

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 информационной системы взаимодействия
с клиентами call-центра ООО «ФИТ»**

Выпускная квалификационная работа

**студентки очной формы обучения
направления подготовки 09.03.03. Прикладная информатика
4 курса группы 07001205
Кизиловой Ольги Геннадиевны**

Научный руководитель
к.т.н., доцент Гахова Н.Н.

БЕЛГОРОД 2016

Содержание

Введение.....	3
1 Аналитическая часть.....	6
1.1 Технико-экономическая характеристика предметной области.....	6
1.1.1 Характеристика предприятия.....	6
1.1.2 Характеристика работы менеджера по взаимодействию с клиентами	8
1.2 Экономические аспекты необходимости автоматизации задачи	10
1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи	15
1.4 Постановка задачи.....	16
1.4.1 Цель и назначение автоматизированного варианта решения задачи	16
1.4.2 Общая характеристика организации решения задачи на компьютере.....	17
1.5 Анализ существующих разработок и обоснование выбора технологии проектирования.....	18
2 Обоснование проектных решений.....	22
2.1 Обоснование проектных решений по техническому обеспечению.....	22
2.2 Обоснование проектных решений по информационному обеспечению	23
2.3 Обоснование проектных решений по программному обеспечению	24
2.4 Обоснование проектных решений по технологическому обеспечению	26
2.5 Обоснование выбора программных средств.....	27
3 Проектная часть.....	29
3.1 Информационное обеспечение задачи автоматизированной системы взаимодействия с клиентами	29
3.1.1 Информационная модель и ее описание	29
3.1.2 Характеристика первичных документов с нормативно-справочной и входной оперативной информацией	31
3.1.3 Характеристика базы данных.....	32
3.1.4 Характеристика выходной информации	35
3.2 Программное обеспечение задачи автоматизированной системы	36
3.2.1 Общие положения	36
3.2.2 Структурная схема пакета и описание программных модулей	38
3.3 Технологическое обеспечение задачи автоматизированной системы	40
3.4 Описание контрольного примера реализации выпускной работы	44
3.5 Расчет экономической эффективности.....	54
Заключение.....	60
Список использованных источников.....	62
Приложение А.....	67
Приложение Б	69
Приложение В.....	94

Введение

С каждым годом рынок товаров и услуг увеличивает объем предложений. В столь стремительном потоке постоянно появляющихся новых компаний и успешно развивающихся старых, необходимо увеличивать объем услуг и производительность труда, при этом не снизив качества товаров и услуг.

Для таких целей производится полный анализ предприятия, его сильных и слабых сторон. Определив наиболее уязвимую и нуждающуюся в реорганизации работы область организации, производится поиск возможных путей решения.

Общество с ограниченной ответственностью «Фабрика информационных технологий», успешно работающая уже много лет на рынке информационных услуг города Белгород, предлагает комплекс услуг в сфере информационных технологий по России. Компания постоянно находится в развитии и стремится усовершенствовать работу своих отделов. На данный момент времени ООО «ФИТ» необходимо автоматизировать процесс работы отдела маркетинга и менеджмента, а именно call-центра компании. Работа с клиентами, их привлечение и дальнейшее удержание, т.е. продолжительный срок востребованности в услугах фирмы, является одним из решающих факторов дальнейшего существования организации. В связи с чем, создание программного продукта, позволяющего автоматизировать работу контактного центра, особенно необходимо.

Таким образом, важность темы выпускной квалификационной работы подтверждается необходимостью постоянного совершенствования работы контактного центра в условиях конкурентной борьбы для устойчивости положения фирмы на рынке.

Актуальность работы заключается в следующем:

- сокращение ручных операций, ускорение обработки и получения информации, повышение точности учета;

- снижение количества ошибок, которые могут совершаться в процессе работы сотрудниками предприятия;
- усовершенствование процесса сбора, передачи, обработки, хранения, защиты целостности информации и выдачи результатов расчетов конечному пользователю

Объектом исследования является общество с ограниченной ответственностью «Фабрика информационных технологий».

Предметом – система взаимодействия с клиентами менеджеров call-центра.

Цель выпускной работы – снижение временных затрат в работе сотрудников call-центра и увеличение производительности отдела менеджмента и маркетинга.

Для выполнения поставленной цели были определены следующие задачи:

- изучить деятельность call-центра организации;
- проанализировать слабые стороны работы отдела и определить наиболее подходящие пути решения выявленных недостатков;
- выполнить проектирование информационной модели call-центра;
- разработать автоматизированную систему взаимодействия с клиентами;
- выполнить тестирование системы;
- рассчитать экономическую эффективность.

Данные задачи реализовываются в трех разделах выпускной квалификационной работе.

В первом разделе:

- определено и описано технико-экономическое состояние предприятия в выбранном для исследования отделе менеджмента и маркетинга call-центра ООО «ФИТ»;

- продемонстрирована работа контактного центра как в модели IDF0, представляющую собой подсистему отражающую взаимодействие между объектами и функциями, так и IDF3, которая показывает причинно-следственные связи между ситуациями и событиями;
- проанализирован способ работы сотрудников, а также выявлены проблемы, устранить которые возможно путем автоматизации определенных процессов фирмы;
- описаны цель и назначение автоматизированного варианта, общая характеристика организации решения задачи;
- проанализированы существующие разработки для автоматизации работы контактных центров.

Во втором разделе произведено обоснование выбора средств, методов и технологий для создания автоматизируемой системы, определены основные требования к различным видам обеспечения. Результатом данного раздела являются обоснованные проектные решения для разрабатываемого программного приложения.

В третьем разделе описан процесс проектирования и разработки системы взаимодействия с клиентами. Отражены структура разработанной информационной системы, функциональные возможности и принцип работы, выполнено тестирование. Наглядно представлены результаты работы разработанного программного продукта для ООО «ФИТ». Рассчитана экономическая эффективность системы.

В заключении произведен анализ действий, произведенных в выпускной квалификационной работе и сделаны соответствующие выводы. Описана степень достижения поставленных целей и задач.

Данная работа состоит из 65 страниц, 33 рисунков, 10 таблиц, 33 литературных источников и 3 приложений.

1 Аналитическая часть

1.1 Технико-экономическая характеристика предметной области

1.1.1 Характеристика предприятия

Общество с ограниченной ответственность «Фабрика информационных технологий», успешно работающая уже много лет на рынке информационных услуг города Белгород, предлагает комплекс услуг в сфере информационных технологий по России.

Клиентами компании являются администрации городов России, не только крупные и известные российские компании, но и небольшие компании, частные предприниматели.

Целью компании «Фабрика информационных технологий», как и любого предприятия, является получению финансовой прибыли. Основными направлениями, позволяющими заработать денежные средства, являются создание веб-сайтов, разработка программного обеспечения, создание трехмерной анимации, продвижение сайтов в сети Интернет, разработка фирменного стиля, дизайна логотипов, буклетов и плакатов, IT-консалтинг, предоставление сервиса по рассылке SMS сообщений, разработка и воплощение различного рода проектов для компаний и предприятий.

Выпускная квалификационная работа связана с оказанием услуг по проекту «Городские парковки», которые осуществляет данная фирма. Для многих городов России создаются системы управления паркоматами – устройствами для оплаты парковки, установленными на платных автостоянках. Данная система предназначена для специальных отделов администрации городов, так же есть программа для клиентов, предназначенная для поиска парковок в определенном городе и свободных мест на них, оплаты парковки.

Система новая и всегда могут возникнуть вопросы по ее использованию, для чего необходим отдельный отдел взаимодействия с клиентами. Кроме того,

нужно следить за тем, чтобы все автомобилисты, занимающие места дольше, чем предоставленное бесплатное время, оплачивали парковку в соответствии со указанными на паркоматах ценами. Для данных целей в фирме был создан контакт-центр, так же именуемый как call-center. На сегодняшний день данный отдел обслуживает жителей Белгорода и Белгородской области, Твери, Екатеринбурга, Тулы.

Общество с ограниченной ответственностью «Фабрика информационных технологий», как и любая фирма, организация, разбита на различные уровни контроля и подчиненности сотрудников. Структура предприятия представлена на рисунке 1.

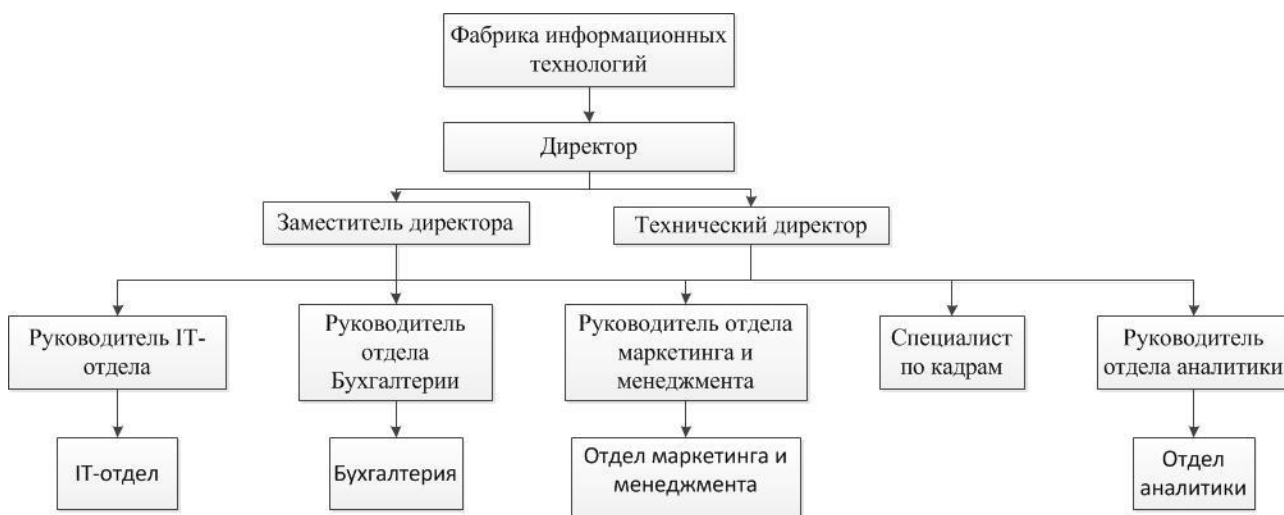


Рисунок 1 – Структура ООО «ФИТ»

В выпускной квалификационной работе для автоматизации был выбран отдел Маркетинга и менеджмента (контактный центр фирмы), структура которого представлена на рисунке 2.



Рисунок 2 – Структура отдела маркетинга и менеджмента

Описание работы сотрудников контактного центра по должностям, представленным на рисунке 2, производится в подразделе 1.1.2.

1.1.2 Характеристика работы менеджера по взаимодействию с клиентами

Менеджер по взаимодействию с клиентами (Account manager) – специалист, в должностные обязанности которого входит обслуживание клиентов.

Основная задача менеджера по работе с клиентами – проведение работ по консультированию.

В отделе имеется главный менеджер (администратор), его заместитель и подчиненные ему менеджеры.

В обязанности администратора входит:

- обучение сотрудников, демонстрирование их обязанностей, проведение тренингов и помощь в работе с трудными клиентами;
- осуществление руководства работниками контактного центра;
- ведение базы данных сотрудников и ее редактирование;
- ведение и редактирование базы данных информации по вопросам;
- получение и создание оповещений о внесенных изменениях в любую базу данных;

- составление отчетов по деятельности работников контактного центра, по городам и по внесенным изменениям;
- отчет руководству по произведенной работе call-центра.

В обязанности менеджера входит:

- принятие входящих звонков, осуществление исходящих, консультирование клиентов;
- применение изученных методик общения с клиентами на практике;
- просмотр всех существующих баз данных;
- оповещение администратора о необходимости внесения изменений в базу данных.

Заместитель руководителя отдела выполняет обязанности менеджера по взаимодействию с клиентами, а также оказывает помощь в работе руководителя отдела.

Менеджер работает с клиентами по следующей схеме:

- на контактный номер менеджера поступает звонок;
- сотрудник принимает звонок, если его телефонная линия не занята.

Иначе клиент пересылается свободному менеджеру. Если таких нет, тогда звонок занимает очередь. Если звонок по каким-либо причинам сорвался, тогда менеджером осуществляется исходящий вызов на номер клиента.

- клиент задает интересующий вопрос, для получения нужной информации;
- менеджер отвечает на него или же задает дополнительный вопрос, чтобы более корректно ответить;
- клиент отвечает на заданные вопросы и получает подробный ответ.

При работе с клиентом менеджеру поступает следующая информация:

- город, в котором находится клиент;
- причина, по которой был осуществлен звонок:
 - а) способ оплаты парковки;
 - б) время работы парковок или время, предоставляемое на оплату;

- с) получение льгот на пользование парковочными местами;
- д) получение абонеента;
- е) другая информация;
- прочие вопросы клиента о парковках города.

На данный момент информация о принятых, пропущенных звонках и их количестве находится на сервере в виде списка, отфильтрованного по дням, но не отфильтрованного по сотрудникам.

Так же менеджеры получают фотографии с мест парковок, которые показывают, сколько времени и какая машина пробыла на стоянке, а также не нарушены ли были правила парковки.

Если стоянка была не оплачена, то по номеру машины имеется возможность отправить данные в администрацию города или сообщить, позвонив.

Каждую неделю администратор составляет отчеты о проделанной работе. В отчетах фиксируется количество работы, произведенной каждым сотрудником. При использовании имеющихся технологий составления отчетов тратится около 15% рабочего времени на то, чтобы перенести имеющуюся и распределенную по дням информацию с сервера в MS Excel и выбрать данные, касающиеся одного сотрудника.

1.2 Экономические аспекты необходимости автоматизации задачи

При анализе работы предприятия была выявлена необходимость оптимизировать работу контактного центра. Информацией, которой пользуются сотрудники, трудно найти, в связи с чем клиентам иногда приходится некоторое время ожидать ответ, что увеличивает очередь поступающих звонков и уменьшает лояльность клиентов. Создание отчетов также занимает большое количество времени администратора. В связи с неудобным для использования способом получения и хранения информации

увеличивается риск создания и получения ошибочных данных. Все эти факторы в целом может негативно сказаться на фирме в целом.

Исходя из всего вышеописанного, была поставлена задача повышения качества обслуживания клиентов и работы фирмы ООО «ФИТ». Данная задача относится к классу задач «управления взаимоотношениями с клиентами» и необходима для увеличения продуктивности взаимодействия с клиентами, обращающимися в контактный центр фирмы. Результаты решения данной задачи окажут благоприятное воздействие на работу структурного подразделения. Поэтому такая задача является важной и неотъемлемой частью контактной деятельности и корректного принятия решений, относящихся к call-центру. Информация для решения задачи предоставляется главным менеджером в текстовом формате. Такими данными являются список сотрудников, перечень вопросов и ответов по различным парковкам городов, вид формы отчетов.

Система главным образом будет предназначена для:

- повышения эффективности реализации основных функций контактного центра, поскольку деятельность данного отдела связана с большим информационным потоком;
- улучшения оперативности взаимодействия с клиентами;
- повышения производительности труда;
- снижения количества ошибок при консультировании клиентов, изменении базы данных, составлении отчетов;
- создания эффективного и безопасного хранения;
- разграничения доступа к информации.

Отражение процессов, происходящих в контактном центре, выполнено с помощью программы AllFusion ERwin Data Modeler (ранее ERwin) – CASE-средства для проектирования и документирования баз данных, которое позволяет создавать, документировать и сопровождать базы данных, хранилища и витрины данных [23].

Описание работы центра было произведено в IDEF0 – методологии функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов [23].

Все используемые сотрудниками программы, входящие и исходящие данные, механизмы и управление отделения менеджмента и маркетинга показаны на рисунке 3.

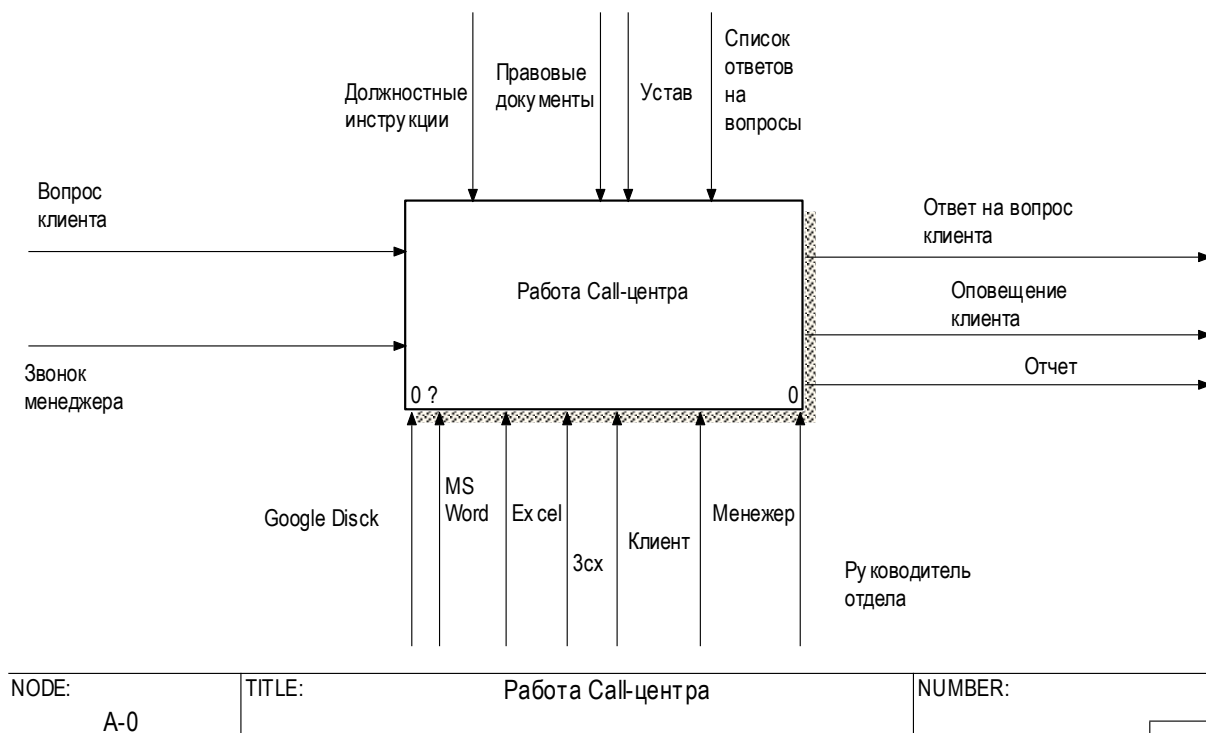


Рисунок 3 – Контекстная диаграмма работы контактного центра

Для более подробного описания процессов, происходящих в деятельности сотрудников call-центра, была создана диаграмма первого уровня, представленная на рисунке 4.

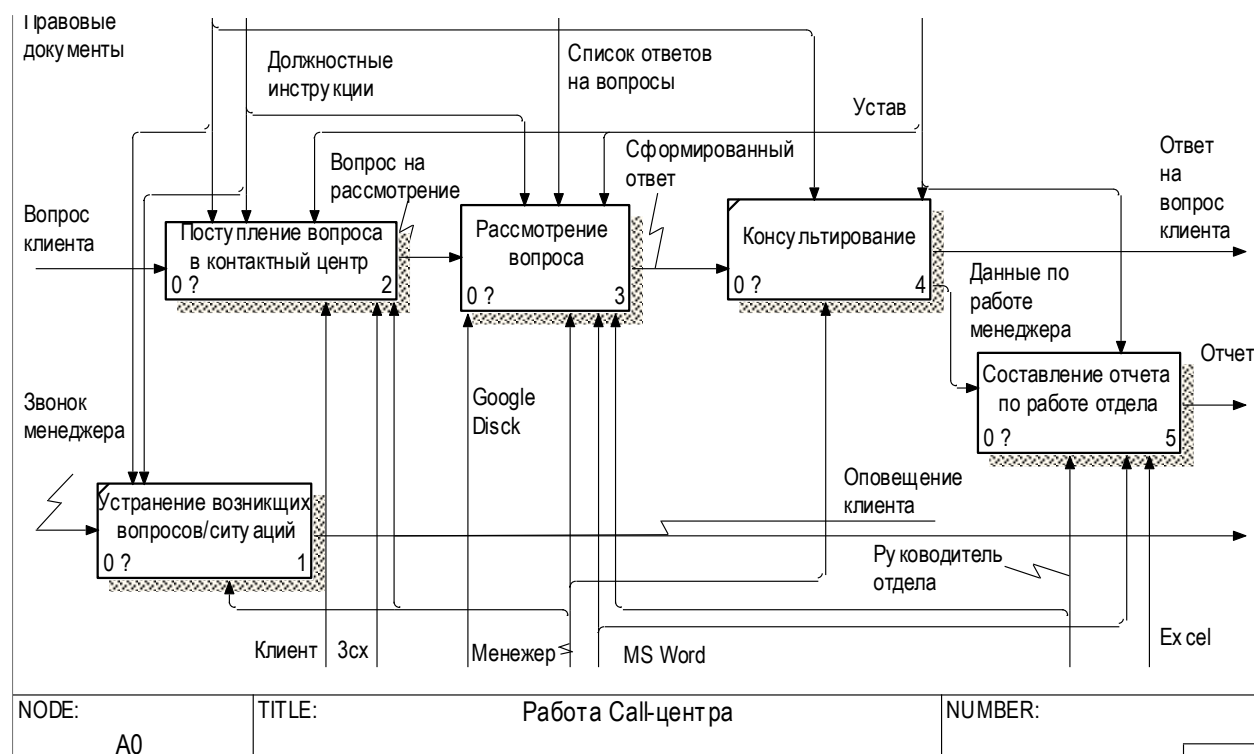


Рисунок 4 – Диаграмма декомпозиции первого уровня

Процесс работы контактного центра представлен в виде пяти основных функций: поступление вопроса в контактный центр, рассмотрение вопроса, консультирование, составление отчета по работе отдела, устранение возникающих вопросов/ситуаций.

Целью каждой функции является учет выполняемых в рамках данного процесса действий и отражение их результата в проектируемой информационной системе. Коротко прокомментируем каждую из них.

Процесс поступления вопроса в контактный центр производится клиентом, посредством осуществления звонка в фирму по указанному на паркоматах телефону. Сам же процесс приема звонка осуществляется менеджером по взаимодействию с клиентами через программу 3CX – основанной на открытых стандартах программной IP АТС, оснащенная “прямо из коробки” полным набором унифицированных коммуникаций [28].

Можно сказать, что данный процесс максимально автоматизирован. В 3CX руководитель отдела определяет длину очереди, в которую попадает клиент, если все операторы в данный момент заняты. Так же программа ищет

свободного оператора и направляет к нему входящий звонок. Данная область не нуждается в автоматизации процесса работы.

Результатом процесса рассмотрения вопроса является найденный и сформулированный ответ на поставленный клиентом вопрос. На данный момент этап рассмотрения вопросов слабо автоматизирован. Менеджерам по взаимодействию с клиентами приходится долго искать информацию среди документов MS Word, хранящихся на Google Drive, что занимает 5-10% рабочего времени. Данное время тратится впустую, а клиенты, ожидающие очереди, могут не дождаться ответа оператора. Все это снижает производительность контактного центра. Поэтому именно данная область была выбрана предметом автоматизации.

Все предыдущие действия производились для выполнения этапа консультирования. При консультировании менеджер подробно объясняет, как решить вопрос клиента или куда необходимо обратиться для решения сложившейся ситуации.

Составление отчета по работе отдела является одним из важнейших компонентов, поскольку показывает необходимость и результативность данного отдела в ООО «ФИТ». Результатом являются показатели работоспособности каждого из сотрудников за определенный период, т.е. число принятых/пропущенных вызовов. На данный момент руководителю отдела для составления отчета необходимо выполнять ряд действий, занимающих большое количество времени, таких как поиск информации по сотруднику в документах MS Word, хранящихся на Google Disc, а также перенос данных в MS Excel. Данную область так же можно автоматизировать, создав в системе события, отражающие поступление звонков.

При устранении возникающих вопросов/ситуаций сотрудниками производится одно из двух действий:

- консультирование с руководителем отдела;

– осуществления исходящего звонка. Происходит если клиент не оплатил парковочное место и в ситуациях, требующих вмешательства. Результатом таких звонков служит получение штрафа клиентом.

Декомпозиции 2-го уровня диаграммы модели «Как есть» представлены в приложении А на рисунках А1-А3.

1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи

На данный момент в фирме используется для связи с клиентами программа 3CX Phone System for Windows 7.

