

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных Технологий  
институт  
Информационные Системы  
кафедра

УТВЕ  
Зав.

РЖДАЮ  
кафедрой ИС  
\_\_\_\_\_ С. А. Виденин  
подпись    инициалы, фамилия  
« \_\_\_\_ » \_\_\_\_\_ 2016г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.02 Информационные системы и технологии

Автоматизированное рабочее место администратора отеля

Руководитель	_____	<u>Н. В. Молокова</u>
	подпись, дата	
Выпускник	_____	<u>В. В. Симонов</u>
	подпись, дата	
Нормоконтролер	_____	<u>Ю. В. Шмагрис</u>
	подпись, дата	

Красноярск 2016

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Автоматизированное рабочее место администратора отеля» содержит 48 страниц текстового документа, 24 рисунка, 1 таблицу, 9 использованных источников.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ, РАЗРАБОТКА ПРОГРАММОГО ПРОДУКТА, АРМ, БАЗА ДАННЫХ, СУБД, ASP.NET MVC 5.

Объектом исследования является работа администратора отеля. Предметом исследования является автоматизация работы администратора отеля.

Цель работы – создание автоматизированного рабочего места администратора отеля.

### Основные задачи:

- анализ предметной области;
- выбор средств разработки АРМ;
- разработка пользовательского интерфейса;
- создание АРМ администратора отеля.

Основные результаты: разработка и создание АРМ администратора отеля.

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра «Информационные системы»

УТВЕРЖДАЮ  
кафедрой ИС

Заведующий

\_\_\_\_\_ С.А.Виденин

подпись

«6» мая 2016 г.

**ЗАДАНИЕ**

**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

**в форме бакалаврской работы**

Студенту (ке) Симонову Виктору Вячеславовичу

Группа:КИ12-13БНаправление:09.03.02 «Информационные системы и технологии»

Тема выпускной квалификационной работы: «Автоматизированное рабочее место администратора отеля»

Утверждена приказом по университету №4729/с от 05.04.2016.

Руководитель ВКР: Н. В. Молокова, доцент кафедры«Информационные системы» ИКИТ СФУ.

Исходные данные для ВКР: список требованийк разрабатываемой системе, методические указания научного руководителя.

Перечень разделов ВКР:введение, общие сведения, разработка АРМ администратора отеля, заключение, список использованных источников.

Перечень графического или иллюстрированного материала с указанием основных чертежей, плакатов, слайдов: презентация, выполненная в MicrosoftOfficePowerPoint 2013.

Руководитель ВКР \_\_\_\_\_  
(подпись)

Н.В.Молокова

Задание принял к исполнению \_\_\_\_\_  
(подпись)

В.В.Симонов

« 6 » \_\_мая\_\_ 2016 г.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
Глава 1 Общие сведения.....	5
1.1 Модель данных .....	5
1.2 Базы данных .....	6
1.3 Система управления базами данных .....	9
1.4 Теоретические аспекты автоматизированного рабочего места .....	11
1.5 Информационные системы.....	15
1.6 ASP.NET MVC .....	18
Глава 2 Разработка АРМ администратора отеля .....	22
2.1 Анализ предметной области.....	22
2.2 Программные средства разработки .....	23
2.3 Диаграммы на концептуальном, логическом и физическом уровнях.....	28
2.4 Описание пользовательского интерфейса .....	38
ЗАКЛЮЧЕНИЕ .....	46
СПИСОК СОКРАЩЕНИЙ.....	47
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	48

## ВВЕДЕНИЕ

В настоящее время, с развитием туризма, ростом международных связей, увеличением сотрудничества в областях науки, культуры и техники, всё большее значение придаётся одной из ведущих отраслей сферы услуг – гостиничному хозяйству.

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

Невозможно представить современный мир без информационных технологий. В сравнительно небольшой срок они вошли в жизнь человечества и охватили все сферы деятельности. Информационные технологии – это накопленный опыт и знания, представленные в формализованном виде, пригодном для прикладного использования.

Количество информации, которую необходимо получать, обрабатывать, использовать и хранить, увеличивается с огромной скоростью. Именно поэтому возникает всё большая потребность в автоматизации производства, начиная от подсчёта простых формул школьного курса и заканчивая автоматизированием цехов и заводов. Большинство рутинных процессов стараются автоматизировать. Невозможно представить функционирование предприятий среднего и малого бизнеса на рынке без использования автоматизированных систем управления.

Поэтому тема комплексной автоматизации деятельности гостиницы по размещению гостей, облегчения работы персонала и ускорения процесса обслуживания клиентов очень актуальна в наши дни.

Целью данной выпускной работы является разработка АРМ администратора отеля, которое решит перечисленные выше проблемы.

## Глава 1 Общие сведения

### 1.1 Модель данных

Согласно классической теории баз данных, модель данных - это формальная теория обработки и представления данных в СУБД, которая состоит из трёх аспектов:

- манипуляции: методы манипулирования данными;
- структуры: методы описания логических структур данных и типов в базе данных;
- целостности: методы поддержки целостности и описания базы данных.

Аспект манипуляции определяет способы извлечения данных из базы данных и способы модификации данных, структурный аспект определяет логическое представление базы данных, аспект целостности определяет средства описаний корректных состояний базы данных.

Модели данных можно разделить на три категории:

1) Концептуальные модели данных – описывают данные высокого уровня: ER-модель или модель типа «сущность-связь», функциональная модель, объектно-ориентированная модель;

2) Логические модели данных – база данных состоит из логических записей фиксированного формата: систем, которые основаны на инвертированных списках, сетевая модель данных, иерархическая модель данных и реляционная модель данных;

3) Физические модели данных – описание того, как данные хранятся в компьютере: обобщающая модель, модель памяти кадров.

На рисунке 1 представлена классификация моделей данных.

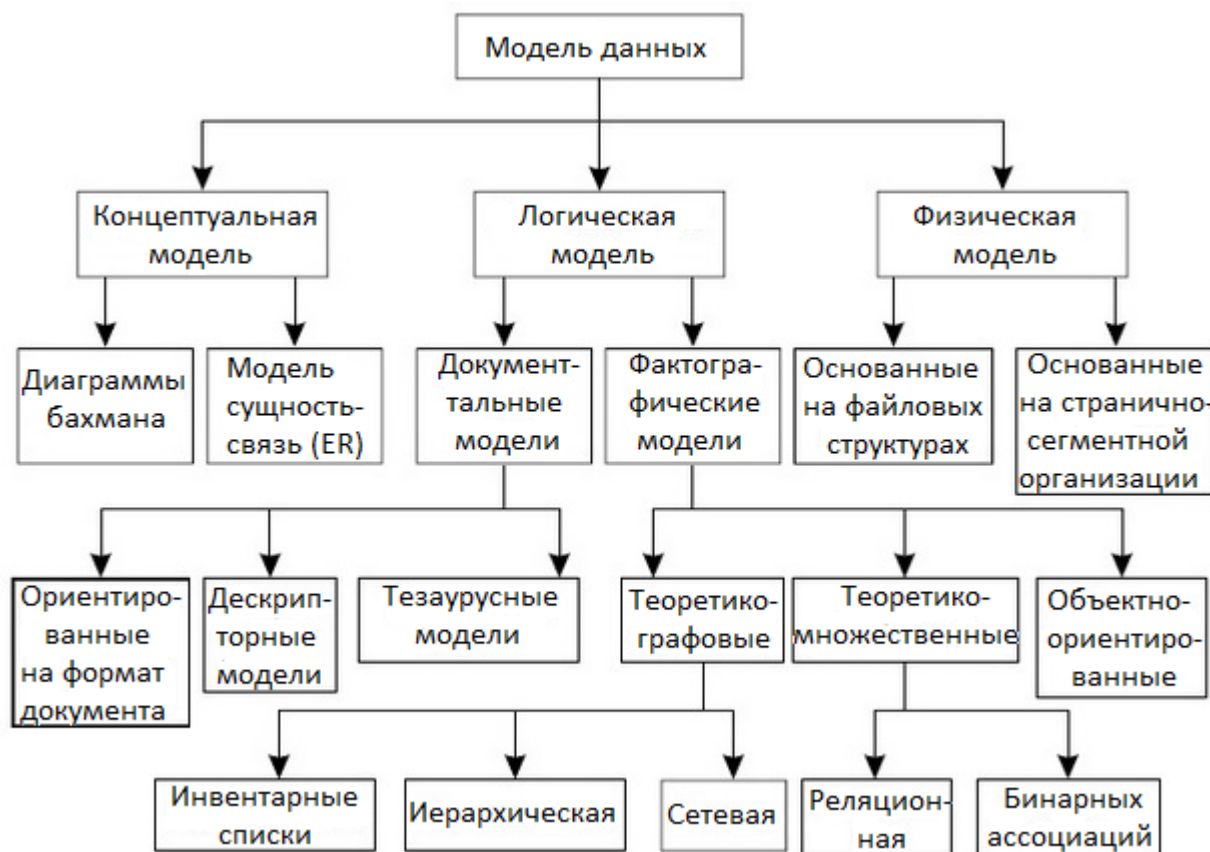


Рисунок 1 – Классификация моделей данных

Модель данных — логическое, самодостаточное, абстрактное определение операторов, объектов и прочих элементов, которые составляют абстрактную машину доступа к данным, взаимодействующую с пользователем. Данные объекты дают возможность моделировать структуру данных, а операторы позволяют моделировать поведение данных.

Любая СУБД и БД строится на основе какой-либо неявной или явной модели данных. Каждую СУБД, построенную на одинаковой модели данных, относят к одному типу.

## 1.2 Базы данных

Базой данных называется совокупность самостоятельных материалов (расчетов, статей, нормативных актов и т.д.), представленных в объективной форме исистематизированных так, чтобы электронная вычислительная



машина могла найти и обработать данные материалы. Существуют и другие определения баз данных, сводящиеся к понятию «совокупность хранимых данных». Но большая часть этих определений не дает отличить базу данных от объектов, которые не являются базой данных, таких как, архивы, картотеки, документы, библиотеки и т.д. Из чего следует, что база данных - это не просто множество хранимых данных, но такое множество, которое обладает тремя важными признаками (свойствами): База данных обрабатывается и хранится в вычислительной системе. Хранилище информации, которое находится вне компьютера не является базой данных.

Данные в базе данных систематизированы и структурированы. В данном случае под этим понимается явное выделение элементов и составных частей, а также связей между ними, типизация связей и элементов, при которой с каждым типом связи или элемента соотносятся допустимые операции и определенная семантика. Структура баз данных обеспечивает обработку данных и эффективный поиск. В данном случае эффективность определяется соотношением гибкости и мощности возможностей (поиска и обработки) с затратами ресурсов и усилий. Из этих трёх признаков только первый является строгим, а остальные допускают различные степени оценки и трактовки. Нет возможности формально определить, является ли совокупность данных на компьютере базой данных или нет. Возможно только установить, насколько она соответствует требованиям к базе данных. В таких ситуациях большую роль играет общепринятая практика. В соответствии с которой не являются базами данных электронные таблицы или файловые архивы, хотя они обладают признаками баз данных.

