МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

###### ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ФГБО ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ

(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)»

ИЖЕНЕРНО-ЭКОНОМИЧЕСКИЙ ИНСТИТУТ

Кафедра «Прикладная и бизнес-информатика»

ОТЧЕТ ПО КУРСОВОМУ ПРОЕКТУ

по дисциплине «Управление разработкой информационных систем»

Тема: «Разработка информационной системы резервирования гостиничных номеров»

Студент группы Лавренов Ю.С. ИЭс-161-16

(Ф.И.О.)

Руководитель к.т.н., Петров С.А.

(уч. степень, звание, Ф.И.О.)

|  |  |  |
| --- | --- | --- |
| Сдана на проверку |  |  |
| Возвращена на доработку |  |  |
| Допущена к защите |  |  |
| Оценка |  |  |
|  |

Москва 2018г.

Оглавление

[1.Описание предметной области 3](#_Toc513653263)

[2. Обоснование выбора объекта автоматизации 6](#_Toc513653264)

[2.1. Назначение, цели создания информационной системы резервирования гостиничных номеров. 6](#_Toc513653265)

[2.2. Общее описание 6](#_Toc513653266)

[2.3. Требования к системе. 6](#_Toc513653267)

[2.4. Требования к функциональным характеристикам. 7](#_Toc513653268)

[2.5. Функциональное назначение системы. 7](#_Toc513653269)

[2.6. Не функциональное назначение системы. 7](#_Toc513653270)

[2.7. Диаграмма вариантов использования. 8](#_Toc513653271)

[3. Описание выбранных для реализации технологий 10](#_Toc513653272)

[4. Проектирование информационной системы 13](#_Toc513653273)

[4.1. Диаграмма базы данных экскурсионного бюро. 15](#_Toc513653275)

[5. Реализация информационной системы 22](#_Toc513653276)

[Заключение 34](#_Toc513653277)

[Литература 35](#_Toc513653278)

## 1.Описание предметной области

Предметной областью информационной системы (ИС) была выбрана деятельность резервирования гостиничных номеров – регистрации постояльцев гостиницы и выбор номера для бронирования.

В настоящее время информационная система используется во многих гостиницах, так как это позволяет улучшить работу сотрудников гостиницы, тем самым уменьшить временные затраты на обработку данных и ведение отчетов.

Гостиница – это средство размещения (дом, здание, часть здания, иные постройки) состоящее из определенного количества номеров для временного проживания, имеющее единое руководство, предоставляющее набор услуг (минимум – заправка постелей, уборка номера и санузла) и сгруппированная в классы в соответствие с предоставляемыми услугами и оборудованию номеров.

Информационная система — предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию (ISO/IEC 2382:2015).

Информационная система предназначена для своевременного обеспечения надлежащих людей надлежащей [информацией](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), то есть для удовлетворения конкретных информационных потребностей в рамках определенной [предметной области](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C), при этом результатом функционирования информационных систем является информационная продукция — документы, [информационные массивы](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2), [базы данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и информационные услуги.

Основные задачи информационных систем:

* **Интерпретация данных**. Под интерпретацией понимается процесс определения смысла данных, результаты которого должны быть согласованными и корректными. Обычно предусматривается многовариантный анализ данных.
* **Диагностика.** Под диагностикой понимается процесс соотношения объекта с некоторым классом объектов и/или обнаружение неисправности в некоторой системе. Неисправность - это отклонение от нормы. Такая трактовка позволяет с единых теоретических позиций рассматривать и неисправность оборудования в технических системах, и заболевания живых организмов, и всевозможные природные аномалии.
* **Мониторинг.** Основная задача мониторинга - непрерывная интерпретация данных в реальном времени и сигнализация о выходе тех или иных параметров за допустимые пределы.
* **Проектирование.** Проектирование состоит в подготовке спецификаций на создание «объектов» с заранее определёнными свойствами. Под спецификацией понимается весь набор необходимых документов - чертёж, пояснительная записка и т.д. Основные проблемы здесь - получение чёткого структурного описания знаний об объекте и проблема «следа».
* **Прогнозирование.** Прогнозирование позволяет предсказывать последствия некоторых событий или явлений на основании анализа имеющихся данных. Прогнозирующие системы логически выводят вероятные следствия из заданных ситуаций.
* **Планирование.** Под планированием понимается нахождение планов действий, относящихся к объектам, способным выполнять некоторые функции. В таких ЭС используются модели поведения реальных объектов с тем, чтобы логически вывести последствия планируемой деятельности.

Обучение. Под обучением понимается использование компьютера для обучения какой-то дисциплине или предмету. Системы обучения диагностируют ошибки при изучении какой-либо дисциплины с помощью ЭВМ и подсказывают правильные решения.

* **Управление.** Под управлением понимается функция организованной системы, поддерживающая определенный режим деятельности. Такого рода ЭС осуществляют управление поведением сложных систем в соответствии с заданными спецификациями.
* **Поддержка принятия решений.** Поддержка принятия решения - это совокупность процедур, обеспечивающая лицо, принимающее решения, необходимой информацией и рекомендациями, облегчающие процесс принятия решения. Эти ЭС помогают специалистам выбрать и/или сформировать нужную альтернативу среди множества выборов при принятии ответственных решений.

В современных гостиницах масса различных номеров с уровнем комфортности сервиса, а за счет этого соответственно, зависит оплата за номер.

Все размещаемые клиенты в гостиницы заполняют карточку регистрации, где необходимо указать: фамилию, имя, отчество, дату рождения, адрес, паспортные данные, время заезда и время выезда. Информационная система предоставляет услугу бронирования номеров.

Информационная резервирования гостиничных номеров должна обеспечить:

* Регистрацию, изменение, удаление персональных данных о клиенте, которая включает в себя фамилию, имя, отчество, дату рождения, адрес, паспортные данные.
* Регистрацию, изменение, удаление полной информации об номерах, которая включает в себя количество комнат, тип номеров.
* Регистрацию, изменение, удаление персональных данных сотрудников, включающая в себя фамилию, имя, отчество, адрес, телефон, зарплату, должность.
* Регистрацию, изменение, удаление бронирования номеров.
* Прием заявок на бронирование.
* Расчет стоимости проживания.
* Справочная информация о услугах и контактах.
* Вход в приложения для сотрудника, используя логин и пароль.

## 

## 2. Обоснование выбора объекта автоматизации

## 2.1. Назначение, цели создания информационной системы резервирования гостиничных номеров.

Назначение ИС:

Информационная система предназначена для регистрации постояльцев гостиницы, и может быть использована администратором или менеджером гостиницы. Каждый сотрудник может обеспечивать ввод, редактирование и удаление имеющийся информации, обладая своим уровнем доступа. Тем самым информационная система сделает работу гостиницы более удобной и эффективной.

Цели создания ИС:

Целью является разработка программного обеспечения для автоматизации деятельности сотрудников гостиничного бизнеса. Она автоматизирует работу по ведению статусов о свободных и занятых номерах, обрабатывать запросы на бронирование номеров и включаемых услугах гостиницы, что позволит сократить объем работы.

Характеристика объектов автоматизации:

В данном курсовом проекте в качестве исследуемой организации рассматривается гостиничный бизнес, который предоставляет бронирование номеров для постояльцев.

## 2.2. Общее описание

Информационная система представляет собой хранение данных, которое обеспечивает сохранность, упорядоченность, поиск и обработку информации резервирования гостиничных номеров.

## 2.3. Требования к системе.

Основное требование к системе – дружественный интерфейс пользователя. Кроме того, следует учесть, что система ориентирована на конечного пользователя, который не должен знать структуру программного кода, внутренние представление данных, средства программирования, которые использовались при разработке системы.

Кроме того, работа в приложении должна быть построена наиболее эффективно, то есть должны быть реализованы самые необходимые пользователю функции. Пользователь должен иметь доступ ко всем данным в БД, а также иметь возможность для добавления, редактирования, удаления.   
С информационной системой «Резервирования гостиничных номеров» должен работать квалифицированный сотрудник, который имеет опыт работы с программными средствами, используемыми в данной работе, и может модернизировать ее.