3CX Phone System – программное решение, которое заменяет традиционную аппаратную АТС на очень простую в управлении IP-АТС, построенную на открытых отраслевых стандартах [28].

Хранение информации по принятым, отклоненным оператором звонков и их общем количестве в день выполняется с использованием документов, созданных в MS Word. Для получения ответа на вопрос клиентов, происходит доступ к данным, хранящимся в облачном хранилище Google Drive for work.

Google Drive for Work – это премиум-версия Диска в составе Google Apps for Work, представленная 25 июня 2014 года на конференции Google I/O. Она включает неограниченное пространство для хранения, расширенные функции аудита и формирования отчетов, а также службы предоставления данных для электронных расследований [29].

Автоматизированная система (АС) для корректной работы call-центра отсутствует. По отдельности применяемые программы не являются эффективным способом организации контактного центра компании.

Для вышеописанной работы call-центра экономически целесообразным является использование вычислительной техники для увеличения производительности работы.

Целями использования вычислительной техники для решения выявленной проблемы являются:

- повышение производительности труда как всего отдела, так и отдельных его сотрудников;
- улучшение качества обслуживания клиентов;
- увеличение конкурентоспособности.

Результатом применения техники станет небольшой рост компании на рынке, а также прирост клиентов.

1.4 Постановка задачи

1.4.1 Цель и назначение автоматизированного варианта решения задачи

В выпускной квалификационной работе осуществляется процесс создания информационной системы взаимодействия с клиентами, предназначенной для автоматизации деятельности менеджеров и систематизации информации о паркоматах и парковках городов.

Основными целями разработки служат:

- сокращение времени обработки входной информации и получения результатных показателей;
- повышение степени достоверности выходной информации;
- снижение трудоемкости и количества ошибок при консультировании и анализе работы отдела.

В программе должны быть реализованы такие функции как:

- разграничение прав доступа к информации;
- хранение, получение и редактирование данных относительно сотрудников, выполняемых ими действий (событий), парковочных мест в городах Белгород, Тверь, Екатеринбург, Тула;
- составление отчетов, в зависимости от требований администратора;

- обеспечение целостности хранимой информации.

Вышеописанные функции в системе выполняются в виде отдельных подзадач.

Автоматизированная работа с информацией происходит в базе данных с использованием экранных форм, таких как окна просмотра и изменения в списках сотрудников, производимых ими операций, парковок городов и окно формирования отчетов.

Ведение условно-постоянной информации, хранящейся в таблицах базы данных происходит через:

- редактирование данных;
- добавление записей;
- удалении записей.

Получение информации, которая содержится в базе, выполняется по сотрудникам, производимых ими действиям и парковкам городов.

Формирование отчетов производится по запросам главного менеджера в виде итоговой отчетности по сотруднику.

1.4.2 Общая характеристика организации решения задачи на компьютере

Информационная система имеет клиент-серверную архитектуру.

Серверная часть будет представлять собой базу данных, разработанную в системе управления базой данных (СУБД), которая будет находиться на сервере компании, написанного вручную на языке Python.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода [32].

Клиентская часть предназначена для конечного пользователя, т.е. для менеджеров ООО «ФИТ», и должна обеспечивать доступ к информации согласно предоставленным правам, ввод новой информации, просмотр, редактирование, а также формирование отчетов.

После внедрения разрабатываемой системы работа менеджера принципиально не изменится – будет оптимизирован участок, помогающий ускорить и увеличить достоверность информации при работе с клиентами. Кроме того, главному менеджеру не придется выполнять достаточно неудобную и трудоемкую работу по составлению еженедельного отчета по проделанной сотрудниками центра работе.

Проектируемая система поможет менеджерам грамотнее организовать работу и выдавать при консультировании более точную информацию.

Руководитель отдела сможет редактировать всю информацию из базы данных, просматривать результаты работы не только по окончанию дня по всем менеджерам, а еще и информацию по одному отдельному сотруднику.

Более того, чтобы просмотреть, добавить информацию, внести изменения или удалить, теперь не надо будет тратить время. Достаточно будет открыть программу и выполнить необходимое действие. А менеджерам не придется долго искать и запоминать, где именно произошли изменения.

1.5 Анализ существующих разработок и обоснование выбора технологии проектирования

Изучая способы решения выявленных недостатков работы контактного центра, был определен наиболее подходящий продукт, автоматизирующий действия, выполняемые менеджерами. Такой является система управления взаимоотношениями с клиентами (CRM).

CRM (Customer Relationship Management) – прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов [33].

TAdviser – информационный веб-портал о российских и иностранных ИТ-решениях, технологиях и поставщиках информационных систем информатизации и интеграции [27]. В ноябре 2015 года на данном сайте был опубликован список наиболее популярных CRM-систем для компаний.

По данным TAdviser, по количеству внедрений лидирует решение Bpm`online. За весь период наблюдений в базе TAdviser насчитывается более 320 проектов, выполненных с помощью этой системы. Следует отметить, что 85% всех CRM-проектов, информация о которых содержится в базе, выполнены в России. Второе место занимает Microsoft Dynamics CRM – 303 проекта. Третье – решение, созданное на базе 1С. Это 1С:CRM (246 проектов соответственно) [27]. Рассмотрим данные системы более подробно.

BPMonline CRM – это веб-приложение, распространяемое по модели SaaS (Software as a Service). Модель предоставления программного обеспечения, как услуги, позволяет заказчикам ускорить развертывание, сократить затраты на внедрение и поддержку, получив гарантированно высокий уровень безопасности и доступности данных.

В зависимости от потребностей и политик компании может быть выбран один из вариантов развертывания BPMonline CRM: доступ пользователей к системе, размещенной на сервере сертифицированного провайдера (On-Demand), или на собственных мощностях клиента (On-Site).

Плюсы:

- отсутствие значительных начальных затрат – стоимость лицензий в аренду составляет 250 евро в год;
- мощный дата-центр, размещенный в г. Франкфурт (Германия), гарантирует доступ к BPMonline CRM в режиме 24x7 из любой точки мира.

Минусы:

- нельзя работать с системными таблицами. Например, нельзя создать пользователя системы;
- принудительная постраничность при возврате списка объектов. По умолчанию запрос возвращает первые 40 объектов;

- главный же минус – высокая цена. Минимальная стоимость продукта – 250 евро в год на одного пользователя. Кроме того, заказать данную CRM менее чем на год нельзя;

- сложность внедрения.

Microsoft Dynamics CRM – это одна из наиболее мощных программ управления бизнесом. Ее можно настроить под любые требования компании. Microsoft Dynamics CRM пользуется большой популярностью в крупных компаниях. Востребована как для сферы продажи услуг, так и для товарной сферы. Систему можно использовать для работы множества менеджеров по продажам, однако для этого потребуются тонкая настройка и наличие специальных знаний по установке.

Плюсы:

- доступ с любых устройств через Интернет;
- возможность настройки под любые задачи;
- надежная сохранность данных.

Минусы:

- сложность настройки;
- большая стоимость.

1С CRM – это система, работающая в комбинации с ресурсами 1С:Предприятие. Ее использование полностью автоматизирует процесс ведения бизнеса. Система идеально подойдет для крупных организаций, занимающихся продвижением товаров, особенно с большим ассортиментом.

1С CRM имеет несколько версий программы: демонстрационную и профессиональную. Каждая из них включает в состав своих функций возможность интеграции и обмена информацией с другими 1С продуктами.

Плюсы:

- интеграция с 1С:Предприятие;
- удобный интерфейс;

- полное и прозрачное управление всеми процессами на предприятии;

- мощная система анализа данных.

Минусы:

- необходимость продления лицензии продукта;
- сложность внедрения и использования для мелких и средних организаций;

- длительный срок обучения сотрудников, в связи с большим набором функций и особенностями программы.

Для данной предметной области существует множество приложений. Одни из них адаптируемые под особенности предприятия, другие – нет.

ООО «ФИТ» требуется не просто CRM-система, но также необходимо в его составе иметь подсистему, хранящую ответы на вопросы клиентов.

С точки зрения программной реализации разрабатываемая система взаимодействия с клиентами будет иметь следующие преимущества относительно вышеописанных систем:

- бесплатное эксплуатирование;
- легкое внедрение;
- простота в использовании;
- наличие ролей и разграничение прав доступа;
- дружественный интерфейс;
- соответствие требованиям компании.

Таким образом, в первом разделе было определено технико-экономическое состояние предприятия в выбранном отделе менеджмента и маркетинга, проанализирован способ работы сотрудников, а также выявлены изъяны, устранить которые возможно будет путем автоматизации определенных процессов фирмы. Кроме того, были определены цель и назначение автоматизированного варианта и общая характеристика организации решения задачи.

2 Обоснование проектных решений

2.1 Обоснование проектных решений по техническому обеспечению

Техническое обеспечение (ТО) – совокупность технических средств, компьютерной техники, средств передачи информации, используемых в автоматизированных системах управления и в информационных системах [11].

Создаваемая система автоматизации работы call-центра будет состоять из набора функций, имеющего стандартные требования к техническому обеспечению компьютера.

Данная программа задействует некоторый объем памяти, и при его недостатке может произойти зависание процесса работы компьютера и потеря данных. Также системе потребуется производительный процессор и необходимые для реализации полных возможностей программного продукта устройства, с установленными на соответствующих компьютерах драйверами.

Для данной системы минимальными требованиями к конфигурации компьютера являются:

- объем оперативной памяти не менее 1,5 Мб;
- процессор не ниже Intel Pentium не менее 1 ГГц.;
- операционная система не менее Windows 7;
- сетевой адаптер;
- манипулятор «Мышь».

Кроме того, необходимо наличие следующих параметров:

- компьютерная сеть;
- доступ к Интернету.

Чтобы в разрабатываемой автоматизированной информационной системе была возможность одновременной работы нескольких сотрудников,

потребуется наличие локальной корпоративной компьютерной сети на основе сервера.

Дополнительных параметров конфигурации для функционирования программы на персональных компьютерах не потребуется.

Для реализации полного функционала системы может потребоваться принтер любой версии любого производителя.

Использование вышеперечисленных проектных решений по техническому обеспечению предоставит возможность выполнять поставленные задачи на должном уровне и пользоваться предоставляемым программным продуктом эффективно.

2.2 Обоснование проектных решений по информационному обеспечению

Информационное обеспечение (ИО) автоматизированной системы – совокупность форм документов, классификаторов, нормативной базы и реализованных решений по объемам, размещению и формам существования информации, применяемой в АС при ее функционировании [8].

Структура информационного обеспечения:

- методические инструктивные материалы;
- информационная база (БД);
- система управления базой данных.

В данной системе информационным обеспечением является база данных. Информация из таблиц базы данных будет вводиться и выводиться через экранные формы, которые проектируются и создаются до начала внедрения системы. Данное решение актуально, так как состав информации используемой информационной системой постоянен и не будет изменяться во время её эксплуатации.

Для решения поставленных задач предлагается использовать свободную реляционную систему управления базами данных MySQL, с которой можно работать не только в текстовом режиме, но и в графическом. Доступ к данной базе для осуществления взаимодействия с ней будет выполняться специальное программное средство. Из множества возможных вариантов была выбрана система FireBirds.

Firebird (FirebirdSQL) – кроссплатформенная система управления базами данных, работающая на Mac OS X, Linux, Microsoft Windows и разнообразных Unix платформах [30].

Данный программный продукт позволяет значительно упростить работу с MySQL, зарегистрировав базу данных в любой совместимой с Firebird системе управления базами данных.

Доступ к базе данных будет осуществляться на различных компьютерах через установленную автоматизированную систему взаимодействия с клиентами и с помощью серверной сетевой технологии. Храниться она будет на сервере компании. Исходя из вышеописанного способом организации информационной базы будет являться централизованно хранимая интегрированная многопользовательская БД в компьютерной сети с «клиент-серверной» архитектурой.

В системе необходимо получать отчеты по заданным параметрам. Однако регламент к данной операции не предоставлен, т.к. общего стандарта на него нет.

Состав экранных форм определялся потребностью в выполнении определенных функций сотрудниками контактного центра.

2.3 Обоснование проектных решений по программному обеспечению

Программное обеспечение (ПО) – все или часть программ, процедур, правил и соответствующей документации системы обработки информации [9].

Структура ПО компьютеров в call-центре компании состоит из системного и прикладного обеспечения. Исходя из чего были определены требования к создаваемой системе:

- надежность;
- дружелюбность и простота интерфейса;
- гибкость;
- открытость;
- эффективность;
- бюджетность.

При определении языка программирования были учтены основополагающие факторы системы, такие как объектно-ориентированный подход и клиент-серверная технология.

Объектно-ориентированное программирование (ООП) – методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования [10].

Для доступа к базе данных была выбрана утилита IVExpert, поскольку для подключения из готового приложения к базе не потребуется установки дополнительных программ, а также из-за простоты в использовании и общедоступности.

IVExpert – GUI-оболочка, предназначенная для разработки и администрирования баз данных InterBase и Firebird, а также для выбора и изменения данных, хранящихся в базах [31].

Учитывая выбранные в подразделе 2.2 технологии доступа к БД, из наиболее подходящих для разработки клиентской части текущей системы предпочтение было отдано изученной в процессе обучения в университете программе – C++ Builder 6.0.

C++ Builder Borland – программный продукт, инструмент быстрой разработки приложений (RAD), интегрированная среда программирования

(IDE), система, используемая программистами для разработки программного обеспечения на языке программирования C++ [13].

Система объектно-ориентированного программирования C++ Builder рассчитана на работу в операционных систем Windows. Интегрированная среда C++ Builder позволяет работать не только в режиме консольного приложения, но и визуальном. Это дает возможность продуктивно использовать компоненты, улучшенные инструментами и разномасштабными средствами доступа к базам данных.

Система объектно-ориентированного программирования C++ Builder также поддерживает серверные сценарии и интеграцию с другими языками, что позволяет расширить функционал создаваемых программ.

2.4 Обоснование проектных решений по технологическому обеспечению

Технологическое обеспечение – EDP (Electronic Data Processing) – это совокупность методов и средств сбора, хранения, передачи, обработки и защиты информации на базе вычислительной техники и средств коммуникаций [25].

Технологический процесс обработки данных – это комплекс взаимосвязанных операций по преобразованию информации в процессе решения функциональных задач на ЭВМ, начиная с момента возникновения информации и до получения на ее основе исходных данных [26].

Технологический процесс обработки данных включает [26]:

- подготовительный этап, на котором осуществляется подготовка к решению задачи (создание справочников, введение в память компьютера необходимых постоянных данных и др.);
- начальный этап, связанный с операциями по сбору, регистрации и размещению документов в базовые массивы (возможна обработка документов,

заполненных вручную, однако более эффективным является электронное документирование);

– основной, завершающий этап работы, обеспечивающий получение необходимых отчетных форм, когда из компьютерной базы данных извлекаются рабочие массивы, подлежащие группировке по соответствующим ключевым признакам, подсчету по ним итоговых данных с распечаткой в дальнейшем полученных отчетных документов.

С внедрением разрабатываемой ИС недостатки, перечисленные в разделах 1.2 и 1.3, будут устранены, посредством выполнения определенных действий, описанных ниже.

Вся информация будет находится в одной базе данных, а не в разрозненных файлах MS Word. В разрабатываемой ИС будут удобные инструменты для хранения и изменения результатов.

Отчеты будут автоматически генерироваться системой на основе вводимой менеджером информации о звонках, встречах и их результатах.

Все вышеперечисленное позволит сократить время на выполнение ежедневных операций, увеличит производительность работы менеджера. Также работу менеджера можно будет контролировать в режиме реального времени.

2.5 Обоснование выбора программных средств

Для обоснования выбора программных средств были определены классы технологий проектирования, которые включают следующие:

– каноническое проектирование является методом ручного проектирования, использующее минимум типовых решений;

– индустриальное проектирование построено на использовании современных средств автоматизации и подразделяется на:

- a) автоматизированное проектирование;
- b) типовое проектирование.

Создавая систему предполагается использование следующих компонентов:

- case-средств;
- объектно-ориентированной технологии;
- системы управление базами данных;
- среды программирования.

Каждый из вышеописанных пунктов подразумевает использование стандартных, типовых решений, автоматизирующих процессы проектирования и разработки информационной системы.

Исходя из чего предпочтение было отдано технологии индустриального проектирования.

Таким образом, во втором разделе были выбраны средства, методы и технологии для создания системы, определены основные требования к обеспечению. Были определены и обоснованы проектные решения для программного средства, предпочтительного в связи с выявленной проблемой. Эти данные позволили перейти к непосредственной разработке автоматизированной системы взаимодействия с клиентами call-центра ООО «ФИТ».

3 Проектная часть

3.1 Информационное обеспечение задачи автоматизированной системы взаимодействия с клиентами

3.1.1 Информационная модель и ее описание

В разделе 1.3 при анализе технико-экономической составляющей предприятия была определена модель «Как есть», выявлены недостатки и уязвимости в работе контактного центра фирмы. Поэтому, перед тем как приступить к созданию информационно системы, необходимо определить область, которая в дальнейшем автоматизируется информационной системой, а также сам процесс работы call-центра с учетом предполагаемых нововведений.

Для выполнения вышеописанных действий в выпускной квалификационной работе использовалась функциональная модель «Как будет». Она позволяет выявить распределение ресурсов между операциями делового процесса, что дает возможность оценить эффективность использования ресурсов после предлагаемого реинжиниринга.

Описание процессов, происходящих в отделе контактного центра, производится с помощью программы Erwin в методологии IDF0.

Данные по работе контактного центра, отраженные с помощью модели «Как будет», представлены на контекстной диаграмме на рисунке 5.

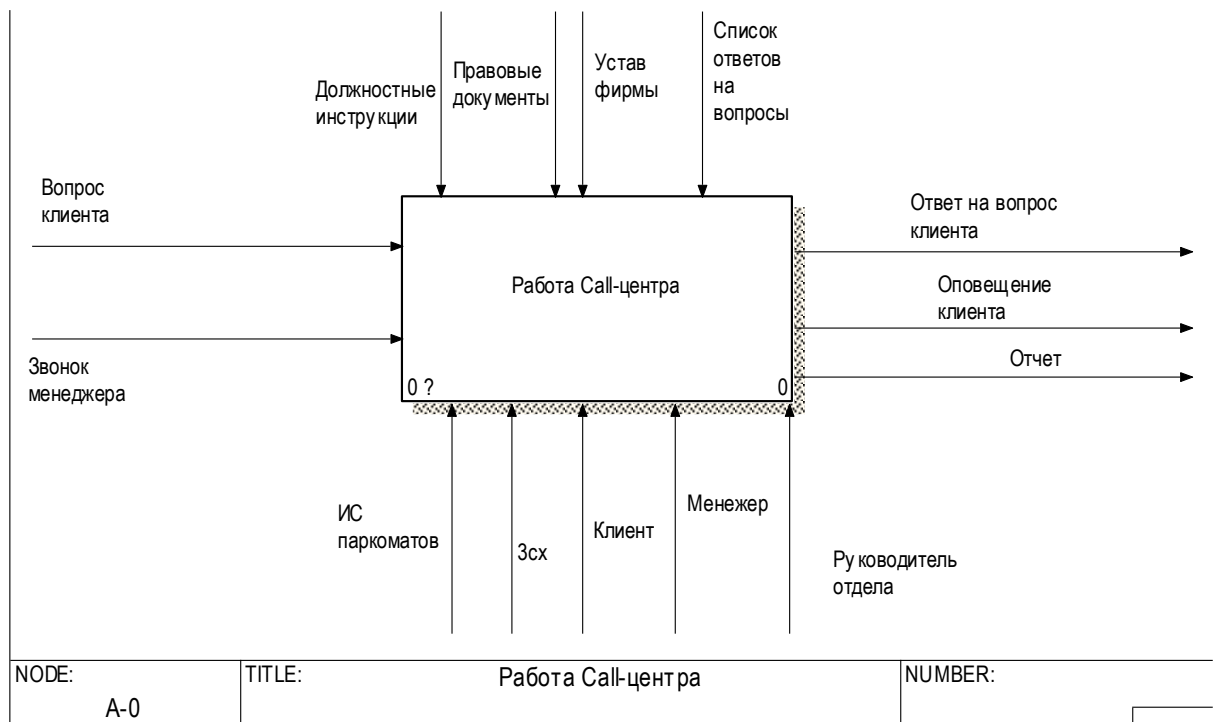


Рисунок 5 – Контекстная диаграмма

Для более подробного описания процесса работы call-центра используется функциональная декомпозиция второго уровня системы, которая приведена на рисунке 6.

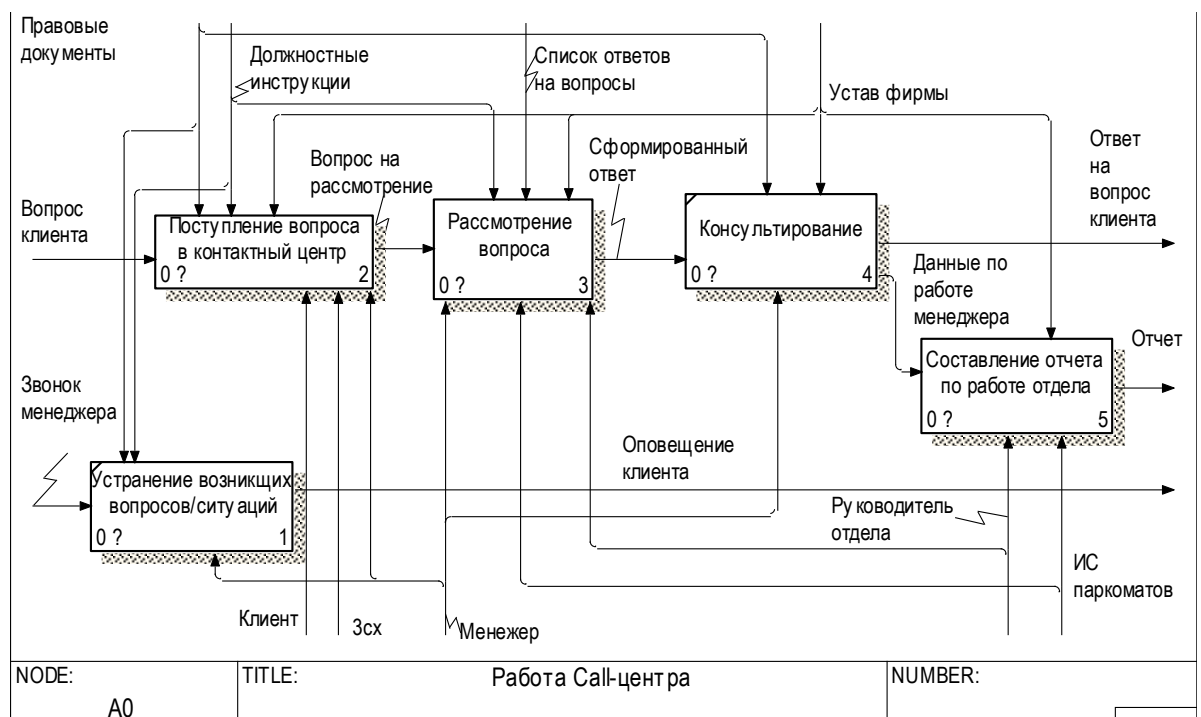


Рисунок 6 – Диаграмма декомпозиции первого уровня

Данная декомпозиция имеет 5 основных блоков. Они полностью совпадают с блоками модели «Как есть». Однако, потоков-механизмов стало меньше, что наглядно показывает, какие именно действия и программы должна заменить информационная система.

Декомпозиции 2-го уровня диаграммы модели «Как будет» представлены в приложении А на рисунках А4-А6.

Автоматизация работы происходит на этапах «Рассмотрение вопроса» и «Составление отчета по работе отдела». Все стандартные операции, совершаемые сотрудниками, в MS Word и Excel теперь находятся в одном приложении и предоставляются в удобной форме.

3.1.2 Характеристика первичных документов с нормативно-справочной и входной оперативной информацией

Первичным документом для программы являются списки сотрудников и ответов на задаваемые клиентами вопросы.

В системе входные данные делятся на данные для менеджеров и для руководителя отдела.

В первом случае, это информация о событиях (действиях, произведенных работниками) и городе, в котором возник вопрос по парковкам.

Во втором, входными данными являются данные о сотрудниках, событиях, парковках.

Во входную информацию о сотрудниках входят такие данные, как уникальный номер, фамилия, имя, отчество.

В данные о событиях входят вид события, этап выполнения, дата начала, дата окончания, описание события, создатель поручения, ответственный за событие.

