Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б.С. Галущака»

ТУРИСТИЧЕСКАЯ ФИРМА «BIGTRAVEL»

Отчёт по учебной практике

ПМ.01 Разработка, администрирование и защита баз данных

МДК11.01 Разработка, администрирование и защита баз данных

МДК11.02 Создание приложений в 1С Предприятие

Разработал:

Студент группы

Косолапова Е.Ю.

2021

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 4
2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 6
3. МОДЕЛИРОВАНИЕ ПРЕЦЕДЕНТОВ 17
4. СПЕЦИФИКАЦИЯ ПРЕЦЕДЕНТОВ 18
5. ДИАГРММА КЛАССОВ АНАЛИЗА 21
6. ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ 22
7. ДИАГРАММА ВЗАИМОДЕЙСТВИЯ 23
8. МОДЕЛИРОВАНИЕ ДАННЫХ 24

9 РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ БАЗЫ ДАННЫХ 30

10 РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ БАЗЫ ДАННЫХ 32

ЗАКЛЮЧЕНИЕ 33

СПИСОК ИСТОЧНИКОВ 34

ПРИЛОЖЕНИЕ А. ER-диаграмма 35

ПРИЛОЖЕНИЕ Б. Диаграмма последовательности 36

ПРИЛОЖЕНИЕ В. Диаграмма видов деятельности 37

ПРИЛОЖЕНИЕ Г. Диаграмма прецедентов 38

ПРИЛОЖЕНИЕ Д. Диаграмма состояний 39

ВВЕДЕНИЕ

Влияние информационных технологий на управление туристической компанией огромно. Ни во многих других областях бизнес настолько не привязан к информации. Ведь, по большей степени, туристическая фирма продает информацию, а турист ее покупает. Таким образом, автоматизация напрямую влияет на повышение эффективности работы, как отдельных менеджеров, так и всей компании в целом. Автоматизация и широкое применение электронной техники становятся одной из актуальнейших задач в отрасли туризма.

Под туристской деятельностью понимается деятельность туристской организации по формированию (производству), продвижению и продаже туристского продукта, оказание услуг по туристскому обслуживанию, а также иная деятельность в сфере туризма.

Актуальность работы – разработка автоматизированной системы работы с клиентами турфирмы, что предопределит успешную реализацию качественного туристского продукта потребителю.

Целью данной учебной практики является закрепление и углубление, полученных знаний во время обучения, приобретение необходимых умений, навыков и опыта в работе с базой данных в программе Microsoft SQL Server и разработке приложения в Visual Studio.

1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Туристическая фирма «BigTravel» — программный продукт, позволяющий создавать, хранить и управлять информацией о клиентах, маршрутах и путевках предоставляемые фирмой. Продукт состоит из веб-приложения, которое разделено на административную часть и клиентскую часть. Административная часть, предоставляет возможность управлять базой данных, то есть создавать, редактировать, удалять, просматривать сведения о путевках, клиентах, транспорте и отелях. Клиентская часть, реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него.

Функциональным назначением программы является автоматизация ввода и обработки информации о списке туров, сведения о клиентах, отелях, маршрутов и транспорта. Разрабатываемый продукт позволит пользователям систематизировать и оптимизировать поиск и оформление заказа конкретного тура.

Данная программа может поддерживать несколько групп пользователей, каждая группа пользователей содержит в себе функциональные требования.

Функциональные требования для пользователя группы «Сотрудник»:

* создание новых туров и добавление в БД;
* добавление в сформированную БД данных о отеле;
* добавление в БД данных о транспорте;
* просмотр данных о отеле;
* просмотр данных о транспорте;
* просмотр информации о туре;
* изменение данных о туре;
* изменение информации о отеле;
* изменение данных о клиентах;
* изменение данных о транспорте;
* удаление данных об отеле;
* удаление информации о туре;
* создание отчета по бронированию в PDF формате;
* отправка сообщения на почту с прикрепленном PDF-файлом.

Функциональные требования для пользователя группы «Клиент»:

* просмотр туров;
* поиск тура по определенным параметрам;
* создание заказа;
* просмотр данных о заказе.

Программный продукт сократит время, затрачиваемое на подбор и поиск тура непосредственно в туристической фирме.

2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Проектирование информационной системы, всегда начинается с определения цели проекта как решение ряда взаимосвязанных задач, включающих в себя обеспечение на момент запуска системы и в течение всего времени ее эксплуатации.

После определения цели был спроектирован прототип и макет приложения в онлайн-сервисе Figma. Figma­ - онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени. Также при проектировании макета было решено какие основные цвета и шрифт будет использовать программа. Основные цвета, используемые в разработке: #d7e6eb, #318bc9 и #ffffff. Шрифт используется только один – Montserrat.

Макеты были сделаны для удобства и понятного использования клиентами и сотрудниками туристической фирмы. Ниже более подробно описана каждая форма со скриншотами.

При открытии приложения пользователь видит главную форму, которая необходима для выбора пользователя, по середине которой есть текстовый логотип туристической фирмы. Ниже представлен макет этой формы (Рисунок 1).

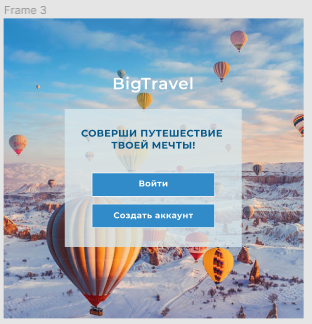


Рисунок 1–Макет главной формы

Если у пользователя отсутствует аккаунт, то он переходит на форму регистрации с помощью кнопки «Создать аккаунт». Макет формы «Регистрации» предоставлен на рисунке 2.

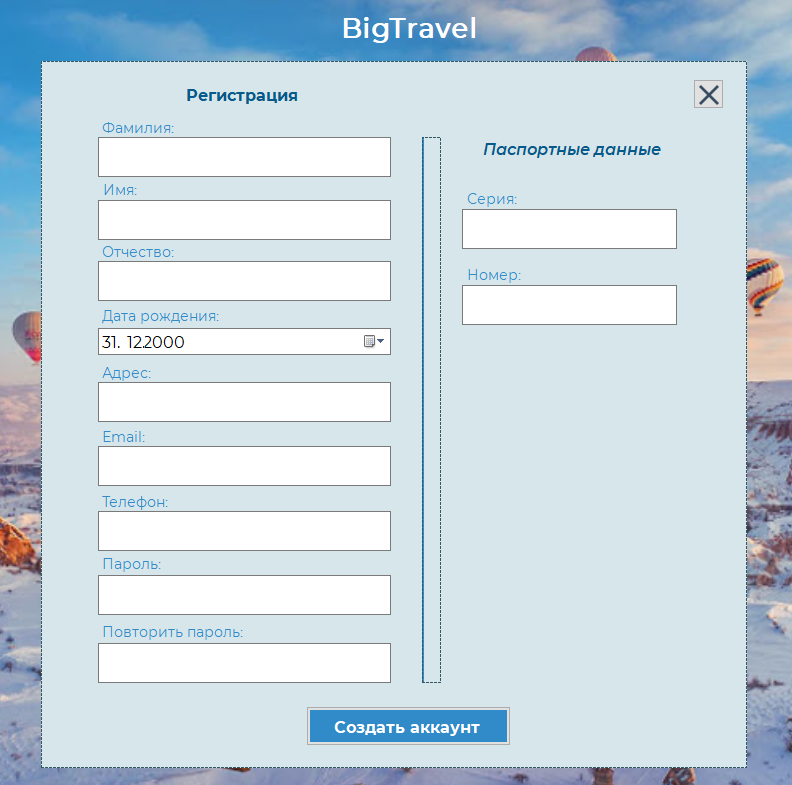


Рисунок 2 –Макет формы «Регистрации»

Макет формы входа показан на рисунке 3.

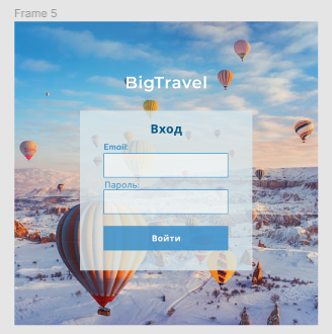


Рисунок 3–Макет формы «Входа»

В зависимости от того, кто входит в систему будет осуществляться переход на форму: «Администратор», «Личный кабинет сотрудника» и «Личный кабинет клиента».

На форме «Личный кабинет клиента» отображается ФИО, телефон, дата рождения, адрес, и электронная почта клиента, а также есть кнопки: «Просмотр туров», «Выход» и «Редактировать». Данная форма предоставлена ниже (Рисунок 4).

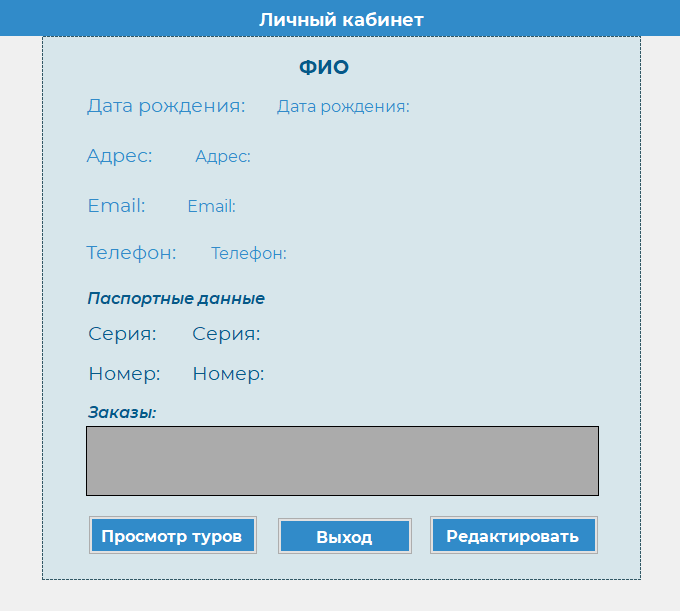


Рисунок 4 –Макет формы «Личный кабинет клиента»

При клике на кнопку «Редактировать» клиент перебрасывает на другую форму, где ему необходимо изменить информацию о себе, на форме также есть кнопки стрелки, что означает действие возвращения в личный кабинет клиента и «Редактировать». Ниже предоставлен макет формы (Рисунок 5).