#### Классификация БД

Существует большое количество разных видов баз данных, которые отличаются по различным критериям.

По технологии хранения:

- в оперативной памяти (in-memorydatabases);

- во вторичной памяти (традиционные);
- в третичной памяти (tertiarydatabases).

По модели данных:

- Многомерные;
- Иерархические;
- Реляционные;
- Объектные;
- Объектно-реляционные;
- Объектно-ориентированные;
- Сетевые.

По степени распределённости:

- Распределённые;
- Сосредоточенные (централизованные);
- Пространственные;
- Временные;
- Пространственно-временные.

По содержанию:

- Исторические;
- Географические;
- Мультимедийные
- Научные и т. д.

На рисунке 2 представлена классификация БД.

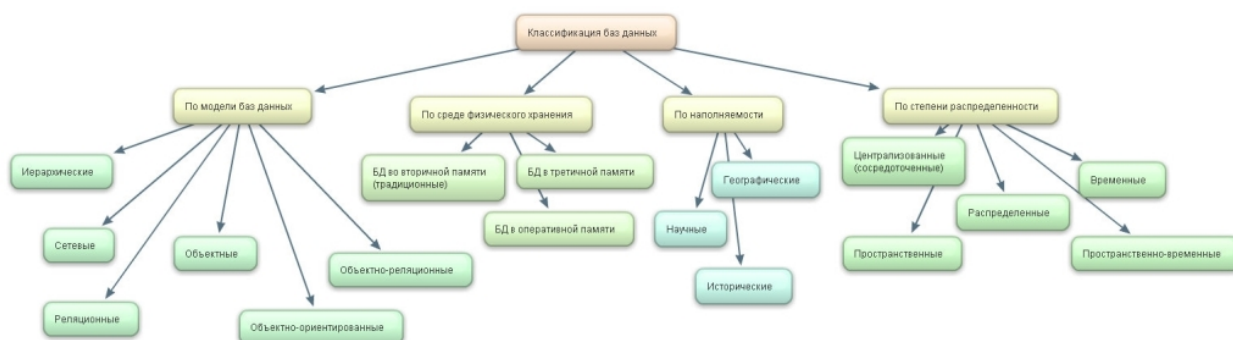


Рисунок 2 – Классификация баз данных

### 1.3 Система управления базами данных

Система управления базами данных (СУБД) – совокупность языковых и программных средств, которые обеспечивают создание, использование, управление и ведение баз данных.

Система управления базами данных состоит из следующих компонентов:

- Ядро СУБД, отвечающее за управление данными во внешней оперативной памяти и журнализацию;
- Процессор языка запросов, с помощью которого обеспечивается оптимизация запросов на извлечение и изменение данных, а также создание исполняемого машинно-независимого внутреннего кода;
- Подсистема поддержки времени исполнения, интерпретирующая программы манипуляции данными, которые создают пользовательский интерфейс с СУБД;
- Сервисные программы, которые обеспечивают дополнительные возможности по обслуживанию информационной системы.

На рисунке 3 представлены компоненты системы управления базами данных.

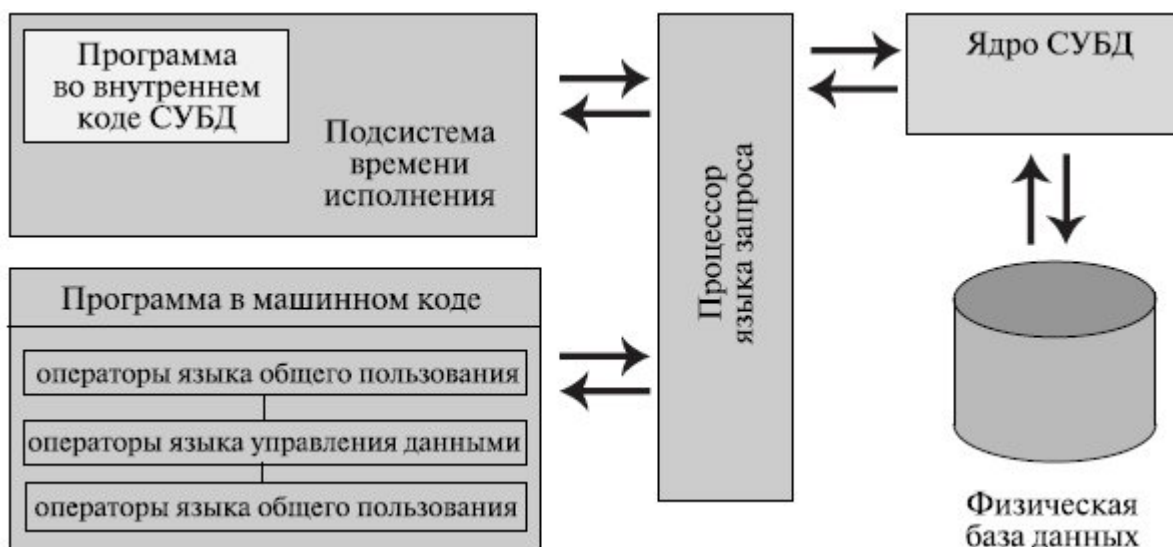


Рисунок 3 – Компоненты СУБД

### Основные функции СУБД:

- управление данными в оперативной памяти;
- управление данными во внешней памяти;
- поддержка языков БД;
- восстановление базы данных, журнализация изменений и резервное копирование.

### Преимущества использования СУБД:

- быстрый поиск информации благодаря её структурированности;
- освобождение человеческих ресурсов;
- уменьшение затрат на покупку, производство и т.д. услуг и товаров;
- быстрый доступ к информации.

Существует множество классификаций СУБД. Ниже представлены некоторые из них:

#### По модели данных различают:

- сетевые;
- реляционные;
- объектно-реляционные;
- объектно-ориентированные;
- иерархические;

#### По степени расположения:

- распределённые;
- локальные;

#### По способу доступа к базе данных:

- встраиваемые;
- клиент-серверные;
- файл-серверные.

## 1.4 Теоретические аспекты автоматизированного рабочего места

Автоматизированное рабочее место (АРМ) – совокупность информационных программных и технических ресурсов, которые обеспечивают пользователю обработку данных и автоматизируют управленческие функции в определенной предметной области. При этом подразумевается, что операции по хранению, переработке и накоплению информации возлагают на вычислительную технику, а пользователь осуществляет часть операций, которые требуют творческого подхода, а также часть ручных операций.

Одной из самых простых функций АРМ является справочно-информационное обслуживание, которое присуще любому АРМ. АРМ могут включать в себя расчетные алгоритмы, которые обеспечивают обработку информации и отображение результатов, экранные формы документов и текстовые системы в зависимости от их назначения.

АРМ, которые созданы на базе ПК – простой и наиболее распространенный вариант, который обеспечивает: удобство и простоту пользования; простоту адаптации к определенным функциям пользователя; простую организацию технического обслуживания; компактность размещения и невысокие требования к условиям эксплуатации.

Полезным режимом работы АРМ является функционирование в пределах локальной вычислительной сети, когда следует распределить информационные и вычислительные ресурсы между пользователями.

Более сложные системы АРМ предусматривают подключение нескольких ПК по каналам связи к главной ЭВМ или через специальное оборудование – к системам общего назначения и информационным службам

Взаимодействие пользователя и программного обеспечения АРМ чаще всего реализуется через меню.

Свойства автоматизированного рабочего места.

В основе конструирования АРМ лежат следующие основные принципы:

- Максимальная ориентация на конечного пользователя, которая достигается путем создания инструментальных средств адаптации автоматизированного рабочего места к уровню подготовки пользователя, к возможностям его обучения и самообучения.

- Формализация профессиональных знаний, возможность предоставления самостоятельно автоматизировать новые функции, а также решать новые задачи помощью АРМ в процессе работы с системой.

- Проблемная ориентация автоматизированного рабочего места на решение конкретного класса задач, которые объединены одной технологией обработки информации, совокупностью режимов эксплуатации и работы, что свойственно специалистам экономических служб.

- Модульность построения, которая обеспечивает сопряжение автоматизированного рабочего места с остальными компонентами системы обработки информации, наращивание и модификацию возможностей АРМ без приостановки его функционирования.

- Создание для пользователя комфортабельных условий труда и благоприятного интерфейса общения с системой.

Существуют общие принципы создания любых АРМ: гибкость, системность, эффективность, устойчивость.

- Системность. АРМ необходимо рассматривать в роли систем, у которых структура определяется функциональным назначением.

- Гибкость. Система должна быть приспособлена к возможным перестройкам, благодаря модульности построения всех подсистем и стандартизации элементов этих подсистем.

- Устойчивость. Система АРМ обязана выполнять основные функции в независимости от воздействия на нее внешних и внутренних возмущающих факторов. То есть неполадки в её отдельных элементах

должны быть легко устранимы, а также система должна быстро восстанавливать свою работоспособность.

- Эффективность автоматизированного рабочего места надлежит рассматривать как интегральный показатель уровня реализации перечисленных выше принципов, отнесенный к затратам на эксплуатацию и создание системы.

Функционирование АРМ может дать желанный эффект при условии того, что будут правильно распределены функции и нагрузка между пользователем и машинными средствами обработки информации.

Автоматизированное рабочее место должно соответствовать следующим требованиям:

- своевременное удовлетворение информационных и вычислительных потребностей пользователя;
- минимальное время ответа на запросы специалиста;
- адаптация к степени подготовки специалиста и его профессиональным запросам;
- простота в освоении приемов работы на АРМ, надежность, легкость общения и простота обслуживания;
- терпимость по отношению к специалисту;
- допустимость быстрого обучения пользователя;
- возможность работы в составе вычислительной сети.

#### Классификация АРМ

Существует множество классификаций АРМ. Ниже представлены некоторые из них:

По степени автоматизации:

- ручные рабочие места — в распоряжении работника имеется специальная мебель (стул, стол, шкафы, линейки, телефон, таблицы и т.д.);
- механизированные рабочие места, помимо того, имеют простейшие или программируемые калькуляторы;

- автоматизированные рабочие места непременно используют ПК с соответствующим программным обеспечением.

По количеству сотрудников, которые используют АРМ, и функциям, выполняемым ими:

- индивидуальные автоматизированные рабочие места, характерные для руководителей всевозможных рангов;
- групповые АРМ, которые используются лицами, готовящими информацию для ее будущего использования и принятия организационных решений руководителями.

По типизации решаемых функциональных задач:

- уникальные автоматизированные рабочие места, которые имеют узкую специализацию для решения большого количества нестандартных задач и проблем;
- массовые АРМ, которые созданы для решения типичных задач во всевозможных отраслях.

По специализации:

- АРМ руководителя - обладает функциональной замкнутостью, которая полностью обеспечивает автономную работу руководителя.
- АРМ специалиста - предоставляет ему возможность решать различные функциональные задачи, стоящие перед ним, максимально эффективно используя нужную ему информацию.
- АРМ технического работника - избавляет его от ежедневно выполняемой шаблонной работы, которая требует некоторых профессиональных умений.

