

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
Кафедра Прикладной информатики и информационных технологий

РАЗРАБОТКА ПОДСИСТЕМЫ БРОНИРОВАНИЯ И ПРОДАЖИ АВИАБИЛЕТОВ

Выпускная квалификационная работа студента

**очной формы обучения
направления подготовки 09.03.03. Прикладная информатика
4 курса группы 07001204
Босого Ярослава Андреевича**

Научный руководитель
кандидат физико-математических наук
доцент Кузьмичева Т.Г.

БЕЛГОРОД 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ИЗУЧЕНИЕ И АНАЛИЗ ИСПОЛЬЗОВАНИЯ ИНФОРМАЦИОННОЙ ПОДСИСТЕМЫ В АО “БЕЛГОРОДАВИА”	5
1.1 Общая характеристика структуры и деятельности “Белгородавиа”	5
1.2 Краткая характеристика подразделения или видов его деятельности .	9
1.3 Анализ существующих подсистем бронирования и продажи авиабилетов	10
1.4 Обоснование необходимости автоматизации и цели создания подсистемы бронирования и продажи авиабилетов	12
1.5 Разработка структурно функциональной диаграммы («КАК ЕСТЬ») по методологии SADT(IDEF0).....	13
1.6 Постановка задачи	17
2 ОБОСНОВАНИЕ ПРОЕКТНЫХ РЕШЕНИЙ.....	19
2.1 Обоснование проектных решений по техническому обеспечению ...	19
2.2 Обоснование проектных решений по информационному обеспечению.....	21
2.3 Обоснование проектных решений по программному обеспечению ..	26
3 ПРОЕКТНАЯ ЧАСТЬ	28
3.1 Разработка структурно функциональной диаграммы («КАК ДОЛЖНО БЫТЬ») по методологии SADT(IDEF0)	28
3.2 Разработка программных модулей	32
3.3 Описание контрольного примера реализации проекта.....	44
3.4 Калькуляция себестоимости научно-технической продукции	51
ЗАКЛЮЧЕНИЕ	58
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	59
ПРИЛОЖЕНИЯ	64

ВВЕДЕНИЕ

Темой данной выпускной квалификационной работы является разработка и создание подсистемы бронирования и продажи авиабилетов.

Актуальность выбранной темы обусловлена тем, что в настоящее время в РФ отсутствует такая отечественная подсистема, которая бы обеспечивала работу специалиста в авиакассе. В основном используются подсистемы из зарубежных стран, на которых хранятся все сведения о пассажирах и их перелётах. Из-за этого очень большое количество средств тратится на обслуживание данного программного обеспечения и для оптимальной работы с данным продуктом зачастую требуется бесперебойное подключение к сети интернет.

Целью работы является разработка подсистемы бронирования и продажи авиабилетов, предназначенной для процесса работы авиакасс в России.

В соответствии с целью задачами выпускной квалификационной работы будут являться:

- изучение предметной области;
- анализ структуры и организации деятельности АО “Белгородавиа”;
- выявление проблем информационной поддержки авиакасс;
- оценка экономической эффективности внедрения разработки бронирования и продажи авиабилетов;
- проведение анализа и выбор средств и технологий для разработки подсистемы;
- разработка функциональных моделей и создание программного продукта для работы в авиакассе;

Выпускная квалификационная работа состоит из введения, 3 разделов:

- Изучение и анализ использования информационной подсистемы в АО “Белгородавиа”;

– Обоснование проектных решений автоматизированной подсистемы бронирования и продажи авиабилетов;

– Проектная часть;

Так же в ВКР включено заключение, список использованных источников и приложения.

В первой разделе выпускной квалификационной работы описывается общая характеристика структуры и деятельности АО “Белгородавиа”, краткая характеристика подразделения или видов его деятельности, анализ существующих автоматизированных подсистем бронирования и продажи авиабилетов, обоснование необходимости автоматизации и цели создания подсистемы, выделяются основные задачи.

Во втором разделе выпускной квалификационной работы выявляется обоснование программных решений автоматизированной подсистемы бронирования и продажи авиабилетов. Принимаются решения по технологическому, информационному и программному обеспечению.

В третьей разделе происходит анализ и оценка эффективности программных решений информационной подсистемы, создается и тестируется готовый программный продукт.

Выпускная квалификационная работа включает в себя 5 таблиц, 45 рисунков

1 ИЗУЧЕНИЕ И АНАЛИЗ ИСПОЛЬЗОВАНИЯ ИНФОРМАЦИОННОЙ ПОДСИСТЕМЫ В АО “БЕЛГОРОДАВИА”

1.1 Общая характеристика структуры и деятельности “Белгородавиа”

Полное фирменное наименование Общества: Акционерное общество «Белгородское авиапредприятие».

Местонахождение Общества: 308000, г. Белгород, Проспект Богдана Хмельницкого, дом 166.

Акционерное общество “Белгородавиа”, именуемое в дальнейшем “Общество”, приобрело свои права юридического лица с момента его государственной регистрации (30 августа 1954 года) и осуществляет свою деятельность в соответствии с положениями настоящего Устава, действующего законодательства Российской Федерации, и на основании полученных сертификатов (Сертификат АС, АБ; Сертификат на международный аэропорт, на обслуживание пассажиров и грузов, на штурманское обеспечение, на аэропортовскую деятельность, на техническое обслуживание; ГОСТ ISO 9001-2011; СПАСОП; ЭСТОП)

Рассмотрим технические характеристики. Аэропорт Белгорода имеет одну ВПП с искусственным асфальтобетонным покрытием длиной 2 500 и шириной 45 метра. Максимальная взлётная масса воздушного судна составляет 190 тонн. Аэропорт может принимать все узкофюзеляжные и некоторые широкофюзеляжные самолеты. При максимальной загруженности аэропорт может осуществлять до 16 взлётно-посадочных операций в час.

Пропускная способность нового аэровокзального комплекса — до 450 пассажиров в час: 300 — на внутренних рейсах и 150 на международных. По сравнению с 2010 годом, в 2011 объём пассажирских перевозок увеличился на 45 % и составил 132 тыс. человек. За 5 месяцев 2012 года увеличение по сравнению с аналогичным периодом прошлого года составило 47 % [30].

Грузовой терминал общей площадью 1000 м² рассчитанный на одновременное хранение международных и внутренних грузов и почты объёмов до шести тонн. Расчетный грузооборот участка по обработке грузов и почты 2,3 тонны в сутки.

На рисунке 1.1 представлена организация подразделений предприятия.

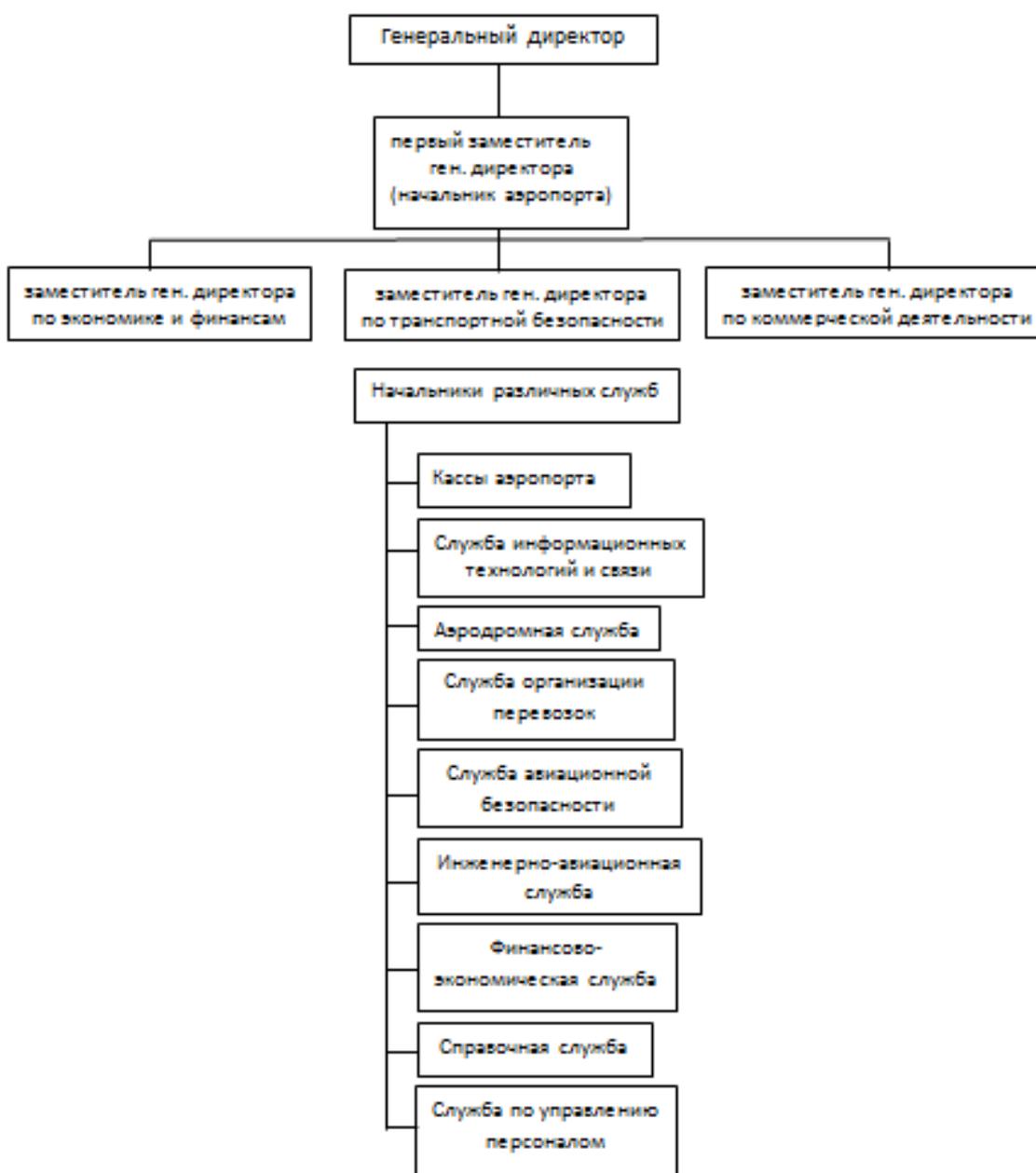


Рисунок 1.1 – Организационная модель предприятия

Первые четыре слоя организационной модели составляют различного рода директора, на них возлагается: общее руководство производственно-хозяйственной и финансово-экономической деятельностью предприятия; организация взаимодействия всех структурных подразделений, цехов и производственных единиц; обеспечение выполнения всех принимаемых предприятием обязательств, включая обязательства перед бюджетами разных уровней и внебюджетными фондами, а также по договорам; создание условий для внедрения новейшей техники и технологии, прогрессивных форм управления и организации труда; принятие мер по обеспечению здоровых и безопасных условий труда на предприятии; контроль за соблюдением законодательства Российской Федерации в деятельности всех служб; защита имущественных интересов предприятия в суде, органах государственной власти [28].

Справочная служба занимается приемом входящих звонков и выдачей необходимой информации для клиентов.

Главной целью службы организации перевозок является обеспечение безопасности и регулярности полетов, поддержание высокого уровня культуры обслуживания пассажиров, грузовой клиентуры. Служба организации перевозок является производственной службой аэропорта, которая обеспечивает обслуживание пассажиров, обработку багажа, груза и почты на внутренних и международных воздушных линиях.

Финансово-экономическая служба:

- ведение бухгалтерского учета в соответствии с российским законодательством;
- представление налоговой и статистической отчетности в контролирующие органы;
- формирование международной финансовой отчетности;
- ведение управленческого учета финансово-хозяйственной деятельности;
- управление финансами;

- планирование и бюджетирование;
- осуществление платежей, привлечение финансирования, проведение финансовых операций;
- внутренний контроль;
- финансово-экономический анализ [29].

Служба по управлению персоналом занимается работой по комплектованию кадров предприятия руководит. А также организует работу на предприятии и проводит систему мероприятий, направленных на повышение квалификации работников.

Служба авиационной безопасности система органов безопасности аэропортов, осуществляющая в пределах своих полномочий решение задач по обеспечению авиационной безопасности

Инженерно-авиационная служба осуществляет: проверку содержания самолётов, вертолётов, планёров, авиационных двигателей, вооружения и оборудования в исправном и готовом к эксплуатации состоянии. Занимается также организацией и осуществлением технически правильного использования авиационной техники, руководит технической подготовкой лётного и инженерно-технического состава, научно-исследовательской работой по совершенствованию авиационной техники, разрабатывает руководства и наставления по техническому обеспечению, информирует командиров о состоянии авиационной техники, оказывает помощь руководителям полётов по руководству действиями лётного состава

Аэродромная служба осуществляет ряд задач:

- эксплуатационное содержание и текущий ремонт аэродромных покрытий, водоотводных и дренажных систем, грунтовой части летного поля, внутриаэропортовых дорог;
- разработка изменений к схемам руления и расстановки воздушных судов на перроне и местах стоянок;
- обеспечение информацией об ограничениях, действующих на аэродроме;

- внесение изменений в документы аэронавигационной информации;
- согласование и контроль за строительством объектов на аэродроме и приаэродромной территории;
- обеспечение контроля за работой сторонних организаций на аэродроме.

1.2 Краткая характеристика подразделения или видов его деятельности

В АО "Белгородавиа" программным обеспечением, которое позволяет производить бронирования и продажу авиабилетов занимаются два подразделения (службы): Кассы аэровокзала и служба информационных технологий и связи.

Кассы аэропорта являются непосредственно связующим звеном между пассажирами и рейсами, то есть они относятся к категории конечного пользователя. В кассах аэропорта организуется снятие с брони и продажа билетов пассажиру или просто продажа авиабилетов без предварительного бронирования.

Служба информационных технологий и связи занимается: предоставлением технической компьютерной поддержки, включая установку и техническое обслуживание серверов, персональных компьютеров и программного обеспечения; соблюдает стандарты в отношении компьютерного оборудования; аппаратным и техническим сопровождением конференций и совещаний; поддержкой инфраструктуры программ планирования общеорганизационных ресурсов и других систем автоматизации делопроизводства; налаживанием работы электронной почты и программное обеспечение для коллективной работы; обеспечением безопасности прикладных и серверных данных и защиты компьютеров от вирусов, хакеров и аналогичных угроз; обслуживанием сотовых телефонов,

коммутационных телефонных узлов и факсимильной связи, а также услуги по проведению аудио- и видеоконференций; обеспечением бесперебойной работы и аварийного восстановления данных. В данной работе нас интересует техническое обслуживание программного обеспечения, так как служба информационных технологий и связи занимается установкой, дальнейшей настройкой и изменением программного обеспечения необходимого для бронирования и продажи авиабилетов.

1.3 Анализ существующих подсистем бронирования и продажи авиабилетов

В современной России отсутствуют подобные подсистемы. Отечественные авиакассы используют следующие системы бронирования.

1) Компьютерная система бронирования “Amadeus”, представленная на российском рынке с 1993 г. Вместе с увеличением интенсивности резервирования растет и число агентств - пользователей системы. Amadeus имеет доступ к ресурсам более 780 зарубежных и 12 российских авиакомпаний

В России, работающих с системой Amadeus;

- Более 800 терминалов, подключенных к системе Amadeus в России;
- Самые низкие цены на подключение;
- Гибкая система оплаты.
- Программные продукты, работающие в операционной среде Windows;
- Автоматическое заполнение и печать бланков ручной выписки билетов на все основные авиакомпании, представленные на российском рынке;
- Доступ к конфиденциальным тарифам, вводимым крупнейшими авиакомпаниями для своих агентов самостоятельно;

- Возможность создания собственной базы данных конфиденциальных тарифов, полученных от авиакомпаний;
- Возможность отправки информации (расписание, подтверждение бронирования, маршрут, счет для оплаты и т.д.) непосредственно из системы.
- Выбор наиболее оптимального способа подключения, конфигурации.

2) Система GALILEO - это целый комплекс встроенных подсистем, каждая из которых предназначена для выполнения задачи получения полной информации и обеспечения простого доступа к ресурсам 527 авиакомпаний.

Galileo Availability - обеспечивает доступ к информации о наличии свободных мест на рейсы более чем 500 авиакомпаний, при этом имеется возможность обращаться непосредственно к базам данных большинства авиакомпаний, а по отдельным авиакомпаниям выбирать свободные места вплоть до последнего на момент запроса;

Galileo Sell - позволяет бронировать места из экрана наличия мест, при этом агент может за один запрос осуществить бронирование мест на рейсы любых типов (прямые, стыковочные и с любым набором промежуточных посадок) по необходимому маршруту;

Galileo Advance Seat Reservation - позволяет осуществлять резервирование конкретного места (с номером) в салоне самолета;

Galileo Enhanced Booking File Servicing - позволяет формировать запросы в системы бронирования авиакомпаний в едином формате;

3) Worldspan работает в России с 1994 года. Представители компании приложили много сил, чтобы практически сразу обеспечить расширенные возможности клиентам по бронированию рейсов авиакомпаний Аэрофлот и Трансаэро и прилагает большие усилия для привлечения авиакомпаний, выполняющих внутренние перевозки.

Из представленных в системе авиакомпаний, более 200 бронируются в режиме прямого доступа, т.е. дают возможность пользователю практически эмулировать экран той или иной авиакомпании на своем компьютере. Для

обеспечения наиболее полной и точной информации по тарифам система Worldspan использует две тарифные базы данных, оригинальную, наиболее полно перекрывающую американский континент, а также SITA Airfare, которую традиционно пользуются перевозчики в остальном мире [27].

1.4 Обоснование необходимости автоматизации и цели создания подсистемы бронирования и продажи авиабилетов

Необходимость разработки обусловлена тем, что на данный момент российские авиаперевозчики используют в основном иностранные сервисы бронирования и продажи авиабилетов. Все сведения о пассажирах и их перелётах хранятся на серверах за границей. Поэтому необходимо создать отечественный программный продукт, который будет использоваться на территории РФ.

Автоматизированная подсистема бронирования и продажи авиабилетов создается с целью:

- уменьшения времени на поиск из списка рейсов;
- снижение стоимости авиабилетов;
- системности в организации архива работ;
- организации рабочего процесса;
- организации процесса взаимодействия между клиентом и сотрудниками;

Так же подразумевается создание соответствующего программного средства для упрощения взаимодействия сотрудников и облегчения выполнения их работы.

Перед разработкой подсистемы заказчик и разработчик должны ясно представлять, какие функциональные возможности будут заложены в подсистему, и как будет организовано функциональное взаимодействие внутри подсистемы.

1.5 Разработка структурно функциональной диаграммы («КАК ЕСТЬ») по методологии SADT(IDEF0)

Построение функциональных моделей должно решить большую часть проблем рассмотренных в предыдущей главе. Построение происходит путём разбиения контекстной диаграммы на функциональные подсистемы, которые делятся на подфункции, а те, в свою очередь, на конкретные процедуры. Для этих целей используем известные методологии IDEF0, IDEF3 и DFD.

Кроме того, данные модели помогают понять, где находятся слабые места, в чем будут состоять преимущества новых процессов и насколько измениться эффективность деятельности работы авиакассы.

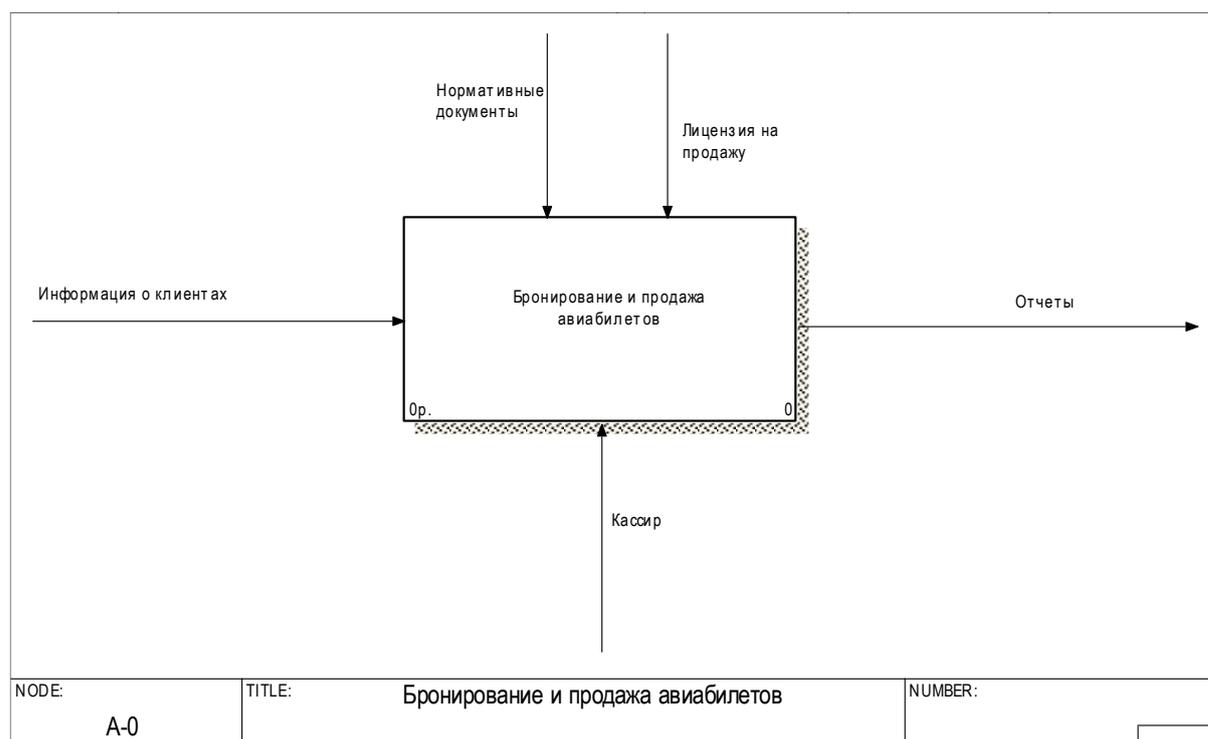


Рисунок 1.1 - Контекстная диаграмма («КАК ЕСТЬ»)

Далее необходимо детализировать деятельность подсистемы, то есть создать диаграмму-декомпозицию.

