

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
**«Владимирский государственный университет
Имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИВлГУ)**

Факультет _____ ИТ _____

Кафедра _____ ПИН _____

КУРСОВАЯ РАБОТА

по Системе управления базами данных

Тема Распределенная информационная система администратора гостиницы

_____	Руководитель
(оценка)	<u>Астафьев А.В.</u>
	(фамилия, инициалы)

	(подпись) (дата)
Члены комиссии	Студент <u>ПИН-119</u>
	(группа)
_____	<u>Буров А.М.</u>
(подпись) (Ф.И.О.)	(фамилия, инициалы)
_____	_____
(подпись) (Ф.И.О.)	(подпись) (дата)

Муром 2022

В данной курсовой работе была разработана распределенной ИС домоуправления для автоматизации предметной области по технологии ASP.NET MVC 5. В ходе выполнения работы была проанализирована предметная область, выявлены требования к работе, разработаны модели данных и диаграммы. На основе разработанных моделей данных был реализован набор классов и разработано приложение на языке C# в среде разработки Microsoft Visual Studio 2019. На заключительном этапе работы произведено тестирование разработанного продукта.

In this course work, a distributed system of house management was developed to automate the subject area by technology ASP.NET MVC 5. In the course of the work, the subject area was analyzed, the requirements for the work were identified, data models and diagrams were developed. Based on the developed data models, a set of classes was implemented and a C# application was developed in the Microsoft Visual Studio 2019 development environment. At the final stage of the work, the developed product was tested.

Содержание

Введение	6
1. Анализ технического задания	7
2. Разработка моделей данных	10
2.1 Выбор доменов	10
2.2. Логическая модель данных	11
2.3. Физическая модель данных	12
3. Проектирование работы системы	13
3.1. Диаграммы UML	13
3.2. Диаграммы DFD	14
3.3. Диаграммы последовательностей	16
3.4. Диаграммы SADT	18
4. Разработка и реализация АИС	20
4.1. Модели данных	20
4.2. Вывод всех номеров	21
4.3. Добавление номера	21
4.4. Изменение данных о номере	22
4.5. Удаление номера	23
4.6. Формирование отчетов	24
5. Тестирование АИС	26
Заключение	28
Список используемой литературы	29
Приложение 1. Модели данных	30
Приложение 2 Снимки окон программы.	33
Приложение 3. Код программы	39

					МИВУ 09.03.04-01.000 ПЗ									
Из	Лис	№ докум.	Подп.	Дата	Распределенная ИС администратора гостиницы					Лит.		Лист	Листов	
Разраб.		Буров А.М.											5	39
Провер.		Астафьев А.В.								МИВУ ПИН - 119				
Н.контр.														
Утв.														

Введение

Проблема автоматизации производственных процессов и процессов управления как средства повышения производительности труда всегда являлась и остается актуальной в производстве, народном хозяйстве, сфере услуг. Необходимость автоматизации управления объясняется задачами облегчения труда управленческого персонала, сдерживанием роста его численности, вызываемым развитием производства; усложнением производственных связей; увеличением объемов управленческой функции. Важную роль играет задача соответствия технической базы управления аналогичной базе производства, в отношении которого производится автоматизация.

Гостиница - средство размещения, состоящее из определенного количества номеров, имеющее единое руководство, предоставляющее определенный набор услуг и сгруппированное в классы в соответствие с предоставляемыми услугами и оборудованием номеров.

Объектом автоматизации является гостиница, а именно место администратора. Именно здесь сосредоточенно много однотипной и рутинной работы, которая за счёт автоматизации может дать значительный прирост быстродействия администратора, организовать эффективное оформления клиентов и учёт свободных номеров. Количество информации, которую необходимо получать, обрабатывать, использовать и хранить, увеличивается с огромной скоростью. Именно поэтому возникает всё большая потребность в автоматизации. Замена ведения бумажного каталога электронным, значительно увеличит скорость поиска необходимой информации.

Актуальность данного проекта заключается в том, что с помощью данного приложения администратор гостиницы сможет вести учёт клиентов и быстро и точно узнавать свободные номера, которые можно забронировать. Это очень удобно, ведь клиентам будут говорить актуальную цену, характеристики и прочие необходимые детали, а пользователь данной программы будет получать большую

					МИВУ 09.03.04-01.000 ПЗ	Лист
Из	Лист	№ докум.	Подп.	Дата		6

прибыль и хорошие отзывы за короткие сроки выполнения своей работы и за её качество.

Целью курсовой работы является разработка распределенной ИС для автоматизации предметной области по технологии ASP.NET MVC 5.

В рамках реализации цели были поставлены задачи:

- Рассмотреть существующие аналоги;
- Разработать алгоритмы программы;
- Изучить основные понятия АРМ, функции, классификации, свойства;
- Исследовать организационную структуру гостиничного хозяйства;
- Определить функции приложения;
- Создать необходимые модели базы данных;
- Ознакомиться и поработать с системами управления базами данных;
- Реализовать программу.

					МИВУ 09.03.04-01.000 ПЗ	Лист
						7
Из	Лист	№ докум.	Подп.	Дата		

1. Анализ технического задания

В данной курсовой работе необходимо разработать автоматизированную информационную систему администратора гостиницы. Она используется с целью учета номеров, постояльцев, проживаний, бронирований и оплаты услуг. Для разработки такой системы необходимы база данных и среда разработки для создания интерфейса и функций по работе с БД.

1.1 Обоснование выбора средств реализации

В качестве среды по разработки, была выбрана среда объектно-ориентированного программирования Microsoft Visual Studio 2019, язык программирования C#. Microsoft Visual Studio включает в себя полный набор новых и улучшенных функций, упрощающих все этапы процесса разработки приложения. В Visual Studio использовался шаблон ASP.NET MVC, который представляет собой фреймворк для создания сайтов и веб-приложений с помощью реализации паттерна MVC. В ASP.NET MVC используется технология .NET Framework доступа к данным, известная как Entity Framework, которая дает возможность создавать и работать с классами сущностями. Поддерживает Entity Framework возможность использования парадигмы разработки Code First. Она позволяет создавать объекты модели путем написания простых классов. Затем можно создавать базу данных в режиме реального времени из классов моделей, которые позволяют очень просто и быстро организовать рабочий процесс.

1.2 Функциональные возможности

Разрабатываемое программное средство должно обеспечивать получение из базы данных всей необходимой информации в полном объеме, а также возможность её редактирования и удаления.

					МИВУ 09.03.04-01.000 ПЗ	Лист
						8
Из	Лист	№ докум.	Подп.	Дата		

На основании произведённого анализа предметной области были сформированы требования к разрабатываемой программе.

Функциональные требования программы:

- Реализовать в БД несколько типов номеров: Люкс, Полулюкс, одноместный и двухместный;
- Реализовать возможность бронирования номеров;
- Реализовать хранение данных о клиентах;
- Реализовать возможность добавления в БД новых услуг и редактирование старых;
- Реализовать поиск свободного номера на требуемые даты по указанным параметрам;
- Реализовать вывод сообщений при некорректном вводе данных;
- Предусмотреть хранение изображений
- Реализовать авторизацию пользователей (хранений хранение пользователей в отдельной БД);
- Реализовать формирование отчетов в виде файлов Excel таблиц;
- Реализовать отправку уведомлений администратору, с отчётами.

					МИВУ 09.03.04-01.000 ПЗ	Лист
						9
Из	Лист	№ докум.	Подп.	Дата		

2 Разработка моделей данных

Этот этап является самым важным при создании автоматизированной информационной системы. Изначально следует создать концептуальную модель базы данных, которая наглядно демонстрирует предметную область, для которой проектируется система. Концептуальная модель не является полностью подходящей для разработки в дальнейшем, поэтому необходимо производить нормализацию всех таблиц для реляционной базы данных. Следующим этапом является построение логической модели, что является подобной концептуальной. Данные в таком представлении легко воспринимаются бизнес-пользователями. Эта модель позволяет точно учесть все аспекты исследуемой предметной области. Далее, на основе составленной модели необходимо спроектировать физическую модель данных, которая будет определять способы размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне, то есть все детали, которые необходимы для создания базы данных: наименования таблиц и столбцов, типы данных, определение первичных и внешних ключей и т.д.

2.1 Концептуальная модель данных

После выделения сущностей идет создание концептуальной модели. Целью ее создания является дальнейшее проектирование базы данных и перевод ее, например, в реляционную базу данных. Описание разрабатываемой системы представлено в разделе под номером 1. Имея всю необходимую информацию достаточно легко составить конечную концептуальную модель. Тем самым можно будет представить то, как будут взаимодействовать таблицы (связи между объектами), и какие данные будут они хранить.

В приложении 1 расположена концептуальная модель данных.

					МИВУ 09.03.04-01.000 ПЗ	Лист
						10
Из	Лист	№ докум.	Подп.	Дата		

2.2 Логическая модель данных

Целью построения логической модели является получение графического представления логической структуры исследуемой предметной области. Логическая модель предметной области иллюстрирует сущности, а также их взаимоотношения между собой. Сущности описывают объекты, являющиеся предметом деятельности предметной области, и субъекты, осуществляющие деятельность в рамках предметной области. Свойства объектов и субъектов реального мира описываются с помощью атрибутов. Взаимоотношения между сущностями иллюстрируются с помощью связей. Правила и ограничения взаимоотношений описываются с помощью свойств-связей. Обычно связи определяют либо зависимости между сущностями, либо влияние одной сущности на другую. В приложении 1 представлена логическая модель данных. Сущность «Клиент» определяет всю необходимую информацию о клиенте, которую должен знать администратор гостиницы при бронировании или регистрации проживания. Согласно интервью с заказчиком, любое проживание или бронь оформляются на одного человека, даже если потенциальных постояльцев при этом будет несколько. Данные об этих постояльцах не хранятся в базе данных. Эта сущность связывает сущности «Проживание» и «Бронирование». В сущности «Бронирование» содержатся данные предварительного проживания. В сущности «Проживание» содержатся данные текущего или оконченного проживания, заносимые при непосредственной регистрации клиента. Сущность «Номера» также, как и сущность «Клиент» связана с сущностями «Проживание» и «Бронирование». В ней хранятся данные о номерах, представленных в гостинице. В связанной с ней сущности «Фотографии» располагаются фотографии каждого номера, при их наличии.

					МИВУ 09.03.04-01.000 ПЗ	Лист
						11
Из	Лист	№ докум.	Подп.	Дата		

2.3 Физическая модель данных

Физическая модель данных составлена на основе логической модели, где уже отражены и учтены все основные аспекты исследуемой предметной области. После проектирования этой модели будет осуществляться непосредственно создание базы данных и ее таблиц. На этом этапе процесс проектирования будет окончен. В приложении 1 представлена физическая модель данных создаваемой АИС.

Определим объем внешней памяти, необходимой для размещения данных за год использования. Для того чтобы оценить объем, занимаемый таблицами базы данных, необходимо оценить объем каждой таблицы. Примерный расчет необходимо объема памяти приведен в таблице 9.

Таким образом, при максимальном заполнении БД объем таблиц составит:
 $V_{\text{данных}} = 341 \text{ байт}$.

Таблица 1

Расчет объема ПЗУ для хранения данных

Таблица	Размер записи, байт	Максимальное (оценочное) количество записей	Всего, Кбайт
Booking	60	1000	58,59
Guest	44	5000	214,84
Client	60	1000	58,59
AdditionalServices	20	5000	97,66
Services	28	20	0,54
Apartment	28	100	2,73
Living	84	1000	82,03
Итого		25610	514,19

3 Проектирование работы системы

Проектирование системы направлено на представление системы, соответствующее предусмотренной цели, принципам и замыслам; оно включает оценку и принятие решений по выбору таких компонентов системы, которые отвечают её архитектуре и укладываются в предписанные ограничения. Результатом проектирования является проект — целостная совокупность моделей, свойств или характеристик, описанных в форме, пригодной для реализации системы. На данном этапе было создано несколько диаграмм: диаграмма вариантов использования, диаграммы SADT («сущность-связь») и DFD (диаграммы потоков данных).