По технической базе создания АРМ:

- Автоматизированное рабочее место на базе универсальных ЭВМ, которые обеспечивают специалистам возможность работать с крупными массивами данных при программной и технической поддержке, которая осуществляется силами работников информационного центра.



Автоматизированное рабочее место на базе ПК являются простейшим и наиболее распространенным вариантом для создания АРМ, в связи с тем, что избавляют от недостатков автоматизированного рабочего места на базе больших ЭВМ.

## 1.5 Информационные системы

Информационные системы – совокупность организационного, технического, программного обеспечения и персонала, которая используется для обработки, выдачи и хранения информации для достижения цели управления.

Структура информационных систем.

Структуру информационной системы составляет совокупность отдельных частей информационной системы, которые называются подсистемами.

Подсистема – часть системы, выделенная по какому-либо признаку.

Среди обеспечивающих подсистем чаще всего выделяют математическое, информационное, правовое, техническое, организационное и программное обеспечение.

- Информационное обеспечение – методы и средства построения информационной базы системы, включающие системы классификации и кодирования информации, унифицированные системы документов, схемы информационных потоков, принципы и методы создания баз данных.

- Техническое обеспечение – это комплекс технических средств, которые задействованы в технологическом процессе преобразования информации в системе. К техническому обеспечению относятся вычислительные машины, каналы передачи данных, аппаратура и периферийное оборудование.

- Программное обеспечение состоит из совокупности программ регулярного применения, которые необходимы для решения функциональных задач, а также программ, которые позволяют с наибольшей эффективностью использовать вычислительную технику и обеспечивать пользователям удобства в работе.

- Математическое обеспечение – это совокупность моделей, математических методов, и алгоритмов обработки информации, которые используются в системе.

- Правовое обеспечение – это комплекс решений, которые регламентируют создание и функционирование информационной системы, порядок получения, преобразования и использования информации.

- Организационное обеспечение – это комплекс решений, которые регламентируют процессы создания и функционирования системы в целом, а также персонала данной системы.

Классификация информационных систем.

В зависимости от степени автоматизации информационных процессов в системе информационные системы делятся на автоматические, ручные и автоматизированные.

- Ручным информационным системам характерно отсутствие современных технических средств переработки информации и выполнение всех операций человеком.

- Автоматическим информационным системам характерно выполнение всех операций по переработке информации без участия человека.

- Автоматизированным информационным системам характерно участие в процессе обработки информации как человека, так и технических средств, при этом главная роль отводится компьютеру.

По характеру использования информации информационные системы подразделяются на:

- Информационно-поисковые системы – информационные системы, которые производят ввод, хранение, выдачу и систематизацию информации по запросу пользователя без сложных преобразований данных.

- Информационно-решающие системы – ИС, которые осуществляют все операции переработки информации, следуя определенному алгоритму. В свою очередь информационно-решающие системы по степени воздействия выработанной результатной информации на процесс принятия решений подразделяются на два класса:

- Управляющие информационные системы – ИС, которые вырабатывают информацию, по которой человек принимает то или иное решение.

- Советующие информационные системы – ИС, которые вырабатывают информацию, принимаемую человеком к сведению и не приводящую к немедленным действиям.

По сфере применения информационные системы делятся на:

- Информационные системы организационного управления – информационные системы, предназначенные для автоматизации функций управленческого персонала. Основными функциями данных систем являются: оперативный учет и анализ, оперативный контроль и регулирование, бухгалтерский учет, перспективное и оперативное планирование, управление сбытом и снабжением, другие организационные и экономические задачи.

- Информационные системы управления технологическими процессами – ИС, которые предназначены для автоматизации всех функций производственного персонала. Широко используются для организации и поддержки технологического процесса в металлургической и машиностроительной промышленности.

- Информационные системы автоматизированного проектирования – ИС, предназначенные для автоматизации функций дизайнеров, инженеров-проектировщиков, архитекторов, конструкторов при создании новой технологии или техники. Основными функциями данных систем являются: инженерные расчеты, создание проектной документации, моделирование проектируемых объектов и создание графической документации.
- Интегрированные (корпоративные) информационные системы – ИС, которые используются для автоматизации всех функций фирмы и охватывающие весь цикл работ, начиная с проектирования и заканчивая сбытом продукции.

## 1.6 ASP.NET MVC

Платформа ASP.NET MVC представляет собой фреймворк для создания сайтов и веб-приложений с помощью реализации паттерна MVC.

Концепция паттерна (шаблона) MVC (model - view - controller) предполагает разделение приложения на три компонента:

Контроллер (controller) представляет класс, обеспечивающий связь между пользователем и системой, представлением и хранилищем данных. Он получает вводимые пользователем данные и обрабатывает их. И в зависимости от результатов обработки отправляет пользователю определенный вывод, например, в виде представления.

Представление (view) - это собственно визуальная часть или пользовательский интерфейс приложения. Как правило, html-страница, которую пользователь видит, зайдя на сайт.

Модель (model) представляет класс, описывающий логику используемых данных.

Общая схема взаимодействия этих компонентов представлена на рисунке 4:



Рисунок 4 – Общая схема взаимодействия MVC

В этой схеме модель является независимым компонентом - любые изменения контроллера или представления не затрагивают модель. Контроллер и представление являются относительно независимыми компонентами, и нередко их можно изменять независимо друг от друга.

Благодаря этому реализуется концепция разделение ответственности, в связи с чем легче построить работу над отдельными компонентами. Кроме того, вследствие этого приложение обладает лучшей тестируемостью. И если нам, допустим, важная визуальная часть или фронтэнд, то мы можем тестировать представление независимо от контроллера. Либо мы можем сосредоточиться на бэкэнде и тестировать контроллер.

Конкретные реализации и определения данного паттерна могут отличаться, но в силу своей гибкости и простоты он стал очень популярным в последнее время, особенно в сфере веб-разработки.

Свою реализацию паттерна представляет платформа ASP.NET MVC. 2013 год ознаменовался выходом новой версии ASP.NET MVC - MVC 5, а также релизом VisualStudio 2013, которая предоставляет инструментарий для работы с MVC5.

Хотя во многих аспектах MVC 5 не слишком сильно будет отличаться от MVC 4, многое из одной версии вполне применимо к другой, но в то же время есть и существенные отличия:

- В MVC 5 изменилась концепция аутентификации и авторизации. Вместо SimpleMembershipProvider была внедрена система ASP.NET Identity, которая использует компоненты OWIN и Katana.

- Для создания адаптивного и расширяемого интерфейса в MVC 5 используется css-фреймворкBootstrap

- Добавлены фильтры аутентификации, а также появилась функциональность переопределения фильтров

- В MVC 5 также добавлены атрибуты маршрутизации

Это наиболее важные нововведения в MVC 5. Кроме того, есть еще ряд менее значимых, например, использование по умолчанию EntityFramework 6, некоторые изменения при создании проекта (концепция One ASP.NET), дополнительные компоненты и т.д.

В любом случае все полученные при работе с MVC 4 навыки можно успешно применять при использовании MVC 5, учитывая, конечно, нововведения.

#### Движок представлений

Чтобы управлять вставками кода и разметкой в представлении используется движок представлений. До версии MVC 5 использовалось два движка:

Razor и WebForms. Начиная с MVC 5 Razor- единственный движок, который встроен по умолчанию. Движком WebForms используются файлы .aspx, а движком Razor-файлы.vbhtml и .cshtml для хранения кода представлений. Основой синтаксиса движка Razor является знак @, после которого осуществляется переход к коду на языках C# и VB.NET. Возможно использование сторонних движков. Файлы представлений не являются статическими стандартными страницами с кодом html, в процессе генерации контроллером ответа с использованием представлений компилируются в классы, из которых в последствии генерируется страница html.

#### Маршрутизация

При обработке запросов фреймворк ASP.NET MVC опирается на

систему маршрутизации, сопоставляющую входящие запросы с определенными маршрутами в системе, указывающими какой метод и контроллер должен обработать данный запрос. Встроенный маршрут предусматривает трехзвенную структуру: контроллер /параметр/действие.

ASP.NET MVC и ASP.NET WebForms

ASP.NET MVC и ASP.NET WebForms – это родственные технологии и в основании которых лежит общая платформа ASP.NET. Но несмотря на это MVC имеет преимущества перед WebForms, такие как:

- разделение ответственности (отдельная разработка различных компонентов - моделей, представлений, контроллера)
- усовершенствованная тестируемость
- соответствие такому протоколу, как HTTP
- повышенная настраиваемость под собственные нужды и гибкость

## **Глава 2    Разработка АРМ администратора отеля**

### **2.1    Анализ предметной области**

Деятельность администратора отеля связана с переработкой и хранением большого количества информации. Огромный документооборот затрудняет работу, снижает эффективность доступа к нужной информации.

Из-за этого и возникает необходимость автоматизации процесса регистрации и работы с клиентами. Автоматизация позволит сократить время необходимое для регистрации клиентов и поиска нужной информации.

Одной из основных задач администратора отеля является бронирование номеров, а также прием, регистрация, размещение и выписка посетителей. В процессе оформления специалистом заносится основная информация о клиенте, предоставленном номере, сроках и конечной цене проживания в номере.

Целью работы является разработка и создание автоматизированного рабочего места администратора отеля для облегчения внесения и учета данных, упрощения ведения документооборота и сокращения затрат.

Выполнение данных требований позволит снизить затраты на создание программного продукта и создать качественную систему, позволяющую повысить экономическую эффективность.

Программа АРМ администратора отеля – автоматизирует работу администратора. Программа служит для эффективной и быстрой работы с клиентами. Данная программа должна предоставить администратору, следующие возможности:

- Добавление клиента с указанием его данных, а также времени остановки в отеле, предоставленного ему номера и цене за проживание;
- Добавление и редактирование информации о категории номера и цене проживания в нем;
- Бронирование номеров администратором;



- Печать базы данных клиентов;
- Осуществление поиска нужного клиента по базе данных;
- Бронирование номеров клиентами через интернет;
- Просмотр забронированных номеров.

Для достижения поставленных целей были сформулированы следующие задачи:

- Анализ предметной области
- Выбор средств разработки АРМ
- Разработка пользовательского интерфейса
- Создание АРМ администратора отеля

## **2.2 Программные средства разработки**

Microsoft Visual Studio — линейка продуктов от Microsoft, которые включают в себя интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Эти продукты предоставляют возможность разрабатывать консольные приложения и приложения с графическим интерфейсом, а также веб-приложения, веб-службы и веб-сайты, и в родном, и в управляемом кодах для всех платформ, которые поддерживают Windows, Windows CE, Windows Mobile, Xbox, .NET Framework, Windows Phone, Silverlight и .NET Compact Framework.