Декомпозируем контекстную диаграмму на 4 функциональных блока (см. рисунок 1.2):

- «Работа в авиакассе»;
- «Работа с бронью»;
- «Создание заказа»;
- «Составление отчета»;

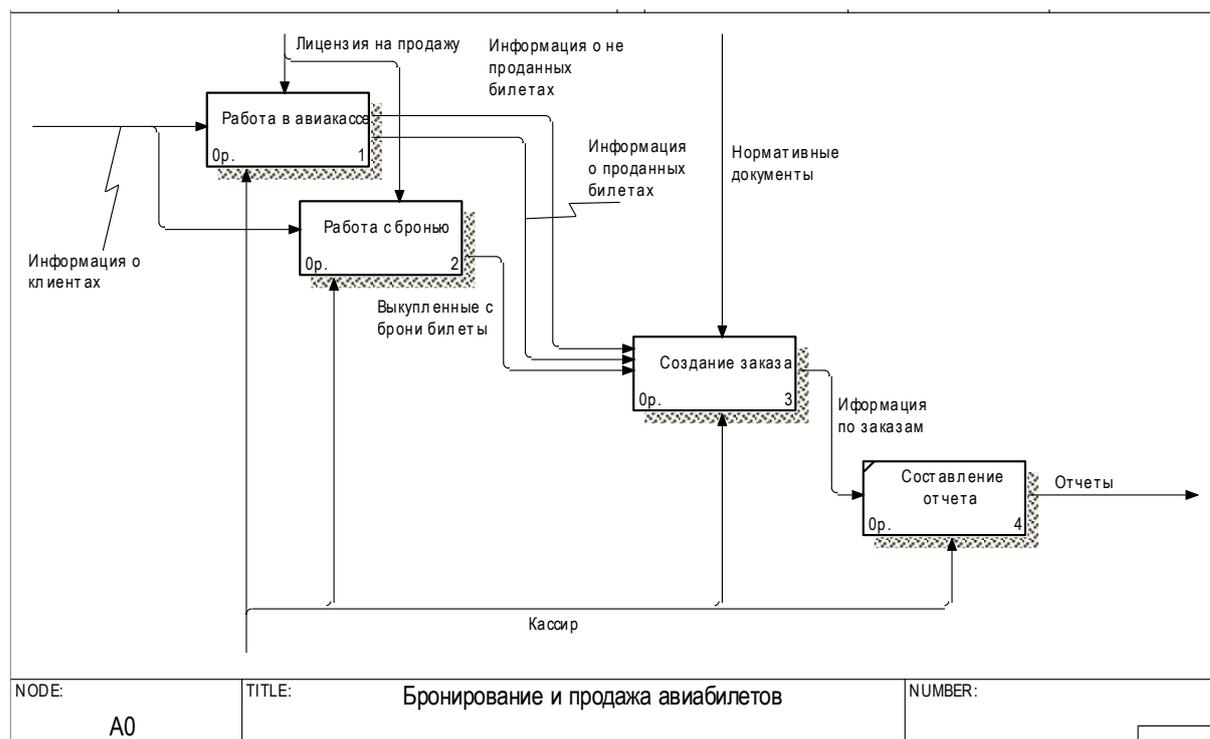


Рисунок 1.1 - Диаграмма декомпозиции

Прежде всего, разрабатываемая подсистема нацелена на автоматизацию деятельности отдела по работе с клиентами. Подсистема должна обеспечивать бронирование и продажу авиабилетов.

Далее каждый из процессов был также детализирован, используя описанные выше методологии. В первую очередь необходимо рассмотреть работу в авиакассе:

Декомпозируем функциональный блок «Работа в авиакассе» еще на 4 действия (см. рисунок 1.3):

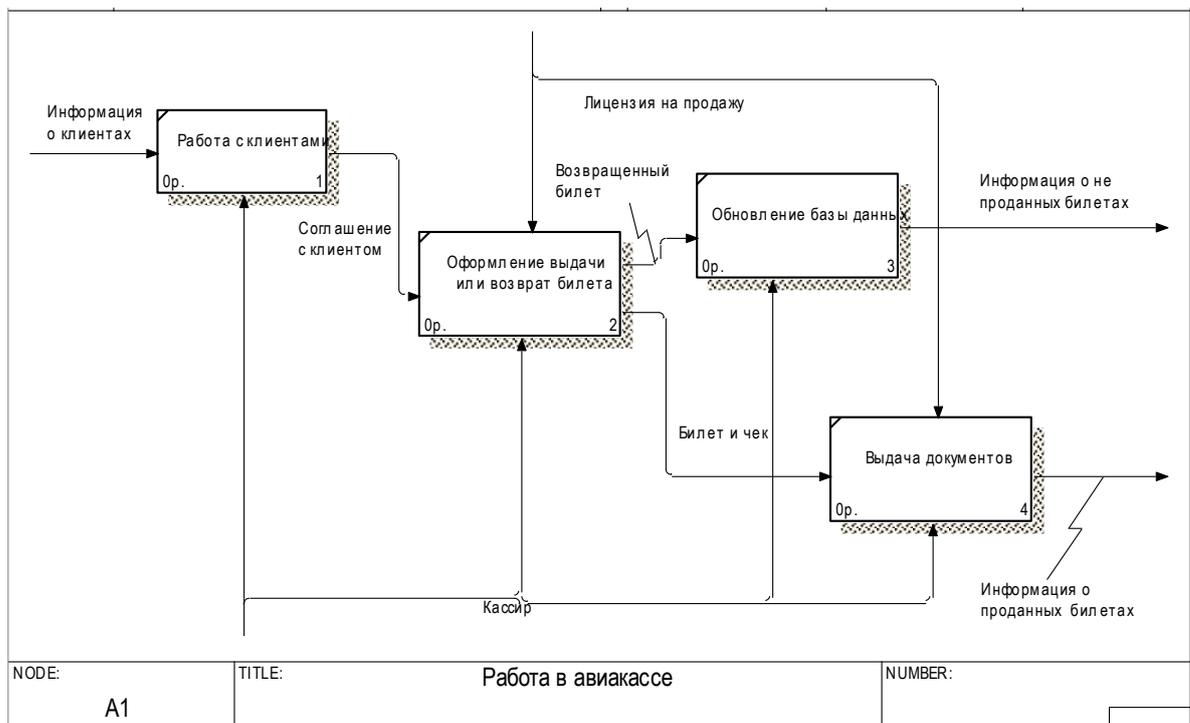


Рисунок 1.2 - Декомпозиция "Работа в авиакассе"

Таким же образом детализируем процессы «Работа с бронью» и «Создание заказа» (см. рисунок 1.4, 1.5), используя методологию IDEF0, которая используется для создания функциональных моделей и рассматривает отношения между всеми видами работ.

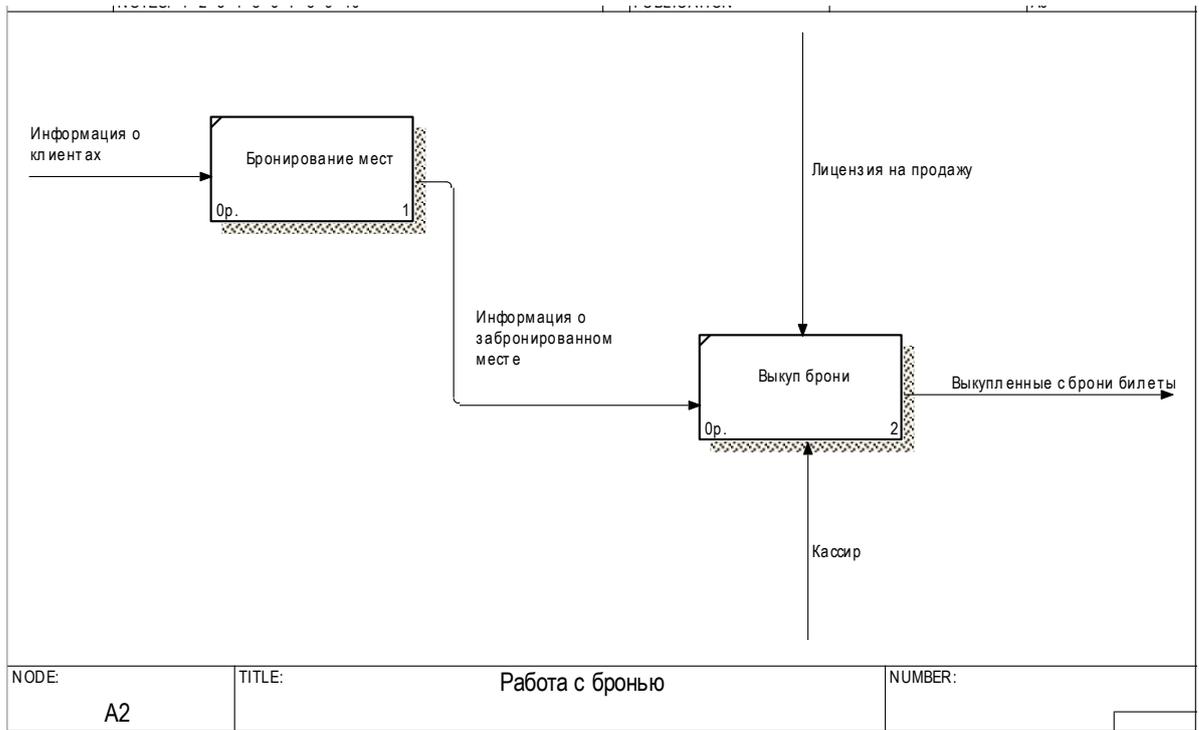


Рисунок 1.3 - Декомпозиция "Работа с бронью"

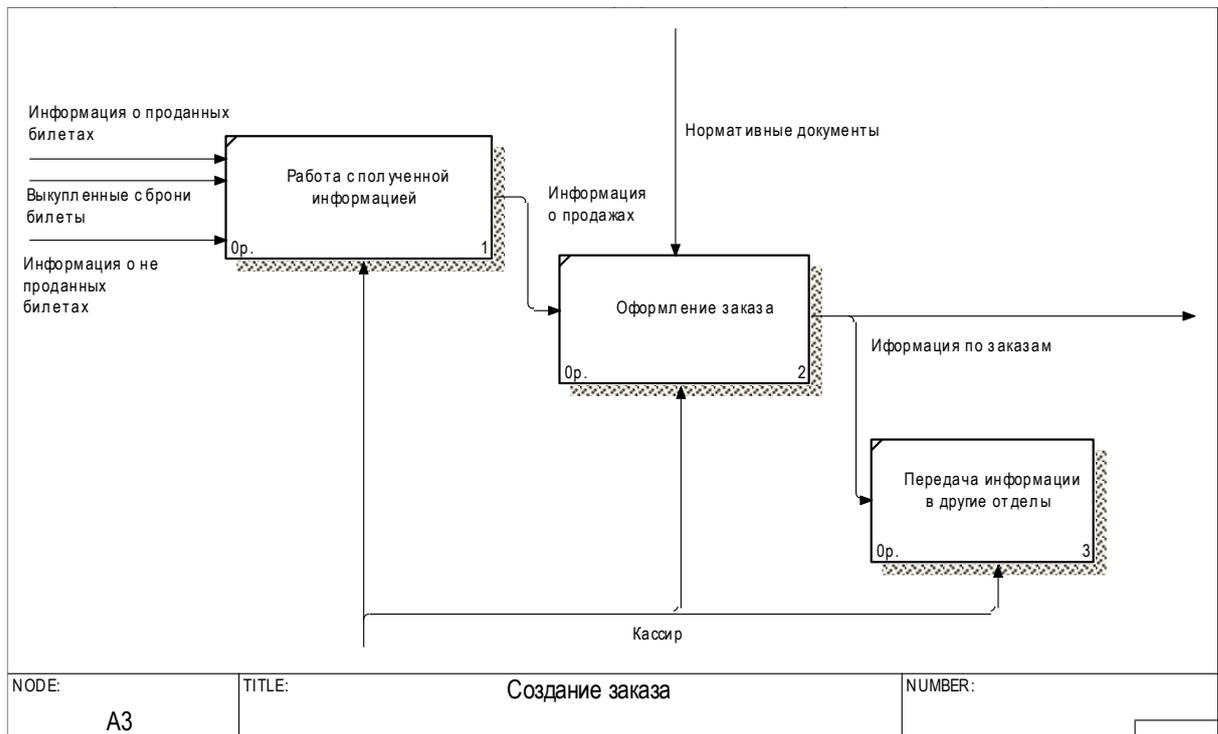


Рисунок 1.4 - Декомпозиция "Создание заказа"

Создаем диаграмму дерева узлов. Диаграмма дерева узлов показывает иерархию работ в модели и позволяет рассмотреть всю модель целиком, но не показывает взаимосвязи между работами

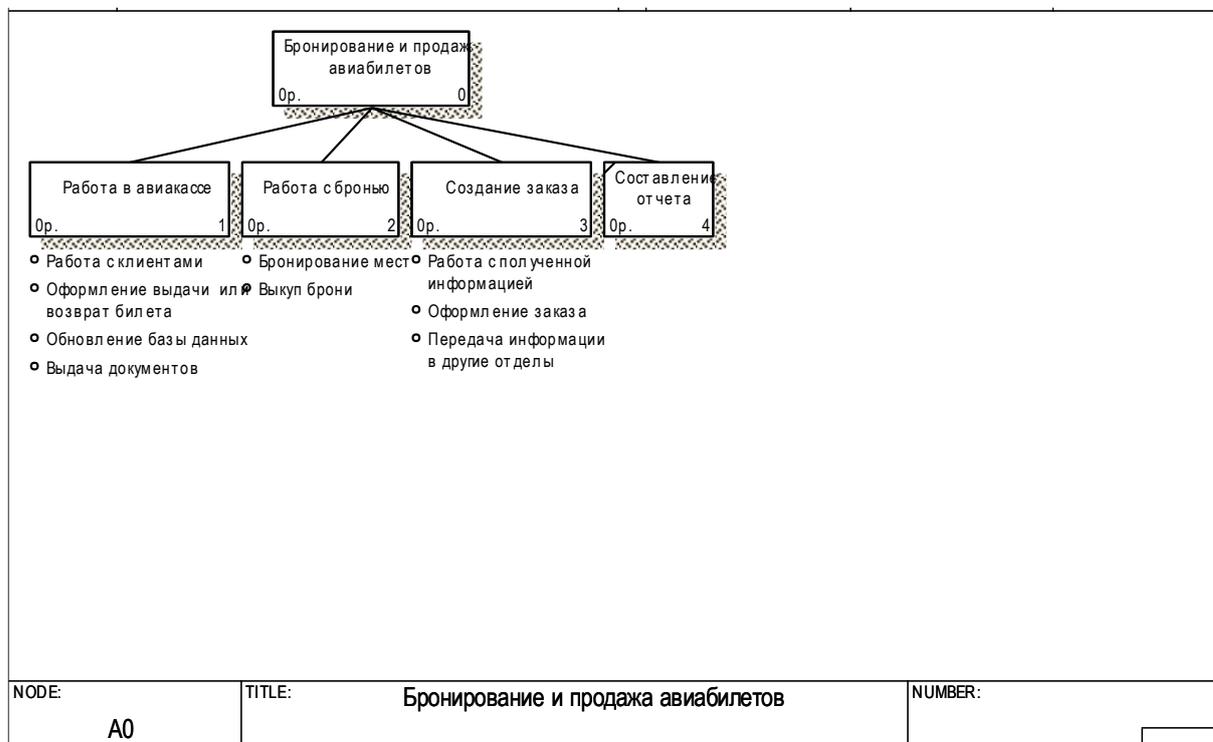


Рисунок 1.6 - Диаграмма узлов подсистемы «Бронирования и продажи авиабилетов»

1.6 Постановка задачи

Основная задача разработки выпускной квалификационной работы состоит в создании подсистемы, которая будет работать в АО «Белгородавиа», и возможно в других отечественных аэропортах, как основная подсистема авиакассы по бронированию и продаже авиабилетов. Тип проектируемой подсистемы: это информационная подсистема решений поставленной задачи, которая будет рассчитывать необходимые для подсистемы решения.

Под разработкой подсистемы подразумевается решение следующих задач:

- Построение единой информационной подсистемы для работы отечественных авиакассы, которая будет включать в себя взаимосвязанные между собой функции (бронирование и продажу);

- Разработка задачи бронирования авиабилета;

- Разработка задачи продажи авиабилета;

- Ведения отчетности по рейсам, клиентам, сотрудникам и т.д.

В результате создания подсистемы должны быть улучшены значения следующих показателей:

- числа будущих клиентов, за счет уменьшения цены на билет;

- времени доступа к списку авиабилетов и другой информации;

- повышения качества (полноты, точности, достоверности, своевременности, согласованности) информации;

- времени взаимодействия сотрудников с клиентами;

- уменьшение затрат на ПО для работы на авиакассе.

Если будут соблюдены данные требования, это позволит говорить о том, что подсистема будет полностью работоспособной и готовой для эксплуатации.

2 ОБОСНОВАНИЕ ПРОЕКТНЫХ РЕШЕНИЙ

2.1 Обоснование проектных решений по техническому обеспечению

Техническое обеспечение – совокупность технических средств (технические средства обработки, хранения, передачи, обмена), предназначенных для создания необходимой автоматизированной подсистемы. Так же к техническому обеспечению относятся различные методические материалы, техническая документация и технический персонал отвечающий за работу подсистемы

Технические средства обработки данных, программное обеспечение и организация БД в совокупности определяют информационно-технологическую архитектуру ИС (ИТА). Различают следующие типы ИТА:

1) централизованная - хранение и обработка данных на центральном компьютере, удобство администрирования ИС. Недостатки: ограничение на рост объемов хранимых данных, увеличение производительности ИС, высокий уровень риска неработоспособности ИС;

2) Система телеобработки данных - наиболее дешевый способ организации одновременной работы большого числа пользователей при использовании мощного центрального компьютера. Высокопроизводительные каналы телекоммуникации позволяют не зависеть от места обработки или хранения данных;

3) Многомашинный комплекс - интеграция нескольких компьютеров, расположенных в непосредственной близости друг от друга, в один "объединенный" компьютер; возможность эффективного выполнения сложных вычислений, повышение надежности ИС, рост объемов хранимых данных, но сохранения централизованного характера хранения и обработки данных и программ, зависимости пользователей от места обработки данных;

4) Телекоммуникационная ИГА - наиболее распространенный вариант построения системы обработки данных для крупномасштабных ИС на базе компьютерных сетей (КС) и их ассоциации. Поддержка программных и технических интерфейсов осуществляется в соответствии со стандартами OSI (Open System Interconnection) [26].

В данной работе организацией является АО “Белгородавиа”, поэтому в этом случае используется многомашинный комплекс. Чтобы организовать связь между большим количеством вычислительной техники используют клиент серверную технологию. Данная технология работает по принципу разделения сетевой нагрузки между клиентом (заказчиком услуг) и сервером (поставщиком услуг). Клиент и сервер взаимодействуют с помощью сетевых протоколов и находятся на разных вычислительных машинах. Клиент посылает запрос на сервер на предоставление ресурса, сервер принимает запрос и посылает клиенту необходимую информацию.

Основные технические характеристики клиентской части для создания подсистемы представлены в таблице 2.1.

Таблица 2.1 - Типовая конфигурация клиента.

<i>Наименование</i>	<i>Характеристика</i>
Процессор	Intel® Core™2 Duo
Жесткие диски	Tosiba MK2555GSX
Модули памяти. Flash-накопители	silicon power 8gb
Видеокарты	ASUS nvidia geforce g105m
Звуковые карты	NVIDIA High Definition Audio
Приводы CD-R/RW DVD-ROM/RW	TSSTcorp CDDVDW TS-L633B
Модемы	Asus RT-G32
Мышь оптическая.	Defender Datum MM-075
Сетевой фильтр	Сетевой фильтр (3 розетки) Pilot GL 5м
Монитор	Универсальный монитор 17”Samsung

Минимальные технические характеристики серверной части, используемой для разработки подсистемы:

Сервер базы данных развернут на HP9000 SuperDome №1, минимальная конфигурация которого: CPU: 16 (32 core); RAM: 128 Gb; HDD: 500 Gb; Network Card: 2 (2 Gbit); Fiber Channel: 4.

Сервер сбора, обработки и загрузки данных развернут на HP9000 SuperDome №2, минимальная конфигурация которого:

CPU: 8 (16 core); RAM: 32 Gb; HDD: 100 Gb; Network Card: 2 (1 Gbit); Fiber Channel: 2.

Сервер приложений развернут на платформе HP Integrity, минимальная конфигурация которого: CPU: 6 (12 core); RAM: 64 Gb; HDD: 300 Gb; Network Card: 3 (1 Gbit) [31].

Приведенные сервера подключены к дисковому массиву HP XP с организацией сети хранения данных. Минимальный объем свободного пространства для хранения данных на дисковом массиве составляет 5 Тб.

2.2 Обоснование проектных решений по информационному обеспечению

Под информационным обеспечением понимается предоставление информационных ресурсов, которые относятся к определенному объекту или субъекту, в данной работе этим объектом является подсистема бронирования и продажи авиабилетов

Приводятся требования:

- 1) к составу, структуре и способам организации данных в системе;
- 2) к информационной совместимости со смежными системами;
- 3) по использованию общесоюзных и зарегистрированных республиканских, отраслевых классификаторов, унифицированных документов и классификаторов, действующих на данном предприятии;

- 4) по применению систем управления базами данных;
 - 5) к структуре процесса сбора, обработки, передачи данных в системе и представлению данных;
 - 6) к защите данных от разрушений при авариях и сбоях в электропитании системы;
 - 7) к контролю, хранению, обновлению и восстановлению данных [31].
- Так же при разработке подсистемы будет использоваться такое понятие как база данных.

База данных представляет собой совокупность некоторых данных, которые взаимосвязаны и над которыми происходит процесс обработки и поиска на определенной вычислительной машине.

На рисунке 2.1 показана модель данных разрабатываемой подсистемы.