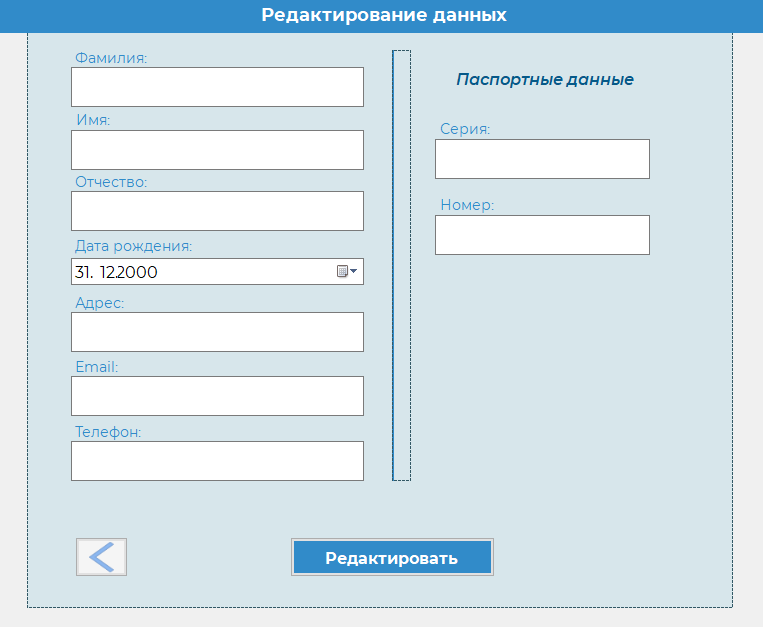


Рисунок 5 –Макет формы «Редактирования»

При возвращении в личный кабинет, клиент, нажимая на кнопку «Просмотр тура» переходит на форму, где может просмотреть туры, которые предлагает турфирма, клиент видит краткую информацию. Макеты форм предоставлены ниже (Рисунок 6-8).

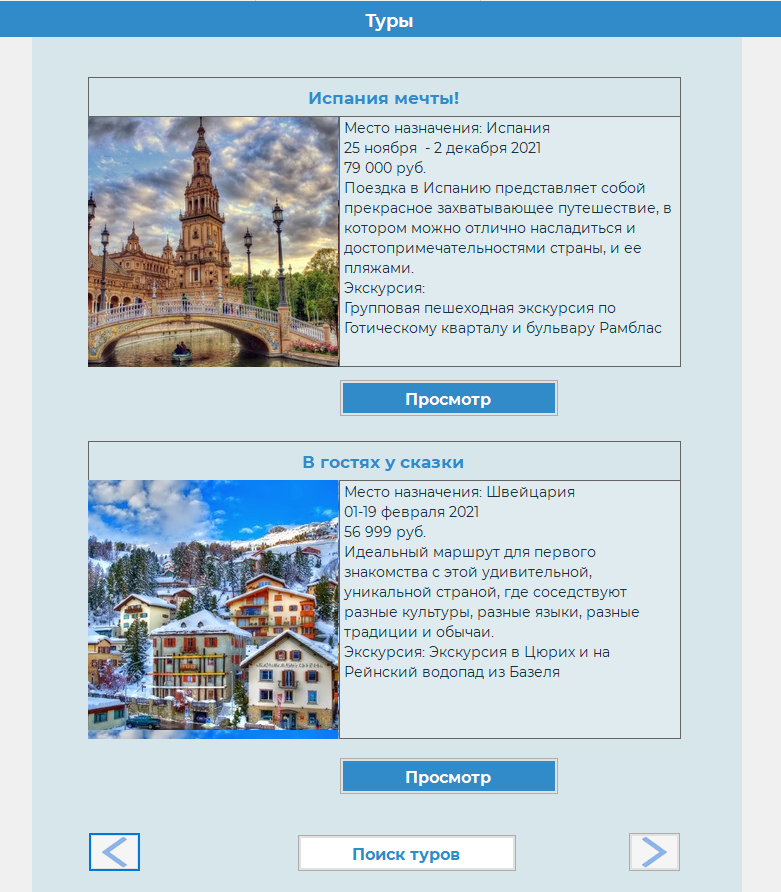
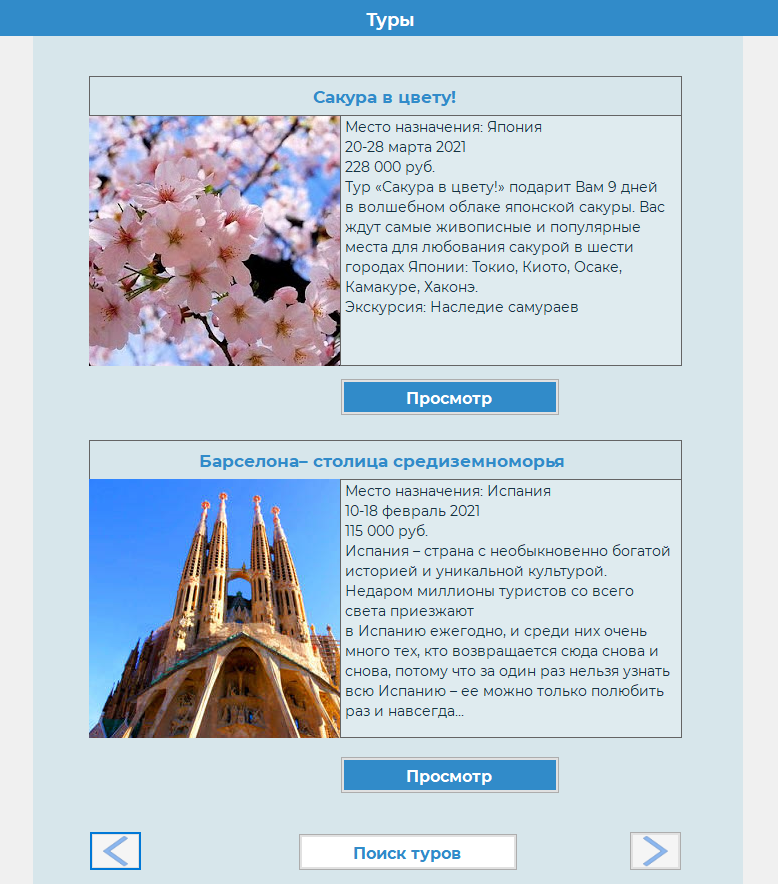


Рисунок 6 – Макеты первой и второй форм «Просмотр тура»

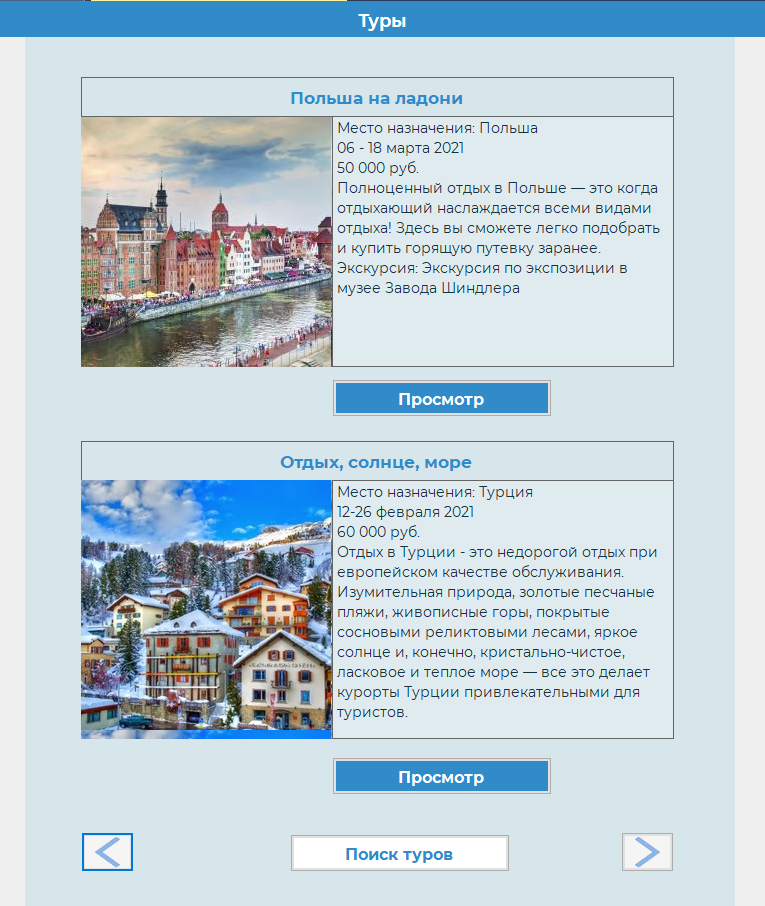


Рисунок 7 – Макет третьей формы «Просмотр тура»

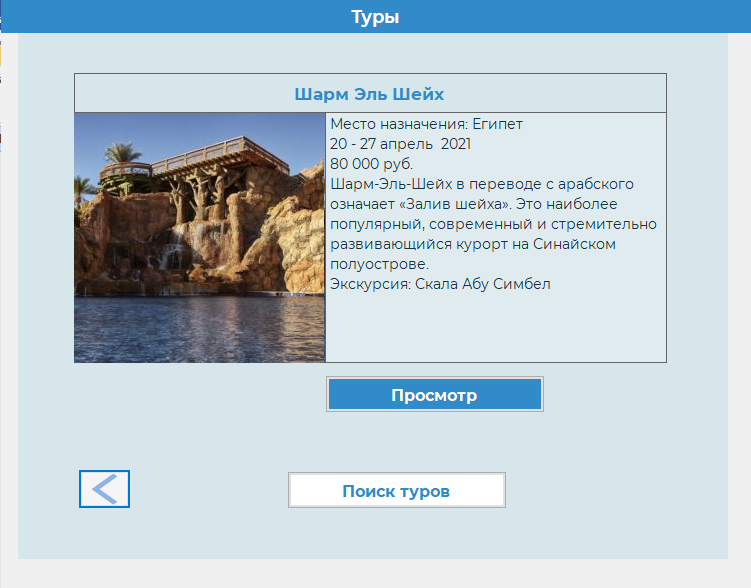


Рисунок 8 –Макет четвертой формы «Просмотр тура»

На всех четырех формах мы видим рядом с каждым туром кнопку «Просмотр», с помощью которой мы можем просмотреть больше информации о выбранном туре. Ниже предоставлен макет формы просмотра определенного тура (Рисунок 9).

