



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ  
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра прикладной информатики  
Институт информационных систем и геотехнологий

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**  
(бакалаврская работа)  
по направлению подготовки 09.03.03 Прикладная информатика  
(квалификация – бакалавр)

На тему «Разработка информационной системы автоматизации бронирования транспорта»

Исполнитель Перевертов Константин Сергеевич

Руководитель к.т.н., доцент кафедры ПИ Яготинцева Наталья Владимировна

«К защите допускаю»

заведующий кафедрой \_\_\_\_\_

доктор технических наук, профессор

Истомин Евгений Петрович

« \_\_\_\_ » \_\_\_\_\_ 2020 г.

Санкт-Петербург  
2020

## ОГЛАВЛЕНИЕ

Введение.....	3
1 Анализ объекта и предмета исследования .....	6
1.1 Описание объекта исследования.....	6
1.2 Описание предмета исследования .....	9
1.3 Постановка задачи на проектирование и информатизацию исследуемого процесса.....	10
2 Проектирование информационной системы.....	16
2.1 Анализ существующего бизнес-процесса по бронированию .....	16
2.2 Анализ существующих разработок .....	17
2.3 Предполагаемая технология выполнения исследуемого процесса .....	20
2.4 Обоснования выбора средств разработки.....	24
3 Разработка частного решения.....	27
3.1 Описание технологии разработки интерфейсов приложения.....	27
3.2 Разработка базы данных. Создание таблиц, установка связей .....	35
3.3 Разработка базы данных. Хранимые процедуры .....	38
3.4 Алгоритм работы информационной системы .....	41
4 Экономическое обоснование .....	59
Заключение .....	64
Список используемой литературы .....	66
Приложения .....	69

## Введение

Современный мир информационных технологий трудно представить себе без использования информации в электронном виде.

Информация — сведения, воспринимаемые человеком как отражение фактов материального мира в процессе коммуникации и передаваемые людьми устным, письменным или другим способом. Включает обмен сведениями между людьми, человеком и автоматом, обмен сигналами в животном и растительном мире, передачу признаков на клеточном уровне (например, генетическая информация).

Но каким образом, кроме традиционных, информация передавалась в нашем обществе? Если говорить об истоках – это рисунки, летопись, позже это различного вида литература на бумажном носителе. В современном же обществе кроме вышеназванных, используются электронные накопители информации.

Используя электронные накопители информации человек получил возможность не только передавать, принимать информацию, но и анализировать, преобразовывать ее. В помощь человеку в этом процессе пришли компьютерные технологии.

Сейчас организовать хранение и обработку информации на электронном носителе возможно самыми различными способами. Если целью хранения информации является ее анализ, преобразование, то одним из способов который является самым предпочтительным станет хранение информации в базе данных. Хранение информации в базе данных помогает ее систематизировать и анализировать гораздо быстрее по сравнению с другими носителями информации.

Базы данных создаются обычно не для решения какой-либо одной задачи для одного пользователя, а для многоцелевого использования. Эта информация должна по возможности фиксироваться в базе данных однократно, и все пользователи, которым эта информация нужна, должны иметь возможность

работать с ней. Отсутствие централизованных методов управления доступом к информации послужило причиной разработки систем управления базой данных, а сами хранилища информации, которые работали под управлением данных систем, назывались базами или банками данных.

Для того чтобы правильно использовать информацию из баз данных создаются информационные системы. Их цель – помощь пользователю в работе с информацией. Это может быть, как введение, сохранение, преобразование, удаление, так и получение аналитических данных, например, отчетов.

Актуальность разработки информационной системы обусловлена отсутствием автоматизированной системы бронирования заявок на транспорт в организации.

Основной целью выпускной квалификационной работы является разработка информационной системы автоматизации бронирования транспорта. Дополнительной целью является разработка информационной системы, которая может использоваться не только на выбранном объекте исследования, но и в других организациях похожей структуры.

Объектом исследования является Государственное бюджетное учреждение здравоохранения «Туапсинская центральная районная больница № 1» МЗ КК.

Предметом исследования является деятельность подразделения, которое находится в подчинении начальника гаража и включает в себя транспортные средства.

Основные задачи:

- Анализ объекта и предмета исследования: описание объекта исследования и предмета исследования, постановка задачи на проектирование и информатизацию исследуемого процесса
- Анализ существующих разработок.
- Описание предполагаемой модели исследуемого процесса
- Описание видов обеспечения предполагаемой модели

- Обоснование выбора средств разработки
- Проектирование интерфейса пользователя/запросы
- Разработка информационной системы

Автоматизация данного процесса позволит сотрудникам избавиться от многих рутинных процессов, что обеспечит наиболее быструю, полную и качественную работу с транспортными средствами, даст возможность оперативно получать всю необходимую информацию.

## 1 Анализ объекта и предмета исследования

### 1.1 Описание объекта исследования

Государственное бюджетное учреждение здравоохранения «Туапсинская центральная районная больница № 1» МЗ КК [10] (далее - ГБУЗ «ТЦРБ № 1» МЗ КК) - крупное многопрофильное лечебно-профилактическое учреждение. Специалисты ГБУЗ «ТЦРБ № 1» МЗ КК оказывают медицинские услуги не только по 29 специальностям в рамках Территориальной программы государственных гарантий оказания медицинской помощи, но и в рамках добровольного медицинского страхования и на платной основе.

Структура учреждения включает в себя следующие основные подразделения:

- Стационар (приемное отделение, отделение переливания крови, терапевтическое, инфекционное, гинекологическое, хирургическое, травматологическое, эндоскопическое, анестезиолого-реанимационное, патологоанатомическое отделения, а также отделение восстановительного лечения)
- Межтерриториальный педиатрический центр (детское соматическое отделение)
- Консультативно диагностический центр (клинико-диагностическая лаборатория, отделение функциональной диагностики, отделение ультразвуковой диагностики, рентгенологическое отделение)
- Межтерриториальный перинатальный центр (акушерское отделение)
- Поликлиническая служба (поликлиника № 1, поликлиника № 2, женская консультация, детская поликлиника, 6 амбулаторий, 10 фельдшерско-акушерских пунктов)
- Первичный сосудистый центр (неврологическое, кардиологическое отделения)

- Вспомогательные службы (экономический, юридический отделы, отдел кадров, информатизации, закупок, гараж и пр.)

Ежегодно больница принимает на стационарное лечение более 13000 пациентов, выполняет более 126000 диагностических и лабораторных исследований.

Начиная с 2011 года ГБУЗ «ТЦРБ № 1» МЗ КК участвует в информатизации учреждений здравоохранения Краснодарского края, что позволило учреждению организовать крупную сеть, которая объединяет в себя все структурные подразделения. На данный момент каждое рабочее место врача специалиста, медицинского регистратора оснащено терминальной станцией, принтером. С помощью данного оборудования производится подключение к серверу медицинской информационной системы в которой ведется электронная медицинская карта пациента. В 2020 году планируется оснащение подобным оборудованием рабочих мест медицинских сестер, а также процедурных кабинетов, кабинетов функциональной диагностики и лаборатории.

Рабочие места врачей специалистов и медицинского персонала функционируют в специальной защищенной сети, настройки которой регламентированы Министерством здравоохранения Краснодарского края.

Типовая схема подключения представлена на рисунке 1.1.1

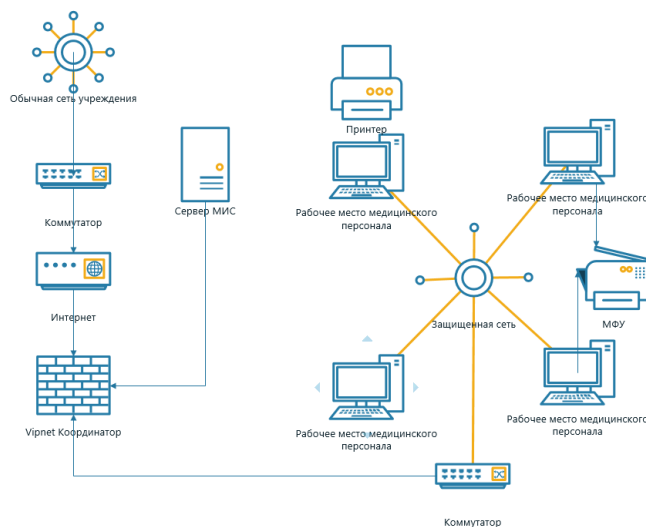


Рисунок 1.1.1 - Типовая схема подключения

Помимо медицинского персонала персональными компьютерами разной комплектации оснащены и рабочие места вспомогательных служб. На каждом компьютере установлено типовое программное обеспечение, разрешенное для использования юридическим лицом. Как правило это операционная система семейства Windows либо Linux, офисные программы для работы с текстовой информацией, почтовые программы и программы узкой направленности, например, «Крипто Про», позволяющая использовать ключ электронной подписи и др.

Каждая служба или отделение учреждения выполняет поставленные перед ними задачи для достижения определенных показателей влияющих на работу учреждения в целом.

Руководителем учреждения является главный врач. В его подчинении находятся заместители по различным направлениям, а также отдельные начальник служб. Общая схема управления учреждением представлена на рисунке 1.1.2

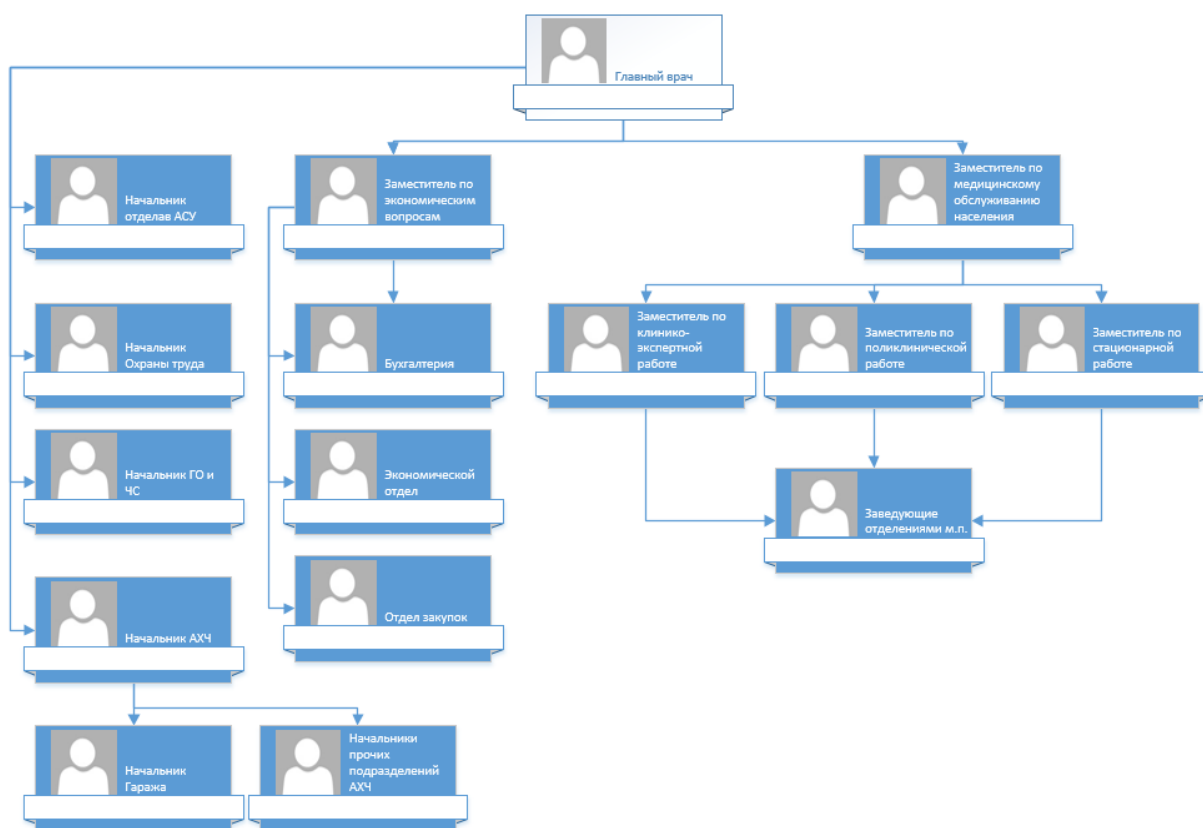


Рисунок 1.1.2 - Схема управления учреждением



## 1.2 Описание предмета исследования

Предметом исследования является деятельность подразделения, которое находится в подчинении начальника транспортного парка. Наличие автопарка в медицинском учреждении является его неотъемлемой частью. Транспорт работает в круглосуточном режиме и используется для следующих целей:

- Транспортировка пациентов
- Транспортировка персонала для оказания медицинской помощи
- Транспортировка прочего персонала для выполнения поручений
- Транспортировка груза: документация, медицинские анализы, медицинское оборудование и пр.

Для достижения этих целей на учете в учреждения находятся различные виды транспортных средств: легковые автомобили, автомобили скорой медицинской помощи, передвижная флюорографическая установка, автомобили повышенной проходимости, грузопассажирские автомобили повышенной проходимости и др.

Результат деятельности подразделения влияет на различные показатели, но главным из них является - качество оказания медицинской помощи. Именно этот показатель отражает работу учреждения в целом.

Деятельность выбранного подразделения зависит от поставленных перед ним задач. Инициатором может выступать заявка от руководителя учреждения, его заместителя или начальника отдела. Основанием для выполнения задачи является разрешение начальника гаража. Критерием при разрешении является в первую очередь срочность выполнения задачи и ее характер. Дополнительными критериями считается место назначения работы, выбранное транспортное средство.

Организация подачи заявки для выполнения работ происходит по следующему алгоритму:

- Инициатор принимает решение о необходимости выполнения работы с использованием транспортного средства;

- В режиме телефонного звонка или при личной встрече с начальником гаража обсуждается характер выполнения работ и место выполнения работ;
- Заявка вносится в журнал «Поступившие заявки»;
- Посредством журнала заявок в бумажном виде начальником гаража подбирается оптимальное транспортное средство;
- Начальник гаража уведомляет инициатора о разрешении на выполнение его заявки или об отказе на этапе формирования первичных форм, например, «Путевой лист»;
- Все разрешенные заявки вносятся в специальный журнал начальником гаража, согласно которому в письменном виде подготавливается общая заявка на транспорт в выбранный период и путевой лист оператором;

Немаловажными факторами, которые повлияли на выбор подразделения объектом информатизации являются:

- Отсутствие возможности оперативного подбора транспорта для выполнения работ в связи с большим количеством времени потраченного на подбор автомобиля;
- Отсутствие возможности получения информации о загрузке транспортного средства и отсутствие возможности оперативно предоставлять руководству отчетность об использовании транспортных средств;
- Наличие автоматизированных рабочих мест для организации автоматизации процесса работы с заявками;
- Наличие требования о предоставлении отчетных форм для руководства в электронном(печатном) виде.

### 1.3 Постановка задачи на проектирование и информатизацию исследуемого процесса

Разрабатываемая информационная система (далее - ИС) должна служить

для автоматизации процесса бронирования транспорта, а также анализа деятельности транспортных средств.

Назначение информационной системы: создание единого информационного пространства в организации; автоматизация процесса сбора, хранения и обработки информации о заявках на транспортные средства.

Цель информационной системы:

- Создание единого информационного пространства;
- Уменьшение финансовых расходов на приобретение расходных материалов;
- Уменьшение временных затрат, связанных с созданием и утверждением заявок;
- Планирование использования транспортного средства;
- Анализ нагрузок на транспортные средства предприятия;
- Прогнозирование проведения ремонтных работ транспортного средства;
- Получение статистической отчетности по различным критериям;

Объектом автоматизации для АИС являются отделы учреждения, у которых возникает потребность в автоматизации следующих функций:

- Ведение информационной базы о заявках на транспорт;
- Планирование использования транспортного средства;
- Получение статистической отчетности по различным критериям;

К пользователям системы относятся:

- Сотрудники служб;
- Заведующий парком транспортных средств;
- Руководитель учреждения;

Требование к разрабатываемой системе:

- ИС должна поддерживать одновременную работу нескольких пользователей одновременно;

- Система должна быть построена по технологии «клиент-сервер». Компоненты системы включают в себя автоматизированные рабочие места пользователя (далее - АРМ) и сервер базы данных (далее - БД) с установленной системой управления базами данных (далее - СУБД) и базой данных;

- Система должна поддерживать разграничение функционала в зависимости от профиля пользователя;

Требование к СУБД:

- Клиент - серверное взаимодействие с клиентскими приложениями по сетевым протоколам TCP/IP;

- Поддержка выполнения хранимых процедур;

- Разграничение прав доступа пользователей на уровне базы данных в соответствии с политикой безопасности системы;

- Создание резервной копии базы данных на случай экстренного восстановления;

- Достаточную производительность для комфортной работы системы в режиме групповой работы;

Требование к базе данных системы. Должна обеспечивать:

- Логическую целостность данных системы при любых действиях со стороны пользователей системы;

- Логическое разделение информации для различных групп и профилей пользователей системы;

Требование к клиентскому приложению:

- Подключение пользовательского приложения к СУБД и базе данных системы с использованием индивидуальной строки подключения;

- Проверка доступности подключения к серверу базы данных;

- Организация обмена данными между клиентским приложением и базой данных;

- Разграничение прав доступа пользователей к данным в соответствии с функциональными требованиями;

- Поддержка 4 основных статусов для заявок: заявка сформирована, дубликат, заявка на утверждении, заявка в работе;
- Выгрузка отчетности в формате .rtf и .pdf по критериям;
- Возможность просмотра изображений транспортного средства в формате .jpg , расположенных в локальной сети учреждения;
- Возможность настройки индивидуального цветового отображения заявок в зависимости от статуса заявки. В том числе сохранение, редактирование набора цветов и возвращения цветового отображения по умолчанию. Набор цветового отображения должен храниться в папке с исполняемым файлом;
- Возможность группового изменения статуса заявок;
- Возможность дублирования выбранной заявки;
- Созданная заявка с длительностью более 1 дня должна создаваться в разрезе каждого дня отдельно;
- Поддержка 2 основных профилей доступа к данным: пользователь, администратор. К особым функциям относятся: - профиль прав «Пользователь» позволяет управлять только собственными заявками; - профиль «Администратор» позволяет управлять всеми заявками ИС.

Профиль доступа к данным «Пользователь». Обеспечить доступ к следующим функциям:

- Проверка доступности сервера БД, авторизация пользователя;
- Настройка палитры цветов для отображения статуса заявок;
- Создание, редактирование, удаление заявки на транспортное средство;
- Поиск заявки;

Профиль доступа к данным «Администратор». Обеспечить доступ к следующим дополнительным функциям:

- Наполнение, редактирование справочника «Службы»;
- Наполнение, редактирование справочника «Сотрудники»;

- Наполнение, редактирование справочника «Транспорт»;
- Утверждение заявки;
- Формирование отчетных форм;
- Создание резервной копии базы данных;

Список сведений для хранения в базе данных:

- Сведения о сотрудниках: ФИО сотрудника, наименование службы, пароль для входа в систему, профиль прав;
- Сведения о службах: наименование службы, номер телефона службы;
- Сведения о транспортном средстве: марка, государственный номер, изображение;
- Сведения о заявке: дата и время выезда, маршрут, состав бригады, дата и время возвращения, пробег, данные о сотруднике, вид транспортного средства, цель, государственный номер транспортного средства;
- Сведения о резервном копировании: дата создания, результат, сведения о сотруднике, путь к сохраненной копии БД;

Список отчетных форм и требований представлен в таблице 1.3.1

Таблица 1.3.1 – Список отчетных форм

Наименование формы	Критерии построения	Данные для отображения
Ф-1 Заявка на транспорт	Дата выезда Статус заявки	Дата и время выезда; Маршрут; Состав бригады; Дата и время возвращения; Пробег; Старший бригады; Вид транспортного средства; Цель
Ф-2 Приказ наряд	Дата выезда Статус заявки	Время выезда; Наименование службы; Ответственный сотрудник; Дата и время возвращения; Вид транспорта; Государственный номер; Маршрут

Продолжение таблицы 1.3.1

Ф-3 Анализ нагрузки на транспортное средство	Период выезда Государственный номер	Сводная информация о: - количество выполненных заявок; - суммарный пробег; - вид транспортного средства; - государственный номер;
Ф-4 Заявки по службам	Период выезда Статус заявки	- наименование службы - количество заявок - статус заявки

## 2 Проектирование информационной системы

### 2.1 Анализ существующего бизнес-процесса по бронированию

Рассмотрим бизнес - процессы, необходимые для создания заявки на выделение транспортного средства:

- Создание автопарка – учет всех видов транспортных средств, которые участвуют в организации работ служб учреждения;
- Создание служб – выделение структурных подразделений внутри организации, имеющих доступ к автопарку;
- Назначение ответственных – выбор бригадира внутри подразделения, который несет ответственность за создание заявки и деятельность своей службы;
- Создание/редактирование заявки – заполнение/редактирование заявки ответственным лицом(бригадиром);
- Утверждение/возврат заявки – утверждение/возврат заявки начальником транспортного парка;

Исходя из вышеперечисленного необходимо определить данные, которые будут использоваться в проектируемой базе данных. Эти сведения можно разделить на типы: сведения об автопарке, сведения о службах, сведения об ответственных лицах, перечень заявок.

Сведения об автопарке включают: наименование, номер, ссылку на изображение.

Сведения о службах включают: наименование, дополнительная информация.

Сведения об ответственных лицах: наименование службы, ФИО ответственного, пароль, уровень доступа.

Сведения о заявках: ФИО ответственного лица, название службы, дата выезда, дата возвращения, цель поездки, расстояние до объекта, наименование объекта, состояние заявки.



После проведенного анализа данных необходимо перейти к исследованию информационных потоков.

К входным данным относятся следующие сведения: сведения о службах, бригадирах, транспортном средстве, типах работ и местах их выполнения, сроке выполнения работ, грузе

Выходные данные: отчет «Заявка на транспорт», отчет «Приказ наряд», отчет «Анализ нагрузки на транспортное средство», отчет «Заявки по службам».

## 2.2 Анализ существующих разработок

В качестве вариантов автоматизации существуют готовые типовые решения, также предлагается рассмотреть вариант индивидуальной разработки.

Готовые решения предназначены для полной автоматизации полной автоматизации транспорта предприятия.