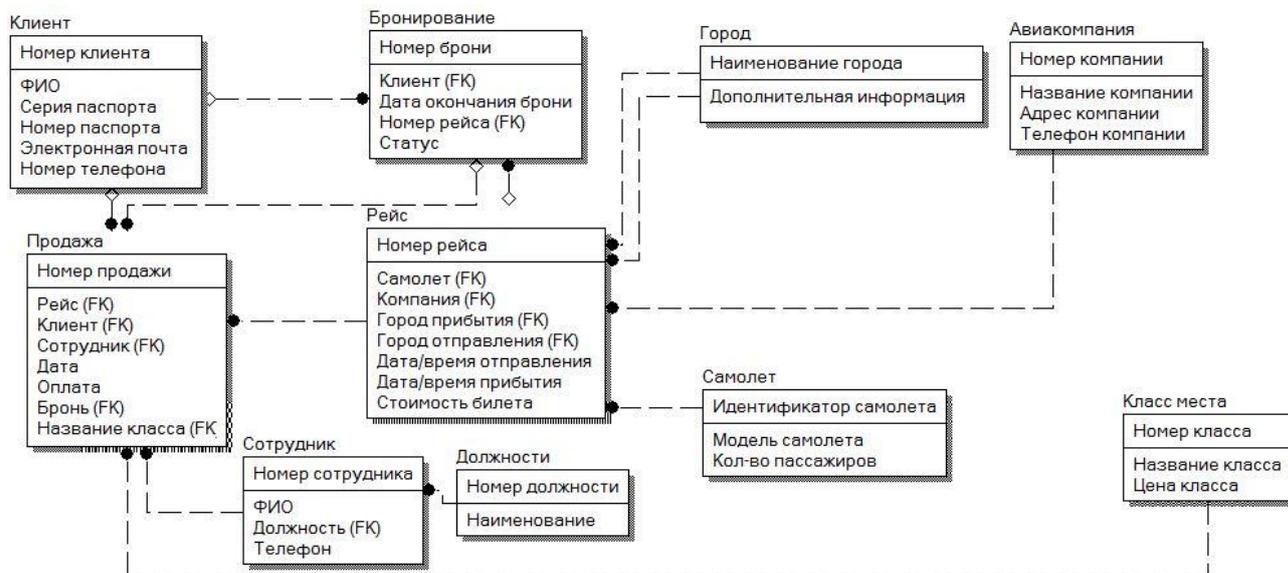


Рисунок 2.1 - Модель данных

Основные объекты системы:

- Самолет – объект, справочный элемент системы, предназначенный для хранения данных о самолетах находящихся в распоряжении предприятия;

- Класс места - объект, справочный элемент системы о названиях классов в самолете и ценах на эти классы;
- Авиакомпания - объект, справочный элемент системы, предназначенный для хранения данных о авиакомпаниях, выполняющих рейсы;
- Клиент - объект, соответствующий реальному клиенту, приходящему в авиакассу.
- Сотрудники - объект, соответствующий реальному лицу предприятия.
- Должности - объект, справочный элемент системы, предназначенный для хранения данных о должностях предприятия;
- Город - объект, справочный элемент системы, предназначенный для хранения данных о городах отправления.
- Продажа - объект, справочный элемент системы, предназначенный для хранения данных о проданных билетах.
- Рейс - справочный элемент системы, предназначенный для хранения данных о рейсах.
- Бронирование - справочный элемент системы, предназначенный для хранения данных о забронированных авиабилетах.

Как видно из рисунка все таблицы связаны между собой отношением «один-ко-многим». Это отношение связывает одну строку первой таблицы с несколькими записями второй с помощью первичного или уникального ключа базовой таблицы и соответствующего ему внешнего ключа связанной таблицы. Также видно, что все связи являются не идентифицирующей, так как ни один из первичных ключей дочерней таблицы не содержит внешний ключ родительской таблицы. Каждую запись таблицы можно однозначно идентифицировать по первичному ключу таблицы.

На основе построенной модели данных строится инфологическая модель данных. Инфологическая модель отражает логические связи между элементами данных вне зависимости от их содержания и среды хранения. По сути

инфологическая модель отражает существующую схему взаимосвязи таблиц на языке определенной СУБД (Система управления базами данных). На рисунке 2.2 представлена инфологическая модель разрабатываемой подсистемы.

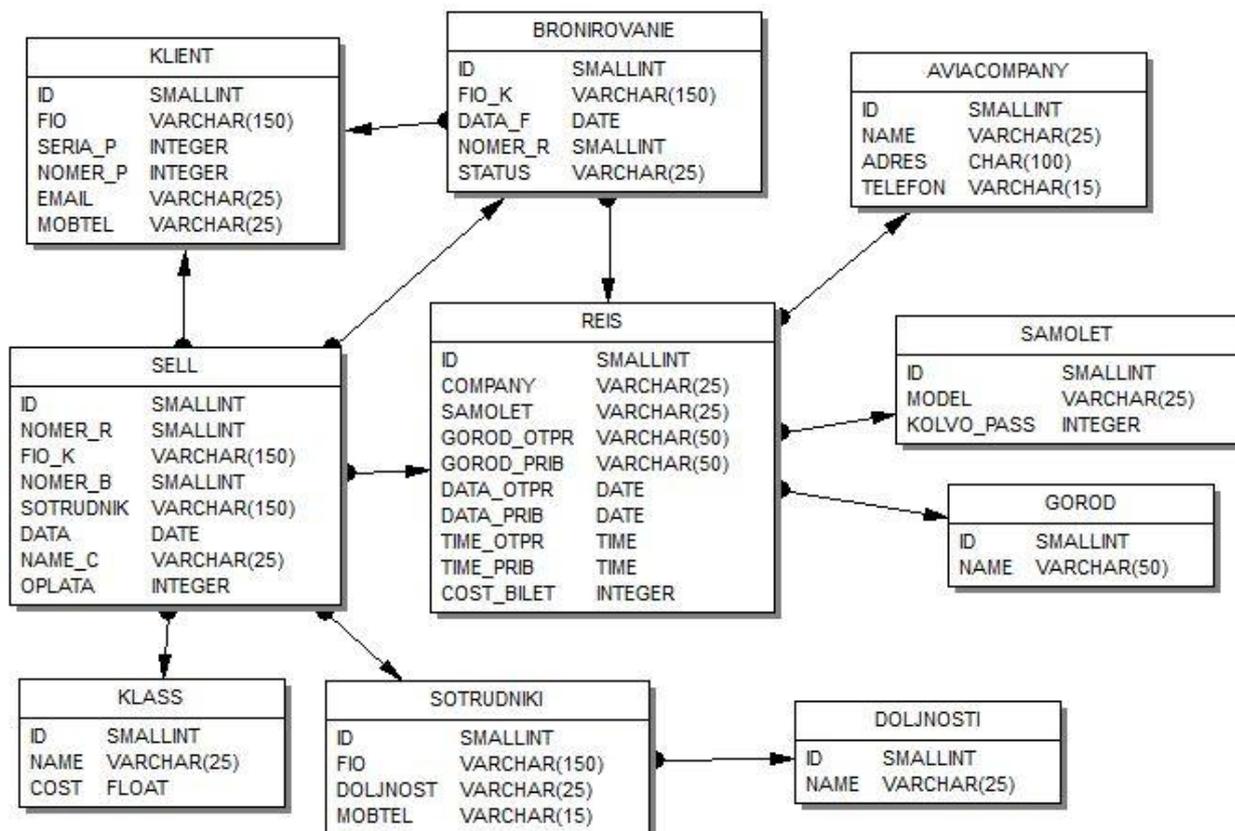


Рисунок 2.2 - Инфологическая модель БД

В таблице 2.2 приведена даталогическая модель подсистемы.

Таблица 2.2 – Таблица даталогической модели.

Сущность	Идентификатор таблицы	Атрибут	Идентификатор поля	Тип поля
1	2	3	4	5
Клиент	KLIENT	Код	ID	SMALLINT
		ФИО	FIO	VARCHAR
		Серия паспорта	SERIA_P	INTEGER
		Номер паспорта	NOMER_P	INTEGER
		EMAIL	EMAIL	VARCHAR
		Мобильный телефон	MOBTEL	VARCHAR

Продолжение таблицы 2.2.

1	2	3	4	5
Продажа	CELL	Код	ID	SMALLINT
		Номер рейса	NOMER_R	SMALLINT
		ФИО клиента	FIO_K	VARCHAR
		Номер брони	NOMER_B	SMALLINT
		ФИО сотрудника	SOTRUDNIK	VARCHAR
		Дата продажи	DATA	DATE
		Название класса	NAME_C	VARCHAR
		Оплата	OPLATA	INTEGER
Класс мест	KLASS	Код	ID	SMALLINT
		Название класса	NAME	VARCHAR
		Коэффициент наценки	COST	FLOAT
Рейс	REIS	Код	ID	SMALLINT
		Авиакомпания	COMPANY	VARCHAR
		Самолет	SAMOLET	VARCHAR
		Город отправления	GOROD_OTPR	VARCHAR
		Город прибытия	GOROD_PRIB	VARCHAR
		Дата отправления	DATA_OTPR	DATE
		Дата прибытия	DATA_PRIB	DATE
		Время отправления	TIME_OTPR	TIME
		Время прибытия	TIME_PRIB	TIME
		Стоимость билета	COST_BILET	INTEGER
Бронирование	BRONIROVANIE	Код	ID	SMALLINT
		ФИО Клиента	FIO_K	VARCHAR
		Дата окончания брони	DATA_F	DATE
		Номер рейса	NOMER_R	SMALLINT
		Статус оплаты	STATUS	VARCHAR
Авиакомпания	AVIAKOMPANY	Код	ID	SMALLINT
		Название	NAME	VARCHAR
		Адрес	ADRES	CHAR
		Телефон	TELEFON	VARCHAR
Сотрудники	SOTRUDNIKI	Код	ID	SMALLINT
		ФИО сотрудника	FIO	VARCHAR
		Должность	DOLJNOST	VARCHAR
		Телефон	MOBTEL	VARCHAR
Самолет	SAMOLET	Код	ID	SMALLINT
		Модель самолета	MODEL	VARCHAR
		Кол-во пассажиров	KOL-VO_PASS	INTEGER
Город	GOROD	Код	ID	SMALLINT
		Название	NAME	VARCHAR
Должности	DOLJNOSTI	Код	ID	SMALLINT
		Название	NAME	VARCHAR

2.3 Обоснование проектных решений по программному обеспечению

Комплекс программ, правил, процедур, используемых при эксплуатации и вся документация, предназначенная для работы системы, называется программным обеспечением.

Все программное обеспечение по назначению делится на 3 типа:

Системное ПО - совокупность программ, которые обеспечивают весь рабочий цикл вычислительных машин и работу сети.

Прикладное ПО – совокупность программ, используемых на компьютере для решения различных задач.

Инструментальное ПО – это совокупность программ, используемых для разработки системного и прикладного программного обеспечения.

Для создания подсистемы в ВКР будет использовано следующее ПО:

К системному ПО, которое мы будем использовать для данной разработки относится:

- операционная система - Microsoft Windows 7 x64;
- стандартные программы;
- служебные программы;
- администрирование и дополнительные средства.

Операционная система – это набор взаимосвязанных программ, с помощью которых происходит управление ресурсами и взаимодействие с пользователем. При разработке использовалась операционная система Microsoft Windows 7.

Так же были использованы стандартные программы типа:

Paint — программа для создания и редактирования изображений

Калькулятор — программа, служащая для выполнения различных математических расчетов.

Проводник — это программа с помощью которой происходит переход между файлами и папками.

К прикладному ПО, используемому в данной разработке относятся:

Microsoft Office Word 2010 – программное обеспечение, служащее для создания и редактирования документов различного формата.

Microsoft Power Point 2010 – программное обеспечение для создания и редактирования презентации.

Иbexpert – программное обеспечение для создания и редактирования баз данных firebird, так же с помощью данной программы можно выбирать и изменять данные, которые хранятся в базе.

Firebird – программное обеспечение для управления базами данных

К инструментальному ПО, относятся:

All fusion Data Modeler – это программное обеспечение, предназначенное для проектирования баз данных.

Для создание готового программного обеспечения была использована программа C++ Builder 6. C++ Builder предназначена для быстрой разработки приложений на языке с++. Данное программное обеспечение содержит в себе набор инструментов, различных библиотек и.т.д. с помощью которых получается качественная виртуальная разработка.

Так же для разработки были использовано программное обеспечение, которое приведено в таблице 2.3.

Таблица 2.3 - Использованное ПО

<i>Наименование</i>	<i>Характеристика</i>
Файловые менеджеры	Far manager. Windows Commander
Антивирусные пакеты	AVG
Архиваторы	WinRAR, WinZIP, UltraISO
Информационно-поисковые системы	Google.ru, Yandex.ru

3 ПРОЕКТНАЯ ЧАСТЬ

3.1 Разработка структурно функциональной диаграммы («КАК ДОЛЖНО БЫТЬ») по методологии SADT(IDEF0)

Под информационным обеспечением решения поставленной задачи понимается создание новой функциональной модели, другими словами модели «КАК ДОЛЖНО БЫТЬ». Она служит наглядным примером того, как изменится существующая организация информационной системы, после разработки подсистемы.

Данная функциональная модель была разработана с помощью методологии IDEF0 и представлена на рисунке 3.1.

Входными ресурсами являются:

«Информация о клиентах» - информация, необходимая при создании документов бронирования и продажи авиабилетов, а также для заполнения справочника о клиентах;

Выходные ресурсы:

«Отчеты» - аналитическая информация, представленная в удобном виде.

Механизмом управления является кассир и созданная информационная подсистема.

Правилами управления включают в себя: нормативные документы и лицензию на продажу авиабилетов.

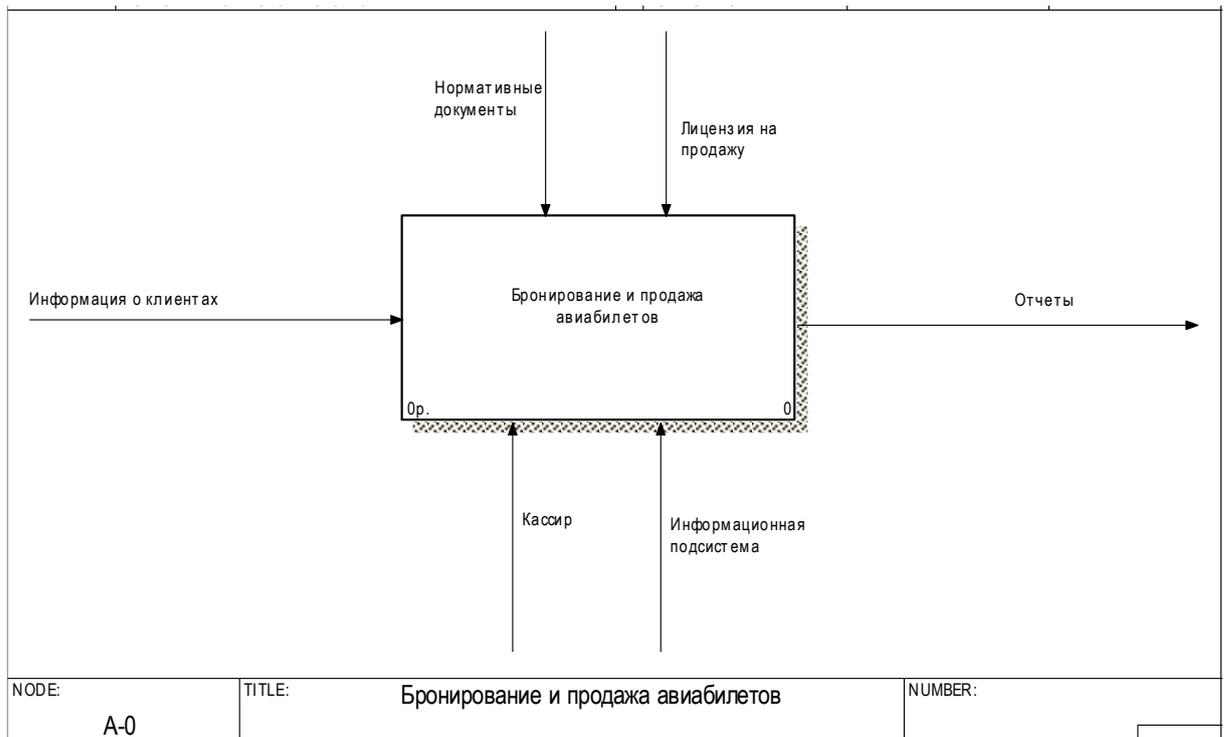


Рисунок 3.1 - Контекстная диаграмма («КАК ДОЛЖНО БЫТЬ»)

Декомпозировали контекстную диаграмму («КАК ДОЛЖНО БЫТЬ») (см. рисунок 3.2).

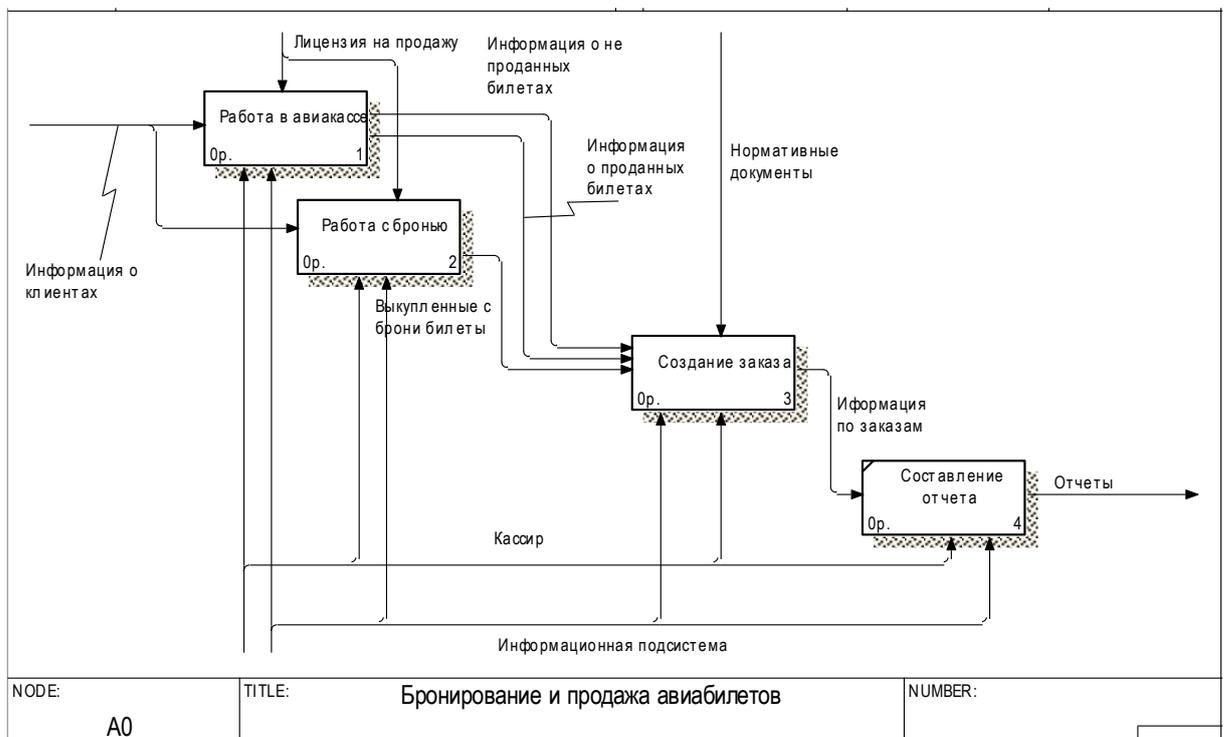


Рисунок 3.2 - Декомпозиция контекстной диаграммы

Декомпозировали функциональный блок «Работа в авиакассе» еще на 4 действия (см. рисунок 3.3):

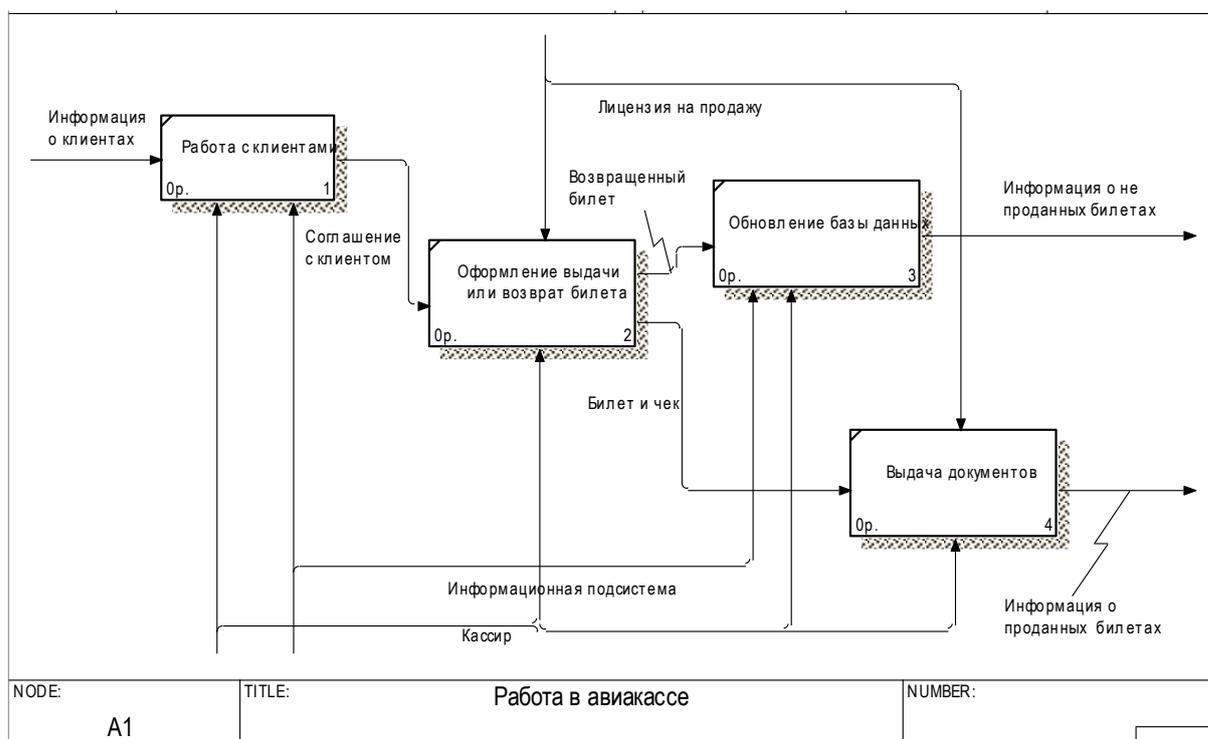


Рисунок 3.5 - Декомпозиция "Работа в авиакассе"

Таким же образом детализировали процессы «Работа с бронью» и «Создание заказа» (см. рисунок 3.4, 3.5), используя методологию IDEF0, которая используется для создания функциональных моделей и рассматривает отношения между всеми видами работ.