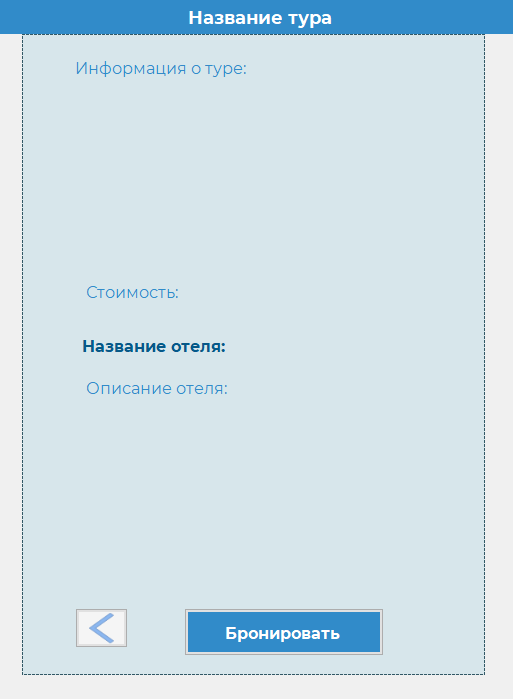


Рисунок 9 – Макет формы «Просмотр определенного тура»

На данной форме есть кнопка, «Бронировать» которая переносит на форму «Бронирования» (Рисунок 10), где необходимо ввести определенные данные и при нажатии на кнопку будет создаваться договор, который отправляется на почту клиенту.

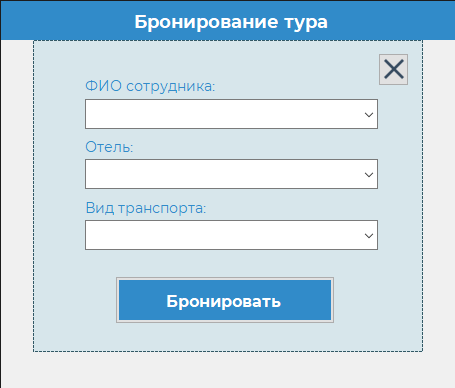


Рисунок 10– Макет формы «Бронирования»

При возвращении на форму «Просмотр тура», также можно заметить кнопку «Поиск тура», при нажатии на нее осуществляется переход на форму где необходимо ввести данные и нажать на кнопку «Поиск тура». Ниже представлен макет формы (Рисунок 11).

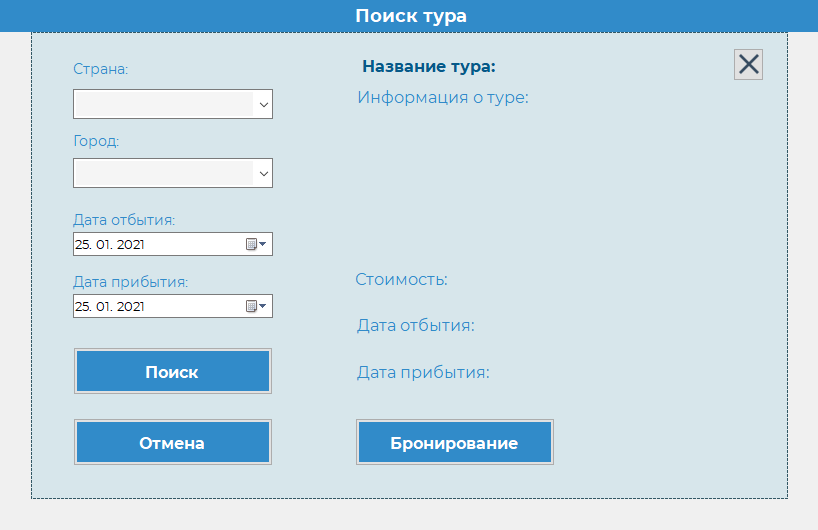


Рисунок 12 – Макет формы «Поиск тура»

При возврате в личный кабинет клиент нажимает выйти и происходит возращение на форму «Входа». Если входит администратор, то осуществляется переход на форму «Администратор», данная форма состоит из шести кнопок: «Данные о турах», «Данные клиентов», «Данные о сотрудниках», «Данные отелей», «Данные о транспорте» и «Выход», как показано на рисунке 13.

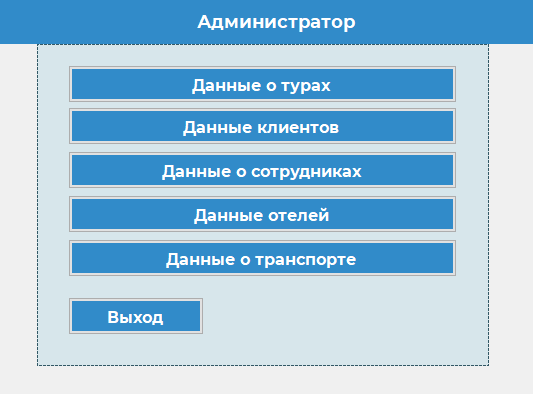


Рисунок 13 – Макет формы «Администратор»

При нажатии на кнопку «Данные о турах», осуществляется переход на другую форму, где отображаются все туры, при нажатии на запись в таблице, выбранная запись высвечивается в полях для ввода, и при изменении записи необходимо нажать на кнопку изменить, осуществляется обновление записи в таблице. Макет данной формы показан на рисунке 14. При клике на кнопку в виде крестика происходит переход обратно на форму «Администратора».

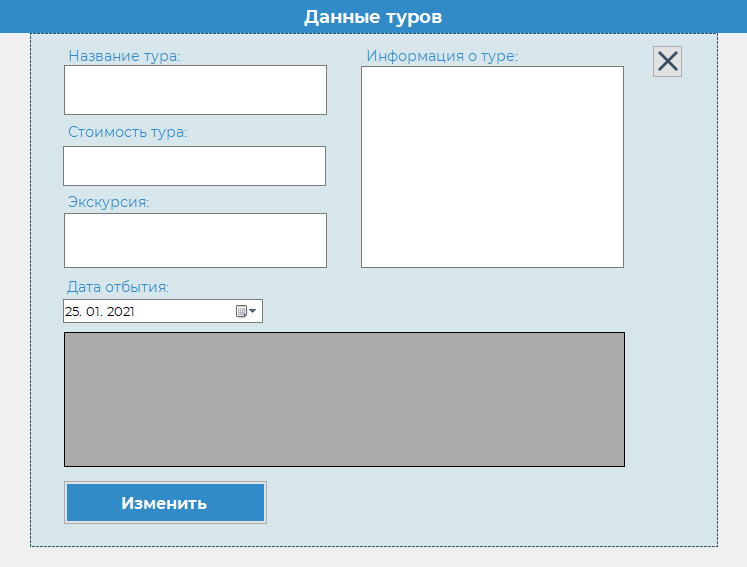


Рисунок 14 – Макет формы «Данные туров»

Кнопка «Данные клиентов» осуществляет переход на форму «Данные клиентов», на которой также есть таблица где отображаются все клиенты и есть кнопка «изменить». При нажатии на крест происходит возврат на форму «Администратора». Ниже показан макет для данной формы (Рисунок 15).

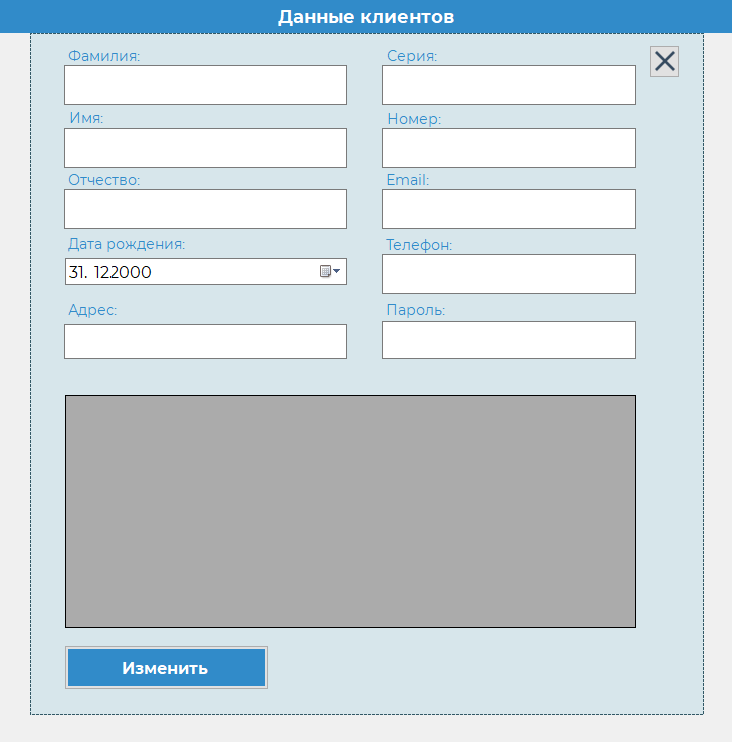


Рисунок 15 – Макет формы «Данные клиентов»

При нажатии на кнопку «Данные о сотрудниках», происходит переход на форму, где в таблице отображаются данные о сотрудников. Макет формы показан на рисунке 16.