Входная информация о парковках содержит сведения о городе, способе оплаты парковочных мест и их дополнительные параметры, такие как номер телефона и сайт, времени работы, времени, в течении которого необходимо

произвести оплату, способ получения льгот и номер телефона для обращений, стоимость парковки, способ получения абонемента.

Входная информация обновляется в процессе работы программы, могут произойти какие-либо корректировки, которые необходимо предусмотреть, такие как изменение данных, увольнение или прием работника и т.д.

3.1.3 Характеристика базы данных

На этапе инфологического проектирования была создана модель «сущность-связь» («Entity-Relationship») для будущей базы данных, которая показана на следующем рисунке 7.

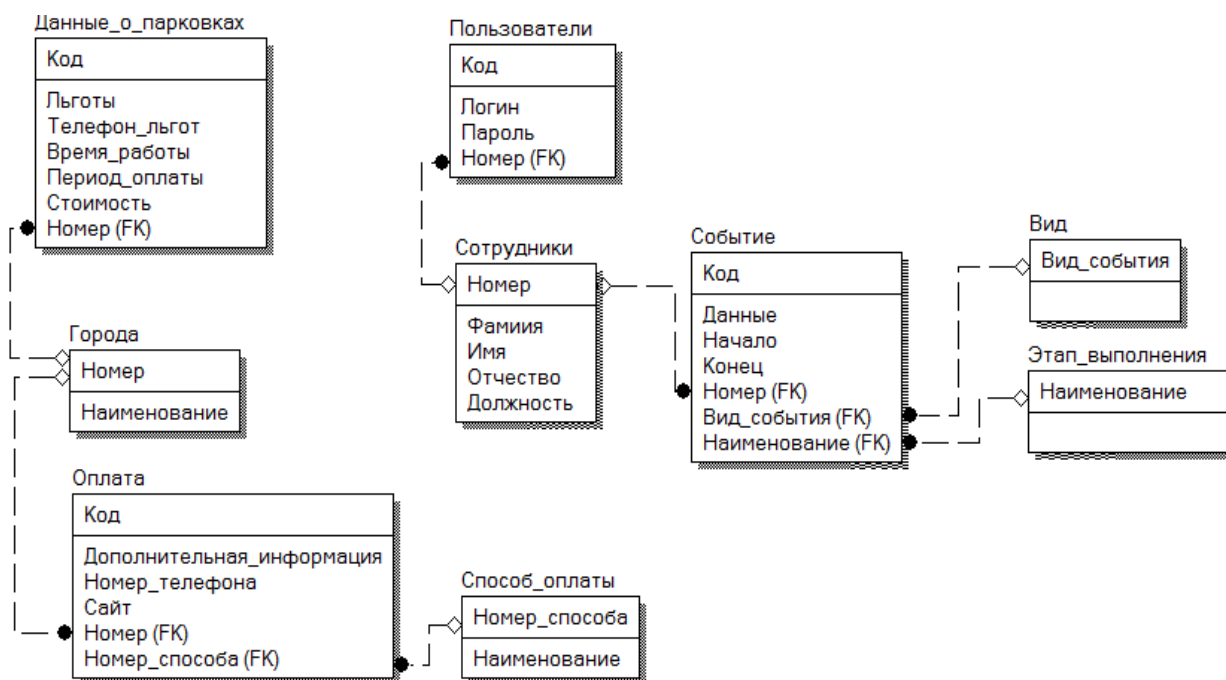


Рисунок 7 – Логическая схема БД

После создания логической модели данных, необходимо приступить к разработке физической модели.

Спроектированная база данных была приведена к третьей нормальной форме. Это означает, что:

- значения всех атрибутов в таблицах преобразованы в атомарные – содержащие только одну запись;
- первичный ключ разделен на части так, чтобы некоторые из ключевых полей зависели только от одной из таких частей;
- созданы новые таблицы для каждой части ключа и перемещены поля, зависящие от них;
- осуществлено логическое разделение на родительские и дочерние сущности, а затем определены идентифицирующие связи.

Физическая модель изображена на рисунке 8.

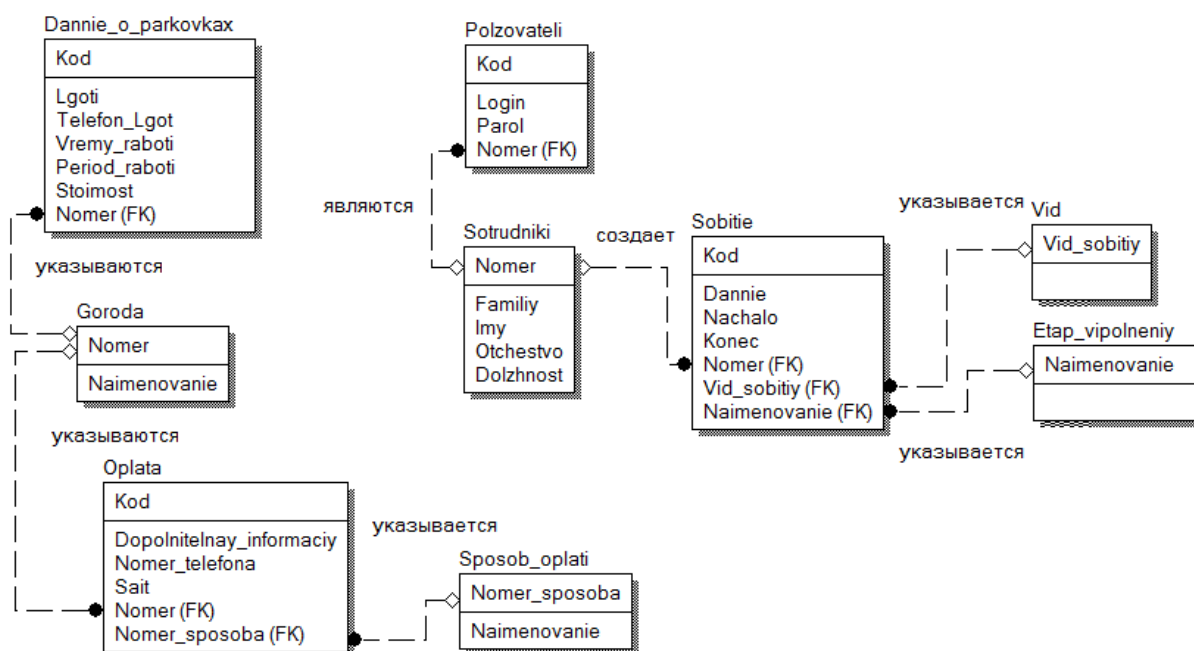


Рисунок 8 – Физическая модель БД

Созданные модели позволяют нагляднее представить будущую структуру автоматизированной системы взаимодействия с клиентами и наиболее полно описывают связи между объектами.

В таблице 1 описаны более подробно каждая сущность модели базы данных, связанные с парковками городов.

Таблица 1 – Описание сущностей, связанных с парковками

Сущность	Идентификатор таблицы	Атрибут	Идентификатор поля	Тип поля
Данные о парковках	DOPOLN_OPL	Код данных	DOPOLN_OPL_KOD	Smallint
		Данные льгот	DOPOLN_OPL_LGOTI	Varchar
		Номер телефона для получения льгот	DOPOLN_OPL_NOME RTEL	Char
		Время работы	DOPOLN_OPL_VREM	Char
		Период оплаты	DOPOLN_OPL_WREM	Char
		Стоимость парковки	DOPOLN_OPL_STOIM	Char
		Номер города	GOROD_KOD	Smallint
Город	GOROD	Номер города	GOROD_KOD	Smallint
		Наименование города	GOROD_GOROD	Char
Способы оплаты парковок город	G_OPL	Номер вида оплаты в городе	G_OPL_KOD	Smallint
		Номер города	GOROD_KOD	Smallint
		Способ оплаты	OPLAT_KOD	Smallint
		Параметры оплаты	G_OPL_DOPOLN	Char
		Номер для оплаты	G_OPL_NOM	Char
		Сайт для оплаты	G_OPL_SAIT	Char
Способы оплаты	OPLAT	Номер способа	OPLAT_KOD	Smallint
		Наименование	OPLAT_NAIM	Char

В таблице 2 описаны более подробно каждая сущность модели базы данных, связанные с событиями и сотрудниками.

Таблица 2 – Описание сущностей, связанных с сотрудниками

Сущность	Идентификатор таблицы	Атрибут	Идентификатор поля	Тип поля
1	2	3	4	5
Пользователи	POLS	Номер пользователя	POLS_KOD	Smallint
		Логин	POLS_LOGIN	Char
		Пароль	POLS_PAROL	Smallint
		Сотрудник	SOTRUDNIKI_ID	
Сотрудники	SOTRUDNIKI	Номер	SOTRUDNIKI_ID	Smallint
		Уникальный код сотрудника	SOTRUDNIKI_KOD	Char
		Фамилия	SOTRUDNIKI_FAM	Char
		Имя	SOTRUDNIKI_IM	Char
		Отчество	G_OPL_NOM_OTCH	Char
		Должность	G_OPL_SAIT_DOLZH	Char

Продолжение таблицы 2

1	2	3	4	5
События	SOBITIE	Номер события	SOBITIE_KOD	Smallint
		Вид события	VIDSOS_VID	Char
		Описание	SOBITIE_NAIM	Varchar
		Дата начала события	SOBITIE_DATAN	Timestamp
		Дата окончания события	SOBITIE_DATAK	Timestamp
		Создатель события	SOTRUDNIKI_ID	Smallint
		Ответственный за событие	SOTRUDNIKI_ID	Smallint
		Этап выполнения	PARAM_NAIM	Char
Состояние	VIDSOS	Наименование	VIDSOS_VID	Char
Этап выполнения	PARAM	Наименование	PARAM_NAIM	Char

Данные таблицы являются наглядным представлением таблиц базы данных и параметров данных, хранящихся в них.

3.1.4 Характеристика выходной информации

В автоматизированной системе выходной информацией являются данные о парковках городов, событиях (действиях, произведенных сотрудниками), выведенные на экранные формы программы.

Данные о парковках поступают из таблиц «Города», «Данные о парковках», «Способ оплаты», «Оплата», где хранятся основные сведения по парковочному пространству различных городов. Информация формируется на экранной форме при выборе города и активизировании кнопки «Получить».

Данные о событиях поступают из таблиц «Событие», «Сотрудник», «Этап выполнения», «Состояние события». Полный состав данных получается путем формирования экранной формы с внесенными данными и нажатия на кнопку «Сохранить». Информация в полном виде необходима для руководителя отдела, который следит за оперативностью и качеством работы

сотрудников. В дальнейшем эти данные можно получить в выходном документе «Отчет по работе сотрудника».

В автоматизированной информационной системе результатным документом является «Отчет по работе сотрудника».

Этот документ получается в результате обработки информации из таблиц «Событие», «Этап выполнения», «Состояние события» и «Сотрудники». При этом производится сортировка по определенному сотруднику. Имеется возможность выбора данных в зависимости от состояния события. Данный документ остается на фирме и подшивается в отдельную папку для ведения отчетности по производительности как работника, так и всего отдела менеджмента и маркетинга.

3.2 Программное обеспечение задачи автоматизированной системы

3.2.1 Общие положения

При разработке ИС, решающей задачи автоматизации, важным этапом является описание иерархии функций управления и обработки данных разрабатываемого программного продукта

Дерево функций представляет собой иерархическую структуру действий, реализованных в ИС. Все действия программного продукта можно разделить на служебные и основные функции.

Служебные функции являются одинаковыми для всех автоматизированных систем на данном этапе реализации аппаратных и программных средств. Функции этого вида необходимы для обеспечения безопасности ввода, обработки и хранения информации, упрощения работы с системой, выполнение ее в наиболее удобной и незатруднительной для конечного пользователя форме.

Основные функции управления и обработки данных встречаются как в информационных системах любой специализации, так и в данной

автоматизированной системе. Данные действия показывают особенности процесса обработки информации, получения результатов, ведения информационной базы. Организация этих функций производилась с учетом условий обработки и управления данной предметной области.

Состав и классификация функций разрабатываемого программного продукта представлены в виде дерева функций на рисунке 9.



Рисунок 9 – Дерево функций ИС

В представленном на рисунке 9 дереве функций четко прослеживается будущий функционал автоматизированной системы взаимодействия с клиентами. Именно эти действия должны быть реализованы в системе.

Рассмотрев основные функции информационной системы, разберем, какие приложения производит основные операции в каждом из экранных форм. На рисунке 10 представлен сценарий диалога программного продукта с пользователем.

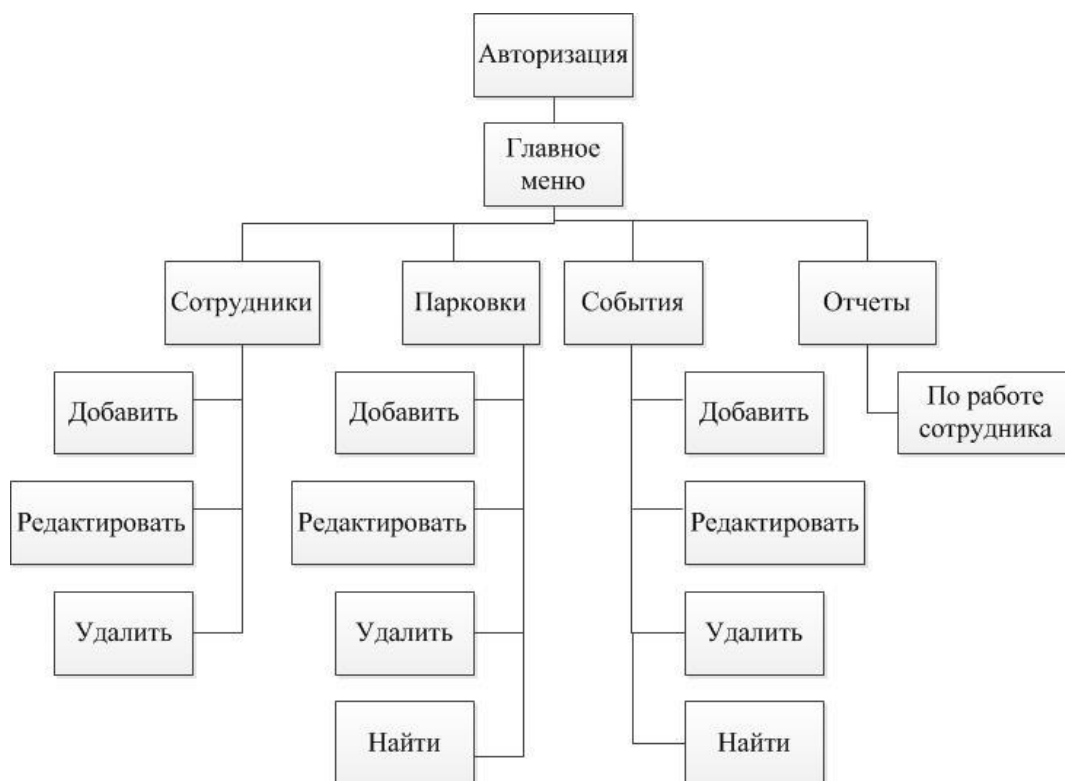


Рисунок 10 – Сценарий диалога

Данные рисунки наглядно показывают функциональные возможности программы и пути обращения к определенным операциям.

3.2.2 Структурная схема пакета и описание программных модулей

На основе результатов, полученных ранее в подразделе 3.2.2, были созданы два дерева программных модулей: для администратора (руководителя отдела менеджмента и маркетинга) и подчиненных ему менеджеров. Структурная схема пакета представлена на рисунках 11 и 12 для описанных выше сотрудников соответственно.

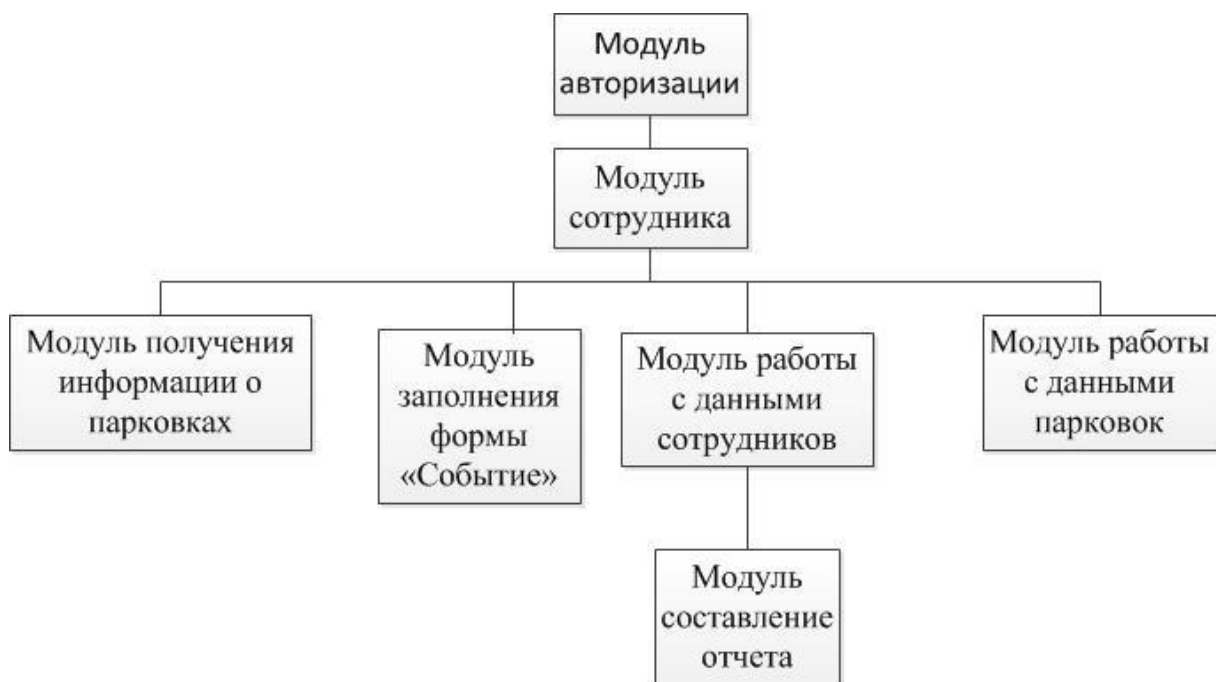


Рисунок 11 – Дерево вызова процедур для администратора

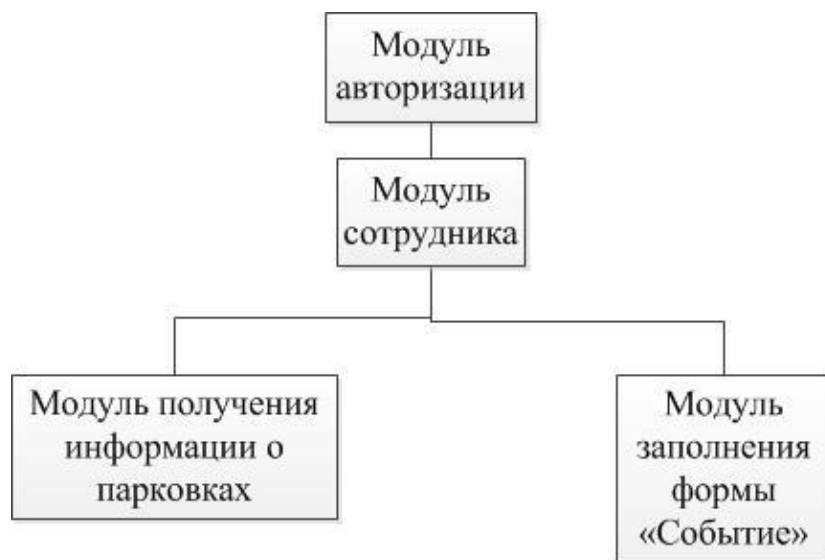


Рисунок 12 – Дерево вызова процедур для менеджера

Каждый из модулей содержит основные и необходимые функции и процедуры для работы с данными таблиц, относящимися к нему.

Характеристики системных модулей представлены в таблице 3.

Таблица 3 – Описание программных модулей

№	Наименование модуля	Функции модуля
1	Модуль авторизации	Содержит процедуры и функции для авторизации пользователя в системе.
2	Модуль сотрудника (главный модуль)	Из данного модуля осуществляются все операции. Содержит процедуры для перехода на другие модули.
3	Модуль получения информации о парковках	Имеет функции для поиска данных, в соответствии с выбранным городом.
4	Модуль заполнения формы «Событие»	Состоит из 2-х частей: сохранения и изменения данных. Остальные функции находятся в модуле сотрудника.
5	Модуль работы с данными парковок	Содержит процедуры, позволяющие вести работу с базой данных парковок. Переход на данный модуль осуществляется из главного модуля.
6	Модуль работы с данными сотрудников	Содержит процедуры, позволяющие вести работу с базой данных сотрудников. Переход на данный модуль осуществляется из главного модуля.
7	Модуль составления отчета	Основное предназначение модуля – создание отчета по работе выбранного сотрудника.

Все вышеописанные модули составляют экранные формы системы, а также ее функционал. Каждый из них разграничен по роли сотрудника в программе.

3.3 Технологическое обеспечение задачи автоматизированной системы

Технологический процесс машинной обработки информации является совокупностью операций, выполняемых строго в определенном порядке для получения желаемого результата. Он состоит из 4 основных этапов работы с информацией: сбор, передача, обработка и выдача.

Первый из них задействует механизмы сбора исходных данных, их регистрацию и передачу системе. После чего все сведения, которые принимаются или регистрируются, подвергаются операциям контроля и переноса на машинные носители, которые являются процедурами второго этапа, а именно передачи.

Обработка является основным этапом в технологическом процессе. Его выполнение задействует процессы способа работы и процедуры выбора действия над информацией.

Последним этапом является выдача информации. На данном шаге выгружается вся обработанная информация в удобном для пользователя виде в разработанной системе.

Данные четвертого этапа отражены на рисунках 13-18.



