Содержание

[Введение. 2](#_Toc375431951)

[1. Рассмотрение предметной области. 4](#_Toc375431952)

[1.1 Проведение декомпозиции 6](#_Toc375431953)

[1.2 Описание получившихся отношений 8](#_Toc375431954)

[2. Описание процесса реализации 11](#_Toc375431955)

[2.1 Реализация структуры базы данных 11](#_Toc375431956)

[2.2 Реализация запросов 13](#_Toc375431957)

[2.3 Реализация графического интерфейса 17](#_Toc375431958)

[2.4 Описание работы приложения 20](#_Toc375431959)

[Вывод 21](#_Toc375431960)

[Список литературы 22](#_Toc375431961)

# Введение.

Современный мир невозможно представить без информации. Она окружает нас повсюду: на работе, дома, в транспорте, и даже на отдыхе. Как известно, Интернет, или, как мы привыкли его называть, «всемирная паутина» - это всемирная система объединения компьютерных сетей, которая используется для хранения и передачи информации. В настоящее время практически каждый индивид способен получить доступ к этому ресурсу, который, вне всякого сомнения, является неотъемлемой частью нашей жизни. Сейчас не составляет труда найти любую интересующую вас информацию за несколько минут. Бесполезно отрицать полезность Интернета и то, какие удобства мы получаем от пользования им. Но в то же время не стоит забывать о проблемах, с которыми столкнулось человечество с момента появления этого места скопления огромного количества информации (как полезной, так и вредной). Например, сейчас, когда практически у каждого человека есть доступ в Интернет, а у определенной части людей этот доступ всегда находится у них под рукой, не обязательно досконально изучать тот или иной аспект жизни и разбираться во всех его мелочах. Вместо этого достаточно пару раз кликнуть по экрану, чтобы найти нужный сайт и прочитать нужную информацию «в нужное время, в нужном месте». Другой наиболее острой проблемой, связанной с использованием всемирной сети является зависимость, как правило, подрастающего поколения, от социальных сетей и «бесполезного» времяпрепровождения в сети Интернет. И, конечно, как показывает мировая практика, каждая страна старается придумать свои способы решения проблем, которые возникают в данной сфере. Но так или иначе, весь тот объем информации, который существует в мире на данный момент и не перестает расти , должен где-то хранится. Какие же способы хранения информации известны на данный момент? На самом деле, этот список очень большой. Начиная от записной книжки или ежедневника, который, возможно, лежит сейчас на вашем рабочем столе, заканчивая облачными хранилищами, данные в которых хранятся на распределенных в сети серверах. Еще одним способом хранения информации являются базы данных.

База данных – это представленная в объективной форме совокупность самостоятельных материалов(статей, расчетов, нормативных актов, судебных решений и т.д.), систематизированных таким образом, что бы эти материалы могли быть найдены или обработаны с помощью электронной вычислительной машины(ЭВМ). Так же, под термином базы данных понимают совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причём такое собрание данных, которое поддерживает одну или более областей применения.

# 1. Рассмотрение предметной области.

Предварительным этапом создания и разработки базы данных является анализ ее предметной области.

Гостиничный бизнес является популярным направлением бизнеса в наши дни. Для успешного функционирования этого рода деятельности необходимо создание надежной платформы по предоставлению качественных услуг своим клиентам.

В зависимости от класса гостиницы подразделяются на три основных типа: бюджетные гостиницы (хостелы), гостиницы бизнес класса и гостиницы элитного класса. Существуют так же и другие классификации гостиниц: звездная система, мотель, специализированные гостиницы, апарт-отели, экзотические гостиницы и др. Но вне зависимости от типа гостиницы, для успешной работы и получения максимальной прибыли каждый руководитель должен заняться вопросом создания качественной программы информационной поддержки своей гостиницы. Наличие такого программного обеспечения позволит автоматизировать весь производственный процесс.

Для эффективного управления деятельностью гостиницы необходимо реализация программы, которая будет осуществлять ведение списка постояльцев, учет списка забронированных мест и ведение архива всех постояльцев.