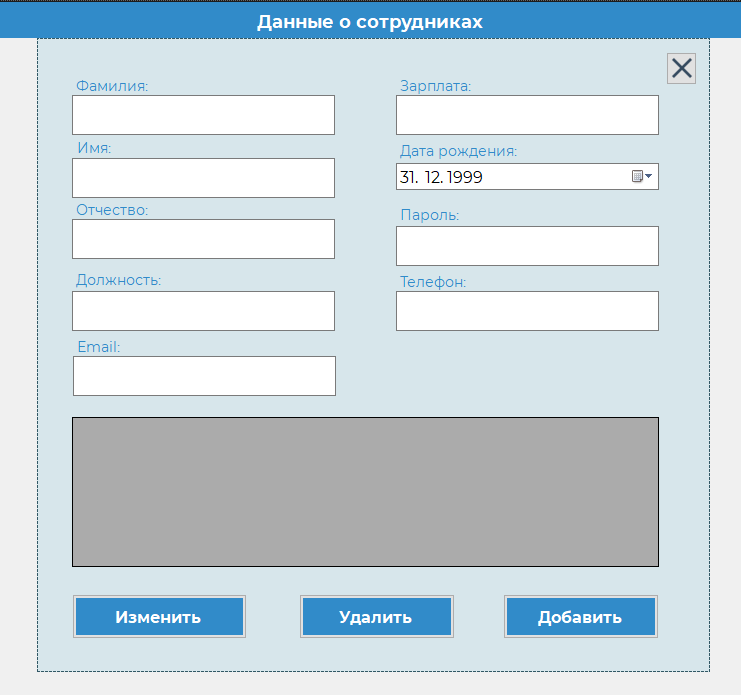


Рисунок 16 – Макет формы «Данные о сотрудниках»

При клике по кнопке «Добавить» переходит на форму «Добавление сотрудника», макет формы предоставлен ниже (Рисунок 17).

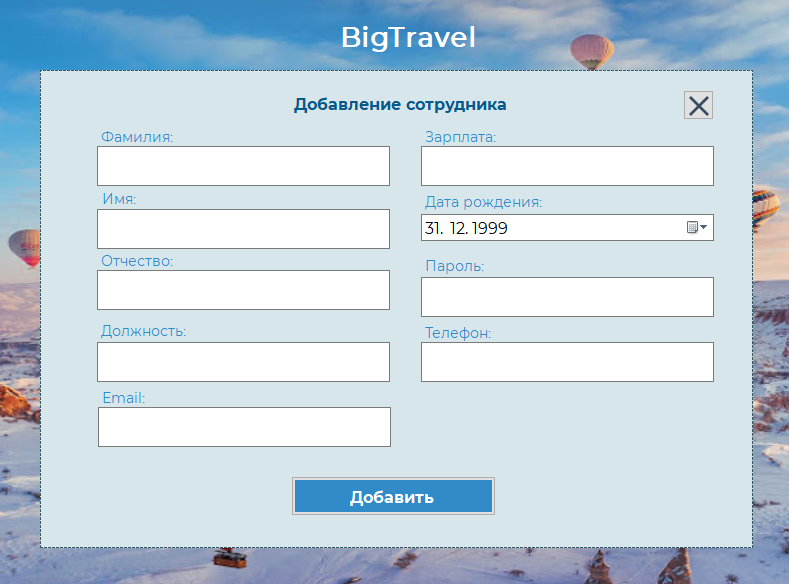


Рисунок 17 – Макет формы «Добавление сотрудника»

При возврате на форму «Администратора», нажав на кнопку «Данные отелей» происходит перебрасывание на форму «Отели». Мы видим, что на ней также, как и на других формах есть таблица где все сведения о отелях и кнопки «Изменить», «Добавить» и «Удалить». Ниже предоставлен макет формы (Рисунок 18).

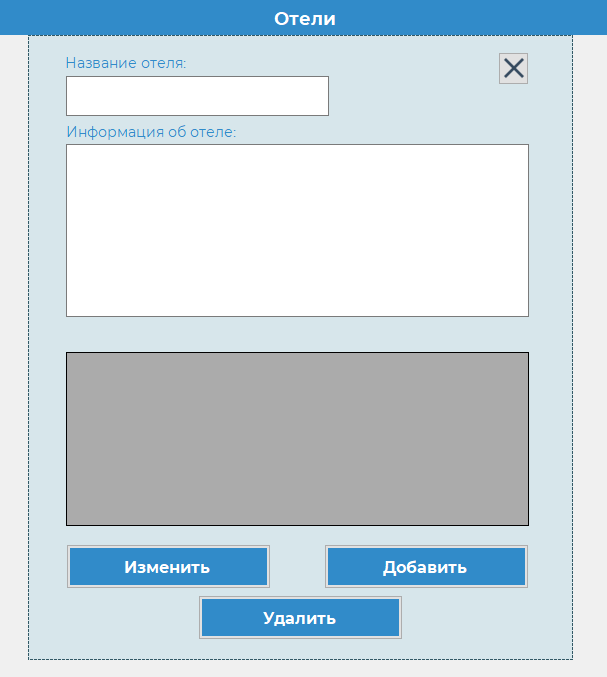


Рисунок 18­ – Макет формы «Отели»

При возврате на форму «Администратора» мы переходим на форму, «Данные о транспорте», которая представлена на рисунке 19. На ней представлена таблица, где просматриваются сведения о транспорте.

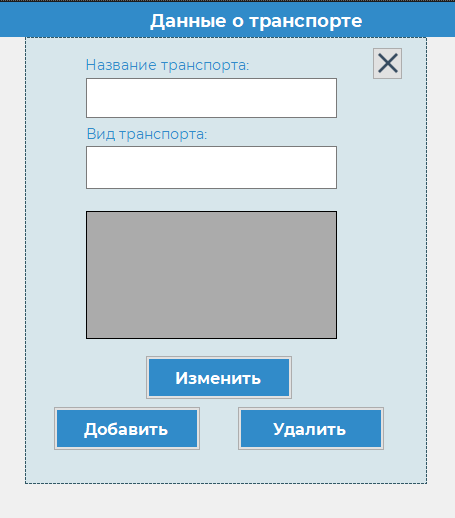


Рисунок 19 – Макет формы «Данные о транспорте»

Нажав на кнопку креста, перебрасывает назад на форму «Администратора». После нажатия на кнопку «Выход», осуществляется переход на главную форму.

При входе сотрудника в систему его перебрасывает на форму «Личный кабинет сотрудника», где отображается информация о нем, а также есть кнопки для перехода на форму «Данные о турах», «Данные клиентов», «Данные отелей», «Данные о транспорте», «Выход» и «Данные по бронированию». Ниже предоставлен макет формы (Рисунок 20).



Рисунок 20 – Макет формы «Личный кабинет сотрудника»

Первые четыре кнопки переходят на те же формы, как и у администратора (Рисунок 21-22).

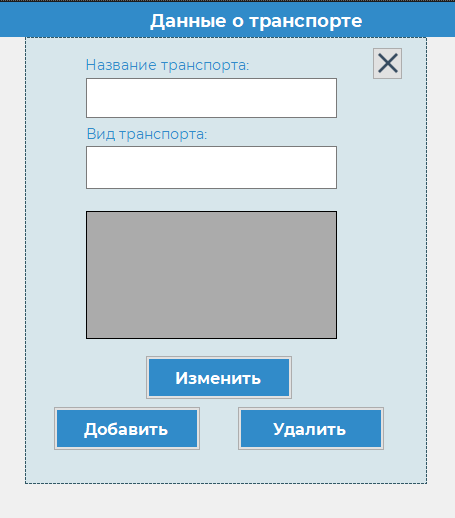
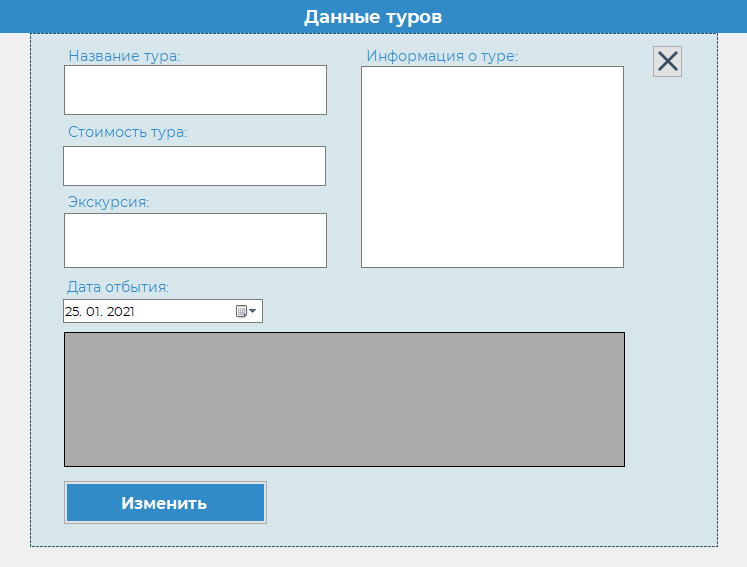


Рисунок 21 – Макет форм «Данные туров» и «Данные о транспорте»