Visual Studio состоит из редактора исходного кода, поддерживающего технологию IntelliSense и имеющего возможность простейшего рефакторинга кода. Встроенный отладчик может работать в роли отладчика уровня исходного кода, а может в роли отладчика машинного уровня. Другие встраиваемые инструменты включают в себя редактор форм, чтобы упростить создание графического интерфейса приложения, дизайнер классов, дизайнер схемы баз данных и веб-редактор. VS даёт возможность подключать и создавать сторонние дополнения, чтобы расширить функциональность на каждом уровне, в том числе добавление поддержки систем контроля версий

исходного кода (VisualSourceSafe и Subversion), а также добавление новых наборов инструментов или инструментов, необходимых для остальных аспектов процесса разработки ПО.

VS 2015 предоставляется в трёх редакциях: бесплатной CommunityEdition, которая объединяет все Express-версии, а также платных ProfessionalEdition для малых проектов и EnterpriseEdition для крупных.

Плюсы VS 2015 Community Edition:

- Процесс установки VisualStudio 2015 поделен на части, поэтому можно установить только необходимые компоненты. Это позволяет ускорить установку для многих типичных сценариев, включая разработку приложений платформы .NET или веб-приложений.

- VisualStudio 2015 обеспечивает крайне простой способ подключения вашего приложения к службам. Новый мастер "Добавление подключенной службы" настраивает ваш проект, добавляет необходимую поддержку проверки подлинности и загружает необходимые пакеты NuGet, чтобы вы могли начать процесс разработки для своей службы быстро и безболезненно.

- Новый компилятор Roslyn для C# не только быстрее компилирует, но также позволяет использовать полностью новые сценарии, такие как динамический анализ кода, который предоставляет подробный и настраиваемый отзыв и предложения непосредственно в редакторе кода по мере ввода данных. В VisualStudio 2015 лампочки отображаются в левой части или во всплывающей подсказке. Лампочка сообщает в режиме реального времени, что компилятор обнаружил проблему в коде и предлагает вариант ее решения.

- Установленные расширения VisualStudio обновляются автоматически при появлении новой версии в коллекции VisualStudio.

## Структура ASP.NET

Существует множество различных видов и типов архитектур, которые успешно применяются. Одной из наиболее используемых является классическая трехуровневая система, которая подразумевает разделение приложения на три уровня.

Тут сразу надо сказать, что многоуровневой архитектурой часто обозначают два не совсем связанных понятия: n-layer и n-tier. И layer, и tier, как правило, обозначаются словом "уровень", иногда по отношению к "layer" еще употребляется слово "слой". Однако в обоих случаях уровни будут разного порядка.

Tier представляет физический уровень. То есть если мы говорим о трехуровневой архитектуре, то n-tier приложение могло быть разделено на такие уровни: сервер базы данных, веб-приложение на веб-сервере и браузер пользователя. То есть каждый уровень представлял бы особый отдельный физический процесс, даже если бы и сервер баз данных, и веб-сервер, и браузер пользователя находились бы на одном компьютере. Если бы в качестве клиента альтернативно использовалось мобильное приложение, то это был бы еще один физический уровень.

Layer представляет логический уровень. То есть у нас может быть уровень доступа к данным, уровень бизнес-логики, уровень представления, уровень сервисов и так далее. При этом логические уровни не совпадают с физическими. Так, обычно уровень предоставления в приложении ASP.NET содержит и контроллеры, которые обрабатывают ввод, и представления, которые отображаются в веб-браузере, то есть разделяется на два физических уровня.

В данном случае мы будем говорить именно о логических уровнях, то есть о n-layer архитектуре.

Классическая трехуровневая система состоит из следующих уровней (схема представлена на рисунке 5):

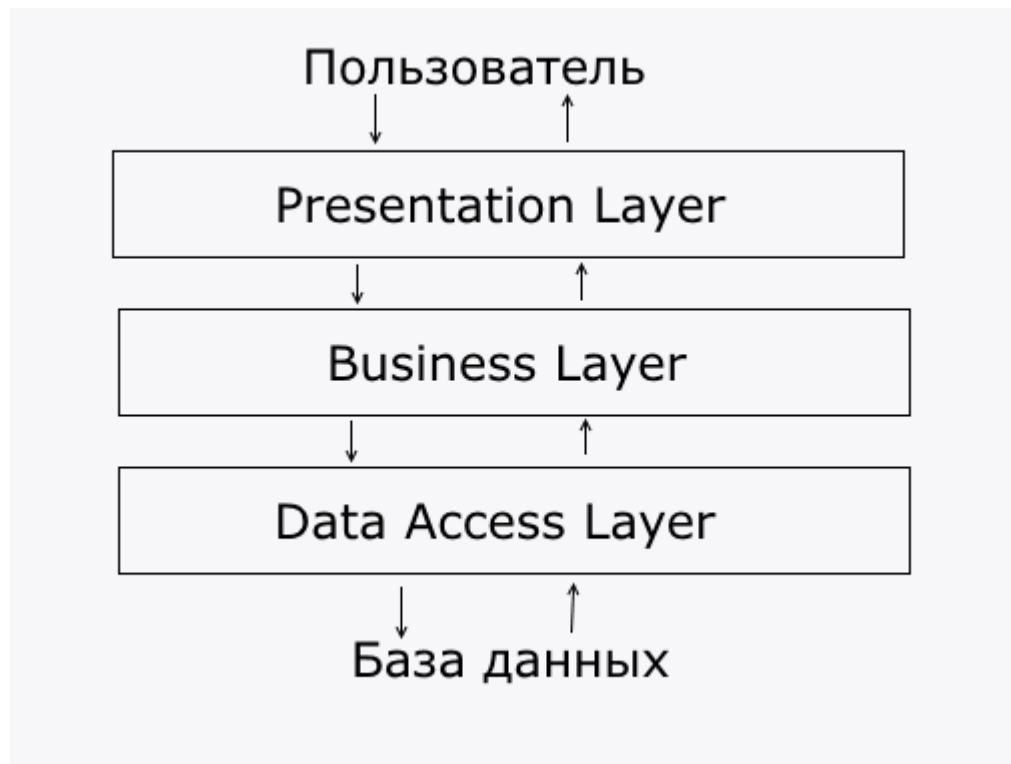


Рисунок 5 – Схема трехуровневой системы

- **Presentationlayer** (уровень представления): это тот уровень, с которым непосредственно взаимодействует пользователь. Этот уровень включает компоненты пользовательского интерфейса, механизм получения ввода от пользователя. Применительно к `asp.net mvc` на данном уровне расположены представления и все те компоненты, который составляют пользовательский интерфейс (стили, статичные страницы `html`, `javascript`), а также модели представлений, контроллеры, объекты контекста запроса.
- **Businesslayer** (уровень бизнес-логики): содержит набор компонентов, которые отвечают за обработку полученных от уровня представлений данных, реализует всю необходимую логику приложения, все вычисления, взаимодействует с базой данных и передает уровню представления результат обработки.
- **DataAccesslayer** (уровень доступа к данным): хранит модели, описывающие используемые сущности, также здесь размещаются специфичные классы для работы с разными технологиями доступа к данным, например, класс контекста данных `EntityFramework`. Здесь также хранятся

репозитории, через которые уровень бизнес-логики взаимодействует с базой данных.

При этом надо отметить, что крайние уровни не могут взаимодействовать между собой, то есть уровень представления (применительно к ASP.NET MVC, контроллеры) не могут напрямую обращаться к базе данных и даже к уровню доступа к данным, а только через уровень бизнес-логики.

Уровень доступа к данным не зависит от других уровней, уровень бизнес-логики зависит от уровня доступа к данным, а уровень представления - от уровня бизнес-логики.

Компоненты, как правило, должны быть слабосвязанными (loosecoupling), поэтому неотъемлемым звеном многоуровневых приложений является внедрение зависимостей.

При чем об ASP.NET MVC мы говорим прежде всего применительно к уровню представления, остальные же уровни могут быть реализованы независимо и могут использоваться в приложениях на других технологиях, как WindowsForms, WPF и т.д. И, как правило, все приложение в целом будет представлять решение (solution) в VisualStudio, а отдельные уровни - проекты. В то же время неверно полагать, что если уровень обязательно должен соответствовать отдельному проекту. При необходимости мы можем раздробить один уровень на несколько проектов, главное, чтобы его функционал представлял единое логическое звено.

## 2.3 Диаграммы на концептуальном, логическом и физическом уровнях

Перед началом проектирования был создан набор диаграмм на основе анализа предметной области и поставленных требований к системе с помощью языка графического описания Unified Modeling Language, а также контекстные диаграммы. UML (унифицированный язык моделирования) — это язык графического описания для моделирования бизнес-процессов, объектного моделирования в области разработки программного обеспечения, отображения организационных структур и системного проектирования.

UML – это язык широкого профиля, открытый стандарт, использующий графические обозначения для создания абстрактной модели системы. UML создан для определения, проектирования, визуализации, документирования программных систем. На основе UML-диаграмм можно сгенерировать код, но сам UML языком программирования не является. В результате моделирования был разработан следующий набор диаграмм:

### Диаграмма деятельности

Диаграмма деятельности - UML-диаграмма, показывающая разложение некоторой деятельности на составные части. Под деятельностью понимается спецификация исполняемого поведения в форме координированного параллельного и последовательного выполнения подчинённых элементов - вложенных видов деятельности и отдельных действий, которые соединены между собой потоками, идущими от выходов одного узла к входам другого.

Диаграмма деятельности используются при моделировании бизнес-процессов, последовательных и параллельных вычислений, а также технологических процессов.

Диаграмма деятельности состоит из ограниченного количества фигур, которые соединены стрелками.

Сценарий последовательности действий варианта «Ведения базы клиентов».

Поток событий - прецедент начинается при выборе администратором окна «Добавить» в списке клиентов.

Базовый поток – Добавления нового клиента:

- 1) Администратор переходит в окно «Добавить».
- 2) Система отображает поля для заполнения.
- 3) Администратор заполняет поле «Фамилия».
- 4) Администратор заполняет поле «Имя».
- 5) Администратор заполняет поле «Отчество».
- 6) Администратор заполняет поле «Номер паспорта».
- 7) Администратор заполняет поле «Место жительства».
- 8) Администратор указывает дату в поле «Дата заселения».
- 9) Администратор указывает дату в поле «Дата выселения».
- 10) Администратор заполняет поле «Номер комнаты».
- 11) Администратор заполняет поле «Цена».
- 12) Администратор нажимает кнопку «Создать».
- 13) Система сохраняет введенные данные в базу данных.

Постусловие: При успешном окончании прецедента администратор, открыв страницу «Клиенты», увидит созданную им запись. При неуспешном – созданная запись отсутствует.

Диаграмма деятельности представлена на рисунке 6.