Программа «Контроль и учет автотранспорта» [8]. Преимущества программы:

- Простой и удобный интерфейс. Вы работаете с программой, специально адаптированной под учет автотранспорта в России. Это значит, что в ней нет ничего лишнего. Основные таблицы называются АТС (автотранспорт), Тракторы и ДСТ (дорожно-строительная техника), Персонал, Путевые листы. Вся информация в представлена компактно. При необходимости получения более подробных сведений Вы просто открываете интересующую вас запись таблицы;
- Информативность. К записи о транспортном средстве, тракторе, сотруднике вы прикрепляете фотографию. Если у вас большой автопарк, то это будет значительным подспорьем. К карточке транспортного средства можно добавить информацию о неисправностях. К записи о сотруднике можно добавить контактные данные. Все сведения, которые вам понадобятся, у вас всегда под рукой и запрашиваются в течение нескольких секунд;

- Быстрое формирование документов и отчетов. В программу загружены типовые формы журналов, путевых листов и всевозможных транспортных отчетов, начиная с лимитно-заборной карты и заканчивая отчетом по топливу. Если вовремя вносить информацию в программу, то отчет формируется за пару кликов. Формат отчетов — doc или xls. Некоторые поля можно заполнить или исправить вручную в печатной форме. Унифицированные формы отчетов составлены в соответствии с действующим законодательством;

- Важные напоминания. Программа автоматически будет оповещать вас о том, когда нужно пройти очередное техобслуживание, техосмотр или поменять страховку. Теперь вам не нужно выписывать всё это в отдельную тетрадь или постоянно просматривать журналы;

- Постоянная техническая поддержка и регулярные обновления версий. Мы даем быструю обратную связь по всем техническим вопросам, возникающим в связи с работой программы. Ваши конструктивные замечания помогают нам делать программу лучше;

Первым недостатком данной программы является стоимость лицензии для 1 рабочего места, которая равна 4000р. Согласно сайту программы стоимость на 10 рабочих мест равна 14000р. На данный момент на предприятии 48 рабочих мест. Общая стоимость из расчета на 50 рабочих мест равна минимум 50000р с учетом скидки. Лицензия выдается на 1 год. Дальнейшее обновление предполагает повторную покупку лицензии на каждое рабочее место.

Вторым недостатком является установка сервера БД, который отличается от версий серверов, разрешенных на предприятии.

Третьим недостатком является отсутствие возможности доработки программного обеспечения.

Программа «Автопарк» [7] позволяет систематизировать информацию по заправкам, расходам, поездкам, приобретенным запчастям и материалам, сервисному обслуживанию и ремонту. Есть возможность планирования сервисного обслуживания и оповещения о предстоящих событиях.

Основные возможности программы:

- Справочники Транспорта, Сотрудников, Запчастей и материалов;
- Учет расхода топлива;
- Учет документов (единый журнал документов);
- Учет сервисного обслуживания и ремонтов;
- Учет запчастей и материалов;
- Учет поездок;
- Планирование сервисного обслуживания;
- Напоминания о запланированных событиях;
- Возможность формирования отчетов;
- Многопользовательский режим работы, разделение прав доступа.

Для кого подходит программа Автопарк:

- Для компаний, имеющих автопарк желающих вести простой и удобный учет автомобилей и других транспортных средств и агрегатов.

Главным недостатком является невозможность доработки программного комплекса по требованию пользователя и стоимость лицензии, которая ограничивается количеством транспортных средств, стоящих на учете и количеством рабочих мест с которых будет производиться работа.

Стоимость лицензии без ограничений на количество транспорта составляет 6000 рублей. Однако программа является однопользовательской и не представляется возможность использования по локальной сети, что делает ее установку на предприятии не доступной.

Сравнительные показатели обоих программных продуктов представлены в таблице 2.2.1.

Таблица 2.2.1 – Сравнительные показатели программных продуктов

Критерий	Контроль и учет	Автопарк
Стоимость 1 лицензии	4000 р + Каждая транспортная ед.- отдельная стоимость	6000 р. Без ограничений по количеству транспорта на 1 год.

### Продолжение таблицы 2.2.1

Техническое сопровождение	да	Да
Возможность доработки ПО	нет	Да
Возможность работы по локальной сети	да	Нет
Разграничение доступа	Да	Да
Формирование отчетности	да	да

### 2.3 Предполагаемая технология выполнения исследуемого процесса

Для описания, бизнес-процессов, которые происходят в исследуемом процессе без использования информационной системы используем ERwin Process Modeler - средство моделирования бизнес-процессов. Посредством набора графических инструментов построили схему процесса, на которой показаны исходные данные, результаты операций, ресурсы, необходимые для их выполнения, управляющие воздействия, взаимные связи между отдельными работами.

Диаграмма потока данных корректно отображает процессы протекающие при создании заявки, но она не способна отобразить одну из главных проблем – время каждого отдельного этапа. Для того чтоб отобразить дополнительные параметры этапов используем карту потока ценностей текущего состояния [9].

Карта потока ценностей - наглядное отображение/документирование процесса, с целью анализа движения информации, документации и наглядного представления задержек в анализируемых потоках, т.е. факторов, увеличивающих ВПП и провоцирующих незавершенное производство. Время протекания процесса–время продвижения заявки от момента возникновения потребности в до момента её завершения.

На рисунке 2.3.1 представлена диаграмма потоков данных [1]. DFD – это нотация, предназначенная для моделирования информационных систем с точки зрения хранения, обработки и передачи данных.

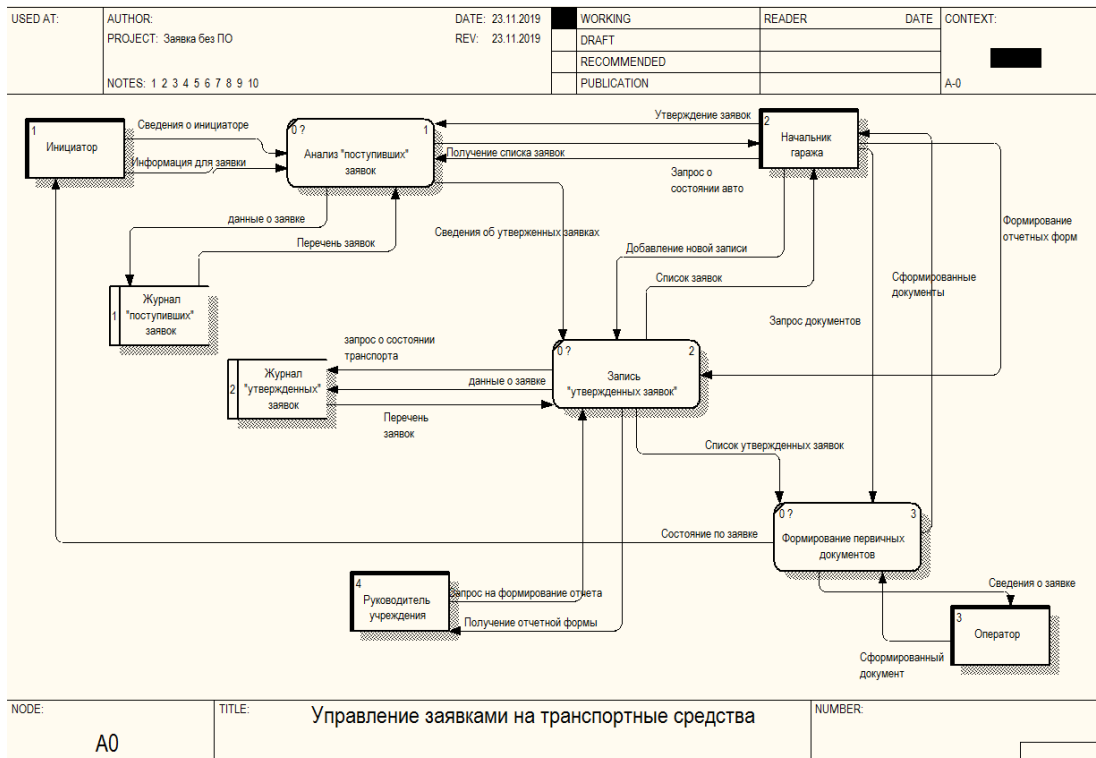


Рисунок 2.3.1 – Диаграмма потоков данных

На рисунке 2.3.2 представлена карта потока ценностей без использования информационной системы. На данной карте отображены основные этапы, время каждого этапа, пользователи, а также сводная информация. На данной карте нет пользователя «Руководитель», так как ответы на запросы руководства занимают очень длительное время.

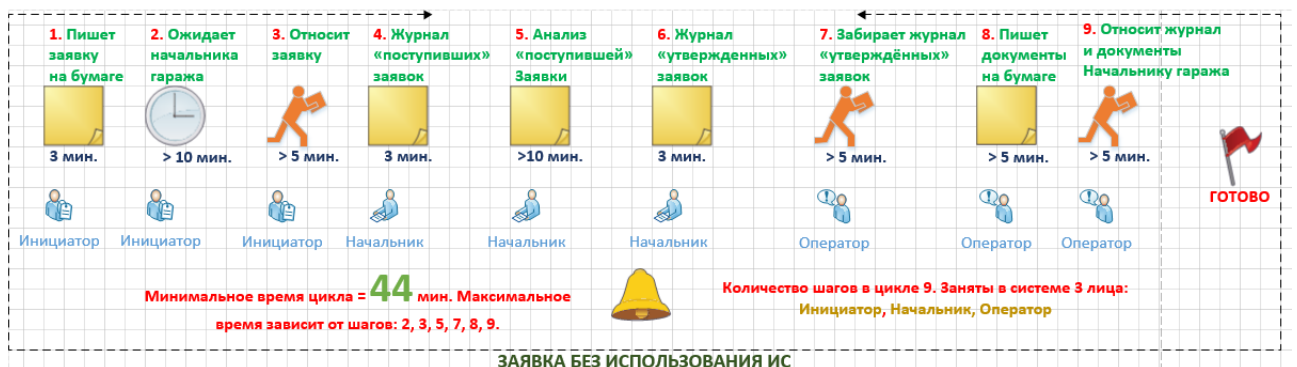


Рис 2.3.2 Карта потока ценностей без ИС

Цель проектирования и информатизации исследуемого бизнес-процесса сводится к достижению следующих показателей:

- Полная автоматизация процесса подачи заявки и ее утверждения;
- Создание единой базы данных для хранения информации о заявках;
- Возможность получение статистической информации по различным критериям;
- Сокращение временных затрат на работу с заявками как от инициатора, так и от начальника транспортного парка.

Для отображения итогового результата используем карту потока ценностей, представленную на рисунке 2.3.3. На данной карте видно, что при наличии автоматизации процесса нет необходимости использовать оператора для ввода информации, также сокращено количество этапов и времени.

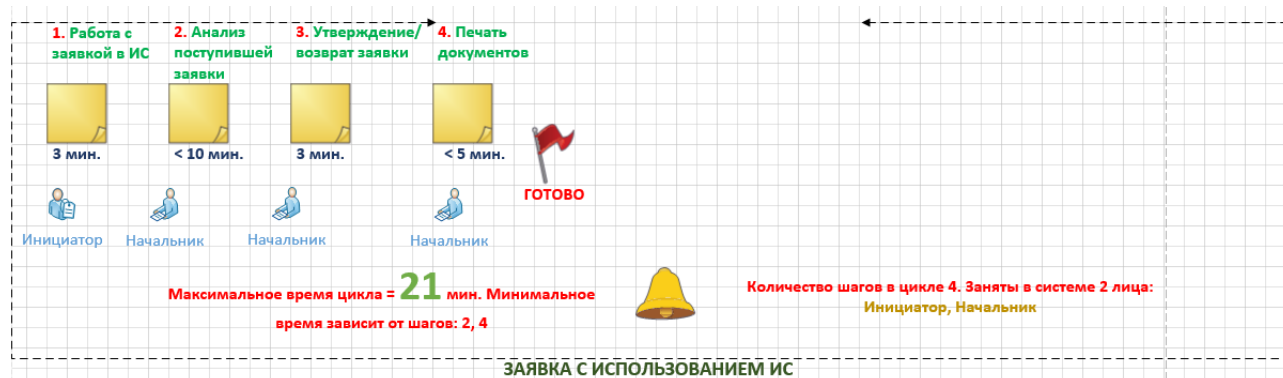


Рисунок 2.3.3 - Карта потока ценностей с использованием ИС

Методология IDEF0 [13] предписывает построение иерархической системы диаграмм - единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция - система разбивается на подсистемы, и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности. На рисунке 2.3.4 представлена диаграмма второго уровня для разрабатываемой системы.

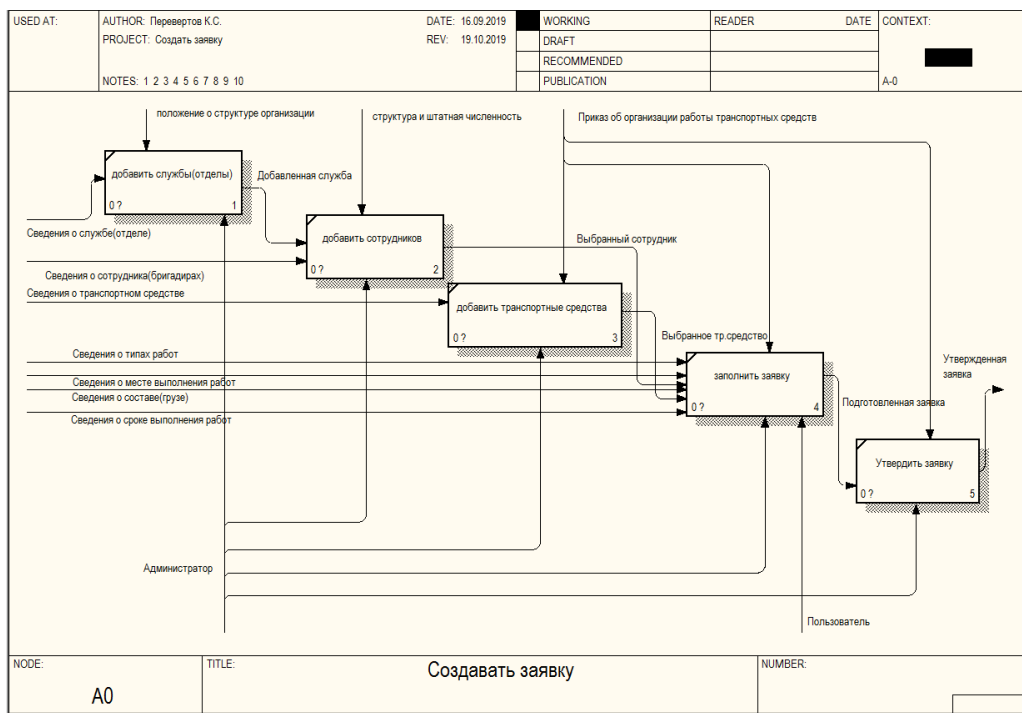


Рисунок 2.3.4 – Диаграмма IDEF0 второго уровня

Преимущества разрабатываемой системы:

- Создание заявки на использование транспортного средства в режиме реального времени, не прерывая текущей работы;
- Возможность выбора транспортного средства по индивидуальным критериям;
- Графическое отображение транспортных средств для наглядности;
- Инициаторы в режиме реального времени получают информацию о состоянии заявки;
- Начальник гаража получает статистическую информацию в режиме реально времени;
- Отсутствие необходимости ведения журнала на бумажном носителе;
- Удобное цветовое отображение состояния заявок;
- Возможность использования информационной системы в локальной сети предприятия;
- Вся информация хранится в единой базе данных;

- Присутствие механизмов резервного копирования данных;
- Разграничение уровня доступа к данным;
- Авторизация пользователя по логину и паролю;
- Возможность доработки программного обеспечения по требованию пользователя;
- База данных разворачивается на существующем сервере;
- Доступ к базе данных дополнительно ограничен на уровне СУБД;

## 2.4 Обоснования выбора средств разработки

В качестве среды разработки информационной системы была выбрана RAD Studio 10.3 Rio [4]. Преимуществами разработки являются следующие:

- Используется программное обеспечение, разрешенное службой безопасности организации;
- Полное соответствие программного продукта требованиям заказчика;
- Отсутствие ненужных (“лишних”) модулей;
- Возможность быстрой доработки;
- Возможность регулярных консультаций (при необходимости таковых, стоит отметить, что в первый год функционирования программного решения такая необходимость возникает достаточно часто);
- Приемлемая стоимость программного решения;

Разрабатываемая информационная система будет являться приложением клиент серверного типа [5]. Информационная система будет взаимодействовать с серверной базой данных, и состоять из двух частей.

Первая часть приложения, реализующая интерфейс пользователя и находящаяся на клиентской рабочей станции. А вторая часть приложения будет отвечать за хранение, и обработка данных осуществляется на стороне сервера.

Для реализации взаимодействия клиента и сервера необходимо



реализовать функции загрузки и отображения данных из базы данных, и пересылка данных в базу данных с последующим сохранением данных в базе.

Для реализации пользовательского интерфейса и подключения к базе данных выбрана среда разработки RAD Studio.

RAD Studio 10.3 Rio - это комплексное решение по разработке программного обеспечения, предназначенное для быстрого проектирования, программирования и расширения так называемых взаимодействующих приложений для Windows, Mac, iOS, Android и IoT. Возможность создавать готовые к работе решения бизнес-класса, включающие разнообразные клиентские native-платформы, мобильные расширения, интеллектуальные устройства, облачные сервисы, корпоративные и встроенные базы данных.

В качестве основной технологии для подключения к базе данных будет использоваться технология ADO [11]. Необходимые компоненты установлены в среде разработки по умолчанию.

ActiveX Data Objects - объекты данных, построенные как объекты ActiveX. ADO усиленно развивается корпорацией Microsoft. На основе этой технологии созданы соответствующие компоненты-наборы TADOTable, TADOQuery, TADOStoredProc, повторяющие в функциональном отношении компоненты TTable, TQuery, TStoredProc, но не требующие развертывания и настройки BDE на клиентской машине. На рисунке 2.4.1 изображена реализация технологии ADO

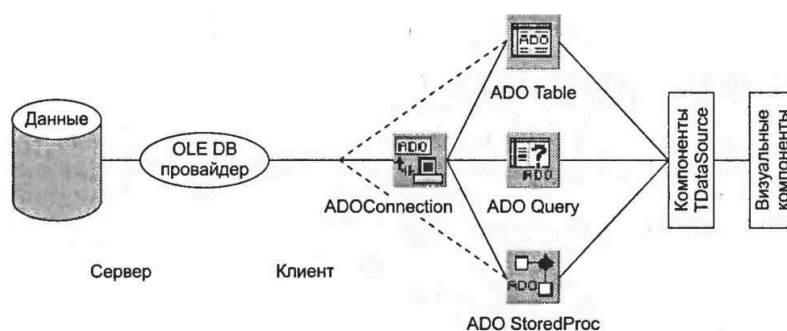


Рисунок 2.4.1- Реализация технологии ADO

В качестве СУБД была использована Microsoft SQL Server [12] — система

управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основным используемым языком запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями.

В качестве среды разработки была выбрана dbForge Studio for SQL Server [14] - это полнофункциональная среда разработки для управления, администрирования, представления и анализа данных SQL Server. Инструмент будет полезен для разработчиков и администраторов баз данных SQL Server при выполнении сложных задач, таких как проектирование баз данных, написание SQL кода, сравнение баз данных, синхронизация схем и данных, генерации реальных тестовых данных и т. п.

СУБД выбранная для разработки уже установлена и настроена в организации.

### 3 Разработка частного решения

#### 3.1 Описание технологии разработки интерфейсов приложения

Реализация программного обеспечения – это процесс перевода системной спецификации в работоспособную систему. Разработка приложений подсистемы осуществлялась на языке Delphi с использованием технологии ADO.

Итогом реализации приложения является работоспособная информационная система. Разрабатываемая информационная система будет являться приложением клиент серверного типа. Информационная система будет взаимодействовать с серверной базой данных, и состоять из двух частей.

Для реализации взаимодействия клиента и сервера необходимо реализовать функции загрузки и отображения данных из базы данных, и пересылка данных в базу данных с последующим сохранением данных в базе. Для работы с данными, а также с базами данных, используются компоненты [15], расположенные на вкладке «dbGo» (рисунок 3.1.1)

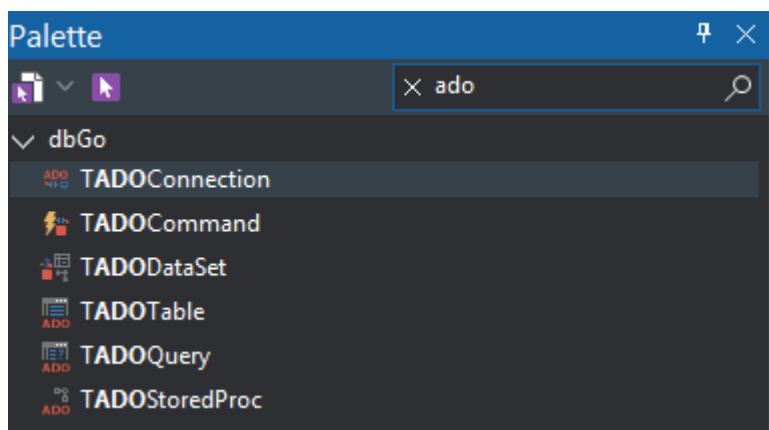


Рисунок 3.1.1 - Компоненты ADO

Первый компонент называется ADOConnection. С помощью компонента ADOConnection можно указывать местоположение базы данных и работать с транзакциями.

Третий компонент закладки ADO носит имя ADODataSet. Его назначение — получение набора данных из одной или нескольких таблиц базы данных.

Кроме того, он также позволяет работать с возвращенным набором данных визуальным компонентам, предназначенным для отображения данных. Это наиболее общий компонент для работы с набором данных, который может использоваться вместо компонентов ADOTable, ADOQuery или ADOStoredProc.

Компонент ADOQuery представляет собой запрос к базе данных. Это может быть как запрос, в результате которого возвращаются данные из базы (например, SELECT), так и запрос, не формирующий результирующего набора данных (например, INSERT).

Еще одним компонентом для взаимодействия является компонент DataSource. Данный компонент находится на другой вкладке и представляет собой источник данных, который обеспечивает связь между набором данных и компонентами отображения и редактирования данных. При использовании данных компонентов необходимо подключить модули Data.DB, Data.Win.ADODB в разделе uses.