В перспективе, в данную программу могут включатся инновационные способы управления персоналом, которые будут полезны не только для учета клиентской базы, а и для успешного функционирования внутренней структуры предприятия. В их числе: внедрение учета заработной платы сотрудников, персонализация программного обеспечения (личный кабинет для каждого сотрудника), составление графика работы сотрудников (приход/уход с рабочего места) и др.

Все эти методы обеспечат автоматизацию деятельности гостиницы.

В рамках учебного процесса будет создана программа, которая будет осуществлять следующие функции:

* получение списка свободных номеров (по количеству мест и классу);
* получение списка номеров (мест), освобождающихся сегодня и завтра;
* выдачу информации по конкретному номеру;
* автоматизацию выдачи счетов на оплату номера и услуг;
* получение списка забронированных номеров;
* проверку наличия брони по имени клиента и/или названию организации.

## 1.1 Проведение декомпозиции

Для начала необходимо представить все необходимые данные, как сложную систему, которая будет состоять из отдельно связанных между собой подсистем и процессов взаимодействия (далее «супер-таблица»). Как правило, в супер-таблице должны быть отображены все данные, которые будут храниться в будущей базе данных.

Следует отметить, что этап декомпозиции является наиболее важным при разработке базы данных, так как выделенные на этом этапе составляющие части супер-таблицы станут базисом нашей будущей базы данных, а взаимосвязи таких таблиц - ее структурой. Настоятельно рекомендуется в дальнейшем не изменять таблицы, полученные на этапе декомпозиции, так как это может привести к сбою в работе всей базы данных.

В ходе работы мне удалось выделить 4 таблицы, каждая из которых имеет собственные элементы:

1. **Guest**

* Guest\_ID
* Name
* Surname
* bDay
* Sex
* Phone

1. **GuestVisiying**

* GuestVisiting\_ID
* Guest\_ID
* Rooms\_ID
* Race
* Exit

1. **Rooms**

* Rooms\_ID
* RoomType\_ID

1. **RoomType**

* RoomType\_ID
* TypeName
* Places
* NumOfRooms
* Price
* Status

Процесс декомпозиции завершен.

## 1.2 Описание получившихся отношений

В данной работе мы будем использовать реляционные базы данных. Такие базы данных характеризуются наличием в них элементов, которые связаны между собой определенным образом, а именно, по средствам отношений. На самом деле, одним из главных преимуществ реляционных баз данных является, то, что в таких базах процесс нормализации используется для устранения недостатков ее структуры, приводящая к избыточности данных, которая, в свою очередь, приводит к аномалиям и нарушению целостности данных.

Для реализации нашей базы данных необходимо использовать отношения «внешний ключ к первичному ключу». Данные отношения используются для установления связи между таблицами. Существуют так же и другие виды отношений между таблицами, но для реализации базы данных «Гостиница» будет достаточно данных отношений.

После того, как определены типы отношений, можно приступать непосредственно к налаживанию связей между таблицами.

Важно так же отменить, что взаимосвязь между полями таблиц возможна только при наличии одинаковых имен и типов хранящихся данных в этих полях. В случае несоблюдения этого правила, связь осуществляться не будет.

#### 1.2.1 Выбор инструмента для реализации

На данный момент существует огромное количество инструментов для реализации базы данных. В их число входят реляционная СУБД корпорации Microsoft- Microsoft Acces, разработка компании Sun Systems/Oracle – Workbench, разработка компании PremiumSoft CyberTech Ltd- Navicat, PHPMyAdmin и другие.

Все эти инструменты имеют свои преимущества и недостатки. Для реализации базы данных «Гостиница» мною была выбрана платформа Platypus.

Platypus Platform — программная платформа для построения приложений различного уровня сложности. Основная область применения — создание информационных систем уровня предприятия, однако приложения на базе платформы могут применяться и в других областях, например, государственном управлении.