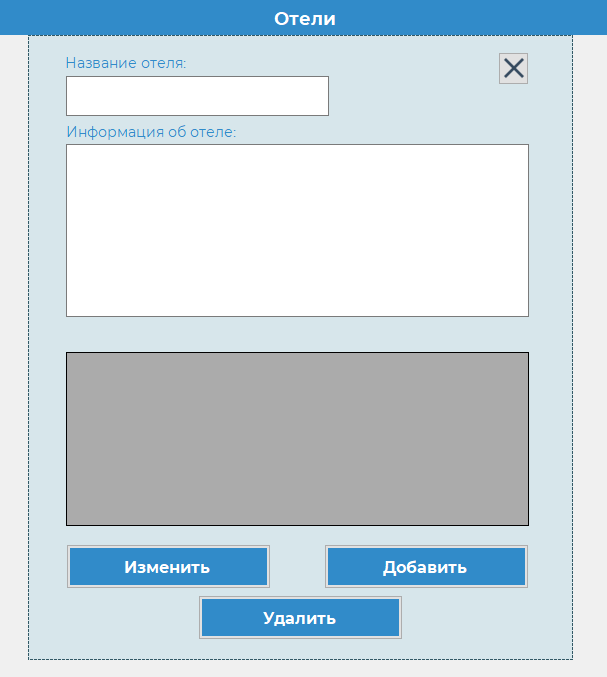
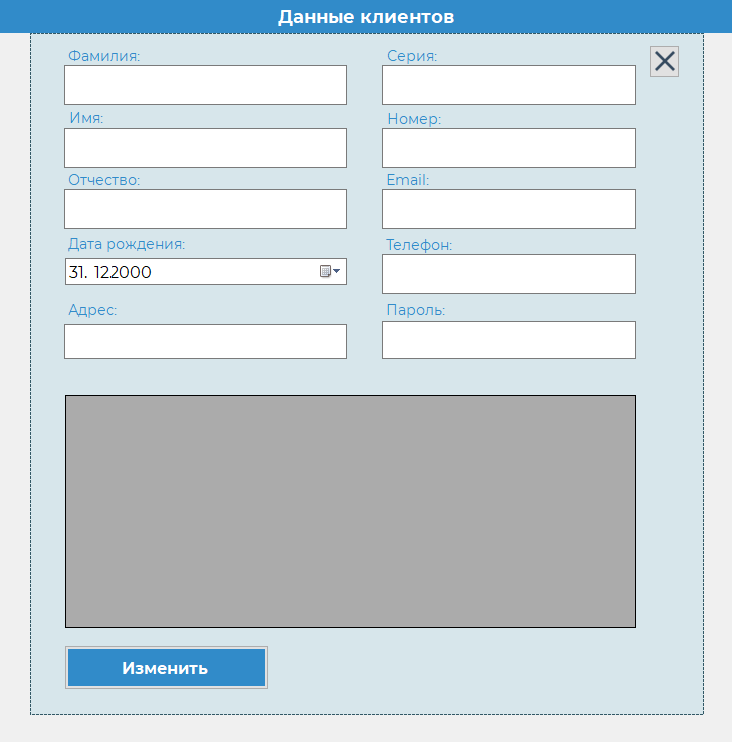


Рисунок 21 – Макет форм «Данные клиентов» и «Данные отелей»

При нажатии же на кнопку «Данные по бронированию» осуществляется переход на форму, где есть таблица по бронированию и кнопка для создания отчета, а также кнопка для возврата в личный кабинет сотрудника.

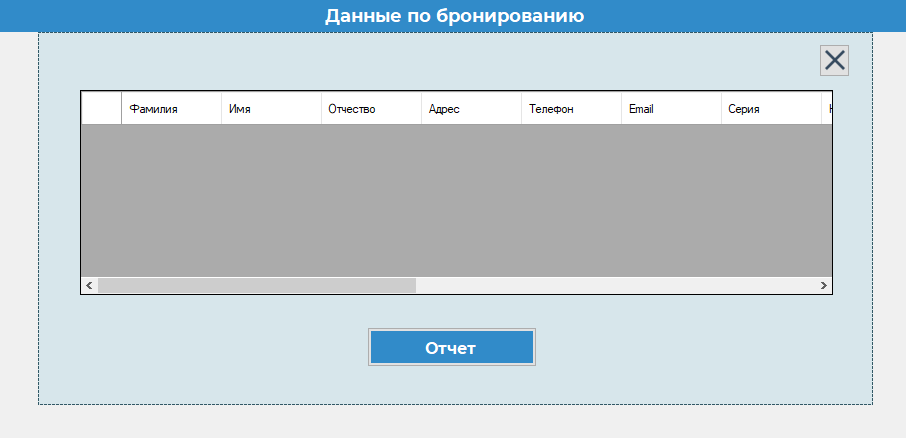


Рисунок 21 – Макет форм «Данные по бронированию»

Теперь, когда сформировался макет форм, в программе Visual Studio 2019 с помощью платформы Windows Forms создаем формы на основе макета.

3 МОДЕЛИРОВАНИЕ ПРЕЦЕДЕНТОВ

Диаграмма прецедентов используются для моделирования динамических аспектов ИС. Диаграммы этого типа позволяют достаточно четко описать и визуализировать поведение системы или ее части с точки зрения способа их использования. В результате, с одной стороны, пользователи системы понимают, как использовать некоторые элементы, а разработчики — как их реализовать. Диаграмма данного типа облегчает понимание системы и ее частей, представляя взгляд извне на то, как данные элементы могут быть использованы в соответствующем контексте. При этом достигается высокий уровень понимания функционирования всей системы в целом. Кроме того, такая диаграмма важна для организации эффективного тестирования систем в процессе прямого проектирования и для понимания их внутреннего устройства при обратном проектировании, когда создается модель уже работающей системы.

Диаграмма прецедентов может быть также использована для описания функциональности любого классификатора (classifier).

Диаграмма прецедентов системы «Туристической фирмы “BigTravel”» представлена в Приложении Г.

Анализируя данную диаграмму видно, что «Клиент» инициирует различные варианты использования системы: «Выбор тура», «Оформление заказа», «Оплата тура» и т.д. Смысл связи «включить» заключается в том, что прецедент, например, «Оформление заказа» включает в себя «Выбор страны», «Выбор даты отбытия», «Выбор даты прибытия», «Выбор отеля» «Выбор транспорта» и «Выбор города».

Ниже представлены спецификации прецедентов. Спецификация прецедента — основной источник информации для выполнения анализа и проектирования системы. Очень важно, чтобы содержание спецификации было представлено в полной и конструктивной форме. В общем случае спецификация включает: главный поток и подпотоки.

4 СПЕЦИФИКАЦИЯ ПРЕЦЕДЕНТОВ ИНФОРМАЦИОННОЙ СИСТЕМЫ «Туристической фирмы «BigTravel»»

Таблица 1 - Спецификация для прецедента «Выбор тура»

|  |  |
| --- | --- |
| Раздел | Описание |
| Краткое описание | Позволяет пользователю выбрать тур, узнать описание тура, название отеля, описание отеля и стоимость |
| Действующие лица | Клиент |
| Предусловия |  |
| Основной поток | 1. Система показывает информацию о туре, его описание, название отеля, описание отеля и цену 2. Пользователь выбирает действие «Выбор тура»   А1. Недостаточное количество товара   1. Сохранение заказа 2. Прецедент завершается |
| Альтернативный поток | А1. Тур не найден   1. Система выводит сообщение, что данного тура нет 2. Пользователь подтверждает просмотр сообщение 3. Поток возвращается на 1 этап основного потока |
| Постусловие | Результатом будет выбранный тур, который в дальнейшем можно будет заказать или отменить заказ |

Таблица 2 - Спецификация для прецедента «Оформление заказа»

|  |  |
| --- | --- |
| Раздел | Описание |
| 1 | 2 |
| Краткое описание | Позволяет пользователю оформить заказ на выбранный тур |
| Действующие лица | Клиент |
| Предусловия | - Выполнен прецедент выбор тура |
| Основной поток | 1. Система предлагает оформить заказ на выбранный тур пользователя 2. Пользователь вводит дату отбытия и дату прибытия, выбирает транспорт, страну, город и отель 3. Пользователь нажимает кнопку «Далее»   А1. Не все поля заполнены   1. Система сохраняет введённую пользователем информацию 2. Система предлагает оплатить заказ с помощью карты 3. Система запрашивает информацию о карте, ее номер, имя владельца, дату завершения срока действия и код безопасности 4. Пользователь вводит данные о банковской карте   А2. Неверно введены данные   1. Система отправляет код для подтверждения пользователю 2. Пользователь подтверждает отплату с помощью кода |

Продолжение таблицы 2

|  |  |
| --- | --- |
| 1 | 2 |
|  | А3. Недостаточно средств на счету   1. Система списывает нужную сумму с счёта банковской карты пользователя   Прецедент завершается |
| Альтернативный поток | А1. Не все поля заполнены   1. Система выводит сообщение, что есть незаполненное поле 2. Пользователь подтверждает просмотр сообщения 3. Поток возвращается на 2 этап основного потока   А2. Неверно введены данные   1. Система выводит сообщение, что пользователь неправильно ввёл данные о своей карты 2. Пользователь подтверждает просмотр сообщения 3. Поток возвращается на 6 этап основного потока   А3. Недостаточно средств на счету   1. Система выводит сообщение, что на счету пользователя недостаточно средств 2. Пользователь подтверждает просмотр сообщения 3. Поток возвращается на 6 этап основного потока |
| Постусловие | Поступление сохранения информации о заказе |

Таблица 3 - Спецификация для прецедента «Учет тура»

|  |  |
| --- | --- |
| Раздел | Описание |
| Краткое описание | Позволяет пользователю вести учёт о туре |
| Действующие лица | Сотрудник |
| Предусловия |  |
| Основной поток | 1. Уменьшение количества тура, в соответствии с заказом |
| Альтернативный поток |  |
| Постусловие |  |