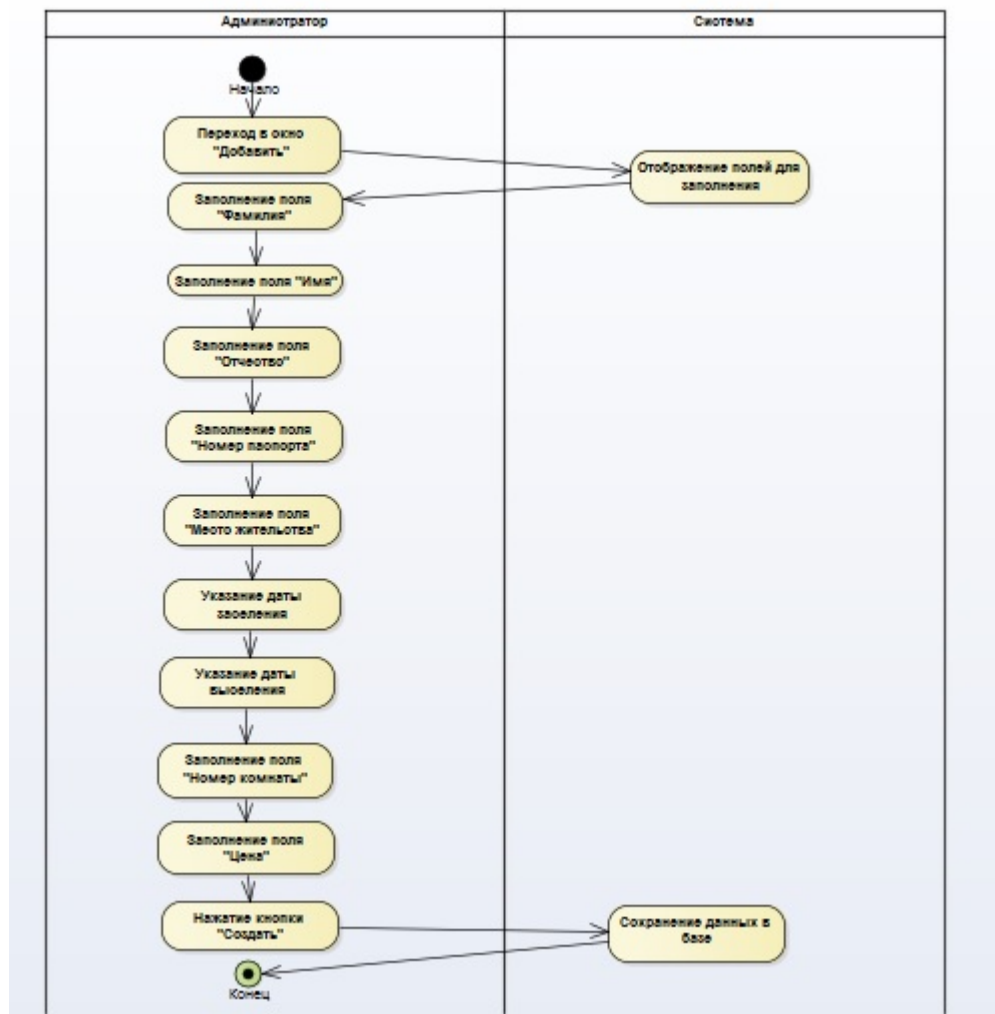


Рисунок 6 – Диаграмма деятельности

### Контекстная диаграмма

Контекстная диаграмма - это модель, представляющая систему как набор иерархических действий, в которой каждое действие преобразует некоторый объект или набор объектов. Высшее действие иерархии называется действием контекста - это самый высокий уровень, который непосредственно описывает систему. Уровни ниже называются порожденными декомпозициями и представляют подпроцессы родительского действия. Контекстная диаграмма определяет внешние для системы объекты, которые взаимодействуют с ней, но ничего не отображает во внутренней структуре или в поведении системы.



При ее разработке сначала строится модель существующей организации работы AS-IS (как есть) на основе должностных инструкций, приказов, отчетов, нормативной документации и т. д. Она позволяет выяснить, «что мы делаем сегодня» перед тем, как «перепрыгнуть» на то, «что мы будем делать завтра». Анализ модели позволяет понять, где находятся слабые места, в чем будут состоять преимущества новых процессов и насколько глубоким изменениям подвергнется существующая организация деятельности. Признаками неэффективной организации деятельности могут быть:

- бесполезные, неуправляемые и дублирующие работы;
- работы без результата;
- неэффективный документооборот (нужный документ не оказывается в нужное время в нужном месте) и т. д.

На рисунках 7 представлена модель AS-IS процесса добавления нового клиента в базу и работы с ним.

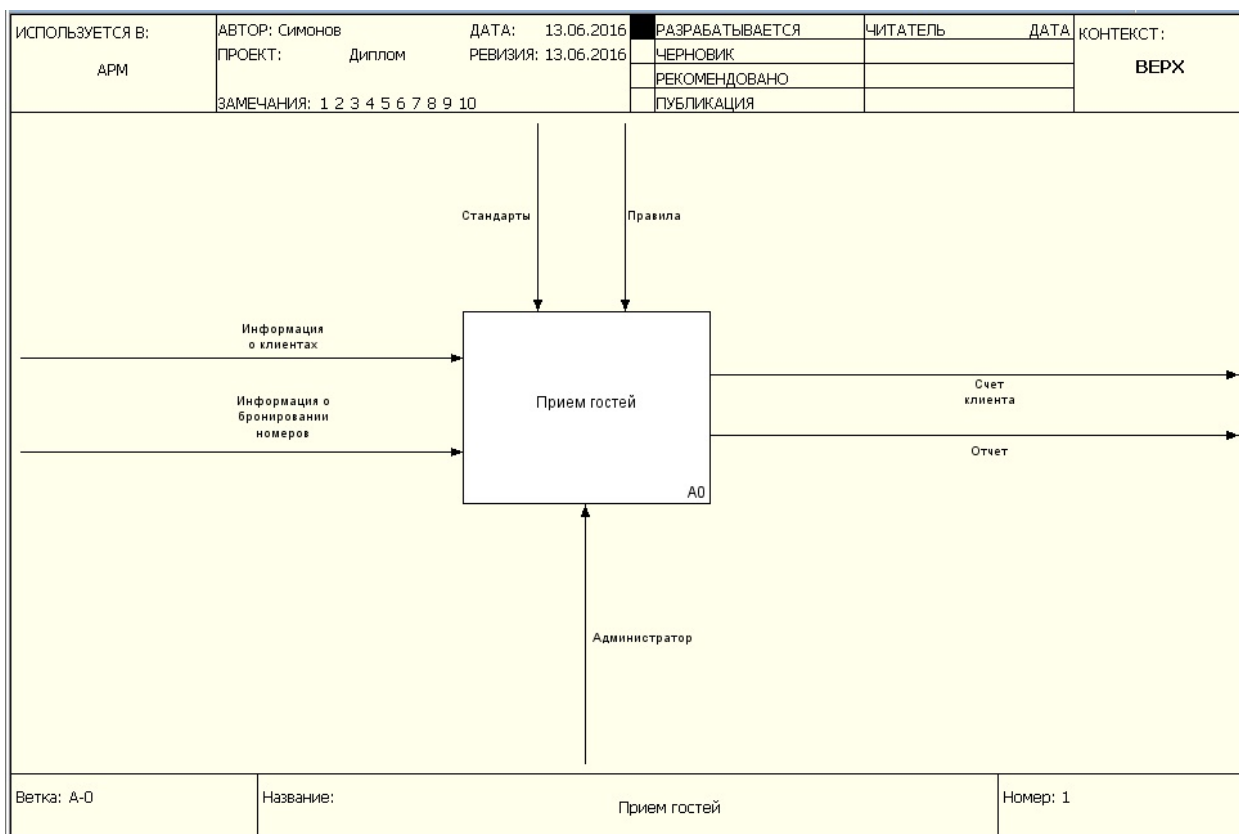


Рисунок 7 – Контекстная диаграмма (AS-IS)

Для большей детализации проходящих процессов проводится декомпозиция контекстной диаграммы, что необходимо для большей наглядности функций системы. Декомпозиция представлена на рисунке 8.

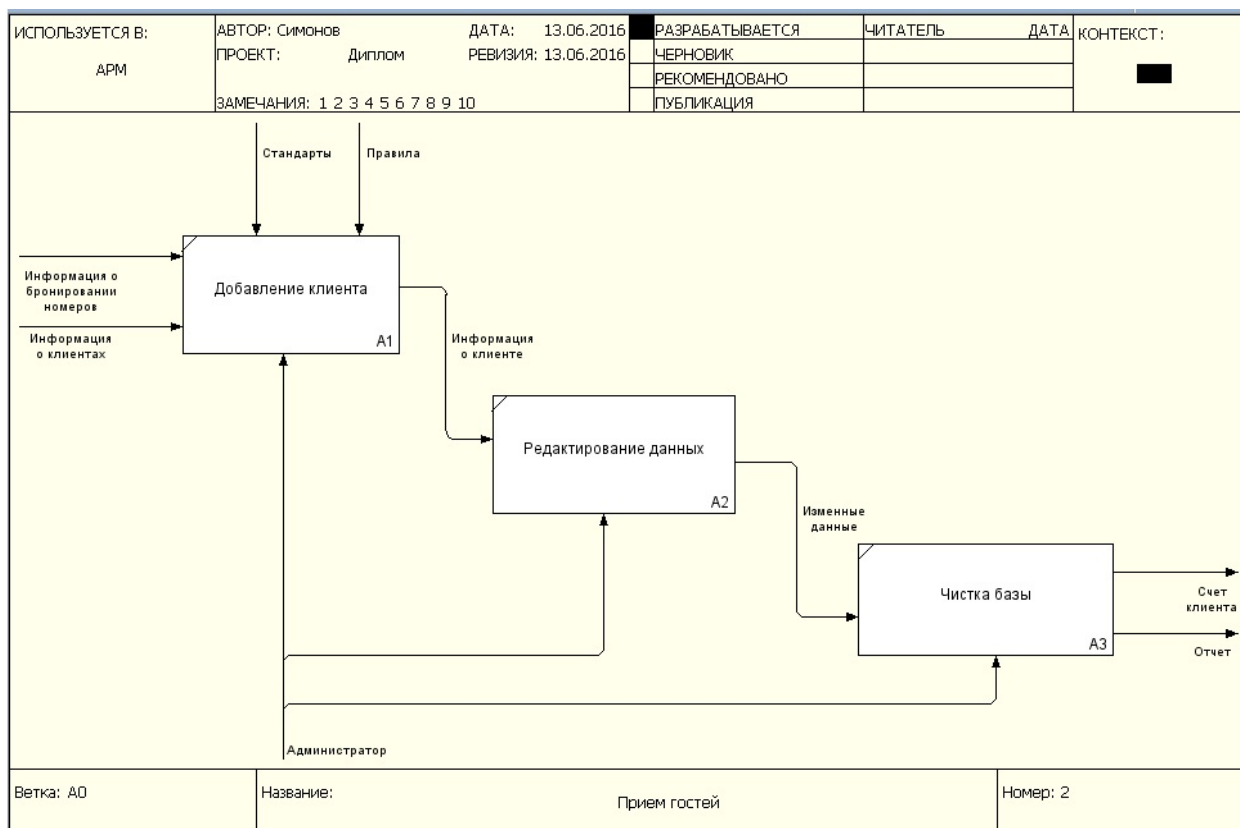


Рисунок 8 – Контекстная диаграмма второго уровня декомпозиции (AS-IS)

На процесс работы с клиентами, демонстрируемый на рисунке 7 тратится большое количество времени. Для уменьшения времени нужно внедрить АРМ, чтобы ускорить скорость обработки данных, уменьшить документооборот и уменьшить возможность допущения ошибок администратором.