#### 1.2.2 Описание инструмента реализации

Наиболее важной особенностью Platypus Platform для меня стало наличие визуальных средств разработки, которые используются при создании приложений:

• визуальный редактор структуры базы данных (БД) приложения;

• визуальный редактор запросов к базе данных;

• визуальный редактор форм;

Так же, Platypus Platform характеризуется возможностью повторного использования SQL запросов, отладкой JavaScript кода как на локальном компьютере, так и на удаленном сервере, выполнением SQL запросов в Platypus Application Designer для их отладки, библиотеками виджетов, работающих непосредственно с моделью данных, которая позволяет легко реализовывать достаточно сложные элементы пользовательского интерфейса.

Вне всякого сомнения, все это делает Platypus Platform мощным инструментом для реализации баз данных.

Именно поэтому, я выбрал Platypus Platform для создания учебного приложения базы данных «Гостиница».

# 2. Описание процесса реализации

## 2.1 Реализация структуры базы данных

После того, как подготовительный этап завершен, можно приступать к реализации структуры базы данных. Для этого необходимо создать новый проект в Platypus Platform и построить JDBC Url. После этого необходимо создать новый элемент приложения: диаграмма структуры базы данных. Затем, необходимо добавить в рабочую область выявленные на предварительном этапе таблицы и установить связи между полями этих таблиц там, где это необходимо.

В итоге должна получиться следующая диаграмма:

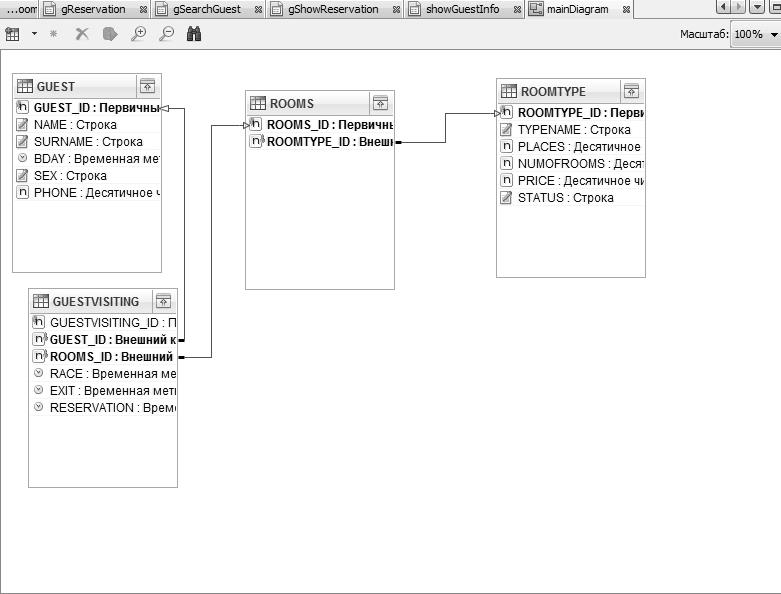


Рис. 1 Диаграмма структуры базы данных.

На данной диаграмме все таблицы, с помощью которых будет осуществляться дальнейшая работа по созданию приложения информационной поддержки гостиницы.

Как отмечалось ранее, необходимым условием для налаживания связей между полями таблиц является наличие одинакового типа и имени полей.

Уже на данном этапе Platypus Platform предоставляет разработчику возможность вносить, изменять и сохранять данные в таблицы.

## 2.2 Реализация запросов

Для успешной работы нашей программы необходимо создать следующие запросы:

* получение списка всех комнат
* получение списка постояльцев, выбывших за последний год
* получение всех данных о постояльцах
* получение списка свободных номеров
* получение списка постояльцев, забронировавших номера