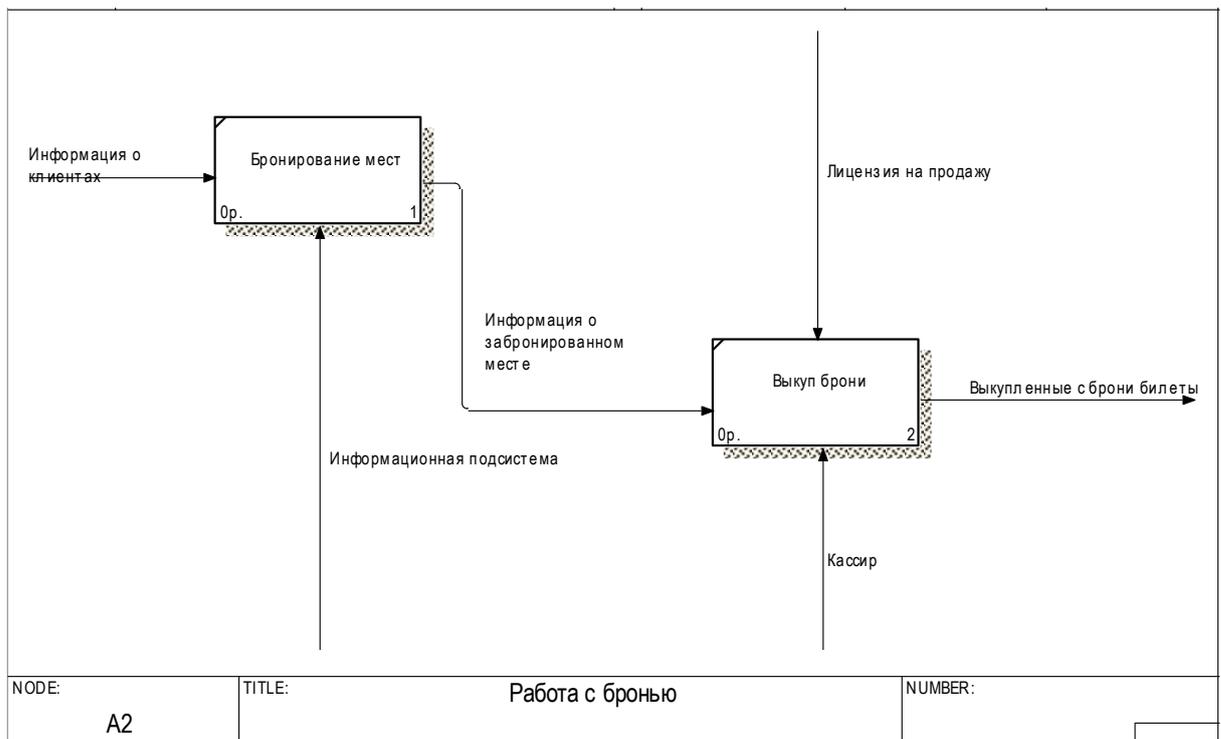


Рисунок 3.6 - Декомпозиция "Работа с бронью"

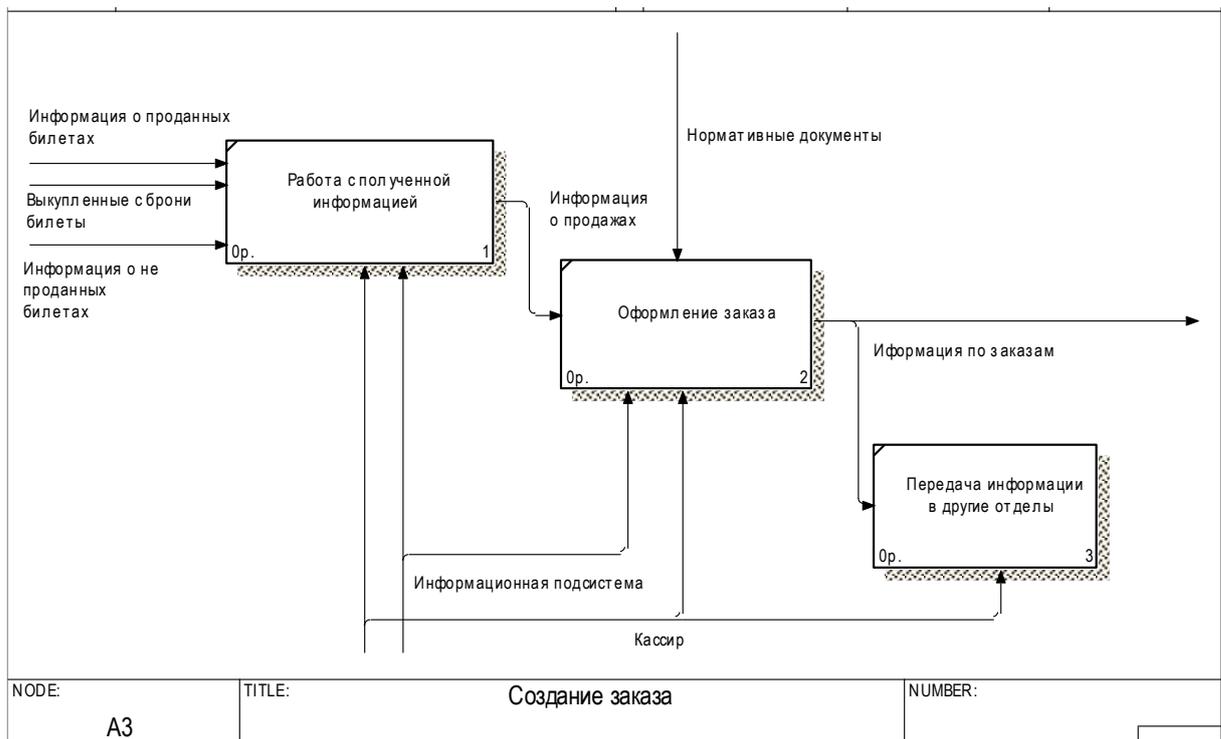


Рисунок 3.7 - Декомпозиция "Создание заказа"

На рисунке 3.3 показана работа кассира в авиакассе. В начале, приходит клиент с просьбой купить или отдать билет на рейс. При продаже

кассир заполняет всю необходимую информацию о клиенте, вносит эти данные в базу данных и в базу по рейсам. После заполненных данных кассир выдает клиенту билет и чек о проданном билете. В случае если клиент пришел в авиакасса за возвратом билета и получением денег за этот билет, то кассир возвращает всю необходимую сумму и удаляет клиента из базы данных о рейсе и обновляет данную базу. На рисунке 3.4 показан процесс бронирования билета. Клиент звонит на кассу, и кассир вносит данные по брони. Затем если клиент успевает до окончания даты бронирования выкупить билет, то кассир вносит данные клиента в рейс, если же этого не происходит, то кассир ничего не предпринимает, просто удаляет клиента из базы брони и тем самым освобождает место на рейсе. На рисунке 3.5 показан процесс создания заказа кассира на рейс. Кассир суммирует полученные данные о проданных и не проданных билетах, так же смотрит остаток мест на рейсе и оформляет все необходимые документы для предоставления их в другие отделы предприятия. На основе полученных заказов в дальнейшем формируются отчеты о проделанной работе и с помощью них предприятия производят свои финансовые расчеты о прибыли полученной от продаж авиабилетов.

3.2 Разработка программных модулей

Разработка подсистему бронирования и продажи авиабилетов была создана с помощью продукта Borland C++ Builder 6. Вся работа была разделена на три этапа: на первом этапе мы создавали базу данных в Ibxpert, вторым этапом был графический интерфейс пользователя, а на третьем этапе все компоненты графического интерфейса были взаимосвязаны и в результате был получен готовый программный продукт.

В самом начале работы создали базу данных с помощью программы Ibxpert, основное меню программы представлена на рисунке 3.6.

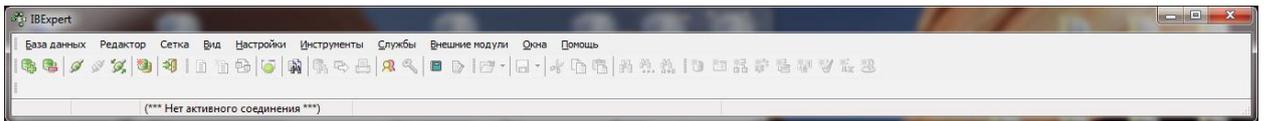


Рисунок 3.6 – Основное меню Ibeexpert.

Открыли вкладку База данных→ Создание базы и внесли данные для создания. Данные о базе представлены на рисунке 3.7.

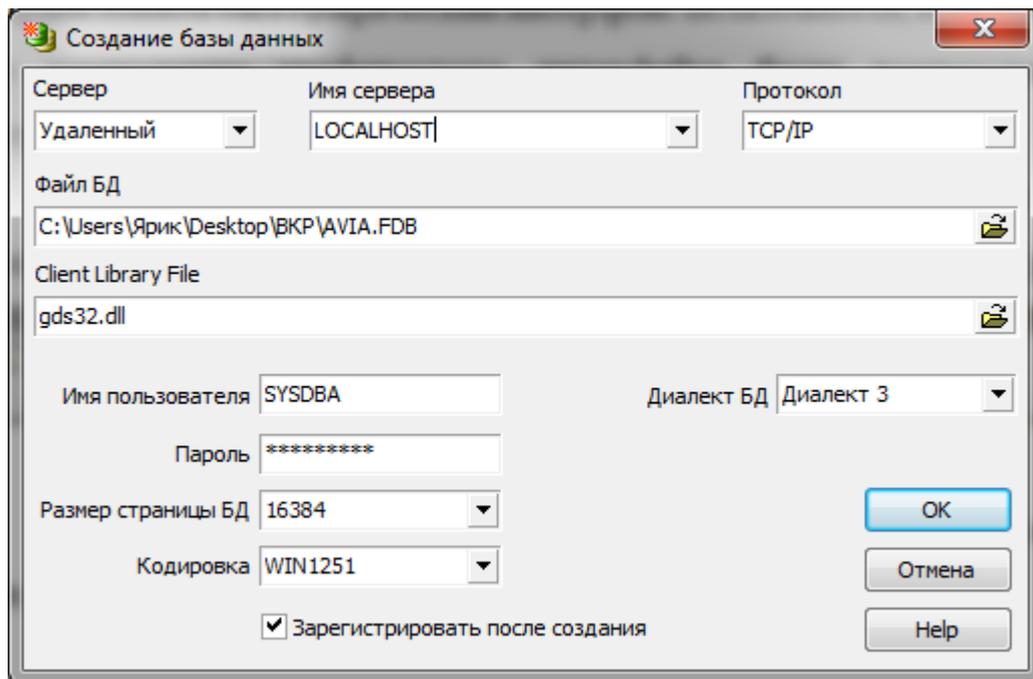


Рисунок 3.7 – Создание базы данных

После этого внесли регистрационную информацию базы. Указали сервер, имя сервера, протокол и версию сервера (см. рисунок 3.8). Нажали кнопку «ок». Быдаа создана база данных необходимая для разработки.

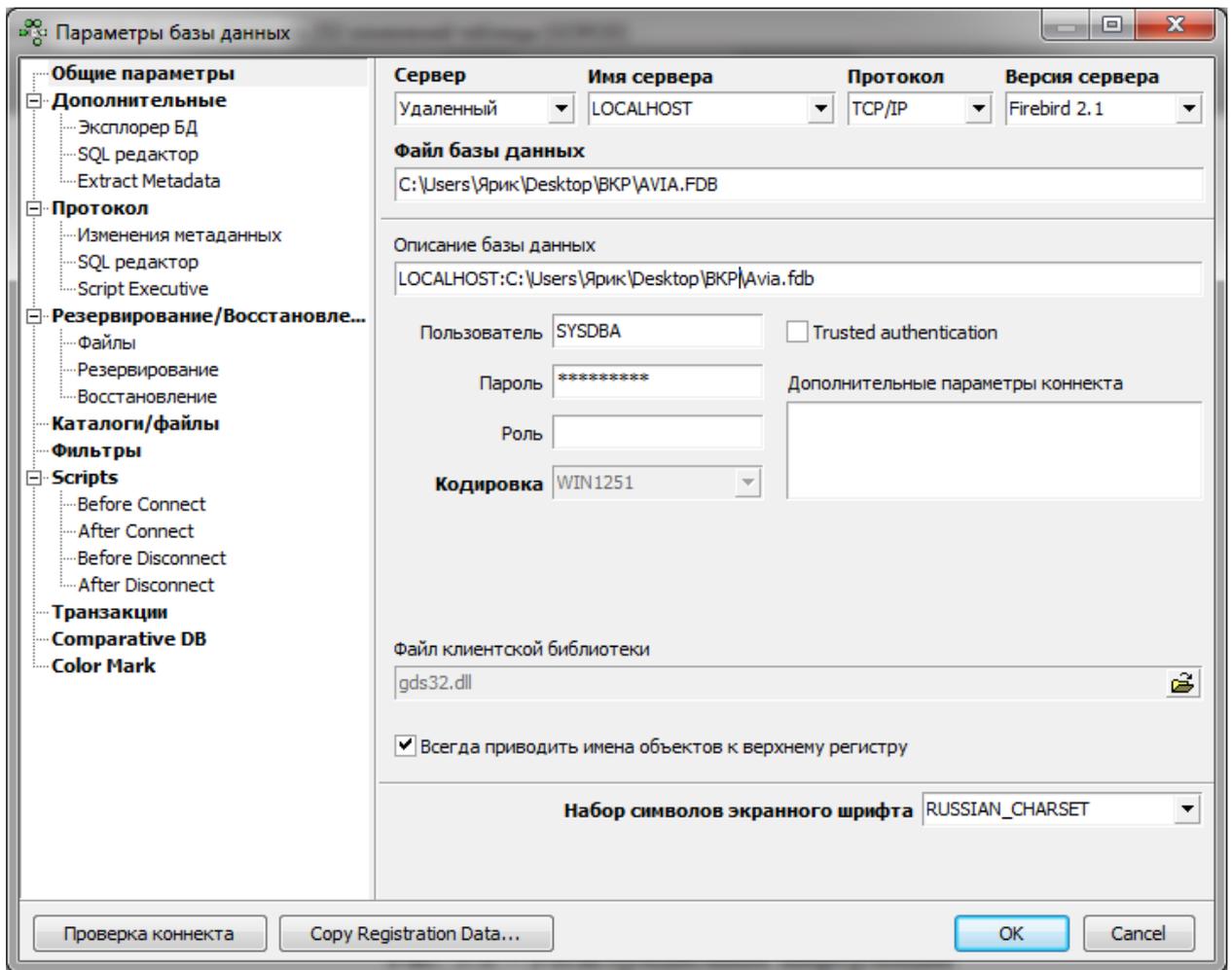


Рисунок 3.8 – Регистрационная информация

Создали таблицы исходя из инфологической модели, приведённой во второй главе. К созданным таблицам относятся: авиакомпания, сотрудники, клиенты, рейсы, бронирование, самолеты, класс цен, город, продажа, должности. Пример заполнения таблицы «Бронирование» представлен на рисунке 3.9, остальные таблицы представлены в приложении В.

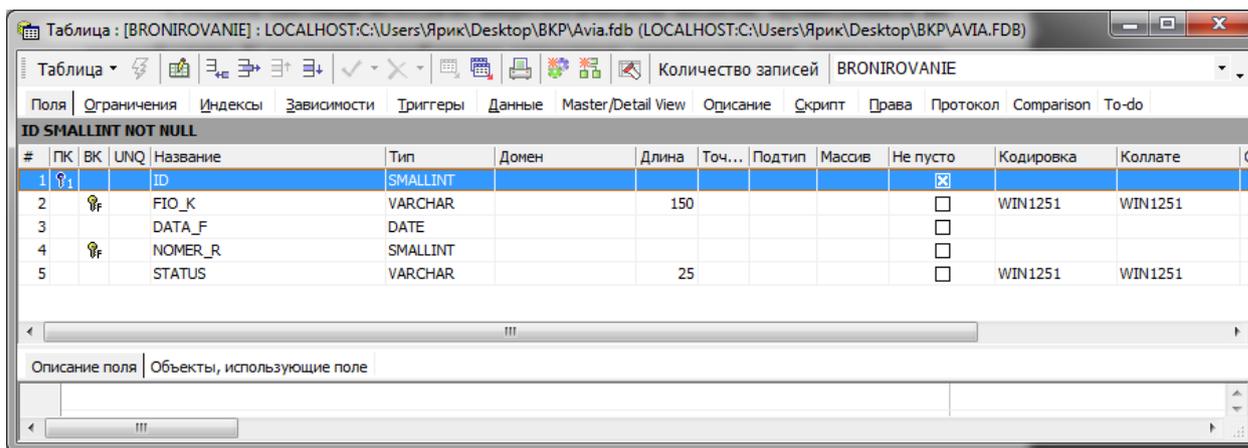


Рисунок 3.9 – Создание таблицы бронирования

Каждую таблицу связали между собой ключами (первичным и внешним) и создали триггеры, генераторы и индексы. Триггер – это хранимая процедура, которая выполняется при изменении каких либо данных. Генератор существует для создания случайных чисел для последовательности записей в таблице. Индексы, как ссылки на какой либо источник создаются с целью упрощения. Это упрощение состоит в том, чтобы с легкостью отыскать таблицу необходимую нам просто указав ее индекс.

Следующим шагом после создания базы данных в Ihexpert идет создание графического интерфейса пользователя. Графический интерфейс должен быть понятен и удобен любому пользователю, работающему на авиакассе. Поэтому создание интерфейса является, можно считать, основной задачей разработки подсистемы бронирования и продажи авиабилетов.

Открыли программу Builder c++ (см. рисунок 3.10). Добавили новую форму, дали ей название «AVIA» и сохранили на жестком диске. Используя компонент «MainMenu1» создали главное меню и в свойстве «Caption» называли форму «Подсистема бронирования и продажи авиабилетов». Меню будет состоять из 6 вкладок:

- Рейсы;
- Оформить продажу;
- Бронирование;

- Справочники;
- Выход;

Так же добавили изображение, чтобы предать нормальный внешний вид главному меню, для этого добавили компонент «Image1» и в свойстве Picture выбирали необходимую картинку. В результате получили главное меню представленное на рисунке 3.11.

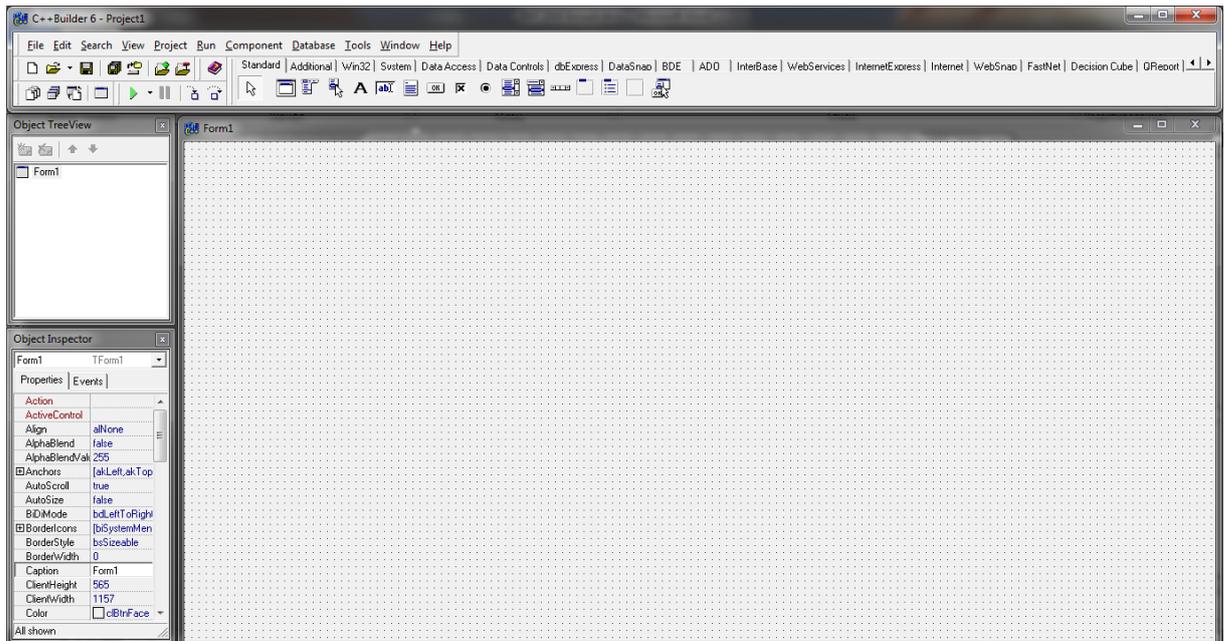


Рисунок 3.10 – Главное меню C++ Builder 6

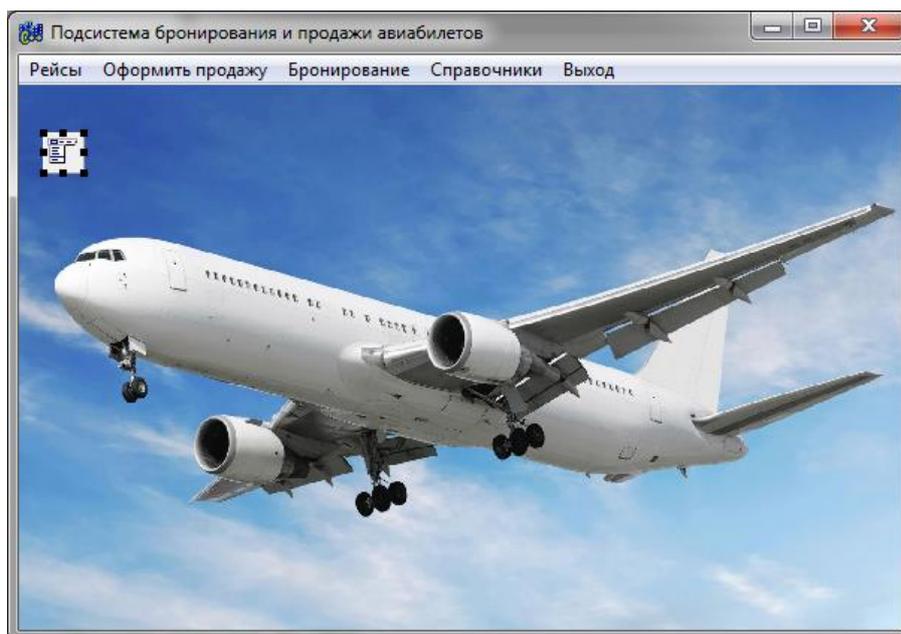


Рисунок 3.11 – Главное меню подсистемы бронирования и продажи авиабилетов

Далее приступили к созданию справочников:

- Должности;
- Самолеты;
- Авиакомпании;
- Города;
- Класс цен;
- Сотрудники.