## 2.4. Требования к функциональным характеристикам.

Система должна обеспечить возможность выполнения перечисленных ниже свойств:

* Гибкость – настраивается на любую структуру, для широкого круга деятельности;
* Безопасность – ограничение общего доступа к заранее определенным ресурсам БД автосалона;
* Целостность и доступность – документы в базе данных букмекерской конторы должны быть представлены в общепринятых в организации форматах, защищенных от изменений;
* Возможность подключения независимых модулей для импорта данных из внешних источников в текущую базу данных.

## 2.5. Функциональное назначение системы.

Программное изделие должно выполнять следующие функции:

* Регистрацию, изменение, удаление персональных данных о клиенте, которая включает в себя фамилию, имя, отчество, дату рождения, адрес, паспортные данные.
* Регистрацию, изменение, удаление полной информации об номерах, которая включает в себя количество комнат, тип номеров.
* Регистрацию, изменение, удаление персональных данных сотрудников, включающая в себя фамилию, имя, отчество, адрес, телефон, зарплату, должность.
* Регистрацию, изменение, удаление бронирования номеров.
* Прием заявок на бронирование.
* Расчет стоимости проживания.
* Справочная информация о услугах и контактах.
* Вход в приложения для сотрудника, используя логин и пароль.

## 2.6. Не функциональное назначение системы.

* ИС должна быть написана на Borland Delphi7.
* ИС должна работать на ОС Windows.

## 2.7. Диаграмма вариантов использования.

Субъект (actor) — это некто или нечто (человек, машина и т.д.) взаимодействующее с системой. Субъект взаимодействует с прецедентом, ожидая получить некий полезный результат.

Типичным графическим изображением субъекта является «штриховой человечек». В общем случае субъект может быть показан в виде прямоугольного символа класса. Подобно обычному классу субъект может обладать атрибутами и операциями (связанными с событиями, сообщения о которых он отправляет и получает). На рисунке 1 показаны субъекты: клиент и менеджер.

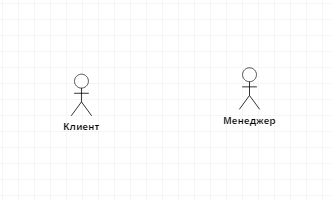


Рисунок 1 - Субъекты системы «резервирования гостиничных номеров»

Прецеденты удовлетворяют функциональные требования за счет предоставления субъекту полезного результата. При этом не имеет значения, в какой последовательности решает бизнес-аналитик свои задачи: сначала обозначает субъектов, а затем прецеденты, или наоборот. Прецедент наглядно показывает варианты использования системы. На рисунке 2 показаны прецеденты моделируемой системы.

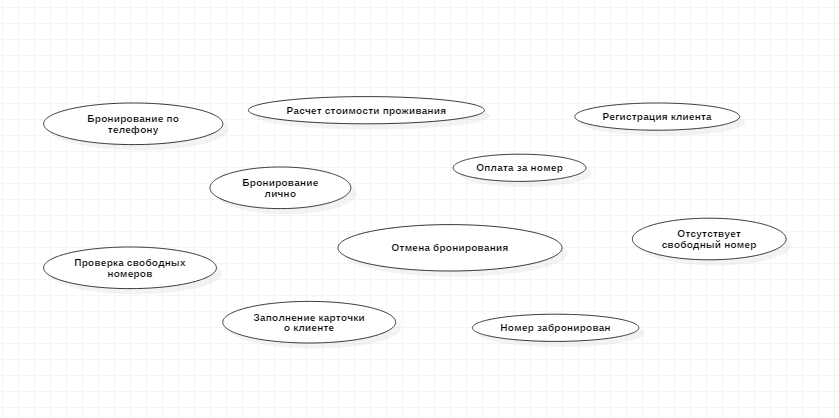


Рисунок 2 - Прецеденты системы «резервирования гостиничных номеров»

Диаграмма вариантов использования – это наглядное графическое представление субъектов и прецедентов и их взаимодействий в системе вместе с любыми дополнительными определениями и спецификациями. Она представляет собой не просто некую схему, а является полностью документированной моделью предполагаемого поведения системы. Диаграмма прецедентов для моделируемой системы представлена на   
рисунке 3.

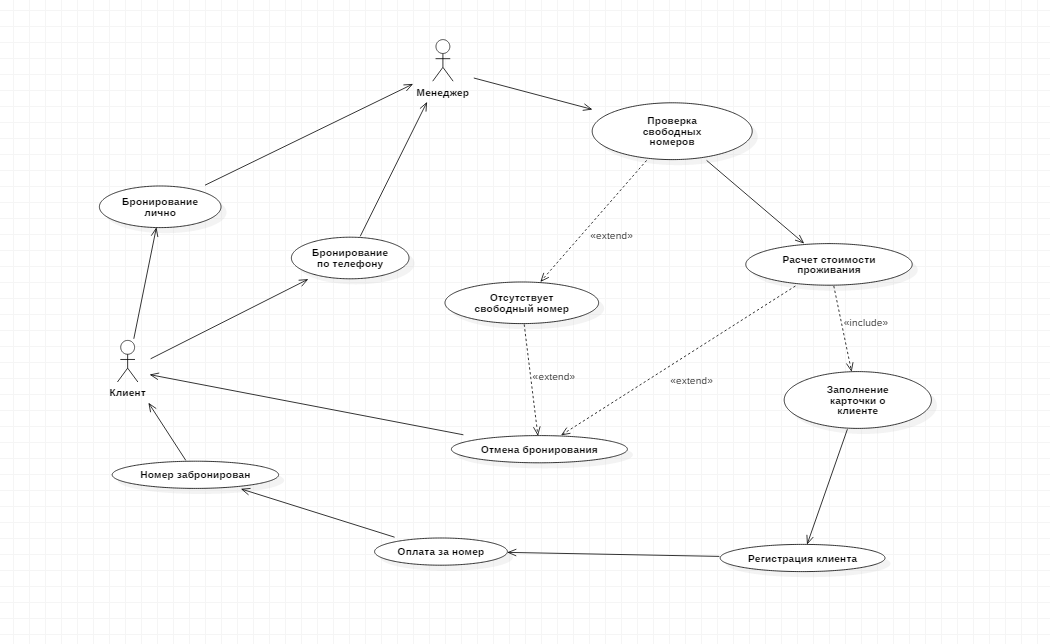


Рисунок 3 - Диаграмма вариантов использования для системы «резервирования гостиничных номеров»

## 3. Описание выбранных для реализации технологий

В качестве технологий для разработки системы выступают:

* Microsoft SQL Server 2008
* Borland Delphi7
* HTML Help Workshop

1) Borland Delphi7 — В качестве среды разработки программы «Информационная система резервирования гостиничных номеров», был выбран Borland Delphi 7.0, так как наиболее популярный и знакомый язык программирования, так как все курсовые проекты были реализованы на нем.

Delphi – императивный, структурированный, объектно – ориенти-рованный язык программирования со строгой статической типизацией переменных.

Borland Delphi удобен тем, что в комплект включает в себя больше ста готовых компонентов, для реализации большого спектра задач. Программа Delphi при ее использовании помогает разработчику, разработать программу в кратчайшие сроки, за счет того, что есть наличие множества готовых компонентов, которые существенно ускоряют процесс проектирования, при этом снижается написание программного кода.

Borland Delphi позволяет реализовать все необходимые функции и задачи, предоставляет возможность удобного и простого интерфейса.

2) Microsoft SQL Server 2008 — доступ к информации всегда и везде.