Таблица 4 - Спецификация для прецедента «Отмена заказа»

|  |  |
| --- | --- |
| Раздел | Описание |
| 1 | 2 |
| Краткое описание | Позволяет пользователю отменить заказ |
| Действующие лица | Клиент |
| Предусловия | Оформление заказа |
| Основной поток | 1. Система выводит сообщение о возможности отменить заказ 2. Пользователь подтверждает отмену заказа 3. Система выводит сообщение о возврате средств |

Продолжение таблицы 4

|  |  |
| --- | --- |
| 1 | 2 |
|  | 1. Система запрашивает информацию о карте, ее номер, имя владельца, дату завершения срока действия и код безопасности 2. Пользователь вводит свои реквизиты   А1. Неверно введены данные   1. Система отправляет код для подтверждения пользователю 2. Пользователь подтверждает отмену с помощью кода 3. Система подтверждает отмену заказа   Прецедент завершён |
| Альтернативный поток | А1. Неверно введены данные   1. Система выводит сообщение, что пользователь неправильно ввёл данные о своей карты 2. Пользователь подтверждает просмотр сообщения 3. Поток возвращается на 5 этап основного потока |
| Постусловие | Увеличение количество туров |

Таблица 5 - Спецификация для прецедента «Формирование тура»

|  |  |
| --- | --- |
| Раздел | Описание |
| 1 | 2 |
| Краткое описание | Позволяет пользователю сформировать новый тур |
| Действующие лица | Сотрудник |
| Предусловия |  |
| Основной поток | 1. Система предлагает создать новый тур 2. Пользователь вводит дату прибытия, дату отбытия, выбор страны, добавить отель и транспорт 3. Пользователь нажимает кнопку «Далее»   А1. Не все поля заполнены   1. Система сохраняет введённую пользователем информацию |
| Альтернативный поток | А1. Не все поля заполнены   1. Система выводит сообщение, что есть незаполненное поле 2. Пользователь подтверждает просмотр сообщения 3. Поток возвращается на 2 этап основного потока |
| Постусловие |  |

После того, как была представлена спецификация прецедентов, разрабатываем диаграмму классов-сущностей.

5 ДИАГРАММА КЛАССОВ

Классы используются в процессе анализа предметной области для составления словаря предметной области разрабатываемой системы. Это могут быть как абстрактные понятия предметной области, так и классы, на которые опирается разработка и которые описывают программные или аппаратные сущности.

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы.

Графически класс отображается в виде прямоугольника, который может быть разделен горизонтальными линиями на секции. В этих секциях указывается имя, атрибуты (свойства) и операции (методы). Секция атрибутов выделяется горизонтальной линией, даже если у класса отсутствуют атрибуты (характерно для классов-интерфейсов).

С точки зрения структурного подхода, атрибуты – это переменные, а методы – это функции, описанные в теле класса. Они могут быть доступны или не доступны для изменения (атрибуты) или выполнения (методы) внешними объектами.

Обязательным элементом обозначения класса на диаграмме является его имя. Оно должно быть уникальным в пределах пакета. Если класс является абстрактным, то его имя пишется курсивом. Абстрактный класс – это класс, на основе которого нельзя создать объекты. Такие классы используются в качестве шаблона для дочерних классов при наследовании.

В секции имени класса может быть указан стереотип (например, "entity", "boundary", "interface" и т. п.).

6 ВИДЫ ДЕЯТЕЛЬНОСТИ

Диаграмма деятельности - это, по существу, блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой, при этом внимание фиксируется на результате деятельности. Результат может привести к изменению состояния системы или возвращению некоторого значения.

Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия или состояния деятельности, а дугами - переходы от одного состояния действия/деятельности к другому. Каждая диаграмма деятельности должна иметь единственное начальное и единственное конечное состояния (на практике иногда можно видеть несколько конечных состояний на одной диаграмме, но это одно и тоже состояние, изображенное несколько раз для лучшей читабельности диаграммы). Саму диаграмму деятельности принято располагать таким образом, чтобы действия следовали сверху вниз. В этом случае начальное состояние будет изображаться в верхней части диаграммы, а конечное - в ее нижней части.

Ниже представлена диаграмма деятельности, построенных для проектируемой системы (Приложение В)

Клиент может выбрать тур, после выбора-заказать тур, а также оплатить свой заказ. Сотрудник, получая информацию о заказе, изменяет количество тура. Оформляет путевку покупателя и отправляет ее на почту клиенту.

1. МОДЕЛИРОВАНИЕ ВЗАИМОДЕЙСТВИЙ

Моделирование взаимодействий (interaction modeling) охватывает вопросы взаимодействия между объектами, необходимыми для выполнения прецедента. Модели взаимодействия используются на более развитых стадиях анализа требований, когда становится известной модель классов, так что ссылки на объекты опираются на модель классов. Моделирование взаимодействий отображает последовательность событий (сообщений) в их связи с действующими в кооперации объектами.

Взаимодействие (interaction) представляет собой набор сообщений, свойственных поведению некоторой системы, которыми обмениваются объекты ъ соответствии с установленными между ними связями. Диаграмма последовательностей представляется двумерным графом. Объекты располагаются по горизонтали. Последовательности сообщений располагаются сверху вниз по вертикали. Каждая вертикальная линия называется линией жизни (lifeline) объекта.

Стрелки представляют каждое сообщение, направляемое от вызывающего объекта (отправителя) к операции (методу) вызываемого объекта (получателя). Для каждого сообщения, как минимум, указывается его имя. Кроме того, для сообщения могут быть указаны фактические аргументы сообщения и другая управляющая информация. Фактические аргументы соответствуют формальным аргументам метода объекта-получателя.

Ниже представлена диаграмма последовательности (Приложение Б).

Клиенту после выбора тура необходимо оформить заказ и оплатить его, после оформления заказа система обрабатывает информацию и передаёт информацию о туре сотруднику, где производится изменение количества туров, а также оформляется путевка для покупателя.

1. ДИАГРАММА СОСТОЯНИЙ

Диаграммы состояний применяются для того, чтобы объяснить, каким образом работают сложные объекты. Диаграмма показывает, как объект переходит из одного состояния в другое. Очевидно, что диаграммы состояний служат для моделирования динамических аспектов системы (как и диаграммы последовательностей, кооперации, прецедентов и, как мы увидим далее, диаграммы деятельности). Диаграмма состояний полезна при моделировании жизненного цикла объекта.

От других диаграмм, диаграмма состояний отличается тем, что описывает процесс изменения состояний только одного экземпляра определенного класса - одного объекта, причем объекта реактивного, то есть объекта, поведение которого характеризуется его реакцией на внешние события. Понятие жизненного цикла применимо как раз к реактивным объектам, настоящее состояние (и поведение) которых обусловлено их прошлым состоянием. Но диаграммы состояний важны не только для описания динамики отдельного объекта. Они могут использоваться для конструирования исполняемых систем путем прямого и обратного проектирования.

Но поговорим об обозначениях на диаграммах состояний. Скругленные прямоугольники представляют состояния, через которые проходит объект в течение своего жизненного цикла. Стрелками показываются переходы между состояниями, которые вызваны выполнением методов описываемого диаграммой объекта. Существует также два вида псевдосостояний: начальное, в котором находится объект сразу после его создания (обозначается сплошным кружком), и конечное, которое объект не может покинуть, если перешел в него (обозначается кружком, обведенным окружностью).

Ниже представлена диаграмма состояний для информационной системы «Туристической фирмы “BigTravel”» (Приложение Д).

ER – диаграмма —это разновидность блок-схемы, где показано, как разные «сущности» связаны между собой внутри системы. ER-диаграммы чаще всего применяются для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса. ER-диаграммы (или ER-модели) полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные линии, для отображения сущностей, их атрибутов и связей. Эти диаграммы устроены по тому же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи — глаголов.

Сущности базы данных описаны в таблице 6.

Таблица 6 – Сущности базы данных

|  |  |
| --- | --- |
| Сущность | Описание сущности |
| administrator | Данные о администраторах турфирмы |
| Country | Список стран |
| Customer | Данные о клиентах |
| Employes | Данные о сотрудниках |
| hotel | Данные о отелях |
| Orders | Данные о забронированных турах |
| reis | Данные о рейсах |
| Route | Список маршрутов |
| Transport | Данные о транспорте |
| Vouchers | Данные о турах |

Далее в таблицах 7-16 представлены структуры описанных сущностей:

Таблица 7 - Схема отношения Клиент (Customer)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Id-клиента | id\_customer | Int | Первичный ключ |
| Серия | Series | Varchar(4) | Обязательное поле |
| Номер | Number | Varchar(6) | Обязательное поле |
| Фамилия | sername | Varchar(50) | Обязательное поле |
| Имя | name | Varchar(30) | Обязательное поле |
| Отчество | patronymic | Varchar(50) | Необязательное поле |
| Дата рождения | date\_of\_birth | Date | Обязательное поле |
| Адрес | adress | Varchar(50) | Обязательное поле |
| Пароль | Password | Varchar(50) | Обязательное поле |
| Email | Email | Varchar(50) | Обязательное поле |
| Телефон | phone | Varchar(11) | Обязательное поле |
| Id-администратора | id\_admin | Int | Внешний ключ |

Таблица 8 - Схема отношения Сотрудник (Employes)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Id-клиента | id\_customer | Int | Первичный ключ |
| Должность | department | Varchar(50) | Обязательное поле |
| Фамилия | sername | Varchar(50) | Обязательное поле |
| Имя | name | Varchar(30) | Обязательное поле |
| Отчество | patronymic | Varchar(50) | Необязательное поле |
| Дата рождения | date\_of\_birth | Date | Обязательное поле |
| Зарплата | salary | Decimal(18, 0) | Обязательное поле |
| Email | Email | Varchar(50) | Обязательное поле |
| Пароль | Password | Varchar(50) | Обязательное поле |
| Телефон | phone | Varchar(11) | Обязательное поле |
| Id-администратора | id\_admin | Int | Внешний ключ |