Каждый из этих справочников содержит 4 компонента (далее приведен пример работы над справочником «должности» остальные справочники представлены в приложении В):

- «DBGrid» используется для отображение таблиц из базы данных ibexpert;
- Кнопка «Button1» имеющая название «Добавить», содержит код для добавление новой записи в таблицу (см. рисунок 3.12);
- Кнопка «Button2» имеющая название «Удалить», содержит код для удаление записи из таблицы (см. рисунок 3.13);
- Кнопка «Button3» имеющая название «Сохранить», содержит код для сохранения всех изменений проделанных с таблицей (см. рисунок 3.14).

```
void __fastcall TFDOLJNOSTI::Button1Click(TObject *Sender)
{
    DM->DOLJNOSTI->Insert ();
    DBGrid1->SetFocus ();
}
```

Рисунок 3.12 – Код кнопки “Добавить”

```
void __fastcall TFDOLJNOSTI::Button2Click(TObject *Sender)
{
    if (Application->MessageBox("Действительно хотите удалить запись?", "Подтвердите удалени
    MB_YESNO + MB_ICONEXCLAMATION) == IDYES)
        DM->DOLJNOSTI->Delete ();
}
```

Рисунок 3.13 – Код кнопки “Удалить”

```
void __fastcall TFDOLJNOSTI::Button3Click(TObject *Sender)
{
if (DM->DOLJNOSTI->Modified) DM->DOLJNOSTI->Post();
}
```

Рисунок 3.14 – Код кнопки “Сохранить”

Суммируя все эти компоненты и свойство «BorderStyle»->«SkyBlue» получили форму справочников, приведённую на рисунке 3.15.

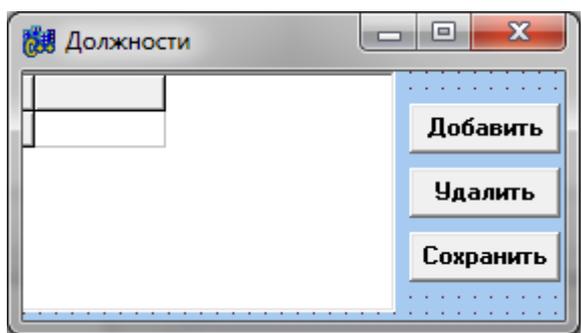


Рисунок 3.15 – Интерфейс справочника “Должности”

Далее создали справочник «клиенты», который отличается от других справочников по составу (см. рисунок 3.16). Отличие этого справочника состояло в наличии в нем компонента «Edit», в который вносились данные, и изменилась кнопка добавить (см. рисунок 3.17). Так же в справочнике «клиент» был добавлен поиск для удобства поиска клиента, код этого компонента представлен на рисунке 3.18.

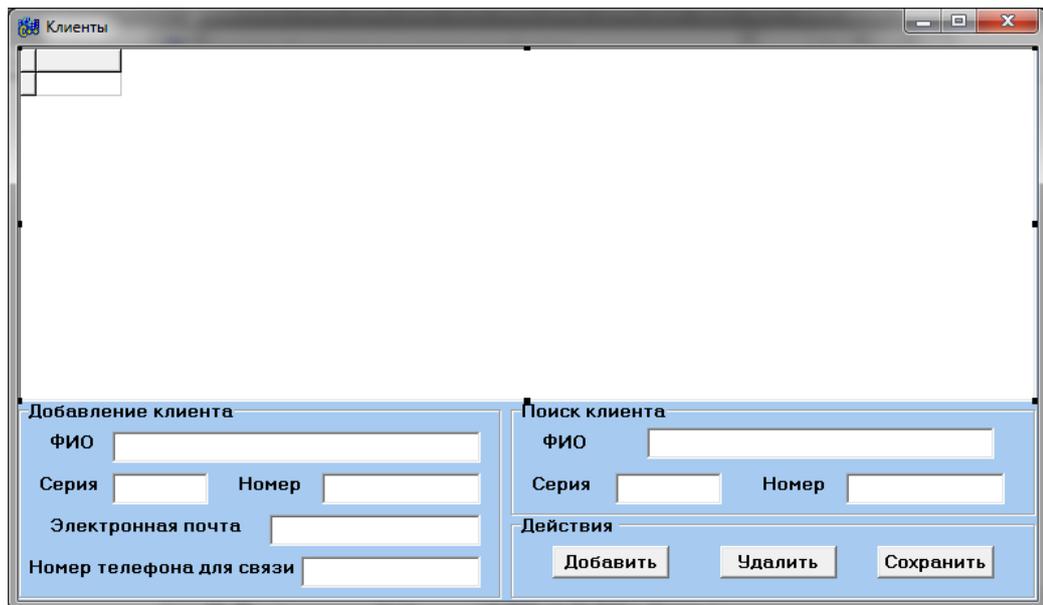


Рисунок 3.16 – Интерфейс справочника “Клиенты”

```
void __fastcall TFKLIENT::Button1Click(TObject *Sender)
{
    DM->KLIENT->Insert();
    DM->KLIENT->Edit();
    DM->KLIENTFIO->AsString = Edit4->Text;
    DM->KLIENTSERIA_P->AsString = Edit5->Text;
    DM->KLIENTNOMER_P->AsString = Edit8->Text;
    DM->KLIENTEMAIL->AsString = Edit6->Text;
    DM->KLIENTMOBTEL->AsString = Edit7->Text;
    DBGrid1->SetFocus();
}
```

Рисунок 3.17 – Код кнопки «Добавить»

```
void __fastcall TFKLIENT::Edit1Change(TObject *Sender)
{
    TLocateOptions SearchOptions;
    Variant locvalues[] = {Edit1->Text};
    DM->KLIENT->Locate("FIO", VarArrayOf(locvalues,1),
    SearchOptions<<loPartialKey<<loCaseInsensitive);
}
```

Рисунок 3.18 – Код поиска клиента по ФИО

Следующим шагом идет создание интерфейса для форм: «рейсы», «продажа» и «бронирование» (см. рисунок 3.19, 3.20, 3.21). Данные формы содержат те же самые компоненты, рассмотренные выше с добавлением

«DBNavigator» и «Panel», которые служат для навигации по таблице и отображение значения соответственно.

Рейсы

Номер рейса	Авиакомпания	Самолет	Город отправления	Город прибытия	Дата отправления	Дата прибытия	Время отправления	Время прибытия	Цена
-------------	--------------	---------	-------------------	----------------	------------------	---------------	-------------------	----------------	------

Действия

Добавить Сохранить

Удалить В отчет

Поиск

Город отправления

Город прибытия

ОЧИСТИТЬ

Рисунок 3.19 – Интерфейс справочника “Рейсы”

Продажа

Номер рейса	ФИО Клиента	Номер брони	Сотрудник	Дата	Имя класса	Оплата
-------------	-------------	-------------	-----------	------	------------	--------

Заполнение данных

Номер рейса

ФИО Клиента

Номер брони

Сотрудник

Дата

Имя класса

Оплата

Очистить

Работа с данными

Добавить

Удалить

Сохранить

Рейсы

Поиск

По клиенту

По сотруднику

Количество свободных мест:

Отчет о продажах

Отчет

Рисунок 3.20 – Интерфейс справочника “Продажа”

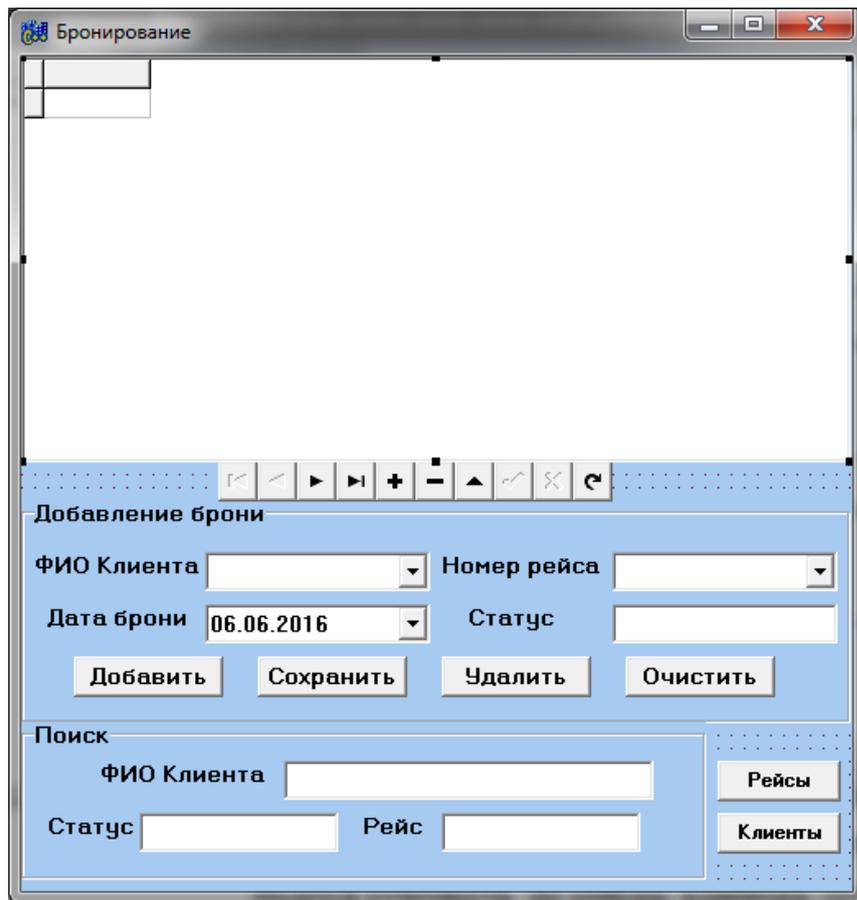


Рисунок 3.21 – Интерфейс справочника “Бронирование”

После создания всех справочников добавили код во вкладки, установленные в главном меню (см. рисунок 3.22).

```

void __fastcall TMainForm::N7Click(TObject *Sender)
{
    FGORODA->Show();
}
//-----
|
void __fastcall TMainForm::N5Click(TObject *Sender)
{
    FREIS->Show();
}
//-----

void __fastcall TMainForm::N12Click(TObject *Sender)
{
    FSAMOLET->Show();
}

```

Рисунок 3.22 – Код вкладок главного меню

Затем создали форму «DM», которая будет отвечать за взаимодействия всех компонентов между собой, и будет содержать в себе все не визуальные компоненты подсистемы. На данной форме установили компонент «IBDatabase1» и в свойстве «DatabaseName» установили соединение с базой данных (см. рисунок 3.23).

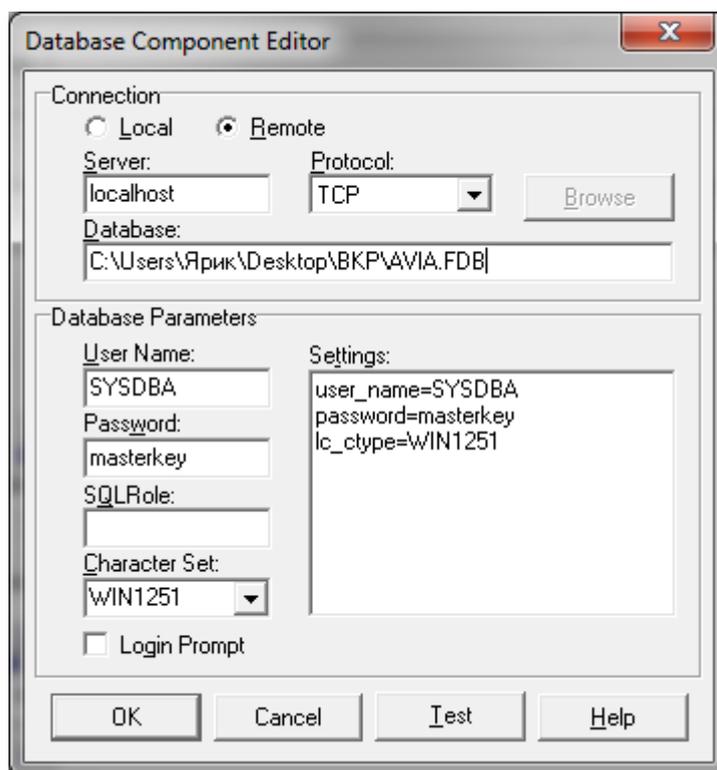


Рисунок 3.23 – Подключение к базе данных «AVIA.FDB»

Так же создали компонент «IBTransaction1» для управления транзакциями. Подключили каждую таблицу через компонент «IBTable», связали с компонентами «IBDatabase» и «IBTransaction1», в свойстве «TableName» указали необходимую таблицу (см. рисунок 3.24).

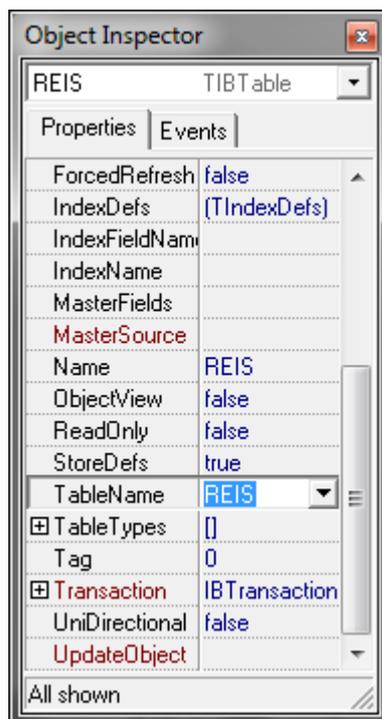


Рисунок 3.24 – Свойства компонента «IBTable»

Для соединения созданных справочников и документов с базой данных, связали компоненты «DataSource» и «DbGrid».

Так же создали запросы, которые будут использоваться на форме «Продажа» для вычисления количества мест на рейсе, оставшихся мест и вывода необходимой к оплате суммы. Код запросов приведен на рисунке 3.25.

```
void __fastcall TFSELL::CBChange(TObject *Sender)
{
    DM->ost->Close();
    DM->ost->SQL->Clear();
    DM->ost->SQL->Add("select KOLVO_PASS from samolet left join reis on samolet.model = reis.samolet where reis.id = '" + CB->Text + "'");
    DM->ost->Open();
    Panel1->Caption = "";
    DM->kolvo->Close();
    DM->kolvo->SQL->Clear();
    DM->kolvo->SQL->Add("select count ( Nomer_r) as kolvo from sell where nomer_r = '" + CB->Text + "'");
    DM->kolvo->Open();
    Panel1->Caption = DM->ost->FieldValues["kolvo_pass"] - DM->kolvo->FieldValues["kolvo"];
    Edit2->Clear();
    DM->cena->SQL->Clear();
    DM->cena->SQL->Add("select COST_BILET from REIS where ID = '" + CB->Text + "'");
    DM->cena->Open();
    if (!DM->cena->FieldByName("COST_BILET")->IsNull)
    Edit2->Text = DM->cena->FieldValues["COST_BILET"];
    else Edit2->Text = "0";
}
```

Рисунок 3.25 – Код запросов «Продажа»

3.3 Описание контрольного примера реализации проекта

Запускаем приложение с помощью файла «Avia.exe». Открывается главное меню с названием «Подсистема бронирования и продажи авиабилетов», представленная на рисунке 3.26.

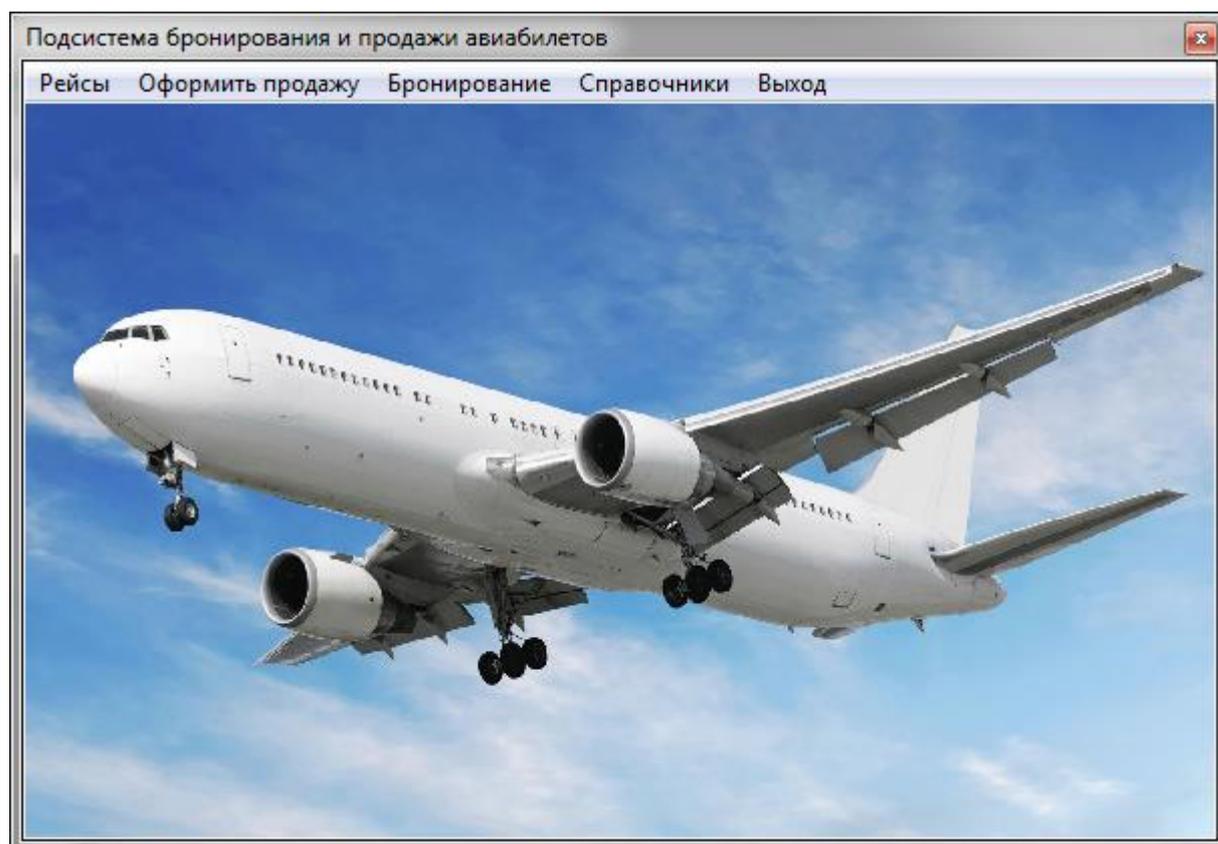


Рисунок 3.26 – Запуск приложения

Приступаем к заполнению справочников. Для этого открываем вкладку «справочники» и выбираем необходимый (см. Рисунок 3.27). Выбираем, к примеру, справочник «города».

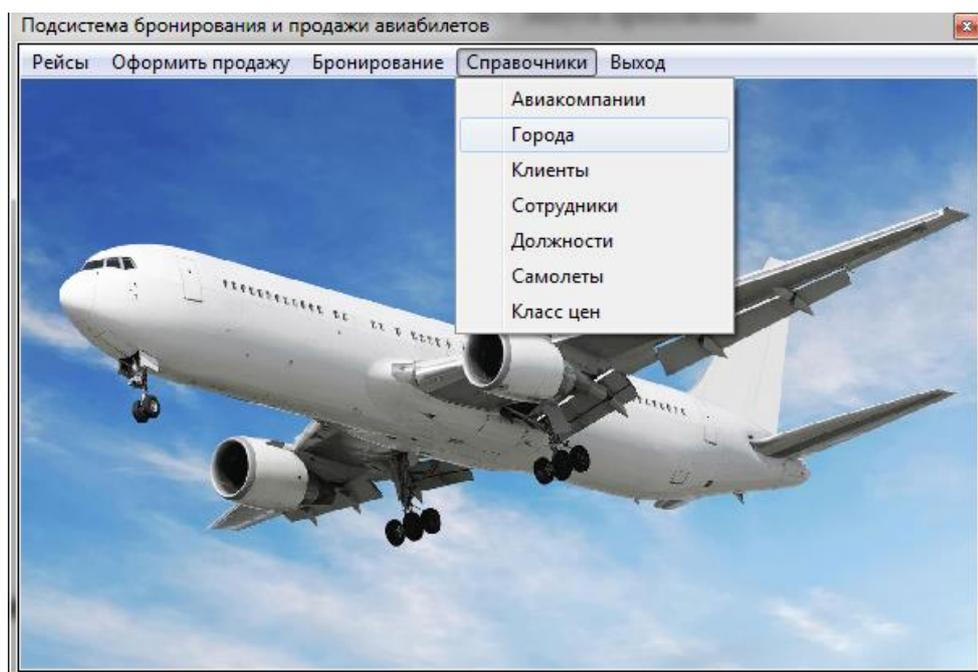


Рисунок 3.27 – Выбор справочника «Города»

Открывается справочник «Города». В этом справочнике добавляем новую запись. Для этого нажимаем кнопку «Добавить». После нажатия данной кнопки курсор переводится на последнюю строчку в таблице, в которую можно вписать название нового города (см. рисунок 3.28).

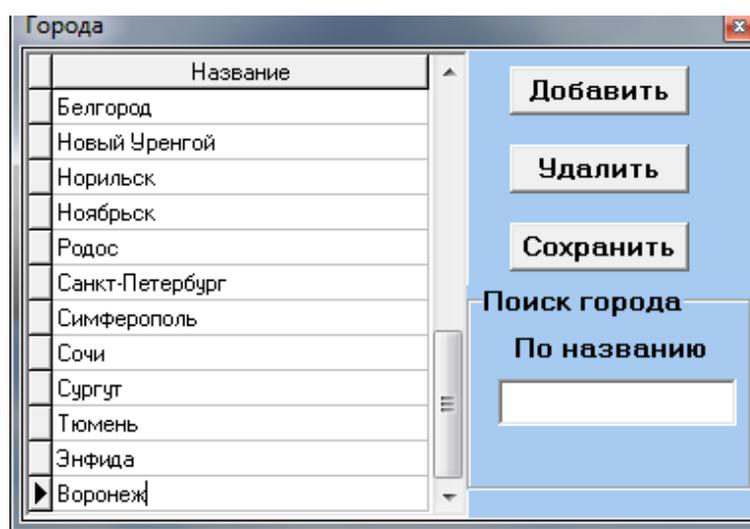


Рисунок 3.28 – Добавление города Воронеж

Если при добавлении ввели не правильно название города, то для этого на форме имеется вкладка «Удалить». При ее нажатии появится запись

с подтверждением удаления (см. рисунок 3.29), выбираем «да» и запись удаляется из таблицы. Есть так же возможность изменить запись, для этого выбираем необходимую запись из таблицы, щелкаем на нее мышью и изменяем ее так как необходимо.