Решение Microsoft SQL Server 2008 представляет собой высокопроизводительную платформу обработки и анализа данных для бизнеса любого размера, которая отвечает всем современным требованиям по работе с данными любых типов и поддержке удобной и быстрой разработки приложений. Продукт реализует концепцию комплексной обработки данных и позволяет преобразовывать ценную информацию в знания и решения на основе анализа собранных данных. В базах данных SQL Server 2008 можно хранить любую структурированную, полуструктурированную или неструктурированную информацию, такую, например, как изображения и мультимедиа из самых разнородных источников данных. Продукт предлагает большой набор интегрированных служб, расширяющих возможности обработки, который позволяет составлять запросы, выполнять поиск, проводить синхронизацию, формировать отчеты и анализировать данные. SQL Server 2008 обеспечивает обращение к данным из любого приложения, разработанного с применением технологий Microsoft .NET и Visual Studio, в том числе — с помощью Microsoft BizTalk Server — из приложений, построенных на базе сервис-ориентированной архитектуры (SOA) и корпоративных бизнес-приложений других разработчиков. Сотрудники, отвечающие за сбор и анализ информации, могут работать с данными, пользуясь привычными приложениями, например, программами Microsoft Office 2007.

На базе Microsoft SQL Server 2008 могут быть построены решения для компаний малого, среднего и крупного бизнеса. SQL Server 2008 выпускается в двух основных редакциях Standard и Enterprise. На основе последней создана также редакция для разработчиков Developer Edition, лицензия на которую позволяет разрабатывать и тестировать системы и приложения.

Enterprise-версия системы SQL Server 2008 представляет собой комплексную платформу, которая позволяет работать даже с самыми требовательными корпоративными OLTP-системами и хранилищами данных. Она обладает значительной масштабируемостью, возможностью создавать громадные хранилища данных, продвинутыми средствами анализа и усиленной безопасностью, что позволяет использовать ее как основу для критически важных бизнес-приложений. Эта редакция позволяет консолидировать серверы и выполнять большое число OLTP-операций и крупные отчеты.

Редакцию Microsoft SQL Server 2008 Enterprise характеризуют:

• **Высокий уровень доступности**. Непрерывность бизнес-процессов обеспечивается благодаря защите данных от дорогостоящих человеческих ошибок и максимальному уменьшению сроков аварийного восстановления.

• **Производительность и масштабируемость**. Инфраструктура на основе SQL Server 2008 Enterprise позволяет справиться с любыми пиковыми нагрузками.

• **Безопасность.** Вопросы конфиденциальности, а также соответствия нормативным требованиям решаются с помощью встроенных средств защиты от несанкционированного доступа.

• **Управляемость.** Автоматические диагностика, калибровка и настройка инфраструктуры позволяют управлять огромными объемами данных, значительно сократив издержки на управление и обслуживание.

• **Бизнес-аналитика**. SQL Server 2008 Enterprise помогает легко собрать и проанализировать большие объемы данных из хранилищ или киосков.

Редакция Microsoft SQL Server 2008 Standard — это законченная платформа для управления данными и выполнения операций бизнес-аналитики. Она демонстрирует лучшие в своем классе показатели простоты использования и степени управляемости ключевых бизнес-приложений.

SQL Server 2008 Standard предлагает следующие преимущества:

• **Простота использования**. Легкость в эксплуатации позволяет снизить операционные издержки и расходы на разработку решений на основе платформы.

• **Управляемость.** Интуитивно понятные средства управления и автоматизированного администрирования помогают эффективно управлять бизнес-приложениями.

• **Создание отчетов и аналитика**. С помощью встроенных технологий анализа и создания отчетности можно легко и быстро получать практическую и значимую информацию и принимать обоснованные решения.

3) **HTML Help Workshop** – Мощный инструмент для создания файлов справок, электронных книг, инструкций и прочих документов в формате CHM. Программа позволяет редактировать гипертекстовое содержание справочной системы, индексировать страницы, а используя CSS, можно привести дизайн многочисленных страниц к общему виду

## 

## 4. Проектирование информационной системы

Проектирование информационных систем всегда начинается с определения цели проекта. Основная задача любого успешного проекта заключается в том, чтобы на момент запуска системы и в течение всего времени ее эксплуатации можно было обеспечить:

* требуемую функциональность системы и степень адаптации к изменяющимся условиям ее функционирования;
* требуемую пропускную способность системы;
* безотказную работу системы в требуемом режиме, иными словами - готовность и доступность системы для обработки запросов пользователей;
* простоту эксплуатации и поддержки системы;
* необходимую безопасность.

Производительность является главным фактором, определяющим эффективность системы. Хорошее проектное решение служит основой высокопроизводительной системы.

Проектирование информационных систем охватывает три основные области:

* проектирование объектов данных, которые будут реализованы в базе данных;
* проектирование программ, экранных форм, отчетов, которые будут обеспечивать выполнение запросов к данным;
* учет конкретной среды или технологии, а именно: топологии сети, конфигурации аппаратных средств, используемой архитектуры (файл-сервер или клиент-сервер), параллельной обработки, распределенной обработки данных и т.п.

В реальных условиях проектирование - это поиск способа, который удовлетворяет требованиям функциональности системы средствами имеющихся технологий с учетом заданных ограничений.

К любому проекту предъявляется ряд абсолютных требований, например, максимальное время разработки проекта, максимальные денежные вложения в проект и т.д. Одна из сложностей проектирования состоит в том, что оно не является такой структурированной задачей, как анализ требований к проекту или реализация того или иного проектного решения.

Считается, что сложную систему невозможно описать в принципе. Это, в частности, касается систем управления предприятием. Одним из основных аргументов является изменение условий функционирования системы, например, директивное изменение тех или иных потоков информации новым руководством. Еще один аргумент - объемы технического задания, которые для крупного проекта могут составлять сотни страниц, в то время как технический проект может содержать ошибки. Возникает вопрос: а может, лучше вообще не проводить обследования и не делать никакого технического проекта, а писать систему «с чистого листа» в надежде на то, что произойдет некое чудесное совпадение желания заказчика с тем, что написали программисты, а также на то, что все это будет стабильно работать?

Жизненный цикл программного обеспечения представляет собой модель его создания и использования. Модель отражает его различные состояния, начиная с момента возникновения необходимости в данном ПО и заканчивая моментом его полного выхода из употребления у всех пользователей. Известны следующие модели жизненного цикла:

* Каскадная модель. Переход на следующий этап означает полное завершение работ на предыдущем этапе.
* Поэтапная модель с промежуточным контролем. Разработка ПО ведется итерациями с циклами обратной связи между этапами. Межэтапные корректировки позволяют уменьшить трудоемкость процесса разработки по сравнению с каскадной моделью; время жизни каждого из этапов растягивается на весь период разработки.
* Спиральная модель. Особое внимание уделяется начальным этапам разработки - выработке стратегии, анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов (макетирования). Каждый виток спирали предполагает создание некой версии продукта или какого-либо его компонента, при этом уточняются характеристики и цели проекта, определяется его качество и планируются работы следующего витка спирали.

В проектировании информационной системы резервирования гостиничных номеров», был применен спиральный подход, то есть происходило поэтапное уточнение требований, частичная реализация в рабочих примерах, тестирование, а дальше цикл повторялся.

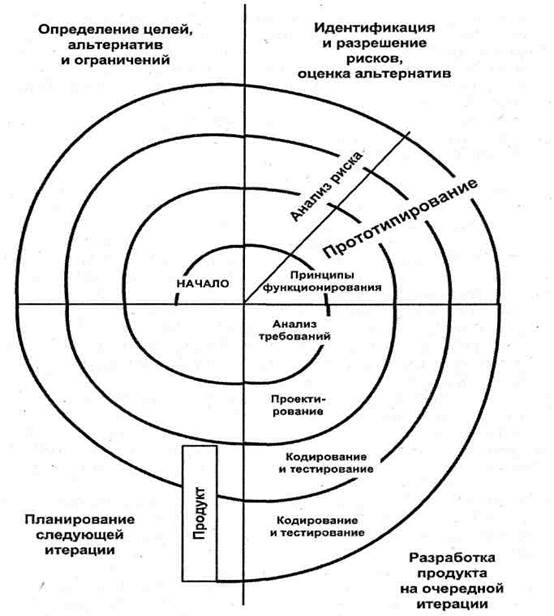


Рисунок 4 – Спиральная модель жизненного цикла программного обеспечения

## 4.1. Диаграмма базы данных экскурсионного бюро.

Целью построения диаграммы базы данных является получение графического представления физической структуры исследуемой предметной области.