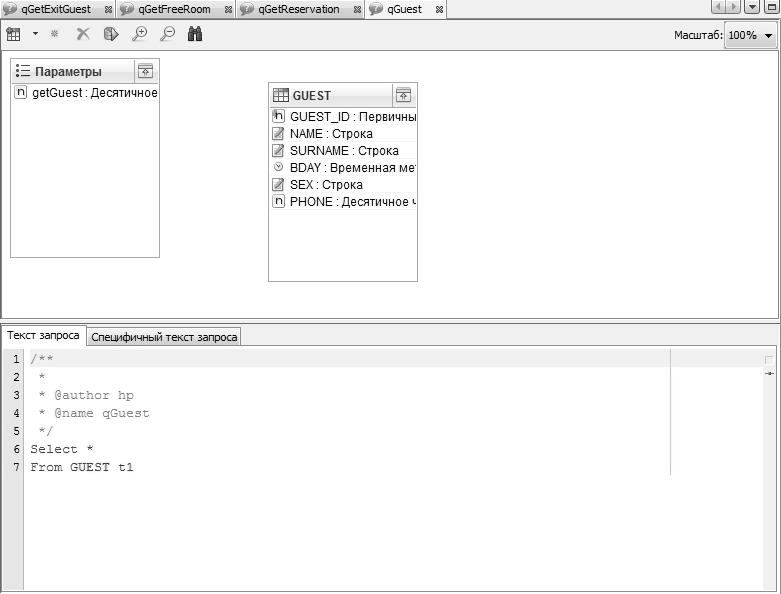
Важно отметить, что каждый запрос создается как новый элемент приложения в виде запроса. Для получения информации о постояльцах необходимо создать новый запрос и добавить в него таблицу Guest (Рис. 2 ) 

Рис. 2 Запрос на получение информации о гостях

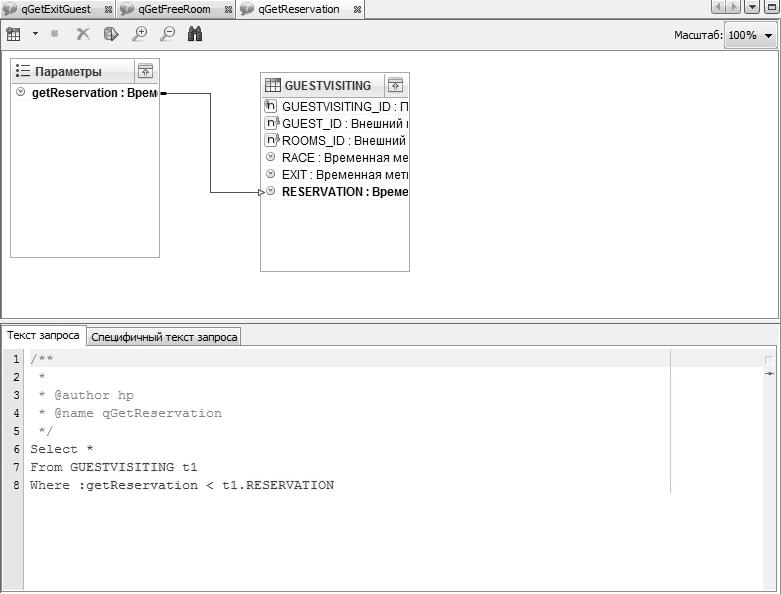
Следующий запрос будет выводить нам список забронированных мест. Для этого создаем новый запрос и добавляем в него таблицу GuestVisiting. Далее необходимо создать новый параметр getReservation типа временная метка, в который мы будем передавать текущую дату. Результатом этого запроса будет являться список людей, которые забронировали номер на определенную дату. (Рис. 3)