3.1 Диаграммы UML

Диаграмма вариантов использования

Создав диаграмму вариантов использования, которая представлена на рисунке 1, были сформулированы общие требования к функциональному поведению проектируемой системы и определены общие границы моделируемой предметной области.

Актерами на данной диаграмме являются администратор и постоялец, они — внешняя часть системы. Администратор имеет доступ ко всем реализованным функциям. Постоялец имеет доступ только к тем данным, которые относятся к его проживанию или бронированию.

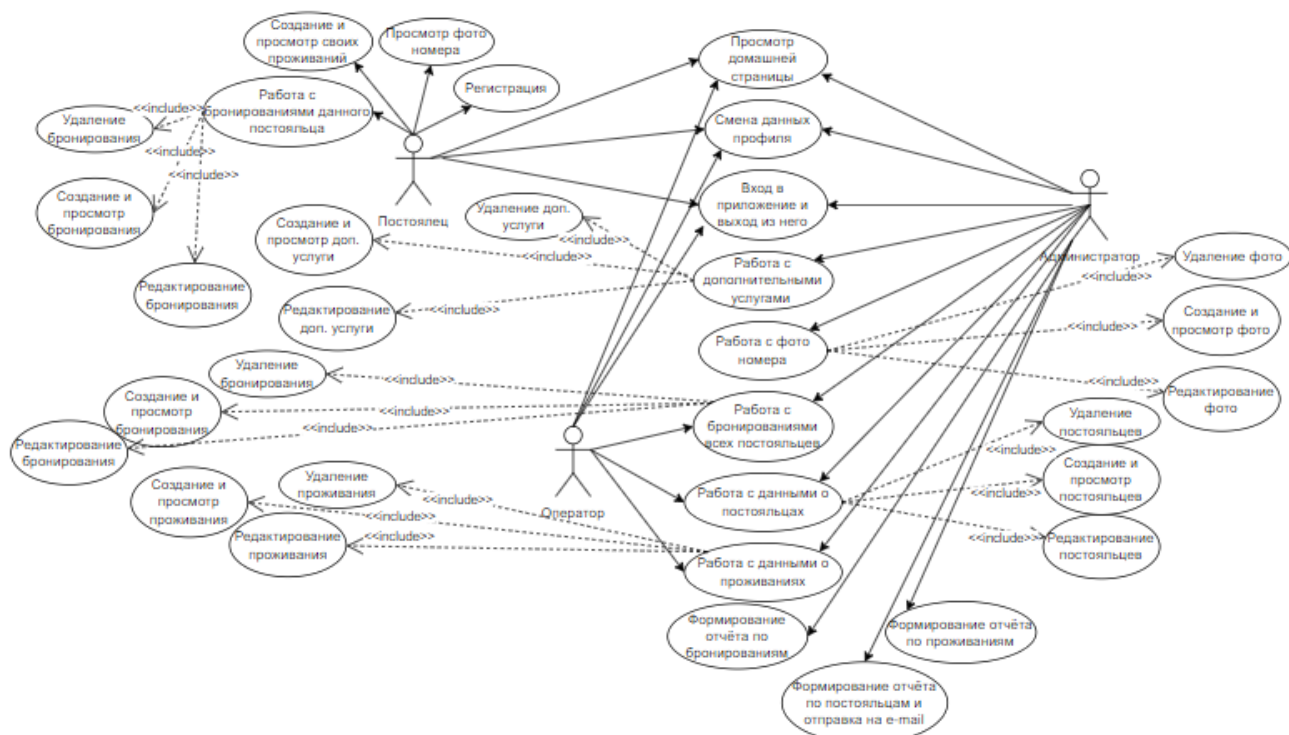


Рисунок 1 – Диаграмма вариантов использования

3. 2 Диаграммы DFD

В соответствии с DFD методологией, модель системы определяется как иерархия диаграмм потоков данных, описывающих процессы преобразования информации от момента ее ввода в систему до выдачи конечному пользователю. Для такого описания модели системы была создана диаграмма потоков данных, которая показана на рисунке 2.

Центральной здесь является работа АИС гостиницы. На ее вход поступают данные по номерам, постояльцам и их гостям, проживаниям, бронированиям, пользователям и услугам. Выходом этой работы будут отчеты по проживаниям и бронированиям за определённый период и постояльцам, которые поступают администратору. Для раскрытия диаграммы потоков данных была построена ее подробная диаграмма. Она показана на рисунке 3.

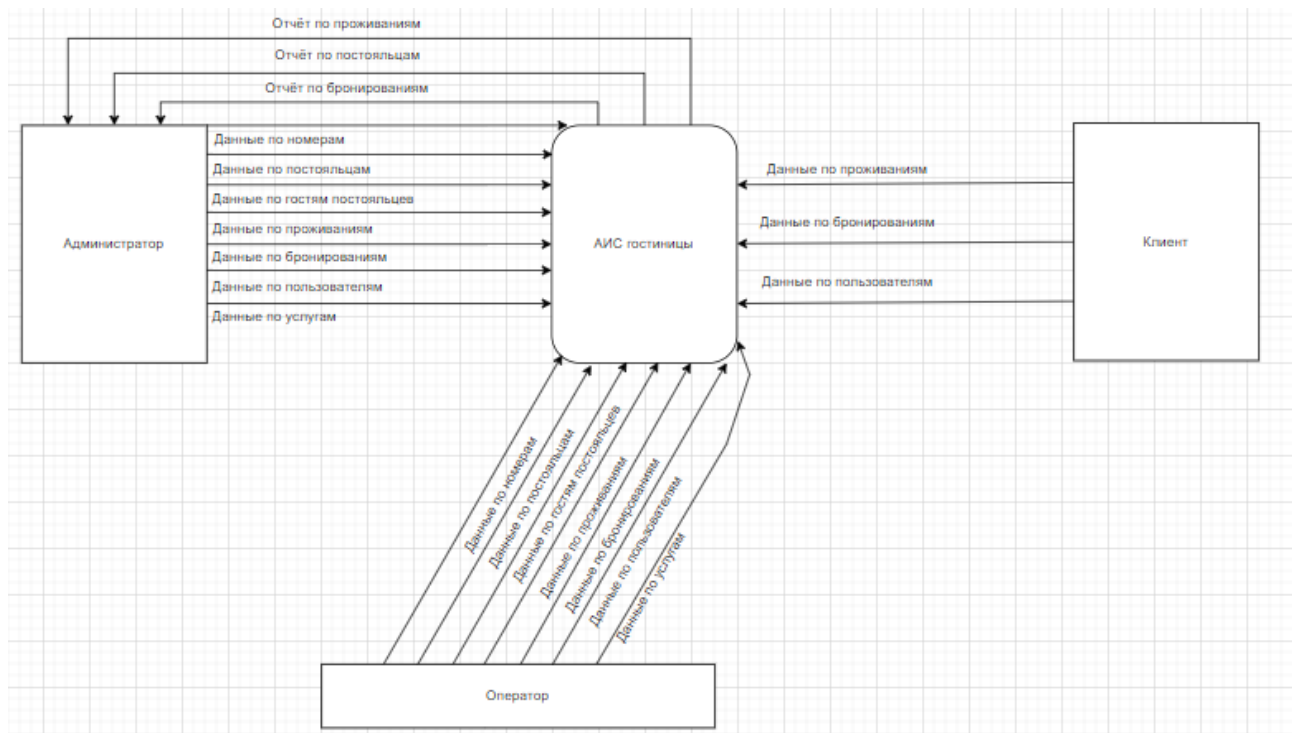


Рисунок 2 – Диаграмма потоков данных АИС гостиницы

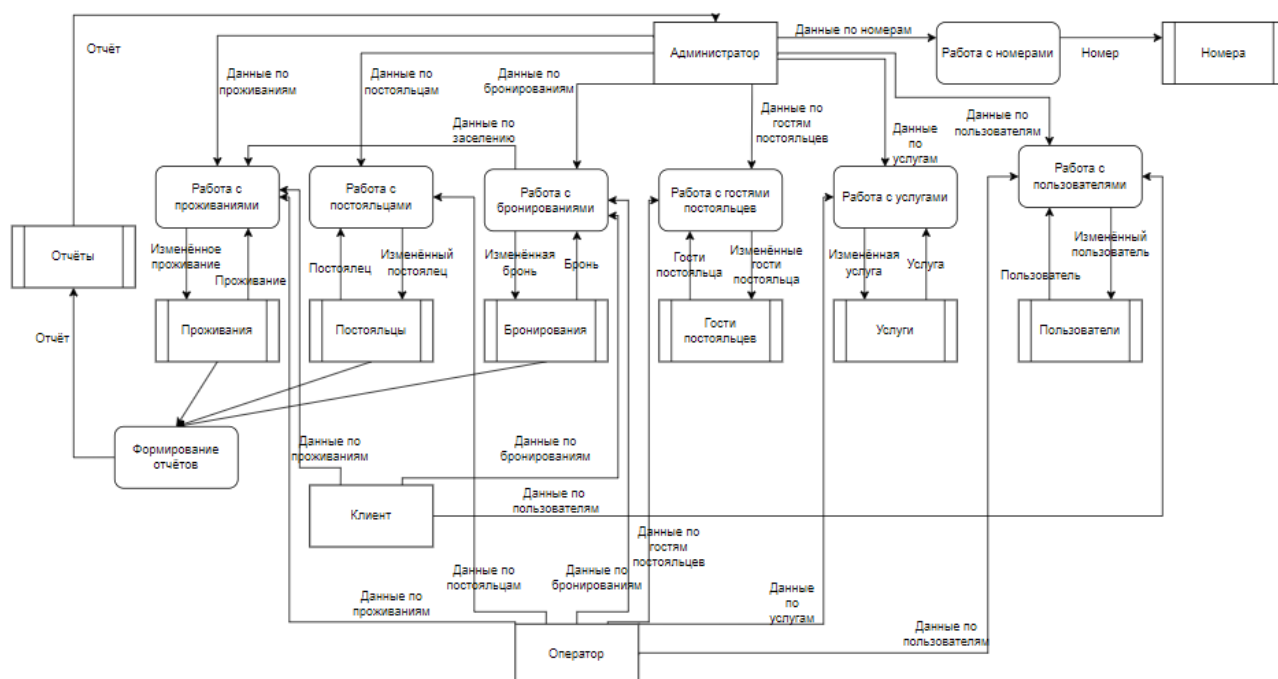


Рисунок 3 – Подробная диаграмма потоков данных

На данной диаграмме подробно расписаны все перемещения данных по системе.

3.3 Диаграммы последовательностей

Построим диаграммы последовательностей, которые позволяют изобразить поведение нескольких объектов в рамках одного прецедента. Диаграмм последовательностей должно быть столько же, сколько и вариантов использования программы, но так как работа со многими из них интуитивно понятна или схожа с остальными, то представим лишь некоторые из них.

Ниже, на рисунке 4, изображена диаграмма последовательности работы с проживаниями.