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 спроектированной базе данных, получились таблицы.

**tStaff (idStaff, idRole, Name, SuName, Login, Tel, Position)** –предназначена для хранения данных о сотрудниках (таблица 1).

Таблица 1–Сотрудники

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| idStaff | int | РК |
| idRole | int | FK |
| Name | nvarchar(255) |  |
| SndName | nvarchar(255) |  |
| SurName | nvarchar(255) |  |
| Login | nvarchar(255) |  |
| Tel | nvarchar(255) |  |
| Position | nvarchar(255) |  |

**tStaffАrc (idStaff, idRole, Name, SuName, Login, Tel, Position)** –предназначена для хранения данных архива сотрудников (таблица 2).

Таблица 2– СотрудникиАрхив

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| idStaff | int | РК |
| idRole | int | FK |
| Name | nvarchar(255) |  |
| SndName | nvarchar(255) |  |
| SurName | nvarchar(255) |  |
| Login | nvarchar(255) |  |
| Tel | nvarchar(255) |  |
| Position | nvarchar(255) |  |

**tBookings (idBooking, idClients, idStaff, idNumber, idBookingStatus, BookingDate, DateFrom, DateTo, Price)** –предназначена для хранения данных о бронировании номеров (таблица 3).

Таблица 3– Бронирование

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| idBooking | int | РК |
| idClients | int | FK |
| idStaff | int | FK |
| idNumber | int | FK |
| idBookingStatus | int | FK |
| BookingDate | datetime |  |
| DateFrom | datetime |  |
| DateTo | datetime |  |
| Price | decimal (9,2) |  |

**tBookingsArc (idBooking, idClients, idStaff, idNumber, idBookingStatus, BookingDate, DateFrom, DateTo, Price)** –предназначена для хранения данных архива бронирования номеров (таблица 4).

Таблица 4– БронированиеАрхив

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| idBooking | int | РК |
| idClients | int | FK |
| idStaff | int | FK |
| idNumber | int | FK |
| idBookingStatus | int | FK |
| BookingDate | datetime |  |
| DateFrom | datetime |  |
| DateTo | datetime |  |
| Price | decimal (9,2) |  |

**tNumbersTypes (idNumberType, NumberType, Price)** – предназначена для хранения данных типов номеров (таблица 6).

Таблица 6– Типы номеров

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| idNumberType | int | РК |
| NumberType | nvarchar(255) |  |
| Price | decimal (9,2) |  |

**tBookingStatuses (IdBookingStatus, StatusCode, StatusDescr)** –предназначена для хранения статуса бронирования (таблица 5).

Таблица 5– Статус бронирования

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| IdBookingStatus | int | PK |
| StatusCode | varchar(1) |  |
| StatusDescr | nvarchar(255) |  |

**tClients (idClient, Name, SurName, Country, email, passroptNo)** –предназначена для хранения данных о клиентах (таблица 7).

Таблица 7– Клиенты

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| idClient | int | РК |
| Name | nvarchar(255) |  |
| SndName | nvarchar(255) |  |
| SurName | nvarchar(255) |  |
| Country | nvarchar(255) |  |
| email | nvarchar(255) |  |
| passroptNo | nvarchar(255) |  |

**tNumbers (idNumber, idNumberType, RoomNo, Floor)** –предназначена для хранения данных о номерах (таблица 8).

Таблица 8– Номера

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| idNumber | int | РК |
| idNumberType | int | FK |
| RoomNo | nvarchar(255) |  |
| Floor | smalint |  |

**tServices (idService, Name, Descr, Price)** –предназначена для хранения услуг (таблица 9).

Таблица 9– Услуги

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| idService | int | PK |
| Name | nvarchar(255) |  |
| Descr | nvarchar(255) |  |
| Price | decimal(9,2) |  |

**tContacts (IdContact, ContactName, Tel, Email, Adress)** –предназначена для хранения контактов (таблица 10).

Таблица 10– Контакты

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| IdContact | int | PK |
| ContactName | nvarchar(255) |  |
| Tel | nvarchar(255) |  |
| Email | nvarchar(255) |  |
| Adress | nvarchar(255) |  |

**tRoles (IdRole, Name, Descr)** –предназначена для хранения ролей сотрудников (таблица 11).

Таблица 11– Роли

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| IdRole | int | PK |
| Name | nvarchar(255) |  |
| Descr | nvarchar(255) |  |

**tFunction (IdFunction, Code, Name, Descr)** –предназначена для хранения функций (таблица 12).

Таблица 12– Функции

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| IdFunction | int | PK |
| Code | varchar(3) |  |
| Name | nvarchar(255) |  |
| Descr | nvarchar(255) |  |

**tAccessLevel (IdAccessLevel, IdFunction, IdRole, Read, Update, Delete)** –предназначена для хранения уровня доступа (таблица 13).

Таблица 13– Уровень доступа

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| IdAccessLevel | int | PK |
| IdFunction | int | FK |
| IdRole | int | FK |
| Read | varchar(1) |  |
| Update | varchar(1) |  |
| Delete | varchar(1) |  |

**tSeasonCosts (IdSeasonCosts, DateFrom, DateTo, Discount)** –предназначена для хранения сезонного коэффициента (таблица 14).

Таблица 14–Сезонный коэффициент

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Ключ |
| IdSeasonCosts | int | PK |
| DateFrom | datetime |  |
| DateTo | datetime |  |
| Discount | decimal(9,2) |  |

На рисунке 5 представлена диаграмма базы данных информационной системы экскурсионного бюро.

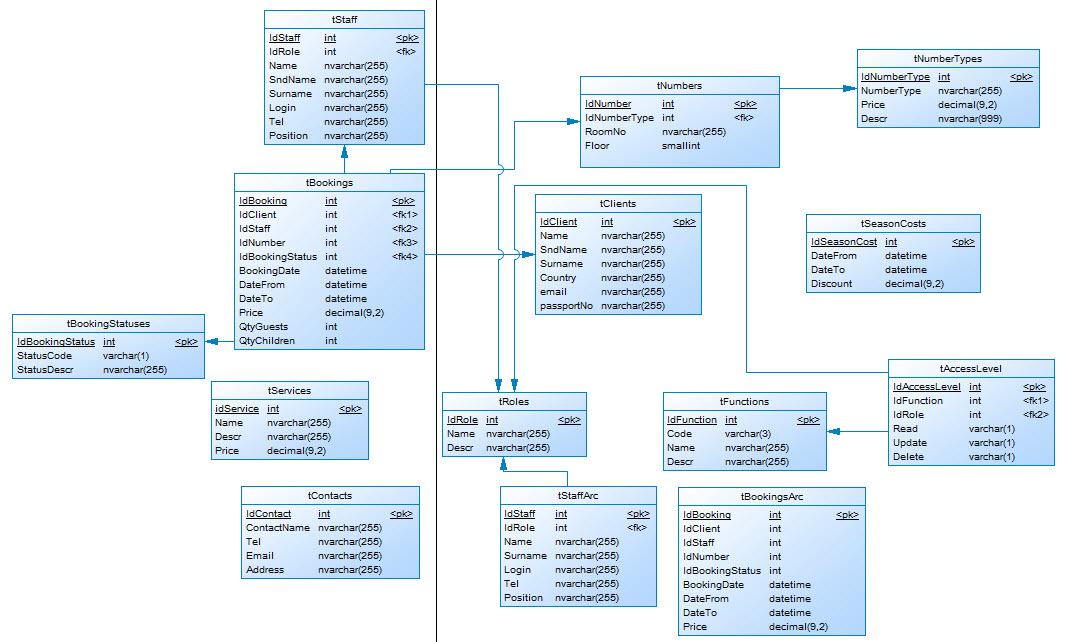


Рисунок 5 – Диаграмма базы данных

## 5. Реализация информационной системы

Для взаимодействия пользователя с системой был разработан графический интерфейс.

Работа программы начинается с окна авторизации пользователя, где пользователь вводит логин и пароль.