Рис. 3 Получение списка забронированных номеров

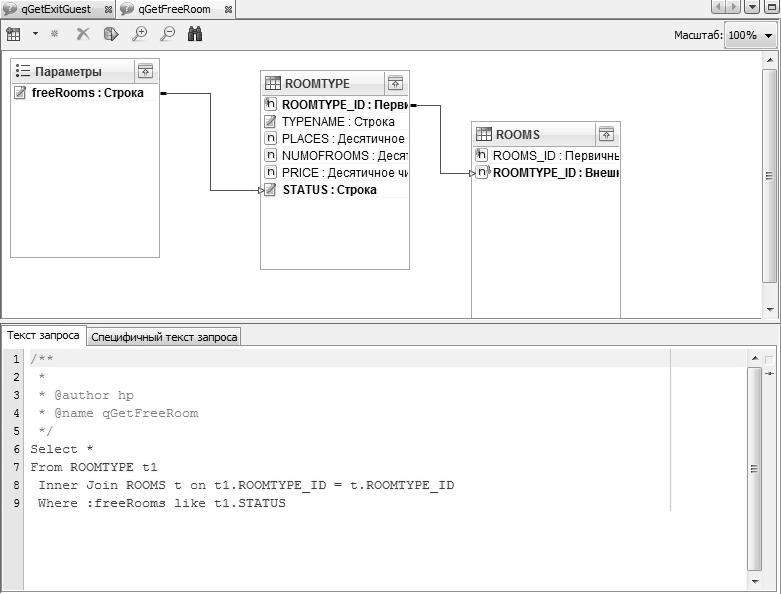
Для получения списка свободных номеров необходимо создать новый запрос и добавить в него таблицы Rooms и RoomType. Следует отметить, что в запросах Platypus Platform автоматически восстановит связи между таблицами, если такие связи имеются. Далее создаем параметр типа строка, который связываем с полем Status, и в который мы будем передавать значение «СВОБОДНО» для того, чтобы получить список свободных номеров (Рис. 4).

Рис. 4 Получение списка свободных/занятых комнат

Далее, создадим запрос, который будет выдавать нам список гостей, которые выбыли за последний год. Для этого в новый запрос добавляем таблицы Guest и GuestVisiting и создаем параметр типа временная метка, в который будет передаваться текущая дата. При том, необходимо наложить условие на параметр, чтобы он был больше значение поля Race из таблицы GuestVisiting. Результатом запроса являться список гостей, которые выбыли из гостиницы за последний год (Рис. 5).

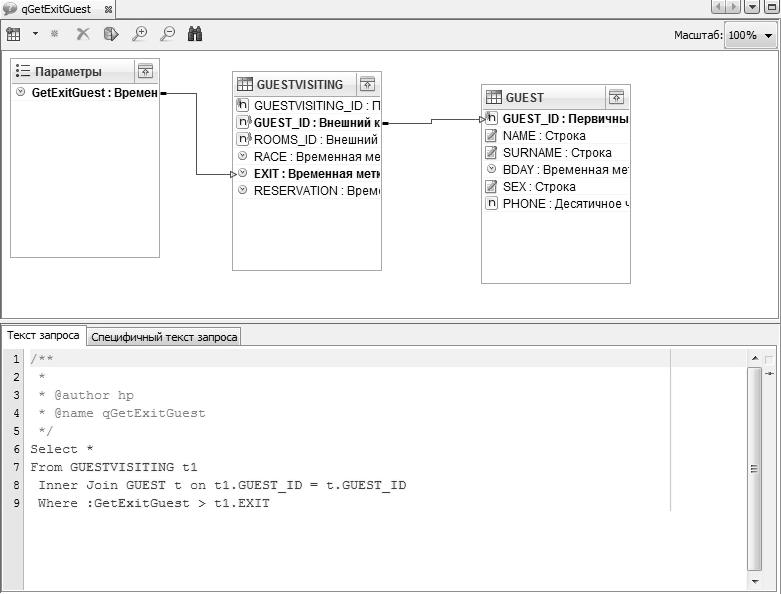


Рис. 5 Получение списка выбывших гостей

Реализация запросов завершена.

## 2.3 Реализация графического интерфейса

Заключительным этапом создания учебного приложения базы данных «Гостиница» является создание графического интерфейса. Следует отметить, что Platypus Platform дает возможность создавать графический интерфейс программы по средствам библиотеки Swing, что делает этот процесс очень удобным. Так же, данная библиотека предоставляет большой набор инструментов для создания удобного и, что самое главное, наглядного пользовательского интерфейса.

Для создания графического интерфейса в Platypus Platform используются представления Platypus или так называемые формы, которые так же как запросы и диаграмма структуры базы данных являются элементами приложения.