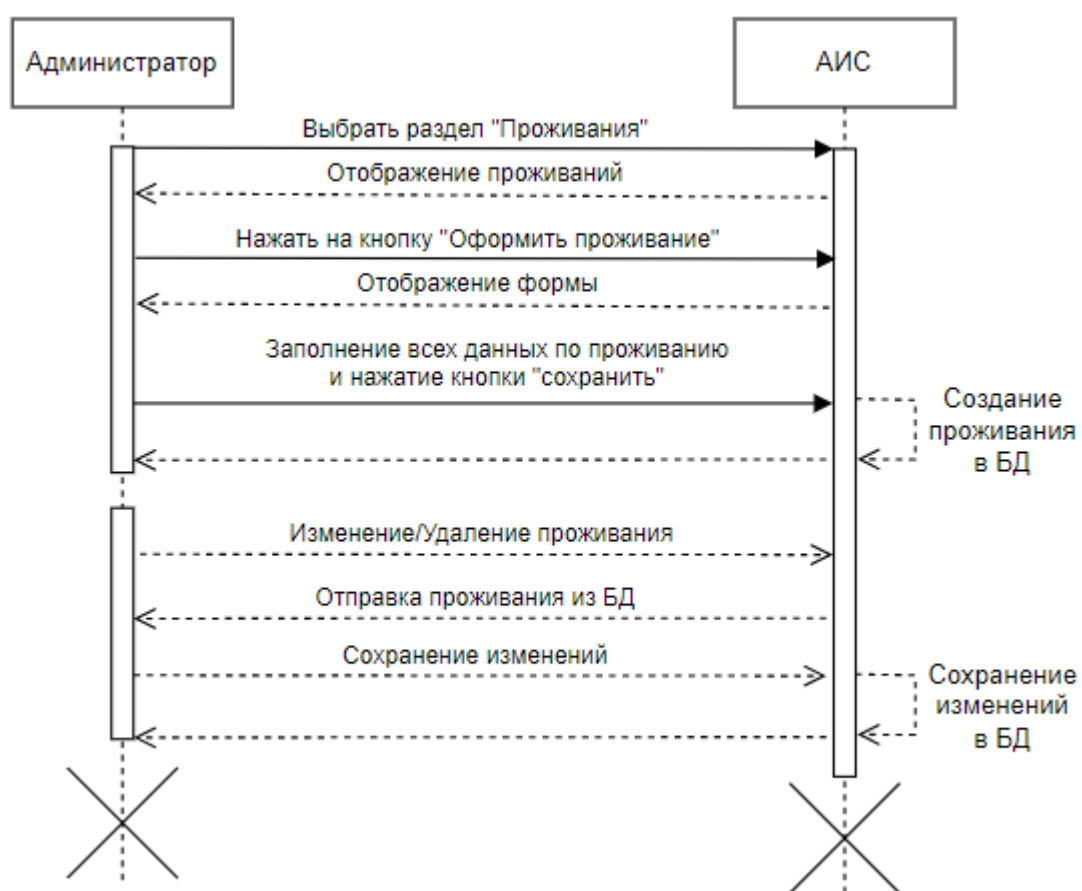


Рисунок 4 – Диаграмма последовательности работы с проживаниями

На данной диаграмме администратор создает номер, после чего АИС сохраняет ее в БД. При просмотре номеров клиентом АИС отправляет ему список номеров. Администратор также может изменять или удалять номер, при этом АИС отправляет ему этот номер из БД, после чего происходит сохранение изменений.

Ниже, на рисунке 5, показана диаграмма последовательности работы с отчётами.

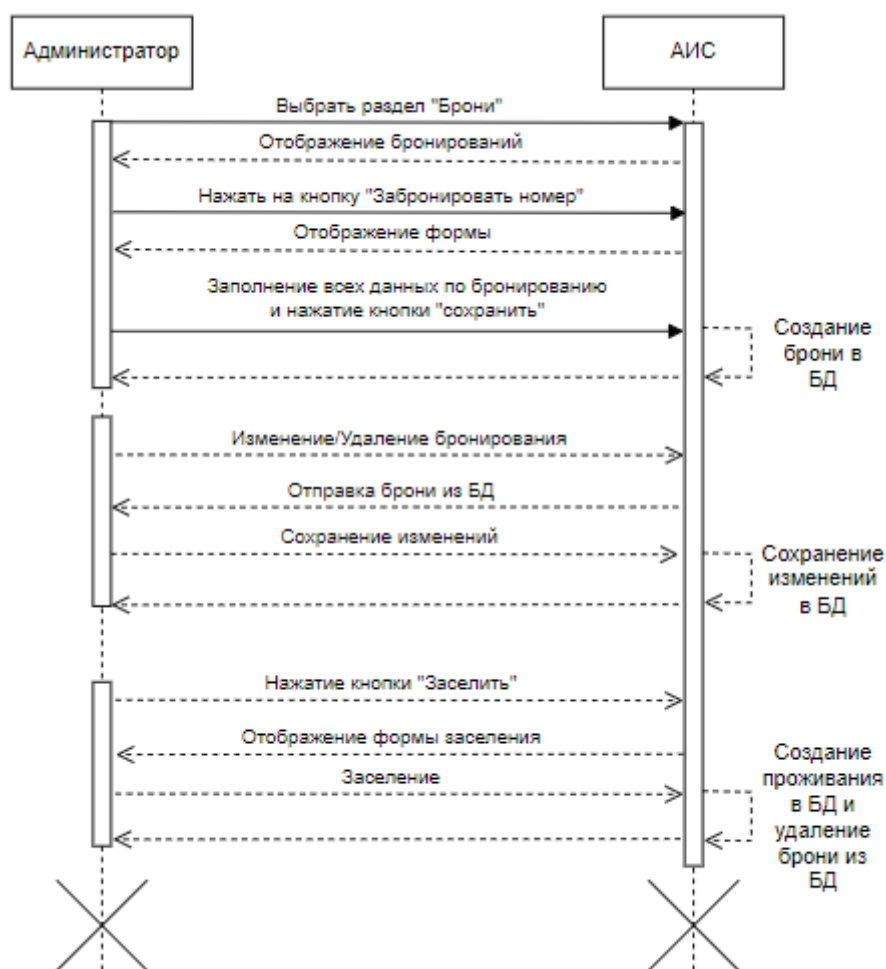


Рисунок 5 – Диаграмма последовательности работы с бронированиями

Работа с бронированиями достаточно похожа на работу с проживаниями.

Ниже, на рисунке 6, показана диаграмма последовательности работы с отчётами.

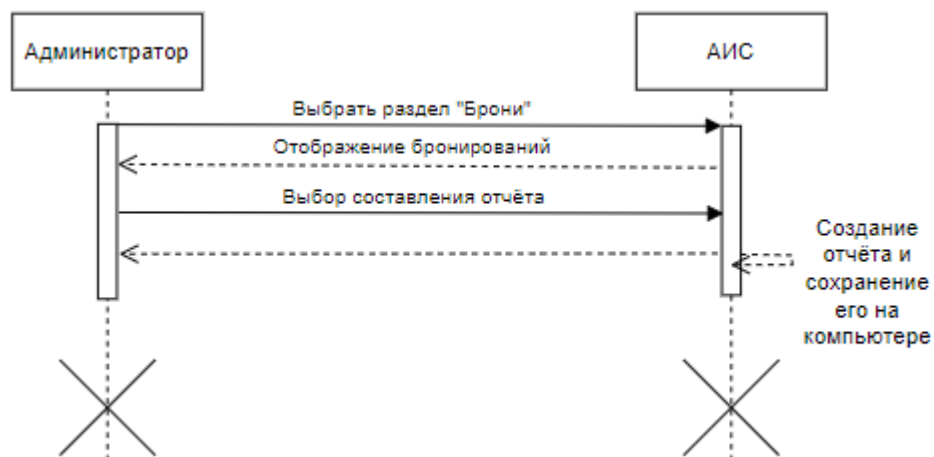


Рисунок 6 – Диаграмма последовательности работы с отчётами

Программа сама создаёт отчёты по бронированиям и отправляет их пользователю.

3.4 Диаграммы SADT

SADT, как и другие методологии проектирования, целесообразно использовать на ранних этапах жизненного цикла: для понимания системы до ее воплощения. Созданная функциональная диаграмма показана на рисунке 7.

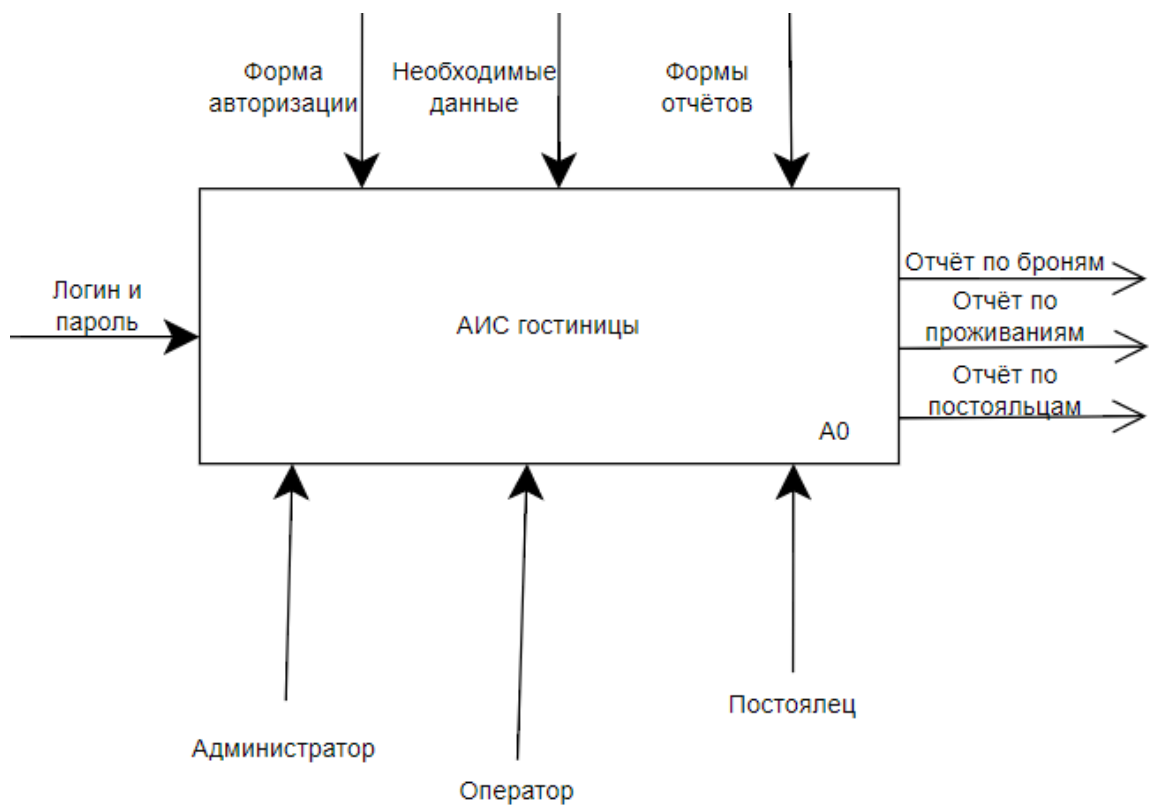


Рисунок 7 – Функциональна диаграмма АИС гостиницы

Для получения подробной информации о функционировании АИС необходимо создать подробную функциональную диаграмму. Она показана на рисунке 8.

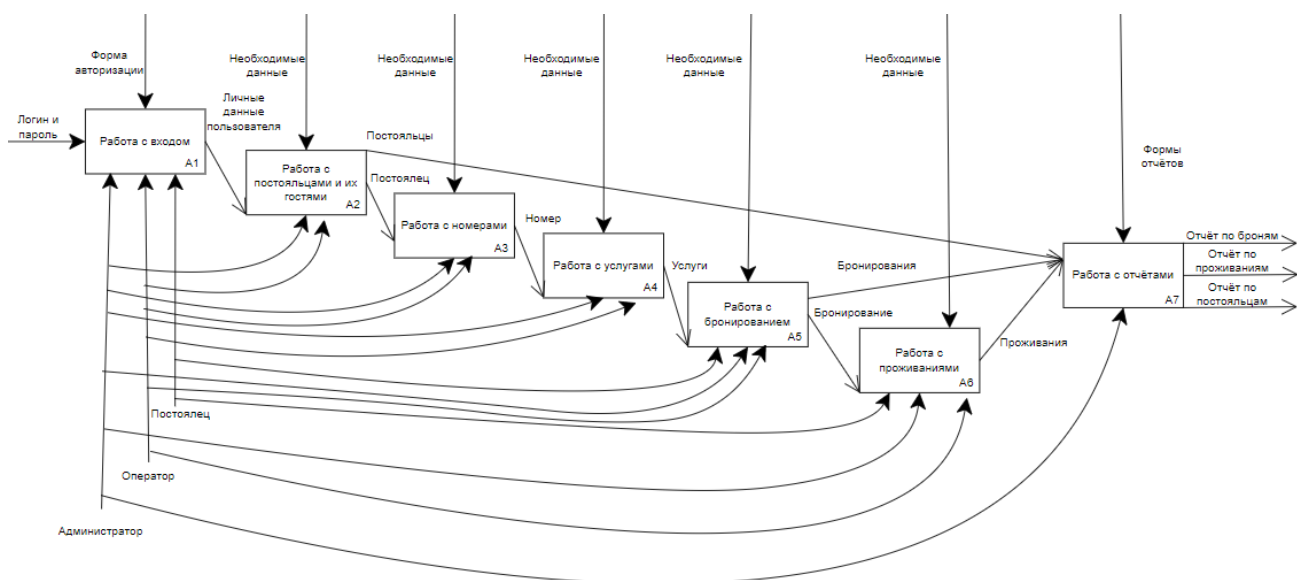


Рисунок 8 - Подробная функциональная диаграмма функционирования АИС гостиницы

4 Разработка и реализация АИС

После разработки моделей данных была создана сама база данных и их классов моделей.