Таблица 9 - Схема отношения Заказы (Orders)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Id-заказа | id\_orders | Int | Первичный ключ |
| Id-клиента | id\_customer | Int | Внешний ключ |
| Id-сотрудника | id\_employes | Int | Внешний ключ |
| Id-тура | id\_vouchers | Int | Внешний ключ |
| Дата регистрации | Date\_registration | date | Обязательное поле |
| Id-администратора | id\_admin | Int | Внешний ключ |

Таблица 10 - Схема отношения Администратор (administrator)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания | |
| Id-пользователя | id\_admin | Int | Первичный ключ | |
| Фамилия | sername | Varchar(50) | Обязательное поле |
| Имя | name | Varchar(30) | Обязательное поле |
| Отчество | patronymic | Varchar(50) | Необязательное поле |
| Дата рождения | date\_of\_birth | Date | Обязательное поле |
| Email | Email | Varchar(50) | Обязательное поле |
| Пароль | Password | Varchar(50) | Обязательное поле |
| Телефон | phone | Varchar(11) | Обязательное поле |
| Зарплата | salary | Decimal(18, 0) | Обязательное поле |

Таблица 11 - Схема отношения Отель (hotel)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Id-отеля | id\_hotel | Int | Первичный ключ |
| Название отеля | Namehotel | Varchar(50) | Обязательное поле |
| Описание отеля | description\_hotel | Varchar(MAX) | Обязательное поле |
| Id-администратора | id\_admin | Int | Внешний ключ |

Таблица 12 - Схема отношения Рейс (reis)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Id-рейса | id\_reis | Int (11) | Первичный ключ |
| Id-заказа | id\_order | Int | Внешний ключ |
| Id-транспорта | id\_transport | Int | Внешний ключ |
| Вид транспорта | view\_transport | Varchar(50) | Обязательное поле |
| Пункт отправления | Point\_departure | Varchar(30) | Обязательное поле |
| Место назначения | Destination | Varchar(20) | Обязательное поле |
| Дата отправления | Departure\_date | date | Обязательное поле |
| Дата прибытия | Date\_arrival | date | Обязательное поле |
| Id-администратора | id\_admin | Int | Внешний ключ |

Таблица 13 - Схема отношения Транспорт (Transport)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Id-транспорта | id\_transport | Int | Первичный ключ |
| Название транспорта | Name\_transport | Varchar(30) | Обязательное поле |
| Вид транспорта | view\_transport | Varchar(50) | Обязательное поле |
| Id-администратора | id\_admin | Int | Внешний ключ |

Таблица 14 - Схема отношения Список маршрута (Route)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Id-маршрута | id\_route | Int | Первичный ключ |
| Id-страны | id\_country | Varchar(3) | Внешний ключ |
| Населенный пункт | human\_settlement | Varchar(50) | Обязательное поле |
| Id-администратора | id\_admin | Int | Внешний ключ |

Таблица 15 - Схема отношения Список стран (Country)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Id-страны | id\_country | Varchar(3) | Первичный ключ |
| Название страны | Name\_country | Varchar(30) | Обязательное поле |
| Описание страны | Description\_country | Varchar(MAX) | Обязательное поле |
| Id-администратора | id\_admin | Int | Внешний ключ |

Таблица 16 - Схема отношения Туры (Vouchers)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Id-тура | id\_vouchers | Int | Первичный ключ |
| Название тура | name\_vouchers | Varchar(50) | Обязательное поле |
| Id-маршрута | id\_route | Int | Внешний ключ |
| Id-отеля | id\_hotel | Int | Внешний ключ |
| Дата отправления | Departure\_date | Date | Обязательное поле |
| Дата прибытия | Date\_arrival | Date | Обязательное поле |
| Информация о туре | Information\_vouchers | Varchar(MAX) | Обязательное поле |
| Экскурсия | Excursions | Varchar(MAX) | Обязательное поле |
| Стоимость | Cost | Decimal(20, 0) | Обязательное поле |
| Количество | number\_trips | Varchar(50) | Обязательное поле |
| Id-администратора | id\_admin | Int | Внешний ключ |

ER – диаграмма базы данных туристическая фирма «BigTravel» представлена ниже (приложении А).

1. РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ БАЗЫ ДАННЫХ

В базе данных туристической фирмы «BigTravel» должна быть отражена информация о клиентах, доступных турах, а также сведения об отелях и транспорте. Поэтому входными данными для проектирования базы данных должны быть следующие характеристики:

* информация о клиентах: ФИО, дата рождения, адрес, телефон, Email, серия и номер паспорта;
* информация о турах: дата начала и конца, название тура, стоимость, название отеля, транспорт;
* информация об отеле: название отеля, описание отеля, место расположения.

База данных туристической фирмы «BigTravel» должна выполнять следующие задачи:

* предоставлять возможность для внесения, изменения и удаления информации о клиентах, турах, отелях и транспорте сотрудникам турфирмы;
* предоставлять информацию о доступных предложениях клиентам;

База данных была разработана в Microsoft SQL Server. При разработке БД было выделено десять сущностей: administrator, country, customer, employes, hotel, orders, reis, route, transport, vouchers. Для атрибутов каждой сущности были выбраны типы данных.

Процесс нормализации состоит в том, чтобы представить данные в виде набора таблиц, в которых все неключевые поля зависят только от целого - возможно, составного - ключа. Тем самым минимизируется избыточность данных, и ограничиваются несвязанные структуры. Поэтому для нормализации были добавлены таблицы Country и Transport. В итоге все таблицы были приведены к третьей нормальной форме.

Также было подключены каскадные обновления и каскадное удаление. При каскадном обновлении, изменение значения первичного ключа таблицы, связанные данные во внешних ключах, относящихся к этой таблице, также изменяются, отражая изменения в первичном ключе. В каскадном удалении, при удалении записи в таблице все записи, связанные с этой записью в соответствующих таблицах, автоматически удаляются.

У каждого актера свои возможности взаимодействия с теми или иными сущностями.

Администратору доступно полное изменение таблиц: customer, employes, hotel, route, transport, vouchers.

Сотрудник может изменять данные таблиц: hotel, route, transport, vouchers.

Клиент же может просматривать таблицу vouchers и hotel.

Когда серверная часть базы данных реализована можно переходить к разработке клиентской части.

1. РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ БАЗЫ ДАННЫХ

После реализации серверной части, разработка выполнялась на платформе Windows Forms в Visual Studio 2019, с использованием Sql- запросов для подключения к БД, а также CRUD-запросов для добавления/изменения удаления и JOIN запросы для соединения. Соединение – это операция, когда таблицы сравниваются между собой построчно и появляется возможность вывода столбцов из всех таблиц, участвующих в соединении.

После того как пользователь авторизируется с помощью соответствующей формы, данные отправляются в базу данных для сравнения в таблицы: administrator, customer и employes, после чего происходит (либо не происходит), непосредственно, вход.

После регистрации клиента, он может войти в свой личный кабинет, где может изменить данные о себе, а также просмотреть заказы и просмотреть доступные туры. Клиент может найти тур введя определенные параметры: страну прибытия, город, дату отбытия и дату прибытия. После ввода параметров будет высвечиваться найденный тур, клиент может забронировать этот тур, а после купить. А также при обратном возвращении в личный кабинет он может выйти из приложения.

ЗАКЛЮЧЕНИЕ

В ходе прохождения учебной практики была реализован программный

продукт для организации работы туристической фирмы, состоящая из базы данных в Microsoft SQL Server и самого приложения в Visual Studio 2019 с использованием платформы Windows Form.

Новые знания, которые получены в ходе прохождения учебной практики:

* Отправка файла на электронную почту;
* Вывод данных в таблицу из базы дынных;
* Оптимизация переходов между формами;
* Работа с разными запросами через C#;
* Создание PDF-файла.

Не было реализовано:

* Бронирование тура;
* Составление договора при бронировании между сотрудником и клиентом;
* Покупка тура;
* Добавление, изменение, удаление маршрута;
* Отмена бронирования и покупки.

В будущем данные функции будут сделаны и доработаны уже имеющиеся.