Рисунок 13 – Схема работы системы

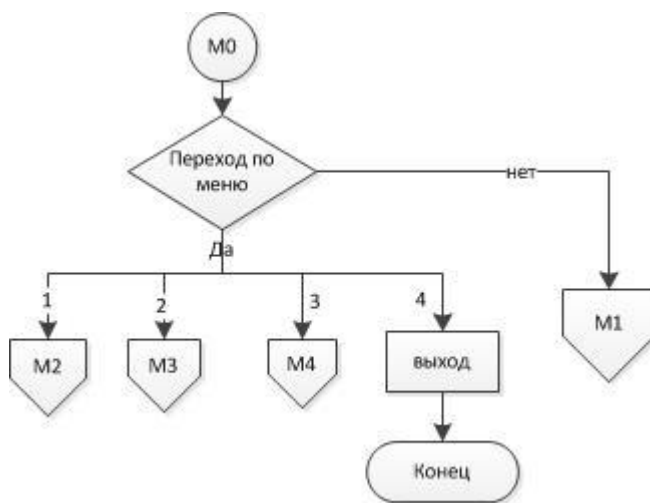


Рисунок 14 – Основной модуль

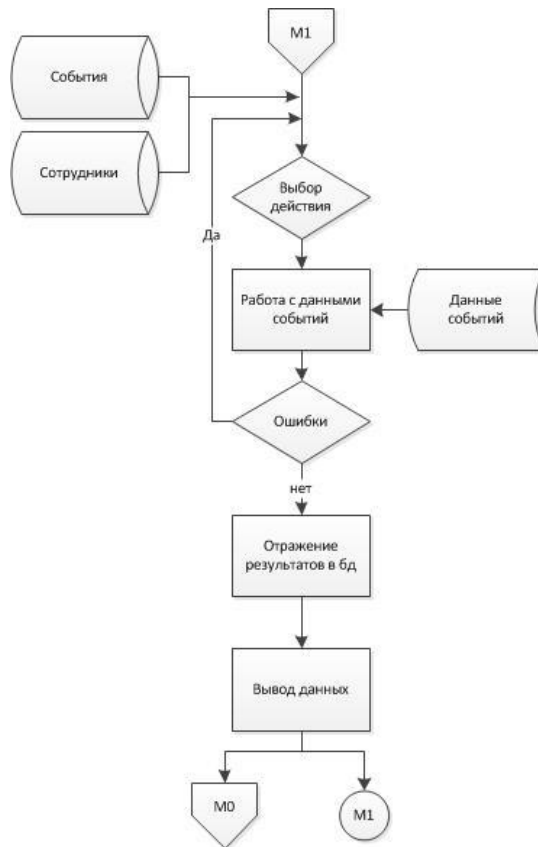


Рисунок 15 – Модуль работы с событиями

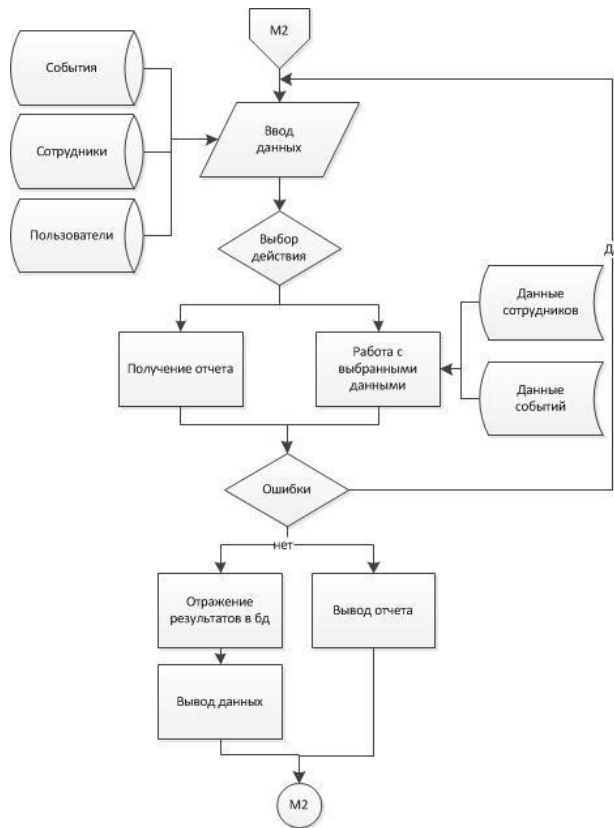


Рисунок 16 – Модуль работы с данными сотрудников

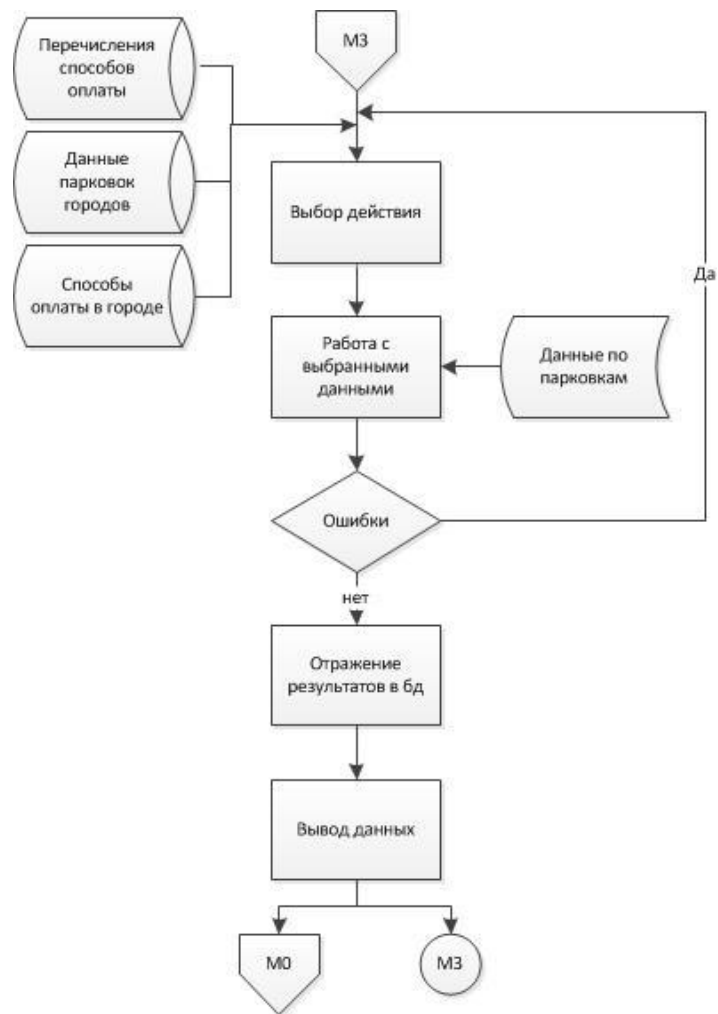


Рисунок 17 – Модуль работы с данными парковок

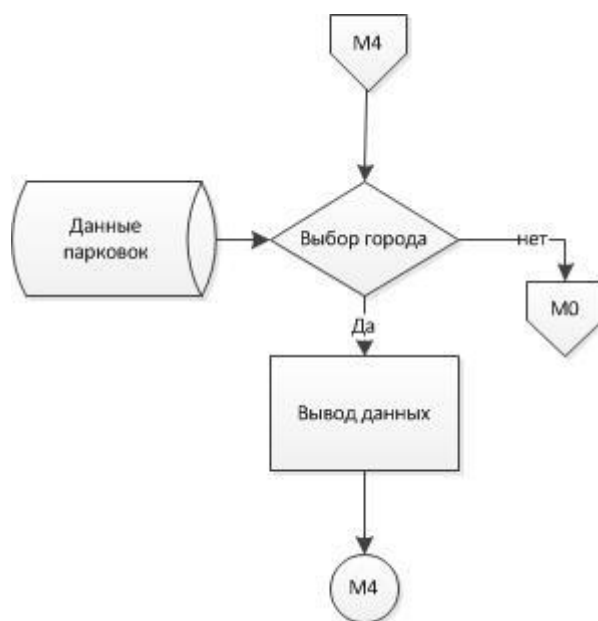


Рисунок 18 – Модуль получения данных по парковкам

Рисунки демонстрируют четыре основных этапа работы с информацией: сбор, передача, обработка и выдача. Благодаря чему, можно наглядно разобрать работу автоматизированной системы взаимодействия с клиентами.

3.4 Описание контрольного примера реализации выпускной работы

Автоматизированная система запускается из файла FIT.exe. При запуске приложения, пока прогружаются все необходимые файлы, на мониторе отображается заставка приложения, представленная в приложении В рисунок В1. После чего, перед входом сотрудника в личный кабинет происходит процесс авторизации сотрудника, показанный на рисунке 19.

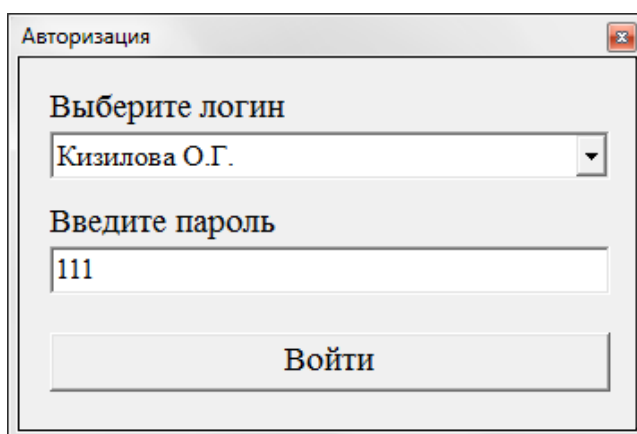
The image shows a standard Windows-style dialog box titled "Авторизация". Inside the dialog, there are three main components: a label "Выберите логин" above a dropdown menu showing "Кизилова О.Г."; a label "Введите пароль" above a text input field containing "111"; and a large button at the bottom labeled "Войти". The dialog has a close button in the top right corner.

Рисунок 19 – Экранная форма авторизации

В программе предусмотрен ввод ошибочных данных. Если введен неверный пароль, программа выдаст сообщение об ошибке, представленное в приложении В на рисунке В2.

После того, как авторизация прошла успешно, открывается главное окно программы, содержащее информацию о сотруднике и его событиях. Личный кабинет сотрудника представлен на рисунке 20.

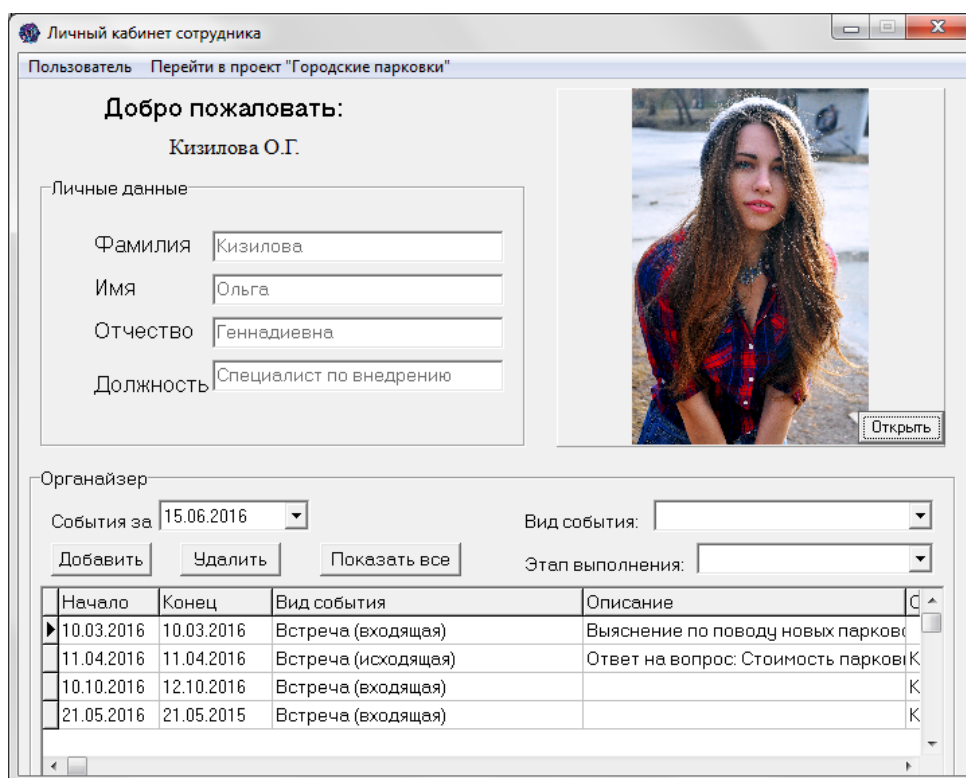


Рисунок 20 – Главная форма системы

На рисунке видно, что личные данные сотрудника недоступны. Также недоступны параметры меню, такие как пункт «Сотрудники» и в пункте «Перейти в проект “Городские парковки”» подпункт «Редактировать информацию». Редактирование данных сотрудника и вышеперечисленные пункты меню открыты для использования только руководителю отдела.

При совершении или планировании какого-либо события менеджеру по взаимодействию с клиентами необходимо будет отразить это в органайзере событий. Для этого следует нажать на кнопку «Добавить». Далее необходимо заполнить все важные параметры. Открывшееся окно с введенными параметрами показано на рисунке 21.

Рисунок 21 – Форма добавления нового события

После нажатия на кнопку «Сохранить», будет выведено сообщение о том, что запись добавлена и событие отразится в главном окне пользователя в организаторе. Полученный результат отражен на рисунке 22.

Начало	Конец	Вид события	Описание
10.03.2016	10.03.2016	Встреча (входящая)	Выяснение по поводу новых парковок
11.04.2016	11.04.2016	Встреча (исходящая)	Ответ на вопрос: Стоимость парковки
10.10.2016	12.10.2016	Встреча (входящая)	
10.10.2016	11.10.2016	Телефонный звонок (исходящий)	Сегодня созвонились. Решить пробл

Рисунок 22 – Форма добавления нового события

Для изменения или просмотра события необходимо нажать на соответствующее событие в таблице. При этом откроется форма, показанная на рисунке 21. Однако, вместо кнопки «Сохранить» на экранной форме приложения будет расположена кнопка «Перезаписать». Принцип изменения записи похож на ее добавление. Вносим новые данные, вместо старых и нажимаем на кнопку «Перезаписать». После чего будет получено сообщение об изменении или ошибке. Если запись отредактирована успешно, результат отразится в таблице органайзера главной формы.

Для удаления события необходимо выделить событие, нажав на него один раз и активировать кнопку «Удалить». После чего появится надпись: «Запись удалена» и изменятся данные таблицы органайзера событий.

События в системе сортируются по сотрудникам, т.е. в окно «Органайзер» выводятся те сведения, где данный работник является ответственным лицом или создателем поручения.

В системе имеется возможность поиска информации. Таблицу событий можно отфильтровать по виду и этапу выполнения. На рисунке 23 представлен результат поиска события «Встреча (исходящая)».

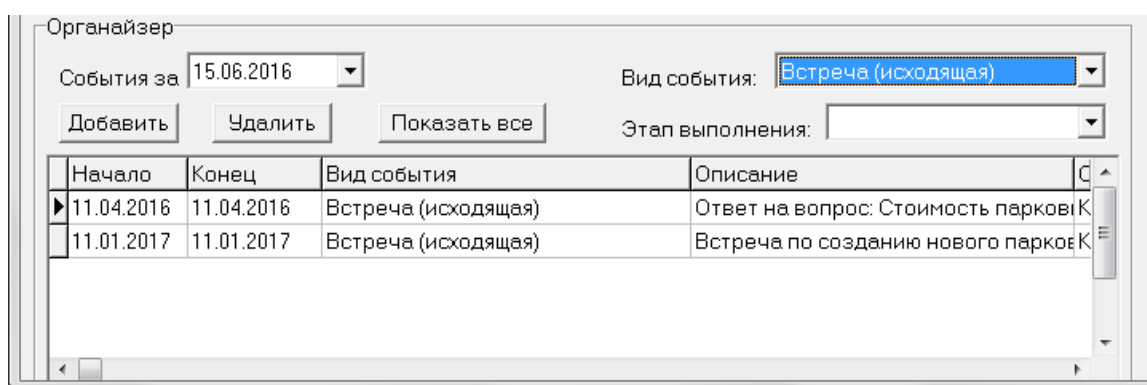


Рисунок 23 – Поиск события

Поиск события сотрудника по параметру «Этап выполнения» представлен в приложении В на рисунке В3.

Для отмены поиска требуется нажать кнопку «Показать все».

Для менеджера по взаимодействию с клиентами, не являющегося администратором, в пункте меню «Перейти в проект «Городские парковки»» открыт доступ к «Получить данные».

При активации данного пункта меню происходит открытие формы «Ответно-вопросная система», где, выбрав город, можно узнать:

- способы оплаты парковки;
- время работы парковок и их стоимость;
- способ получения льгот на пользование платными парковками;
- способ приобретения абонемента.

Результат работы данной формы отражен на рисунках 24-25.

The screenshot shows a web application window titled "Ответно-вопросная страница". At the top, there is a dropdown menu labeled "Веберите город:" with "Белгород" selected. Below this are four tabs: "Оплата парковки", "Время работы парковок и стоимость", "Льготы на пользование платными парковками", and "Приобретение абонемента". The "Оплата парковки" tab is active. The main content area is titled "Вы можете оплатить парковку:" and contains a list of payment methods under the heading "Способы оплаты". The first item is highlighted in blue: "Банковской картой с помощью паркомата или парковочной картой её можно приобрести в МКУ Белгородское Парковочное Пространство номиналом 1000 руб. по адресу ул. Вокзальная, 10". Other items include "СМС-сообщением, отправив запрос с текстом р31*номер парковочной зоны*номер автомобиля*количество часов на номер", "Из личного кабинета веб-портала, зарегистрировавшись на сайте или пополнив счёт с помощью Банк.картой, телефона, СберБанкОнлайн, Яндекс.Деньги", and "С помощью мобильного приложения или пополнив счёт с помощью Банк.картой, телефона, СберБанкОнлайн, Яндекс.Деньги". At the bottom of the form are two buttons: "Получить ответ" and "Сбросить".

Рисунок 24 – Данные по Белгороду

The screenshot shows the same web application window as in Figure 24, but with "Екатеринбург" selected in the "Веберите город:" dropdown menu. The "Оплата парковки" tab is active. The list of payment methods under "Способы оплаты" is different. The first item is highlighted in blue: "Банковской картой с помощью паркомата или Е-картой (с тарифом "Электронный кашелек")". Other items include "СМС-сообщением, отправив запрос с текстом р66*номер парковочной зоны*номер автомобиля*количество часов на номер", "Из личного кабинета веб-портала, зарегистрировавшись на сайте или пополнив счёт с помощью мобильного телефона", and "С помощью мобильного приложения или пополнив счёт с помощью мобильного телефона". The "Получить ответ" and "Сбросить" buttons are also present at the bottom.

Рисунок 25 – Данные по Екатеринбургу

Результат, полученный на остальных закладках, для города Белгород представлен в приложении В на рисунках В4-В6.

Для того чтобы начать поиск заново необходимо нажать на кнопку «Сбросить», осуществить выбор города и нажать на кнопку «Получить ответ».

Также для всех пользователей доступен пункт меню «Пользователь» и его подпункты «Сменить пользователя» и «Выход из системы», при выполнении которых происходит возврат к окну авторизации и закрытие приложения соответственно.

Далее будет осуществлен вход в роли администратора с функциями, доступными руководителю отделения контактного центра.

Личный кабинет администратора по внешнему виду и функциям главной формы не отличается от кабинета любого менеджера. Однако, в меню появляется пункт «Сотрудники» и в пункте «Перейти в проект «Городские парковки»» добавляется подпункт «Редактировать информацию», представленные на рисунке 26.

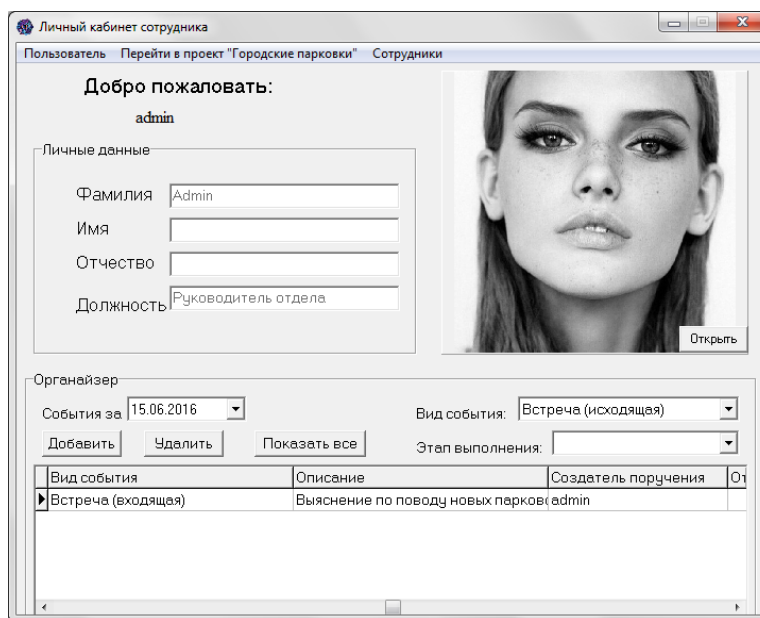


Рисунок 26 – Меню администратора

При нажатии на пункт меню «Сотрудники» происходит открытие одноименной формы, представленной на рисунке 27.

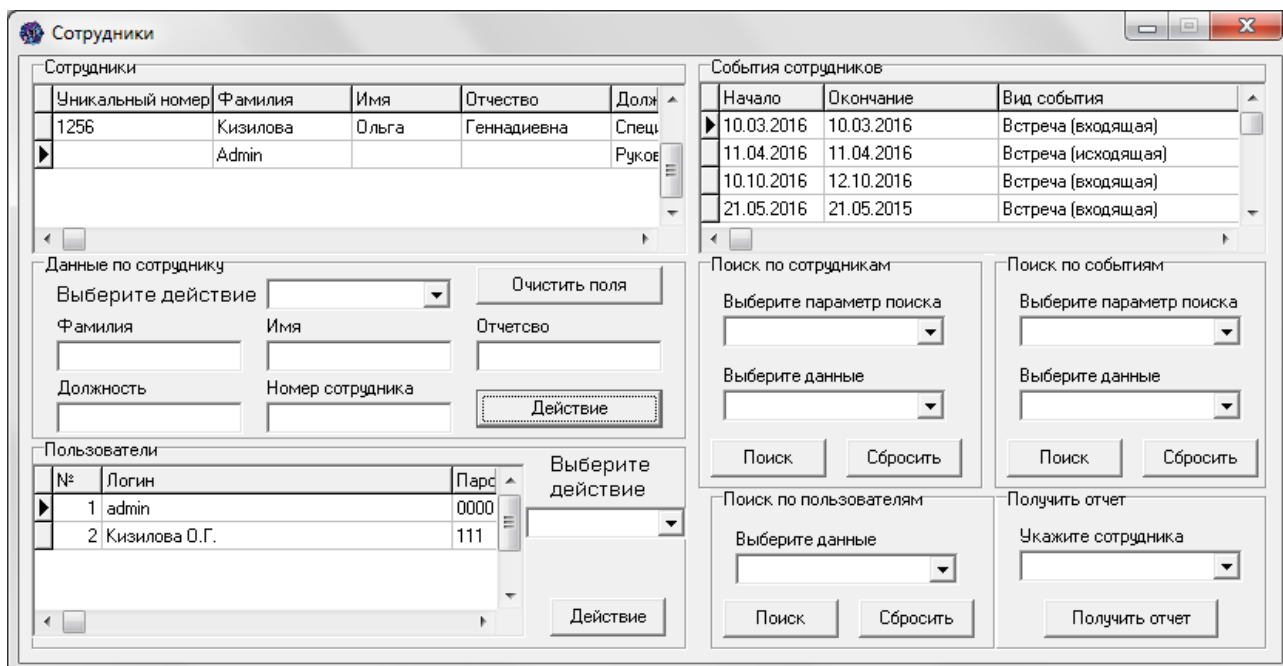


Рисунок 27 – Окно с данными работников

Данная форма состоит из 3-х частей: непосредственно личные данные сотрудников, пользователи и события, совершенные ими.

Для осуществления действия над информацией работников необходимо выбрать соответствующее действие из выпадающего списка действий. При выполнении данной операции появляется окно, оповещающее о пунктах, которые доступны для заполнения и блокируются пункты, которые не потребуются. При недостатке введенной информации, появляется окно, сообщающее об ошибке и извещающее, как ее исправить.

Если добавляется новый сотрудник, тогда после нажатия на кнопку «Добавить» открывается окно для добавления нового пользователя системы, куда нужно ввести логин и пароль пользователя.

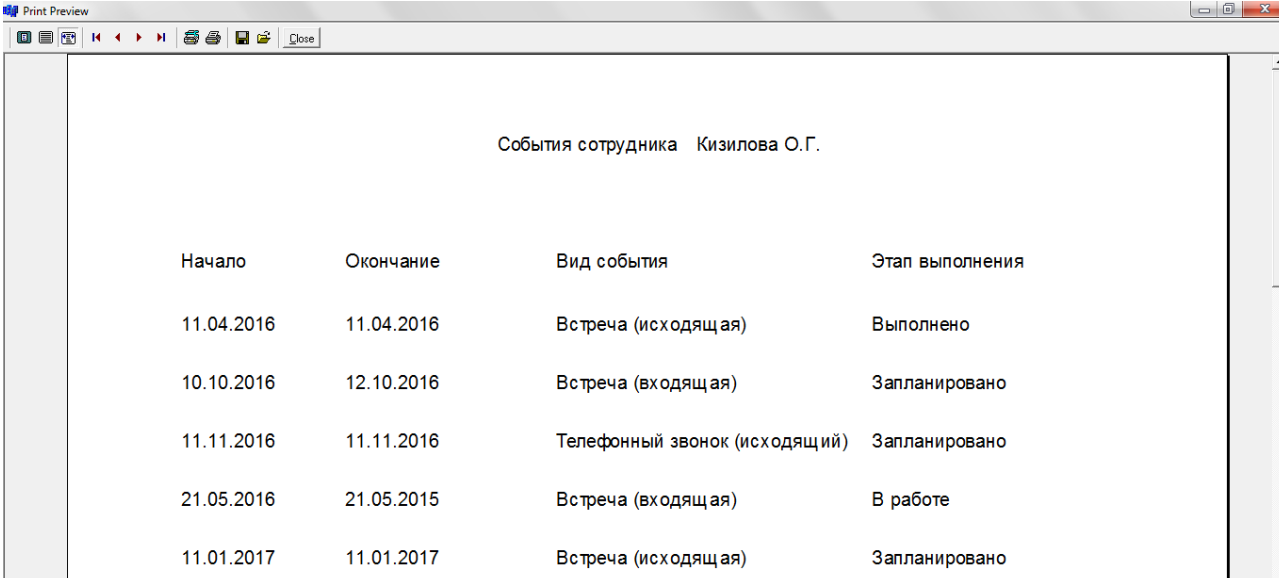
Для изменения и удаления пользователей необходимо ввести номер пользователя и нажать соответствующую кнопку.