Ключевое слово Uses определяет список одного или более модулей, которые используются текущим модулем, программой или библиотекой. Каждый модуль эффективно импортирует - все public или published подпрограммы и данные.

Например, включение SysUtils делает доступными множество подпрограмм манипулирования данными, таких как IntToStr. Порядок модулей в списке важен. Те, что объявлены позже в этом списке имеют приоритет над объявленными ранее. В редкой ситуации, где та же самая подпрограмма или данные определены в двух или более модулях, вы можете принудить Delphi использовать желаемый модуль приписыванием имени модуля к имени подпрограммы или данных.

Для модулей, Uses может находиться в разделах Interface и Implementation. Если модули используются только в разделе implementation, то их следует объявлять именно там, это поможет избежать излишних перекомпиляций.

Для удобства подключения приложения к источнику данных было

принято решение хранить строку подключения в отдельном файле. Данные из файла автоматически считываются и передаются свойству `ConnectionString` компонента `AdoConnection` при использовании функции «проверка соединения». Далее происходит считывание имени сервера базы данных и проверка его на соответствие. В случае удачной проверки происходит включение компонентов запроса `AdoQuery`. Если проверка не прошла успешно появляется окно с описанием ошибок. Фрагмент кода реализующий подключение представлен на рисунке 3.1.2

```
Memol.Lines.LoadFromFile('conf.txt');
//присваиваем строку подключения
AdoConnection1.ConnectionString:=Memol.Text;
//получаем имя сервера БД
Edit3.Text:=AdoConnection1.Properties.Item['Data Source'].Value;
//проверяем соответствие
if (edit3.Text='ASU3\SQLEXPRESS') or (Edit3.Text='kulpu-wsus-s01') or (Edit3.Text='mynout')
begin
//удачное соединение. включаем компоненты запроса adoquery
try
AdoConnection1.LoginPrompt:=false;
AdoConnection1.Connected:=true;
ShowMessage('Соединение успешно');
AdoQuery_v_lic.Active:=true;
AdoQuery_lic.Active:=true;
AdoQuery_lic_spr.Active:=true;
AdoQuery_park.Active:=true;
AdoQuery_ADD_z.Active:=true;
AdoQuery_f3.Active:=true;
AdoQuery_del_z1.Active:=true;
AdoQuery_flexp.Active:=true;
AdoQuery_park_r.Active:=true;
AdoQuery_otdel.Active:=true;
AdoQuery_otdel_r.Active:=true;
AdoQuery_spr_otdel.Active:=true;
AdoQuery_profil.Active:=true;
DBLookupComboBox1.Enabled:=true;
AdoQuery_lic_spr_r.Active:=true;
AdoQuery_glob.Active:=true;

Edit1.Enabled:=true;

BitBtn2.Visible:=false;
//неудачное соединение. Вывод ошибки в сообщении
except
on E:Exception do begin
DBLookupComboBox1.Enabled:=false;
Edit1.Enabled:=false;

ShowMessage('Ошибка: '+e.message+#13#10+'идент');
end;
```

Рисунок 3.1.2 - Подключение к источнику данных

Для выполнения различных запросов необходимо передавать параметры запроса компоненту `AdoQuery`. Существует несколько способов передачи параметров. Первый – хранение параметров запроса в свойстве `sql` данного компонента. Такой способ удобен при выполнении статического запроса, параметры которого не изменятся. Например: `select * from otdel`. Если же запрос использует значения каких-либо переменных, то необходимо передавать эти значения непосредственно перед выполнением запроса. На рисунке 3.1.3 представлен запрос с передачей дополнительных параметров.

```

procedure TForm_GL.BitBtn4Click(Sender: TObject);
var
td:Tdatetime;
begin
td:=now;
BitBtn4.Visible:=false;
BitBtn7.Visible:=false;
Form_login.ADOQuery_ADD_z.Close;
//очищаем свойство sql компонента запроса
Form_login.ADOQuery_ADD_z.SQL.Clear;
//вызываем хранимую процедуру, перечисляем доп. параметры запроса. : параметр
Form_login.ADOQuery_ADD_z.SQL.Add('update zi_full :dtz, :idp, :d1, :c1,:pr, :t2, :d2, :cel, :ml, :km, :sti, :IDI');
//присваиваем параметрам значение компонентов
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('dtz').Value:=StrToDateTime(Form_GL.StatusBar1.Panels[7].Text);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('idp').Value:=DBLookupComboBox1.KeyValue;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('d1').Value:=Edit2.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('c1').Value:=MaskEdit1.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('pr').Value:=Edit7.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('t2').Value:=MaskEdit2.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('d2').Value:=StrToDate(Edit1.Text);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('cel').Value:=Edit8.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('ml').Value:=Edit3.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('km').Value:=StrToInt(Edit10.Text);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('sti').Value:='B pa6ore';
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('IDI').Value:=StrToInt(id_zi_del.Text);
//выполняем запрос
Form_login.ADOQuery_ADD_z.ExecSQL;

```

Рисунок 3.1.3 - Запрос с передачей дополнительных параметров

При необходимости вывода результата запроса используется конструкция присвоения. Для этого сначала выполняется запрос, затем результат запроса присваивается свойству компонента. Пример конструкции по присвоению свойству Caption компонента GroupBox результата запроса имеет вид:

```

Form_login.ADOQuery_sum.Close; Form_login.ADOQuery_sum.SQL.Clear;
//вызываем хранимую процедуру, перечисляем доп. параметры запроса. :
параметр

```

```

Form_login.ADOQuery_sum.SQL.Add('sum :d1');

```

```

//присваиваем параметрам значение компонентов

```

```

Form_login.ADOQuery_sum.Parameters.ParamByName('d1').Value:=DateTo
Str(td); Form_login.ADOQuery_sum.Open; GroupBox7.Caption:='Сводная
информация';// результат запроса передаем свойству

```

```

GroupBox7.Caption:=GroupBox7.Caption+'
'+IntToStr(Form_login.ADOQuery_full.RecordCount);

```

Результат выполнения запроса представлен на рисунке 3.1.4.

Дата выезда	Количество заявок
2018-06-01	47
2018-06-04	45
2018-06-05	47

Рисунок 3.1.4 - Присвоение результата запроса компоненту

При необходимости вывода информации в табличном виде из базы данных используется компонент DBGrid. Для того чтобы связать данный компонент с компонентом запроса ADOQuery необходимо подключить дополнительный компонент DataSource. Компонент DataSource подключается к ADOQuery через свойство DataSet. В свою очередь компонент DBGrid подключается к DataSource через одноименное свойство.

На рисунке 3.1.5 представлен фрагмент компонентов ADO подключенных к источнику данных. На фрагменте видно, что к одному из компонентов запроса ADOQuery подключен компонент DataSource, так как результат запроса данного компонента необходимо вывести в компонент DBGrid.

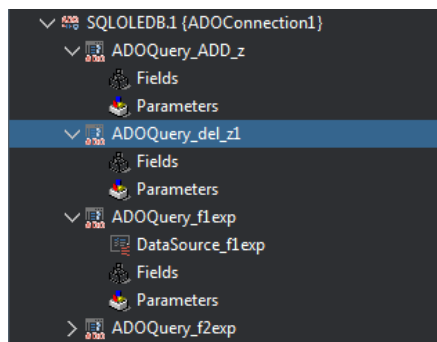


Рисунок 3.1.5 – Подключение к источнику данных

Полный листинг приложения представлен в Приложении А .

FrxFReport - Компонент отвечает за навигацию по записям и обращение к полям. Это позволяет не привязывать ядро FastReport [16] к какой-либо библиотеке доступа к данным. Именно этот компонент позволяет построить визуальную составляющую отчета.

Для каждого отчета в проекте создается отдельная форма, на которой в обязательном порядке присутствует FrxFReport, frxDBDataset- аналог компонента DataSource. А также компоненты для выгрузки отчета в определенном формате, например, компонент frxPDFExport, позволяющий выгрузить отчет в формате .pdf

FrxFReport представляет из себя лист отчета, который состоит из 3 основных блоков: ReportTitle, MasterData, PageFooter.

Для подключения к источнику используем меню-отчет-данные и выбираем источники данных которые подключены к компонентам запроса ADOQuery.

С помощью текстового компонента Мемо создаем заголовки столбцов таблицы и информативные поля, такие как название отчета

Поле [frxDBDataset1.""] – используя данные виды полей получаем информацию из запроса. В кавычках вводится имя поля из запроса.

Текстовые поля, заключенные в квадратные кавычки, позволяют использовать системные переменные и переменные значение которые необходимо вывести на форму. Например: [Line]-системная переменная, позволяющая получить порядковый номер строки в таблице, а [da]- переменная в которую передается Дата создания заявки, взятая как критерий построения отчета. Общий вид отчета представлен на рисунке 3.1.6.

The screenshot shows a report design grid with the following elements:

- ReportTitle:** ReportTitle1
- Text:**
  - УТВЕРЖДАЮ
  - Начальник управления
  - Нав\*\*\* Р. В.
  - ЗАЯВКА
  - На транспорт на [da]
- Header:** Header1
 

№ п/п	Дата, время выезда	Маршрут движения	Состав бригады (груз)	Дата, время возвращения	Пробег	Старший бригады	Марка автомобиля	Цель
[Line]	[frxDBDataset1."Дата и	[frxDBDataset1."Маршрут	[frxDBDataset1."Co	[frxDBDataset1."Дат	[frxDBDataset1."Пробег	[frxDBDataset1."Старший	[frxDBDataset1."Марка	[frxDBDataset1."Цель"]
- PageFooter:** PageFooter1
 

[Page#]

Рисунок 3.1.6 – Проектирование отчета

Для построения отчета в общем случае необходимо выполнить запрос на выборку, передать значение переменным, сохранить отчет под определенным именем, открыть отчет. В проекте для данного построения отчета используется конструкция:

```
Form_login.ADOQuery_f2exp.SQL.Text:=Form_login.ADOQuery_f2exp.SQL.Text + 'AND d1=:d1 and status=:st';
```

```
Form_login.ADOQuery_f2exp.Parameters.ParamByName('d1').Value:=StrToDate(edit1.Text);Form_login.ADOQuery_f2exp.Parameters.ParamByName('st').Value:='Сформирована';Form_login.ADOQuery_f2exp.Open;Form_login.ADOQuery_FULL.Ac
```



```

tive:=false;Form_login.ADOQuery_FULL.Active:=true;ShowMessage('Заявка
сформирована. Таблица обновлена');
//присваиваем отчету имя используя дополнительно дату
Form_f2_e.frxPDFExport1.FileName:='Ф1_Заявка на транспорт'+ ' '+Edit1.text;
//присваиваем переменной дату построения отчета
Form_f2_e.frxReport1.Variables['da']:='"' + Edit1.Text + '"';
//показываем отчет
Form_f2_e.frxReport1.ShowReport;

```

При разработке визуальной части интерфейса ИС используется ряд стандартных визуальных компонентов [17]. Сюда можно отнести компоненты: кнопка (button, radiobutton, speedbutton), меню (mainmenu, popupmenu), таблица (dbgrid), текстовые поля для ввода информации (edit, labelededit), календарь(calendar), выпадающие списки (dblookupcombobox, combobox), панель группировки информации (panel, groupbox), компонент просмотра изображения (image), надписи(label). Вспомогательными не визуальными компонентами, которые дополняют работу визуальных компонентов, являются компоненты: таймер (timer), диалоговое окно выбора цвета (colordialog), диалоговое окно выбора каталога или файла (opendialog).

Общий вид визуальных компонентов представлен на рисунке 3.1.7

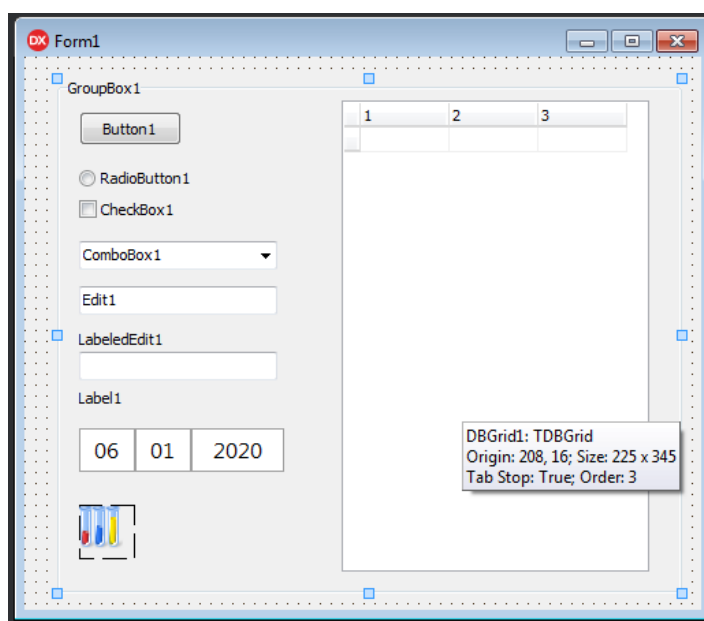


Рисунок 3.1.7 – Внешний вид визуальных компонентов

При реализации функциональной части ИС используются операции присваивания, например: DBLookupComboBox1.Enabled:=false.

Также используются операции по преобразованию типа данных, например приведение строчного типа к числовому: a:=StrToInt(Edit1.text).

Различного вида алгоритмы используются в ИС для получения определенного результата, например, проверка доступности БД выглядит следующим образом:

```
except
on E:Exception do begin
DBLookupComboBox1.Enabled:=false;
Edit1.Enabled:=false;
ShowMessage('Ошибка:'+e.message+#13#10+'идент');
end;end; end else
ShowMessage('Сервер не зарегистрирован в приложении');
```

Простейшим алгоритмом является также конструкция: if then else.

Примером цикла в ИС является конструкция добавления строки, пока не выполнится условие:

```
var
Data: TDate;
i:integer;
s:string;
begin
Memo1.Lines.Add(edit2.Text);
while Edit13.text<>edit1.text do
begin
edit1.text:=DateToStr(DateTimePicker1.Date);
data:=StrToDate(Edit13.text)+1;
Memo1.Lines.Add(datetostr(data));
Edit13.Text:=DateToStr(data);
end;
```

```
if Memo1.Lines.Count > 0 then
begin
for i:=0 to Memo1.Lines.Count-1 do
begin
s:=Memo1.Lines.Strings[i];
end;
memo1.Clear
end;
```

### 3.2 Разработка базы данных. Создание таблиц, установка связей

Для графического представления проектируемой базы данных используем ER-модель [18]. ER-модель (от англ. entity-relationship model, модель «сущность — связь») — модель данных, позволяющая описывать концептуальные схемы предметной области.

ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.).

ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма «сущность-связь» (англ. entity-relationship diagram, ERD, ER-диаграмма). В приложении В представлена диаграмма «сущность-связь» проектируемой базы данных.

Спроектированная логическая модель данных имеет следующие сущности, представленные в таблице 3.2.1:

Таблица 3.2.1- Сущности проектируемой БД

«Заявки»			
Имя столбца	Тип поля	Описание	Ограничение
id	int	Код заявки	Первичный ключ
dt_z	Varchar(30)	Дата и время создания заявки	
Id_ot	int	Код службы	
Id_lic	int	Код сотрудника	Внешний ключ на таблицу «Сотрудники»
Id_p	int	Код транспорта	Внешний ключ на таблицу «Транспорт»
d1	date	Дата выезда	
t1	Varchar(30)	Время выезда	
Pr	Varchar(100)	Состав бригады (груз)	
status	Varchar(30)	Состояние заявки	
t2	date	Дата прибытия	
t2	Varchar(30)	Время прибытия	
cel	Varchar(100)	Цель поездки	
m1	Varchar(100)	Маршрут	
km	int	Предполагаемый пробег	
«Транспорт»			
id	int	Код транспорта	Первичный ключ
marka	Varchar(40)	Описание марки	
gos_n	Varchar(40)	Номер транспорта	
pr	Varchar(30)	Признак доступности	
img	Varchar(100)	Ссылка на изображение	
«Службы»			
id	int	Код службы	Первичный ключ
name_otd	Varchar(40)	Наименование службы	
tel	Varchar(50)	Номер телефона(доп инф)	
«Профиль доступа»			
id	int	Код профиля	Первичный ключ
name_p	Varchar(50)	Наименование профиля	
«Сотрудники»			
id	int	Код сотрудника	Первичный ключ
Id_otd	int	Код службы	Внешний ключ на таблицу «Службы»
Fio	Varchar(50)	ФИО сотрудника	
spec	int	Код профиля	Внешний ключ на таблицу «Профиль доступа»
pas	Varchar(20)	пароль для Авторизации	
«Резервные копии»			
id	int	Код копии	Первичный ключ
d1	date	Дата копии	
Id_lic	int	Код сотрудника	Внешний ключ на таблицу «Сотрудники»
status	Varchar(50)	Статус выполнения	
P1	Varchar(50)	Примечание	

В качестве СУБД была использована Microsoft SQL Server — система управления реляционными базами данных (СУБД), разработанная корпорацией

Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями.

В качестве среды разработки была выбрана dbForge Studio for SQL Server - это полнофункциональная среда разработки для управления, администрирования, представления и анализа данных SQL Server. Используя среду разработки создаем новое соединение, указав имя сервера, выбрав тип аутентификации.

После успешного соединения нажимаем ПКМ и выбираем «Новая База данных». Задаем имя «transport», заполняем описание и сохраняем. Созданная БД будет представлять из себя дерево объектов, представленное на рисунке 3.2.1.

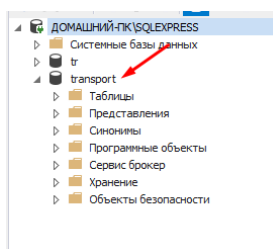


Рисунок 3.2.1 - Дерево объектов

Используя дерево объектов выбираем «Таблицы» и нажимаем ПКМ-Новая таблица. Для каждой из созданных таблиц указывается название таблицы, наименование полей и типы полей. При необходимости при выборе типа поля в круглых скобках указывается его размерность. В проекте структуре БД построено 5 таблиц. На рисунке 3.2.2 представлен пример создания таблицы z1.

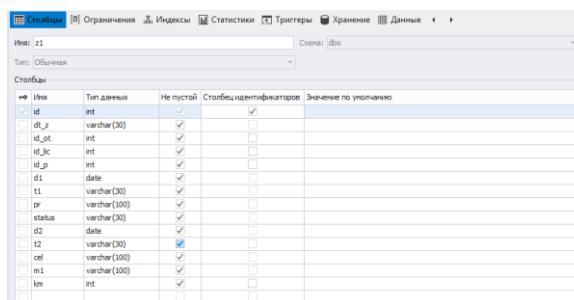


Рисунок 3.2.2 - Проект Таблицы z1

Для создания связи между таблицами используется блок «Ограничения» выбранной таблицы. В данном блоке создаются внешние ключи для связанных таблиц. Для создания ключа необходимо выбрать поле в базовой таблице и поле в таблице, на которую ссылается ключ. На рисунке 3.2.3 представлен внешний ключ «Номер авто» которой ссылается на таблицу «Парк авто»

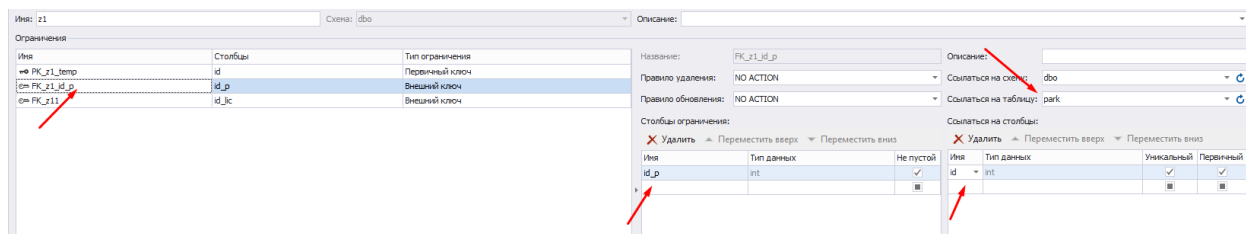


Рисунок 3.2.3 - Создание внешнего ключа

Общий вид топологии, проектируемой БД представлен на рисунке 3.2.4

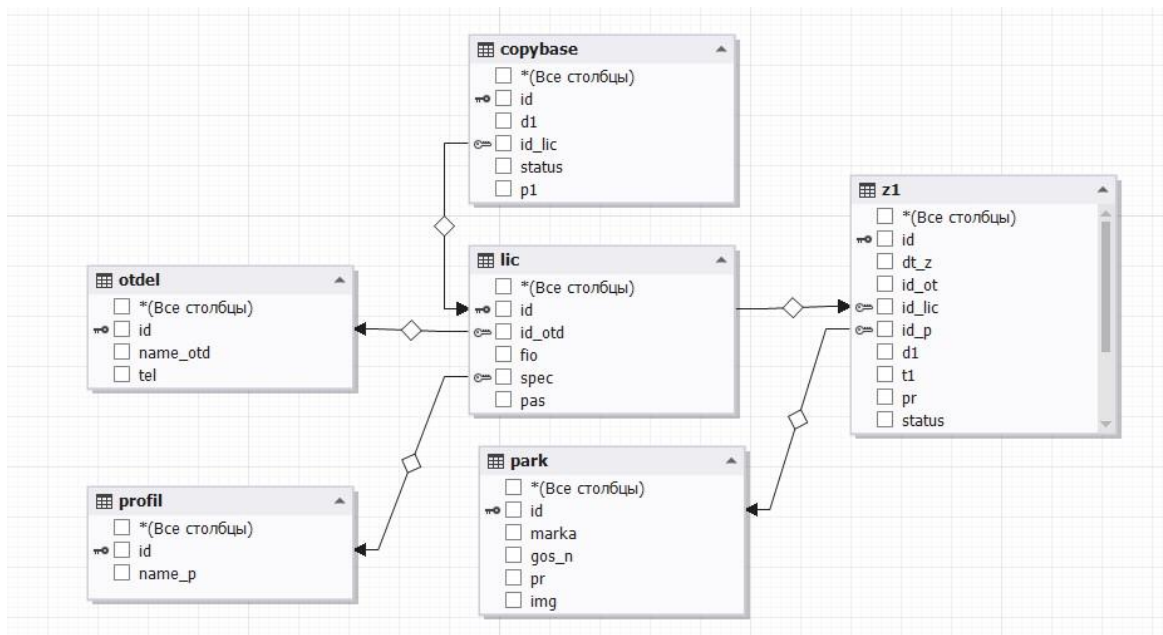


Рисунок 3.2.4 – Топология проектируемой БД

### 3.3 Разработка базы данных. Хранимые процедуры

Нередко операция с данными представляет набор инструкций, которые необходимо выполнить в определенной последовательности. Например, если требуется удалить подразделение из базы данных, то необходимо провести проверку на наличие работающих сотрудников в подразделении. Возможно,

при этом понадобится проверить еще ряд дополнительных условий. То есть удаление подразделения охватывает несколько действий, которые должны выполняться в определенной последовательности. И в этом случае более оптимально будет инкапсулировать все эти действия в один объект - хранимую процедуру (stored procedure) [6].