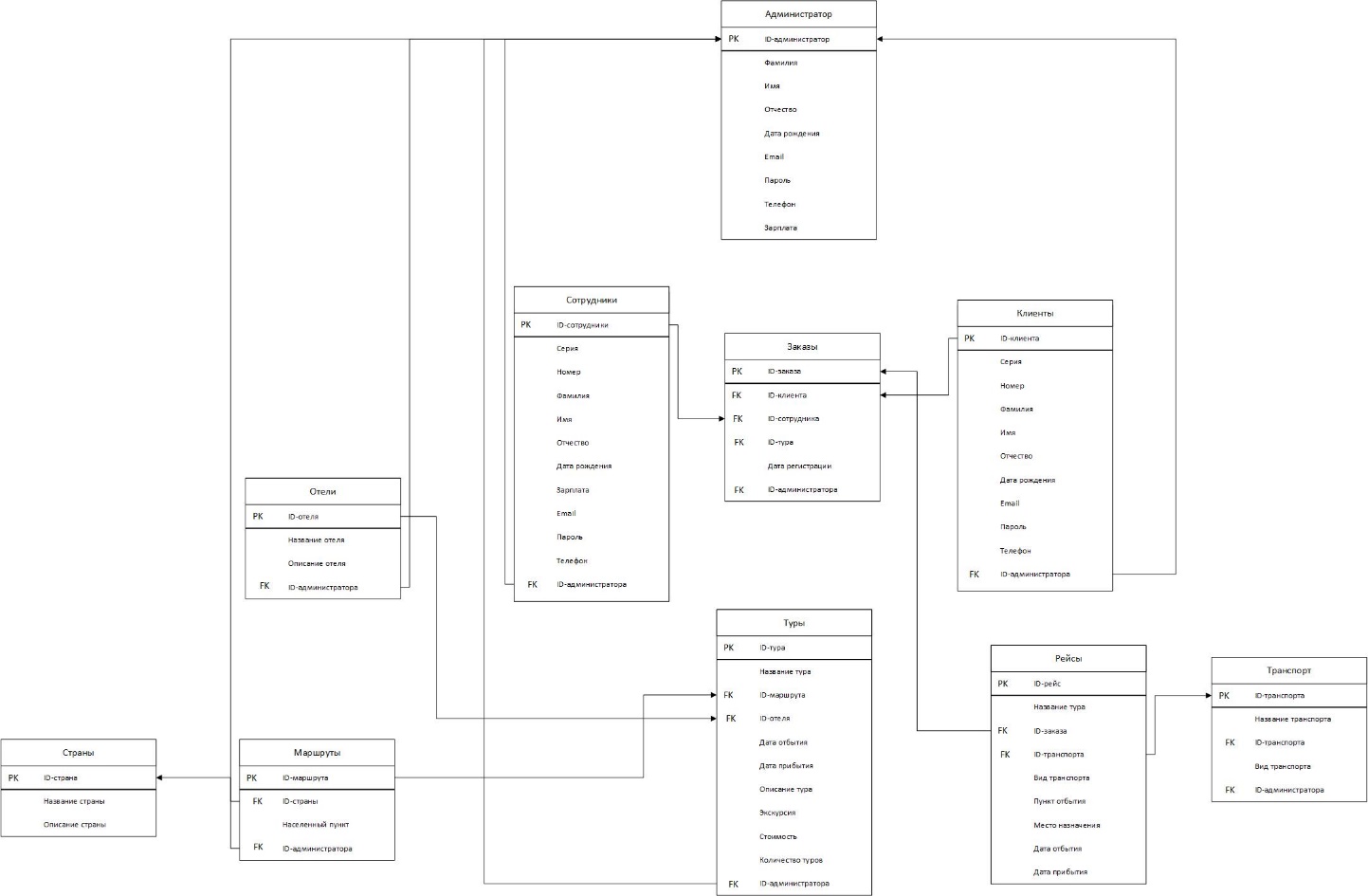
СПИСОК ИСТОЧНИКОВ

1. Хелпикс.Орг [Электронный ресурс]: Диаграммы прецедентов- Режим доступа к руководству:  <https://helpiks.org/9-53447.html>
2. ИНТУИТ [Электронный ресурс]: Диаграмма классов- Режим доступа к руководству: <https://intuit.ru/studies/courses/1007/229/lecture/5954?page=2>
3. ИНТУИТ [Электронный ресурс]: Диаграмма деятельности и особенности ее построения- Режим доступа к руководству: <https://intuit.ru/studies/courses/32/32/lecture/1020>
4. StudFiles [Электронный ресурс]: Моделирование взаимодействий- Режим доступа к руководству: <https://studfile.net/preview/4452586/page:4/>
5. Хелпикс.Орг [Электронный ресурс]: Диаграмма состояний-Режим доступа к руководству: <https://intuit.ru/studies/courses/1007/229/lecture/5954?page=4>

ПРИЛОЖЕНИЕ А

(информационное)

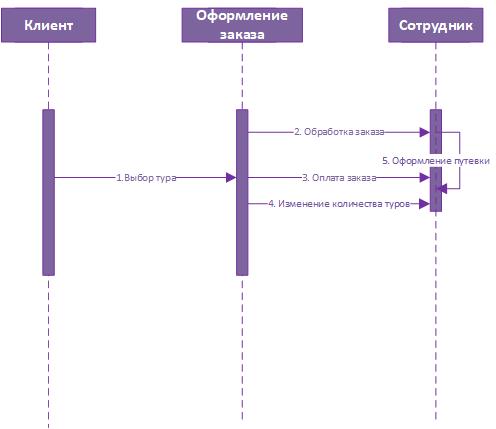
ER – диаграмма



ПРИЛОЖЕНИЕ Б

(информационное)

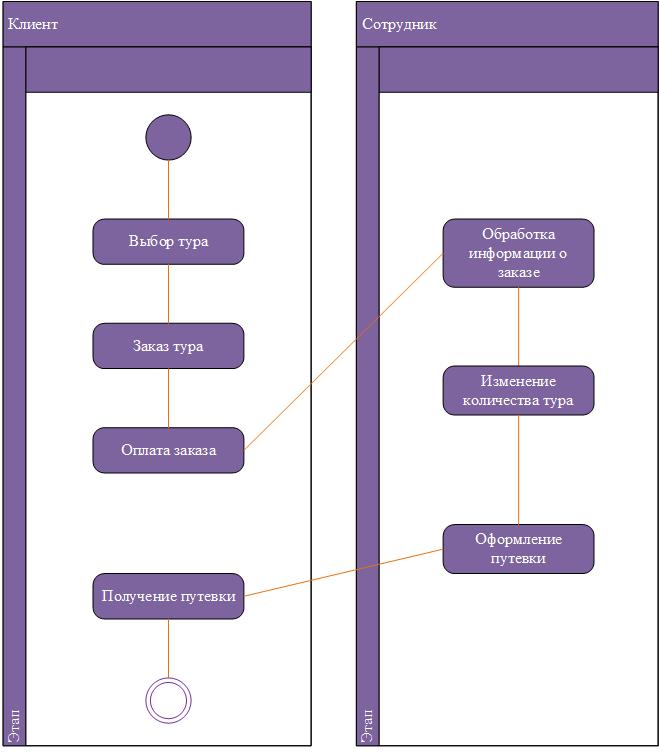
Диаграмма последовательности



ПРИЛОЖЕНИЕ В

(информационное)

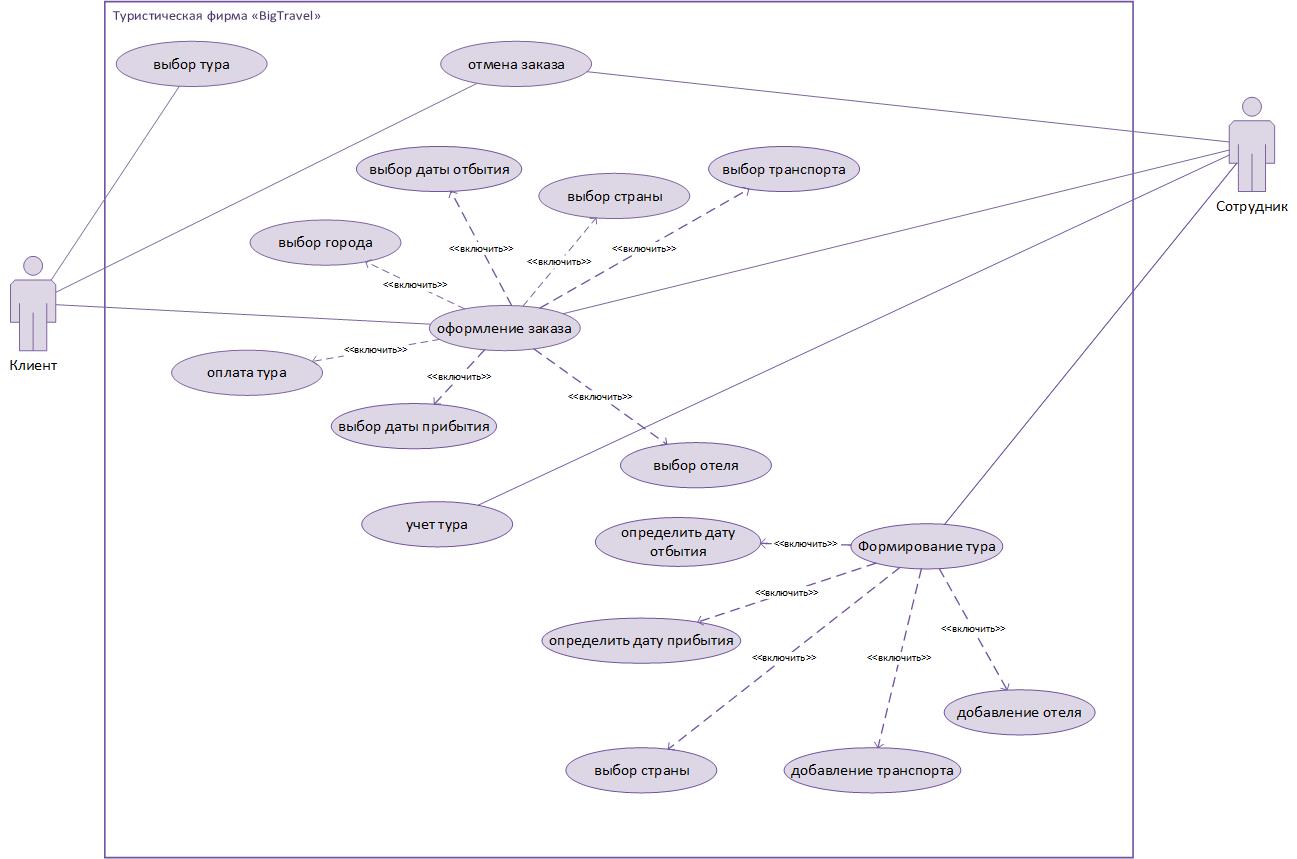
Диаграмма видов деятельности



ПРИЛОЖЕНИЕ Г

(информационное)

Диаграмма прецедентов



ПРИЛОЖЕНИЕ Д

(информационное)

Диаграмма состояний