Все изменения, происходящие с данными сотрудников, пользователей отображаются в таблицах, находящихся на форме.

Также имеется возможность поиска по сотрудникам, событиям и пользователям, выбирая для этого необходимые поля в соответствующем выпадающем списке с подписью «Выберите данные».

Результат работы с данными пользователей и сотрудников представлен в приложении В на рисунках В7-В9.

Для составления отчета по событиям, произведенным любым сотрудником необходимо выбрать сотрудника в группе «Получить отчет» и нажать на одноименную кнопку. Результат полученного отчета по работнику «Кизилова О.Г.» представлен на рисунке 28.



События сотрудника Кизилова О.Г.

Начало	Окончание	Вид события	Этап выполнения
11.04.2016	11.04.2016	Встреча (исходящая)	Выполнено
10.10.2016	12.10.2016	Встреча (входящая)	Запланировано
11.11.2016	11.11.2016	Телефонный звонок (исходящий)	Запланировано
21.05.2016	21.05.2015	Встреча (входящая)	В работе
11.01.2017	11.01.2017	Встреча (исходящая)	Запланировано

Рисунок 28 – Сформированный отчет

При нажатии в пункте меню «Перейти в проект “Городские парковки”» на подпункт «Редактировать информацию» появляется окно, представленное на рисунке 29.

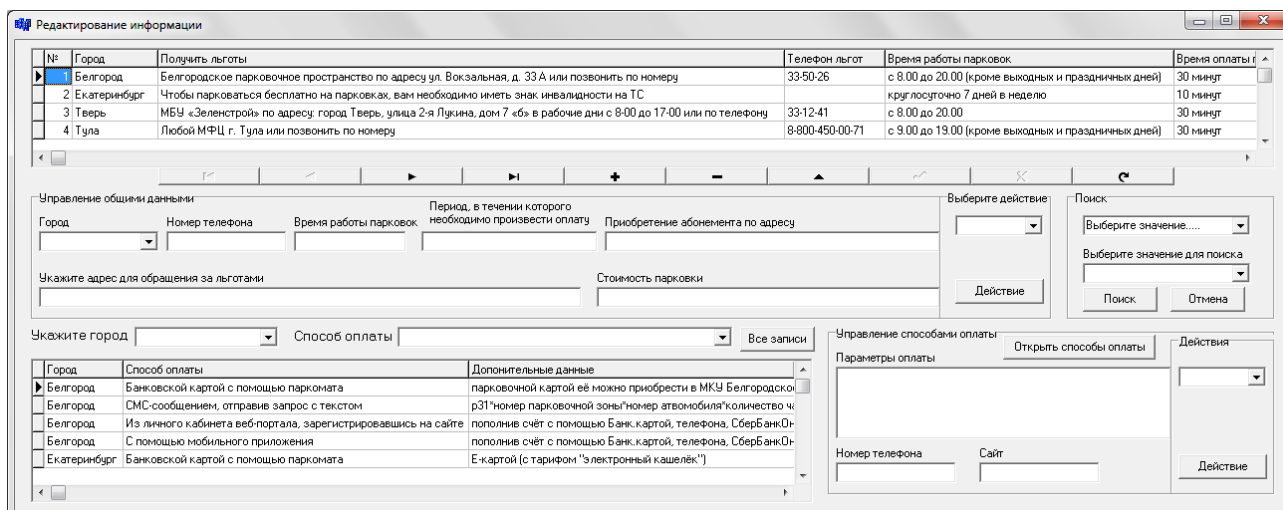


Рисунок 29 – Окно работы с данными парковочных зон

На данной форме предоставлена возможность редактирования данных по способу оплаты в городах, данным по парковкам городов. Для изменения любых данных необходимо выбрать действие в выпадающем списке. После подтверждения в окне оповещения о доступных полях, вводятся необходимые данные и нажимается соответствующая кнопка в соответствующей зоне.

Пример добавления данных по парковкам представлен на рисунке 30.

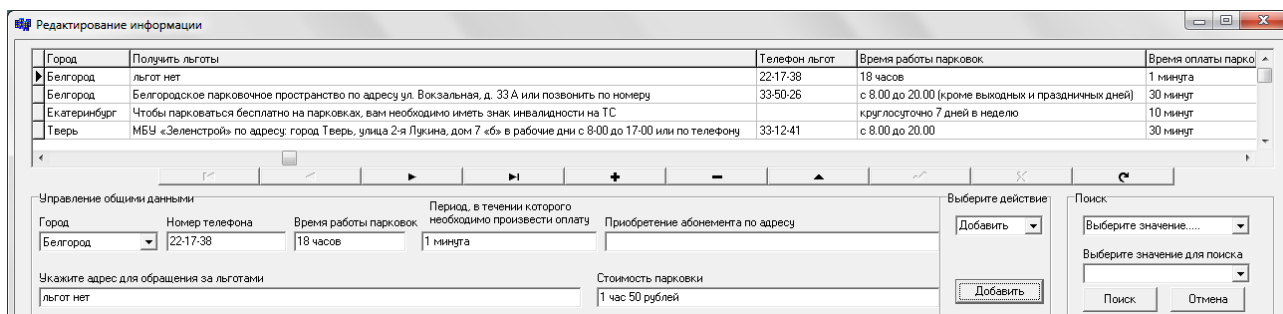


Рисунок 30 – Добавление данных по парковкам

Теперь произведем операцию удаления только что добавленной записи. При выборе нового действия происходит очищение всех доступных для ввода полей и блокировка недоступных. Результат удаления записи представлен на рисунке 31.

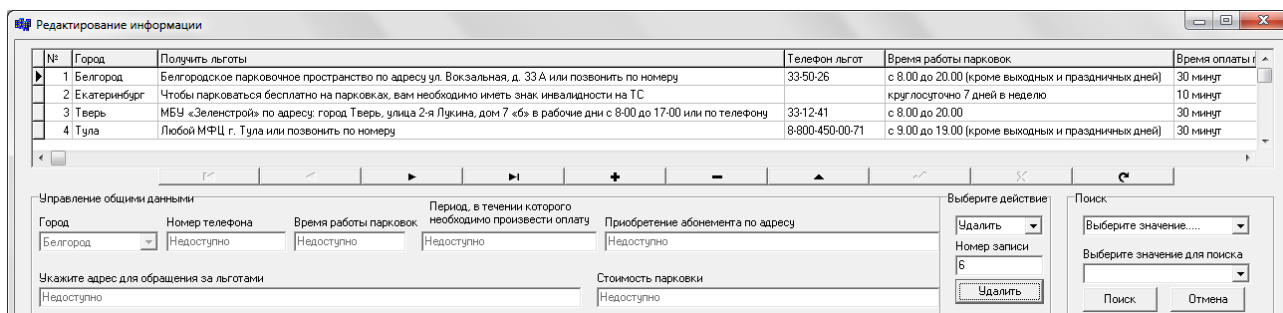


Рисунок 31 – Удаление данных по парковкам

Таким же образом производится и работа с данными по способам оплаты парковок в городах.

Данные способов оплаты в городе можно фильтровать по городу или способу оплаты. Поиск данных, осуществленный по городу «Белгород», представлен на рисунке 32.

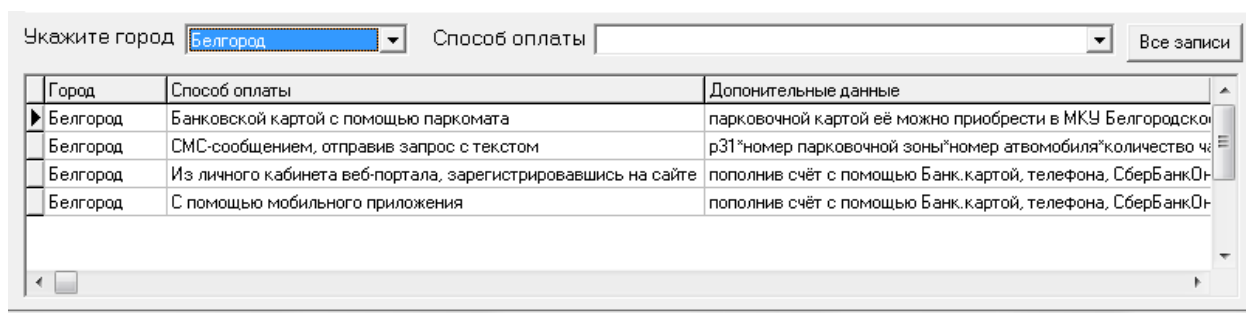


Рисунок 32 – Поиск способов оплаты парковок г.Белгород

Для отмены фильтрации необходимо нажать на кнопку «Все записи».

Таким же образом можно осуществить поиск парковок в городах, удовлетворяющим выбранным параметрам из выпадающего списка.

При нажатии на кнопку «Открыть способы оплаты» выводится окно, представленное на рисунке 33.

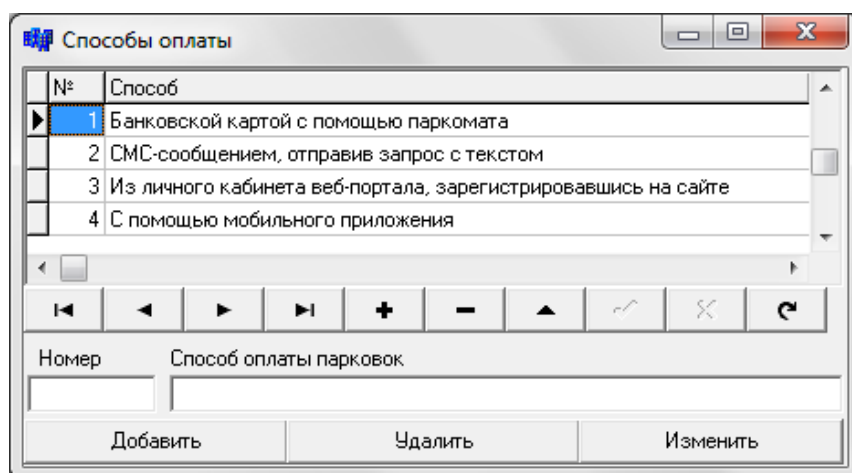


Рисунок 33 – Окно редактирования данных по способам оплаты

Данная форма позволяет добавлять, изменять и удалять записи таблицы способов оплаты. Пример работы с данными в окне «Способы оплаты», представлен в приложении В на рисунке В10.

На формах системы также имеется возможность перемещения по данным таблицам при помощи соответствующих панелей действий.

3.5 Расчет экономической эффективности

При проектировании системы взаимодействия с клиентами были определены этапы разработки и их продолжительность:

- 1) анализ технико-экономического состояния предприятия и составление технического задания ТЗ – 4 часа;
- 2) изучение технической, программной и информационной составляющей фирмы – 2 часа;
- 3) составление списка требований call-центра ООО «ФИТ» к автоматизированной системе взаимодействия с клиентами – 1 час;
- 4) определение класса информационной системы и, исходя из требований фирмы, выбор средств программирования – 1 час;
- 5) поиск и выбор информационных источников – 5 часов;
- 6) разработка алгоритма и структуры программы – 10 часов;

- 7) программирование – 100 часов;
- 8) тестирование программного продукта – 3 часа;
- 9) исправление выявленных недоработок или несоответствия полученным ранее требованиям – 10 часов;
- 10) разработка инструкции и описаний системы – 10 часов.

Для выполнения вышеописанных этапов были определены области и соответствующие специалисты ООО «ФИТ», которые могли бы осуществлять консультирование при создании программного продукта. В соответствии с поставленной задачей, были выделены специалист по разработке программного обеспечения и системный администратор.

При оценке прибыльности (безубыточности) проекта, были определены и рассчитаны затраты на разработку системы.

Рассматривая область расходов, необходимо подразделять их на капитальные (единовременные) – разработка самой системы и эксплуатационные (текущие).

Расчет был выполнен по отдельным статьям:

- прямые материальные затраты;
- фонд оплаты труда;
- отчисления на социальные нужды;
- амортизационные отчисления;
- накладные расходы;
- прочие расходы.

Расчет материальных затрат приведен в таблице 4.

Таблица 4 – Расчет прямых материальных затрат

Материалы	Ед. измер.	Кол – во	Цена за ед. (руб.)	Стоимость (руб.)
Бумага	Лист	200	1	200
Печать документов	стр.	200	2	400
Заправка картриджа	шт.	1	400	400
Итого				1000
Расходы на энергию	кВт/ч	13	3,53	45,89
ИТОГО				1045,89

Прямые материальные затраты составили 1045,89 рубля.

Для осуществления проекта никаких единовременных затрат не потребуется.

При расчете фонда оплаты труда необходимо вычислить заработную плату разработчиков ИС, составить баланс рабочего времени.

Данные по общему количеству трудового времени организации ООО «ФИТ» отображены в таблице 5.

Таблица 5 – Баланс рабочего времени

П/п	Наименование показателей	ИТР
1	Число календарных дней в году	365
2	Число выходных и нерабочих дней в году	119
3	Число рабочих дней в году	247
4	Невыходы на работу:	0
5	Фактическое число рабочих дней в году	247
6	Продолжительность рабочего дня	8
7	Годовой фонд рабочего времени (час.)	1976

Часовая ставка заработной платы ($Чс$) была определена по следующей формуле 3.1:

$$Чс = (З * М * К) / \Phi \quad (3.1)$$

$З$ – среднемесячная зарплата;

$М$ – число месяцев в году, исключая отпуск;

$К$ – коэффициент, учитывающий премии из фонда зарплаты;

Φ – фактический годовой фонд рабочего времени, час.

В среднем в месяц разработчик программного обеспечения и системный администратор получают 19000 руб.

При $М = 11$, $К=1,4$, а $\Phi = 2024$ часа, получилось $Чс = 148,08$ руб./час.

Размер основной заработной платы был определен исходя из времени, затрачиваемого на выполнение работ и стоимости часа работы исполнителя.

Основная заработная плата определяется по графику основных этапов работ. Дополнительная заработная плата будет составлять 0 рублей.

Расчет фонда оплаты труда приведен в таблице 6.

Таблица 6 – Расчет фонда оплаты труда

Этапы разработки	Время (час)	Часовая ставка (руб.)	Сумма (руб.)
Анализ технико-экономического состояния предприятия и составление технического задания ТЗ	4	148,08	592,32
Изучение технической, программной и информационной составляющей фирмы	2	148,08	296,16
Составление списка требований call-центра ООО «ФИТ» к автоматизированной системе взаимодействия с клиентами	1	148,08	148,08
Определение класса информационной системы и, исходя из требований фирмы, выбор средств программирования	1	148,08	148,08
Поиск и выбор информационных источников	5	148,08	740,4
Разработка алгоритма и структуры программы	10	148,08	1480,8
Программирование	100	148,08	14808
Тестирование программного продукта	3	148,08	444,24
Исправление выявленных недоработок или несоответствия полученным ранее требованиям	10	148,08	1480,8
Разработка инструкции и описаний системы	10	148,08	1480,8
Основная заработная плата(итого)			21619,68
Дополнительная заработная плата			0
ИТОГО			21619,68

Размер отчислений на социальные нужды определяется исходя из размера фонда оплаты труда. Расчет приведен в таблице 7.

Таблица 7 – Отчисления на социальные нужды

Отчисления	Доля от фонда оплаты труда (%)	Сумма (руб.)
В пенсионный фонд	22	4756,33
В фонд занятости	0,2	43,3
Медицинское страхование	5,1	1102,6
Социальное страхование	2,9	626,97
ИТОГО	30,2	6529,2

Амортизационные отчисления на оргтехнику в данном случае не производились. В таблице 8 приведена полная смета оплаты труда.

Таблица 8 – Расчет затрат на разработку ИС

Статья расхода	Сумма (руб.)
Фонд оплаты труда	21619,68
Отчисления на социальные нужды	6529,2
Материальные затраты	1045,89
Амортизационные отчисления	0
Прочие расходы	0
Накладные расходы	0
ИТОГО	29194,77

Расчет ежемесячных затрат на эксплуатацию системы взаимодействия с клиентами приведен в таблице 9.

Таблица 9 – Расчет затрат на эксплуатацию системы

Статья расхода	Сумма (руб.)
Зарплата системного администратора	18000
Отчисления на социальные нужды	6810,2
Затраты на электроэнергию	321,23
ИТОГО	25041,2

При оценке инвестиционной работы автоматизированной системы взаимодействия с клиентами были рассчитаны планируемые поступления денежных средств от реализации услуг по месяцам. Расчет данных показателей приведен в таблице 10.

Таблица 10 – Расчет планируемых поступлений

Месяц	Увеличение объема продаж, %	Сумма, руб.	Затраты, руб.	CF, руб.	$сF_t^{(\Sigma)}$ руб.
1	2	3	4	5	6
1	0	0	25041,2	- 25041,2	- 25041,2
2	1	15500	25041,2	- 9541,2	- 34582,4
3	1	15500	25041,2	- 9541,2	- 44123,6
4	1	31000	25041,2	- 9541,2	- 53664,8
5	2	31000	25041,2	5958,8	-47706
6	2	31000	25041,2	5958,8	-41747,2
7	2	46500	25041,2	5958,8	-35788,4
8	3	46500	25041,2	21458,8	-14329,6
9	3	46500	25041,2	21458,8	7129,2

Продолжение таблицы 10

1	2	3	4	5	6
10	3	46500	25041,2	21458,8	28588
11	3	46500	25041,2	21458,8	50046,8
12	3	46500	25041,2	21458,8	71505,6

Проведем оценку инвестиционного проекта автоматизированной системы взаимодействия с клиентами по формулам 3.2-3.4:

$$NPV = \sum_{t=1}^n \frac{CF_t}{k + q} - I_0, \text{ где} \quad (3.2)$$

NPV – чистая текущая стоимость инвестиций;

CF – поступление денежных средств в конце t-ого периода;

Q – банковская ставка;

I – стоимость реализации инвестиционного проекта (инвестиции).

$NPV = 71505,6 / (1 + 0.105) - 41310,46 = 30300,14$ рублей.

$$PP = \frac{I_0}{CF_t^{(\Sigma)}}, \quad (3.4)$$

где PP – период окупаемости (лет);

I_0 – первоначальные инвестиции;

$CF_t^{(\Sigma)}$ – годовая сумма денежных поступлений от реализации инвестированного проекта.

$PP = 41310,46 / (71505,6 / (1 + 0.105)) = 0,63$ лет.

Таким образом, затраты на разработку и внедрение системы окупятся за 10 месяцев со дня введения системы в действие. Внедрение системы позволило увеличить качество обслуживания и количество принятых звонков, уменьшая время поиска данных.

Итак, в третьем разделе представлены модели «Как Есть» и инфологическая, продемонстрирована работа и функции системы по средствам предоставленных схем. Были наглядно представлены результаты работы получившегося программного продукта. После чего, была рассчитана экономическая эффективность системы для ООО «ФИТ».

Заключение

Результатом выпускной квалификационной работы стала автоматизированная система взаимодействия с клиентами.

Для достижения поставленных целей и задач был выполнен ряд действий.

Было определено технико-экономического состояния предприятия в выбранном отделе менеджмента и маркетинга был проанализирован способ работы сотрудников, а также выявлены изъяны, устранить которые возможно будет путем автоматизации определенных процессов фирмы. Определены цель и назначение автоматизированного варианта и общая характеристика организации решения задачи.

Выбраны средства, методы и технологии для создания системы, определены основные требования к обеспечению. Были определены и обоснованы проектные решения для программного средства, предпочтительного в связи с выявленной проблемой.

Была описана созданная система, ее структура и работа. Были наглядно представлены результаты работы получившегося программного продукта. После чего, была рассчитана экономическая эффективность системы для ООО «ФИТ».

Разработанная автоматизированная система удовлетворяет всем поставленным требованиям компании. Функциональные возможности и интерфейс просты и интуитивно понятны, что позволило сократить время обучения персонала работе с новым приложением.

Рассматривая систему, как новый продукт, можно заметить, что она автоматизирует лишь часть функций. В дальнейшем наблюдается возможность минимизации затрат времени путем интегрирования полученной системы с программой ЗСХ подробно, что позволит менеджерам по взаимодействию с клиентами не вводить данные по произведенным действиям вручную, а автоматически получать информацию о пропущенных или принятых звонках.

В выпускной квалификационной работе были решены все поставленные задачи, а именно:

- изучена деятельность call-центра организации;
- проанализированы слабые стороны работы отдела и определены наиболее подходящие пути решения выявленных недостатков;
- спроектирована информационная модель call-центра;
- разработана автоматизированная система взаимодействия с клиентами;
- выполнено тестирование системы;
- рассчитана экономическая эффективность системы.

Цель выпускной квалификационной работы была достигнута.

Список использованных источников

1 ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления [Текст]. – Взамен ГОСТ 7.32-91; Введ. 01.07.2002. – М: Стандартинформ, 2008. – 20 с. – (Система стандартов по информации, библиотечному и издательскому делу).

2 ГОСТ 2.105-95. ОБЩИЕ ТРЕБОВАНИЯ К ТЕКСТОВЫМ ДОКУМЕНТАМ [Текст]. – Взамен ГОСТ 2.105-79. ГОСТ 2.906-71; Введ. 01.06.1996. – Минск: Межгосударственный совет по стандартизации, метрологии и сертификации, 2007. – 30 с. – (Единая система конструкторской документации).

3 Положение о выпускных квалификационных работах дипломированного специалиста, бакалавра, по программам получения дополнительных квалификаций от 15 октября 2007 г. (с изменениями и дополнениями от 22.06.2009 г.) [Текст] – Белгород: Изд-во БелГУ, 2009. – 20 с.

4 ГОСТ 7.1-2003. БИБЛИОГРАФИЧЕСКАЯ ЗАПИСЬ. БИБЛИОГРАФИЧЕСКОЕ ОПИСАНИЕ. Общие требования и правила составления [Текст]. – Взамен ГОСТ 7.1-84, ГОСТ 7.16-79, ГОСТ 7.18-79, ГОСТ 7.34-81, ГОСТ 7.40-82; Введ. 01.07.2004. – М: Стандартинформ, 2006. – 52 с. – (Система стандартов по информации, библиотечному и издательскому делу).

5 ГОСТ 7.82-2001. БИБЛИОГРАФИЧЕСКАЯ ЗАПИСЬ. БИБЛИОГРАФИЧЕСКОЕ ОПИСАНИЕ. ЭЛЕКТРОННЫХ РЕСУРСОВ. Общие требования и правила составления [Текст]. – Введ. 01.06.2002. – Минск: Межгосударственный совет по стандартизации, метрологии и сертификации, 2007. – 27 с. – (Система стандартов по информации, библиотечному и издательскому делу).

6 Фот, А.П. Оформление библиографических ссылок на использованные источники в научных работах [Электронный документ]: методические рекомендации соискателям ученых степеней. – Оренбург: ГОУ

«ОГУ», 2005. – 24 с. (<http://www.osu.ru/docs/official/nauka/bibllinks.doc>).
Проверено 11.04.2010.

7 ГОСТ 7.0.5-2008. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая ссылка. Общие требования и правила составления [Текст]. Введ. 01.01.2009. – М: Стандартинформ, 2009. – 23 с. – (Система стандартов по информации, библиотечному и издательскому делу).

8 ГОСТ 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения [Текст]. – Взамен ГОСТ 24.003-84, ГОСТ 22487-77; Введ. 01.01.1992. – М.: ГосСтандарт СССР, 1980. – 26 с.

9 Стандарт ISO/IEC 2382 Информационные технологии – Словарь – Первый Выпуск [Текст]. – Взамен ISO/IEC 2382-1:1993 – Женева, Швейцария, 2015. – 6 с.

10 Объектно-ориентированный анализ и проектирование с примерами приложений (UML 2). Третье издание / Гради Буч, Роберт А. Максимчук, Майкл У. Энгл и др. – М.: Издательство: Вильямс, 2010. – 720 с.

11 Кураков, Л. П. Экономика и право: словарь-справочник / Л. П. Кураков, В. Л. Кураков, А. Л. Кураков – М.: Вуз и школа, 2004. – 868 с.

12 Михелёв, В.М. Базы данных и СУБД: учебное пособие / В.М. Михелёв – Белгород: Изд-во БелГУ, 2007 г. – 200 с

13 Архангельский, А. Я. Программирование в С++ Builder 7-е изд / А. Я. Архангельский – М.: ООО «Бином-Пресс», 2010г. – 896 с.

14 Хомоненко, А. Д. Работа с базами данных в С++ Builder / А. Д. Хомоненко, С. Е. Ададунов – СПб.: БХВ-Петербург, 2008. – 496 с.

