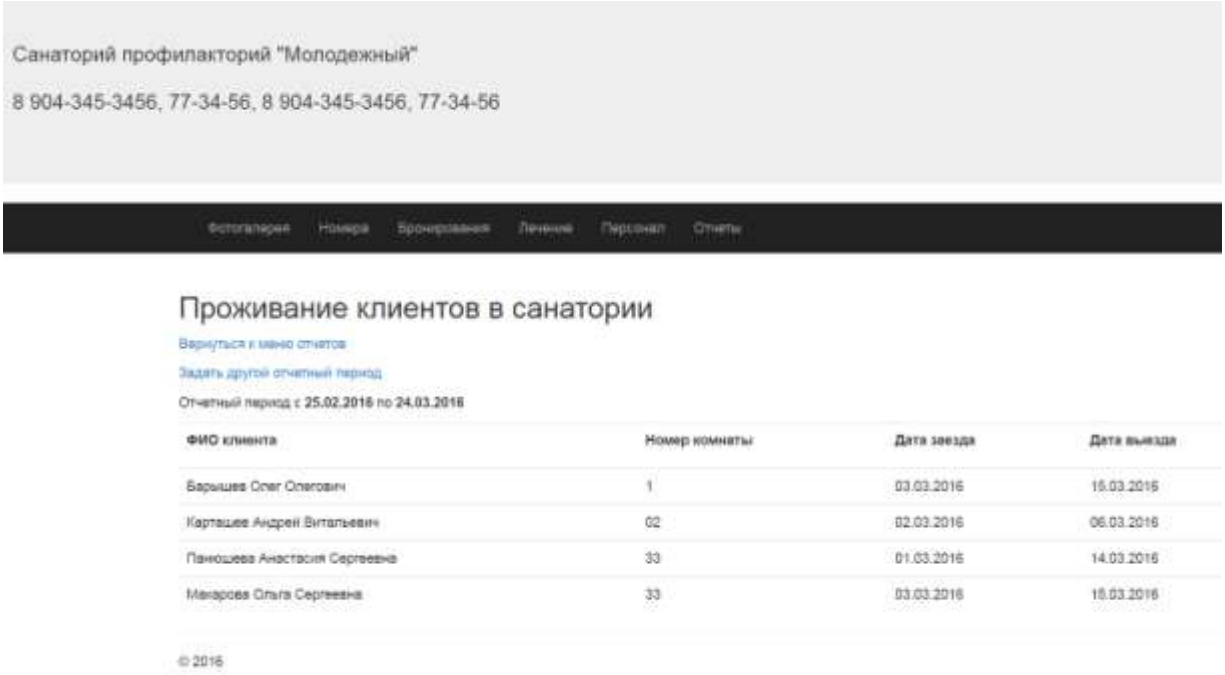

Рисунок. 4.2 – «Отчет по клиентам»

Формируется отчет по клиентам, проживающих в санатории.

Выводы по разделу.

Показан процесс работы Web-приложения в режиме Пользователя и в режиме Администратора.

ПРИЛОЖЕНИЕ А

КОД СТРАНИЦЫ «АВТОРИЗАЦИЯ»

```
@using WebMvc.Models
@model LoginViewModel
@{
   ViewBag.Title = "Вход";
}

<h2>@ViewBag.Title</h2>
<div class="row">
   <div class="col-md-8">
      <section id="loginForm">
         @using (Html.BeginForm("Login", "Account", new { ReturnUrl = ViewBag.ReturnUrl
}, FormMethod.Post, new { @class = "form-horizontal", role = "form" }))
         {
            @Html.AntiForgeryToken()
            <h4>Укажите регистрационные данные для входа в систему с правами
администратора.</h4>
            <hr />
            @Html.ValidationSummary(true, "", new { @class = "text-danger" })
            <div class="form-group">
               @Html.LabelFor(m => m.Email, new { @class = "col-md-2 control-label" })
               <div class="col-md-10">
                  @Html.TextBoxFor(m => m.Email, new { @class = "form-control" })
                  @Html.ValidationMessageFor(m => m.Email, "", new { @class = "text-danger"
})
               </div>
            </div>
            <div class="form-group">
               @Html.LabelFor(m => m.Password, new { @class = "col-md-2 control-label" })
```

Окончание приложения А

```
               <div class="col-md-10">
                  @Html.PasswordFor(m => m.Password, new { @class = "form-control" })
                  @Html.ValidationMessageFor(m => m.Password, "", new { @class = "text-
danger" })
               </div>
            </div>
            <div class="form-group">
               <div class="col-md-offset-2 col-md-10">
                  <div class="checkbox">
                     @Html.CheckBoxFor(m => m.RememberMe)
                     @Html.LabelFor(m => m.RememberMe)
                  </div>
               </div>
            </div>
            <div class="form-group">
               <div class="col-md-offset-2 col-md-10">
                  <input type="submit" value="Войти" class="btn btn-default" />
```

```
                </div>
            </div>

        </section>
    </div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
```