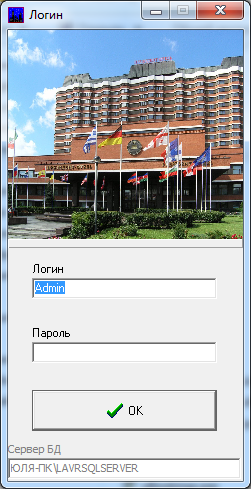


Рисунок 6 – Форма авторизации пользователя.

Код проверки логина и пароля, при помощи кнопки вход и переход в клиентское приложения. Код кнопки “ОК”.

procedure TfrmLogin.btOkKeyPress(Sender: TObject; var Key: Char);

begin

if key = #13 then

btOkClick(sender);

end;

// авторизация логина в БД системы

procedure TfrmLogin.btOkClick(Sender: TObject);

begin

try

DM.Conn.connected:= false; // отключаем соединение с БД

DM.Conn.DefaultDatabase:= 'HotelDB'; // по умолчанию

DM.Conn.ConnectionString:= 'Provider=SQLOLEDB.1;   
Persist Security Info=False; // данные для входа из sql

User ID=' + edUser.text + '; Password=' + edPass.text +'; Initial Catalog=HotelDB; Data Source=' + edServer.text;

DM.Conn.connected:= true; // соединение с БД

frmMain.sBar.Panels[0].Text:= edUser.Text; // статус бар главной формы

frmMain.sBar.Panels[1].Text:= edServer.Text; // сервер бар главной формы

frmMain.show;

frmLogin.Hide;

except

on E:Exception do ShowMessage(E.Message); // вывод текста из ошибки

end;

end;

// считывает пользователя и сервер

procedure TfrmLogin.FormCreate(Sender: TObject);

var

ini : TIniFile;

begin

If not FileExists(ExtractFilePath(Application.ExeName)+ 'Hotel.ini') then Exit;

Ini:= TIniFile.Create(ExtractFilePath(Application.ExeName)+ 'Hotel.ini');

edUser.Text:= Ini.ReadString('User', 'Login', '');

edServer.Text:= Ini.ReadString('Server', 'Host', '');

end;

// запоминает пользователя и сервер

procedure TfrmLogin.FormClose(Sender: TObject; var Action: TCloseAction);

var

ini : TIniFile; // ini файл

begin

Ini:= TIniFile.Create(ExtractFilePath(Application.ExeName)+ 'Hotel.ini');

Ini.WriteString('User', 'Login', edUser.Text);

Ini.WriteString('Server', 'Host', edServer.Text);

end;

После нажатие кнопки “ОК”, при указании правильного логина и пароля, открывается главная форма программы, зависит от доступа. У администратора полный доступ без ограничений на (рисунок 7), а у менеджера (рисунок 8), после чего нам открывается доступ к разным данным программы.

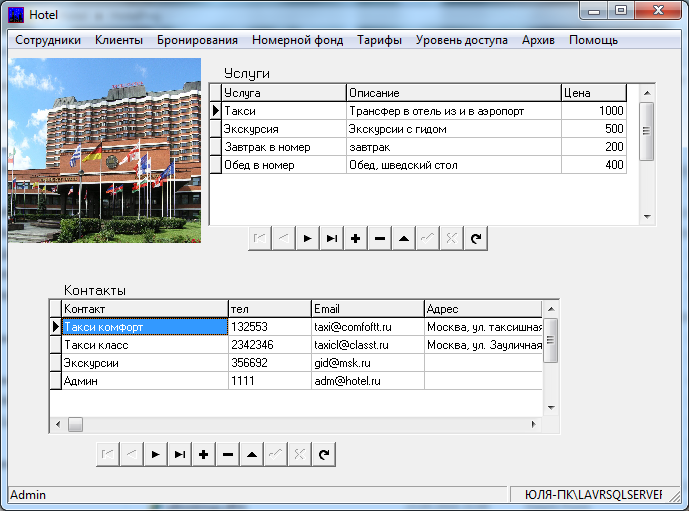


Рисунок 7 – Главная форма администратора

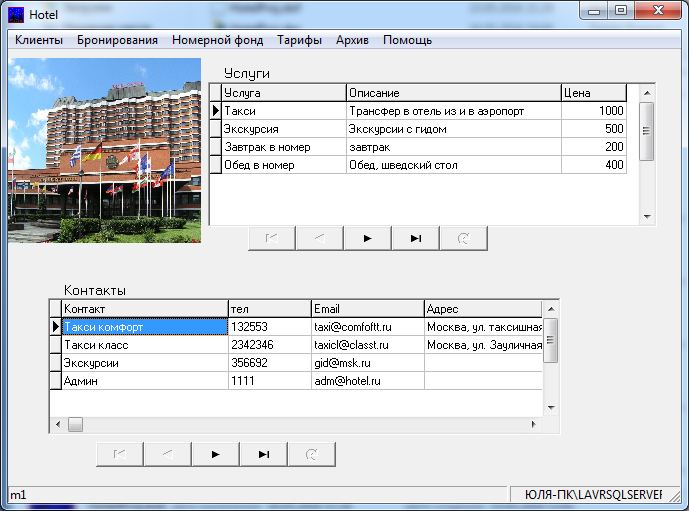


Рисунок 8 – Главная форма менеджера

Код главной формы.

procedure TfrmMain.FormShow(Sender: TObject);

begin

if DM.Conn.Connected = false then

DM.Conn.Connected:= true;

dsContacts.Active:= true;

dsServices.Active:= true;

// проверяем права и отключаем меню, если нужно

if DM.fCheckAccess(fStaff).Read = '0' then begin

Menu.Items.Find('Сотрудники').visible:= false;  
 Menu.Items[7].items[1].Visible:= false;

end;  
 if DM.fCheckAccess(fClients).Read = '0' then

Menu.Items.Find('Клиенты').visible:= false;

if DM.fCheckAccess(fBookings).Read = '0' then

Menu.Items.Find('Бронирования').visible:= false;

if DM.fCheckAccess(fNumbers).Read = '0' then

Menu.Items.Find('База номеров').visible:= false;

if DM.fCheckAccess(fNumberTypes).Read = '0' then begin

Menu.Items.Find('Типы номеров').visible:= false;

Menu.Items.Find('Цены на номера').visible:= false;

end;

if DM.fCheckAccess(fSeasonCosts).Read = '0' then

Menu.Items.Find('Сезонные коэффициенты').visible:= false;

if DM.fCheckAccess(fBookingStatuses).Read = '0' then

Menu.Items.Find('Статусы бронирования').visible:= false;

if DM.fCheckAccess(fRoles).Read = '0' then

Menu.Items[5].items[0].Visible:= false; // .Find('Роли').visible:= false;

if DM.fCheckAccess(fFunctions).Read = '0' then

Menu.Items[5].items[1].Visible:= false; // Menu.Items.Find('Функции системы').visible:= false;

if DM.fCheckAccess(fAccessLevel).Read = '0' then

Menu.Items[5].items[2].Visible:= false; // Menu.Items.Find('Раграничение уровня доступа').visible:= false;

if (DM.fCheckAccess(fRoles).Read = '0') and (DM.fCheckAccess(fFunctions).Read = '0') and (DM.fCheckAccess(fAccessLevel).Read = '0') then

Menu.Items[5].Visible:= false;

if DM.fCheckAccess(fServices).Update = '0' then begin

dsServices.Close;

dsServices.ReadOnly:= true;

dsServices.Active:= true;

grServices.ReadOnly:= true;

DBNavServices.VisibleButtons:= [nbFirst, nbPrior, nbNext, nbLast, nbRefresh];

end;

if DM.fCheckAccess(fContacts).Update = '0' then begin

grContacts.ReadOnly:= true;

dsContacts.Close;

dsContacts.ReadOnly:= true;

dsContacts.Active:= true;

DBNavContacts.VisibleButtons:= [nbFirst, nbPrior, nbNext, nbLast, nbRefresh]; // скрываются кнопки навигатора

end;end;

Рассмотрим первую вкладку "Сотрудники", доступна только администратору, для добавления сотрудника используется форма (рисунок 9) заполняем данными и нажимаем "Ок".