Найденные в модели недостатки исправляются при создании модели ТО-ВЕ (как будет). Модель ТО-ВЕ нужна для анализа альтернативных путей решения задачи и выбора наилучшего из них. Модели ТО-ВЕ представлены на рисунках 9 и 10.

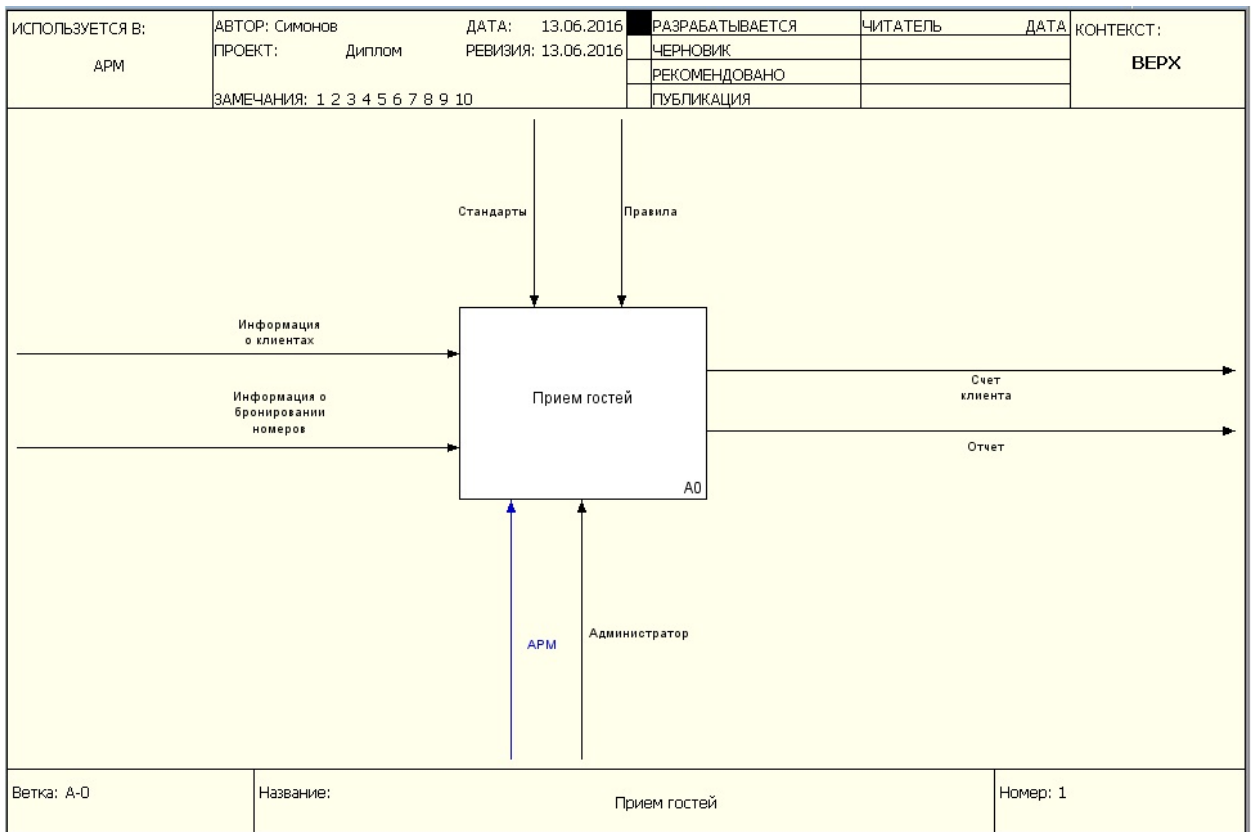


Рисунок 9 - Контекстная диаграмма (ТО-ВЕ)

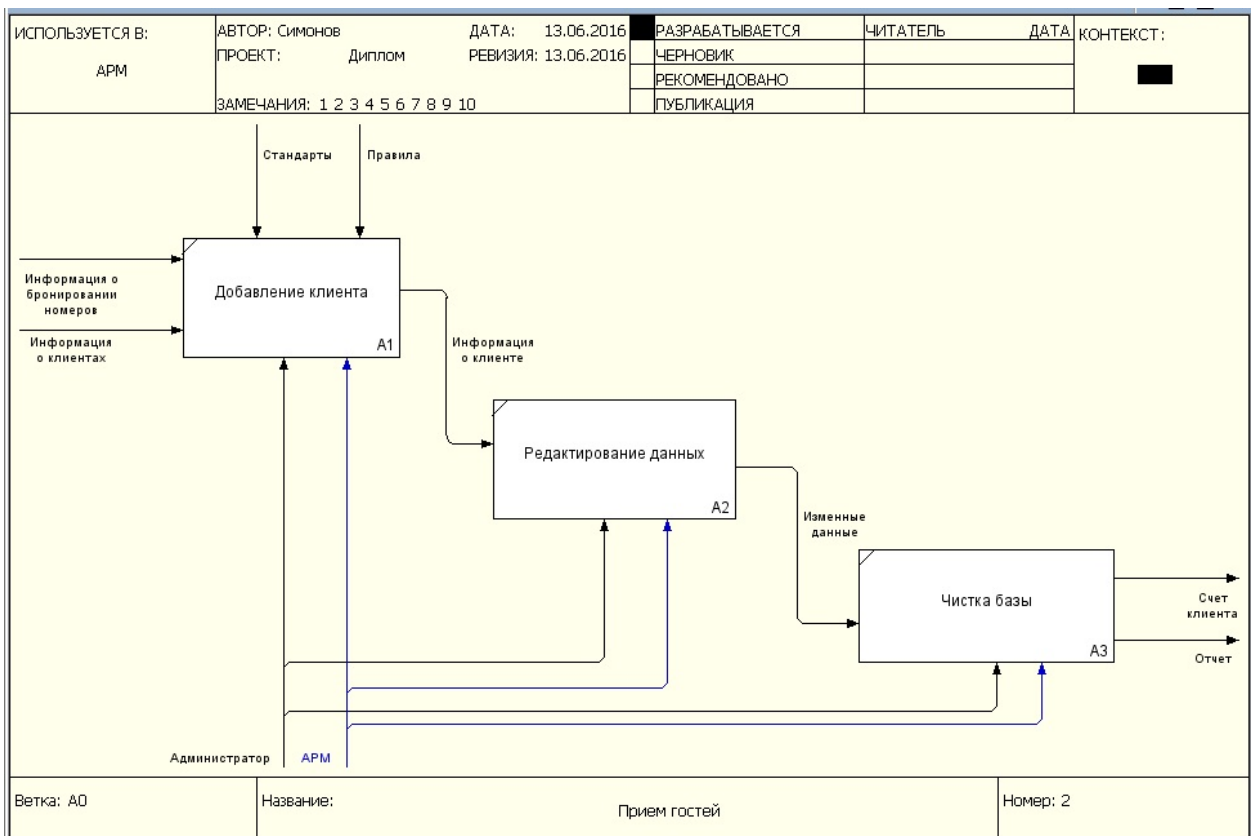


Рисунок 10 – Контекстная диаграмма второго уровня декомпозиции (ТО-ВЕ)

В результате проведения анализа предметной области была разработана структурная модель системы «AS-IS». Данная модель показывает процесс работы с клиентом.

Для улучшения существующего процесса была создана модель «ТО-ВЕ», которая позволяет ускорить и повысить эффективность работы администратора.

По модели «ТО-ВЕ» выявлены такие функции, как:

- Формирование отчета
- Запись, удаление и редактирование информации о клиенте в базе данных
- Возможность бронирования комнаты пользователем и получение информации о бронировании администратором из базы данных.

#### Use-case диаграмма

Use-case диаграмма (диаграмма прецедентов или диаграмма вариантов использования) описывает взаимоотношения и зависимости между группами вариантов использования и действующими лиц, участвующими в процессе функционирования системы.

При работе с вариантами использования важно помнить несколько простых правил:

- каждый прецедент относится как минимум к одному действующему лицу;
- каждый прецедент имеет инициатора;
- каждый прецедент приводит к соответствующему результату.

Таблица 1 - Варианты использования системы

Актер	Прецедент	Формулировка
Администратор	Ведение базы данных	Данный вариант использования позволяет администратору создавать и редактировать записи о клиентах.
Администратор	Чистка базы данных	Данный вариант использования позволяет администратору удалять записи из базы данных о клиентах и комнатах.
Администратор	Добавление и изменение информации о комнатах	Данный вариант использования позволяет администратору добавлять и изменять информацию о комнатах отеля.
Администратор	Бронирование	Данный вариант использования позволяет администратору бронировать комнаты для клиентов.
Администратор	Поиск по клиентам	Данный вариант использования позволяет администратору осуществлять быстрый поиск нужного клиента.
Администратор	Печать базы клиентов	Данный вариант использования позволяет администратору распечатывать базу данных клиентов.
Администратор	Просмотр информации о комнатах	Данный вариант использования позволяет администратору просматривать информацию о комнатах отеля.
Пользователь	Бронирование	Данный вариант использования позволяет клиентам бронировать номера самостоятельно.
Пользователь	Просмотр информации о комнатах	Данный вариант использования позволяет клиентам просматривать информацию о комнатах отеля.
Гость	Регистрация	Данный вариант использования позволяет зарегистрироваться клиентам для доступа к бронированию.
Гость	Просмотр информации о комнатах	Данный вариант использования позволяет неавторизованному пользователю просматривать информацию о комнатах отеля.

Use-case диаграмма представлена на рисунке 11.