То есть по сути хранимые процедуры представляет набор инструкций, которые выполняются как единое целое. Тем самым хранимые процедуры позволяют упростить комплексные операции и вынести их в единый объект. Изменится процесс удаления подразделения, соответственно достаточно будет изменить код процедуры. То есть процедура также упрощает управление кодом.

Также хранимые процедуры позволяют ограничить доступ к данным в таблицах и тем самым уменьшить вероятность преднамеренных или неосознанных нежелательных действий в отношении этих данных.

И еще один важный аспект - производительность. Хранимые процедуры обычно выполняются быстрее, чем обычные SQL-инструкции. Все потому что код процедур компилируется один раз при первом ее запуске, а затем сохраняется в скомпилированной форме.

Для создания хранимой процедуры применяется команда CREATE PROCEDURE или CREATE PROC.

Таким образом, хранимая процедура имеет три ключевых особенности: упрощение кода, безопасность и производительность.

Хранимые процедуры могут возвращать множества результатов, то есть результаты запроса SELECT. Такие множества результатов могут обрабатываться, используя курсоры, другими сохранёнными процедурами, возвращая указатель результирующего множества, либо же приложениями. Хранимые процедуры могут также содержать объявленные переменные для обработки данных и курсоров, которые позволяют организовать цикл по нескольким строкам в таблице. Стандарт SQL предоставляет для работы выражения IF, LOOP, REPEAT, CASE и многие другие. Хранимые процедуры

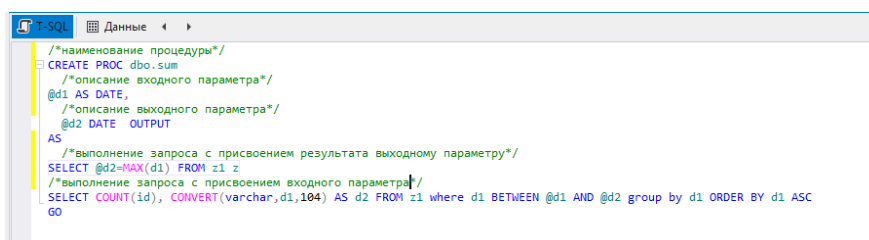
могут принимать переменные, возвращать результаты или изменять переменные и возвращать их, в зависимости от того, где переменная объявлена.

Вместо хранения часто используемого запроса, клиенты могут ссылаться на соответствующую хранимую процедуру. При вызове хранимой процедуры её содержимое сразу же обрабатывается сервером.

Кроме собственно выполнения запроса, хранимые процедуры позволяют также производить вычисления и манипуляцию данными — изменение, удаление, выполнять DDL-операторы (не во всех СУБД) и вызывать другие хранимые процедуры, выполнять сложную транзакционную логику. Единственный оператор позволяет вызвать сложный сценарий, который содержится в хранимой процедуре, что позволяет избежать пересылки через сеть сотен команд и, в особенности, необходимости передачи больших объёмов данных с клиента на сервер.

Для создания хранимой процедуры необходимо в дереве объектов выбрать «Программные объекты-Хранимые процедуры».

Структура хранимой процедуры следующая: наименование процедуры, запрос. Нередко используются входные и выходные параметры, которые имеют собственное наименование, тип поля, тип параметра. Блок с параметрами находится перед блоком с запросом. На рисунке 3.3.1 представлена структура хранимой процедуры по выбору данных из таблицы заявок. В качестве входного параметра будет использоваться произвольная дата выезда, а в качестве выходного максимальная дата выезда из таблицы.



```
/*наименование процедуры*/
CREATE PROC dbo.sum
/*описание входного параметра*/
@d1 AS DATE,
/*описание выходного параметра*/
@d2 DATE OUTPUT
AS
/*выполнение запроса с присвоением результата выходному параметру*/
SELECT @d2=MAX(d1) FROM z1 z
/*выполнение запроса с присвоением входного параметра*/
SELECT COUNT(id), CONVERT(varchar,104) AS d2 FROM z1 where d1 BETWEEN @d1 AND @d2 group by d1 ORDER BY d1 ASC
GO
```

Рисунок 3.3.1 - Структура хранимой процедуры

Полное описание всех хранимых процедур представлено в Приложении Б



### 3.4 Алгоритм работы информационной системы

Работа в АИС условно делиться на 10 основных этапов:

- - Этап № 1. Проверка доступности сервера БД, авторизация пользователя;
- - Этап № 2. Наполнение, редактирование справочника «Службы»;
- - Этап № 3. Наполнение, редактирование справочника «Сотрудники»;
- - Этап № 4. Наполнение, редактирование справочника «Транспорт»;
- - Этап № 5. Настройка палитры цветов для отображения статуса заявок;
- - Этап № 6. Создание, редактирование, удаление заявки на транспортное средство;
  - - Этап № 7. Поиск заявки;
  - - Этап № 8. Утверждение заявки;
  - - Этап № 9. Формирование отчетных форм;

В АИС существует 2 профиля прав: пользователь, администратор. Доступные этапы для «Пользователь»: № 1, 5, 6,7. Доступные этапы для «Администратор»: № 1-9. Профиль прав «Пользователь» позволяет управлять только собственными заявками, профиль «Администратор» позволяет управлять всеми заявками АИС.

Рассмотрим каждый из представленных этапов отдельно по порядку.

Этап № 1. Проверка доступности сервера БД, авторизация пользователя. При каждом запуске АИС появляется окно авторизации и проверки доступности сервера БД. В окне расположена кнопка проверки доступности сервера «Проверить соединения». Нажав данную кнопку, пользователь запускает функцию, которая проверяет доступность сервера БД. На данном этапе невозможно выбрать пользователя системы и ввести пароль.

В случае недоступности сервера при проверке соединения АИС

предупреждает пользователя о неправильной конфигурации подключения.

Пример проверки подключения при незарегистрированном сервере представлен на рис. 3.4.1

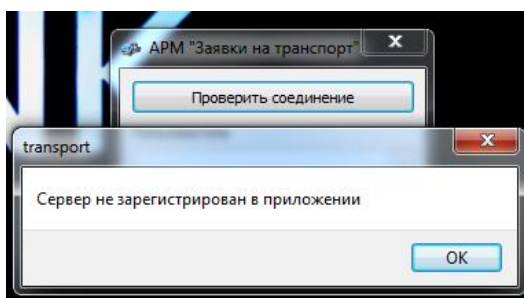


Рисунок 3.4.1 – Проверка подлинности сервера

В случае успешной проверки необходимо из выпадающего списка выбрать пользователя системы, а также ввести пароль (рис. 3.4.2).

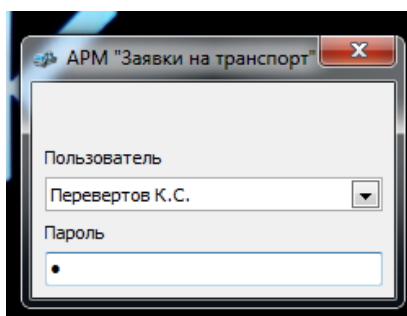


Рисунок 3.4.2 - Окно авторизации после успешного соединения

Введённый пароль проверяется АИС на корректность. В случае ввода некорректного пароля система выведет соответствующее сообщение. Сообщение представлено на рис. 3.4.3

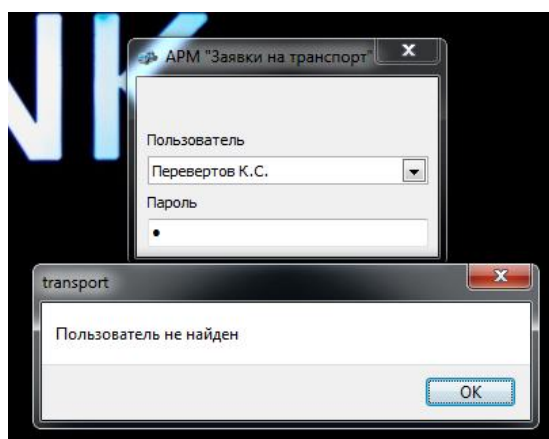


Рисунок 3.4.3 - Проверка корректности ввода пароля

В случае успешной авторизации пользователь переходит к этапу № 2 и попадает на главное окно АИС меню которого отличается в зависимости от прав авторизованного

Этап № 2. Наполнение, редактирование справочника «Службы». Для перехода к данному справочнику необходимо воспользоваться меню и перейти к разделу «Настройки-Справочник-Службы».

Данный справочник представляет из себя отдельную форму. Форма состоит из таблицы, которая содержит наименование служб, дополнительную информацию. Также в окне присутствуют поля ввода наименования и доп. информации. К таблице подключено контекстное меню, позволяющее добавлять и редактировать информацию о службах. Также присутствует быстрый поиск по наименованию службы. На рисунке 3.4.5 изображено окно справочника «Службы».

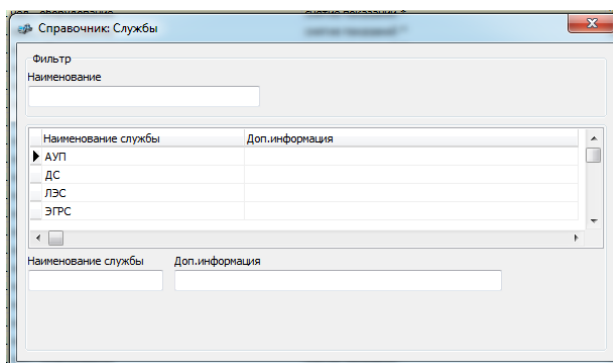


Рисунок 3.4.5 - Справочник «Службы»

Для добавления новой службы необходимо в таблице нажать ПКМ - Добавить\*. Ввести в одноименные поля наименование новой службы, дополнительную информацию. Нажать кнопку «сохранить» для сохранения данных в таблице. На рис. 3.4.6 изображено добавление новой службы.

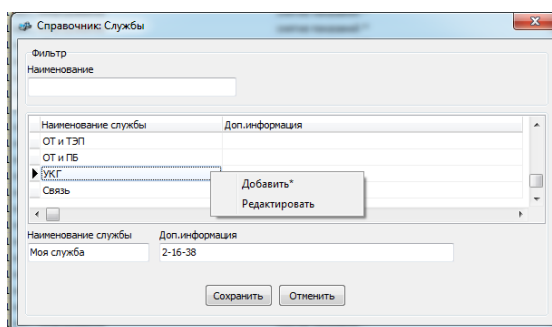


Рисунок 3.4.6 - Добавление новой службы

Для редактирования службы необходимо выбрать службу в таблице и нажать ПКМ - Редактировать. Внести необходимые изменения, нажать кнопку «сохранить» для обновления данных в таблице.

Для быстрого поиска службы пользователю необходимо ввести часть наименования в поле «Наименование». Поиск происходит автоматически, результат поиска представлен на рис. 3.4.7

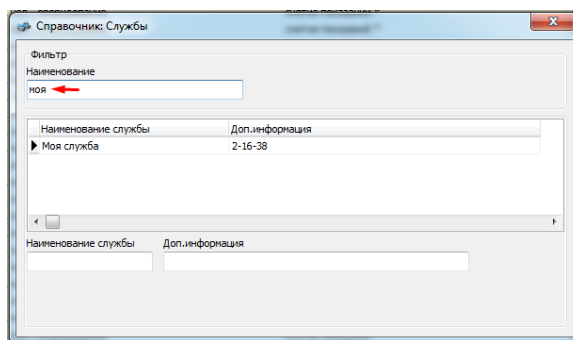


Рисунок 3.4.7 - Поиск службы

Этап № 3. Наполнение, редактирование справочника «Сотрудники». Для перехода к данному справочнику необходимо воспользоваться меню и перейти к разделу «Настройки-Справочник-Сотрудники».

Данный справочник представляет из себя отдельную форму. Форма состоит из таблицы, которая содержит наименование служб, фио сотрудника, пароль для авторизации, профиль прав. Также в окне присутствуют поля ввода основной информации. К таблице подключено контекстное меню, позволяющее добавлять и редактировать информацию о сотрудниках. Также присутствует быстрый поиск по фио сотрудника. На рис. 3.4.8 изображено окно справочника «Сотрудники»

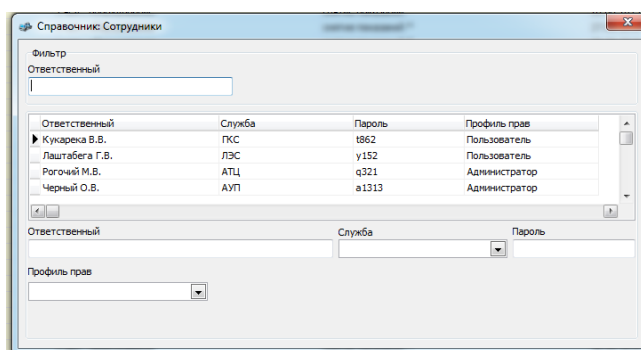


Рис. 3.4.8 - Справочник «Сотрудники»

Для добавления новой службы необходимо в таблице нажать ПКМ - Добавить\*. Ввести в одноименные поля необходимую информацию, выбрать наименование службы и профиль прав. Нажать кнопку «сохранить» для сохранения данных в таблице. На рисунке 3.4.9 изображено добавление новой службы.

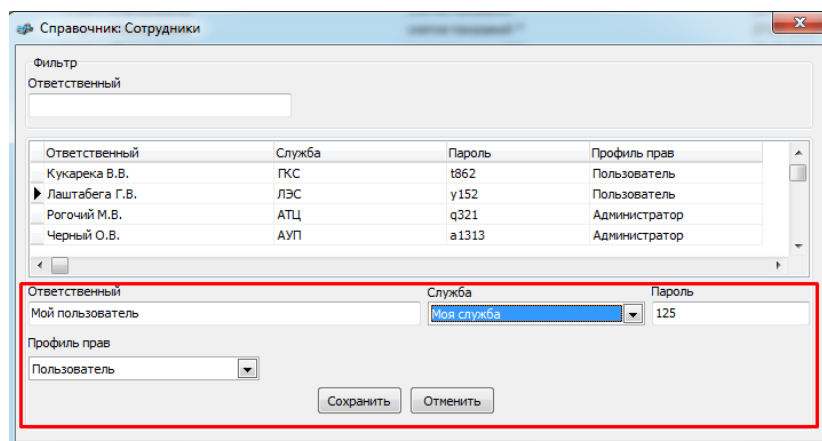


Рисунок 3.4. 9 - Добавление нового пользователя

Для редактирования пользователя необходимо выбрать его в таблице и нажать ПКМ - Редактировать. Внести необходимые изменения, нажать кнопку «сохранить» для обновления данных в таблице.

Для быстрого поиска пользователя необходимо ввести часть фио пользователя в поле «Ответственный». Поиск происходит автоматически, результат поиска представлен на рисунке 3.4.10

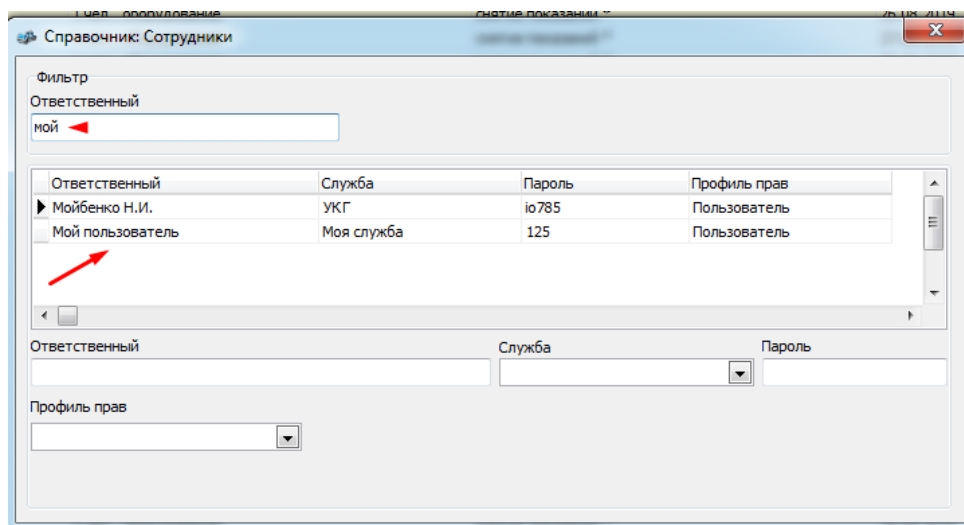


Рисунок 3.4.10 - Поиск пользователя

Этап № 4. Наполнение, редактирование справочника «Транспорт». Для перехода к данному справочнику необходимо воспользоваться меню и перейти к разделу «Настройки-Справочник-Транспорт».

Данный справочник представляет из себя отдельную форму. Форма состоит из таблицы, которая содержит марку транспортного средства, идентификационный номер, ссылку на изображение. Также в окне присутствуют поля ввода марки, номера и кнопка загрузки изображения. К таблице подключено контекстное меню, позволяющее добавлять и редактировать информацию о транспортном средстве. Также присутствует быстрый поиск по марке транспортного средства. На рисунке 3.4.11 изображено окно справочника «Транспорта»

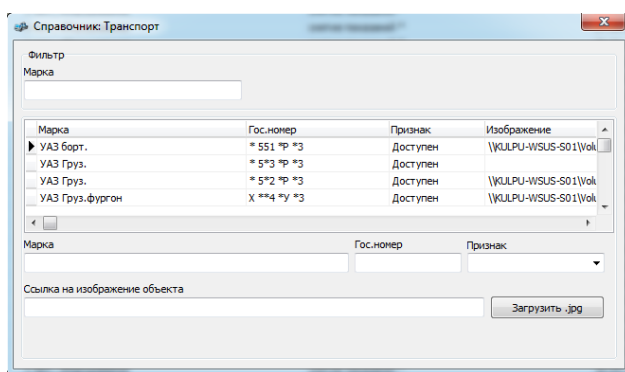


Рисунок 3.4.11 - Справочник «Транспорт»

Для добавления новой службы необходимо в таблице нажать ПКМ - Добавить\*. Ввести в одноименные поля необходимую информацию. Нажать кнопку «сохранить» для сохранения данных в таблице. На рисунке 3.4.12 изображено добавление нового транспортного средства.

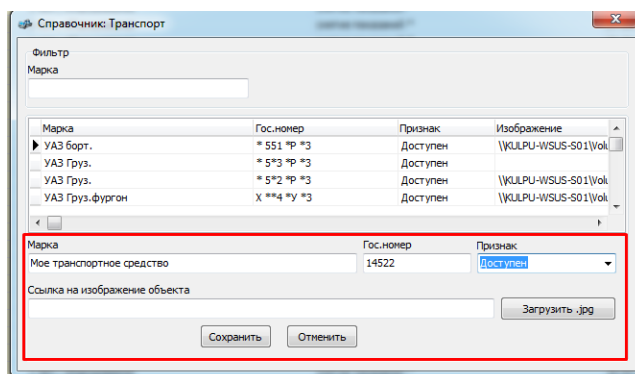


Рисунок 3.4.12 - Добавление нового транспорта

Для редактирования транспортного средства необходимо выбрать его в таблице и нажать ПКМ - Редактировать. Внести необходимые изменения, нажать кнопку «сохранить» для обновления данных в таблице.

Для быстрого поиска транспорта необходимо ввести часть марки в поле «Марка». Поиск происходит автоматически, результат поиска представлен на рисунке 3.4.13

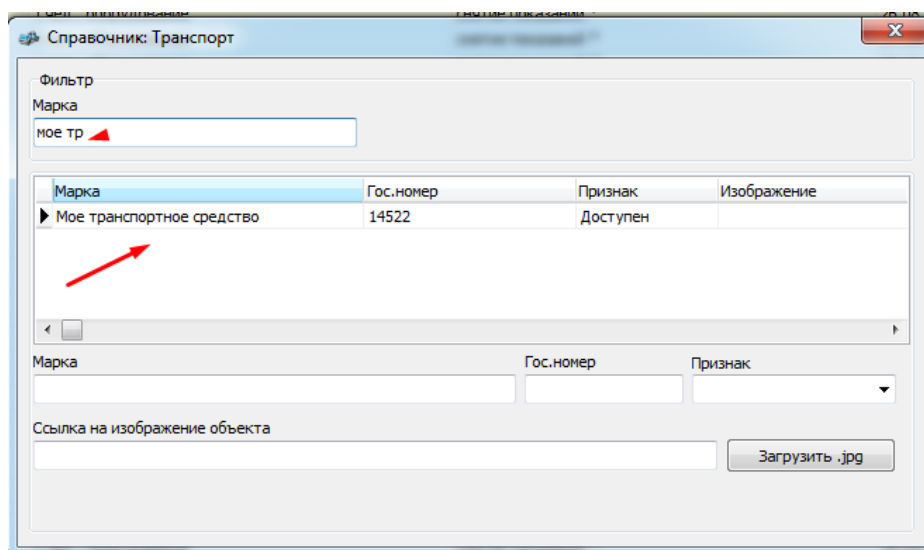


Рисунок 3.4.13 - Поиск транспорта

Для загрузки изображения транспортного средства необходимо нажать кнопку «Загрузить .jpg» и выбрать изображение на жестком диске. Пример выбора представлен на рисунке 3.4.14

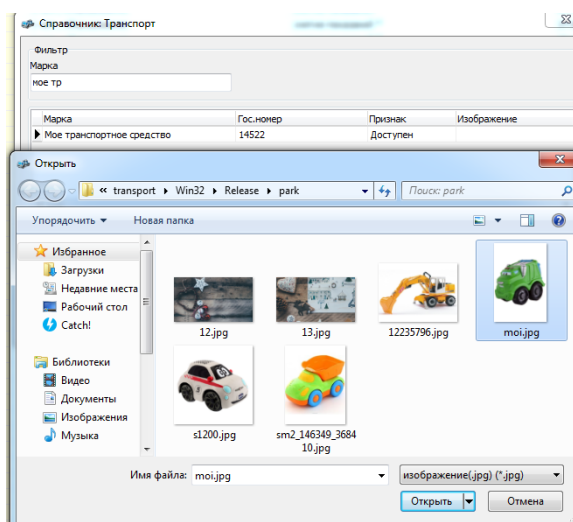


Рисунок 3.4.14 - Выбор изображения

После выбора изображения и сохранения информации пользователь доступен его просмотр при создании заявки и выборе этого транспортного средства. На рисунке 3.4.15 изображена обновленная информация о транспорте с ссылкой на изображение.