Так как основу программы составляет взаимодействие с базой данных, то рассмотрим его на примере работы с номером.

4.1 Модели данных

Рассмотрим работу приложения на примере работы с номерами. Листинг класса Apartment.cs:

```
public class Apartment
{
    public int Id { get; set; }

    [Display(Name = "Номер")]
    [Required(ErrorMessage = "Введите номер")]
    [StringLength(4, MinimumLength = 1)]
    public string Number { get; set; }

    [Display(Name = "Тип номера")]
    [Required(ErrorMessage = "Введите тип номера")]
    [StringLength(50, MinimumLength = 3)]
    public string Type { get; set; }

    [Display(Name = "Стоимость")]
    [Required(ErrorMessage = "Введите цену")]
    [Range(100, 1000000)]
    [DataType(DataType.Currency)]
    public decimal Price { get; set; }

    [Display(Name = "Максимальное количество гостей")]
    [Required(ErrorMessage = "Введите максимальное количество гостей")]
    [Range(0, 15)]
    public int MaxGuests { get; set; }

    public List<Photo> Photos { get; set; }
    public List<Living> Living { get; set; }
    public List<Booking> Booking { get; set; }

    public Apartment()
    {
        Photos = new List<Photo>();
        Living = new List<Living>();
        Booking = new List<Booking>();
    }
}
```

Данный класс соответствует таблице «Apartment» в базе данных.

					МИВУ 09.03.04-01.000 ПЗ	Лист
						20
Из	Лист	№ докум.	Подп.	Дата		

Перед полями, которые заносятся в БД, указываются аннотации данных – инструмент языка для настройки сущностей с помощью атрибутов.

[Key] – атрибут, который указывается для установки поля в качестве первичного ключа.

[Display] – атрибут, который задает параметры отображения свойства.

[MinLenght] и [MaxLenght] – атрибуты, которые устанавливают минимальное и максимальное количество символов в строке.

[RegularExpression] – атрибут, который используется для проверки соответствия шаблону.

[Required] – атрибут, который указывает, что данное свойство обязательно для установки, то есть будет иметь определение NOT NULL в БД.

[Range] – атрибут, который позволяет указывать диапазон значений.

[ForeignKey] – атрибут, который указывает внешний ключ для навигационного свойства.

У некоторых из этих атрибутов есть свойство “ErrorMessage”, которое позволяет указывать текст сообщения об ошибке.

4.2 Вывод всех номеров

Листинг метода Index():

```
public ActionResult Index()
{
    return View(db.Apartments.ToList());
}
```

В данном методе во Views передаются списком все номера.

4.3 Добавление номера

При добавлении номера есть 2 метода: с GET и POST запросами. GET запрос отображает нужную модель для ввода данных, а POST запрос обрабатывает данные, которые пользователь ввел.

Листинг метода Create() GET запроса:

```
[Authorize(Roles = "Admin")]
public ActionResult Create()
{
    return View();
}
```

Перед методом есть скобка [Authorize(Roles = "Admin")], которая отвечает за то, что данный метод доступен только авторизованному пользователю под ролью Администратор. Сам метод возвращает View модель, которая отвечает за добавление номера.

Листинг метода Create() POST запроса:

```
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Admin")]
public ActionResult Create([Bind(Include = "Id,Number,Type,Price,MaxGuests")]
Apartment apartment)
{
    if (ModelState.IsValid)
    {
        db.Apartments.Add(apartment);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(apartment);
}
```

Атрибут [ValidateAntiForgeryToken] предназначен для противодействия подделке межсайтовых запросов.

Атрибут [HttpPost] указывает на то, что данный метод может вызываться только для POST запросов.

Данный метод доступен также только администратору.

4.4 Изменение данных о номере

В изменении квартиры есть также GET и POST запросы. Только в данном случае GET запрос принимает на вход id квартиры, которую необходимо изменить.

Листинг метода Edit() Get запроса:

					МИВУ 09.03.04-01.000 ПЗ	Лист
						22
Из	Лист	№ докум.	Подп.	Дата		

```

[Authorize(Roles = "Admin")]
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Apartment apartment = dbApartments.Find(id);
    if (apartment == null)
    {
        return HttpNotFound();
    }
    return View(apartment);
}

```

Можно увидеть, что изменение номера доступно также только администратору. В методе реализована проверка на то, чтобы id изменяемого номера не был равен null, иначе выводится ошибка о том, что запрос имел неверный формат.

Листинг метода Edit() POST запроса:

```

[HttpPost]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Admin")]
public ActionResult Edit([Bind(Include = "Id,Number,Type,Price,MaxGuests")]
Apartment apartment)
{
    if (ModelState.IsValid)
    {
        db.Entry(apartment).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(apartment);
}

```

Если данные введены корректно — данные квартиры изменяются и пользователя возвращают к списку всех номеров.

4.5 Удаление номера

В удалении номера, как и всегда, участвуют два метода с разными запросами. Один из них GET, который возвращает нужную View модель с подтверждением удаления квартиры. Этот метод принимает на вход также как и при изменении — id номера.

В POST запросе обрабатывается подтверждение пользователем удаления квартиры. Он также получает на вход id соответствующего номера. После этого происходит непосредственно удаление и пользователя перенаправляют на страницу со списком всех номеров.

Листинг DeleteConfirmed() POST запроса:

```
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Admin")]
public ActionResult DeleteConfirmed(int id)
{
    Apartment apartment = dbApartments.Find(id);
    dbApartments.Remove(apartment);
    db.SaveChanges();
    return RedirectToAction("Index");
}
```

4.6 Формирование отчетов

В данном пункте будет рассмотрен пример формирования отчета по проживаниям за последние 30 дней.

Листинг Index() GET запроса:

```
[Authorize(Roles = "Admin, Operator")]
public ActionResult LivingStat()
{
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial;
    //Открытие существующего файла
    FileInfo fi = new
    FileInfo(@"E:\Саня\Лаб\ARM_Hotel\Content\Reports\LivingsReport.xlsx");
    using (ExcelPackage excelPackage = new ExcelPackage(fi))
    {
        //Получение листа по имени. Если лист с таким именем не существует -
        //будет сгенерировано исключение
        ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets["Лист1"];
        //Занесение информации
        TimeSpan ts = new TimeSpan(-30, 0, 0, 0, 0);
        DateTime settling = DateTime.Now.Date + ts;
        List<Living> livings = db.Livings
            .Include(m => m.Client)
            .Where(l => l.Settling > settling).ToList();

        int i = 3;
        foreach (var living in livings)
        {
            worksheet.Cells[i, 1].Value = i - 2;
            worksheet.Cells[i, 2].Value = living.Client.LastName + " " +
            living.Client.FirstName;
            worksheet.Cells[i, 3].Value = living.Settling;
            worksheet.Cells[i, 4].Value = living.Eviction;
            worksheet.Cells[i, 5].Value = living.Type;
            worksheet.Cells[i, 6].Value = Convert.ToInt32(living.Number);
        }
    }
}
```

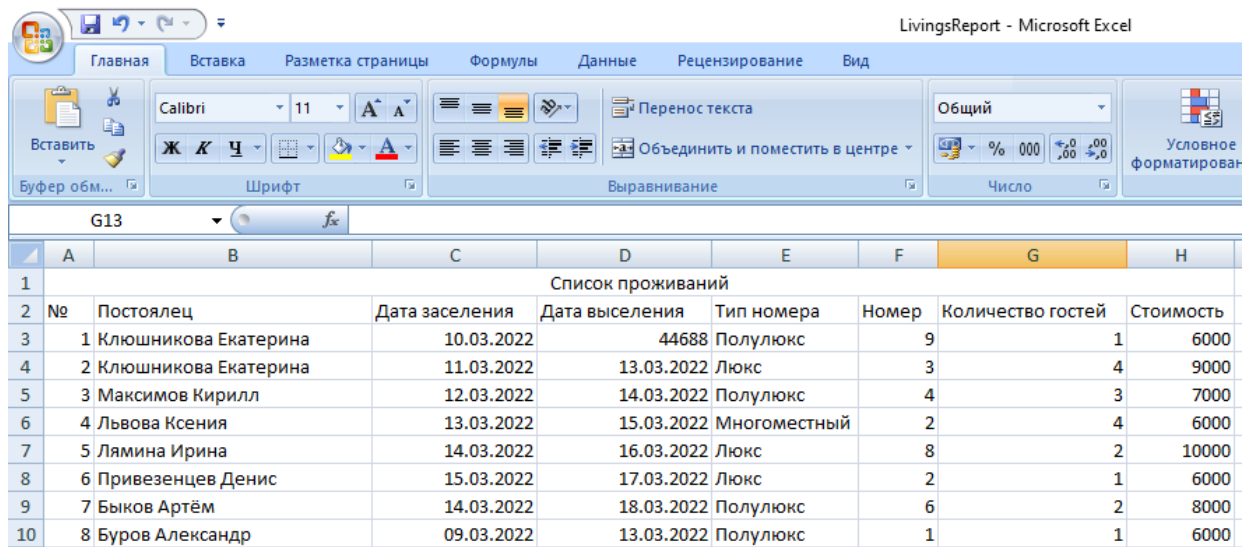
					МИВУ 09.03.04-01.000 ПЗ	Лист
Из	Лист	№ докум.	Подп.	Дата		24

```

        worksheet.Cells[i, 7].Value = living.ValueOfGuests;
        worksheet.Cells[i, 8].Value = living.Price;
        i++;
    }
    //Сохраняем файл
    excelPackage.Save();
}
return RedirectToAction("Index", "Livings");
}

```

В методе создается экземпляр класса FileInfo, который работает с нашим документом. После чего задается автор документа, его тема и название. Далее идет заполнение ячеек файла. Отчет создается с расширением .xlsx. Ниже представлен пример сформированного отчета.



№	Постоялец	Дата заселения	Дата выселения	Тип номера	Номер	Количество гостей	Стоимость
1	Клюшникова Екатерина	10.03.2022	44688	Полулюкс	9	1	6000
2	Клюшникова Екатерина	11.03.2022	13.03.2022	Люкс	3	4	9000
3	Максимов Кирилл	12.03.2022	14.03.2022	Полулюкс	4	3	7000
4	Львова Ксения	13.03.2022	15.03.2022	Многоместный	2	4	6000
5	Лямина Ирина	14.03.2022	16.03.2022	Люкс	8	2	10000
6	Привезенцев Денис	15.03.2022	17.03.2022	Люкс	2	1	6000
7	Быков Артём	14.03.2022	18.03.2022	Полулюкс	6	2	8000
8	Буров Александр	09.03.2022	13.03.2022	Полулюкс	1	1	6000

Рисунок 11 - Пример сформированного отчета

5. Тестирование

Целью проведения тестирования является подтверждение реализации требуемой функциональности системы. В процессе тестирования, результат каждого теста фиксируется. Считается, что тест прошел успешно в случае, если результат совпадает с ожидаемым результатом. Если результат отличается от ожидаемого, он фиксируется в протоколе тестирования.