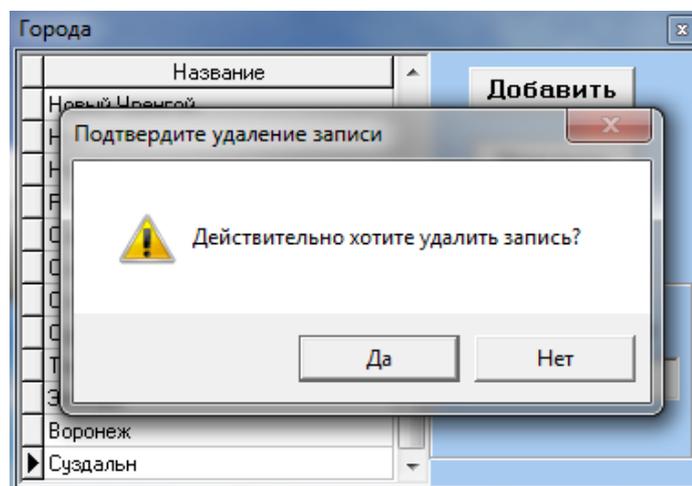


Рисунок 3.29 – Подтверждение удаления записи

Так же в справочнике имеется возможность поиска города. Для того чтобы найти город необходимо навести курсор мыши на панель поиска и ввести необходимый город. Найдем, к примеру, город Курск (см. рисунок 3.30).

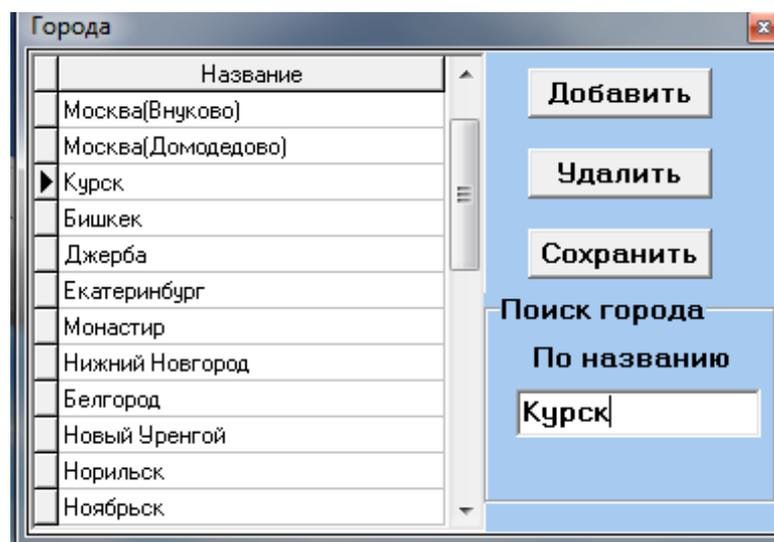
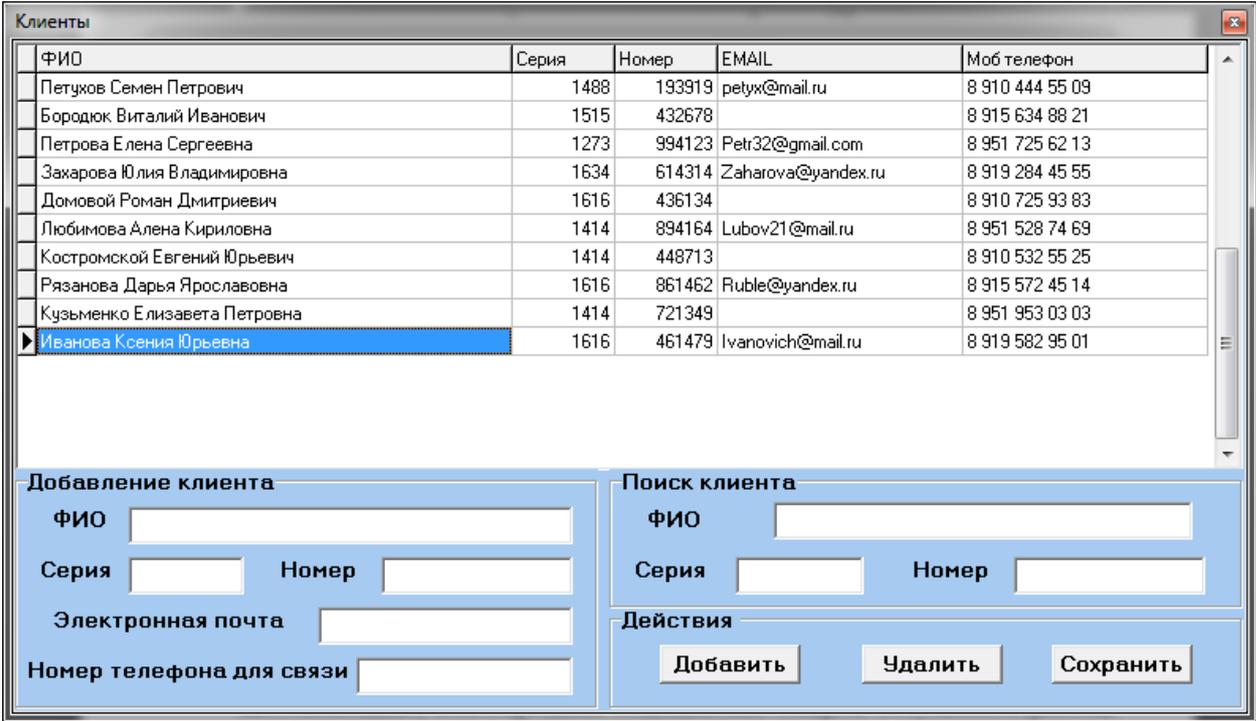


Рисунок 3.30 – Поиск города Курск

После работы со справочником нажимаем кнопку «Сохранить» и выходим в главное меню с помощью красного креста в правом верхнем углу формы.

Создадим, например клиента Степанова Дмитрия Владимировича забронировавшего билет 25.05.2016 на рейс 243 Белгород – Джерба во 2-й класс и оплатившего его 30.05.2016.

Открываем справочник «Клиенты» (см. рисунок 3.31).



ФИО	Серия	Номер	EMAIL	Моб телефон
Петухов Семен Петрович	1488	193919	petyx@mail.ru	8 910 444 55 09
Бородюк Виталий Иванович	1515	432678		8 915 634 88 21
Петрова Елена Сергеевна	1273	994123	Petr32@gmail.com	8 951 725 62 13
Захарова Юлия Владимировна	1634	614314	Zaharova@yandex.ru	8 919 284 45 55
Домовой Роман Дмитриевич	1616	436134		8 910 725 93 83
Любимова Алена Кириловна	1414	894164	Lubov21@mail.ru	8 951 528 74 69
Костромской Евгений Юрьевич	1414	448713		8 910 532 55 25
Рязанова Дарья Ярославовна	1616	861462	Ruble@yandex.ru	8 915 572 45 14
Кузьменко Елизавета Петровна	1414	721349		8 951 953 03 03
Иванова Ксения Юрьевна	1616	461479	Ivanovich@mail.ru	8 919 582 95 01

Добавление клиента

ФИО

Серия Номер

Электронная почта

Номер телефона для связи

Поиск клиента

ФИО

Серия Номер

Действия

Рисунок 3.31 – Справочник «Клиенты»

Производим добавление нового клиента Степанова Дмитрия Владимировича. Остальные данные берем произвольно. Для этого вносим данные в окно «Добавление клиента» и нажимаем кнопку «Добавить». Появившиеся новая запись представлена на рисунке 3.32.

ФИО	Серия	Номер	EMAIL	Моб телефон
Петухов Семен Петрович	1488	193919	petyx@mail.ru	8 910 444 55 09
Бородюк Виталий Иванович	1515	432678		8 915 634 88 21
Петрова Елена Сергеевна	1273	994123	Petr32@gmail.com	8 951 725 62 13
Захарова Юлия Владимировна	1634	614314	Zaharova@yandex.ru	8 919 284 45 55
Домовой Роман Дмитриевич	1616	436134		8 910 725 93 83
Любимова Алена Кириловна	1414	894164	Lubov21@mail.ru	8 951 528 74 69
Костромской Евгений Юрьевич	1414	448713		8 910 532 55 25
Рязанова Дарья Ярославовна	1616	861462	Ruble@yandex.ru	8 915 572 45 14
Кузьменко Елизавета Петровна	1414	721349		8 951 953 03 03
* Степанов Дмитрий Владимирович	1515	943751	Stepanovich@mail.ru	8 910 362 65 84
Иванова Ксения Юрьевна	1616	461479	Ivanovich@mail.ru	8 919 582 95 01

Добавление клиента

ФИО

Серия Номер

Электронная почта

Номер телефона для связи

Поиск клиента

ФИО

Серия Номер

Действия

Рисунок 3.32 – Добавление нового клиента

Сохраняем базу и выходим в главное меню. Затем оформляем бронь. Для этого открываем вкладку «Бронирование» (см. рисунок 3.33).

ФИО Клиента	Дата брони	Номер рейса	Статус
Бородюк Виталий Иванович	07.04.2016	243	Оплачен
Иванова Ксения Юрьевна	12.03.2016	243	Не оплачен
Домовой Роман Дмитриевич	24.05.2016	243	Оплачен
Захарова Юлия Владимировна	05.05.2016	243	Оплачен
Костромской Евгений Юрьевич	24.05.2016	243	Не оплачен
Кузьменко Елизавета Петровна	30.05.2016	243	Оплачен
▶ Петрова Елена Сергеевна	09.05.2016	243	Оплачен

Добавление брони

ФИО Клиента Номер рейса

Дата брони Статус

Поиск

ФИО Клиента

Статус Рейс

Рисунок 3.33 – Форма «Бронирование»

В окне добавление брони с помощью выпадающих списков выбираем ФИО клиента забронировавшего билет (см. рисунок 3.34).

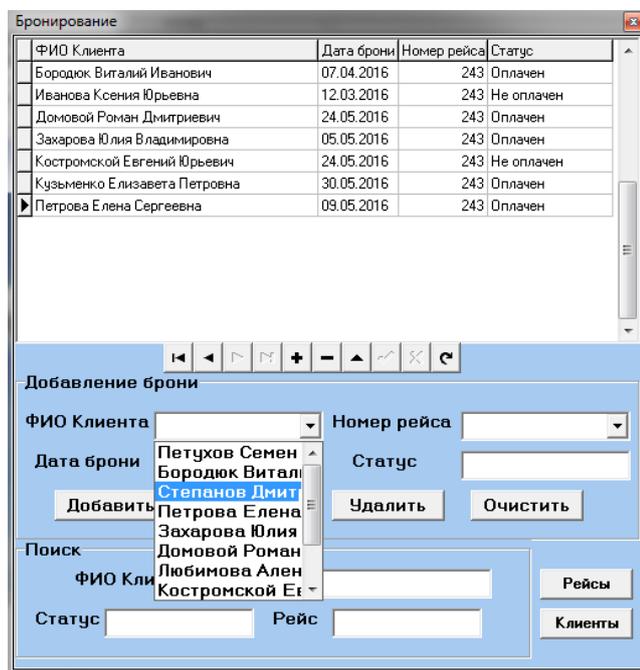


Рисунок 3.34 – Выбор клиента для брони

С помощью такого же списка выбираем номер рейса 243. Выставляем дату брони 25.05.2016 и статус оплачен. Нажимаем кнопку «Добавить». Появляется новая запись забронированного билета. Сохраняем форму и закрываем. Переходим на вкладку «Оформить продажу» (см. рисунок 3.35).

Номер рейса	ФИО Клиента	Номер брони	Сотрудник	Дата	Название класса	Оплата
243	Бородюк Виталий Иванович		1 Березкина Елена Ивановна	07.05.2016	Эконом	12000
243	Домовой Роман Дмитриевич		3 Журова Валентина Борисовна	30.05.2016	Бизнес	24000
243	Захарова Юлия Владимировна		4 Березкина Елена Ивановна	07.05.2016	Эконом	12000
243	Кузьменко Елизавета Петровна		6 Журова Валентина Борисовна	01.06.2016	Бизнес	24000
243	Петрова Елена Сергеевна		7 Журова Валентина Борисовна	22.05.2016	Эконом	12000
243	Рязанова Дарья Ярославовна		Березкина Елена Ивановна	15.06.2016	2-й	18000

Заполнение данных

Номер рейса: Дата:

ФИО Клиента: Имя класса:

Номер брони: Оплата:

Сотрудник:

Работа с данными

Поиск

По клиенту:

По сотруднику:

Количество свободных мест:

Отчет о продажах

Рисунок 3.35 – Форма «Продажа»

С помощью выпадающих списков в окне «Заполнение данных» выбираем «номер рейса – 243» и при выборе рейса в окне появляется информация о количестве свободных мест и сумме к оплате (см. рисунок 3.36)

Номер рейса	ФИО Клиента	Номер брони	Сотрудник	Дата	Название класса	Оплата
243	Бородюк Виталий Иванович		1 Березкина Елена Ивановна	07.05.2016	Эконом	12000
243	Домовой Роман Дмитриевич		3 Журова Валентина Борисовна	30.05.2016	Бизнес	24000
243	Захарова Юлия Владимировна		4 Березкина Елена Ивановна	07.05.2016	Эконом	12000
243	Кузьменко Елизавета Петровна		6 Журова Валентина Борисовна	01.06.2016	Бизнес	24000
243	Петрова Елена Сергеевна		7 Журова Валентина Борисовна	22.05.2016	Эконом	12000
243	Рязанова Дарья Ярославовна		Березкина Елена Ивановна	15.06.2016	2-й	18000

Заполнение данных

Номер рейса: Дата:

ФИО Клиента: Имя класса:

Номер брони: Оплата:

Сотрудник:

Работа с данными

Поиск

По клиенту:

По сотруднику:

Количество свободных мест:

Отчет о продажах

Рисунок 3.36 – Выбор номера рейса

После этого выбираем:

- ФИО Клиента – Степанов Дмитрий Владимирович;
- Номер брони – 8;
- Сотрудник – произвольно;
- Дату – 30.05.2016;
- Имя класса – 2-й класс.

После выбора класса сумма к оплате будет автоматически посчитана, исходя из коэффициента на данный класс, и выведена на экран. В итоге мы получаем оформленную продажу, представленную на рисунке 3.37.

Номер рейса	ФИО Клиента	Номер брони	Сотрудник	Дата	Название класса	Оплата
243	Бородюк Виталий Иванович	1	Березкина Елена Ивановна	07.05.2016	Эконом	12000
243	Домовой Роман Дмитриевич	3	Журова Валентина Борисовна	30.05.2016	Бизнес	24000
243	Захарова Юлия Владимировна	4	Березкина Елена Ивановна	07.05.2016	Эконом	12000
243	Кузьменко Елизавета Петровна	6	Журова Валентина Борисовна	01.06.2016	Бизнес	24000
243	Петрова Елена Сергеевна	7	Журова Валентина Борисовна	22.05.2016	Эконом	12000
243	Рязанова Дарья Ярославовна		Березкина Елена Ивановна	15.06.2016	2-й	18000
243	Степанов Дмитрий Владимирович	8	Журова Валентина Борисовна	30.05.2016	2-й	18000

Заполнение данных

Номер рейса: 243 Дата: 30.05.2016

ФИО Клиента: Степанов Дмитрий Вл Имя класса: 2-й

Номер брони: 8 Оплата: 18000

Сотрудник: Журова Валентина Бс

Очистить

Работа с данными

Добавить

Удалить

Сохранить

Рейсы

Поиск

По клиенту

По сотруднику

Количество свободных мест: 156

Отчет о продажах

Отчет

Рисунок 3.37 – Добавление продажи билета

3.4 Калькуляция себестоимости научно-технической продукции

Для анализа экономической эффективности разрабатываемой подсистемы необходимо рассчитать трудоемкость разработки, все затраты, которые будут использоваться при разработке и в итоге рассчитать срок окупаемости подсистемы на предприятии.

Чтобы рассчитать затраты на оплату труда разработчиков необходимо произвести оценку трудоемкости разработки.

Данные о трудоемкости разработки представлены в таблице 3.1.

Таблица 3.1 – Оценка трудоемкости разработки

Характер работы	Производительность
1) изучение задачи	30
2) обработка текстовой и графической информации	50
3) разработка схем	55
4) информационное наполнение	85
5) создание интерфейса	80
6) отладка	60
7) Составление и согласование проекта пакета служебных инструкций	40

Исходя из таблицы 2, трудоемкость для данной разработки составит 400 часов или 50 рабочих дней при восьмичасовом рабочем дне. Месячный фонд времени работы программиста 168 часов, среднемесячная заработная плата 15000 рублей [32].

Расчет основной заработной платы программиста ($Z_{осн}$) произведен по формуле 1.

$$Z_{осн} = \frac{15000}{168} \times 8 \times 30 = 21428,57(\text{руб.}) \quad (1)$$

Дополнительная заработная плата составляет 12 %. Расчет дополнительной заработной платы ($Z_{доп}$) произведен по формуле 2, общая заработная плата рассчитана по формуле 3 .

$$Z_{\text{доп}} = Z_{\text{осн}} \times 0,12 = 21428,57 \times 0,12 = 2571,42 (\text{руб.}) \quad (2)$$

$$Z_{\text{общ}} = Z_{\text{осн}} + Z_{\text{доп}} = 21428,57 + 2571,42 = 24000 (\text{руб.}) \quad (3)$$

В соответствии с Федеральным законом от 24 июля 2009 года N 212-ФЗ "О страховых взносах в Пенсионный фонд РФ, Фонд социального страхования РФ, Федеральный фонд обязательного медицинского страхования и территориальные фонды обязательного медицинского страхования" (в редакции Федерального закона от 03.12.2011 № 379-ФЗ) страховые взносы следует считать по следующим тарифам:

– отчисления в Пенсионный фонд – 22% от затрат на оплату труда (см. формулу 4):

$$ПФР = Z_{\text{общ}} \times 0,22 = 24000 \times 0,22 = 5280 (\text{руб.}) \quad (4)$$

– отчисления в Фонд Социального страхования – 2,9% от затрат на оплату труда (см. формулу 5):

$$ФСС = Z_{\text{общ}} \times 0,029 = 24000 \times 0,029 = 696 (\text{руб.}) \quad (5)$$

– отчисления в Федеральный Фонд обязательного медицинского страхования – 5,1% от затрат на оплату труда (см. формулу 6):

$$ФФОМС = Z_{\text{общ}} \times 0,051 = 24000 \times 0,051 = 1224 (\text{руб.}) \quad (6)$$

Накладные расходы составляют 15 % от затрат на оплату труда [32].

Расчет накладных расходов (P_H) произведен по формуле 7.

$$P_H = Z_{\text{ОБЩ}} \times 0,15 = 24000 \times 0,15 = 3600 (\text{руб.}) \quad (7)$$

Итого единый социальный налог 10800 рублей.

Страховой взнос на обязательное социальное страхование от 48 несчастных случаев на производстве и профессиональных заболеваний (0,2% от затрат на оплату труда)

$$0,002 * 24000 = 48$$

Итого отчисления на социальные нужды 10848 руб.

К прямым расходам относятся расходы на всю систему связи: интернет, для связи с сервером и телефон, для подтверждения бронирования. В среднем расходы на интернет (на примере компании NETBYNET – тариф «Мой тариф 100») составляет 750 рублей в месяц. Телефонные расходы составят 200 рублей в месяц (на примере компании NETBYNET – тариф «Комбинированный 100»). Так же к прочим прямым расходам относится приобретение лицензионного ключа для C++BUILDER, они составляют 206 999 рублей. Итого к прочим прямым расходам относится сумма 207949.

На основе произведенных расчетов по отдельным статьям калькуляции заполним таблицу 3.

Таблица 3.2 – Калькуляция себестоимости

№	Наименование статей затрат	Сумма (руб.)
1	Материалы	0
3	Затраты на оплату труда работников, непосредственно занятых созданием научно-технической продукции	24000
4	Отчисления на социальные нужды	10848
5	Прочие прямые расходы	207949
6	Накладные расходы	3600
7	Итого:	246397

После расчета себестоимости разрабатываемого программного обеспечения для предприятия «Белгородавиа», составляющего 246397 рублей производим расчеты срока окупаемости внедряемой подсистемы.

Для этого необходимо определить чистый дисконтированный доход (ЧДД), он показывает величину денежных средств, которую инвестор ожидает получить от проекта, после того, как его инвестиции окупят его первоначальные вложения. Если значение ЧДД будет больше 0, то проект можно внедрять. Если значение ЧДД окажется меньше 0, то проект будет считаться убыточным и внедрять его не следует.

Для расчета ЧДД необходимы значения доходов «Белгородавиа» за несколько лет. Возьмем доходы «Белгородавиа» за 4 предыдущих года:

За 2012 год – 92 342 000 руб.

За 2013 год – 161 720 000 руб.

За 2014 год – 248 563 640 руб.

За 2015 год – 260 991 822 руб.

Ставка дисконта на 2016 год ЦБ РФ составляет 11%. Рассчитаем значение ЧДД по формуле 8.

$$ЧДД = \sum_{i=1}^N \frac{\Pi_i + АО}{(1+r)^n} - K \quad (8)$$

Где Π_i – величина чистой прибыли; $АО_i$ – амортизационные отчисления; K – единовременные капиталовложения. Подставим значения в формулу 8 и получим результат.

$$ЧДД = \left(\frac{92,3}{(1+0,11)} + \frac{161,7}{(1+0,11)^2} + \frac{248,5}{(1+0,11)^3} + \frac{260,9}{(1+0,11)^4} \right) - 246,4 = 321,55(\text{руб.})$$

Определим индекс доходности (ИД), который отражает эффективность инвестиционного проекта, по формуле 9.

$$ИД = \frac{\sum_{i=1}^N \frac{\Pi_i + АО}{(1+r)^n}}{K} = \frac{\left(\frac{92,3}{(1+0,11)} + \frac{161,7}{(1+0,11)^2} + \frac{248,5}{(1+0,11)^3} + \frac{260,9}{(1+0,11)^4} \right)}{246,4} = 2,3 \quad (9)$$

Значение ИД получилось $2,3 > 0$, проект по внедрению системы можно реализовать, так как значение больше 0, проект не будет убыточным.

Далее рассчитаем срок окупаемости внедрения разработанной системы по формуле 10.

$$T_{ок} = \frac{K}{\sum_{i=1}^n \frac{D_i}{(1+r)^n}} \quad (10)$$

Где D_i – совокупное сальдо денежного потока от операций, инвестиционной деятельности; r – ставка дисконта; n – номер периода. Подставим значение в формулу 10 и произведем расчеты.

$$T_{ок} = \frac{246,4}{\frac{\left(\frac{92,3}{(1+0,11)} + \frac{161,7}{(1+0,11)^2} + \frac{248,5}{(1+0,11)^3} + \frac{260,9}{(1+0,11)^4} \right)}{4}} = 0,1$$

Срок окупаемости проекта составил 10 месяцев. Полученные расчеты позволяют говорить о том, что все показатели являются положительными и разработанную подсистему можно внедрять на предприятие.