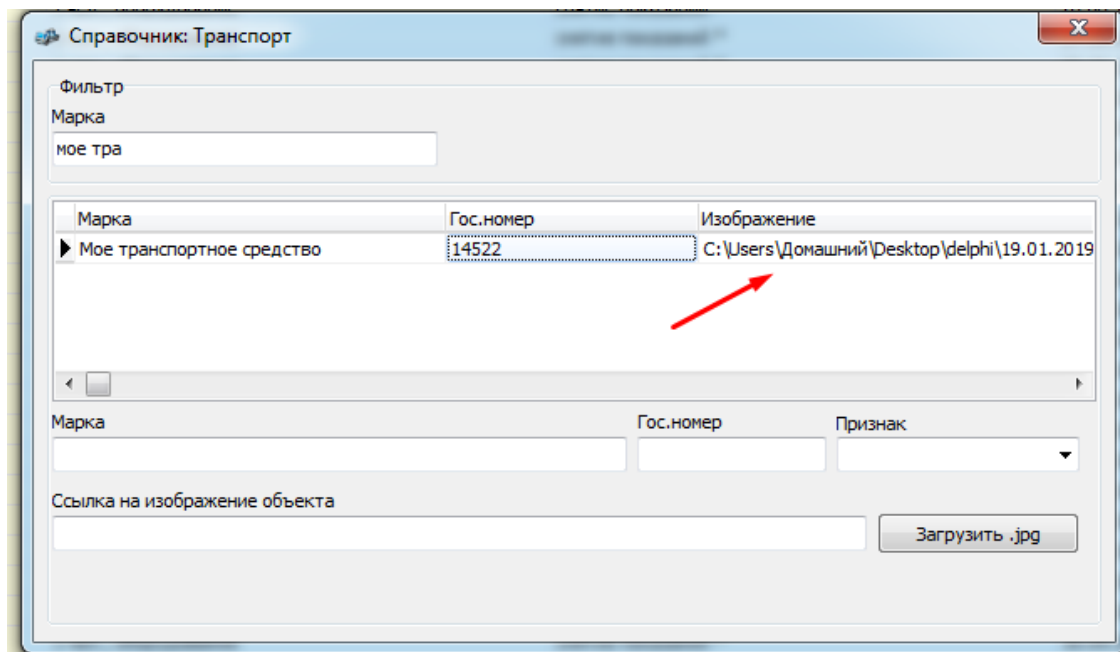


Рисунок 3.4.15 - Информация о загрузке изображения

Этап № 5. Настройка палитры цветов для отображения статуса заявок. На данном этапе каждый пользователь может настроить палитру цветов, которая отображает созданные заявки различными цветами. Цвет зависит от состояния заявки. По умолчанию палитра представлена следующими цветами: светло-зеленый- «Заявка сформирована», светло-голубой- «Заявка на утверждении», светло-желтый-«Заявка в работе», оливковый-«Дубликат». Изображение палитры представлено на рисунке 3.4.16

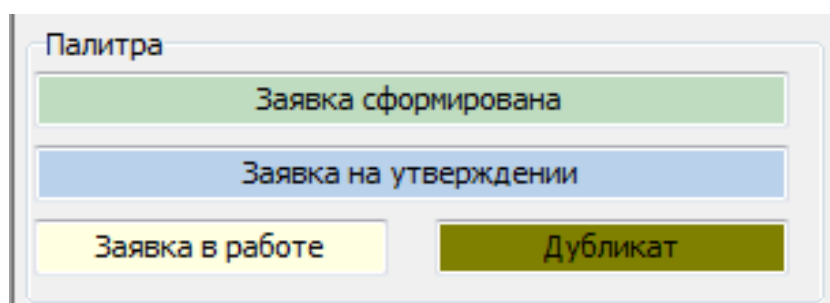


Рисунок 3.4.16 – Палитра



Для изменения цвета отображения заявки необходимо нажать ЛКМ на необходимом состоянии и выбрать понравившийся цвет либо определить свой. Выбор цвета на примере состояния «Заявка в работе» представлен на рисунке 3.4.17

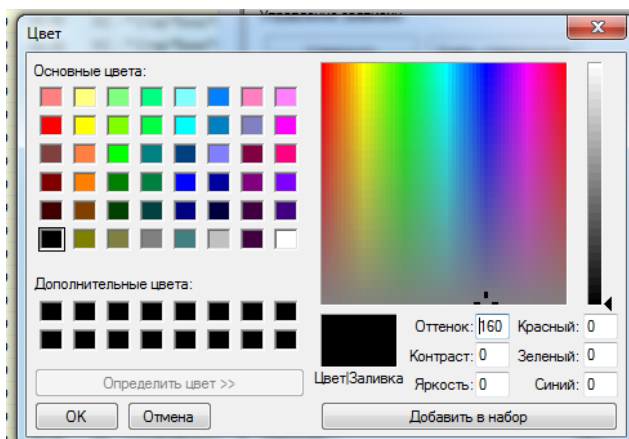


Рисунок 3.4.17 - Определение цвета заявки

Отображение заявки с цветами по умолчанию представлено на рисунке 3.4.18. После выбора цвета необходимо нажать кнопку «ОК» и выбрать любую заявку в главной таблице для изменения цвета. Результат выбора розового цвета для состояния «Заявка в работе» представлен на рисунке 3.4.19

Служба	Ответственный	Транспорт	Гос.номер
САиМО	Гулько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3
САиМО	Гулько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3
САиМО	Гулько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3

Рисунок 3.4.18 - Палитра по умолчанию

Служба	Ответственный	Транспорт	Гос.номер
САиМО	Гулько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3
САиМО	Гулько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3
САиМО	Гулько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3
САиМО	Гулько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3

Рисунок 3.4.19 - Измененная палитра

Выбор цветов не по «умолчанию» сохраняется только в текущей сессии. Если необходимо сохранить понравившуюся палитру для дальнейшего использования необходимо перейти в «меню-настройки-палитра-сохранить»

набор». Здесь же можно вернуть настройки палитры «по умолчанию» или загрузить последний сохраненный набор. Результатом сохранения палитры будет сообщение АИС, представленное на рисунке 3.4.20

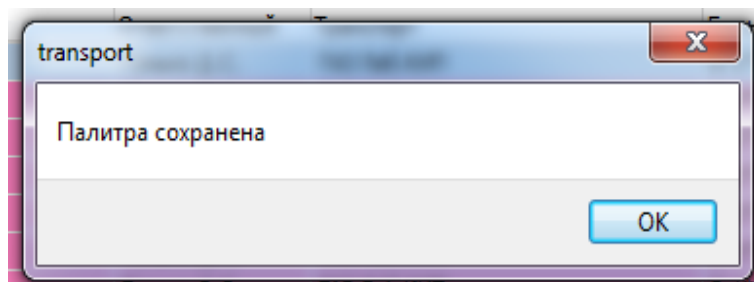


Рисунок 3.4.20 - Результат сохранения

Наборы цветов сохраняются для каждого рабочего места отдельно в файле с расширением .ini который находится в папке АИС и имеет название «color.ini». Загрузка данного файла происходит после авторизации пользователя в ИС.

Этап № 6. Создание, редактирование, удаление заявки на транспортное средство. Перед созданием заявки рекомендуется ознакомиться с интерфейсом главного окна, которое имеет следующую структуру: меню, перечень заявок, сводная информация, палитра, управление заявками, фильтр, информация о заявке, панель состояния. На рисунке 3.4.21 изображено меню, перечень заявок, панель состояния. Панель состояния отображает данные авторизованного пользователя, состояние палитры, версию АИС, организацию и текущее время. Перечень заявок представлен в виде таблицы в которой хранятся заявки пользователей.

Служба	Ответственный	Транспорт	Гос.номер	Бригада(групп)	Цель	Дата в.	Время в.	Дата п.	Время п.	Маршрут
САИМО	Гулько Д.С.	ГАЗ Лаб.КУП	О 1* ОА *3	1 чел., оборудование	снятие показаний *	13.08.2019	08:30	13.08.2019	16:30	КС - * Старобыно*
САИМО	Гулько Д.С.	ГАЗ Лаб.КУП	О 1* ОА *3	1 чел., оборудование	снятие показаний *	14.08.2019	08:30	14.08.2019	16:30	КС - * Старобыно*
САИМО	Гулько Д.С.	ГАЗ Лаб.КУП	О 1* ОА *3	1 чел., оборудование	снятие показаний *	15.08.2019	08:30	15.08.2019	16:30	КС - * Старобыно*

Рисунок 3.4.21 - Палитра, меню, перечень заявок

С правой стороны расположены следующие блоки: сводная информация,

палитра, управление заявками, фильтр, информация о заявке. На рисунке 3.4.22 представлены перечисленные блоки. Блок «Сводная информация» обновляется автоматически после любой операции с заявками в основном блоке.

The screenshot displays a software interface with several sections:

- Сводная информация: 9910**: A table with columns 'Дата выезда' and 'Количество заявок'.

Дата выезда	Количество заявок
29.07.2020	1
24.07.2028	1
13.03.2032	1
- Палитра**: A set of colored buttons: 'Заявка сформирована' (green), 'Заявка на утверждении' (blue), 'Заявка в работе' (pink), and 'Дубликат' (yellow).
- Управление заявками**: Buttons for 'Утвердить' and 'Снять утверждение'.
- Фильтр**: Fields for 'Дата выезда' and 'Состояние', with 'Применить' and 'Сбросить/Обновить' buttons.
- Информация о дате/времени заявки**: Fields for 'Дата выезда', 'Дата прибытия', 'Время выезда', and 'Время прибытия'.
- Информация о транспорте**: Fields for 'Транспорт', 'Маршрут', and 'Планируемый пробег (км.)'.
- Дополнительные сведения**: Fields for 'Состав бригады(груз)' and 'Цель поездки(наим.работ)'.

Рисунок 3.4.22 -Блоки дополнительных действий

Для создания новой заявки необходимо в таблице заявок нажать «ПКМ-Добавить» и внести информацию в блок «Информация о заявке». Дата выезда и дата прибытия выбираются из календаря и проверяются на корректность на этапе выбора даты. В случае неправильной даты АИС предупредит пользователя сообщением, изображенным на рисунке 3.4.23

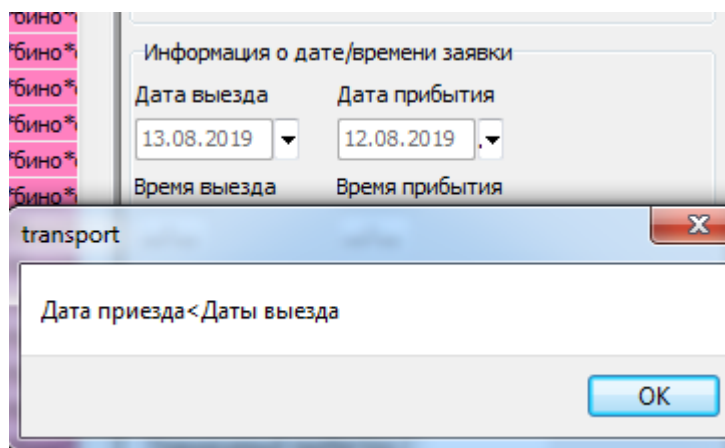


Рисунок 3.4.23 - Предупреждение о неверной дате

Транспорт выбирается из выпадающего списка справочника «Транспорт». Если требуется просмотреть изображения транспортного средства необходимо нажать «enter». Пример просмотра изображения представлен на рисунке 3.4.24

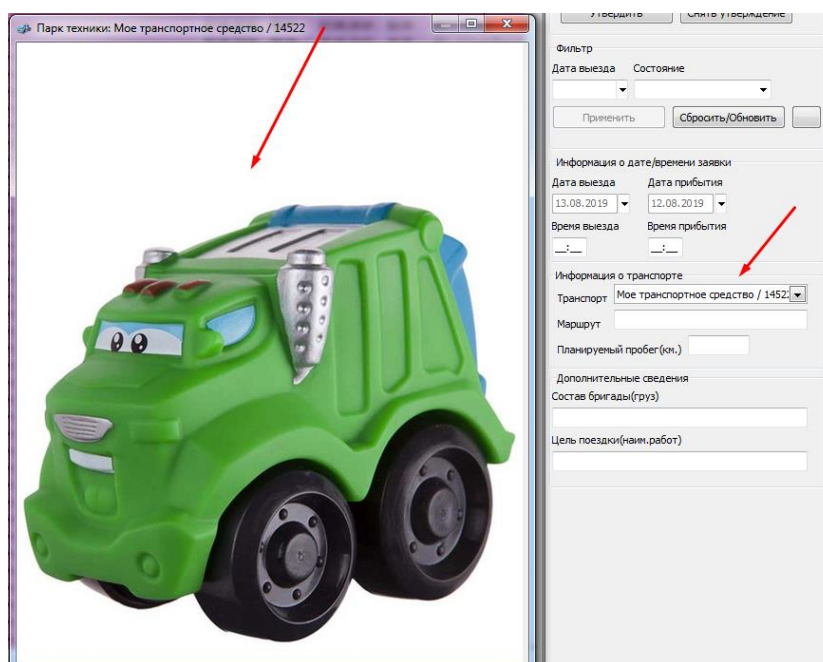


Рисунок 3.4.24 - Просмотр изображения

Далее заполняются дополнительные поля: маршрут, пробег, груз, цель. Для сохранения заявки необходимо нажать кнопку «Сохранить». Созданная заявка получит статус «В работе», отобразится в таблице заявок. Пример заполнения заявки представлен на рисунке 3.4.25

The image shows a form titled "Информация о дате/времени заявки". It contains the following fields: "Дата выезда" (14.08.2019), "Дата прибытия" (15.08.2019), "Время выезда" (12:00), "Время прибытия" (16:00). Below this is "Информация о транспорте" with "Транспорт" (Мое транспортное средство / 14522), "Маршрут" (с. Высокое), and "Планируемый пробег(км.)" (100). The "Дополнительные сведения" section includes "Состав бригады(груз)" (4 человека) and "Цель поездки(наим.работ)" (ремонт моста). At the bottom are "Сохранить" and "Отмена" buttons.

Рисунок 3.4.25 - Заполнение заявки

Для редактирования заявки необходимо в таблице заявок нажать «ПКМ-Редактировать выбранное» и внести изменения в блок «Информация о заявке». Правила проверки дат аналогичны. В случае, когда состояние заявки отлично от «Заявка в работе», «Дубликат» редактирование невозможно. АИС предупредит пользователя сообщением, представленным на рисунке 3.4.26

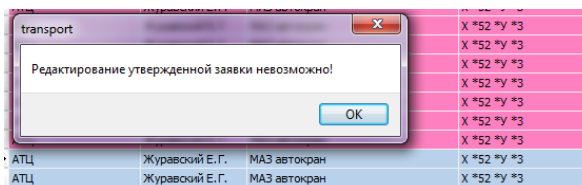


Рисунок 3.4.26 - Сообщение о редактировании

Для удаления заявки необходимо в таблице заявок нажать «ПКМ-Удалить». В случае, когда состояние заявки отлично от «Заявка в работе», «Дубликат» редактирование невозможно. АИС предупредит пользователя сообщением.

Для создания дубликата необходимо в таблице заявок нажать «ПКМ-Дубликат». АИС создаст дубликат выбранной заявки, которой будет присвоено состояние «Дубликат». Дубликаты заявок не рассматриваются администраторами системы для утверждения. Дубликаты необходимо отредактировать для присвоения статуса «В работе».

Этап № 7. Поиск заявки. Для поиска заявок необходимо воспользоваться фильтром, который представлен выпадающим списком с состояниями заявок и датой выезда и нажать кнопку «Применить». Результатом выполнения данного действия (рисунок 3.4.27) станет поиск заявок и отображение их в таблице. Воспользоваться поиском может любой пользователь не в зависимости от профиля прав.

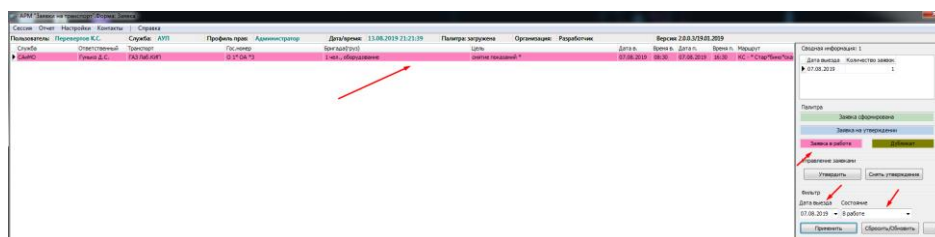


Рисунок 3.4.27 - Поиск заявок

Этап № 8. Утверждение заявки. Утверждение заявки доступно только для профиля прав «Администратор». Для утверждения заявки требуется выделить строку/группу строк. И нажать в блоке «Управление заявками» кнопку «Утвердить». Результатом данного действия станет присвоение выбранным заявкам состояния «Заявка на утверждении». Аналогичным образом используется кнопка «снять утверждение». Результатом выполнения которой станет присвоение выбранным заявкам состояния «В работе». На рисунке 3.4.28 представлен массовый выбор заявок на утверждение.

• САиМО	Гуныко Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3
• САиМО	Гуныко Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3
• САиМО	Гуныко Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3
➤ САиМО	Гуныко Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3

Рисунок 3.4.28 - Массовый выбор

Результат утверждения заявок представлен на рисунке 3.4.29

Служба	Ответственный	Транспорт	Гос.номер
▶ САиМО	Гуныко Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3
САиМО	Гуныко Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3
САиМО	Гуныко Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3
САиМО	Гуныко Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3
САиМО	Гуныко Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3

Рисунок 3.4.29 - Массовое утверждение.

Результат проверки возможности обычным пользователем утвердить заявку представлен на рисунке 3.4.30

Пользователь: **Кукарека В.В.** Служба: **ГКС** Профиль прав: **Пользователь**

Служба	Ответственный	Транспорт	Гос.номер	Бр
▶ САиМО	Гуныко Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3	1
САиМО			1* ОА *3	1
САиМО			1* ОА *3	1
САиМО			1* ОА *3	1
САиМО			1* ОА *3	1
САиМО			1* ОА *3	1
САиМО			1* ОА *3	1
САиМО			1* ОА *3	1

transport

Недостаточно прав!

OK

Рисунок 3.4.30 - Проверка разграничения прав доступа

Этап № 9. Формирование отчетных форм. Для формирования отчетных форм необходимо воспользоваться меню и выбрать «Отчет». Данный раздел доступен только пользователям с профилем прав «Администратор». Формирование отчетных форм происходит только с заявками, состояние которых «Заявка на утверждении». После формирования отчетной формы, заявкам, попавшим в отчет, присваивается состояние «Заявка сформирована». Пример формирования отчетной формы за 13.08.2019 представлен на рисунке 3.4.31. Для удобства формирования формы необходимо выбрать заявку с таблицы заявок и тогда дата заявки будет использована для формирования. Также возможно выбрать формат выгрузки отчета.