На всех формах, где пользователь вводит данные, присутствуют проверки, которые срабатывают до нажатия на кнопки. Это аннотации, которые были описаны до этого в пункте 4.1.

Таблица 2- Методика тестирования разработанной программы

Выполненное действие	Полученный результат
Запустить программу	Отображается домашняя страница.
Регистрация нового клиента	Внесение данных клиента в базу данных.
Бронирование номера	Создание записи в соответствующей таблице в БД.
Заселение в номер	Создание записи в соответствующей таблице в БД.
Запрос на отображение детальной информации записи любой из таблиц	Отображение информации, содержащейся в этой записи в лицеприятном виде.
Попытка регистрации клиента с незаполненными данными	Выводится сообщение об ошибке «Все данные должны быть заполнены».
Удаление номера, в котором ещё проживают люди/ в котором собираются проживать люди	Вывод сообщения о невозможности данного действия и отсутствие кнопки «Удалить».
Удаление номера, в котором никто не проживает в данный момент	Удаление соответствующей записи с номером из БД.
Добавление нового номера	Создание записи в соответствующей таблице в БД.
Вход в аккаунт	Отображение особой домашней страницы
Попытка добавления большего количества гостей, чем позволяет номер	Выводится сообщение об ошибке: «Превышено количество постояльцев в данном номере».

Выполненное действие	Полученный результат
Нажатие на кнопку «Заселить выбранного клиента»	Происходит перенос данных из таблицы «Бронирования» в таблицу «Проживания».
Попытка зарегистрироваться в приложении	Добавление нового аккаунта и доступ к входу по зарегистрированным данным
Добавление услуги	Создание записи в БД с соответствующей услугой

Результаты, полученные в ходе тестирования разработанного программного продукта, позволяют сделать заключение в том, что разработанная программа соответствует требованиям технического задания.

Заключение

В данной курсовой работе в соответствии с заданием была разработана АИС администратора гостиницы для автоматизации предметной области по технологии ASP.NET MVC 5.

В ходе выполнения курсовой работы были выполнены следующие задачи:

- выявлены требования к программе;
- разработаны модели данных;
- разработана программа;
- осуществлено ее тестирование.

Разработанная программа обеспечивает осуществление следующих функций:

- Возможность бронирования номеров;
- Хранение данных о клиентах;
- Возможность добавления в БД новых услуг и редактирование старых;
- Поиск свободного номера на требуемые даты по указанным параметрам;
- Авторизация пользователей (хранение пользователей в отдельной БД);
- Отправка уведомлений квартиросъемщикам, которые не передали показания, в виде e-mail писем.
- Формирование отчетов в формате .xlsx.

Внедрение данного приложения в работу предприятия положительно повысит производительность труда и упростит работу администратора гостиницы. Таким образом, в курсовой работе были реализованы все пункты технического задания, а в перспективе список функциональных возможностей системы может расширяться.

Список литературы

1. Руководство по Entity Framework [сайт]. URL: <https://metanit.com/sharp/entityframework/>
2. Работа с Entity Framework [сайт]. URL: <https://professorweb.ru/my/entity-framework/6/level1/>
3. Аннотации в Entity Framework [сайт]. URL: <https://docs.microsoft.com/en-us/ef/ef6/modeling/code-first/data-annotations>
4. Столбовский, Д. Н. Разработка Web-приложений ASP.NET с использованием Visual Studio .NET: учебное пособие / Д. Н. Столбовский. — 3-е изд. — Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 375 с.
5. Фримен Адам. ASP.NET MVC 5 с примерами на C# 5.0 для профессионалов : Вильямс, 2018, 736с.
6. Brian L Gorman. Practical Entity Framework: Database Access for Enterprise Applications. — Apress, 2020 — 433pp.
7. Эспозито Дино Разработка современных веб-приложений. Анализ предметных областей и технологий. — Вильямс, 2017, 464с.
8. Span<T> Структура [сайт]. <https://docs.microsoft.com/ru-ru/dotnet/api/system.span-1?view=net-6.0>