15 Васвани, В. MySQL: использование и администрирование / В. Васвани – М.: «Питер», 2011. – 368 с.

16 Кузнецов, С. Д. Основы баз данных / С. Д. Кузнецов – 2-е изд – М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2007. – 484 с.

- 17 Бочаров, Е.П. Интегрированные корпоративные информационные системы / Е.П. Бочаров, А.И. Колдина – М.: Финансы и статистика, 2010. – 350 с.
- 18 Глушаков, С.В. Язык программирования C++, учебный курс, ANSI / С.В. Глушаков, А.В. Коваль, С.В. Смирнов – Харьков: Издательство: Фолио, 2010. – 340 с.
- 19 Емельянова, Н.З Основы построения автоматизированных информационных систем. Учебное пособие / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов – М.: Academia, 2010. – 220 с.
- 20 Коберн, А. Быстрая разработка программного обеспечения / А. Коберн. – М.: Лори, 2013. – 336 с.
- 21 Павловская, Т.А. С/С++. Структурное и объектно-ориентированное программирование / Т.А. Павловская – СПб.: Питер, 2010. – 352 с.
- 22 Кузин, А. В. Работа с изображениями в БД [Текст]: методические указания к лабораторным работам «СУБД MySQL Borland» / Кузин, А. В.; Сургутский Государственный Университет – Сургут: ИНФРА-М, 2005. – 75 с.
- 23 Маклаков, С. В. BPWin и ERWin. CASE-средства разработки информационных систем / С. В. Маклаков – М.: Диалог-МИФИ, 2001. – 455 с.
- 24 Кузин, А. В. Создание отчетов [Текст]: методические указания к лабораторным работам «СУБД MySQL Borland» / Кузин, А. В.; Сургутский Государственный Университет – Сургут: ИНФРА-М, 2005. – 75 с.
- 25 Информационные системы и технологии [Электронный ресурс] – Электрон. текстовые дан. – 2010. – Режим доступа: http://edulib.pgta.ru/els/_/disk0/38.03.02%20-%20Менеджмент/Информационные%20технологии/Технологическое%20обеспечение%20ИТ.pdf, свободный.
- 26 Технологический процесс обработки данных [Электронный ресурс] – Электрон. текстовые дан. – 2015. – Режим доступа:

http://studopedia.ru/5_152014_tehnologicheskij-protsess-obrabotki-dannih.html, свободный.

27 Рейтинги CRM-систем и их поставщиков [Электронный ресурс] – Электрон. текстовые дан. – 2015. – Режим доступа: http://www.tadviser.ru/index.php/Статья:Рейтинги_CRM-систем_и_их_поставщиков, свободный.

28 Унифицированные коммуникации на основе программной IP АТС [Электронный ресурс] – Электрон. текстовые дан. – 2016. – Режим доступа: <http://www.3cx.ru>, свободный.

29 Google Диск [Электронный ресурс] – Электрон. текстовые дан. – 2015. – Режим доступа: https://ru.wikipedia.org/wiki/Google_Диск, свободный.

30 Firebird [Электронный ресурс] – Электрон. текстовые дан. – 2015. – Режим доступа: <https://ru.wikipedia.org/wiki/Firebird>, свободный.

31 ИБExpert [Электронный ресурс] – Электрон. текстовые дан. – 2015. – Режим доступа: <https://ru.wikipedia.org/wiki/ИБExpert>, свободный.

32 Python [Электронный ресурс] – Электрон. текстовые дан. – 2015. – Режим доступа: https://ru.wikipedia.org/wiki/Python#cite_note-4, свободный.

33 Система управления взаимоотношениями с клиентами [Электронный ресурс] – Электрон. текстовые дан. – 2015. – Режим доступа: https://ru.wikipedia.org/wiki/Система_управления_взаимоотношениями_с_клиентами, свободный.

Приложения

Приложение А

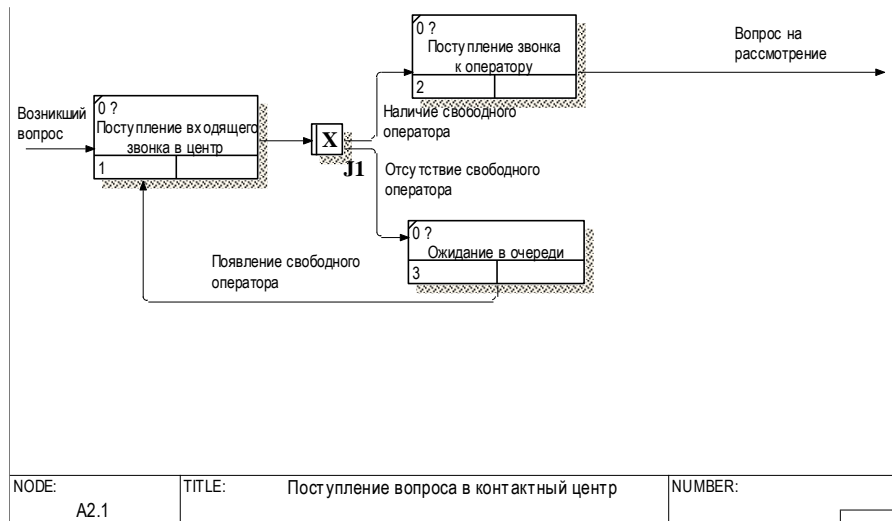


Рисунок А1 – Декомпозиция «Поступление вопроса в контактный центр»

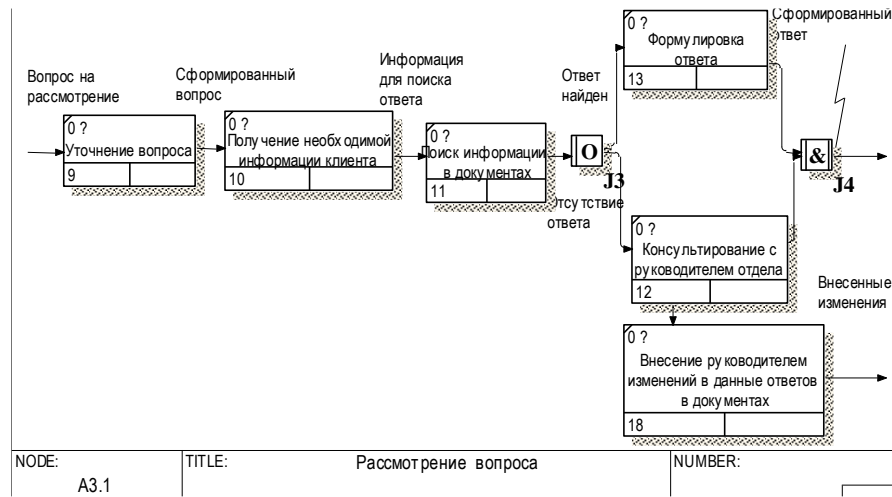


Рисунок А2 – Декомпозиция «Рассмотрение вопроса»

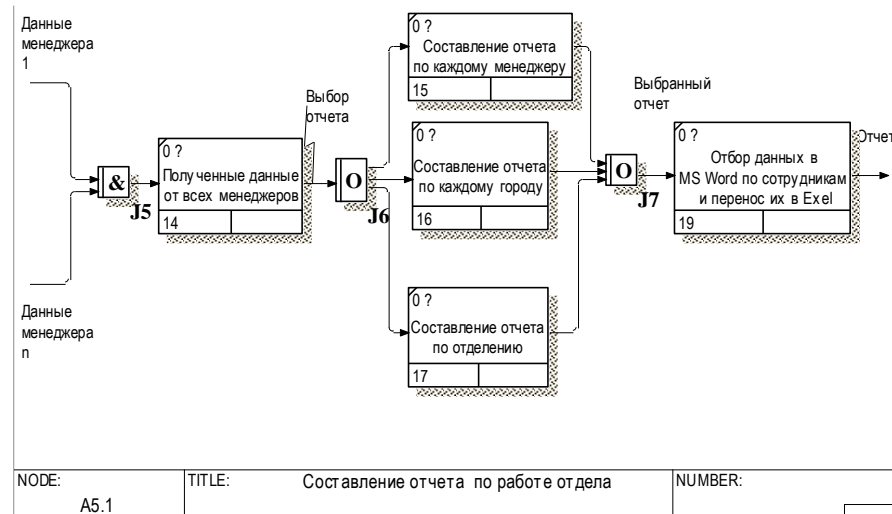


Рисунок А3 – Декомпозиция «Составление отчета по работе отдела»

Диаграммы модели «Как будет» представлены на рисунках 33-35.

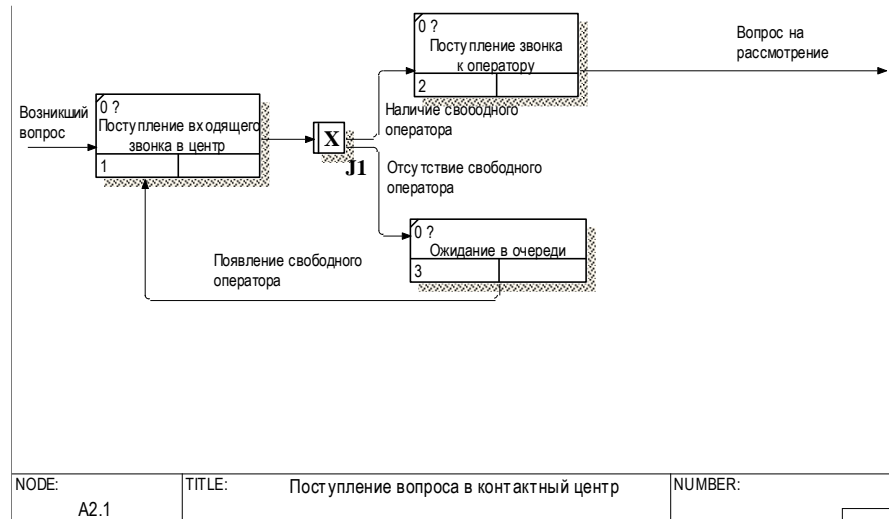


Рисунок А4 – Декомпозиция «Поступление вопроса в контактный центр»

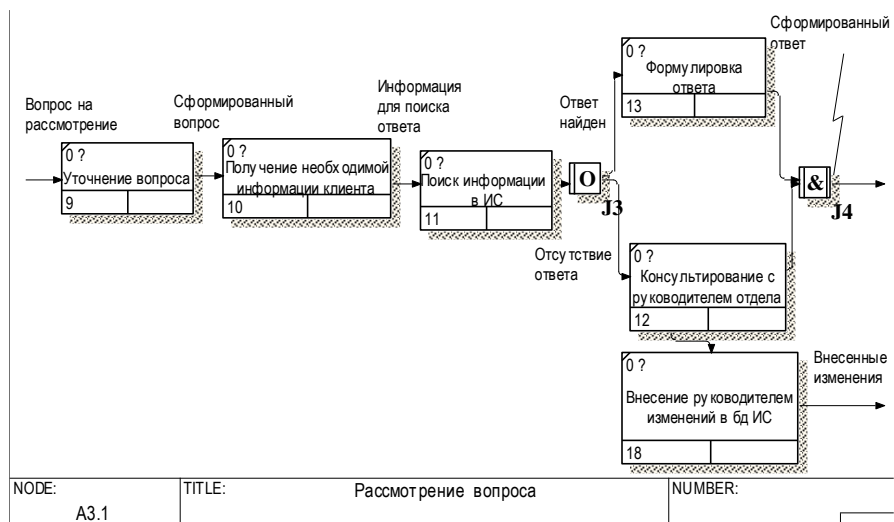


Рисунок А5 – Декомпозиция «Рассмотрение вопроса»

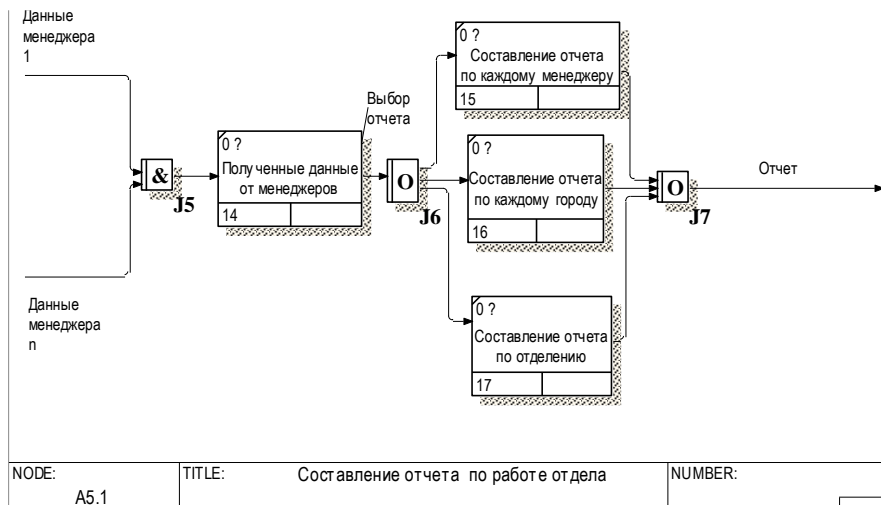


Рисунок А6 – Декомпозиция «Составление отчета по работе отдела»

Приложение Б

Код модуля авторизации:

```
#include <vcl.h>
#pragma hdrstop

#include "Glavnay.h"
#include "Unit1.h"
#include "Forma.h"
#include "Splash_form.h"
#include "Load_thread.h"
#include "Data_unit.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    StatInfo = new TStringList;
}
void __fastcall TForm1::AppLoaded(TObject *Sender)
{
    IsApploading = false;
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    DataModule1->IBTable1->Filtered=false;
    DataModule1->IBTable1->FilterOptions << foCaseInsensitive;
    DataModule1->IBTable1->Filter="(LOGIN='"+DBLookupComboBox1->Text+"'";
    DataModule1->IBTable1->Filtered=true;
    AnsiString str;
    str=Trim(DataModule1->IBTable1->FieldByName("PAROL")->AsString);
    if(str==Edit1->Text)
    {Form2->Show();
    Form1->Visible=False;
    Form2->Label1->Caption=DBLookupComboBox1->Text;
    DataModule1->IBSotr->Filtered=false;
    DataModule1->IBSotr->FilterOptions << foCaseInsensitive;
    DataModule1->IBSotr->Filter="(ID='"+DataModule1->IBTable1->FieldByName("Sotr")-
    >AsString+"'";
    DataModule1->IBSotr->Filtered=true;
    Form2->Edit1->Text=DataModule1->IBSotr->FieldByName("Fam")->AsString;
    Form2->Edit2->Text=DataModule1->IBSotr->FieldByName("Im")->AsString;
    Form2->Edit3->Text=DataModule1->IBSotr->FieldByName("Otch")->AsString;
    Form2->Edit4->Text=DataModule1->IBSotr->FieldByName("Dolzh")->AsString;
```

```

TStream *MyStream;
TJPEGImage *jpg = new TJPEGImage();
if (!DataModule1->IBSotr->FieldByName("FOTO")->IsNull)
{
MyStream = DataModule1->IBSotr->CreateBlobStream(DataModule1->IBSotr-
>FieldByName("FOTO"), bmRead);
jpg->LoadFromStream(MyStream);
Form2->Image1->Picture->Assign(jpg);
MyStream->Free();
}
if(Trim(DataModule1->IBTable1->FieldByName("Rol")->AsString)=="Администратор")
{
DataModule1->MainMenu1->Items->operator [](1)->operator [](1)->Visible=true;
DataModule1->MainMenu1->Items->operator [](2)->Visible=true;}
else {DataModule1->MainMenu1->Items->Items[1]->Items[1]->Visible=false;
DataModule1->MainMenu1->Items->operator [](2)->Visible=false;}
Form2->Menu=DataModule1->MainMenu1;
DataModule1->IBSotr->Filtered=false;
DataModule1->IBTable1->Filtered=false;
}
else {ShowMessage("Введен неправильный логин или пароль!!!");
DataModule1->IBTable1->Filtered=false;}
}
//-----
void __fastcall TForm1::FormDestroy(TObject *Sender)
{
delete StatInfo;
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
IsApploading = true;
TLoadApp *thr = new TLoadApp(true);
thr->InfoField = SplashForm->StaticText1;
thr->OnTerminate = AppLoaded;
thr->FreeOnTerminate = true;
thr->Resume();
while(IsApploading)
{
Application->HandleMessage();
}
}
//-----
void __fastcall TForm1::FormShow(TObject *Sender)
{
Form1->Show();}
//-----

```

Код Data_unit:

```

#pragma hdrstop
#include "Data_unit.h"

```

```

//-----
#pragma package(smart_init)
TStringList *StatInfo;

Код полосы загрузки:

#include <vcl.h>
#pragma hdrstop
#include "Load_thread.h"
#include "Data_unit.h"
#include <IniFiles.hpp>
#include <memory>
//-----
#pragma package(smart_init)
//-----
bool IsApploading;

__fastcall TLoadApp::TLoadApp(bool CreateSuspended)
: TThread(CreateSuspended)
{}
void __fastcall TLoadApp::UpdateField()
{
    InfoField->Caption = info_str;
}
//-----
void __fastcall TLoadApp::Execute()
{
    String fn = "first.ini";
    String s;
    info_str = String().sprintf(_T("Загрузка: %s"), fn);
    UpdateField();
    Sleep(1000);
    TIniFile *f_ini = new TIniFile(ExpandFileName(ExtractFilePath(ParamStr(0))) +
_T("ini\\first.ini"));
    std::auto_ptr<TStringList> section (new TStringList);
    f_ini->ReadSection(_T("data"), section.get());
    for(int i = 0; i < section->Count; i++)
    {
        info_str = String().sprintf(_T("Загрузка: %s"), (s = f_ini->ReadString(_T("data"), section-
>Strings[i], _T("oops"))));
        StatInfo->Add(s + _T(" processed"));
        UpdateField();
        Sleep(1000);
    }
    delete f_ini;
    section->Clear();
    fn = "second.ini";
    info_str = String().sprintf(_T("Загрузка: %s"), fn);
    UpdateField();
    TIniFile *s_ini = new TIniFile(ExpandFileName(ExtractFilePath(ParamStr(0))) +
_T("ini\\second.ini"));
    s_ini->ReadSection(_T("users"), section.get());

```

```

    for(int i = 0; i < section->Count; i++)
    {
        info_str = String().sprintf(_T("Загрузка: %s"), (s = s_ini->ReadString(_T("users"), section-
>Strings[i], _T("oops"))));
        StatInfo->Add(s + _T(" processed"));
        UpdateField();
        Sleep(1000);
    }
    delete s_ini;
}
//-----

```

Код личного кабинета:

```

#include <vcl.h>
#pragma hdrstop
#include "Forma.h"
#include "Glavnay.h"
#include "Unit1.h"
#include <jpeg.hpp>
#include "Parcovki.h"
#include "Redact.h"
#include "Sobitiy.h"
//-----
#pragma package(smart_init)
#pragma link "CCALENDR"
#pragma resource "*.dfm"
TForm2 *Form2;
//-----
__fastcall TForm2::TForm2(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm2::Button1Click(TObject *Sender)
{
    if (OpenPictureDialog1->Execute())
    {
        Image1->Picture->LoadFromFile(OpenPictureDialog1->FileName);
        DataModule1->IBSotr->Filtered=false;
        DataModule1->IBSotr->FilterOptions << foCaseInsensitive;
        DataModule1->IBSotr->Filter="(ID='"+DataModule1->IBTable1->FieldByName("Sotr")-
>AsString+"")";
        DataModule1->IBSotr->Filtered=true;
        DataModule1->IBStorFoto->ParamByName("i")->AsString=DataModule1->IBSotr-
>FieldByName("id")->AsString;
        DataModule1->IBStorFoto->ParamByName("f")->LoadFromFile(OpenPictureDialog1-
>FileName,ftBlob);
        DataModule1->IBStorFoto->Prepare();
        DataModule1->IBStorFoto->ExecProc();
    }
}

```



```

//-----
void __fastcall TForm2::DBGrid1CellClick(TColumn *Column)
{
Form5->Button2->Visible=true;
Form5->Button1->Visible=false;
Form5->Edit1->Text=DBGrid1->Fields[0]->AsString;
Form5->Edit2->Text=DBGrid1->Fields[1]->AsString;
if(DBGrid1->Fields[2]->AsString!="")
Form5->DBLookupComboBox1->KeyValue=DBGrid1->Fields[2]->AsString;
else Form5->DBLookupComboBox1->KeyValue=0;
Form5->Memo1->Text="";
Form5->Memo1->Text=DBGrid1->Fields[3]->AsString;
if(DBGrid1->Fields[4]->AsString!="")
Form5->DBLookupComboBox2->KeyValue=DBGrid1->Fields[7]->AsString;
else Form5->DBLookupComboBox2->KeyValue=0;
if(DBGrid1->Fields[5]->AsString!="")
Form5->DBLookupComboBox3->KeyValue=DBGrid1->Fields[8]->AsString;
else Form5->DBLookupComboBox3->KeyValue=0;
if(DBGrid1->Fields[6]->AsString!="")
Form5->DBLookupComboBox4->KeyValue=DBGrid1->Fields[6]->AsString;
else Form5->DBLookupComboBox4->KeyValue=0;
DataModule1->IBStorOrgUp->ParamByName("Kod")->AsString= Form2->DBGrid1->Fields[9]-
>AsString;
Form5->Button1->Visible=false;
Form5->Show();
}
//-----
void __fastcall TForm2::Button2Click(TObject *Sender)
{
Form5->Memo1->Text="";
Form5->Edit1->Text="";
Form5->Edit2->Text="";
Form5->DBLookupComboBox1->KeyValue=NULL();
Form5->DBLookupComboBox2->KeyValue=NULL();
Form5->DBLookupComboBox3->KeyValue=NULL();
Form5->DBLookupComboBox4->KeyValue=NULL();
Form5->Button2->Visible=false;
Form5->Button1->Visible=true;
DataModule1->IBSotr->Filtered=false;
Form5->Show();
}
//-----
void __fastcall TForm2::Button3Click(TObject *Sender)
{
DataModule1->IBStorOrUD->ParamByName("Kod")->AsString= Form2->DBGrid1->Fields[9]-
>AsString;
DataModule1->IBStorOrUD->Prepare();
DataModule1->IBStorOrUD->ExecProc();
ShowMessage("Запись удалена");
DataModule1->IBOrg->Active=false;
DataModule1->IBOrg->Active=true;
}

```

```

//-----
void __fastcall TForm2::DBLookupComboBox2Click(TObject *Sender)
{
DataModule1->IBOrg->Filtered=false;
DataModule1->IBOrg->FilterOptions << foCaseInsensitive;
DataModule1->IBOrg->Filter="(Vid like '"+DBLookupComboBox2->KeyValue+'")";
DataModule1->IBOrg->Filtered=true;
}
//-----
void __fastcall TForm2::DBLookupComboBox1Click(TObject *Sender)
{
DataModule1->IBOrg->Filtered=false;
DataModule1->IBOrg->FilterOptions << foCaseInsensitive;
DataModule1->IBOrg->Filter="(Pa like '"+DBLookupComboBox1->KeyValue+'")";
DataModule1->IBOrg->Filtered=true;
}
//-----
void __fastcall TForm2::Button4Click(TObject *Sender)
{
DataModule1->IBOrg->Filtered=false;
}
//-----

```

Код вопросно-ответной системы:

```

#include <vcl.h>
#pragma hdrstop
#include "Parcovki.h"
#include "Forma.h"
#include "Glavnay.h"
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm3 *Form3;
//-----
__fastcall TForm3::TForm3(TComponent* Owner)
: TForm(Owner)
{
StringGrid1->Cells[0][0]="Способы оплаты";
Label2->Caption="";
Label4->Caption="";
Label5->Caption="";
Label6->Caption="";
Label7->Caption="";
Label8->Caption="";
Label9->Caption="";
}
//-----
void __fastcall TForm3::Button1Click(TObject *Sender)
{
int i, b;