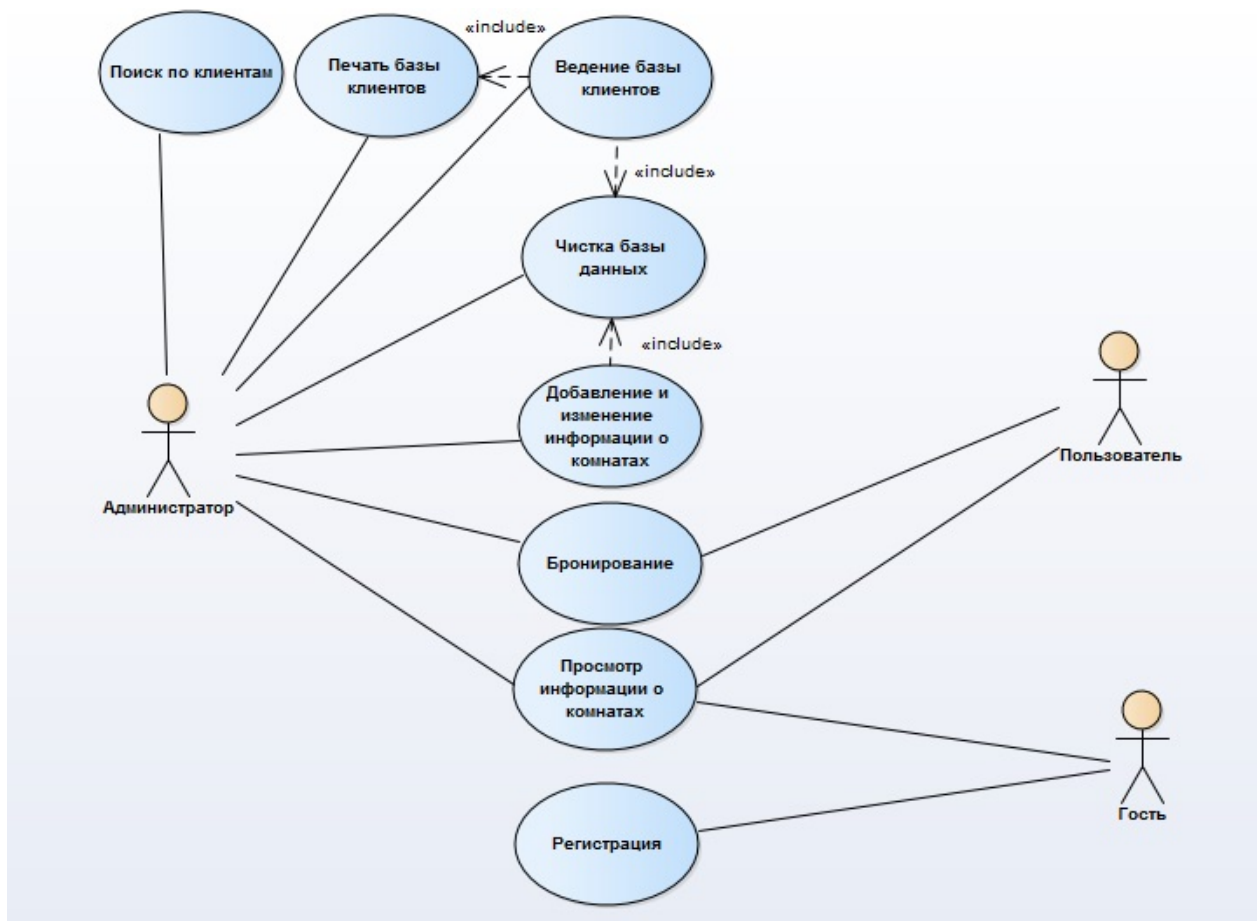


Рисунок 11 – Диаграмма вариантов использования

### Диаграмма классов

Диаграмма классов — UML-диаграмма, которая демонстрирует классы системы, атрибуты классов, их методы, а также взаимосвязи между ними.

Диаграмма классов используется для моделирования данных. Анализ предметной области позволяет выявить основные характерные для нее сущности и связи между ними. Что удобно моделируется с помощью диаграмм классов. Основой построения концептуальной схемы базы данных являются данные диаграммы.

Диаграмма классов является одним из ключевых элементов в объектно-ориентированном моделировании. На диаграмме классы представлены в рамках, содержащих три компонента:

- В верхней части указывается имя класса.

- Посередине располагаются поля (атрибуты) класса.
- Нижняя часть содержит методы класса.

На рисунке 12 представлена диаграмма классов.

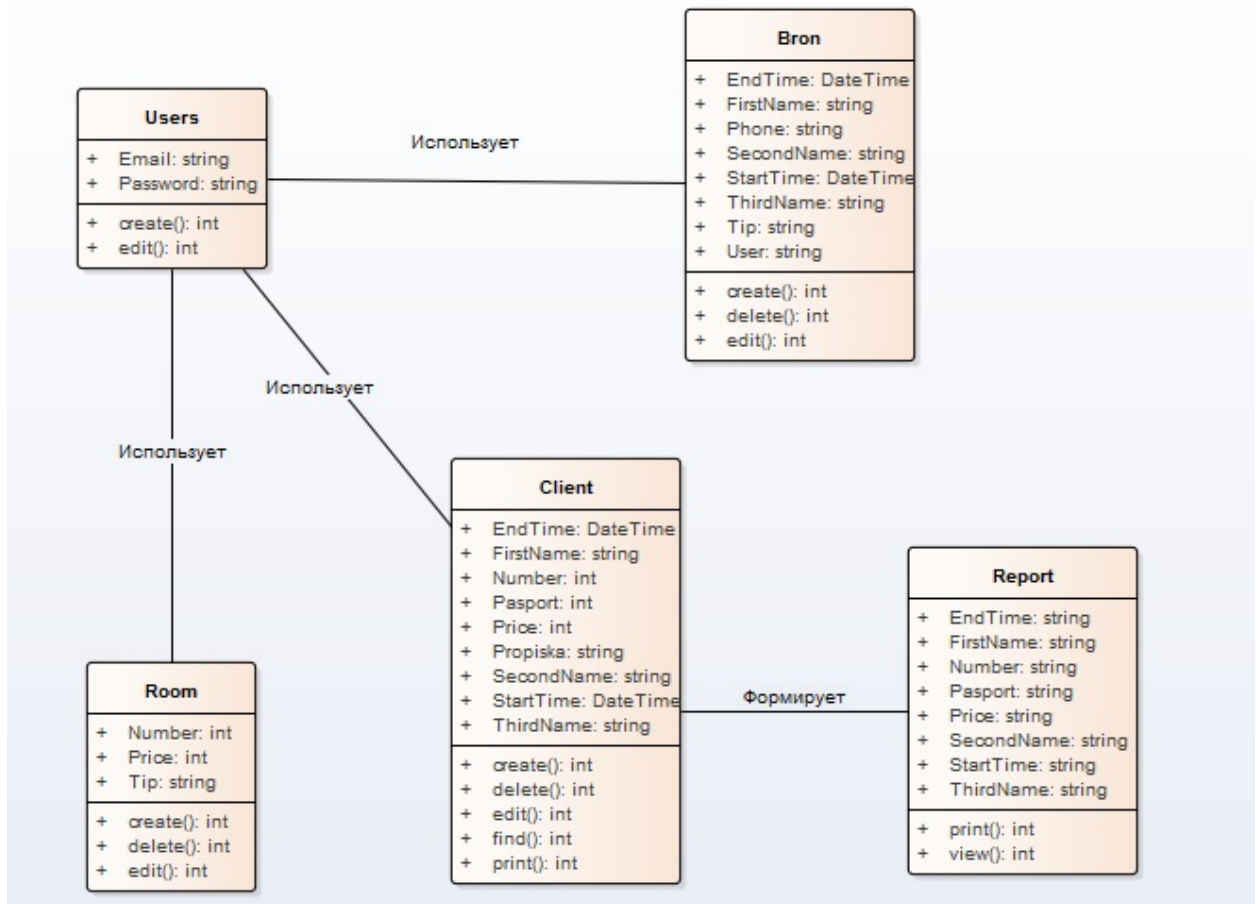


Рисунок 12 – Диаграмма классов

## 2.4 Описание пользовательского интерфейса

Главное меню для незарегистрированного пользователя представлено на рисунке 13. Незарегистрированный пользователь может посмотреть список комнат и информацию о них. При нажатии на кнопку «Забронировать» пользователю будет открываться меню авторизации, представленное на рисунке 14. Где пользователь вводит адрес электронной почты и пароль, либо переходит в меню регистрации. Так же в верхней части окна расположена кнопка регистрации и авторизации.

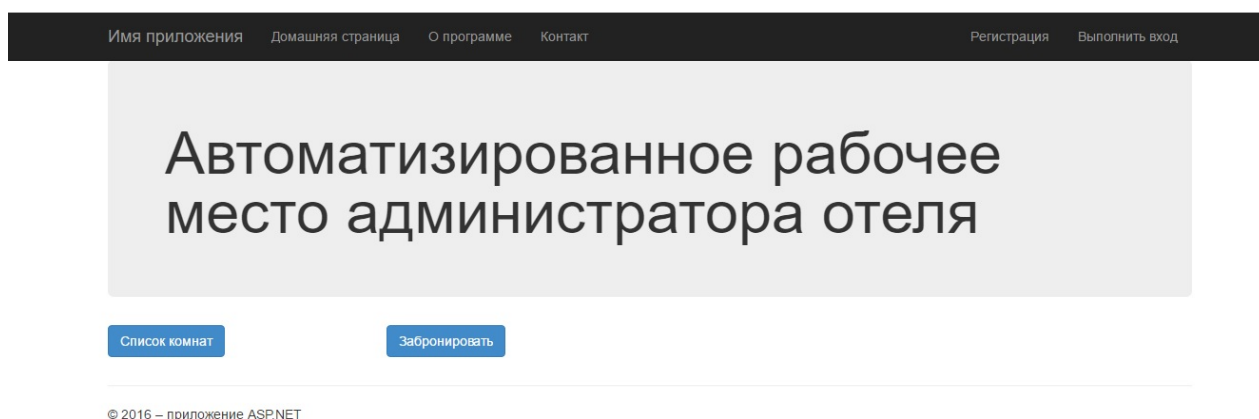


Рисунок 13 – Главное меню



Имя приложения    Домашняя страница    О программе    Контакт    Регистрация    Выполнить вход

---

### Выполнить вход.

Используйте локальную учетную запись для входа.

Адрес электронной почты

Пароль

Запомнить меня

[Регистрация нового пользователя](#)

Рисунок 14 – Авторизация

При нажатии на кнопку «регистрация» пользователь переходит в окно регистрации, которое представлено на рисунке 15. Здесь новый пользователь вводит:

- Адрес электронной почты;
- Пароль;
- Подтверждение пароля.

Имя приложения    Домашняя страница    О программе    Контакт    Регистрация    Выполнить вход

---

### Регистрация.

Создайте новую учетную запись.

Адрес электронной почты

Пароль

Подтверждение пароля

© 2016 – приложение ASP.NET

Рисунок 15 – Окно регистрации

При нажатии кнопки «регистрация» в базу данных добавляется новый пользователь, у которого есть права просматривать информацию о номерах отеля и появляется возможность забронировать номер.

На рисунке 16 представлено главное меню для пользователя с правами администратора. Оно содержит три кнопки:

- Список клиентов;
- Список комнат;
- Забронировать.