Приложение 1. Модели данных

1. Концептуальная модель данных

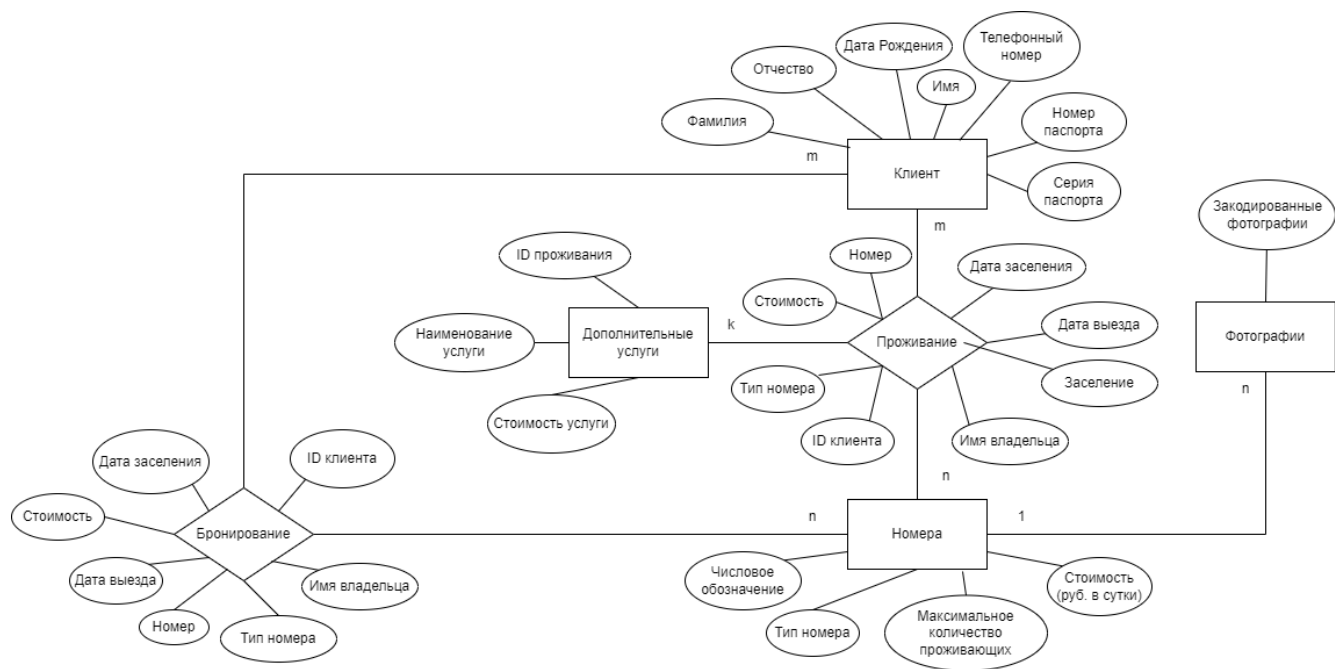


Рисунок 12 - Концептуальная модель данных

2. Логическая модель данных

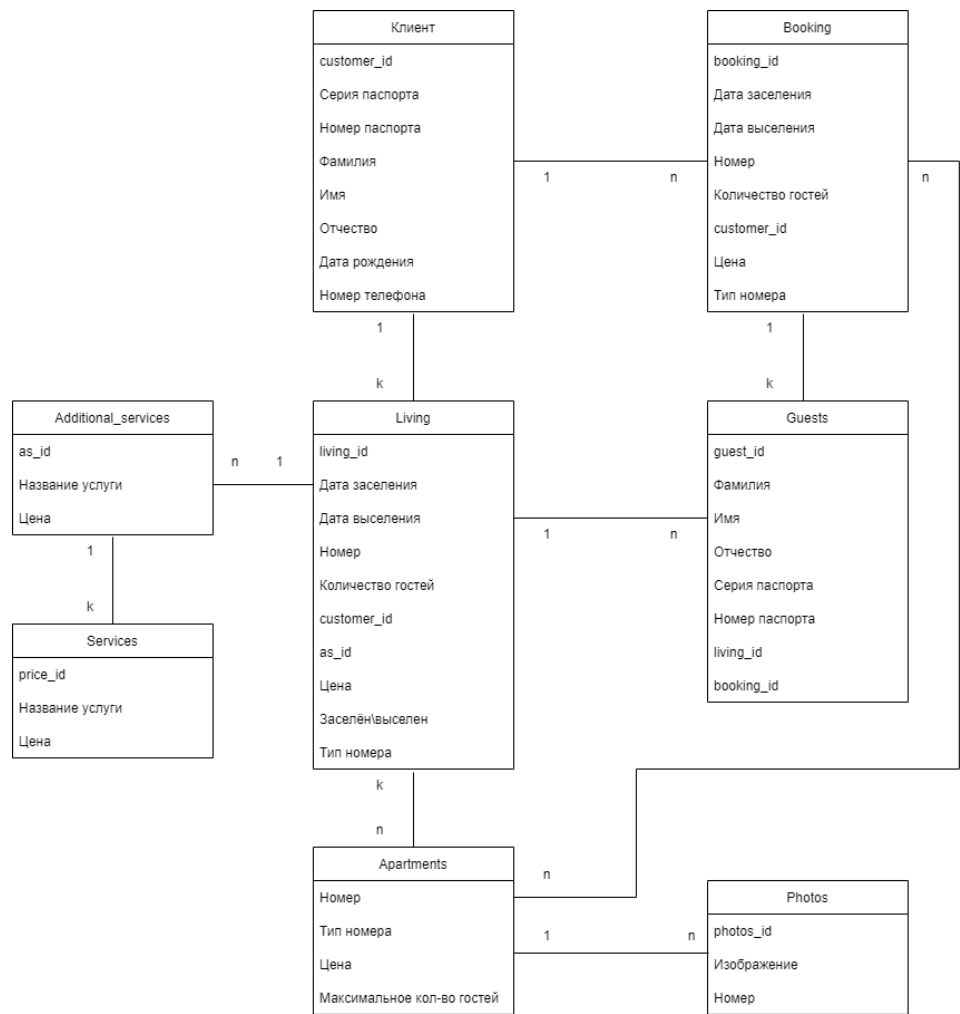


Рисунок 13 - Логическая модель данных

3. Физическая модель данных

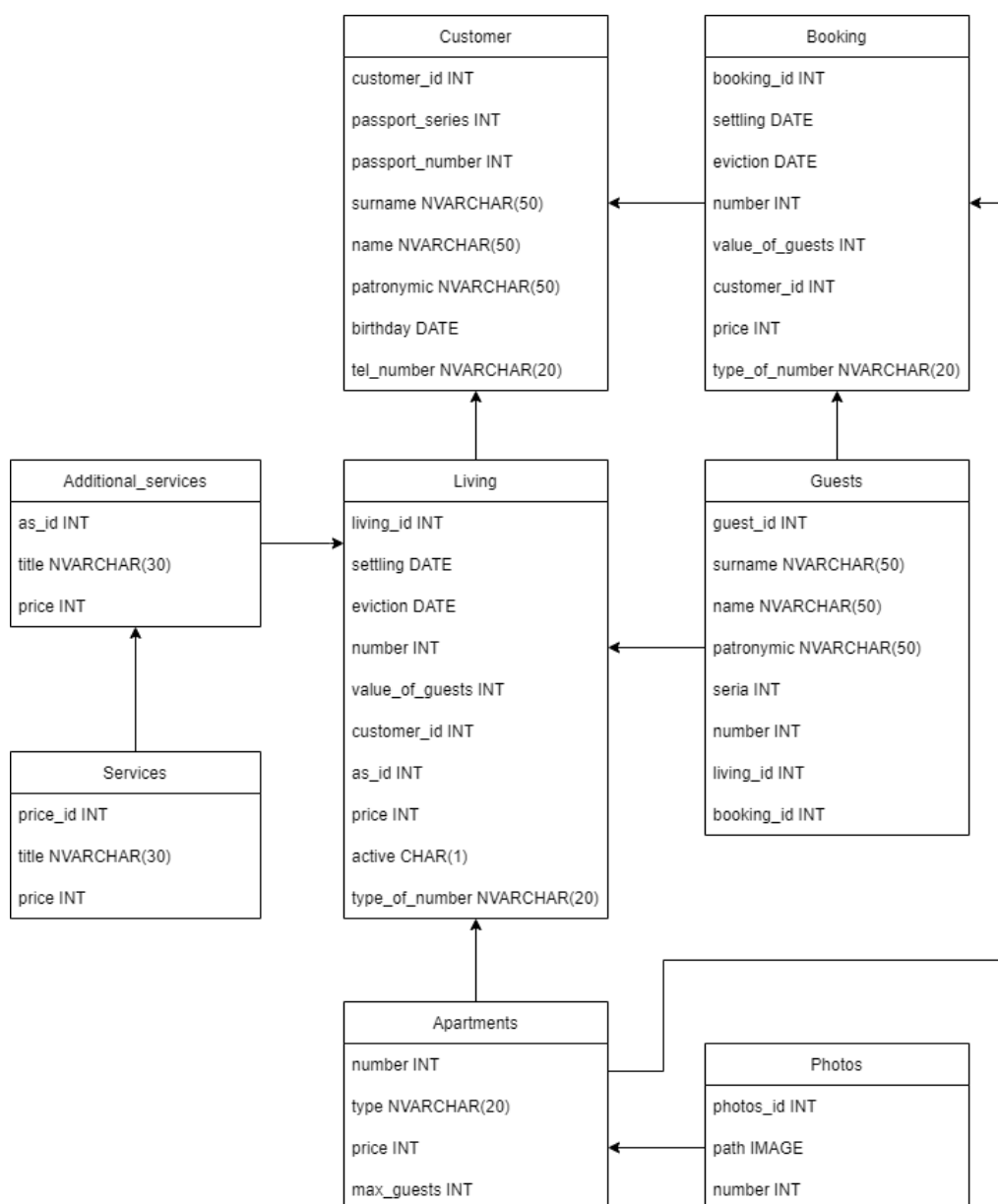


Рисунок 14 - Физическая модель данных

Приложение 2. Снимки окон программы (скриншоты программы)