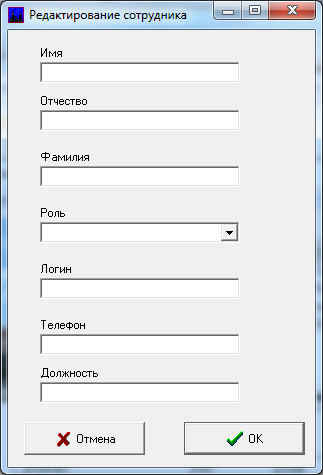


Рисунок 9 – Добавление сотрудника

Просмотр сотрудников так же для удобства отсюда возможно редактирование, добавление и удаление сотрудника. Есть возможность вывести отчет в Ехсеl.

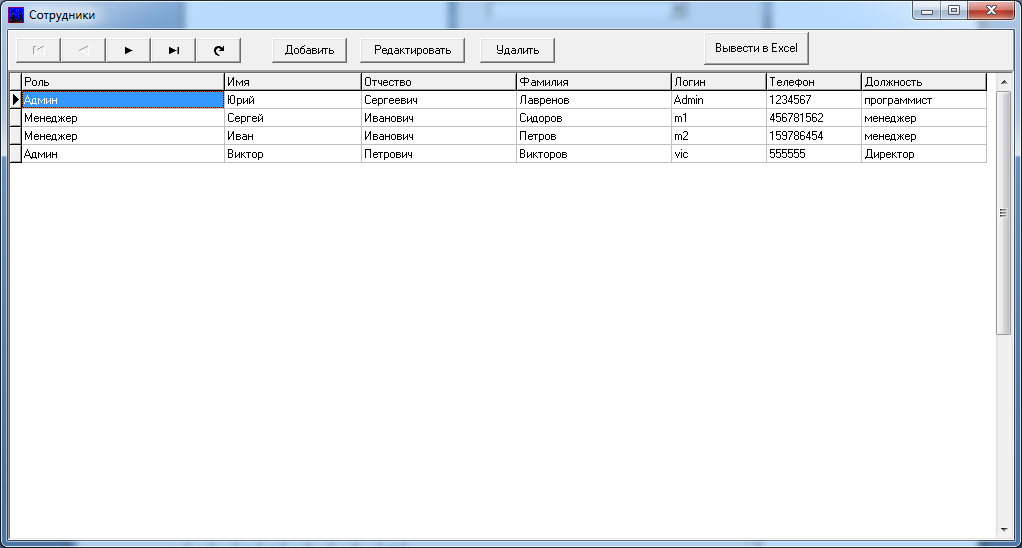


Рисунок 10 – Просмотр сотрудников

Для добавления нового клиента нажимаем на вкладку "Клиенты", открывается форма, добавление и редактирование доступно напрямую сразу в таблице. Для редактирования необходимо два раза щелкнуть в той ячейке которую необходимо изменить, после чего на панель навигатор нажать на галочку. А для добавления на панели выбрать +, после чего добавиться новая строка, в которую можно добавлять данные.

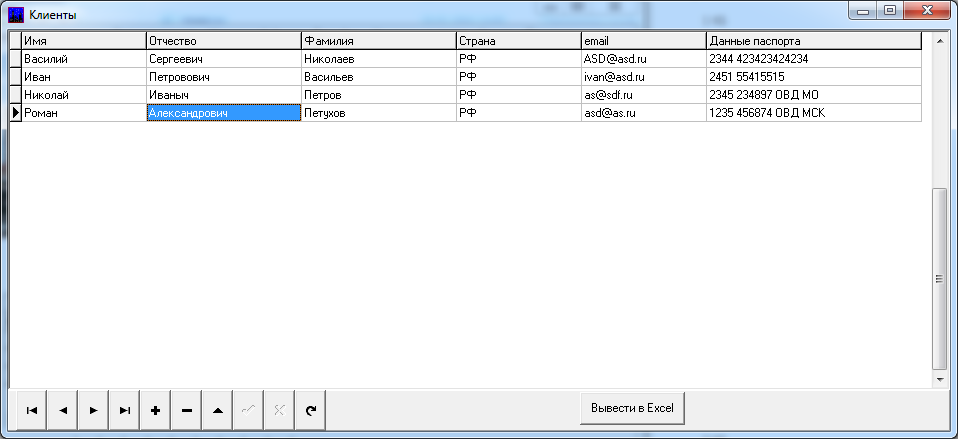


Рисунок 11 – Клиенты

Основная форма для бронирования номеров, и просмотра статуса.

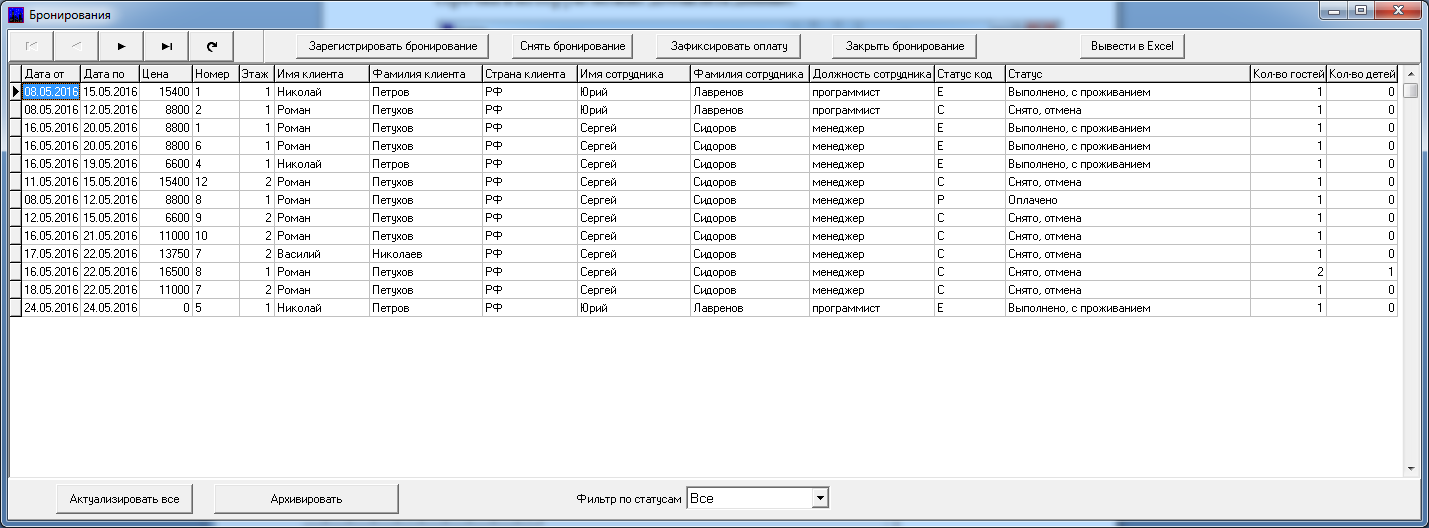


Рисунок 12 – Бронирование

Для регистрирования бронирования используется форма, в которой заполняются данные, выбирается клиент. Прежде, чем зарегистрировать номер, необходимо зарегистрировать клиента. После того, как выбрали клиента, выбираем номер, ставим количество взрослых и количество детей, если есть такое.