```

```

DataModule1->IBOsнова->Filtered=false;
DataModule1->IBOsнова->FilterOptions << foCaseInsensitive;
DataModule1->IBOsнова->Filter="(GOR like '"+DBLookupComboBox1->KeyValue+'"");
DataModule1->IBOsнова->Filtered=true;
b=5;
if(DBLookupComboBox1->Text=="Тула")
{b=6;}
DataModule1->IBPlat->First();
for (i=1; i<b; i++)
{
DataModule1->IBInfa->Filtered=false;
DataModule1->IBInfa->FilterOptions << foCaseInsensitive;
DataModule1->IBInfa->Filter="(GOR like '"+DBLookupComboBox1->KeyValue+"' and Spopl
like "+i+"");
DataModule1->IBInfa->Filtered=true;
if(i==2)
StringGrid1->Cells[0][i]=Trim(DataModule1->IBInfa->FieldByName("Sposob")->AsString)+"
"+DataModule1->IBInfa->FieldByName("Dopoln")->AsString+DataModule1->IBInfa-
->FieldByName("Nom")->AsString+DataModule1->IBInfa->FieldByName("Sait")->AsString;
else
StringGrid1->Cells[0][i]=Trim(DataModule1->IBInfa->FieldByName("Sposob")->AsString)+"
или "+DataModule1->IBInfa->FieldByName("Dopoln")->AsString+DataModule1->IBInfa-
->FieldByName("Nom")->AsString+DataModule1->IBInfa->FieldByName("Sait")->AsString;
DataModule1->IBPlat->Next();}
StringGrid1->RowCount=i;
if(DataModule1->IBOsнова->FieldByName("VREM")->AsString=="")
Label2->Caption="Данные отсутствуют!";
else
Label2->Caption="Время работы платных парковок в г."+Trim(DBLookupComboBox1-
->Text)+": "+DataModule1->IBOsнова->FieldByName("VREM")->AsString;
if(DataModule1->IBOsнова->FieldByName("WREM")->AsString=="")
Label5->Caption="Данные отсутствуют!";
else
Label5->Caption="Предоставляется "+Trim(DataModule1->IBOsнова-
->FieldByName("WREM")->AsString)+" на оплату парковки.";
Label4->Caption="Для того чтобы получить льготы на пользование платными парковками в
г."+Trim(DBLookupComboBox1->Text)+" необходимо обратиться в";
if(DataModule1->IBOsнова->FieldByName("Lgoti")->AsString=="")
Label6->Caption="Данные отсутствуют!";
else {if (DataModule1->IBOsнова->FieldByName("Nomertel")->AsString=="")
Label6->Caption=Trim(DataModule1->IBOsнова->FieldByName("Lgoti")->AsString);
else Label6->Caption=Trim(DataModule1->IBOsнова->FieldByName("Lgoti")->AsString)+
"+DataModule1->IBOsнова->FieldByName("Nomertel")->AsString;}
Label7->Caption="Приобрести резидентского абонеента на пользование платными
парковками можно, обратившись ";
Label8->Caption="Стоимость парковки: "+DataModule1->IBOsнова->FieldByName("stoim")-
->AsString;
if(DataModule1->IBOsнова->FieldByName("Abonem")->AsString=="")
Label9->Caption="Данные отсутствуют!";
else
Label9->Caption=Trim(DataModule1->IBOsнова->FieldByName("Abonem")->AsString);
}

```

```
//-----
void __fastcall TForm3::Button2Click(TObject *Sender)
{
DataModule1->IBTGoroda->Filtered=false;
for (int i=1; i < StringGrid1->RowCount; i++) {
StringGrid1->Rows[i]->Clear();
}
Label2->Caption="";
Label4->Caption="";
Label5->Caption="";
Label6->Caption="";
Label7->Caption="";
Label8->Caption="";
Label9->Caption="";
}
//-----
```

Код модуля редактирования данных парковок:

```
#include <vcl.h>
#pragma hdrstop
#include "Redact.h"
#include "Glavnay.h"
#include "Forma.h"
#include "Unit1.h"
#include "SPOPL.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm4 *Form4;
String U;
//-----
__fastcall TForm4::TForm4(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm4::DBLookupComboBox2Click(TObject *Sender)
{
DataModule1->IBInfa->Filtered=false;
DataModule1->IBInfa->FilterOptions << foCaseInsensitive;
DataModule1->IBInfa->Filter="(GOR like '"+DBLookupComboBox2->KeyValue+'")";
DataModule1->IBInfa->Filtered=true;
}
//-----
void __fastcall TForm4::FormCreate(TObject *Sender)
{
Memo1->Text="";
Edit6->Visible=false;
Edit2->Visible=false;
Label2->Visible=false;
Edit7->Visible=false;
```

```

Label13->Visible=false;
DBGrid2->Columns->operator [](0)->Visible=false;
}
//-----
void __fastcall TForm4::ComboBox2Change(TObject *Sender)
{
if (ComboBox2->ItemIndex==0)
{U="Gor";
DBLookupComboBox4->Visible=true;
Edit6->Visible=false; }
if (ComboBox2->ItemIndex==2)
{DBLookupComboBox4->Visible=false;
Edit6->Visible=true;
U="Stoim";}
if (ComboBox2->ItemIndex==1)
{DBLookupComboBox4->Visible=false;
Edit6->Visible=true;
U="Wrem";}
}
//-----
void __fastcall TForm4::DBLookupComboBox5Click(TObject *Sender)
{
DataModule1->IBInfa->Filtered=false;
DataModule1->IBInfa->FilterOptions << foCaseInsensitive;
DataModule1->IBInfa->Filter="(Spopl like '"+DBLookupComboBox5->KeyValue+"")";
DataModule1->IBInfa->Filtered=true;
}
//-----
void __fastcall TForm4::Button4Click(TObject *Sender)
{
DataModule1->IBInfa->Filtered=false;
}
//-----
void __fastcall TForm4::Button6Click(TObject *Sender)
{
DataModule1->IBOsnova->Filtered=false;
DataModule1->IBOsnova->FilterOptions << foCaseInsensitive;
if(DBLookupComboBox4->Visible==false)
DataModule1->IBOsnova->Filter="("+U+"='%"+Edit6->Text+"%')";
else
DataModule1->IBOsnova->Filter="("+U+" like '"+DBLookupComboBox4->KeyValue+"")";
DataModule1->IBOsnova->Filtered=true;
}
//-----
void __fastcall TForm4::Button9Click(TObject *Sender)
{
DataModule1->IBOsnova->Filtered=false;
Edit6->Text="";
}
//-----
void __fastcall TForm4::Button5Click(TObject *Sender)
{

```

```

if (ComboBox3->ItemIndex==0)
{
if(DBLookupComboBox1->Text=="")
ShowMessage("Введите, пожалуйста, город!");
else{
DataModule1->IBStorOSNNew->ParamByName("g")->AsString=DBLookupComboBox1-
>KeyValue;
DataModule1->IBStorOSNNew->ParamByName("l")->AsString=Edit9->Text;
DataModule1->IBStorOSNNew->ParamByName("n")->AsString=Edit5->Text;
DataModule1->IBStorOSNNew->ParamByName("v")->AsString=Edit3->Text;
DataModule1->IBStorOSNNew->ParamByName("w")->AsString=Edit8->Text;
DataModule1->IBStorOSNNew->ParamByName("s")->AsString=Edit1->Text;
DataModule1->IBStorOSNNew->ParamByName("a")->AsString=Edit4->Text;
DataModule1->IBStorOSNNew->Prepare();
DataModule1->IBStorOSNNew->ExecProc();
ShowMessage("Запись добавлена");
DataModule1->IBOsнова->Active=false;
DataModule1->IBOsнова->Active=true;
} }
if (ComboBox3->ItemIndex==1)
{
if(DBLookupComboBox1->Text=="")
ShowMessage("Введите, пожалуйста, город!");
else{
DataModule1->IBStorOSNUP->ParamByName("kod")->AsString=Edit2->Text;
DataModule1->IBStorOSNUP->ParamByName("g")->AsString=DBLookupComboBox1-
>KeyValue;
DataModule1->IBStorOSNUP->ParamByName("l")->AsString=Edit9->Text;
DataModule1->IBStorOSNUP->ParamByName("n")->AsString=Edit5->Text;
DataModule1->IBStorOSNUP->ParamByName("v")->AsString=Edit3->Text;
DataModule1->IBStorOSNUP->ParamByName("w")->AsString=Edit8->Text;
DataModule1->IBStorOSNUP->ParamByName("s")->AsString=Edit1->Text;
DataModule1->IBStorOSNUP->ParamByName("a")->AsString=Edit4->Text;
DataModule1->IBStorOSNUP->Prepare();
DataModule1->IBStorOSNUP->ExecProc();
ShowMessage("Запись изменена");
DataModule1->IBOsнова->Active=false;
DataModule1->IBOsнова->Active=true;
} }
if(ComboBox3->ItemIndex==2)
{
DataModule1->IBStorOSNDel->ParamByName("Kod")->AsString=Edit2->Text;
DataModule1->IBStorOSNDel->Prepare();
DataModule1->IBStorOSNDel->ExecProc();
ShowMessage("Запись удалена");
DataModule1->IBOsнова->Active=false;
DataModule1->IBOsнова->Active=true;}
}
//-----
void __fastcall TForm4::ComboBox3Click(TObject *Sender)
{
if((ComboBox3->ItemIndex==0) || (ComboBox3->ItemIndex==1))

```

```

{
Edit1->Text="";
Edit3->Text="";
Edit4->Text="";
Edit5->Text="";
Edit8->Text="";
Edit9->Text="";
Edit1->Enabled=true;
Edit3->Enabled=true;
Edit4->Enabled=true;
Edit5->Enabled=true;
Edit8->Enabled=true;
Edit9->Enabled=true;
DBLookupComboBox1->Enabled=true;
ShowMessage("Введите необходимые данные в соответствующие доступные поля!"); }
else { Edit1->Enabled=false;
Edit1->Text="Недоступно";
Edit3->Enabled=false;
Edit3->Text="Недоступно";
Edit4->Enabled=false;
Edit4->Text="Недоступно";
Edit5->Enabled=false;
Edit5->Text="Недоступно";
Edit8->Enabled=false;
Edit8->Text="Недоступно";
Edit9->Enabled=false;
Edit9->Text="Недоступно";
DBLookupComboBox1->Enabled=false;
ShowMessage("Введите код!");}
if(ComboBox3->ItemIndex==0)
Button7->Caption="Добавить";
if(ComboBox3->ItemIndex==1)
{ Button7->Caption="Редактировать";
ShowMessage("Введите номер!");
Edit2->Visible=true;
Label2->Visible=true;}
if(ComboBox3->ItemIndex==2)
{ Button7->Caption="Удалить";
Edit2->Visible=true;
Label2->Visible=true;}
}
//-----
void __fastcall TForm4::ComboBox1Click(TObject *Sender)
{
if(ComboBox1->ItemIndex==0)
{Memo1->Text="";
Memo1->Enabled=true;
LabeledEdit1->Text="";
LabeledEdit1->Enabled=true;
LabeledEdit2->Text="";
LabeledEdit2->Enabled=true;
DBLookupComboBox2->Enabled=true;

```

```

DBLookupComboBox5->Enabled=true;
DBGrid2->Columns->operator [](0)->Visible=false;
Button1->Caption="Добавить";
Edit7->Visible=false;
Label13->Visible=false;
ShowMessage("Выберите город, способ оплаты и введите все необходимые доступные
параметры!");}
if(ComboBox1->ItemIndex==1)
{Memo1->Text="";
Memo1->Enabled=true;
LabeledEdit1->Text="";
LabeledEdit1->Enabled=true;
LabeledEdit2->Text="";
LabeledEdit2->Enabled=true;
DBLookupComboBox2->Enabled=true;
DBLookupComboBox5->Enabled=true;
DBGrid2->Columns->operator [](0)->Visible=true;
Button1->Caption="Редактировать";
Edit7->Visible=true;
Label13->Visible=true;
ShowMessage("Введите номер!");}
if(ComboBox1->ItemIndex==2)
{DBGrid2->Columns->operator [](0)->Visible=true;
Memo1->Text="Недоступно";
Memo1->Enabled=false;
LabeledEdit1->Text="Недоступно";
LabeledEdit1->Enabled=false;
LabeledEdit2->Text="Недоступно";
LabeledEdit2->Enabled=false;
DBLookupComboBox2->Enabled=false;
DBLookupComboBox5->Enabled=false;
Edit7->Visible=true;
Label13->Visible=true;
Button1->Caption="Удалить";
ShowMessage("Введите номер!");}
}
//-----
void __fastcall TForm4::Button1Click(TObject *Sender)
{
if (Button1->Caption=="Добавить")
{
if((DBLookupComboBox2->Text=="")||(DBLookupComboBox5->Text==""))
ShowMessage("Введите, пожалуйста, город и способ оплаты!");
else{
DataModule1->IBStorOplNew->ParamByName("g")->AsString=DBLookupComboBox2-
>KeyValue;
DataModule1->IBStorOplNew->ParamByName("s")->AsString=DBLookupComboBox5-
>KeyValue;
DataModule1->IBStorOplNew->ParamByName("d")->AsString=Memo1->Text;
DataModule1->IBStorOplNew->ParamByName("n")->AsString=LabeledEdit1->Text;
DataModule1->IBStorOplNew->ParamByName("sa")->AsString=LabeledEdit2->Text;
DataModule1->IBStorOplNew->Prepare();
}
}
}

```



```

DataModule1->IBStorOplNew->ExecProc();
ShowMessage("Запись добавлена");
DataModule1->IBInfa->Active=false;
DataModule1->IBInfa->Active=true;
}
}
if (Button1->Caption=="Редактировать")
{
if(Edit7->Text=="")
ShowMessage("Введите, пожалуйста, номер записи!");
else{
DataModule1->IBStorOplUp->ParamByName("kod")->AsString=Edit7->Text;
DataModule1->IBStorOplUp->ParamByName("g")->AsString=DBLookupComboBox2-
>KeyValue;
DataModule1->IBStorOplUp->ParamByName("s")->AsString=DBLookupComboBox5-
>KeyValue;
DataModule1->IBStorOplUp->ParamByName("d")->AsString=Memo1->Text;
DataModule1->IBStorOplUp->ParamByName("n")->AsString=LabeledEdit1->Text;
DataModule1->IBStorOplUp->ParamByName("sa")->AsString=LabeledEdit2->Text;
DataModule1->IBStorOplUp->Prepare();
DataModule1->IBStorOplUp->ExecProc();
ShowMessage("Запись изменена");
DataModule1->IBInfa->Active=false;
DataModule1->IBInfa->Active=true;
}
}
if (Button1->Caption=="Удалить")
{
DataModule1->IBStorOplDel->ParamByName("Kod")->AsString=Edit7->Text;
DataModule1->IBStorOplDel->Prepare();
DataModule1->IBStorOplDel->ExecProc();
ShowMessage("Запись удалена");
DataModule1->IBInfa->Active=false;
DataModule1->IBInfa->Active=true;
}
}
//-----
void __fastcall TForm4::Button2Click(TObject *Sender)
{Form8->Show();}
//-----

```

Код органайзера событий:

```

#include <vcl.h>
#pragma hdrstop
#include "Sobitiy.h"
#include "Unit1.h"
#include "Forma.h"
//-----
#pragma package(smart_init)

```

```

#pragma resource "*.dfm"
TForm5 *Form5;
//-----
__fastcall TForm5::TForm5(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm5::Button1Click(TObject *Sender)
{
if(DBLookupComboBox1->Text!="")
DataModule1->IBStorOrg->ParamByName("v")->AsString=DBLookupComboBox1->Text;
DataModule1->IBStorOrg->ParamByName("n")->AsString=Memo1->Text;
DataModule1->IBStorOrg->ParamByName("dn")->AsString=Edit1->Text;
DataModule1->IBStorOrg->ParamByName("dk")->AsString=Edit2->Text;
if(DBLookupComboBox2->Text!="")
DataModule1->IBStorOrg->ParamByName("S")->AsString=DBLookupComboBox2->KeyValue;
if(DBLookupComboBox3->Text!="")
DataModule1->IBStorOrg->ParamByName("o")->AsString=DBLookupComboBox3->KeyValue;
if(DBLookupComboBox4->Text!="")
DataModule1->IBStorOrg->ParamByName("p")->AsString=DBLookupComboBox4->Text;
DataModule1->IBStorOrg->Prepare();
DataModule1->IBStorOrg->ExecProc();
ShowMessage("Запись добавлена");
DataModule1->IBOrg->Active=false;
DataModule1->IBOrg->Active=true;
}
//-----
void __fastcall TForm5::Button2Click(TObject *Sender)
{
if(DBLookupComboBox1->Text!="")
DataModule1->IBStorOrgUp->ParamByName("v")->AsString=DBLookupComboBox1->Text;
DataModule1->IBStorOrgUp->ParamByName("n")->AsString=Memo1->Text;
DataModule1->IBStorOrgUp->ParamByName("dn")->AsString=Edit1->Text;
DataModule1->IBStorOrgUp->ParamByName("dk")->AsString=Edit2->Text;
if(DBLookupComboBox2->Text!="")
DataModule1->IBStorOrgUp->ParamByName("s")->AsString=DBLookupComboBox3-
>KeyValue;
if(DBLookupComboBox3->Text!="")
DataModule1->IBStorOrgUp->ParamByName("o")->AsString=DBLookupComboBox3-
>KeyValue;
if(DBLookupComboBox4->Text!="")
DataModule1->IBStorOrgUp->ParamByName("p")->AsString=DBLookupComboBox4->Text;
DataModule1->IBStorOrgUp->Prepare();
DataModule1->IBStorOrgUp->ExecProc();
ShowMessage("Запись изменена");
DataModule1->IBOrg->Active=false;
DataModule1->IBOrg->Active=true;
}
//-----
void __fastcall TForm5::FormShow(TObject *Sender)
{

```

```
DataModule1->IBTable1->Filtered=false;
}
//-----
```

Код модуля «Сотрудники»:

```
#include <vcl.h>
#pragma hdrstop
#include "Sotrudniki.h"
#include "Unit1.h"
#include "Polz.h"
#include "Otchet.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm6 *Form6;
//-----
__fastcall TForm6::TForm6(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm6::ComboBox1Click(TObject *Sender)
{
if((ComboBox1->ItemIndex==0) || (ComboBox1->ItemIndex==1))
{
LabeledEdit1->Text="";
LabeledEdit2->Text="";
LabeledEdit3->Text="";
LabeledEdit4->Text="";
LabeledEdit6->Text="";
LabeledEdit1->Enabled=true;
LabeledEdit2->Enabled=true;
LabeledEdit3->Enabled=true;
LabeledEdit4->Enabled=true;
LabeledEdit6->Enabled=true;
ShowMessage("Введите необходимые данные в соответствующие доступные поля!"); }
else { LabeledEdit1->Enabled=false;
LabeledEdit1->Text="Недоступно";
LabeledEdit2->Enabled=false;
LabeledEdit2->Text="Недоступно";
LabeledEdit3->Enabled=false;
LabeledEdit3->Text="Недоступно";
LabeledEdit4->Enabled=false;
LabeledEdit4->Text="Недоступно";
ShowMessage("Введите уникальный номер сотрудника!"); }
if(ComboBox1->ItemIndex==0)
{ Button3->Caption="Добавить"; }
if(ComboBox1->ItemIndex==1)
{ Button3->Caption="Редактировать"; }
ShowMessage("Введите уникальный номер сотрудника!"); }
if(ComboBox1->ItemIndex==2)
```

```

{Button3->Caption="Удалить";}
}
//-----
void __fastcall TForm6::Button3Click(TObject *Sender)
{
if (ComboBox1->ItemIndex==0)
{
if(LabeledEdit6->Text=="")
ShowMessage("Введите, пожалуйста, уникальный номер сотрудника");
DataModule1->IBSotr->Filtered=false;
DataModule1->IBSotr->FilterOptions << foCaseInsensitive;
DataModule1->IBSotr->Filter="(kod = '"+LabeledEdit6->Text+"'";
DataModule1->IBSotr->Filtered=true;
if((LabeledEdit6->Text!="")&(DataModule1->DataSource5->DataSet->RecordCount!=0))
ShowMessage("Вы ввели не уникальный номер!");
else{
DataModule1->IBStorSotrNew->ParamByName("k")->AsString=LabeledEdit6->Text;
DataModule1->IBStorSotrNew->ParamByName("f")->AsString=LabeledEdit1->Text;
DataModule1->IBStorSotrNew->ParamByName("i")->AsString=LabeledEdit2->Text;
DataModule1->IBStorSotrNew->ParamByName("o")->AsString=LabeledEdit3->Text;
DataModule1->IBStorSotrNew->ParamByName("d")->AsString=LabeledEdit4->Text;
Form7->Show();
DataModule1->IBStorSotrNew->Prepare();
DataModule1->IBStorSotrNew->ExecProc();
ShowMessage("Запись добавлена");
DataModule1->IBSotr->Active=false;
DataModule1->IBSotr->Active=true;
Form7->Edit1->Text=LabeledEdit6->Text;
} }
if (ComboBox1->ItemIndex==1)
{
if(LabeledEdit6->Text=="")
ShowMessage("Введите, пожалуйста, уникальный номер сотрудника");
else{
DataModule1->IBStorSotrUp->ParamByName("k")->AsString=LabeledEdit6->Text;
DataModule1->IBStorSotrUp->ParamByName("f")->AsString=LabeledEdit1->Text;
DataModule1->IBStorSotrUp->ParamByName("i")->AsString=LabeledEdit2->Text;
DataModule1->IBStorSotrUp->ParamByName("o")->AsString=LabeledEdit3->Text;
DataModule1->IBStorSotrUp->ParamByName("d")->AsString=LabeledEdit4->Text;
DataModule1->IBStorSotrUp->Prepare();
DataModule1->IBStorSotrUp->ExecProc();
ShowMessage("Запись изменена");
DataModule1->IBSotr->Active=false;
DataModule1->IBSotr->Active=true;
} }
if(ComboBox1->ItemIndex==2)
{
DataModule1->IBStorSotrDel->ParamByName("Kod")->AsString=LabeledEdit6->Text;
DataModule1->IBStorSotrDel->Prepare();
DataModule1->IBStorSotrDel->ExecProc();
ShowMessage("Запись удалена");
DataModule1->IBSotr->Active=false;
}
}
}

```

```

DataModule1->IBSotr->Active=true;}
DataModule1->IBSotr->Filtered=false;
DataModule1->IBTable1->Filtered=false;
}
//-----
void __fastcall TForm6::Button5Click(TObject *Sender)
{
DataModule1->IBOrg->Filtered=false;
DataModule1->IBOrg->FilterOptions << foCaseInsensitive;
DataModule1->IBOrg->Filter="(Sozd="" +DBLookupComboBox5->KeyValue+"");
DataModule1->IBOrg->Filtered=true;
Form9->QuickRep1->Preview();
}
//-----
void __fastcall TForm6::ComboBox2Change(TObject *Sender)
{
if(ComboBox2->ItemIndex==0)
{DBLookupComboBox1->ListField="KOD";
J=DBLookupComboBox1->ListField;}
if(ComboBox2->ItemIndex==1)
{DBLookupComboBox1->ListField="Fam";
J=DBLookupComboBox1->ListField;}
if(ComboBox2->ItemIndex==2)
{DBLookupComboBox1->ListField="FIO";
J=DBLookupComboBox1->KeyField;}
}
//-----
void __fastcall TForm6::ComboBox3Change(TObject *Sender)
{
DBLookupComboBox2->CleanupInstance();
if(ComboBox3->ItemIndex==0)
{ DBLookupComboBox2->ListSource=DataModule1->DataSource6;
DBLookupComboBox2->KeyField="Datan";
DBLookupComboBox2->ListField="Datan";
J="Datan";}
if(ComboBox3->ItemIndex==1)
{ DBLookupComboBox2->ListSource=DataModule1->DataSource6;
DBLookupComboBox2->KeyField="Datak";
DBLookupComboBox2->ListField="Datak";
J="Datak";}
if(ComboBox3->ItemIndex==2)
{DBLookupComboBox2->ListSource=DataModule1->DataSource8;
DBLookupComboBox2->KeyField="Vid";
DBLookupComboBox2->ListField="Vid";
J="VID";}
if(ComboBox3->ItemIndex==3)
{DBLookupComboBox2->ListSource=DataModule1->DataSource7;
DBLookupComboBox2->KeyField="Naim";
DBLookupComboBox2->ListField="Naim";
J="Pa";}
if(ComboBox3->ItemIndex==4)
{DBLookupComboBox2->ListSource=DataModule1->DataSource5;