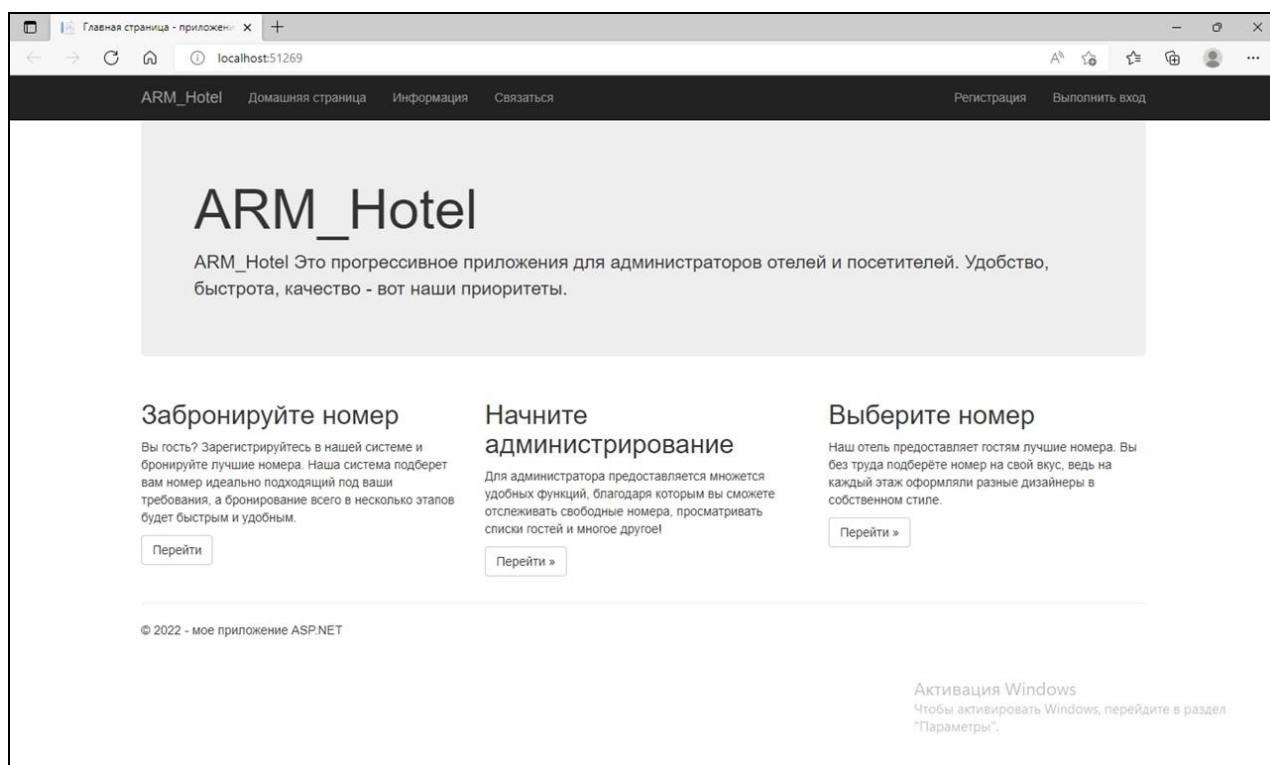


Рисунок 15 – Домашняя страница приложения

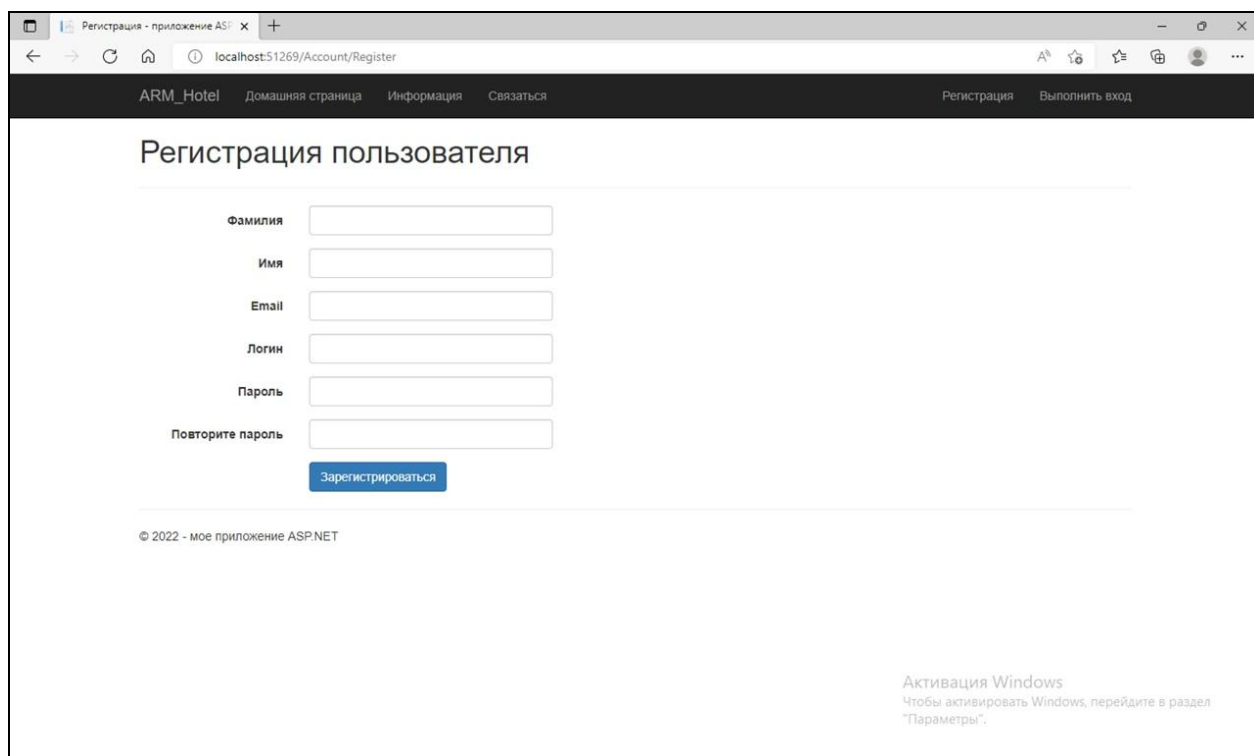


Рисунок 16 – Регистрация пользователя

Из	Лист	№ докум.	Подп.	Дата

МИВУ 09.03.04-01.000 ПЗ

Лист

33

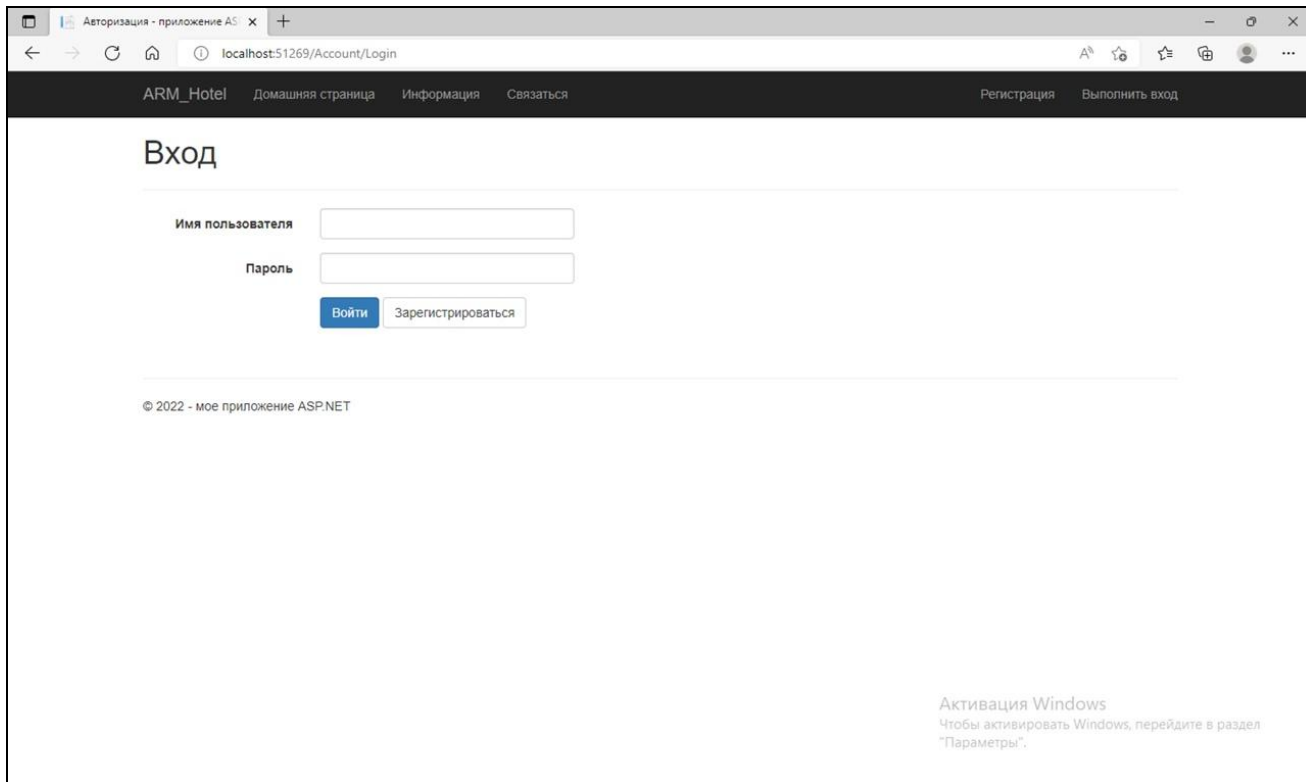


Рисунок 17 – Вход в аккаунт

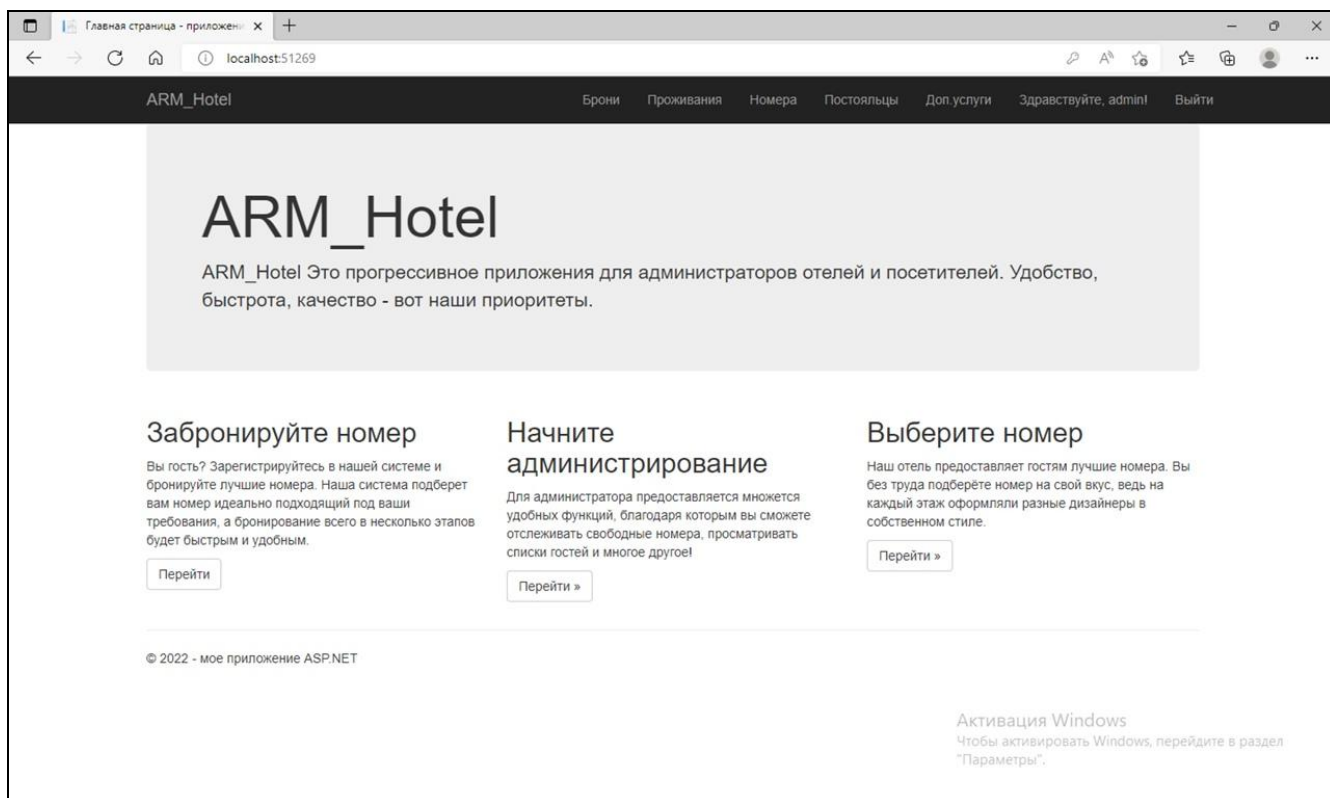


Рисунок 18 – Домашняя страница после входа в аккаунт администратора

Из	Лист	№ докум.	Подп.	Дата

МИВУ 09.03.04-01.000 ПЗ

Лист

34

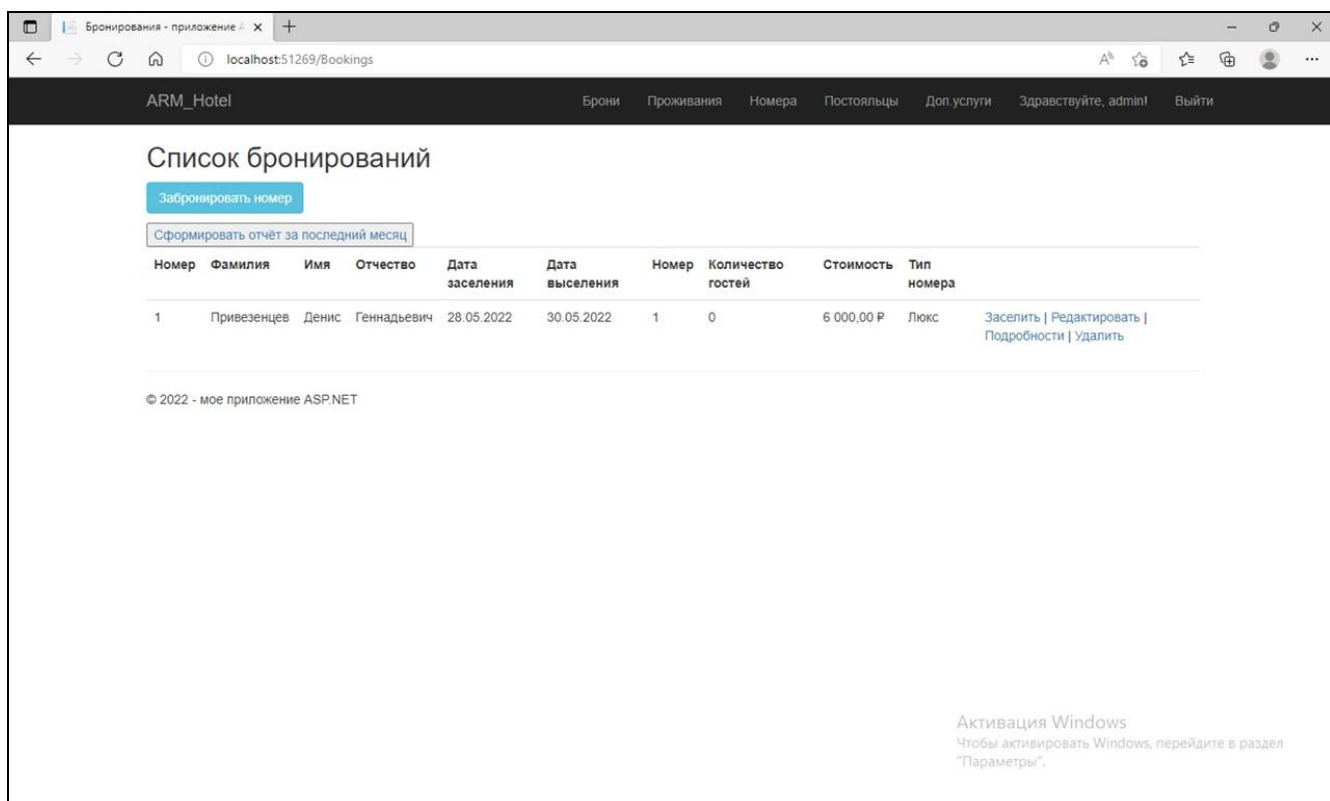


Рисунок 19 – Таблица бронирований

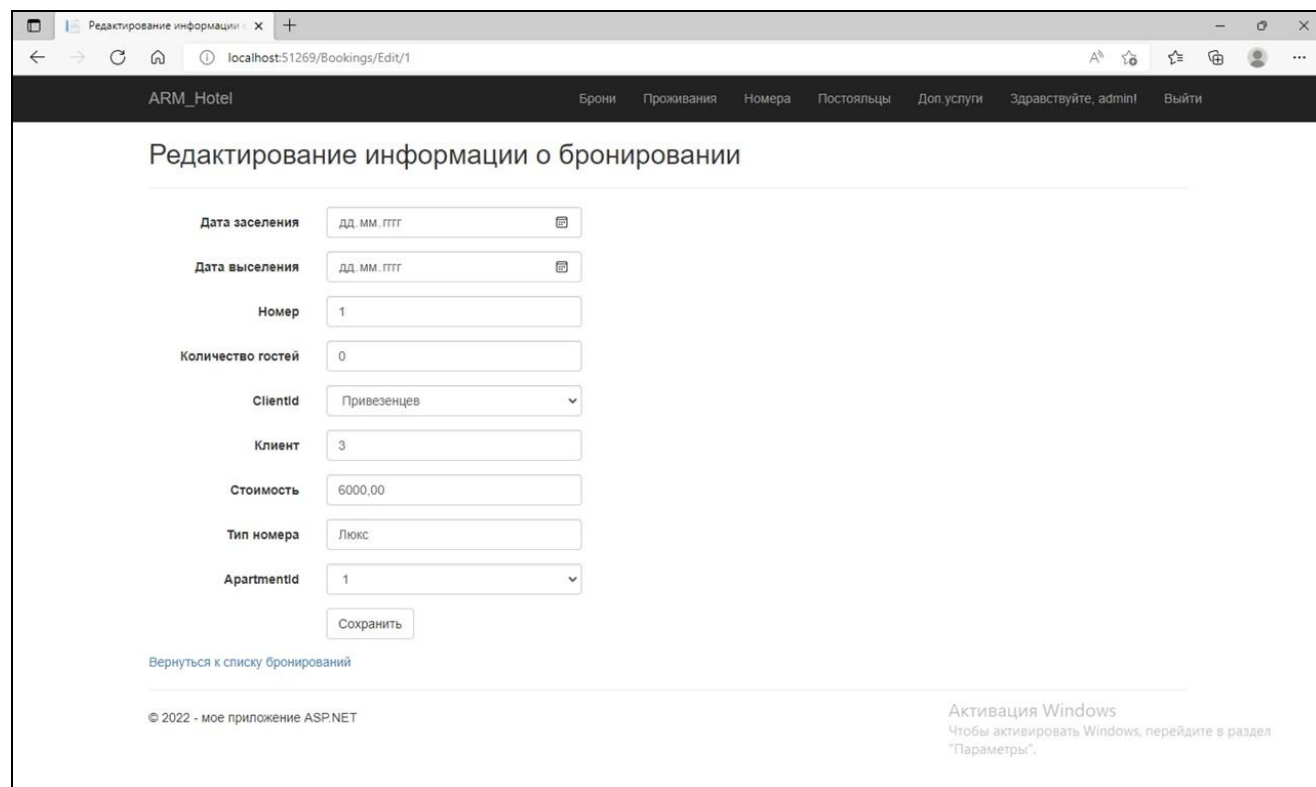


Рисунок 20 – Редактирование информации о бронировании

ARM_Hotel									
Брони Проживания Номера Постояльцы Доп. услуги Здравствуйте, admin! Выйти									
Список проживаний									
Оформить проживание									
Сформировать отчёт за последний месяц									
Фамилия	Имя	Отчество	Дата заселения	Дата выселения	Номер	Количество гостей	Стоимость	Тип номера	
Ключникова	Екатерина	Александровна	05.05.2022	07.05.2022	1	1	6 000,00 Р	Люкс	Редактировать Подробности Удалить
Ключникова	Екатерина	Александровна	12.05.2022	14.05.2022	1	1	6 000,00 Р	Люкс	Редактировать Подробности Удалить
Ключникова	Екатерина	Александровна	09.05.2022	12.05.2022	2	2	9 000,00 Р	Люкс	Редактировать Подробности Удалить
Ключникова	Екатерина	Александровна	18.05.2022	21.05.2022	1	2	9 000,00 Р	Люкс	Редактировать Подробности Удалить
Привезенцев	Денис	Геннадьевич	18.05.2022	31.05.2022	2	0	39 000,00 Р	Люкс	Редактировать Подробности Удалить
Ключникова	Екатерина	Александровна	22.05.2022	24.05.2022	1	0	6 000,00 Р	Люкс	Редактировать Подробности Удалить
Максимов	Кирилл	Алексеевич	25.05.2022	27.05.2022	1	0	6 000,00 Р	Люкс	Редактировать Подробности Удалить
Петров	Вася	Иванович	04.06.2022	07.06.2022	1	0	9 000,00 Р	Люкс	Редактировать Подробности Удалить
Кукина	Анастасия	Владимировна	31.05.2022	01.06.2022	1	2	3 000,00 Р	Люкс	Редактировать Подробности Удалить

Рисунок 21 – Список проживаний

Номера - приложение ASP.NET

localhost:51269/Apartments

ARM_Hotel

Брони

Проживания

Номера

Постояльцы

Доп. услуги

Здравствуй, admin!