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе была рассмотрена проблема автоматизации учета и продажи медицинских товаров. Результатом работы является готовое приложение. Для решения поставленной цели были решены следующие задачи:

- Построена единая информационная подсистема для работы отечественных авиакасс;
- Решена проблема с бронированием авиабилетов;
- Разработана подсистема продажи авиабилетов;
- Ведется отчетность по рейсам, клиентам, сотрудникам и т.д.

В результате создания подсистемы были улучшены значения следующих показателей:

- числа будущих клиентов;
- времени доступа к списку авиабилетов и другой информации;
- повышения качества (полноты, точности, достоверности, своевременности, согласованности) информации;
- времени взаимодействия сотрудников с клиентами;
- эффективности распределение задач.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Мескон, М.Х. Основы Менеджмента: пер. с англ. [Текст]/ М. Х. Мескон, М. Алберт, Ф. Хедоури. – М.: Дело, 2011. – 342 с.
2. Мшиенин, А.И. Теория экономических информационных систем: [Текст]/ А.И. Мшиенин. — М.: Финансы и статистика, 2011. – 234 с.
3. Петров, В.Н. - «Информационные системы» [Текст]/ В.Н. Петров. - Изд. Питер, 2012г. – 340 с.
4. Алан Р. Саймон Стратегические технологии баз данных: менеджмент на 2010 год [Текст]/ Алан Р. Саймон. - М.: Финансы и статистика, 2009. – 625 с.
5. Кононов, В.Н., «Базы данных: проектирование структуры предметной области» [Текст]/ В.Н. Кононов – Магнитогорск, 2015. – 543 с.
6. Коротков, Э.М. Концепция менеджмента [Текст]/ Э.М. Коротков. – М.: Дека, 2015. – 230 с.
7. Кузнецов, С.Д. Основы современных баз данных [Текст]/ - С.Д. Кузнецов. – Калининград: Издательская группа ВHV, 2012. – 332 с.
8. Баронов, В.В. Автоматизация управления предприятием [Текст]/ В.В. Баронов, Г.Н. Калянов, Ю.Н. Попов. - М.: Инфра-М, 2013. – 543 с.
9. Беспалов, Р. С. Инструментарий разработчика бизнес-процессов [Текст] / Р.С. Беспалов. – М.: Акцион-Медиа, 2013. – 400 с.
10. Болтенков, В.И. Конфигурирование и настройка автоматизированных информационных систем [Текст]/ В.И. Болтенков, А.Л. Литвинов, Н.В. Лычёва. - Белгород: Издательство БелГУ, 2012. – 25 с.
11. Вендров, А.М. Проектирование программного обеспечения экономических информационных систем [Текст]/ А.М. Вендров. – М.: Финансы и статистика, 2012. - 352 с.
12. Годин, В.В. «Информационное обеспечение управленческой деятельности» [Текст]/ В.В. Годин. - М: «Высшая школа», 2012 г. – 345 с.

13. Михелёв, В.М. Базы данных и СУБД [Текст]/ В.М. Михелёв. – Белгород: Издательство БелГУ, 2007. – 200 с.
14. Автоматизированные информационные технологии в экономике под ред. Титоренко, Г.А. [Текст]/ Г.А. Титоренко. – М.: Компьютер, ЮНИТИ, 2015. – 400 с.
15. Архангельский, А.Я. С++ Builder 6 [Текст]/ А.Я. Архангельский. - М.: Бином-Пресс, 2014. – 824 с.
16. Дейт К. Дж. Введение в системы баз данных. [Текст]/ Дейт К. Дж. - М., СПб., Киев: Изд. дом Вильяме, 2014. – 234 с.
17. Иванова, Г.С. Технология программирования [Текст]/ Г.С. Иванова. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2012. – 320 с
18. Методические указания по содержанию, выполнению и защите выпускных квалификационных работ [Электронный ресурс]/ Зайцева Т.В., Пусная О.П., Путивцева Н.П., Игрунова С.В., Нестерова Е.В. – Белгород: Изд-во БелГУ, 2015. – с. 94
19. Исаенко, Е.В. Методические рекомендации по выполнению дипломной работы [Текст]/ Е.В. Исаенко. – Белгород: Изд_во БУПК, 2014. – 25 с.
20. Каминский, А.М. Информатизация бизнеса [Текст]/ А.М. Каминский. – М.: Финансы и статистика, 2015. – 190 с.
21. Липаев, В.В. Системное проектирование сложных программных средств для информационных систем [Текст]/ В.В. Липаев. - М.: Синтег, 2011. – 231 с.
22. Маклаков, С.В. ВРwin и Erwin. Case – средства разработки информационных систем [Текст]/ С.В. Маклаков. – М.: ДИАЛОГ – МИФИ, 2011. – 150 с.
23. Семенов, М.И. и др. Автоматизированные информационные технологии в экономике [Текст]/ М.И. Семенов. — М.: Финансы и статистика, 2012. – 432 с.

24. Смирнова, Г.Н. Проектирование экономических информационных систем: [Текст]/ Г.Н. Смирнова. – М.: Финансы и статистика, 2011.

25. Амириди, Ю. В. Современные IT-решения для финансовой индустрии [Текст]/ Ю. В. Амириди, Н. Е. Анненская – М.: Издательская группа «БДЦ-пресс», 2012.

26. Жизненный цикл информационных систем [Электронный ресурс]. Режим доступа: http://studopedia.ru/13_115077_zhiznenniy-tsikl-informatsionnih-sistem.html, свободный.

27. [Электронный ресурс]. Режим доступа: <http://ref.by/refs/46/9355/1.html>, свободный.

28. Совершенствование организационной структуры организации [Электронный ресурс]. Режим доступа: <http://diplomba.ru/work/22277>, свободный.

29. Организация, планирование и управление производством на примере ООО Производственная компания "Кузнецкий цементный завод" [Электронный ресурс]. Режим доступа: <http://westud.ru/work/261989/Organizaciya-planirovanie-i-upravlenie>, свободный.

30. Белгород (аэропорт) [Электронный ресурс]. Режим доступа: [https://ru.wikipedia.org/wiki/Белгород_\(Аэропорт\)](https://ru.wikipedia.org/wiki/Белгород_(Аэропорт)), свободный.

31. [Электронный ресурс]. Режим доступа: http://fullref.ru/job_6d44e54b860ee1cfa53e215197748772.html, свободный.

32. Разработка программного обеспечения поддержки процессов закупа и учёта хранения товаров на склад [Электронный ресурс]. Режим доступа: [http://coolreferat.com/Разработка программного обеспечения поддержки процессов закупа и учёта хранения товаров на склад](http://coolreferat.com/Разработка_программного_обеспечения_поддержки_процессов_закупа_и_учёта_хранения_товаров_на_склад), свободный.

33. Автоматизированное рабочее место менеджера сервисного центра [Электронный ресурс]. Режим доступа:

http://knowledge.allbest.ru/programming/2c0b65635b3bc68a5c53b89421216d36_0.html, свободный

34. Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Белгородский государственный национальный исследовательский университет» [Электронный ресурс]. Режим доступа: <http://edu.znate.ru/docs/114/index-5317.html?page=14>, свободный

35. Создание и ведение баз данных v <http://ref.by/refs/67/15521/1.html>, свободный

36. Модели и структуры базы данных экспертной системы анализа ускоренных испытаний технических и программных средств при обеспечении надежности [Электронный ресурс]. Режим доступа: <http://cyberleninka.ru/article/n/modeli-i-struktury-bazy-dannyh-ekspertnoy-sistemy-analiza-uskorenyih-ispytaniy-tehnicheskikh-i-programmnyh-sredstv-pri-obespechenii#ixzz4BcvoVcjz>, свободный

37. Давыдов, В.А. Visual C++. Разработка Windows –приложений с помощью MFC и API функций [Текст]/ В.А. Давыдов – М.: БВХ-Петербург, - 2015 г. – 546 с.

38. Румянцев, П.В. Работа с файлами Win32 API [Текст]/ П.В. Румянцев – М.: Горячая линия – Телеком, 2012 г. – 324 с.

39. Мозговой, М.В. C++ мастер класс. 85 нетривиальных проектов, решений и задач [Текст]/ М.В. Мозговой – М.: Наука и техника, 2012 г. – 234 с.

40. Джозеф Джарратано Экспертные системы: принципы разработки и программирование [Текст]/ Джозеф Джарратано, Гари Райли – М.: Издательский дом «Вильямс», 2011. – 156 с.

41. Борри, Х. Firebird: руководство разработчика баз данных [Текст]/ Х. Борри – БХВ.: Петербург 2011. – 342 с.

42. Хомоненко, А. Д. Работа с базами данных в С++ Builder [Текст]/ А. Д. Хомоненко, С. Е. Ададунов — СПб.:БХВ – Петербург, 2013. – 300 с.
43. Оболенски, Н. Практический реинжиниринг бизнеса [Текст]/ Н. Оболенски – М: ЛОРИ, 2014. – 368 с.
44. Пилон, Д. Управление разработкой ПО [Текст]/ Д. Пилон – СПб.: Питер, 2015. – 455 с.
45. Ворожцов, А.В Алгоритмы: Построение, анализ и реализация на языке Си [Текст]/ А.В Ворожцов, Н.А. Винокуров – М.: МФТИ, 2015 г. – 453 с.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Программный код

«DataModule.cpp»

```
#include <vcl.h>
#pragma hdrstop
#include "DataModule.h"
#include "Main.h"
#include "REIS.h"
#include "GORODA.h"
#include "SAMOLET.h"
#include "AVIACOMPANY.h"
#include "DOLJNOSTI.h"
#include "KLIENT.h"
#include "SOTRUDNIKI.h"
#include "KLASS.h"
#include "SELL.h"
#include "BRONIROVANIE.h"
#include "ReportReis.h"
#include "ReportSell.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TDM *DM;
//-----
__fastcall TDM::TDM(TComponent* Owner)
    : TDataModule(Owner)
{
}
//-----
```

«Main.cpp»

```
#include <vcl.h>
#pragma hdrstop
#include "Main.h"
#include "DataModule.h"
#include "REIS.h"
#include "GORODA.h"
#include "SAMOLET.h"
#include "AVIACOMPANY.h"
#include "DOLJNOSTI.h"
#include "KLIENT.h"
#include "SOTRUDNIKI.h"
#include "KLASS.h"
#include "SELL.h"
#include "BRONIROVANIE.h"
#include "ReportReis.h"
#include "ReportSell.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TMainForm *MainForm;
//-----
__fastcall TMainForm::TMainForm(TComponent* Owner)
```

```

        : TForm(Owner)
    {
    }
    //-----
    void __fastcall TMainForm::N7Click(TObject *Sender)
    {
        FGORODA->Show();
    }
    //-----
    void __fastcall TMainForm::N5Click(TObject *Sender)
    {
        FREIS->Show();
    }
    //-----
    void __fastcall TMainForm::N12Click(TObject *Sender)
    {
        FSAMOLET->Show();
    }
    //-----
    void __fastcall TMainForm::N6Click(TObject *Sender)
    {
        FAVIACOMPANY->Show();
    }
    //-----
    void __fastcall TMainForm::N10Click(TObject *Sender)
    {
        FDOLJNOSTI->Show();
    }
    //-----
    void __fastcall TMainForm::N8Click(TObject *Sender)
    {
        FKLIENT->Show();
    }
    //-----
    void __fastcall TMainForm::N9Click(TObject *Sender)
    {
        FSOTRUDNIKI->Show();
    }
    //-----
    void __fastcall TMainForm::N11Click(TObject *Sender)
    {
        FKLASS->Show();
    }
    //-----
    void __fastcall TMainForm::N1Click(TObject *Sender)
    {
        FSELL->Show();
    }
    //-----
    void __fastcall TMainForm::N2Click(TObject *Sender)
    {
        FBRONIROVANIE->Show();
    }
    //-----

    void __fastcall TMainForm::N4Click(TObject *Sender)
    {
        Close();
    }

```

```

#include <vcl.h>
#pragma hdrstop
#include "REIS.h"
#include "DataModule.h"
#include "Main.h"
#include "GORODA.h"
#include "BRONIROVANIE.h"
#include "ReportReis.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TFREIS *FREIS;
//-----
__fastcall TFREIS::TFREIS(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TFREIS::FormCreate(TObject *Sender)
{
DM->GORODA->Active = true;
DM->GORODA->First();
while (!DM->GORODA->Eof) {
CB3->Items->Add(DM->GORODANAME->AsString);
CB4->Items->Add(DM->GORODANAME->AsString);
DM->GORODA->Next(); }
DM->AVIA->Active = true;
DM->AVIA->First();
while (!DM->AVIA->Eof) {
CB1->Items->Add(DM->AVIANAME->AsString);
DM->AVIA->Next(); }
DM->SAMOLET->Active = true;
DM->SAMOLET->First();
while (!DM->SAMOLET->Eof) {
CB2->Items->Add(DM->SAMOLETMODEL->AsString);
DM->SAMOLET->Next(); }
}
//-----
void __fastcall TFREIS::Button1Click(TObject *Sender)
{
DM->REIS->Insert();
DM->REIS->Edit();
DM->REISID->AsString = Edit1->Text;
DM->REISCOMPANY->AsString = CB1->Text;
DM->REISSAMOLET->AsString = CB2->Text;
DM->REISGOROD_OTPR->AsString = CB3->Text;
DM->REISGOROD_PRIB->AsString = CB4->Text;
DM->REISDATA_OTPR->AsString = DateTimePicker1->Date;
DM->REISDATA_PRIB->AsString = DateTimePicker3->Date;
DM->REISTIME_OTPR->AsString = DateTimePicker2->Time;
DM->REISTIME_PRIB->AsString = DateTimePicker4->Time;
DM->REISCOST_BILET->AsString = Edit2->Text;
DBGrid1->SetFocus();
}
//-----
void __fastcall TFREIS::Button3Click(TObject *Sender)
{
if (DM->REIS->Modified) DM->REIS->Post();
}

```

```

//-----
void __fastcall TFREIS::Button4Click(TObject *Sender)
{
CB1->Clear();
CB2->Clear();
CB3->Clear();
CB4->Clear();
Edit1->Clear();
Edit2->Clear();
}
//-----
void __fastcall TFREIS::Button2Click(TObject *Sender)
{
if (Application->MessageBox("Действительно хотите удалить запись?", "Подтвердите удаление записи", MB_YESNO + MB_ICONEXCLAMATION) == IDYES)
DM->REIS->Delete();
}
//-----
void __fastcall TFREIS::Button5Click(TObject *Sender)
{
RepReis->QuickRep1->Preview();
}
//-----
void __fastcall TFREIS::Edit3Change(TObject *Sender)
{
TLocateOptions SearchOptions;
Variant locvalues[] = {Edit3->Text};
DM->REIS->Locate("GOROD_OTPR", VarArrayOf(locvalues,1), SearchOptions<<loPartialKey<<loCaseInsensitive);
}
//-----
void __fastcall TFREIS::Edit4Change(TObject *Sender)
{
TLocateOptions SearchOptions;
Variant locvalues[] = {Edit3->Text};
DM->REIS->Locate("GOROD_PRIB", VarArrayOf(locvalues,1), SearchOptions<<loPartialKey<<loCaseInsensitive);
}

                                     «Sell.cpp»

#include <vcl.h>
#pragma hdrstop
#include "SELL.h"
#include "DataModule.h"
#include "Main.h"
#include "REIS.h"
#include "ReportSell.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TFSELL *FSELL;
//-----
__fastcall TFSELL::TFSELL(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TFSELL::FormCreate(TObject *Sender)
{

```

```

CB->Clear();
DM->REIS->Active = true;
DM->REIS->First();
while (!DM->REIS->Eof) {
CB->Items->Add(DM->REISID->AsString);
DM->REIS->Next(); }
CB1->Clear();
DM->KLIENT->Active = true;
DM->KLIENT->First();
while (!DM->KLIENT->Eof) {
CB1->Items->Add(DM->KLIENTFIO->AsString);
DM->KLIENT->Next(); }
CB2->Clear();
DM->BRONIROVANIE->Active = true;
DM->BRONIROVANIE->First();
while (!DM->BRONIROVANIE->Eof) {
CB2->Items->Add(DM->BRONIROVANIEID->AsString);
DM->BRONIROVANIE->Next(); }
CB3->Clear();
DM->SOTRUDNIKI->Active = true;
DM->SOTRUDNIKI->First();
while (!DM->SOTRUDNIKI->Eof) {
CB3->Items->Add(DM->SOTRUDNIKIFIO->AsString);
DM->SOTRUDNIKI->Next(); }
CB4->Clear();
DM->KLASS->Active = true;
DM->KLASS->First();
while (!DM->KLASS->Eof) {
CB4->Items->Add(DM->KLASSNAME->AsString);
DM->KLASS->Next(); }
}
//-----
void __fastcall TFSELL::Button1Click(TObject *Sender)
{
    DM->SELL->Insert();
    DM->SELL->Edit();
    DM->SELLNOMER_R->AsString = CB->Text;
    DM->SELLFIO_K->AsString = CB1->Text;
    DM->SELLNOMER_B->AsString = CB2->Text;
    DM->SELLSOTRUDNIK->AsString = CB3->Text;
    DM->SELLDATA->AsString = DTP->Date;
    DM->SELLNAME_C->AsString = CB4->Text;
    DM->SELLOPLATA->AsString = Edit2->Text;
}
//-----
void __fastcall TFSELL::Edit1Change(TObject *Sender)
{
    TLocateOptions SearchOptions;
    Variant locvalues[] = {Edit1->Text};
    DM->SELL->Locate("FIO_K", VarArrayOf(locvalues,1),
    SearchOptions<<loPartialKey<<loCaseInsensitive);
}
//-----
void __fastcall TFSELL::Edit3Change(TObject *Sender)
{
    TLocateOptions SearchOptions;
    Variant locvalues[] = {Edit3->Text};
    DM->SELL->Locate("SOTRUDNIK", VarArrayOf(locvalues,1),
    SearchOptions<<loPartialKey<<loCaseInsensitive);
}

```

```

//-----
void __fastcall TFSELL::CBChange(TObject *Sender)
{
DM->ost->Close();
DM->ost->SQL->Clear();
DM->ost->SQL->Add("select KOLVO_PASS from samolet left join reis on
samolet.model = reis.samolet where reis.id = '" + CB->Text + "'");
DM->ost->Open();
Panell1->Caption = "";
DM->kolvo->Close();
DM->kolvo->SQL->Clear();
DM->kolvo->SQL->Add("select count ( Nomer_r) as kolvo from sell where
nomer_r = '" + CB->Text + "'");
DM->kolvo->Open();
Panell1->Caption = DM->ost->FieldValues["kolvo_pass"] - DM->kolvo-
>FieldValues["kolvo"];
Edit2->Clear();
DM->cena->SQL->Clear();
DM->cena->SQL->Add("select COST_BILET from REIS where ID = '" + CB-
>Text + "'");
DM->cena->Open();
if (!DM->cena->FieldByName("COST_BILET")->IsNull)
Edit2->Text = DM->cena->FieldValues["COST_BILET"];
else Edit2->Text = "0";
}
//-----
void __fastcall TFSELL::CB4Change(TObject *Sender)
{
Edit2->Clear();
CBChange(CB);
float a, b;
if (CB4->ItemIndex == 0) a = 1;
if (CB4->ItemIndex == 1) a = 1.5;
if (CB4->ItemIndex == 2) a = 2;
b = StrToFloat(Edit2->Text) * a;
Edit2->Text = FloatToStr(b);
}
//-----
void __fastcall TFSELL::Button4Click(TObject *Sender)
{
RepSell->QuickRep1->Preview();
}
//-----
-----
void __fastcall TFSELL::Button2Click(TObject *Sender)
{
if (Application->MessageBox("Действительно хотите удалить
запись?", "Подтвердите удаление записи",
MB_YESNO + MB_ICONEXCLAMATION) == IDYES)
DM->SELL->Delete();
}
//-----
void __fastcall TFSELL::Button3Click(TObject *Sender)
{
if (DM->SELL->Modified) DM->SELL->Post();
}
//-----
void __fastcall TFSELL::Button5Click(TObject *Sender)
{
FREIS->Show();
}

```

```

}
//-----
void __fastcall TFSELL::Button6Click(TObject *Sender)
{
CB->Clear();
CB1->Clear();
CB2->Clear();
CB3->Clear();
CB4->Clear();
Edit1->Clear();
Edit2->Clear();
Edit3->Clear();
}

                                     «Bronirovanie.cpp»

#include <vcl.h>
#pragma hdrstop
#include "BRONIROVANIE.h"
#include "DataModule.h"
#include "Main.h"
#include "KLIENT.h"
#include "REIS.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TFBRONIROVANIE *FBRONIROVANIE;
//-----
__fastcall TFBRONIROVANIE::TFBRONIROVANIE(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TFBRONIROVANIE::Edit3Change(TObject *Sender)
{
TLocateOptions SearchOptions;
Variant locvalues[] = {Edit3->Text};
DM->BRONIROVANIE->Locate("FIO_K", VarArrayOf(locvalues,1),
SearchOptions<<loPartialKey<<loCaseInsensitive);
}
//-----
void __fastcall TFBRONIROVANIE::Edit4Change(TObject *Sender)
{
TLocateOptions SearchOptions;
Variant locvalues[] = {Edit4->Text};
DM->BRONIROVANIE->Locate("STATUS", VarArrayOf(locvalues,1),
SearchOptions<<loPartialKey<<loCaseInsensitive);
}
//-----
void __fastcall TFBRONIROVANIE::FormCreate(TObject *Sender)
{
CB->Clear();
DM->KLIENT->Active = true;
DM->KLIENT->First();
while (!DM->KLIENT->Eof) {
CB->Items->Add(DM->KLIENTFIO->AsString);
DM->KLIENT->Next(); }
CB1->Clear();
DM->REIS->Active = true;
DM->REIS->First();
while (!DM->REIS->Eof) {