ПРИЛОЖЕНИЕ Б

КОД СТРАНИЦЫ «ФОТОГАЛЕРЕЯ»

```
@using WebMvc.Views
@model string []
@{
   ViewBag.Title = "Photogalery";
   Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}

<h2>Фотографии санатория</h2>
@using (Html.BeginForm("Upload", "Admin", FormMethod.Post, new { enctype = "multipart/form-data" }))
{
   <table>
      <tr>
         <td>Для загрузки фотографии  </td>
         <td><input type="file" name="upload" /></td>
         <td>и нажмите кнопку   <input type="submit" value="Загрузить" /></td>
      </tr>
      <tr><td><p> </p></td></tr>
   </table>
}
@{
   foreach (var imgFile in Model)
   {
      var file = "/Content/Files/" + imgFile;
      <span><img src="@file" width="200" />  </span>
   }
}
```

ПРИЛОЖЕНИЕ В

КОД СТРАНИЦЫ «ОТЧЕТ: СПЕЦИАЛИЗАЦИЯ ВРАЧЕЙ В САНАТОРИИ»

```
@model List<WebSanatorium.Api.Doctor>
@{
   ViewBag.Title = "Специализация врачей в санатории";
   Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}

<h2>@ViewBag.Title</h2>

<p>@Html.ActionLink("К меню отчетов", "Reports") </p>


@{
   var specialties = Model.SelectMany(x => x.Specialties.Select(s=> new
                    {
                       Fio = x.FullName,
                       Specialty = s.Name
                    }));
}
<table width="100%" class="table">
      <tr>
         <th>
            <p>ФИО Врача</p>
         </th>
         <th>
            <p>Название специальности</p>
         </th>
```

Окончание приложения В

```
      </tr>
      @{
         foreach (var doctor in specialties)
         {
            <tr>
               <td>@doctor.Fio</td>
               <td>@doctor.Specialty</td>
            </tr>

         }
      }
   </table>
```

ПРИЛОЖЕНИЕ Г

КОД СТРАНИЦЫ «ОТЧЕТ: ПРОЖИВАНИЕ КЛИЕНТОВ»

```
@using WebMvc.Models
@model List<WebSanatorium.Api.ClientPermit>
@{
   ViewBag.Title = "Проживание клиентов в санатории";
   Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}

<h2>@ViewBag.Title</h2>

<p>@Html.ActionLink("Вернуться к меню отчетов", "Reports") </p>
<p>@Html.ActionLink("Задать другой отчетный период", "ReportByClientsCreteria") </p>

@{
   var clients = Model.Select(x => new
   {
      Fio = x.Client.FullName,
      Room = x.Room.Number,
      BeginDate = x.BeginDate,
      EndDate = x.EndDate
   });

   var period = (Period)ViewBag.Period;

   }
   <p>
      Отчетный период с <b>@period.BeginDate.ToShortDateString()</b> по
<b>@period.EndDate.ToShortDateString()</b>
```

Окончание приложения Г

```
   </p>
   <table width="100%" class="table">
      <tr>
         <th>
            <p>ФИО клиента </p>
         </th>
         <th>
            <p>Номер комнаты</p>
         </th>
         <th>
            <p>Дата заезда</p>
         </th>
         <th>
            <p>Дата выезда</p>
         </th>
      </tr>
      @{
```

```
        foreach (var client in clients)
        {
            <tr>
                <td>@client.Fio</td>
                <td>@client.Room</td>
                <td>@client.BeginDate.ToShortDateString()</td>
                <td>@client.EndDate.ToShortDateString()</td>

            </tr>
        }
    }
</table>
```

ПРИЛОЖЕНИЕ Д

КОД СТРАНИЦЫ «ДОБАВЛЕНИЕ СОТРУДНИКА»

```
@using WebMvc.Views
@model WebSanatorium.Api.Doctor

@{
   ViewBag.Title = "Добавление сотрудника";
   Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}

<h2>@ViewBag.Title</h2>


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

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

Окончание приложения Д

```
      <div class="form-group">
         @Html.LabelFor(model => model.Old, htmlAttributes: new { @class = "control-label
col-md-2" })
         <div class="col-md-10">
            @Html.EditorFor(model => model.Old, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.Old, "", new { @class = "text-danger"
})
         </div>
      </div>
      <div class="form-group">
         <div class="col-md-offset-2 col-md-10">
            <input type="submit" value=@Str.Create class="btn btn-default" />
         </div>
      </div>
   </div>
```