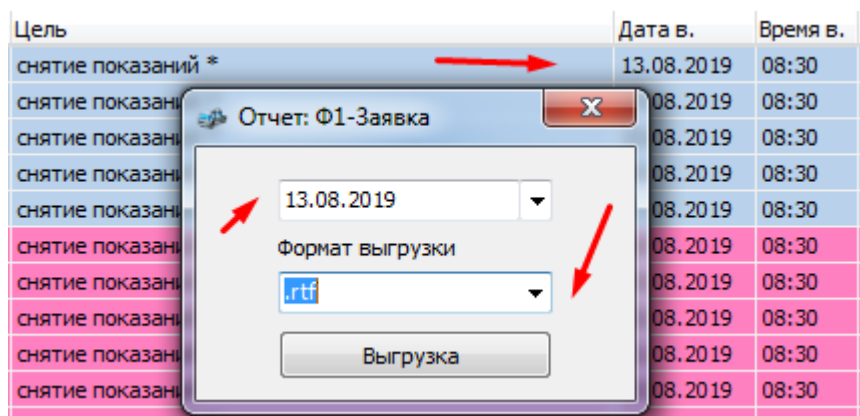


Рисунок 3.4.31 - Пример формирования отчетной формы

Результат формирования отчетной формы представлен на рисунке 3.4.32

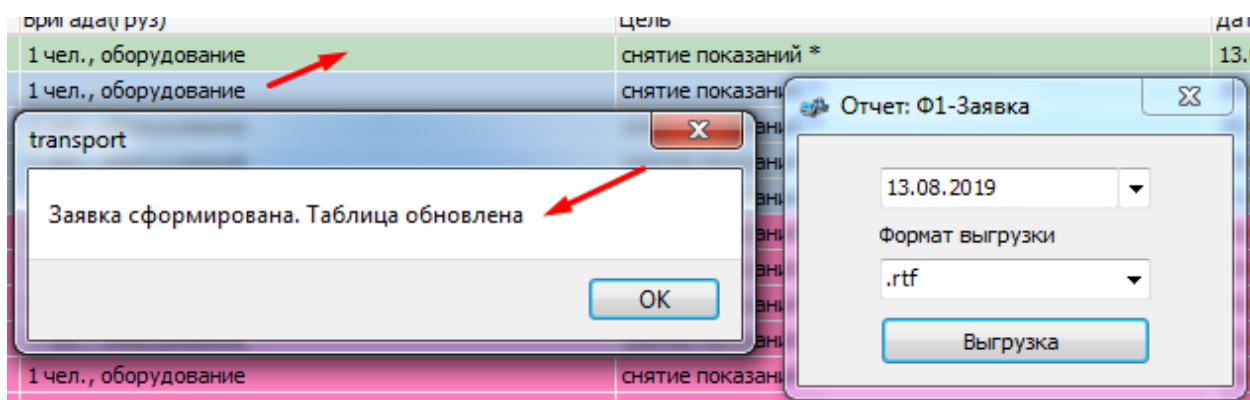


Рисунок 3.4.32 - Результат формирования отчетной формы

Сформированная заявка выгружается в выбранном формате и содержит различную информацию в зависимости от формы отчета. Данную заявку

необходимо сохранить на жестком диске для последующего использования (рисунок 3.4.33)

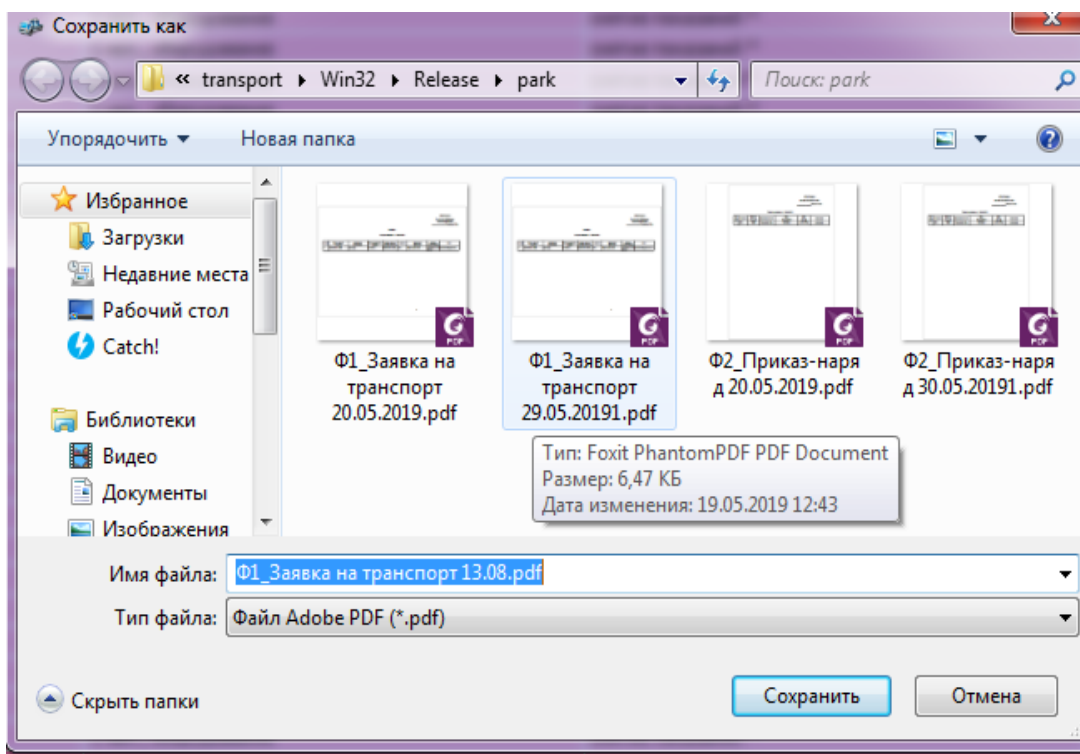


Рисунок 3.4.33 - Сохранение заявки

Внешний вид выбранного отчета представлен на рисунке 3.4.34

Пользователь: Перевертов К.С. | Служба: АУП | Профиль прав: Администратор | Дата/время: 13.08.2019 21:50:41 | Палитра: загружена | Организация: Разработчик | Версия: 2.0.0.3/19.01.2019

Служба	Ответственный	Транспорт	Гос.номер	Бригада(груз)	Цель	Дата в.	Время в.	Дата п.	Время п.	Ма
САИМО	Гунько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3	1 чел., оборудование	снятие показаний *	13.08.2019	08:30	13.08.2019	16:30	КС
САИМО	Гунько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3	1 чел., оборудование	снятие показаний	08.2019	08:30	14.08.2019	16:30	КС
САИМО	Гунько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3	1 чел., оборудование	снятие показаний	08.2019	08:30	15.08.2019	16:30	КС
САИМО	Гунько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3	1 чел., оборудование	снятие показаний	08.2019	08:30	16.08.2019	16:30	КС
САИМО	Гунько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3	1 чел., оборудование	снятие показаний	08.2019	08:30	17.08.2019	16:30	КС
САИМО	Гунько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3	1 чел., оборудование	снятие показаний	08.2019	08:30	18.08.2019	16:30	КС
САИМО	Гунько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3	1 чел., оборудование	снятие показаний	08.2019	08:30	19.08.2019	16:30	КС
САИМО	Гунько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3	1 чел., оборудование	снятие показаний	08.2019	08:30	20.08.2019	16:30	КС
САИМО	Гунько Д.С.	ГАЗ Лаб.КИП	О 1* ОА *3	1 чел., оборудование	снятие показаний	08.2019	08:30	21.08.2019	16:30	КС

Ф1\_Заявка на транспорт 13.08.pdf - Foxit PhantomPDF

УТВЕРЖДАЮ  
Начальник управления  
[Подпись]

**ЗАЯВКА**  
На транспорт на 13.08.2019

№ п/п	Дата, время выезда	Маршрут движения	Состав бригады (груз)	Дата, время возвращения	Пробег	Старший бригады	Марка автомобиля	Цель
1	13.08.2019 08:30	КС - * Стар*бино*ская, Глафиро*ка, Николае*ка	1 чел., оборудование	13.08.2019 16:30	280	Гунько Д.С.	ГАЗ Лаб КИП О 1* ОА *3	снятие показаний *

Рисунок 3.4.35 - Внешний вид отчета



Приложение Г представляет собой форму отчета № 1. Приложение Д представляет собой отчетную форму № 2. Приложение Е представляет собой форму отчета № 3. Приложение Ё представляет собой отчетную форму № 4

Этап № 10. Создание резервной копии базы данных. Для создания резервной копии необходимо воспользоваться меню и выбрать «Настройки-Резервное копирование». Данный раздел доступен только пользователям с профилем прав «Администратор». На рисунке 3.4.36 представлен интерфейс окна для создания резервной копии.

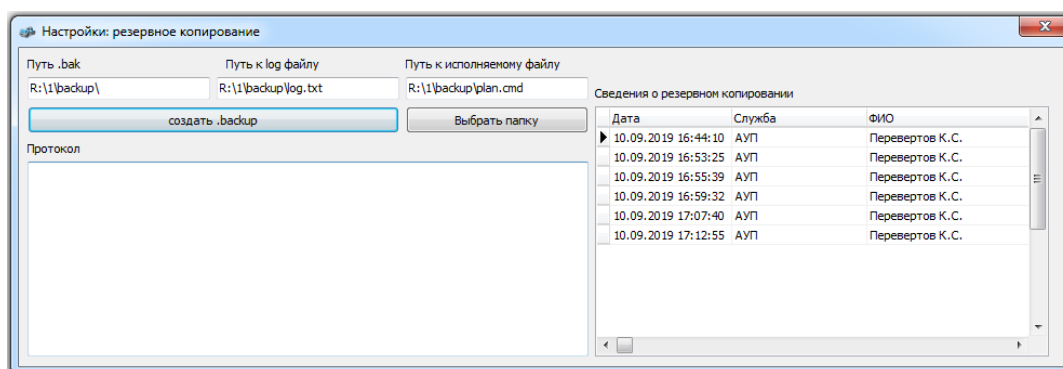


Рисунок 3.4.36 – Интерфейс окна резервной копии

Для выполнения резервной копии необходимо нажать кнопку «Выбрать папку» и подтвердить место сохранения копии базы данных. Далее нажать кнопку «Создать .backup» для создания копии. В окне «протокол» появятся сведения о ходе выполнения операции. В таблице «Сведения о резервном копировании» отобразятся итоговые результаты операции, а также сведения о сотруднике, выполнившем операцию. Итоговый результат представлен на рисунке 3.4.37

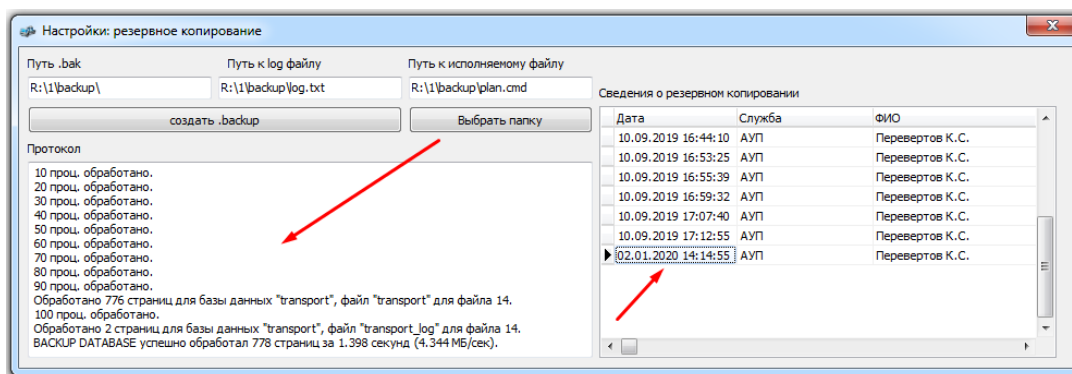


Рисунок 3.4.37 – Результат создания копии БД

Открыв директорию с сохраненной копией можно убедиться в успешном выполнении операции. В файле «log.txt» содержится информация с протокола.

Файл «plan.cmd» содержит команду для запуска скрипта по выполнению копии БД, а также по сохранению данных в файл «log.txt»

На рисунке 3.4.38 представлен результат выполнения операции по резервному копированию. В результате выполнения операции формируется файл с расширением «.bak» который содержит наименования базы данных и дату формирования копии.

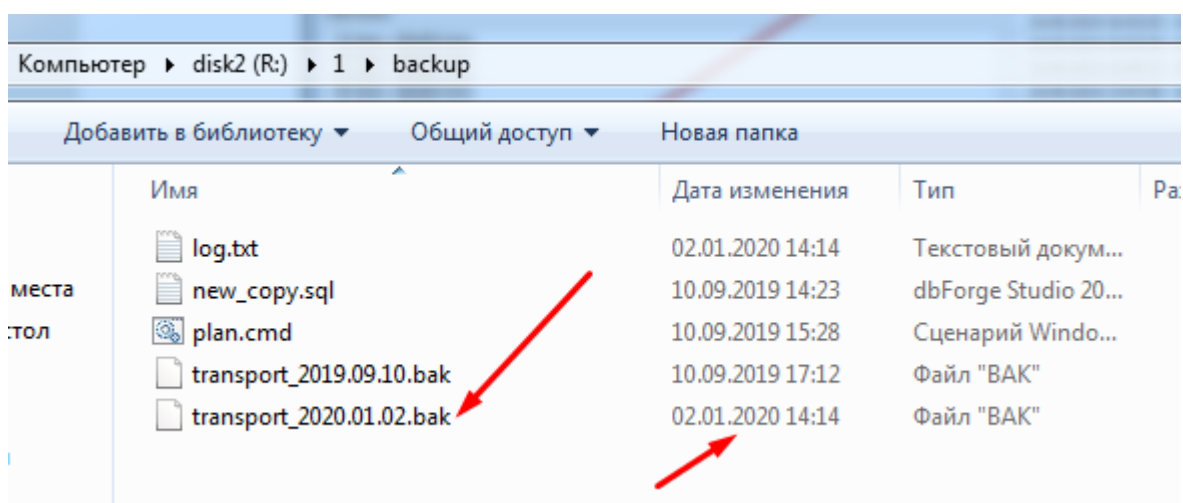


Рисунок 3.4.39 – Создание файла копии базы данных

#### 4 Экономическое обоснование

За основную модель разработки ПО была взята «Incremental Model» [19] (инкрементная модель).

В инкрементной модели полные требования к системе делятся на различные сборки. Терминология часто используется для описания поэтапной сборки ПО. Имеют место несколько циклов разработки, и вместе они составляют жизненный цикл «мульти-водопад». Цикл разделен на более мелкие легко создаваемые модули. Каждый модуль проходит через фазы определения требований, проектирования, кодирования, внедрения и тестирования. Процедура разработки по инкрементной модели предполагает выпуск на первом большом этапе продукта в базовой функциональности, а затем уже последовательное добавление новых функций, так называемых «инкрементов». Процесс продолжается до тех пор, пока не будет создана полная система.

Так как выбрана именно такая модель разработки, то и конечная стоимость продукта неизвестна. На момент написания дипломного проекта АИС находится на этапе расширения функционала. В связи с чем стоимость продукта изменится.

Однако в выбранной модели разработки присутствуют основные этапы, стоимость которых возможно рассчитать с учетом количества времени, потраченного на данный этап.

Основные этапы разработки представлены на рисунке 4.1.



Рисунок 4.1 - Этапы разработки

Для того чтобы получить стоимость выполнения услуг введем в разработку проекта 4 основных должности и обозначим их обязанности. В таблице 4.1 представлен список должностей и обязанности.

Таблица 4.1 – Список основных должностей

Должность	Обязанность
Менеджер проекта	Разработка ТЗ; Организация процесса разработки; Управление изменениями в требованиях; Ведение проектной документации и полной отчетности по проекту.
Дизайнер	Разработки интерфейса; Подготовка и нарезка дизайна для разработчиков.
Разработчик	Написание кода приложения, редактирование кода согласно изменений проектной документации
Тестировщик	Тестирование приложения на рабочем месте, получение детальных сведений о ходе проведения тестирования

Рассмотрим детально сам проект разработки, в связи с выбранной моделью разработки, проект рассчитан на длительный срок. На рисунке 4.2 показана средняя заработная ставка каждого сотрудника. Были использованы данные о минимальных ставках оплаты труда

Пользователь	Значение	Стоимость
 Дизайнер	в час ▾	400
 Менеджер проекта	в час ▾	250
 Разработчик	в час ▾	350
 Тестировщик	в час ▾	200

Рисунок 4.2 - Стоимость работы сотрудника в час

Разработка минимальной версии проекта достаточной для нормального функционирования заняла 30 дней. На рисунке 4.3 представлен базовый план разработки проекта

Задача	исполнитель	начал	окончание	Длительность (дни)	Трудозатраты (часы)	Стоимость	Итоговая стоимость
<b>1. Анали предметной области</b>				<b>2</b>			<b>2000</b>
1.1 Сбор требований	Менеджер	15.10.2019	16.10.2019	1	4	250	1000
1.2 Постановка задачи	Менеджер	16.10.2019	17.10.2019	1	4	250	1000
<b>2. Разработка ТЗ</b>				<b>5</b>			<b>3000</b>
2.1 Составление подробного описания функционала	Менеджер	17.10.2019	21.10.2019	4	8	250	2000
2.2 Определение временных рамок	Менеджер	21.10.2019	22.10.2019	1	4	250	1000
<b>3. Разработка ИС</b>				<b>12</b>			<b>13150</b>
3.1 Выбор модели разработки	Менеджер	22.10.2019	23.10.2019	1	1	250	250
3.2 Проектирование БД	Разработчик	23.10.2019	27.10.2019	4	16	350	5600
3.3 Разработка интерфейса	Дизайнер	27.10.2019	30.10.2019	3	12	400	4800
3.4 Разработка алгоритма работы АИС	Менеджер	30.10.2019	03.11.2019	4	10	250	2500
<b>4. Разработка ПО</b>				<b>8</b>			<b>11200</b>
4.1 Разработка первой версии АИС	Разработчик	03.11.2019	11.11.2019	8	32	350	11200
<b>5. Тестирование</b>				<b>3</b>			<b>2600</b>
5.1 Формирования списка недоработок	Тестировщик	11.11.2019	12.11.2019	1	3	200	600
5.2 Определение сроков исправления	Разработчик	12.11.2019	12.11.2019	0	1	350	350
5.3 Доработка ПО	Разработчик	12.11.2019	13.11.2019	1	3	350	1050
5.4 Компиляция ПО	Разработчик	13.11.2019	13.11.2019	0	1	350	350
5.5 Установка ПО	Менеджер	14.11.2019	15.11.2019	1	1	250	250
<b>ВСЕГО</b>				<b>30</b>	<b>100</b>		<b>31950</b>

Рис. 4.3 - Базовый план разработки

Диаграмма Ганта [20] представлена на рисунке 4.4. Данная диаграмма является одним из методов планирования проектов, которая используется для иллюстрации графика работ по проекту.

На диаграмме предусмотрены все этапы и соблюдена очерёдность, причём каждый этап начинается только после завершения предыдущего. Как показано, весь процесс разделён на 6 задач, каждая из которых имеет свои подзадачи, с конкретным сроком выполнения. Вследствие данных были сформированы затраты на трудовые ресурсы, которые составили 31950.

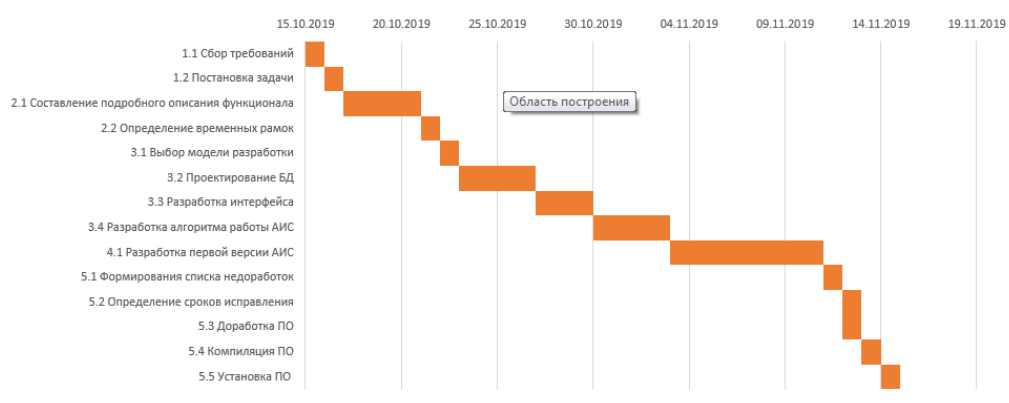


Рис. 4.4 - Диаграмма Ганта

Для того чтобы определить экономическую эффективность [21] необходимо сравнить таких два показателя как, расходы и доходы.

Экономической эффективностью является отношение величины выручки, к величине всех расходов.

$$\mathcal{E}_{\text{эф}} = \Pi / \mathcal{Z} \quad (1)$$

Где:  $\mathcal{E}_{\text{эф}}$  – экономическая эффективность;  $\Pi$  – прибыль;  $\mathcal{Z}$  – сумма всех затрат.

Чаще всего экономическую эффективность рассчитывают по данным за год. Он определяется как разность между годовой прибылью и долей затрат, на этот год.

$$\mathcal{E}_{\text{эф}} = \mathcal{E}_{\text{год}} - (K * E) \quad (2)$$

Где:  $\mathcal{E}_{\text{год}}$  – годовой экономический эффект от реализации АИС;  $\mathcal{E}_{\text{эф}}$  – экономическая эффективность;  $K$  – затраты на реализацию;  $E$  – коэффициент затрат.

Критерии для принятия решения о создании ИС:  $\mathcal{E} > 0$ ;  $E > E_n$ ;  $T < 1/E_n$

Для того чтобы оценить экономическую эффективность необходимо определить затраты на разработку ИС и прибыль за год. Разовые затраты на разработку ИС составили 31950 рублей, такое значение получилось с условием, что не будут затраты на съём помещения и работа будет написана полностью бесплатно. Осталось посчитать предлагаемый доход после продажи АИС. Первым способом получения дохода является заключение договора на сопровождение АИС с ежемесячной оплатой. Вторым способом является разработка специализированных отчетных форм и доработка функционала посредством заключения разового договора на обслуживание.

В случае заключения договора с ежемесячной оплатой организация выплачивает 3000р. Итого за год: 36000р.

Рассчитаем коэффициент экономической эффективности:

$$E = \mathcal{E}_{\text{год}} / K \quad (3)$$

$$E = 36\,000 / 31\,950 \approx 1,13$$

Посчитаем срок окупаемости в годах:

$$T = K / \mathcal{E}_{\text{год}} \quad (4)$$

$$T = 128\,000 / 188\,000 \approx 0,89$$

Следовательно, проект окупиться за год. Все критерии соответствуют вышеизложенному условию, экономическая эффективность достаточно высокая. Разработка и внедрение АИС является эффективной и оправданной.

## Заключение

В рамках выпускной квалификационной работы (далее - ВКР) была разработана информационная система автоматизации бронирования транспорта.

В процессе разработки были решены следующие задачи:

- Проведен анализ объекта и предмета исследования: описаны объект исследования и предмет исследования, постановлены задачи на проектирование и информатизацию исследуемого процесса
- Проведен существующих разработок
- Описана предполагаемая модель исследуемого процесса
- Описаны виды обеспечения предполагаемой модели
- Обоснован выбор средств разработки
- Разработан интерфейс ИС, база данных

Разработанная информационная система является готовым автономным программным обеспечением, которое позволяет полностью решить поставленную перед ним задачу.

Информационная система разрабатывалась с учетом будущего развития согласно требованиям, что подразумевает не только создание дополнительных отчетных форм, но также и дополнительного функционала.

Для развития информационной системы в будущем будет использован следующий функционал:

- Получение координат транспортного средства с сохранением их в базе данных. Данный функционал позволит перестроить логистику для максимальной оптимизации маршрута транспортного средства. Доработка возможна только при согласовании информационной безопасности. Требуется дополнительная настройка оборудования провайдера.
- Сохранение настроек цветовой палитры пользователя в базе данных.



Данный функционал позволит авторизованному пользователю использовать ИС в привычном виде не в зависимости от места авторизации пользователя.

- Выполнение скриптов, необходимых для обновления без использования вспомогательных средств. Данный функционал позволит выполнять скрипты из клиента ИС. Также стоит отметить, что скрипты будут находится в зашифрованном виде и открываться только в ИС. Обязательным пунктом будет ведение отчета о ходе выполнения скриптов.

- Обновление клиента ИС из общей директории локальной сети.

Функционал позволит обновлять АРМ специалистов полностью автоматически без использования ручного обновления.

Стоит отметить, что разработанная ИС на данный момент функционирует только в локальной сети учреждения, что не позволяет связать ее с сетью Интернет. Данное ограничение выполнено согласно требованиям информационной безопасности организации.