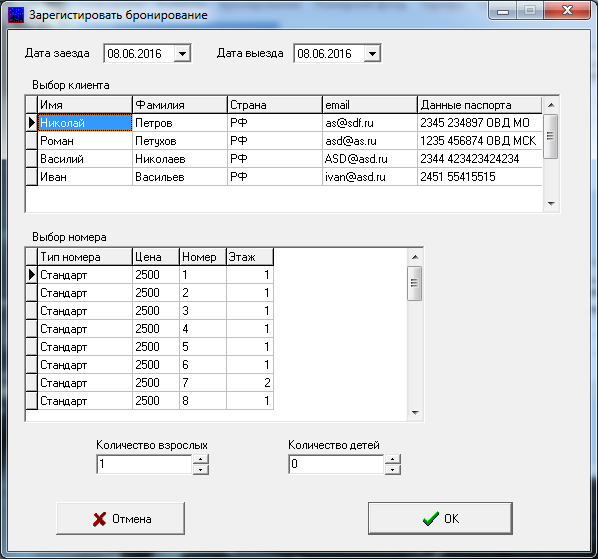


Рисунок 13 – Окно регистрации брони

Код программы для бронирования.

procedure TfrmRegBooking.FormShow(Sender: TObject);

begin

try

DSClients.Active:= true;

finally

end;

updateNumbers; // вызов процедуры

end;

// обновление списка доступных номеров (нет брони в указанные даты)

procedure TfrmRegBooking.updateNumbers;

begin

sp1.Close;

sp1.SQL.Clear;

sp1.SQL.Text:= 'exec dbo.sp\_getFreeNumbers ''' +

formatDateTime('yyyymmdd', dt1.DateTime) + ''', ''' + formatDateTime('yyyymmdd', dt2.DateTime) + '''';

sp1.Open;

sp1.Active:= true;

end;

procedure TfrmRegBooking.dt1Change(Sender: TObject);

begin

updateNumbers;

end;

// отмена бронирования

procedure TfrmRegBooking.btCancelClick(Sender: TObject);

begin

try

DSClients.Close;

sp1.Close;

finally

Close;

end;

end;

procedure TfrmRegBooking.btOkClick(Sender: TObject);

begin

try

sp\_regBooking.Cancel;

sp\_regBooking.Parameters.ParamByName('@IdClient').Value:= DSClients.FieldByName('IdClient').asInteger;

sp\_regBooking.Parameters.ParamByName('@IdNumber').Value:= sp1.FieldByName('IdNumber').asInteger;

sp\_regBooking.Parameters.ParamByName('@DateFrom').Value:= dt1.DateTime;

sp\_regBooking.Parameters.ParamByName('@DateTo').Value:= dt2.DateTime;

sp\_regBooking.Parameters.ParamByName('@QtyGuests').Value:= StrToInt(edQtyGuests.Text);

sp\_regBooking.Parameters.ParamByName('@QtyChildren').Value:= StrToInt(edQtyChildren.Text);

sp\_regBooking.ExecProc;

if frmBookings.DSBookings.active = false then frmBookings.DSBookings.active := true;

frmBookings.DSBookings.requery;

frmBookings.DSBookings.refresh;

finally

close;

end;

end;

procedure TfrmRegBooking.FormCreate(Sender: TObject); begin

dt1.DateTime:= now;

dt2.DateTime:= now;

end;

end.

Для работы с номерами предусмотрена вкладка "Номерной фонд", в котором отображается база номеров и тип номеров. Для удобства было сделано добавление/редактирование, удаление. Так же, здесь есть просмотр статуса номеров, для этого необходимо выбрать дату заезда и дату выезда, и в таблице будет отображаться статус на эти номера в определенный период.

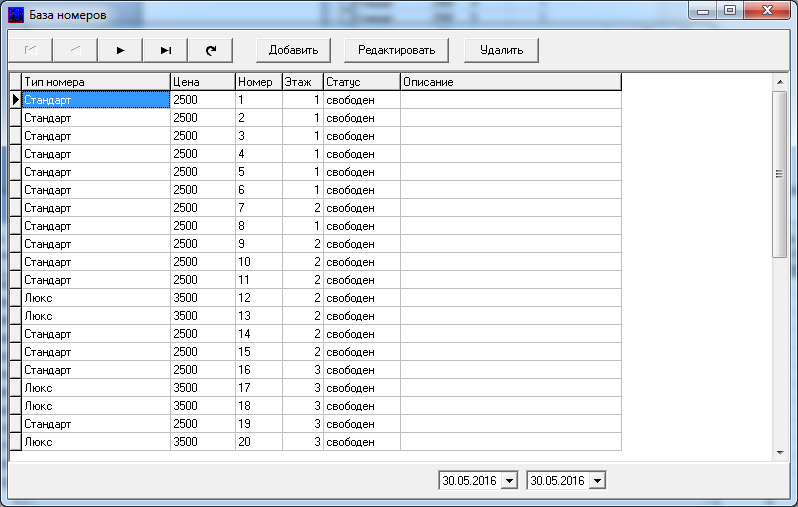


Рисунок 14 – База номеров

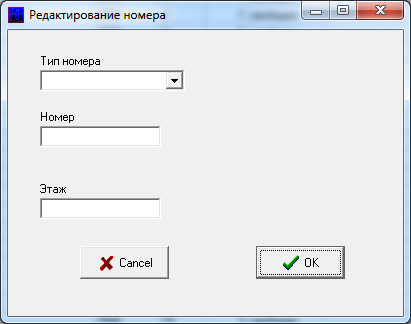


Рисунок 15 – Редактирование номера

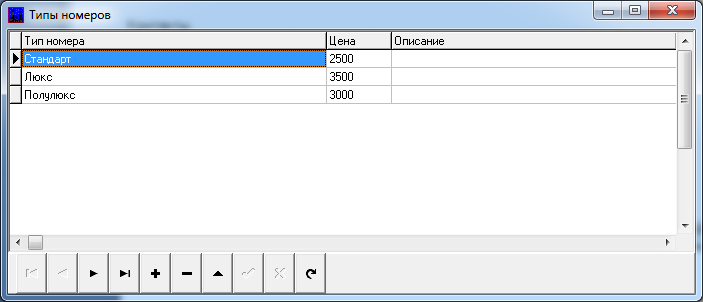


Рисунок 16 – Тип номеров

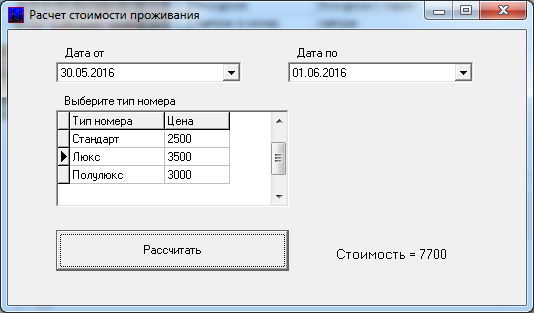


Рисунок 17 – Расчет стоимости проживания

Для ограничения прав менеджера была сделана вкладка "Уровень доступа", доступна только администратору. В ней описывается функция, ее код функции, роль. В данном окне отображается возможности менеджера.

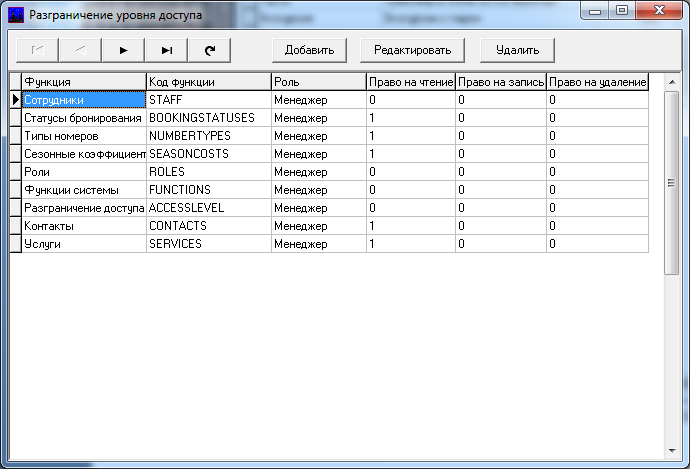


Рисунок 18 – Разграничение уровня доступа

Для просмотра архива есть вкладка "Архив", в ней возможно просмотр сотрудников, доступно администратору, и бронирования.