Для того что бы вывести на экран тот или иной запрос, его нужно добывать в модель данных и привязать к нему параметр. Затем необходимо создать новую сетку данных и привязать ее к запросу из модели данных. Далее нужно заполнить сетку данных нажав ПКМ- заполнить столбцы (Рис. 6)

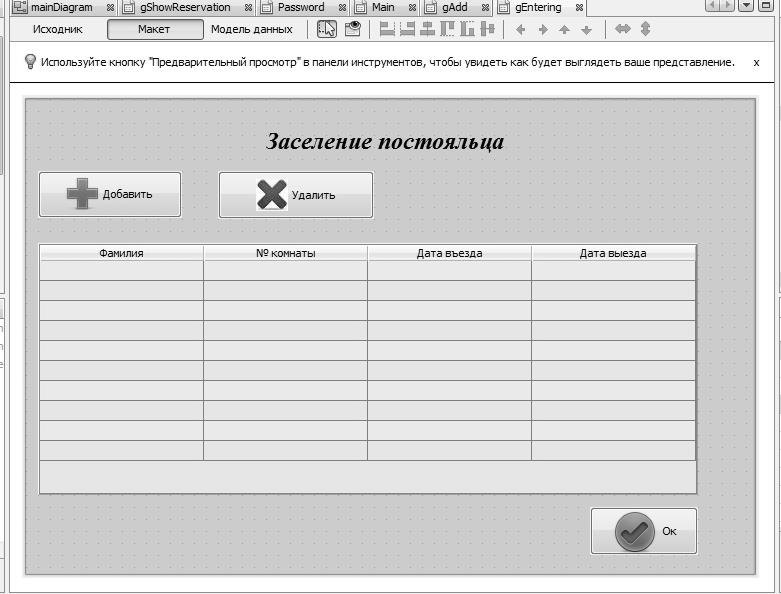


Рис. 6 Заселение гостей

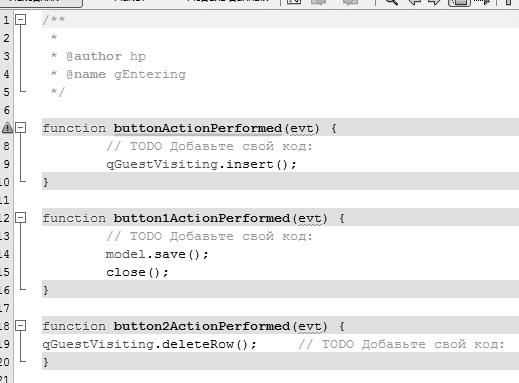
Потом необходимо создать новые кнопки и написать для них обработчик (Рис. 7).

Рис. 7 Исходник для «Заселение гостей»

Данный процесс позволяет нам добавлять, изменять, удалять и сохранять данные в таблицах.

Таким же образом можно реализовать формы с другими запросами, где необходимо добавить, изменить или удалить данные их таблиц.

В случае, когда нам необходимо получить какие-либо данные, нам нужно воспользоваться такими связанными элементами управления, как текстовое поле или Дата-Время. Например, чтобы получить результат запроса, который выдает пользователю список забронированных мест, необходимо сделать следующие действия:

Во-первых, так же как и в случае с заселением гостей, добавляем сетку данных и привязываем к ней запрос на получение забронированных мест.

Во-вторых, необходимо создать элемент Дата-Время и привязать к нему параметр, который возвращает пользователю список забронированных мест.