```
}

<div>
    @Html.ActionLink(Str.ToList, "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

ПРИЛОЖЕНИЕ Е

КОД СТРАНИЦЫ «ДОБАВЛЕНИЕ ЗАЯВКИ»

```
@using WebMvc.Views
@model WebSanatorium.Api.Order

@{
   ViewBag.Title = "Добавление заявки";
   Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}

<h2>@ViewBag.Title</h2>

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

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

Продолжение приложения Е

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

Продолжение приложения Е

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

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value=@Str.Create class="btn btn-default" />
        </div>
    </div>
  </div>
}
```

Окончание приложения Е

```
<div>
  @Html.ActionLink(Str.ToList, "Index")
```

ПРИЛОЖЕНИЕ Ж

КОД СТРАНИЦЫ «ДОБАВЛЕНИЕ НОМЕРА»

```
@using WebMvc.Views
@using WebSanatorium.Api
@model WebSanatorium.Api.Room

@{
   ViewBag.Title = "Добавление номера";
   Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}

<h2>@ViewBag.Title</h2>

@using (Html.BeginForm("Create", "Rooms", FormMethod.Post, new { enctype = "multipart/form-data" }))
{
   @Html.AntiForgeryToken()

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

Продолжение приложения Ж

```
      </div>

      <div class="form-group">
         @Html.LabelFor(model => model.RoomClass, htmlAttributes: new { @class = "control-label col-md-2" })
         <div class="col-md-10">
            @{ //Подготовка списка для dropdown
               var classList = ((IEnumerable<RoomClass>)ViewBag.RoomClassList).Select(x => new SelectListItem
               {
                  Text = x.Name,
                  Value = x.Id.ToString()
               }).AsEnumerable();
            }
            @Html.DropDownListFor(m => m.RoomClass.Id, classList, new { htmlAttributes = new { @class = "form-control" } })
```

```
                @Html.ValidationMessageFor(m => m.RoomClass.Id, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.PictureFile, htmlAttributes: new {@class = "control-label col-md-2"})
            <div class="col-md-10">
                <input type="file" name="upload"/>
            </div>
        </div>
        <div class="form-group">

            <div class="col-md-offset-2 col-md-10">
```

Окончание приложения Ж

```
                <input type="submit" value=@Str.Create  class="btn btn-default" />
            </div>
        </div>
    </div>
}

<div>
    @Html.ActionLink(Str.ToList, "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

ПРИЛОЖЕНИЕ 3

КОД СТРАНИЦЫ «ДОБАВЛЕНИЕ УСЛУГИ»

```
@using WebMvc.Views
@model WebSanatorium.Api.Specialty

@{
   ViewBag.Title = "Добавление услуги";
   Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}

<h2>@ViewBag.Title</h2>


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

   <div class="form-horizontal">
      <hr />
      @Html.ValidationSummary(true, "", new { @class = "text-danger" })
      <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>
```

Окончание приложения 3

```
      </div>

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

      <div class="form-group">
         <div class="col-md-offset-2 col-md-10">
            <input type="submit" value=@Str.Create class="btn btn-default" />
         </div>
```

```
        </div>
    </div>
}

<div>
    @Html.ActionLink(Str.ToList, "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

ЗАКЛЮЧЕНИЕ

Цель выпускной квалификационной работы – разработка Web-приложения для автоматизации работы санатория.

Для решения поставленной задачи использовались следующие программные инструменты – HTML, CSS, XML, ASP.Net MVC, ADO.NET Entity Framework.

В ходе выполнения дипломного проекта решены задачи:

- выполнен анализ предметной области;
- выполнен анализ существующих систем;
- описана структуры программы;
- выбраны инструменты для разработки;
- разработан проект программного продукта;
- разработаны интерфейсы и бизнес-логика;
- выполнено тестирование.

Итогом работы над дипломным проектом является готовое Web-приложение, которое отвечает всем установленным требованиям. Поставленные задачи выполнены, и цель дипломного проекта была достигнута.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Титтел, Э. HTML, XHTML и CSS для чайников, 7-е издание = HTML, XHTML & CSS For Dummies, 7th Edition / Э. Титтел, Д. Ноубл. – М.: Диалектика, 2011. 400 с.

2 Ливингстон, Д. CSS и DHTML: Web – профессионалам / Д. Ливингстон , М. Браун. – Киев: BHV, 2001. 264 с.

3 «MSDN» – информационный сервис для разработчиков. Электронный сайт: http://msdn.microsoft.com/ru-RU/

4 «КиберФорум» – компьютерный форум. Электронный сайт: http://www.cyberforum.ru/.