Информационная система была разработана таким образом, что может использоваться не только в выбранной организации, но и в сторонних организациях. Это позволило мне внедрить данную ИС в другую организацию и сопровождать ее.

В ходе выполнения ВКР были выявлен ряд ограничений при выбранном средстве разработки. Функционал был полностью реализован, однако было принято решение о смене языка программирования на более развивающиеся. Моим выбором стал язык программирования С#. Стоит также обратить внимание на web и мобильные разработки, так как развитие ИС в частности передача координат, напрямую с мобильной разработкой.

Знания, полученные при выполнении ВКР, используются мной при реализации других проектов.

## Список используемой литературы

1. Александров, Д.В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы: учеб. пособие / Д.В. Александров. - М.: Финансы и статистика, 2017. - 224 с.
2. Информационные технологии и вычислительные системы: Обработка информации и анализ данных. Программная инженерия. Математическое моделирование. Прикладные аспекты информатики / Под ред. С.В. Емельянова. - М.: Ленанд, 2015. - 104 с.
3. Информационные технологии и вычислительные системы. Вычислительные системы. Компьютерная графика. Распознавание образов. Математическое моделирование / Под ред. С.В. Емельянова. - М.: Ленанд, 2015. - 100 с.
4. Delphi XE. Изучение функционала [Электронный ресурс]. URL <http://www.delphiplus.org/projects/books/index.html> (дата обращения: 12.12.2019).
5. Реализация методов предметно-ориентированного проектирования / Под редакцией В.Вон – М.: Вильямс, 2018. – 54 с.
6. Хранимые процедуры MS SQL Server [Электронный ресурс]. URL <https://docs.microsoft.com/ru-RU/sql/relational-databases/stored-procedures/create-a-stored-procedure?view=aps-pdw-2016> (дата обращения: 12.12.2019).
7. Программа «Автопарк» [Электронный ресурс]. URL <https://www.pisoft.ru/auto//> (дата обращения: 12.12.2019).
8. Программа «Контроль и учет автотранспорта» [Электронный ресурс]. URL <https://www.avto-uchet.ru/> (дата обращения: 12.12.2019).
9. Учись видеть бизнес-процессы. Построение карт потоков создания ценности / Под ред. Майк Ротер, 2015. - 98 с.
10. Официальный сайт ГБУЗ «ТЦРБ № 1» МЗ КК [Электронный ресурс]. URL <https://trb1-tuapse.ru/> (дата обращения: 12.12.2019).

11. Технология ADO [Электронный ресурс]. URL <http://www.delphiplus.org/programirovanie-baz-dannih-delphi7/tehnologiya-ado.html> (дата обращения: 12.12.2019).
12. MS SQL Server [Электронный ресурс]. URL <https://metanit.com/sql/sqlserver/2.1.php> (дата обращения: 12.12.2019).
13. Знакомство с нотацией IDEF0 [Электронный ресурс]. URL <https://habr.com/ru/company/trinion/blog/322832/> (дата обращения: 12.12.2019).
14. Универсальный инструмент для работы с базами данных SQL Server [Электронный ресурс]. URL <https://www.devart.com/ru/dbforge/sql/studio/> (дата обращения: 08.01.2020).
15. Delphi 7 для профессионалов / Под редакцией Кэнту М. – М.: Питер, 2004. – 702 с.
16. FastReport 4.6 Руководство пользователя [Электронный ресурс]. URL <https://www.fastreport.ru/documentation/UserMan-ru/index.html> (дата обращения: 08.01.2020).
17. Embarcadero. Product Documentation Wikis [Электронный ресурс]. URL <http://docwiki.embarcadero.com/> (дата обращения: 08.01.2020).
18. Academic. ER-модель данных [Электронный ресурс]. URL <https://dic.academic.ru/dic.nsf/ruwiki/147094> (дата обращения: 08.01.2020).
19. Habr. 7 основных моделей разработки [Электронный ресурс]. URL <https://habr.com/ru/company/edison/blog/269789/> (дата обращения: 08.01.2020).
20. Рассел, Джесси Диаграмма Ганта / Джесси Рассел. - М.: VSD, 2012. - 591 с.
21. Экономист года 2016. Сборник статей II Международного научно-практического конкурса: сборник статей / под ред. ИП Гуляев Г.Ю. Издатель: Наука и просвещение, 2016. -101 с.
22. Культин, Н.Б. Основы программирования в Delphi XE / Н.Б. Культин. - СПб.: BHV, 2011. - 416 с.

23. Черпаков, И.В. Основы программирования: Учебник и практикум для прикладного бакалавриата / И.В. Черпаков. - Люберцы: Юрайт, 2016. - 219 с.
24. Юдин, Д.Б. Задачи и методы линейного программирования: Математические основы и практические задачи / Д.Б. Юдин, Е.Г. Гольштейн. - М.: КД Либроком, 2010. - 320 с.
25. Голицына, О.Л. Основы алгоритмизации и программирования: Учебное пособие / О.Л. Голицына, И.И. Попов. - М.: Форум, 2013. - 205 с.
26. Основы программирования. Учебник с практикумом / Под ред. Макаровой Н.В.. - М.: КноРус, 2017. - 352 с.
27. Зыков, С.В. Основы современного программирования: Учебное пособие для вузов / С.В. Зыков. - М.: ГЛТ, 2012. - 444 с.
28. Кундиус, В.А. Теоретические основы разработки и реализации языков программирования / В.А. Кундиус. - М.: КноРус, 2013. - 184 с.
29. Довек, Ж. Введение в теорию языков программирования / Ж. Довек, Ж.-Ж. Леви. — М.: ДМК, 2016. — 134 с.
30. Кауфман, В.Ш. Языки программирования. Концепции и принципы / В.Ш. Кауфман. — М.: ДМК, 2017. — 464 с.

# Приложение А

## Листинг приложения

```
unit Unit_login;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.Win.ADODB, Vcl.Grids,
  Vcl.DBGrids, Vcl.StdCtrls, Vcl.Buttons, Vcl.DBCtrls, IniFiles, ExtCtrls;
type
TForm_login = class(TForm)
  ADOConnection1: TADOConnection;
  DataSource1: TDataSource;
  ADOQuery_lic: TADOQuery;
  ADOQuery_v_lic: TADOQuery;
  DBLookupComboBox1: TDBLookupComboBox;
  Edit1: TEdit;
  Label1: TLabel;
  Label2: TLabel;
  ADOQuery_park: TADOQuery;
  DataSource_park: TDataSource;
  ADOQuery_ADD_z: TADOQuery;
  ADOQuery_FULL: TADOQuery;
  DataSource_FULL: TDataSource;
  ADOQuery_del_z1: TADOQuery;
  ADOQuery_z1_r: TADOQuery;
  DataSource_f1exp: TDataSource;
  ADOQuery_f1exp: TADOQuery;
  Edit2: TEdit;
  BitBtn2: TBitBtn;
  Memo1: TMemo;
  ADOQuery_f2exp: TADOQuery;
  DataSource_f2exp: TDataSource;
  ADOQuery_sum: TADOQuery;
  DataSource_sum: TDataSource;
  ADOQuery1: TADOQuery;
  ADOQuery_park_r: TADOQuery;
  DataSource_park_r: TDataSource;
  ADOQuery_park_img: TADOQuery;
  ADOQuery_otdel: TADOQuery;
  DataSource_otdel: TDataSource;
  ADOQuery_otdel_r: TADOQuery;
  ADOQuery_lic_spr: TADOQuery;
  DataSource_lic_spr: TDataSource;
  ADOQuery_spr_otdel: TADOQuery;
  DataSource_spr_otdel: TDataSource;
  ADOQuery_profil: TADOQuery;
  DataSource_profil: TDataSource;
  ADOQuery_lic_spr_r: TADOQuery;
  Edit3: TEdit;
  ADOQuery_f3: TADOQuery;
  DataSource_f3: TDataSource;
  ADOQuery_glob: TADOQuery;
  ADOQuery_copy: TADOQuery;
  DataSource_copy: TDataSource;
  DataSource_f3exp: TDataSource;
  ADOQuery_f3exp: TADOQuery;
  ADOQuery_f5: TADOQuery;
  DataSource_f5: TDataSource;
```

## Продолжение приложения А

```
ADOQuery_f4_p: TADOQuery;
  DataSource_f4_p: TDataSource;
  ADOQuery_f5_p: TADOQuery;
  DataSource_f5_p: TDataSource;
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure BitBtn5Click(Sender: TObject);
  procedure Edit1KeyPress(Sender: TObject; var Key: Char);
  procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form_login: TForm_login;
  inifile:TIniFile;
  a,b,c,d, i,f:string;
implementation
  {$R *.dfm}
  uses Unit_GL;
  procedure TForm_login.BitBtn1Click(Sender: TObject);
  var
    td:TDateTime;
  begin
    Form_login.ADOQuery_lic.Close;
    Form_login.ADOQuery_lic.SQL.Clear;
    Form_login.ADOQuery_lic.SQL.Add('lic_pas :FIO, :PAS');
    Form_login.ADOQuery_lic.Parameters.ParamByName('FIO').Value:=Form_login.DBLookupComboBox1.KeyValue;
    Form_login.ADOQuery_lic.Parameters.ParamByName('PAS').Value:=Form_login.Edit1.Text;
    Form_login.ADOQuery_lic.Open;
    if Form_login.ADOQuery_lic.RecordCount>0
    then begin
      Form_GL.show;
      Form_login.Visible:=false;
      Form_GL.StatusBar1.Panels[1].Text:=Form_login.ADOQuery_lic.FieldByName('fio').AsString;
      Form_GL.spec.Text:=Form_login.ADOQuery_lic.FieldByName('spec').AsString;
      Form_GL.id_lic.Text:=Form_login.ADOQuery_lic.FieldByName('id').AsString;
      Form_GL.id_ot.Text:=Form_login.ADOQuery_lic.FieldByName('id_otd').AsString;
      td:=Now;
      Form_GL.StatusBar1.Panels[3].Text:=DateTimeToStr(td);
      Form_login.ADOQuery_sum.Close;
      Form_login.ADOQuery_sum.SQL.Clear;
      Form_login.ADOQuery_sum.SQL.Add('sum_1 :d1');
      Form_login.ADOQuery_sum.Parameters.ParamByName('d1').Value:=DateToStr(td);
      Form_login.ADOQuery_sum.Open;
      Form_login.ADOQuery_FULL.Close;
      Form_login.ADOQuery_FULL.SQL.Clear;
      Form_login.ADOQuery_FULL.SQL.Add('obsh_v :d1');
      Form_login.ADOQuery_FULL.Parameters.ParamByName('d1').Value:=DateToStr(td);
      Form_login.ADOQuery_FULL.Open;
    end else
      ShowMessage('Пользователь не найден');
    end;
  procedure TForm_login.BitBtn2Click(Sender: TObject);
  begin
    //считываем данные из файла
```

## Продолжение приложения А

```
Memo1.Lines.LoadFromFile('conf.txt');
//присваиваем строку подключения
ADOConnection1.ConnectionString:=Memo1.Text;
//получаем имя сервера БД
Edit3.Text:=ADOConnection1.Properties.Item['Data Source'].Value;
//проверяем соответствие
if (edit3.Text='ASU3\SQLEXPRESS') or (Edit3.Text='kulpu-wsus-s01') or (Edit3.Text='mynout') or
(Edit3.Text='kulpu-s05') or (Edit3.Text='ДОМАШНИЙ-ПК\SQLEXPRESS') then
begin
//удачное соединение. включаем компоненты запроса adoquery
try
ADOConnection1.LoginPrompt:=false;
ADOConnection1.Connected:=true;
ShowMessage('Соединение успешно');
ADOQuery_v_lic.Active:=true;
ADOQuery_lic.Active:=true;
ADOQuery_lic_spr.Active:=true;
ADOQuery_park.Active:=true;
ADOQuery_ADD_z.Active:=true;
ADOQuery_f3.Active:=true;
ADOQuery_del_z1.Active:=true;
ADOQuery_f1exp.Active:=true;
ADOQuery_park_r.Active:=true;
ADOQuery_otdel.Active:=true;
ADOQuery_otdel_r.Active:=true;
ADOQuery_spr_otdel.Active:=true;
ADOQuery_profil.Active:=true;
DBLookupComboBox1.Enabled:=true;
ADOQuery_lic_spr_r.Active:=true;
ADOQuery_glob.Active:=true;
ADOQuery_copy.Active:=true;
ADOQuery_f3exp.Active:=true;
ADOQuery_f5.Active:=true;
ADOQuery_f4_p.Active:=true;
ADOQuery_f5_p.Active:=true;
Edit1.Enabled:=true;
BitBtn2.Visible:=false;
//неудачное соединение. Вывод ошибки в сообщении
except
on E:Exception do begin
DBLookupComboBox1.Enabled:=false;
Edit1.Enabled:=false;
ShowMessage('Ошибка:'+e.message+#13#10+'ident');
end;
end;
end
else
ShowMessage('Сервер не зарегистрирован в приложении');
end;
procedure TForm_login.BitBtn3Click(Sender: TObject);
begin
ADOConnection1.Connected:=false;end;
procedure TForm_login.BitBtn5Click(Sender: TObject);
begin
Memo1.Lines.LoadFromFile('conf.txt');
end;
procedure TForm_login.Edit1KeyPress(Sender: TObject; var Key: Char);
var
TD:TDateTime;
```

## Продолжение приложения А

```
TD1:TDateTime;
begin
if Ord(Key) = VK_RETURN then
  begin
Form_login.ADOQuery_lic.Close;
Form_login.ADOQuery_lic.SQL.Clear;
Form_login.ADOQuery_lic.SQL.Add('lic_v :fio, :pas');
Form_login.ADOQuery_lic.Parameters.ParamByName('FIO').Value:=Form_login.DBLookupComboBox1.KeyValue;
Form_login.ADOQuery_lic.Parameters.ParamByName('PAS').Value:=Form_login.Edit1.Text;
Form_login.ADOQuery_lic.Open;
if Form_login.ADOQuery_lic.RecordCount>0
then begin
Form_GL.show;
Form_login.Visible:=false;
Form_GL.StatusBar1.Panels[1].Text:=Form_login.ADOQuery_lic.FieldByName('fio').AsString;
Form_GL.StatusBar1.Panels[3].Text:=Form_login.ADOQuery_lic.FieldByName('name_otd').AsString;
Form_GL.StatusBar1.Panels[5].Text:=Form_login.ADOQuery_lic.FieldByName('name_p').AsString;
a:=Form_login.ADOQuery_lic.FieldByName('fio').AsString;
b:=Form_login.ADOQuery_lic.FieldByName('name_otd').AsString;
c:=Form_login.ADOQuery_lic.FieldByName('name_p').AsString;
Form_GL.spec.Text:=Form_login.ADOQuery_lic.FieldByName('spec').AsString;
Form_GL.id_lic.Text:=Form_login.ADOQuery_lic.FieldByName('id').AsString;
Form_GL.id_ot.Text:=Form_login.ADOQuery_lic.FieldByName('id_otd').AsString;
Form_GL.edit11.Text:=Form_login.ADOQuery_lic.FieldByName('id').AsString;
td:=Now;
Form_GL.StatusBar1.Panels[7].Text:=DateTimeToStr(td);
Form_login.ADOQuery_sum.Close;
Form_login.ADOQuery_sum.SQL.Clear;
Form_login.ADOQuery_sum.SQL.Add('sum_v :d1,:d2' );
Form_login.ADOQuery_sum.Parameters.ParamByName('d1').Value:=DateToStr(td);
Form_login.ADOQuery_sum.Parameters.ParamByName('d2').Value:=StrToDate('31.12.2099');
Form_login.ADOQuery_sum.Open;
Form_login.ADOQuery_FULL.Close;
Form_login.ADOQuery_FULL.SQL.Clear;
Form_login.ADOQuery_FULL.SQL.Add('obsh_v :d1, :d2');
Form_login.ADOQuery_FULL.Parameters.ParamByName('d1').Value:=DateToStr(td);
Form_login.ADOQuery_full.Parameters.ParamByName('d2').Value:=StrToDate('31.12.2099');
Form_login.ADOQuery_FULL.Open;
Form_GL.GroupBox7.Caption:=Form_GL.GroupBox7.Caption+'
'+IntToStr(Form_login.ADOQuery_full.RecordCount);
Form_GL.BitBtn5.Enabled:=false;
if Form_GL.spec.Text='2'
then Form_GL.N18.Visible:=true
else
Form_GL.N2.Visible:=true;
end
else
ShowMessage('Пользователь не найден');
  end;
end;
procedure TForm_login.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Application.Terminate;
end;
procedure TForm_login.FormCreate(Sender: TObject);
begin
end;end.
unit Unit_copy;
interface
```



## Продолжение приложения А

```
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, ShellApi, Data.DB,
  Vcl.Grids, Vcl.DBGrids, TIHelp32, Vcl.ExtCtrls;
type
  TForm_copy = class(TForm)
    Button1: TButton;
    Edit1: TEdit;
    Memo1: TMemo;
    Label1: TLabel;
    DBGrid1: TDBGrid;
    Label2: TLabel;
    Label3: TLabel;
    Edit3: TEdit;
    Label4: TLabel;
    Edit4: TEdit;
    Label5: TLabel;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form_copy: TForm_copy;
implementation
  {$R *.dfm}
  uses Unit_GL, Unit_login;
  function processExists(exeFileName: string): Boolean;
  var
    ContinueLoop: BOOL;
    FSnapshotHandle: THandle;
    FProcessEntry32: TProcessEntry32;
  begin
    FSnapshotHandle := CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);
    FProcessEntry32.dwSize := SizeOf(FProcessEntry32);
    ContinueLoop := Process32First(FSnapshotHandle, FProcessEntry32);
    Result := False;
    while Integer(ContinueLoop) <> 0 do
      begin
        if ((UpperCase(ExtractFileName(FProcessEntry32.szExeFile)) =
          UpperCase(ExeFileName)) or (UpperCase(FProcessEntry32.szExeFile) =
          UpperCase(ExeFileName))) then
          begin
            Result := True;
          end;
        ContinueLoop := Process32Next(FSnapshotHandle, FProcessEntry32);
      end;
    CloseHandle(FSnapshotHandle);
  end;
  procedure TForm_copy.Button1Click(Sender: TObject);
  var Find:string; // Искомая строка
      i:integer;
      td:TDateTime;
      stat:string;
  begin
```

## Продолжение приложения А

```
ShellExecute(handle,nil,(pchar(edit1.Text)),nil,nil,sw_normal);
Memo1.Lines.LoadFromFile(Edit3.Text);
stat:=Memo1.Lines.Strings[Memo1.Lines.Count-1];
Form_login.ADOQuery_copy.Close;
Form_login.ADOQuery_copy.SQL.Clear;
Form_login.ADOQuery_copy.SQL.Add('copy_new :d1,:id_lic,:status, :p1');
td:=Now;
Form_login.ADOQuery_copy.Parameters.ParamByName('d1').Value:=DateTimeToStr(td);
Form_login.ADOQuery_copy.Parameters.ParamByName('ID_lic').Value:=StrToInt(Form_GL.id_lic.Text);
Form_login.ADOQuery_copy.Parameters.ParamByName('status').Value:=stat;
Form_login.ADOQuery_copy.Parameters.ParamByName('p1').Value:=edit4.Text;
Form_login.ADOQuery_copy.Open
end;
procedure TForm_copy.Button2Click(Sender: TObject);
begin
if processExists('transport.exe') then
ShowMessage('process is running')
else
ShowMessage('process not running');
end;
procedure TForm_copy.FormCreate(Sender: TObject);
begin
end;
end.
unit Unit_lic_spr;
interface
uses
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.DBCtrls, Vcl.Menus,
Vcl.StdCtrls, Vcl.Buttons, Vcl.Grids, Vcl.DBGrids, Vcl.ExtCtrls;
type
TForm_lic_spr = class(TForm)
GroupBox2: TGroupBox;
LabeledEdit3: TLabeledEdit;
GroupBox1: TGroupBox;
Label1: TLabel;
Label2: TLabel;
DBGrid1: TDBGrid;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
BitBtn3: TBitBtn;
BitBtn4: TBitBtn;
LabeledEdit1: TLabeledEdit;
LabeledEdit2: TLabeledEdit;
Edit1: TEdit;
PopupMenu1: TPopupMenu;
N1: TMenuItem;
N2: TMenuItem;
DBLookupComboBox1: TDBLookupComboBox;
DBLookupComboBox2: TDBLookupComboBox;
procedure N1Click(Sender: TObject);
procedure N2Click(Sender: TObject);
procedure LabeledEdit3Change(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
private
{ Private declarations }
```