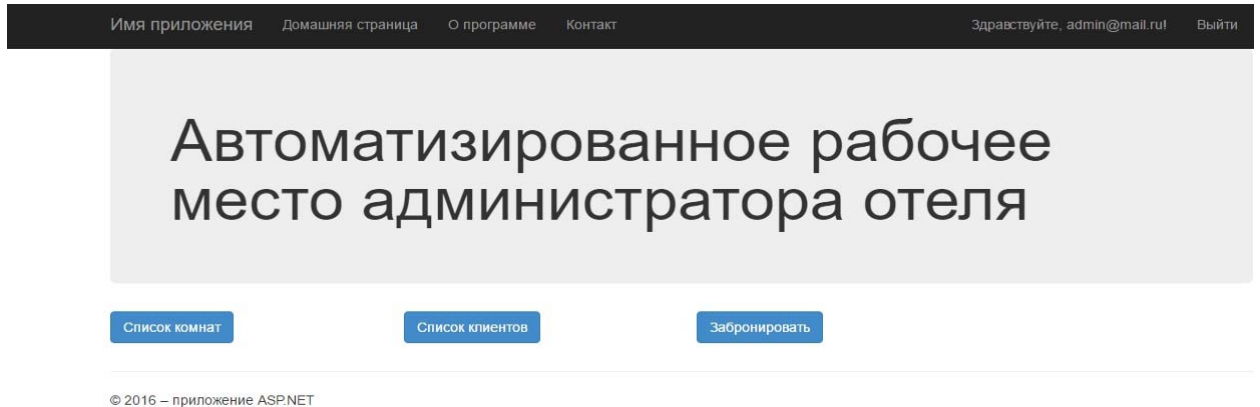


Рисунок 16 – Главное меню администратора

При нажатии кнопки «Список комнат» программа открывает страницу «Комнаты», которая представлена на рисунке 17.

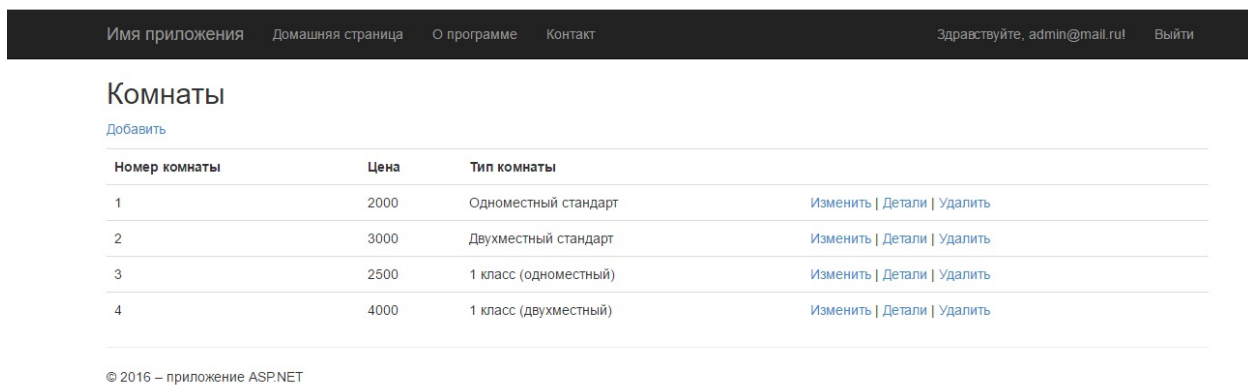


Рисунок 17 – Окно «Комнаты»

В данном окне показаны данные, представленные в виде таблицы, содержащей такие столбцы, как:

- Номер комнаты;
- Цена;
- Тип комнаты.

А также кнопки управления строками таблицы:

- Добавить
- Изменить;
- Детали;
- Удалить.

Для неавторизованного пользователя и пользователя без прав администратора на странице «Комнаты» будет присутствовать только кнопка «Детали», как показано на рисунке 18, функции добавления, удаления и изменения будут недоступны.

Номер комнаты	Цена	Тип комнаты	
1	2000	Одноместный стандарт	<a href="#">  Детали  </a>
2	3000	Двухместный стандарт	<a href="#">  Детали  </a>
3	2500	1 класс (одноместный)	<a href="#">  Детали  </a>
4	4000	1 класс (двухместный)	<a href="#">  Детали  </a>

© 2016 – приложение ASPNET

Рисунок 18 – Окно «Комнаты» пользователя

При нажатии в главном меню на кнопку «Список клиентов» администратор переходит в окно «Клиент», представленное на рисунке 19.

## Клиент

[Добавить](#)

Введите фамилию для поиска

Фамилия	Имя	Отчество	Номер паспорта	Место жительства	Дата заселения	Дата выселения	Номер комнаты	Цена	
test	test	test	2423	test	18/05/2016	19/05/2016	1	2000	<a href="#">Изменить</a>   <a href="#">Детали</a>   <a href="#">Удалить</a>
Иванов	Иван	Иванович	123432432	г. Красноярск	17/05/2016	20/05/2016	3	7500	<a href="#">Изменить</a>   <a href="#">Детали</a>   <a href="#">Удалить</a>

[Распечатать](#)

© 2016 – приложение ASP.NET

Рисунок 19 – Окно «Клиент»

В данном окне показаны данные, представленные в виде таблицы, содержащей такие столбцы, как:

- Фамилия;
- Имя;
- Отчество;
- Номер паспорта;
- Место жительства;
- Дата заселения;
- Дата выселения;
- Номер комнаты;
- Цена.

А также кнопки управления строками таблицы:

- Добавить
- Изменить;
- Детали;
- Удалить.

Присутствует строка поиска и кнопка «Распечатать».

Работа поиска по базе данных показана на рисунке 20.

## Клиент

[Добавить](#)

Фамилия	Имя	Отчество	Номер паспорта	Место жительства	Дата заселения	Дата выселения	Номер комнаты	Цена
Иванов	Иван	Иванович	123432432	г. Красноярск	17/05/2016	20/05/2016	3	7500

[Изменить](#) | [Детали](#) | [Удалить](#)[Распечатать](#)

© 2016 – приложение ASP.NET

Рисунок 20 – Поиск

При нажатии на кнопку распечатать формируется и скачивается PDF-файл, представленный на рисунке 21.

### Отчет "Список клиентов"

<u>Фамилия</u>	<u>Имя</u>	<u>Отчество</u>	<u>Паспортные данные</u>	<u>Дата заселения</u>	<u>Дата выселения</u>	<u>Комната</u>	<u>Цена</u>
test	test	test	2 423	май 18 2016	май 19 2016	1	2 000
Иванов	Иван	Иванович	123 432 432	май 17 2016	май 20 2016	3	7 500

Рисунок 21 – Пример печати отчета «Список клиентов»

При нажатии на кнопку «Детали» появляется окно, представленное на рисунке 22.

## Детали

Клиент

Фамилия	Иванов
Имя	Иван
Отчество	Иванович
Номер паспорта	123432432
Место жительства	г. Красноярск
Дата заселения	17/05/2016
Дата выселения	20/05/2016
Номер комнаты	3
Цена	7500

[Изменить](#) | [Назад](#)

© 2016 – приложение ASPNET

Рисунок 22 – Окно «Детали»

На рисунке 23 представлено окно «Бронирования номера», которое содержит строки:

- Фамилия;
- Имя;
- Отчество;
- Дата заселения;
- Дата выселения;
- Тип комнаты;
- Номер телефона.

После заполнения данные запишутся в таблицу бронирования. И у администратора будет возможность при нажатии кнопки «Детали» посмотреть, какой пользователь забронировал комнату. Что представлено на картинке 24.

## Создать

### Бронирование

<b>Фамилия</b>	<input type="text" value="Смирнов"/>
<b>Имя</b>	<input type="text" value="Александр"/>
<b>Отчество</b>	<input type="text" value="Петрович"/>
<b>Дата заселения</b>	<input type="text" value="17.06.2016"/>
<b>Дата выселения</b>	<input type="text" value="21.06.2016"/>
<b>Тип комнаты</b>	<input type="text" value="Выберите вариант"/>
<b>Номер телефона</b>	<input type="text" value=""/>

Выберите вариант

- Выберите вариант
- Одноместный стандарт
- Двухместный стандарт
- 1 класс (одноместный)
- 1 класс (двухместный)

[Назад](#)

© 2016 – приложение ASP.NET

Рисунок 23 – Окно «Бронирование номера»

## Детали

### Бронирование

<b>Фамилия</b>	Иванов
<b>Имя</b>	Иван
<b>Отчество</b>	Иванович
<b>Дата заселения</b>	17/06/2016
<b>Дата выселения</b>	21/06/2016
<b>Тип комнаты</b>	Двухместный стандарт
<b>Номер телефона</b>	89121231232
<b>Пользователь</b>	Sdad1@mail.ru

[Изменить](#) | [Назад](#)

© 2016 – приложение ASP.NET

Рисунок 24 – Окно «Детали бронирования»

## ЗАКЛЮЧЕНИЕ

Поставленные в начале проектирования задачи были реализованы в результате выполнения выпускной квалификационной работы. Было создано автоматизированное рабочее место администратора отеля. Данная система предоставляет специалисту ряд возможностей:

- Автоматизация процесса работы с клиентами;
- Удобный и понятный интерфейс для эффективной работы с клиентами;
- Добавление и изменение информации о имеющихся номерах;
- Поиск клиента по базе данных;
- Печать базы данных клиентов;
- Установление цен.

Реализована возможность бронирования клиентами номеров через сеть интернет.

При выполнении работы был проведен анализ предметной области в результате которого, можно сделать вывод об актуальности, и полезных свойствах приложения.



## **СПИСОК СОКРАЩЕНИЙ**

АРМ – Автоматизированное рабочее место.

БД – База данных.

ВКР – Выпускная квалификационная работа.

ИС – Информационная система.

ПО – Программное обеспечение.

СУБД – Система управления базами данных.

MVC – Model – View - Controller

UML – Unified Modeling Language.

VS – VisualStudio.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. -Введ. 30.12.2013. – Красноярск: СФУ, 2014.–60 с.
2. Автоматизированное рабочее место [электронный ресурс]: Энциклопедия по машиностроению – Режим доступа:<http://mash-xxl.info/info/16271/>
3. Информационные системы [электронный ресурс]: Информационные системы и сети –Режим доступа: [http://tsput.ru/res/informat/sist\\_seti\\_fm0/lekcii/lekcij-1.html](http://tsput.ru/res/informat/sist_seti_fm0/lekcii/lekcij-1.html)
4. Коюпченко, И. Н. Автоматизированное рабочее место (АРМ) специалиста: учебное пособие, Ч. 1 / И.Н. Коюпченко, Н.Ф. Телешова; Краснояр. гос. торг-эконом. инстит. – Красноярск, 2009. – 84 с.
5. Модель данных [электронный ресурс]: Википедия – Режим доступа:[https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C\\_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)
6. Советов, Б. Я. Информационные технологии: учебное пособие для вузов/ Б. Я. Советов, В. В. Цехановский. – Москва: Высш. шк., 2003. – 263 с.
7. Трофимов, С. Варианты использования (UseCase) [Электронный ресурс]: Клуб разработчиков программных систем – Режим доступа: [http://www.caseclub.ru/articles/use\\_case.html](http://www.caseclub.ru/articles/use_case.html)
8. Федотова, Е. Л. Информационные технологии и системы: учебное пособие / Е. Л. Федотова. – Москва: ИД «Форум»: ИНФРА-М, 2014. – 352 с.
9. Хомоненко, А. Д. Базы данных: учебник для высших учебных заведений. – 4-е изд., доп. и перераб. / под ред. проф. А.Д. Хомоненко. – Санкт-Петербург: КОРОНА принт, 2004. – 736с.