```

```

    CB1->Items->Add(DM->REISID->AsString);
    DM->REIS->Next(); }
}
//-----
void __fastcall TFBRONIROVANIE::Button1Click(TObject *Sender)
{
    DM->BRONIROVANIE->Insert();
    DM->BRONIROVANIE->Edit();
    DM->BRONIROVANIIEFIO_K->AsString = CB->Text;
    DM->BRONIROVANIIEFIO_F->AsString = DTP->Date;
    DM->BRONIROVANIENOMER_R->AsString = CB1->Text;
    DM->BRONIROVANIESTATUS->AsString = Edit2->Text;
    DBGrid1->SetFocus();
}
//-----
void __fastcall TFBRONIROVANIE::Button3Click(TObject *Sender)
{
    if (Application->MessageBox("Действительно хотите удалить запись?", "Подтвердите удаление записи", MB_YESNO + MB_ICONEXCLAMATION) == IDYES)
        DM->BRONIROVANIE->Delete();
}
//-----
void __fastcall TFBRONIROVANIE::Button2Click(TObject *Sender)
{
    if (DM->BRONIROVANIE->Modified) DM->BRONIROVANIE->Post();
}
//-----
void __fastcall TFBRONIROVANIE::Button4Click(TObject *Sender)
{
    CB->Clear();
    CB1->Clear();
    Edit2->Clear();
}
//-----
void __fastcall TFBRONIROVANIE::Button6Click(TObject *Sender)
{
    FKLIENT->Show();
}
//-----
void __fastcall TFBRONIROVANIE::Button5Click(TObject *Sender)
{
    FREIS->Show();
}
}

```

«Klient.cpp»

```

#include <vcl.h>
#pragma hdrstop
#include "KLIENT.h"
#include "DataModule.h"
#include "Main.h"
#include "BRONIROVANIE.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TFKLIENT *FKLIENT;
//-----
fastcall TFKLIENT::TFKLIENT(TComponent* Owner)
    : TForm(Owner)
{

```

```

}
//-----
void __fastcall TFKLIENT::Button1Click(TObject *Sender)
{
DM->KLIENT->Insert();
DM->KLIENT->Edit();
DM->KLIENTFIO->AsString = Edit4->Text;
DM->KLIENTSERIA_P->AsString = Edit5->Text;
DM->KLIENTNOMER_P->AsString = Edit8->Text;
DM->KLIENTEMAIL->AsString = Edit6->Text;
DM->KLIENTMOBTEL->AsString = Edit7->Text;
DBGrid1->SetFocus();
}
//-----
void __fastcall TFKLIENT::Button2Click(TObject *Sender)
{
if (Application->MessageBox("Действительно хотите удалить запись?", "Подтвердите удаление записи", MB_YESNO + MB_ICONEXCLAMATION) == IDYES)
DM->KLIENT->Delete();
}
//-----
void __fastcall TFKLIENT::Button3Click(TObject *Sender)
{
if (DM->KLIENT->Modified) DM->KLIENT->Post();
}
//-----
void __fastcall TFKLIENT::Edit1Change(TObject *Sender)
{
TLocateOptions SearchOptions;
Variant locvalues[] = {Edit1->Text};
DM->KLIENT->Locate("FIO", VarArrayOf(locvalues,1), SearchOptions<<loPartialKey<<loCaseInsensitive);
}
//-----
void __fastcall TFKLIENT::Edit2Change(TObject *Sender)
{
TLocateOptions SearchOptions;
Variant locvalues[] = {Edit1->Text};
DM->KLIENT->Locate("SERIA_P", VarArrayOf(locvalues,1), SearchOptions<<loPartialKey<<loCaseInsensitive);
}

                                        «Klass.cpp»

#include <vcl.h>
#pragma hdrstop

#include "KLASS.h"
#include "DataModule.h"
#include "Main.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TFKCLASS *FKLASS;
//-----
__fastcall TFKCLASS::TFKCLASS(TComponent* Owner)
: TForm(Owner)
{
}
}
//-----

```

```

void __fastcall TFKLASS::Button1Click(TObject *Sender)
{
    DM->KLASS->Insert();
    DBGrid1->SetFocus();
}
//-----
void __fastcall TFKLASS::Button2Click(TObject *Sender)
{
    if (Application->MessageBox("Действительно хотите удалить запись?", "Подтвердите удаление записи", MB_YESNO + MB_ICONEXCLAMATION) == IDYES)
        DM->KLASS->Delete();
}
//-----
void __fastcall TFKLASS::Button3Click(TObject *Sender)
{
    if (DM->KLASS->Modified) DM->KLASS->Post();
}

```

«Goroda.cpp»

```

#include <vcl.h>
#pragma hdrstop
#include "GORODA.h"
#include "DataModule.h"
#include "Main.h"
#include "REIS.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TFGORODA *FGORODA;
//-----
__fastcall TFGORODA::TFGORODA(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TFGORODA::Button1Click(TObject *Sender)
{
    DM->GORODA->Insert();
    DBGrid1->SetFocus();
}
//-----
void __fastcall TFGORODA::Button2Click(TObject *Sender)
{
    if (Application->MessageBox("Действительно хотите удалить запись?", "Подтвердите удаление записи", MB_YESNO + MB_ICONEXCLAMATION) == IDYES)
        DM->GORODA->Delete();
}
//-----
void __fastcall TFGORODA::Button3Click(TObject *Sender)
{
    if (DM->GORODA->Modified) DM->GORODA->Post();
}
//-----
void __fastcall TFGORODA::Edit1Change(TObject *Sender)
{
    TLocateOptions SearchOptions;
    Variant locvalues[] = {Edit1->Text};
    DM->GORODA->Locate("NAME", VarArrayOf(locvalues,1),

```

```
SearchOptions<<loPartialKey<<loCaseInsensitive);
}
```

«Doljnosti.cpp»

```
#include <vcl.h>
#pragma hdrstop
#include "DOLJNOSTI.h"
#include "DataModule.h"
#include "Main.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TFDOLJNOSTI *FDOLJNOSTI;
//-----
__fastcall TFDOLJNOSTI::TFDOLJNOSTI(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TFDOLJNOSTI::Button1Click(TObject *Sender)
{
    DM->DOLJNOSTI->Insert();
    DBGrid1->SetFocus();
}
//-----
void __fastcall TFDOLJNOSTI::Button2Click(TObject *Sender)
{
    if (Application->MessageBox("Действительно хотите удалить запись?", "Подтвердите удаление записи", MB_YESNO + MB_ICONEXCLAMATION) == IDYES)
        DM->DOLJNOSTI->Delete();
}
//-----
void __fastcall TFDOLJNOSTI::Button3Click(TObject *Sender)
{
    if (DM->DOLJNOSTI->Modified) DM->DOLJNOSTI->Post();
}
}
```

«Aviacompany.cpp»

```
#include <vcl.h>
#pragma hdrstop
#include "AVIACOMPANY.h"
#include "DataModule.h"
#include "GORODA.h"
#include "Main.h"
#include "SAMOLET.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TFAVIACOMPANY *FAVIACOMPANY;
//-----
__fastcall TFAVIACOMPANY::TFAVIACOMPANY(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TFAVIACOMPANY::Button1Click(TObject *Sender)
{
    DM->AVIA->Insert();
    DBGrid1->SetFocus();
}
```

```

}
//-----

void __fastcall TFAVIACOMPANY::Button2Click(TObject *Sender)
{
    if (Application->MessageBox("Действительно хотите удалить запись?", "Подтвердите удаление записи", MB_YESNO + MB_ICONEXCLAMATION) == IDYES)
        DM->AVIA->Delete();
}
//-----

void __fastcall TFAVIACOMPANY::Button3Click(TObject *Sender)
{
    if (DM->AVIA->Modified) DM->AVIA->Post();
}
//-----

void __fastcall TFAVIACOMPANY::Edit1Change(TObject *Sender)
{
    TLocateOptions SearchOptions;
    Variant locvalues[] = {Edit1->Text};
    DM->AVIA->Locate("NAME", VarArrayOf(locvalues,1), SearchOptions<<loPartialKey<<loCaseInsensitive);
}

                                     «Sotrudniki.cpp»

#include <vcl.h>
#pragma hdrstop
#include "SOTRUDNIKI.h"
#include "DataModule.h"
#include "Main.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TFSOTRUDNIKI *FSOTRUDNIKI;
//-----
__fastcall TFSOTRUDNIKI::TFSOTRUDNIKI(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TFSOTRUDNIKI::Button1Click(TObject *Sender)
{
    DM->SOTRUDNIKI->Insert();
    DBGrid1->SetFocus();
}
//-----

void __fastcall TFSOTRUDNIKI::Button2Click(TObject *Sender)
{
    if (Application->MessageBox("Действительно хотите удалить запись?", "Подтвердите удаление записи", MB_YESNO + MB_ICONEXCLAMATION) == IDYES)
        DM->SOTRUDNIKI->Delete();
}
//-----

void __fastcall TFSOTRUDNIKI::Button3Click(TObject *Sender)
{
    if (DM->SOTRUDNIKI->Modified) DM->SOTRUDNIKI->Post();
}

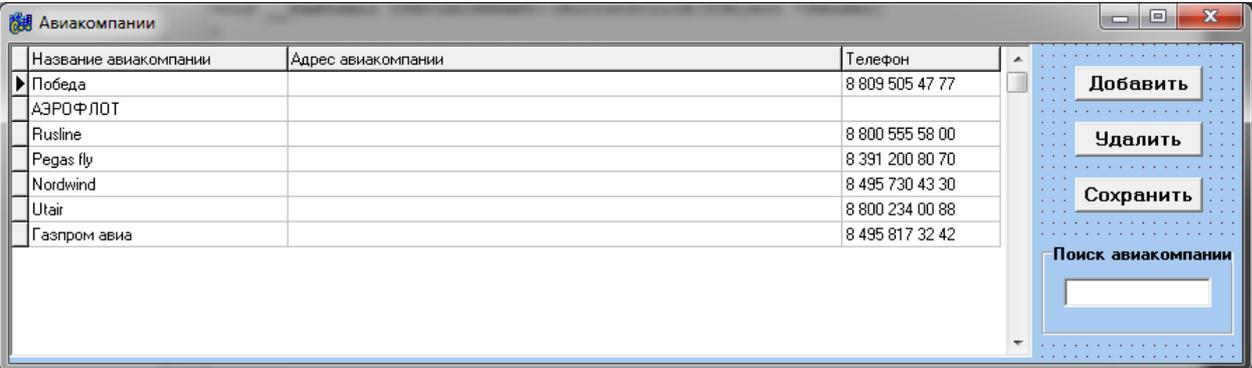
```

«Samolet.cpp»

```
#include <vcl.h>
#pragma hdrstop
#include "SAMOLET.h"
#include "DataModule.h"
#include "Main.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TFSAMOLET *FSAMOLET;
//-----
__fastcall TFSAMOLET::TFSAMOLET(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TFSAMOLET::Button1Click(TObject *Sender)
{
    DM->SAMOLET->Insert();
    DBGrid1->SetFocus();
}
//-----
void __fastcall TFSAMOLET::Button2Click(TObject *Sender)
{
    if (Application->MessageBox("Действительно хотите удалить запись?", "Подтвердите удаление записи", MB_YESNO + MB_ICONEXCLAMATION) == IDYES)
        DM->SAMOLET->Delete();
}
//-----
void __fastcall TFSAMOLET::Button3Click(TObject *Sender)
{
    if (DM->SAMOLET->Modified) DM->SAMOLET->Post();
}
```

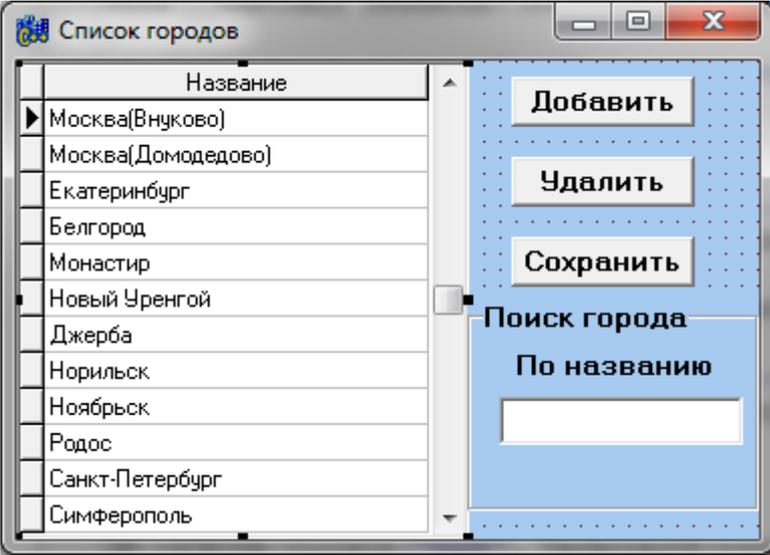
ПРИЛОЖЕНИЕ Б

ИСХОДНЫЕ ДАННЫЕ



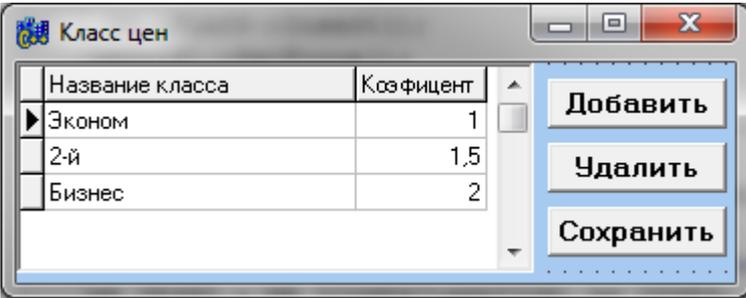
Название авиакомпании	Адрес авиакомпании	Телефон
Победа		8 809 505 47 77
AZRFLLOT		
Rusline		8 800 555 58 00
Pegas fly		8 391 200 80 70
Nordwind		8 495 730 43 30
Utair		8 800 234 00 88
Газпром авиа		8 495 817 32 42

Рисунок Б.1 - Данные из справочника «Авиакомпании»



Название
Москва(Внуково)
Москва(Домодедово)
Екатеринбург
Белгород
Монастир
Новый Уренгой
Джерба
Норильск
Ноябрьск
Родос
Санкт-Петербург
Симферополь

Рисунок Б.2 - Данные из справочника «Города»



Название класса	Коэффициент
Эконом	1
2-й	1,5
Бизнес	2

Рисунок Б.3 - Данные из справочника «Класс цен»

Модель самолета	Количество пассажиров
Tu-154	175
B-737-300	122
CRJ-200	50
A-321	185
B-737-800	162
SU9	125
Як-42	120
B-757-200	235

Рисунок Б.4 - Данные из справочника «Самолет»

ФИО	Должности	Моб телефон
Журова Валентина Борисовна	Кассир	8 910 201 30 43
Березкина Елена Ивановна	Кассир	8 915 564 32 11

Рисунок Б.5 - Данные из справочника «Сотрудники»

ПРИЛОЖЕНИЕ В

Исходные таблицы

Таблица : [AVIACOMPANY] : LOCALHOST:C:\Users\Ярик\Desktop\BKP\Avia.fdb (LOCALHOST:C:\Users\Ярик\Desktop\BKP\AVIA.FDB)

Поля | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Количество записей | Протокол | Comparison | To-do

ID SMALLINT NOT NULL

#	ПК	ВК	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка	Коллате
1	<input checked="" type="checkbox"/>			ID	SMALLINT						<input checked="" type="checkbox"/>		
2			<input checked="" type="checkbox"/>	NAME	VARCHAR		25				<input type="checkbox"/>	WIN1251	WIN1251
3				ADRES	CHAR		50				<input type="checkbox"/>	WIN1251	WIN1251
4				TELEFON	VARCHAR		15				<input type="checkbox"/>	WIN1251	WIN1251

Описание поля | Объекты, использующие поле

Рисунок В.1 - Создание справочника «Авиакомпаний»

Таблица : [DOLJNOSTI] : LOCALHOST:C:\Users\Ярик\Desktop\BKP\Avia.fdb (LOCALHOST:C:\Users\Ярик\Desktop\BKP\AVIA.FDB)

Поля | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол | Comparison | To-do

ID SMALLINT NOT NULL

#	ПК	ВК	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка	Коллате
1	<input checked="" type="checkbox"/>			ID	SMALLINT						<input checked="" type="checkbox"/>		
2			<input checked="" type="checkbox"/>	NAME	VARCHAR		25				<input type="checkbox"/>	WIN1251	WIN1251

Описание поля | Объекты, использующие поле

Рисунок В.2 - Создание справочника «Должности»

Таблица : [SOTRUDNIKI] : LOCALHOST:C:\Users\Ярик\Desktop\BKP\Avia.fdb (LOCALHOST:C:\Users\Ярик\Desktop\BKP\AVIA.FDB)

Поля | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол | Comparison | To-do

ID SMALLINT NOT NULL

#	ПК	ВК	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка	Коллате
1	<input checked="" type="checkbox"/>			ID	SMALLINT						<input checked="" type="checkbox"/>		
2			<input checked="" type="checkbox"/>	FIO	VARCHAR		150				<input type="checkbox"/>	WIN1251	WIN1251
3			<input checked="" type="checkbox"/>	DOLJNOST	VARCHAR		25				<input type="checkbox"/>	WIN1251	WIN1251
4				MOBTEL	VARCHAR		15				<input type="checkbox"/>	WIN1251	WIN1251

Описание поля | Объекты, использующие поле

Рисунок В.3 - Создание справочника «Сотрудники»

Таблица : [SELL] : LOCALHOST:C:\Users\Ярик\Desktop\BKP\Avia.fdb (LOCALHOST:C:\Users\Ярик\Desktop\BKP\AVIA.FDB)

Таблица | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол | Comparison | To-do

Количество записей | SELL

Поля | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол | Comparison | To-do

ID SMALLINT NOT NULL

#	ПК	ВК	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка	Коллате
1	1			ID	SMALLINT						<input checked="" type="checkbox"/>		
2				NOMER_R	SMALLINT						<input type="checkbox"/>		
3				FIO_K	VARCHAR		150				<input type="checkbox"/>	WIN1251	WIN1251
4				NOMER_B	SMALLINT						<input type="checkbox"/>		
5				SOTRUDNIK	VARCHAR		150				<input type="checkbox"/>	WIN1251	WIN1251
6				DATA	DATE						<input type="checkbox"/>		
7				NAME_C	VARCHAR		25				<input type="checkbox"/>	WIN1251	WIN1251
8				OPLATA	INTEGER						<input type="checkbox"/>		

Описание поля | Объекты, использующие поле

Рисунок В.4 - Создание справочника «Продажа»

Таблица : [SAMOLET] : LOCALHOST:C:\Users\Ярик\Desktop\BKP\Avia.fdb (LOCALHOST:C:\Users\Ярик\Desktop\BKP\A...

Таблица | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол

Количество записей | SAMOLET

Поля | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол

ID SMALLINT NOT NULL

#	ПК	ВК	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка
1	1			ID	SMALLINT						<input checked="" type="checkbox"/>	
2				MODEL	VARCHAR		25				<input type="checkbox"/>	WIN1251
3				KOLVO_PASS	INTEGER						<input type="checkbox"/>	

Описание поля | Объекты, использующие поле

Рисунок В.5 - Создание справочника «Самолет»

Таблица : [REIS] : LOCALHOST:C:\Users\Ярик\Desktop\BKP\Avia.fdb (LOCALHOST:C:\Users\Ярик\Desktop\BKP\AVIA.FDB)

Таблица | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол | Comparison | To-do

Количество записей | REIS

Поля | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол | Comparison | To-do

ID SMALLINT NOT NULL

#	ПК	ВК	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка	Колла
1	1			ID	SMALLINT						<input checked="" type="checkbox"/>		
2				COMPANY	VARCHAR		25				<input type="checkbox"/>	WIN1251	WIN12
3				SAMOLET	VARCHAR		25				<input type="checkbox"/>	WIN1251	WIN12
4				GOROD_OTPR	VARCHAR		50				<input type="checkbox"/>	WIN1251	WIN12
5				GOROD_PRIIB	VARCHAR		50				<input type="checkbox"/>	WIN1251	WIN12
6				DATA_OTPR	DATE						<input type="checkbox"/>		
7				DATA_PRIIB	DATE						<input type="checkbox"/>		
8				TIME_OTPR	TIME						<input type="checkbox"/>		
9				TIME_PRIIB	TIME						<input type="checkbox"/>		
10				COST_BILET	INTEGER						<input type="checkbox"/>		

Описание поля | Объекты, использующие поле

Рисунок В.6 - Создание справочника «Рейс»

Таблица : [KLIENT] : LOCALHOST:C:\Users\Ярик\Desktop\BKP\Avia.fdb (LOCALHOST:C:\Users\Ярик\Desktop\BKP\AVIA.FDB)

Таблица | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол | Comparison | To-do

Количество записей | KLIENT

Поля | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол | Comparison | To-do

ID SMALLINT NOT NULL

#	ПК	ВК	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка	Коллате
1	PK			ID	SMALLINT						<input checked="" type="checkbox"/>		
2			U	FIO	VARCHAR		150				<input type="checkbox"/>	WIN1251	WIN1251
3				SERIA_P	INTEGER						<input type="checkbox"/>		
4				NOMER_P	INTEGER						<input type="checkbox"/>		
5				EMAIL	VARCHAR		25				<input type="checkbox"/>	WIN1251	WIN1251
6				MOBTEL	VARCHAR		25				<input type="checkbox"/>	WIN1251	WIN1251

Описание поля | Объекты, использующие поле

Рисунок В.7 - Создание справочника «Клиент»

Таблица : [KLASS] : LOCALHOST:C:\Users\Ярик\Desktop\BKP\Avia.fdb (LOCALHOST:C:\Users\Ярик\Desktop\BKP\AVIA.FDB)

Таблица | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол | Comparison | To-do

Количество записей | KLASS

Поля | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол | Comparison | To-do

ID SMALLINT NOT NULL

#	ПК	ВК	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка	Коллате
1	PK			ID	SMALLINT						<input checked="" type="checkbox"/>		
2			U	NAME	VARCHAR		25				<input type="checkbox"/>	WIN1251	WIN1251
3				COST	FLOAT						<input type="checkbox"/>		

Описание поля | Объекты, использующие поле

Рисунок В.8 - Создание справочника «Класс цен»

Таблица : [GOROD] : LOCALHOST:C:\Users\Ярик\Desktop\BKP\Avia.fdb (LOCALHOST:C:\Users\Ярик\Desktop\BKP\AVIA.FDB)

Таблица | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол | Comparison | To-do

Количество записей | GOROD

Поля | Ограничения | Индексы | Зависимости | Триггеры | Данные | Master/Detail View | Описание | Скрипт | Права | Протокол | Comparison | To-do

ID SMALLINT NOT NULL

#	ПК	ВК	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка	Коллате
1	PK			ID	SMALLINT						<input checked="" type="checkbox"/>		
2			U	NAME	VARCHAR		50				<input type="checkbox"/>	WIN1251	WIN1251

Описание поля | Объекты, использующие поле

Рисунок В.9 - Создание справочника «Город»