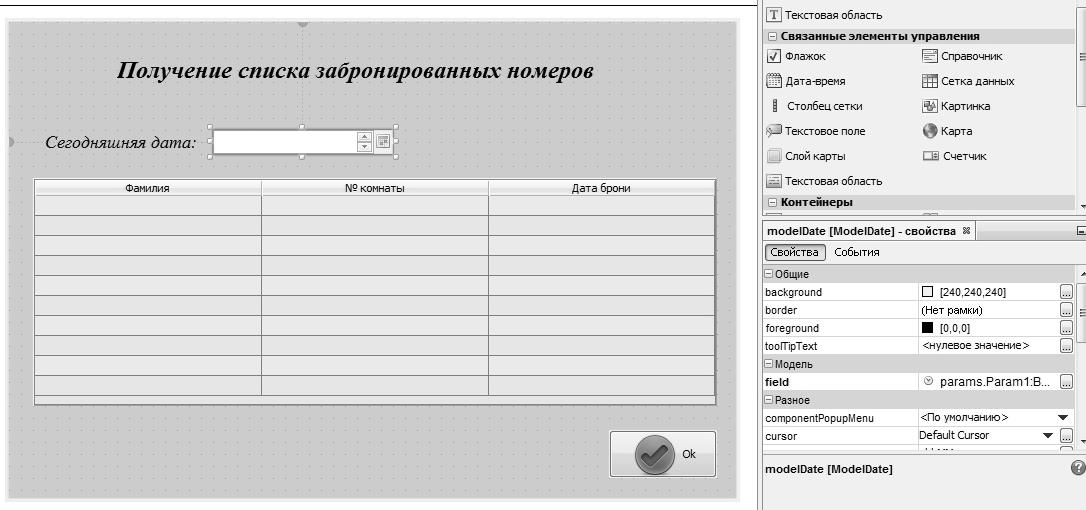


Рис. 8 Графика "Получение забронированных мест"

Обработчик необходимо добавить только для кнопки «Ок», которая будет закрывать окно, так как после того, как пользователь введет данные в поле Время-дата, программа автоматически выполнит запрос и выведет его результат на экран.

Описанных выше методов создания графического представления достаточно для того, что бы привязать все запросы к формам и получить необходимую информацию из таблиц.

## 2.4 Описание работы приложения

Первое окно, которое видит пользователь после запуска моем программы - это окно авторизации, в котором он должен ввести имя пользователя и пароль для того, чтобы войти в программу информационной поддержки отеля. После этого перед пользователем появляется окно, где он может добавить пользователя, посмотреть информацию о номерах или постояльцах.

- Если пользователь захочет добавить постояльца, то перед ним появится окно, в котором он сможет добавить нового постояльца, а затем заселить его в номер или забронировать ему место в отеле.

- Если пользователь захочет получить информацию о гостях отеля, то перед ним появится окно в котором он сможет посмотреть список постояльцев с подробной информацией о них, или же увидеть список, тех гостей, которые покинули отель за последний год.

- Если же пользователь захочет посмотреть информацию о номерах, то перед ним появится окно, в котором он сможет проверить список свободных или занятых номеров, или же увидеть список забронированных мест.

С исходным кодом моего приложения можно ознакомиться, перейдя по ссылке: <https://github.com/kalyagin/Kalyagin-s-Repository/tree/Kalyagin>

# Вывод

В заключении, хотелось бы еще раз подчеркнуть, что базы данных играют огромную роль в жизни современного человека: они позволяют не только хранить информацию, но и создают все необходимые условия для ее управления.

Базы данных используются повсеместно. И, наверное, трудно сейчас найти такую сферу деятельности, где-бы не могли использоваться базы данных. На мой взгляд, это объясняется лишь тем, что базы данных очень сильно упрощают процесс управлением информацией.

Для разработки приложения информационной поддержки отеля я использовал программу Platypus Platform. Это был мой первый опыт работы в данной сфере, поэтому в процессе написания приложения возникали некоторые трудности, связанные с синтаксисом JavaScript и некоторых нюансов работы программы Platypus Platform. Но я считаю, что я выбрал наиболее удобный инструмент для написания своего собственного приложения.

В ходе работы я научился проводить декомпозицию данных, создавать структуру базы данных и выполнять простые и сложные запросы SQL. Так же, в процессе работы мне частично пришлось освоить [прототипно-ориентированный](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [сценарный](http://ru.wikipedia.org/wiki/%D0%A1%D1%86%D0%B5%D0%BD%D0%B0%D1%80%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) [язык программирования](http://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) JavaScript.

Я уверен, что все приобретенные в процессе написания программы навыки понадобятся мне в будущем.

# Список литературы

1. Руководство по быстрому старту

Разработка приложений на платформе Platypus

1. Электронный ресурс - [http://ru.wikipedia.org](http://ru.wikipedia.org/)
2. Руководство по разработке приложений платформы Platypus