```

```

DBLookupComboBox2->KeyField="id";
DBLookupComboBox2->ListField="FIO";
J="otv";
}
if(ComboBox3->ItemIndex==5)
{DBLookupComboBox2->ListSource=DataModule1->DataSource5;
DBLookupComboBox2->KeyField="id";
DBLookupComboBox2->ListField="FIO";
J="sozd";}
}
//-----
void __fastcall TForm6::Button4Click(TObject *Sender)
{
DataModule1->IBSotr->Filtered=true;
}
//-----
void __fastcall TForm6::Button6Click(TObject *Sender)
{
DataModule1->IBOrg->Filtered=true;
}
//-----
void __fastcall TForm6::Button9Click(TObject *Sender)
{
DataModule1->IBOrg->Filtered=false;
}
//-----
void __fastcall TForm6::DBLookupComboBox2Click(TObject *Sender)
{
DataModule1->IBOrg->Filtered=false;
DataModule1->IBOrg->FilterOptions << foCaseInsensitive;
if((ComboBox3->ItemIndex==4)||((ComboBox3->ItemIndex==5))
{DataModule1->IBOrg->Filter="("+J+"="+DBLookupComboBox2->KeyValue+");"}
else
{DataModule1->IBOrg->Filter="("+J+"="+DBLookupComboBox2->Text+");"}
}
//-----
void __fastcall TForm6::Button8Click(TObject *Sender)
{
DataModule1->IBSotr->Filtered=false;
}
//-----
void __fastcall TForm6::DBLookupComboBox1Click(TObject *Sender)
{
DataModule1->IBSotr->Filtered=false;
DataModule1->IBSotr->FilterOptions << foCaseInsensitive;
if(ComboBox2->ItemIndex==2)
DataModule1->IBSotr->Filter="("+J+"="+DBLookupComboBox1->KeyValue+");"
else
DataModule1->IBSotr->Filter="("+J+"="+DBLookupComboBox1->Text+");"
}
//-----
void __fastcall TForm6::FormCreate(TObject *Sender)

```

```

{
LabeledEdit5->Visible=false;
}
//-----
void __fastcall TForm6::ComboBox4Click(TObject *Sender)
{
if(ComboBox4->ItemIndex==0)
{Form7->Show();
LabeledEdit5->Visible=false;}
if(ComboBox4->ItemIndex==1)
{
Button2->Caption="Изменить";
ShowMessage("Введите номер пользователя, нажмите изменить и в новой форме исправьте
необходимые данные.");
LabeledEdit5->Visible=true;
}
if(ComboBox4->ItemIndex==2)
{
Button2->Caption="Удалить";
ShowMessage("Введите номер пользователя для удаления.");
LabeledEdit5->Visible=true;
}
}
//-----
void __fastcall TForm6::Button2Click(TObject *Sender)
{
if(ComboBox4->ItemIndex==1)
Form7->Show();
if(ComboBox4->ItemIndex==2)
{DataModule1->IBStorPolDel->ParamByName("KOD")->AsString=LabeledEdit5->Text;
DataModule1->IBStorPolDel->Prepare();
DataModule1->IBStorPolDel->ExecProc();
ShowMessage("Запись удалена");
DataModule1->IBTable1->Active=false;
DataModule1->IBTable1->Active=true;}
}
//-----
void __fastcall TForm6::Button1Click(TObject *Sender)
{
LabeledEdit1->Text="";
LabeledEdit2->Text="";
LabeledEdit3->Text="";
LabeledEdit4->Text="";
LabeledEdit6->Text="";
LabeledEdit1->Enabled=true;
LabeledEdit2->Enabled=true;
LabeledEdit3->Enabled=true;
LabeledEdit4->Enabled=true;
LabeledEdit6->Enabled=true;
ComboBox1->Text="";
}
//-----

```

Код модуля «Пользователи»:

```
#include <vcl.h>
#pragma hdrstop
#include "Polz.h"
#include "Unit1.h"
#include "Sotrudniki.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm7 *Form7;
//-----
__fastcall TForm7::TForm7(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm7::FormShow(TObject *Sender)
{
if((Form6->ComboBox4->ItemIndex==0) || (Form6->ComboBox1->ItemIndex==0))
{Edit1->Text="";
LabeledEdit1->Text="";
LabeledEdit2->Text="";
DBLookupComboBox1->KeyValue=NULL();
}
if(Form6->ComboBox4->ItemIndex==1)
{
DataModule1->IBTable1->Filtered=false;
DataModule1->IBTable1->FilterOptions << foCaseInsensitive;
DataModule1->IBTable1->Filter="(KOD="+Form6->LabeledEdit5->Text+")";
DataModule1->IBTable1->Filtered=true;
LabeledEdit1->Text=Trim(DataModule1->IBTable1->FieldByName("Login")->AsString);
LabeledEdit2->Text=Trim(DataModule1->IBTable1->FieldByName("parol")->AsString);
DataModule1->IBSotr->Filtered=false;
DataModule1->IBSotr->FilterOptions << foCaseInsensitive;
DataModule1->IBSotr->Filter="(ID="+DataModule1->IBTable1->FieldByName("sotr")->AsString+")";
DataModule1->IBSotr->Filtered=true;
Edit1->Text=Trim(DataModule1->IBSotr->FieldByName("KOD")->AsString);
DBLookupComboBox1->KeyValue=DataModule1->IBTable1->FieldByName("rol")->AsString;
DataModule1->IBSotr->Filtered=false;
}
}
//-----
void __fastcall TForm7::Button1Click(TObject *Sender)
{
if(Form6->ComboBox4->ItemIndex==0)
{
DataModule1->IBTable1->Filtered=false;
DataModule1->IBTable1->FilterOptions << foCaseInsensitive;
DataModule1->IBTable1->Filter="(login ="+LabeledEdit1->Text+")";
DataModule1->IBTable1->Filtered=true;
}
```



```

if(DataModule1->DataSource1->DataSet->RecordCount!=0)
ShowMessage("Такой логин уже существует. Введите уникальные данные!");
else{
if(DBLookupComboBox1->Text=="")
ShowMessage("Вы не указали роль пользователя!");
else{
DataModule1->IBStorPolNew->ParamByName("l")->AsString=LabeledEdit1->Text;
DataModule1->IBStorPolNew->ParamByName("p")->AsString=LabeledEdit2->Text;
DataModule1->IBStorPolNew->ParamByName("po")->AsString=DBLookupComboBox1->Text;
DataModule1->IBSotr->Filtered=false;
DataModule1->IBSotr->FilterOptions << foCaseInsensitive;
DataModule1->IBSotr->Filter="(kod ="+Edit1->Text+"";
DataModule1->IBSotr->Filtered=true;
if((Edit1->Text=="")||(DataModule1->DataSource5->DataSet->RecordCount==0))
ShowMessage("Вы не указали уникальный номер сотрудника или введенный номер не
существует!");
else{
DataModule1->IBStorPolNew->ParamByName("s")->AsString=DataModule1->IBSotr-
>FieldByName("id")->AsString;
DataModule1->IBStorPolNew->Prepare();
DataModule1->IBStorPolNew->ExecProc();
ShowMessage("Запись добавлена");
DataModule1->IBSotr->Filtered=false;
DataModule1->IBTable1->Filtered=false;
DataModule1->IBTable1->Active=false;
DataModule1->IBTable1->Active=true;} } }
if(Form6->ComboBox4->ItemIndex==1)
{
DataModule1->IBTable1->Filtered=false;
DataModule1->IBTable1->FilterOptions << foCaseInsensitive;
DataModule1->IBTable1->Filter="(login ="+LabeledEdit1->Text+"";
DataModule1->IBTable1->Filtered=true;
if(DataModule1->DataSource1->DataSet->RecordCount>1)
ShowMessage("Такой логин уже используется другим пользователем. Введите уникальные
данные!");
else{
DataModule1->IBStorPolUp->ParamByName("l")->AsString=LabeledEdit1->Text;
DataModule1->IBStorPolUp->ParamByName("p")->AsString=LabeledEdit2->Text;
DataModule1->IBSotr->Filtered=false;
DataModule1->IBSotr->FilterOptions << foCaseInsensitive;
DataModule1->IBSotr->Filter="(kod ="+Edit1->Text+"";
DataModule1->IBSotr->Filtered=true;
if((Edit1->Text=="")||(DataModule1->DataSource5->DataSet->RecordCount==0))
ShowMessage("Вы не указали уникальный номер сотрудника или введенный номер не
существует!");
else{
DataModule1->IBStorPolNew->ParamByName("r")->AsString=DBLookupComboBox1->Text;
DataModule1->IBStorPolUp->ParamByName("s")->AsString=DataModule1->IBSotr-
>FieldByName("id")->AsString;
DataModule1->IBStorPolUp->ParamByName("kod")->AsString=Form6->LabeledEdit5->Text;
DataModule1->IBStorPolUp->Prepare();
DataModule1->IBStorPolUp->ExecProc();

```

```

ShowMessage("Запись изменена");
DataModule1->IBSotr->Filtered=false;
DataModule1->IBTable1->Filtered=false;
DataModule1->IBTable1->Active=false;
DataModule1->IBTable1->Active=true;
Form6->LabeledEdit5->Text="";}}}
}
//-----

```

Код модуля «Способ оплаты»:

```

#include <vcl.h>
#pragma hdrstop

#include "SPOPL.h"
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm8 *Form8;
//-----
__fastcall TForm8::TForm8(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm8::ComboBox1Change(TObject *Sender)
{
if(ComboBox1->ItemIndex==0)
{Button2->Caption="Добавить";
Edit2->Enabled=false;
Edit2->Text="Недоступно";
Edit1->Enabled=true;
Edit1->Text="";}
if(ComboBox1->ItemIndex==1)
{Button2->Caption="Изменить";
Edit2->Enabled=true;
Edit2->Text="";
Edit1->Enabled=true;
Edit1->Text="";}
if(ComboBox1->ItemIndex==2)
{Button2->Caption="Удалить";
Edit2->Enabled=true;
Edit2->Text="";
Edit1->Enabled=false;
Edit1->Text="Недоступно";}
}
//-----
void __fastcall TForm8::Button2Click(TObject *Sender)
{
if(ComboBox1->ItemIndex==0)
{

```

```

DataModule1->IBPlat->Filtered=false;
DataModule1->IBPlat->FilterOptions << foCaseInsensitive;
DataModule1->IBPlat->Filter="(naim="" +Edit1->Text+)";
DataModule1->IBPlat->Filtered=true;
if(DataModule1->DataSource3->DataSet->RecordCount!=0)
ShowMessage("Данный способ оплаты уже существует!");
else{
DataModule1->IBStoredOplatVNew->ParamByName("n")->AsString=Edit1->Text;
DataModule1->IBStoredOplatVNew->Prepare();
DataModule1->IBStoredOplatVNew->ExecProc();
ShowMessage("Запись добавлена");
DataModule1->IBPlat->Active=false;
DataModule1->IBPlat->Active=true;
}}
if(ComboBox1->ItemIndex==1)
{
DataModule1->IBStoredOplatVUp->ParamByName("kod")->AsString=Edit2->Text;
DataModule1->IBStoredOplatVUp->ParamByName("n")->AsString=Edit1->Text;
DataModule1->IBStoredOplatVUp->Prepare();
DataModule1->IBStoredOplatVUp->ExecProc();
ShowMessage("Запись изменена");
DataModule1->IBPlat->Active=false;
DataModule1->IBPlat->Active=true;
}
if(ComboBox1->ItemIndex==2)
{
DataModule1->IBStoredOplatVDel->ParamByName("k")->AsString=Edit2->Text;
DataModule1->IBStoredOplatVDel->Prepare();
DataModule1->IBStoredOplatVDel->ExecProc();
ShowMessage("Запись удалена");
DataModule1->IBPlat->Active=false;
DataModule1->IBPlat->Active=true;
}
DataModule1->IBPlat->Filtered=false;
}
//-----

```

Код окна «Заставка»:

```

#include <vcl.h>
#pragma hdrstop
#include "Splash_form.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TSplashForm *SplashForm;
//-----
__fastcall TSplashForm::TSplashForm(TComponent* Owner)
: TForm(Owner)
{
}

```

```

//-----
void __fastcall TSplashForm::Timer1Timer(TObject *Sender)
{
    static int SplashTime = 3000;
    if((::GetTickCount() - FStart > SplashTime) && CanCloseSplash)
    {
        this->Close();
    }
}
//-----
void __fastcall TSplashForm::FormCreate(TObject *Sender)
{
    FStart = ::GetTickCount();
    CanCloseSplash = false;
    Timer1->Interval = 100;
    FormStyle = fsStayOnTop;
}
//-----
void __fastcall TSplashForm::FormClose(TObject *Sender,
    TCloseAction &Action)
{
    Action = caFree;
}
//-----
void __fastcall TSplashForm::FormDestroy(TObject *Sender)
{
    SplashForm = 0;
}
//-----

```

Код проекта:

```

#include <vcl.h>
#pragma hdrstop
#include "Splash_form.h"
//-----
USEFORM("Glavnay.cpp", Form1);
USEFORM("Unit1.cpp", DataModule1); /* TDataModule: File Type */
USEFORM("Forma.cpp", Form2);
USEFORM("Parcovki.cpp", Form3);
USEFORM("Redact.cpp", Form4);
USEFORM("Sobitiy.cpp", Form5);
USEFORM("Sotrudniki.cpp", Form6);
USEFORM("Polz.cpp", Form7);
USEFORM("SPOPL.cpp", Form8);
USEFORM("Otchet.cpp", Form9);
USEFORM("Splash_form.cpp", SplashForm);
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try

```

```

{
    SplashForm = new TSplashForm(0);
    SplashForm->Show();
    Application->Initialize();
    Application->Title = "FIT";
        Application->CreateForm(__classid(TForm1), &Form1);
        Application->CreateForm(__classid(TForm6), &Form6);
        Application->CreateForm(__classid(TForm7), &Form7);
        Application->CreateForm(__classid(TForm4), &Form4);
        Application->CreateForm(__classid(TForm8), &Form8);
        Application->CreateForm(__classid(TForm5), &Form5);
        Application->CreateForm(__classid(TForm3), &Form3);
        Application->CreateForm(__classid(TForm2), &Form2);
        Application->CreateForm(__classid(TDataModule1), &DataModule1);
        Application->CreateForm(__classid(TForm9), &Form9);
        if(SplashForm)
        {
            SplashForm->CanCloseSplash = true;
        }

        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    catch (...)
    {
        try
        {
            throw Exception("");
        }
        catch (Exception &exception)
        {
            Application->ShowException(&exception);
        }
    }
    return 0;
}
//-----

```

Приложение В



Рисунок В1 – Заставка приложения

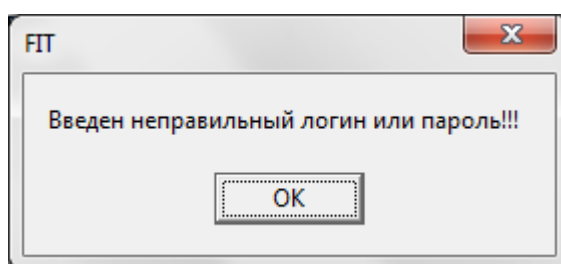


Рисунок В2 – Извещение об ошибке



Рисунок В3 – Поиск события с этапом выполнения «Запланировано»

Ответно-вопросная страница

Веберите город:

Оплата парковки | **Время работы парковок и стоимость** | Льготы на пользование платными парковками | Приобретение абонемента

Время работы платных парковок в г.Белгород: с 8.00 до 20.00 (кроме выходных и праздничных дней)

Предоставляется 30 минут на оплату парковки.

Стоимость парковки: 1 час 30 рублей

Получить ответ Сбросить

Рисунок В4 – Данные по времени работы парковок и их стоимости

Ответно-вопросная страница

Веберите город:

Оплата парковки | Время работы парковок и стоимость | **Льготы на пользование платными парковками** | Приобретение абонемента

Для того чтобы получить льготы на пользование платными парковками в г.Белгород необходимо обратиться в

Белгородское парковочное пространство по адресу ул. Вокзальная, д. 33 А или позвонить по номеру 33-50-26

Получить ответ Сбросить

Рисунок В5 – Данные по льготам

Ответно-вопросная страница

Веберите город:

Оплата парковки | Время работы парковок и стоимость | Льготы на пользование платными парковками | **Приобретение абонемента**

Приобрести резидентского абонемента на пользование платными парковками можно, обратившись

по адресу ул. Вокзальная, д. 33 А или позвонить по номеру 33-50-26

Получить ответ Сбросить

Рисунок В6 – Данные по способам приобретения абонементов

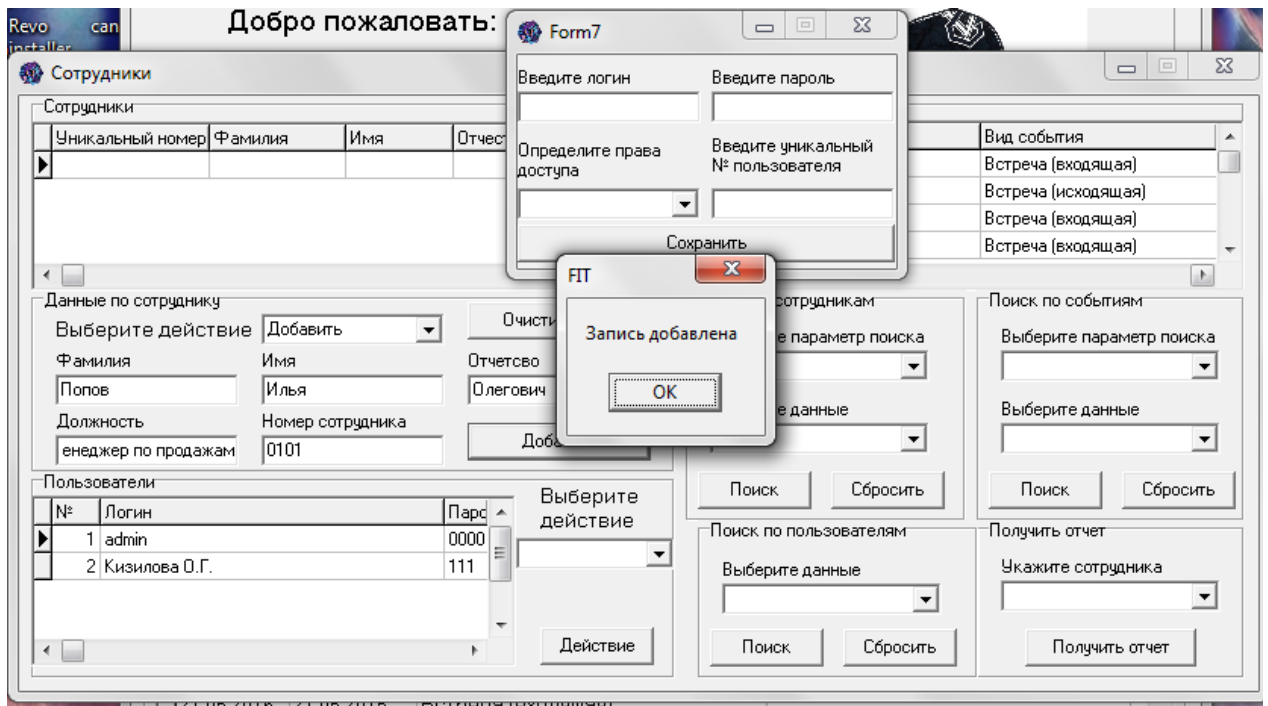


Рисунок В7 – Добавление нового пользователя

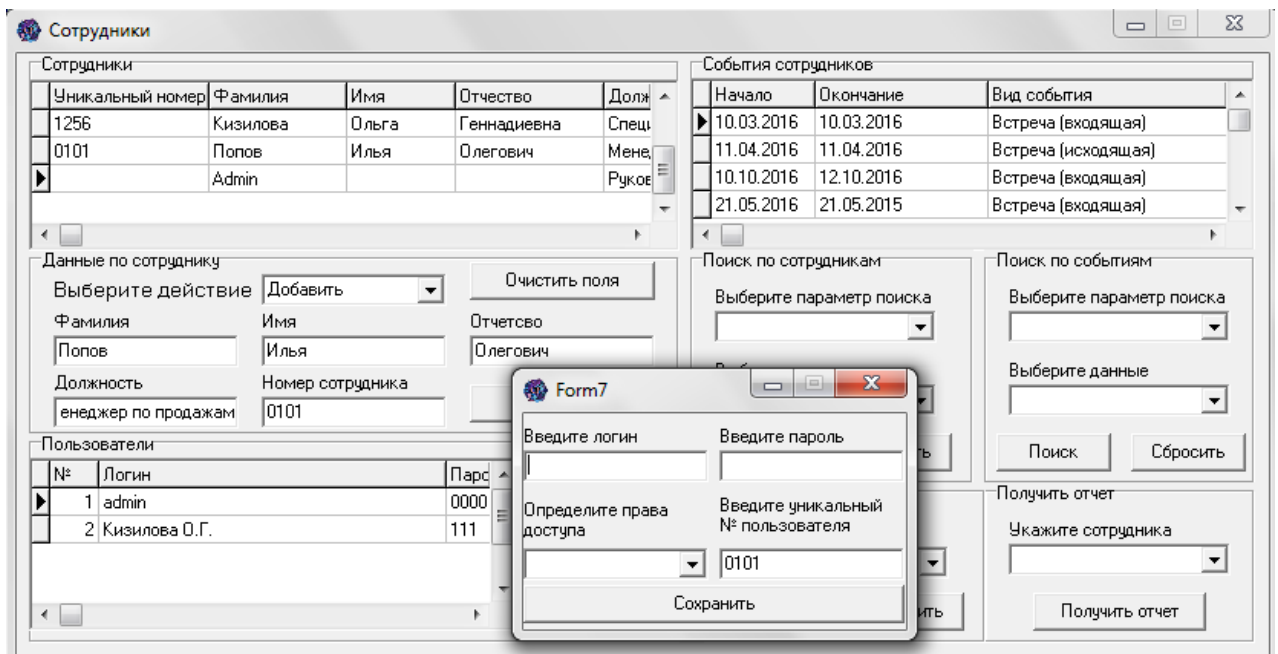


Рисунок В8 – Открывшееся окно добавления пользователя

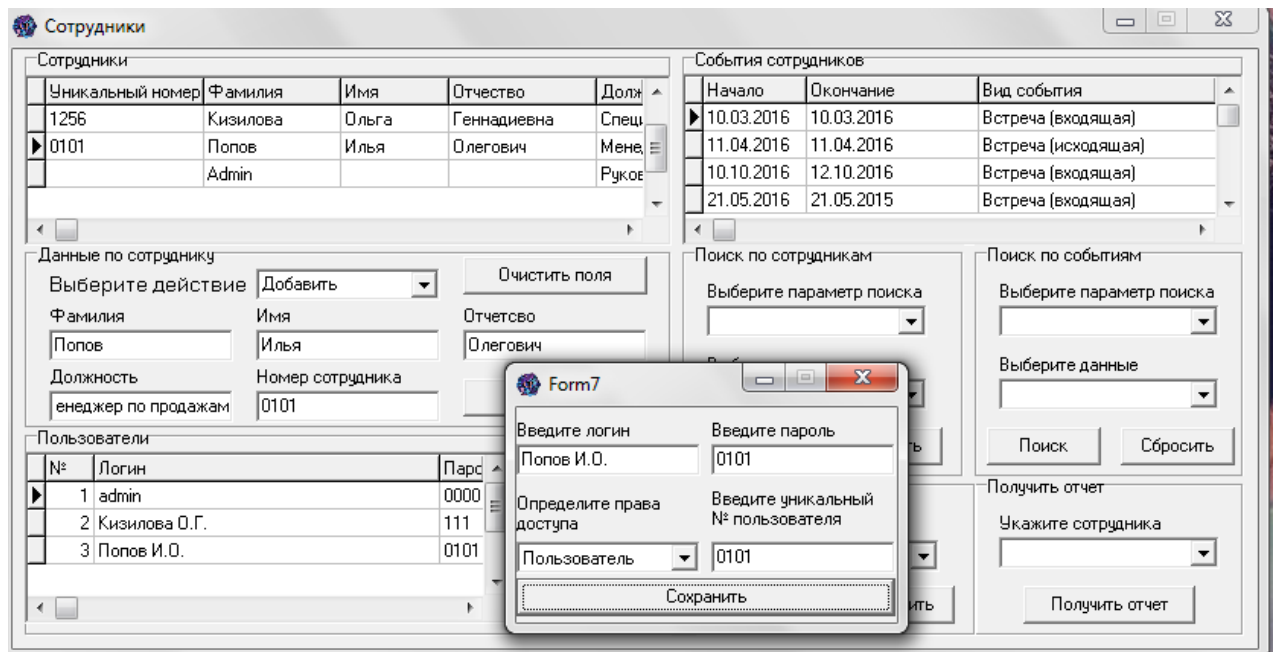


Рисунок В9 – Добавление нового пользователя

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

« ___ » _____ Г.

(подпись) (Ф.И.О.)