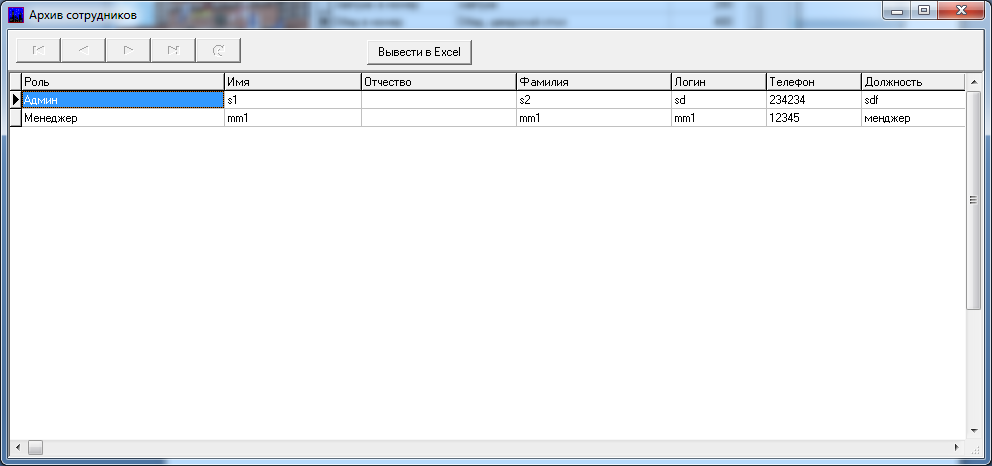


Рисунок 19 – Архив сотрудников

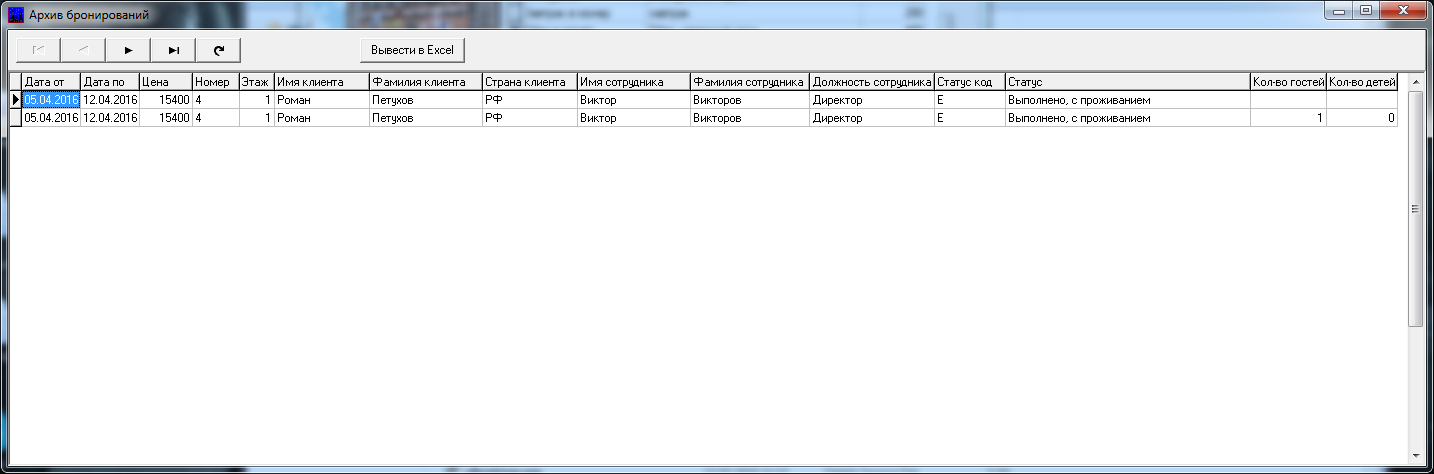


Рисунок 20 – Архив бронирования

Для удобства пользователю сделана справка "Помощь". В которой описана инструкция по работе с приложением.

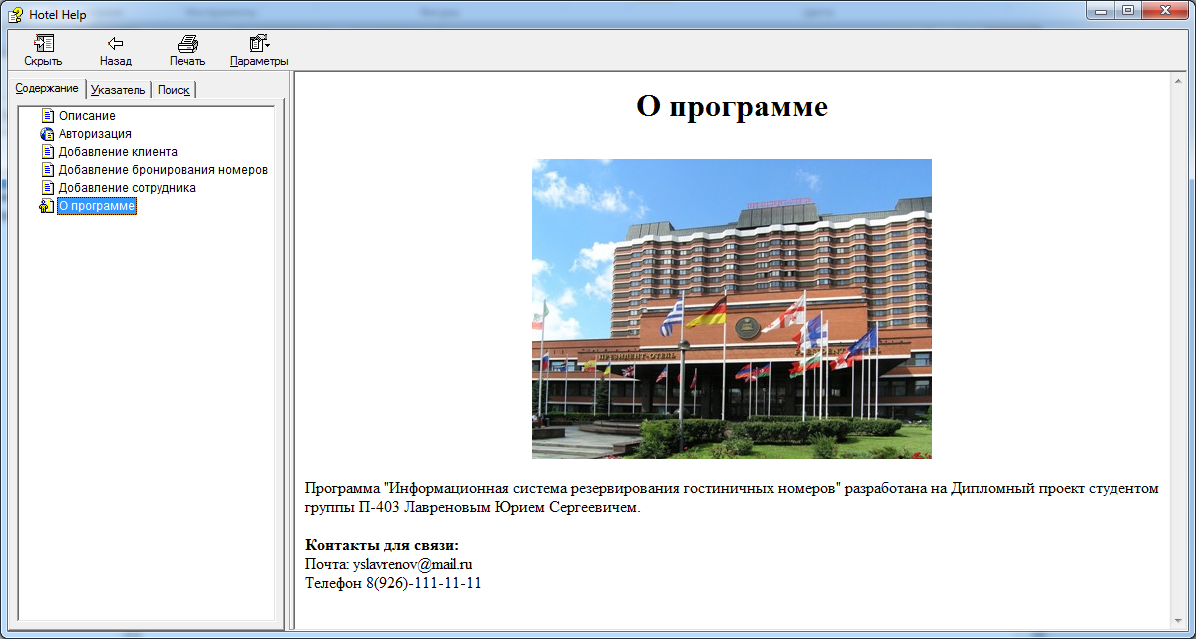


Рисунок 21 – Помощь

## Заключение

В результате курсового проекта была спроектирована и реализована информационная система «Резервирования гостиничных номеров». Данная система удовлетворяет всем требованиям, предъявленным в задании, и реализует большинство необходимых функций. Приложение выполняет следующий перечень требований:

* представление данных в удобном для пользователя виде;
* добавление, удаление и изменение данных;
* просмотр даты и времени, и т.д.

Данная программа разработана и предназначена для автоматизации рабочего места менеджера/администратора гостиницы, обеспечивающая предоставление и хранение полной информации о сотрудниках гостиницы, так же о клиентах и бронировании номеров.

Для того чтобы программа функционировала, необходима операционная система Windows (созданная специально для этой среды). Любой неквалифицированный сотрудник гостиницы должен иметь навыки работы с ПК, и знать основы OC Windows, т.к. все используемые компоненты и формы разработаны при помощи компонентов стандартных Windows, обеспечивающие диалога между пользователем и программой.

Программа включает в себя интуитивно понятный и практичный пользовательский интерфейс, подходящий под OC Windows. Для работы с программой пользователю необходимо владеть элементарными навыками работы с данной оболочкой.

# Литература

1. Описание гостиницы: [Электронный ресурс]// Статья. URL: http://dic.academic.ru/dic.nsf/ruwiki/131260.
2. Описание гостиницы: [Электронный ресурс]// Википедия. URL: https://ru.wikipedia.org/wiki/Гостиница.
3. Описание информационной системы: [Электронный ресурс]// Википедия. URL: https://ru.wikipedia.org/wiki/Информационная\_система.
4. Описание языка программирования: [Электронный ресурс]// Википедия. URL: https://ru.wikipedia.org/wiki/Delphi.
5. Логическая модель предметной области: [Электронный ресурс]// Статья. URL: http://analyst.by/diagrams/logicheskaya-model-predmetnoy-oblasti
6. Стандарт ISO/IEC 2382:2015.