Выйти

Список номеров

Добавить новый номер

Номер	Тип номера	Стоимость	Максимальное количество гостей	
1	Люкс	3 000,00 Р	3	Редактировать Добавить фото Подробности Удалить
2	Люкс	3 000,00 Р	5	Редактировать Добавить фото Подробности Удалить

© 2022 - мое приложение ASP.NET

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".

Рисунок 22 – Список номеров

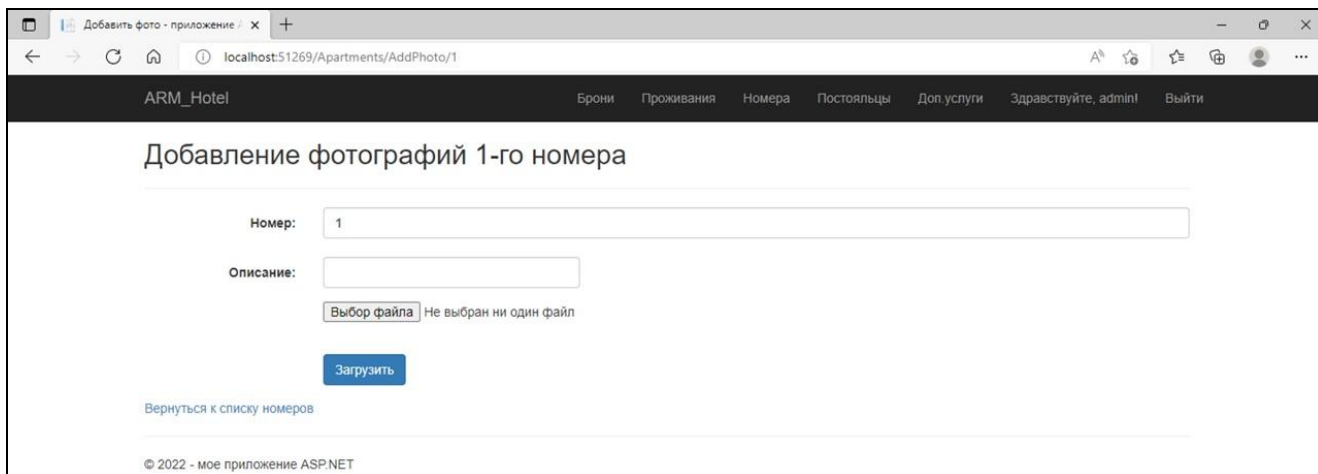


Рисунок 23 – Добавление фото номера

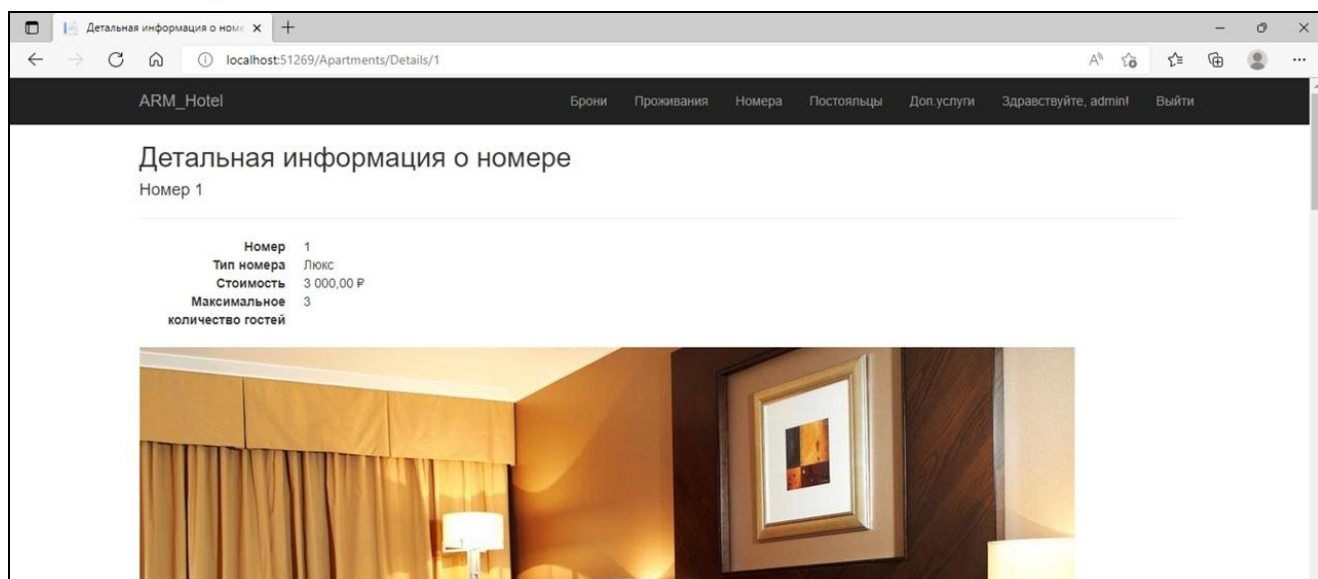


Рисунок 24 – Просмотр информации о конкретном номере

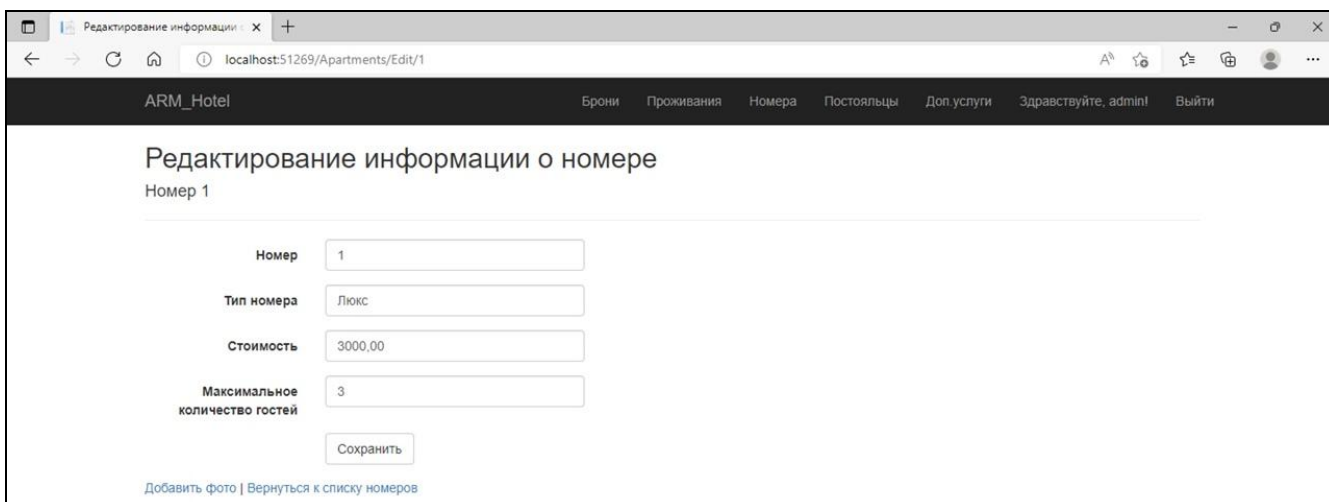


Рисунок 25 – Редактирование информации о номере

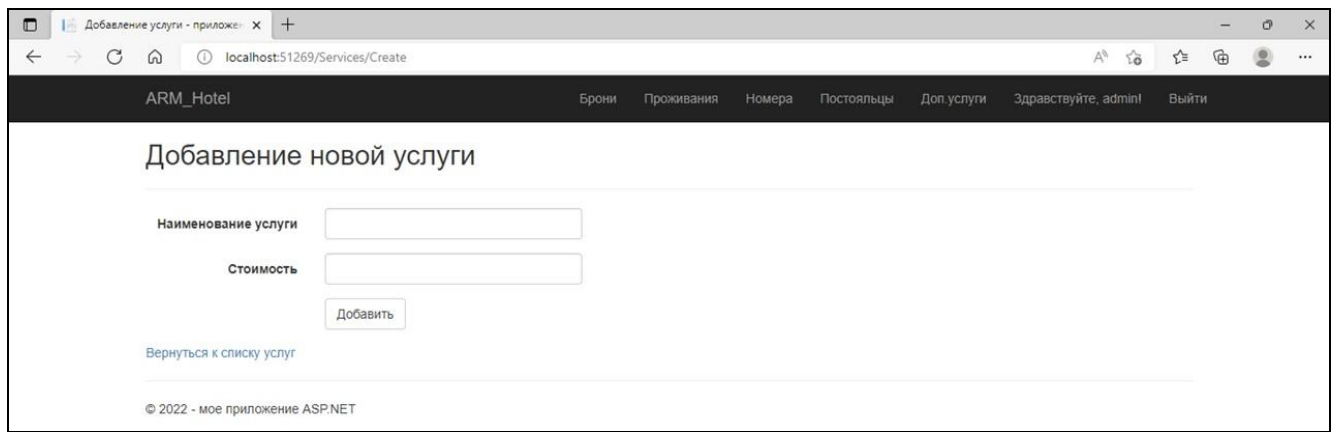


Рисунок 26 – Добавление новой услуги

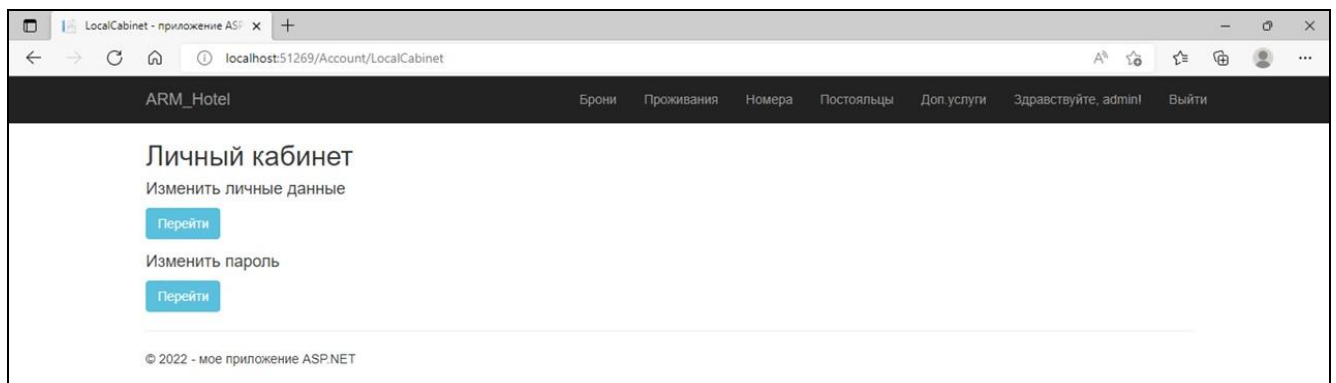


Рисунок 27 – Личный кабинет

Из	Лист	№ докум.	Подп.	Дата

МИВУ 09.03.04-01.000 ПЗ

Приложение 3. Код программы

Код программы можно посмотреть по ссылке: https://github.com/Aleks-de-Bur/ARM_Hotel_RSOD

					МИВУ 09.03.04-01.000 ПЗ	Лист
						39
Из	Лист	№ докум.	Подп.	Дата		