## Продолжение приложения А

```
public
  { Public declarations }
end;
var
  Form_lic_spr: TForm_lic_spr;
implementation
  {$R *.dfm}
uses Unit_login;
procedure TForm_lic_spr.BitBtn1Click(Sender: TObject);
begin
  BitBtn1.Visible:=false;
  BitBtn2.Visible:=False;
  Form_login.ADOQuery_lic_spr_r.Close;
  Form_login.ADOQuery_lic_spr_r.SQL.Clear;
  Form_login.ADOQuery_lic_spr_r.SQL.Add('INSERT INTO lic (fio, pas, id_otd, spec)');
  Form_login.ADOQuery_lic_spr_r.SQL.Add('VALUES (:fio, :pas, :ido, :spec)');
  Form_login.ADOQuery_lic_spr_r.Parameters.ParamByName('fio').Value:=LabeledEdit1.Text;
  Form_login.ADOQuery_lic_spr_r.Parameters.ParamByName('pas').Value:=LabeledEdit2.Text;
  Form_login.ADOQuery_lic_spr_r.Parameters.ParamByName('IDO').Value:=DBLookupComboBox1.KeyValue;
  Form_login.ADOQuery_lic_spr_r.Parameters.ParamByName('spec').Value:=DBLookupComboBox2.KeyValue;
  Form_login.ADOQuery_lic_spr_r.ExecSQL;
  ShowMessage('Добавление выполнено. Таблица обновлена');
  LabeledEdit1.Text:="";
  LabeledEdit2.Text:="";
  Edit1.Text:="";
  DBLookupComboBox1.KeyValue:=0;
  DBLookupComboBox2.KeyValue:=0;
  Form_login.ADOQuery_lic_spr_r.Close;
  Form_login.ADOQuery_lic_spr_r.SQL.Clear;
  Form_login.ADOQuery_lic_spr_r.SQL.Add('SELECT lic.id, lic.fio, profil.name_p, lic.pas, otdel.name_otd, profil.id,
otdel.id FROM lic , otdel, profil WHERE id_otd=otdel.id and profil.id=lic.spec');
  Form_login.ADOQuery_lic_spr_r.Open;
end;
procedure TForm_lic_spr.BitBtn2Click(Sender: TObject);
begin
  BitBtn3.Visible:=false;
  BitBtn4.Visible:=False;
  ShowMessage('Добавление отменено!');
  LabeledEdit1.Text:="";
  LabeledEdit2.Text:="";
  DBLookupComboBox1.KeyValue:=0;
  DBLookupComboBox2.KeyValue:=0;
  Edit1.Text:="";
end;
procedure TForm_lic_spr.BitBtn3Click(Sender: TObject);
begin
  BitBtn3.Visible:=false;
  BitBtn4.Visible:=False;
  Form_login.ADOQuery_lic_spr_r.Close;
  Form_login.ADOQuery_lic_spr_r.SQL.Clear;
  Form_login.ADOQuery_lic_spr_r.SQL.Add('UPDATE lic SET fio=:fio, pas=:pas, id_otd=:IDO, spec=:spec where
ID=:IDI');
  Form_login.ADOQuery_lic_spr_r.Parameters.ParamByName('IDI').Value:=StrToInt(Edit1.text);
  Form_login.ADOQuery_lic_spr_r.Parameters.ParamByName('fio').Value:=LabeledEdit1.Text;
  Form_login.ADOQuery_lic_spr_r.Parameters.ParamByName('pas').Value:=LabeledEdit2.Text;
  Form_login.ADOQuery_lic_spr_r.Parameters.ParamByName('IDO').Value:=DBLookupComboBox1.KeyValue;
  Form_login.ADOQuery_lic_spr_r.Parameters.ParamByName('spec').Value:=DBLookupComboBox2.KeyValue;
  Form_login.ADOQuery_lic_spr_r.ExecSQL;
  ShowMessage('Редактирование выполнено. Таблица обновлена');
```

## Продолжение приложения А

```
LabeledEdit1.Text:="";
LabeledEdit2.Text:="";
Edit1.Text:="";
DBLookupComboBox1.KeyValue:=0;
DBLookupComboBox2.KeyValue:=0;
Form_login.ADOQuery_lic_spr.Close;
Form_login.ADOQuery_lic_spr.SQL.Clear;
Form_login.ADOQuery_lic_spr.SQL.Add('SELECT lic.id, lic.fio, profil.name_p, lic.pas, otdel.name_otd, profil.id,
otdel.id FROM lic , otdel, profil WHERE id_otd=otdel.id and profil.id=lic.spec');
Form_login.ADOQuery_lic_spr.Open;
end;
procedure TForm_lic_spr.BitBtn4Click(Sender: TObject);
begin
BitBtn3.Visible:=false;
BitBtn4.Visible:=False;
ShowMessage('Редактирование отменено!');
LabeledEdit1.Text:="";
LabeledEdit2.Text:="";
DBLookupComboBox1.KeyValue:=0;
DBLookupComboBox2.KeyValue:=0;
Edit1.Text:="";
end;
procedure TForm_lic_spr.LabeledEdit3Change(Sender: TObject);
begin
Form_login.ADOQuery_lic_spr.Close;
Form_login.ADOQuery_lic_spr.SQL.Clear;
Form_login.ADOQuery_lic_spr.SQL.Add('SELECT lic.id, lic.fio, profil.name_p, lic.pas, otdel.name_otd, profil.id,
otdel.id FROM lic , otdel, profil WHERE id_otd=otdel.id and profil.id=lic.spec and lic.fio like ' + #39 + '%' +
LabeledEdit3.Text + '%' + #39' ');
Form_login.ADOQuery_lic_spr.Open;
end;
procedure TForm_lic_spr.N1Click(Sender: TObject);
begin
BitBtn3.Visible:=false;
BitBtn4.Visible:=False;
BitBtn1.Visible:=True;
BitBtn2.Visible:=True;
end;
procedure TForm_lic_spr.N2Click(Sender: TObject);
begin
BitBtn1.Visible:=false;
BitBtn2.Visible:=False;
BitBtn3.Visible:=True;
BitBtn4.Visible:=True;
Form_lic_spr.LabeledEdit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('fio').AsString;
Form_lic_spr.LabeledEdit2.Text:=DBGrid1.DataSource.DataSet.fieldbyname('pas').AsString;
Form_lic_spr.DBLookupComboBox1.KeyValue:=DBGrid1.DataSource.DataSet.fieldbyname('id_2').AsString;
Form_lic_spr.Edit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id').AsString;
Form_lic_spr.DBLookupComboBox2.KeyValue:=DBGrid1.DataSource.DataSet.fieldbyname('id_1').AsString;
end;
end.
unit Unit_otdel_spr;
interface
uses
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.Grids, Vcl.DBGrids,
Vcl.StdCtrls, Vcl.ExtCtrls;
type
TForm_otdel_spr = class(TForm)
```

## Продолжение приложения А

```
GroupBox2: TGroupBox;
  LabeledEdit3: TLabeledEdit;
  GroupBox1: TGroupBox;
  DBGrid1: TDBGrid;
  procedure LabeledEdit3Change(Sender: TObject);
  procedure GroupBox2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form_otdel_spr: TForm_otdel_spr;
implementation
{$R *.dfm}
uses Unit_login;
procedure TForm_otdel_spr.GroupBox2Click(Sender: TObject);
begin
end;
procedure TForm_otdel_spr.LabeledEdit3Change(Sender: TObject);
begin
  Form_login.ADOQuery_otdel.Close;
  Form_login.ADOQuery_otdel.SQL.Clear;
  Form_login.ADOQuery_otdel.SQL.Add('SELECT * from otdel WHERE name_otd like ' + #39 + '%' +
  LabeledEdit3.Text + '%' + #39' ');
  Form_login.ADOQuery_otdel.Open;
end;
end.
unit Unit_otdel;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.StdCtrls, Vcl.ExtCtrls,
  Vcl.Buttons, Vcl.Grids, Vcl.DBGrids, Vcl.Menus;
type
  TForm_otdel = class(TForm)
    GroupBox1: TGroupBox;
    DBGrid1: TDBGrid;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    LabeledEdit1: TLabeledEdit;
    LabeledEdit2: TLabeledEdit;
    Edit1: TEdit;
    GroupBox2: TGroupBox;
    LabeledEdit3: TLabeledEdit;
    PopupMenu1: TPopupMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    procedure LabeledEdit3Change(Sender: TObject);
    procedure N2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure N1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
private
  { Private declarations }
```

## Продолжение приложения А

```
public
  { Public declarations }
end;
var
  Form_otdel: TForm_otdel;
implementation
  {$R *.dfm}
uses Unit_login;
procedure TForm_otdel.BitBtn1Click(Sender: TObject);
begin
  BitBtn1.Visible:=false;
  BitBtn2.Visible:=False;
  Form_login.ADOQuery_otdel_r.Close;
  Form_login.ADOQuery_otdel_r.SQL.Clear;
  Form_login.ADOQuery_otdel_r.SQL.Add('INSERT INTO otdel (name_otd, tel)');
  Form_login.ADOQuery_otdel_r.SQL.Add('VALUES (:name, :tel)');
  Form_login.ADOQuery_otdel_r.Parameters.ParamByName('name').Value:=LabeledEdit1.Text;
  Form_login.ADOQuery_otdel_r.Parameters.ParamByName('tel').Value:=LabeledEdit2.Text;
  Form_login.ADOQuery_otdel_r.ExecSQL;
  Form_login.ADOQuery_otdel.Close;
  Form_login.ADOQuery_otdel.Open;
end;
procedure TForm_otdel.BitBtn2Click(Sender: TObject);
begin
  BitBtn1.Visible:=false;
  BitBtn2.Visible:=False;
  ShowMessage('Добавление отменено!');
  LabeledEdit1.Text:="";
  LabeledEdit2.Text:="";
  Edit1.Text:="";
end;
procedure TForm_otdel.BitBtn3Click(Sender: TObject);
begin
  BitBtn3.Visible:=false;
  BitBtn4.Visible:=False;
  Form_login.ADOQuery_otdel_r.Close;
  Form_login.ADOQuery_otdel_r.SQL.Clear;
  Form_login.ADOQuery_otdel_r.SQL.Add('UPDATE otdel SET name_otd=:name_otd, tel=:tel where ID=:IDI');
  Form_login.ADOQuery_otdel_r.Parameters.ParamByName('IDI').Value:=StrToInt(Edit1.text);
  Form_login.ADOQuery_otdel_r.Parameters.ParamByName('name_otd').Value:=LabeledEdit1.Text;
  Form_login.ADOQuery_otdel_r.Parameters.ParamByName('tel').Value:=LabeledEdit2.Text;
  Form_login.ADOQuery_otdel_r.ExecSQL;
  ShowMessage('Редактирование выполнено. Таблица обновлена');
  LabeledEdit1.Text:="";
  LabeledEdit2.Text:="";
  Edit1.Text:="";
  Form_login.ADOQuery_otdel.Close;
  Form_login.ADOQuery_otdel.SQL.Clear;
  Form_login.ADOQuery_otdel.SQL.Add('select * from otdel');
  Form_login.ADOQuery_otdel.Open;
end;
procedure TForm_otdel.BitBtn4Click(Sender: TObject);
begin
  BitBtn3.Visible:=false;
  BitBtn4.Visible:=False;
  ShowMessage('Редактирование отменено!');
  LabeledEdit1.Text:="";
  LabeledEdit2.Text:="";
  Edit1.Text:="";
```

## Продолжение приложения А

```
end;
procedure TForm_otdel.LabeledEdit3Change(Sender: TObject);
begin
Form_login.ADOQuery_otdel.Close;
Form_login.ADOQuery_otdel.SQL.Clear;
Form_login.ADOQuery_otdel.SQL.Add('SELECT * from otdel WHERE name_otd like ' + #39 + '%' +
LabeledEdit3.Text + '%' + #39' ');
Form_login.ADOQuery_otdel.Open;
end;
procedure TForm_otdel.N1Click(Sender: TObject);
begin
BitBtn3.Visible:=false;
BitBtn4.Visible:=False;
BitBtn1.Visible:=True;
BitBtn2.Visible:=True;
end;
procedure TForm_otdel.N2Click(Sender: TObject);
begin
BitBtn1.Visible:=false;
BitBtn2.Visible:=False;
BitBtn3.Visible:=True;
BitBtn4.Visible:=True;
Form_otdel.LabeledEdit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('name_otd').AsString;
Form_otdel.LabeledEdit2.Text:=DBGrid1.DataSource.DataSet.fieldbyname('tel').AsString;
Form_otdel.Edit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id').AsString;
end;
end.
unit Unit_img;
interface
uses
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls;
type
TForm_img = class(TForm)
Image1: TImage;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Image1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form_img: TForm_img;
implementation
{$R *.dfm}
procedure TForm_img.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Form_img.Caption:='Парк техники:';
end;
procedure TForm_img.Image1Click(Sender: TObject);
begin
end;
end.
unit Unit_park;
interface
uses
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.Grids, Vcl.DBGrids,
```

## Продолжение приложения А

```
Vcl.StdCtrls, Vcl.ExtCtrls, Vcl.Buttons, Vcl.Menus;
type
  TForm_spr_park = class(TForm)
    GroupBox1: TGroupBox;
    DBGrid1: TDBGrid;
    PopupMenu1: TPopupMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    LabeledEdit1: TLabeledEdit;
    LabeledEdit2: TLabeledEdit;
    ComboBox1: TComboBox;
    Edit1: TEdit;
    OpenFileDialog1: TOpenDialog;
    BitBtn5: TBitBtn;
    Edit2: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    GroupBox2: TGroupBox;
    LabeledEdit3: TLabeledEdit;
    procedure N2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure N1Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn5Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure LabeledEdit3Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form_spr_park: TForm_spr_park;
implementation
{$R *.dfm}
uses Unit_login;
procedure TForm_spr_park.BitBtn1Click(Sender: TObject);
begin
  BitBtn1.Visible:=false;
  BitBtn2.Visible:=False;
  Form_login.ADOQuery_park_r.Close;
  Form_login.ADOQuery_park_r.SQL.Clear;
  Form_login.ADOQuery_park_r.SQL.Add('INSERT INTO park (marka, gos_n, pr, img)');
  Form_login.ADOQuery_park_r.SQL.Add('VALUES (:marka, :gos_n, :pr)');
  Form_login.ADOQuery_park_r.Parameters.ParamByName('marka').Value:=LabeledEdit1.Text;
  Form_login.ADOQuery_park_r.Parameters.ParamByName('gos_n').Value:=LabeledEdit2.Text;
  Form_login.ADOQuery_park_r.Parameters.ParamByName('pr').Value:=ComboBox1.Text;
  Form_login.ADOQuery_park_r.Parameters.ParamByName('img').Value:=Edit2.Text;
  Form_login.ADOQuery_park_r.ExecSQL;
  Form_login.ADOQuery_park.Close;
  Form_login.ADOQuery_park.Open;
  ShowMessage('Добавление выполнено. Таблица обновлена');
  LabeledEdit1.Text:=;
```



## Продолжение приложения А

```
LabeledEdit2.Text:="";
ComboBox1.Text:="";
Edit1.Text:="";
Edit2.Text:="";
Form_login.ADOQuery_park_r.Close;
Form_login.ADOQuery_park_r.SQL.Clear;
Form_login.ADOQuery_park_r.SQL.Add('select * from park');
Form_login.ADOQuery_park_r.Open;
end;
procedure TForm_spr_park.BitBtn2Click(Sender: TObject);
begin
  BitBtn1.Visible:=false;
  BitBtn2.Visible:=False;
  ShowMessage('Добавление отменено!');
  LabeledEdit1.Text:="";
  LabeledEdit2.Text:="";
  ComboBox1.Text:="";
  Edit1.Text:="";
  Edit2.Text:="";
end;
procedure TForm_spr_park.BitBtn3Click(Sender: TObject);
begin
  BitBtn3.Visible:=false;
  BitBtn4.Visible:=False;
  Form_login.ADOQuery_park_r.Close;
  Form_login.ADOQuery_park_r.SQL.Clear;
  Form_login.ADOQuery_park_r.SQL.Add('UPDATE park SET marka=:marka, gos_n=:gos_n, pr=:pr, img=:img where ID=:IDI');
  Form_login.ADOQuery_park_r.Parameters.ParamByName('IDI').Value:=StrToInt(Edit1.text);
  Form_login.ADOQuery_park_r.Parameters.ParamByName('marka').Value:=LabeledEdit1.Text;
  Form_login.ADOQuery_park_r.Parameters.ParamByName('gos_n').Value:=LabeledEdit2.Text;
  Form_login.ADOQuery_park_r.Parameters.ParamByName('pr').Value:=ComboBox1.Text;
  Form_login.ADOQuery_park_r.Parameters.ParamByName('img').Value:=Edit2.Text;
  Form_login.ADOQuery_park_r.ExecSQL;
  ShowMessage('Редактирование выполнено. Таблица обновлена');
  LabeledEdit1.Text:="";
  LabeledEdit2.Text:="";
  ComboBox1.Text:="";
  Edit1.Text:="";
  Edit2.Text:="";
  Form_login.ADOQuery_park_r.Close;
  Form_login.ADOQuery_park_r.SQL.Clear;
  Form_login.ADOQuery_park_r.SQL.Add('select * from park');
  Form_login.ADOQuery_park_r.Open;
  Form_login.ADOQuery_park.Close;
  Form_login.ADOQuery_park.Open;
end;
procedure TForm_spr_park.BitBtn4Click(Sender: TObject);
begin
  BitBtn3.Visible:=false;
  BitBtn4.Visible:=False;
  ShowMessage('Редактирование отменено!');
  LabeledEdit1.Text:="";
  LabeledEdit2.Text:="";
  ComboBox1.Text:="";
  Edit1.Text:="";
  Edit2.Text:="";
end;
procedure TForm_spr_park.BitBtn5Click(Sender: TObject);
```

## Продолжение приложения А

```
begin
OpenDialog1.Execute;
edit2.Text:=openDialog1.FileName ;
end;
procedure TForm_spr_park.Button2Click(Sender: TObject);
begin
LabeledEdit3.Text:="";
Form_login.ADOQuery_park_r.Close;
Form_login.ADOQuery_park_r.SQL.Clear;
Form_login.ADOQuery_park_r.SQL.Add('select * from park');
Form_login.ADOQuery_park_r.Open;
end;
procedure TForm_spr_park.LabeledEdit3Change(Sender: TObject);
begin
Form_login.ADOQuery_park_r.Close;
Form_login.ADOQuery_park_r.SQL.Clear;
Form_login.ADOQuery_park_r.SQL.Add('SELECT * from park WHERE marka like ' + #39 + '%' +
LabeledEdit3.Text + '%' + #39' ');
Form_login.ADOQuery_park_r.Open;
end;
procedure TForm_spr_park.N1Click(Sender: TObject);
begin
BitBtn3.Visible:=false;
BitBtn4.Visible:=False;
BitBtn1.Visible:=True;
BitBtn2.Visible:=True;
end;
procedure TForm_spr_park.N2Click(Sender: TObject);
begin
BitBtn1.Visible:=false;
BitBtn2.Visible:=False;
BitBtn3.Visible:=True;
BitBtn4.Visible:=True;
Form_spr_park.LabeledEdit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('marka').AsString;
Form_spr_park.LabeledEdit2.Text:=DBGrid1.DataSource.DataSet.fieldbyname('gos_n').AsString;
Form_spr_park.ComboBox1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('pr').AsString;
Form_spr_park.Edit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id').AsString;
Form_spr_park.Edit2.Text:=DBGrid1.DataSource.DataSet.fieldbyname('img').AsString;
end;
end.
unit Unit_GL;
interface
uses
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ComCtrls, Vcl.ExtCtrls, Data.DB,
Vcl.Menus, Vcl.Grids, Vcl.DBGrids, Vcl.StdCtrls, Vcl.Buttons, Vcl.DBCtrls,
Vcl.Mask, IniFiles, ShellApi;
type
TForm_GL = class(TForm)
StatusBar1: TStatusBar;
Timer1: TTimer;
DBGrid1: TDBGrid;
PopupMenu1: TPopupMenu;
N1: TMenuItem;
N2: TMenuItem;
N3: TMenuItem;
MainMenu1: TMainMenu;
N4: TMenuItem;
N5: TMenuItem;
```

## Продолжение приложения А

N6: TMenuItem;  
N7: TMenuItem;  
N8: TMenuItem;  
N11: TMenuItem;  
N12: TMenuItem;  
Panel1: TPanel;  
N9: TMenuItem;  
N10: TMenuItem;  
N13: TMenuItem;  
N14: TMenuItem;  
GroupBox1: TGroupBox;  
spec: TEdit;  
id\_ot: TEdit;  
id\_lic: TEdit;  
id\_lic\_del: TEdit;  
id\_z1\_del: TEdit;  
status: TEdit;  
LabeledEdit1: TLabeledEdit;  
LabeledEdit2: TLabeledEdit;  
LabeledEdit3: TLabeledEdit;  
StatusBar2: TStatusBar;  
ScrollBar1: TScrollBar;  
GroupBox7: TGroupBox;  
DBGrid2: TDBGrid;  
GroupBox8: TGroupBox;  
Edit4: TEdit;  
Edit5: TEdit;  
Edit6: TEdit;  
GroupBox2: TGroupBox;  
BitBtn1: TBitBtn;  
BitBtn2: TBitBtn;  
GroupBox9: TGroupBox;  
Label7: TLabel;  
Label8: TLabel;  
Edit9: TEdit;  
DateTimePicker3: TDateTimePicker;  
ComboBox1: TComboBox;  
BitBtn5: TBitBtn;  
BitBtn6: TBitBtn;  
GroupBox5: TGroupBox;  
Label2: TLabel;  
Label3: TLabel;  
Label1: TLabel;  
Label5: TLabel;  
DateTimePicker2: TDateTimePicker;  
Edit2: TEdit;  
MaskEdit1: TMaskEdit;  
Edit1: TEdit;  
DateTimePicker1: TDateTimePicker;  
MaskEdit2: TMaskEdit;  
GroupBox4: TGroupBox;  
Label4: TLabel;  
Label6: TLabel;  
Label9: TLabel;  
DBLookupComboBox1: TDBLookupComboBox;  
Edit3: TEdit;  
Edit10: TEdit;  
GroupBox6: TGroupBox;  
Label10: TLabel;

## Продолжение приложения А

```
Label11: TLabel;
Edit7: TEdit;
Edit8: TEdit;
BitBtn3: TBitBtn;
BitBtn4: TBitBtn;
BitBtn8: TBitBtn;
BitBtn7: TBitBtn;
Button1: TButton;
Edit_img: TEdit;
Edit_park: TEdit;
Edit_g: TEdit;
Edit11: TEdit;
N18: TMenuItem;
N19: TMenuItem;
Edit12: TEdit;
ColorDialog1: TColorDialog;
N17: TMenuItem;
N20: TMenuItem;
N21: TMenuItem;
N22: TMenuItem;
N23: TMenuItem;
N24: TMenuItem;
Edit13: TEdit;
Memo1: TMemo;
N26: TMenuItem;
N27: TMenuItem;
N28: TMenuItem;
N29: TMenuItem;
N30: TMenuItem;
N31: TMenuItem;
N15: TMenuItem;
N32: TMenuItem;
N16: TMenuItem;
N33: TMenuItem;
N41: TMenuItem;
procedure Timer1Timer(Sender: TObject);
procedure DateTimePicker1Change(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure BitBtn1Click(Sender: TObject);
procedure N1Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure DateTimePicker2Change(Sender: TObject);
procedure N3Click(Sender: TObject);
procedure N2Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure DBGrid1DrawColumnCell(Sender: TObject; const Rect: TRect;
  DataCol: Integer; Column: TColumn; State: TGridDrawState);
procedure N5Click(Sender: TObject);
procedure N7Click(Sender: TObject);
procedure N11Click(Sender: TObject);
procedure DBGrid1CellClick(Column: TColumn);
procedure DateTimePicker3Change(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure N12Click(Sender: TObject);
procedure N13Click(Sender: TObject);
```

## Продолжение приложения А

```
procedure N14Click(Sender: TObject);
  procedure ScrollBox1MouseWheelDown(Sender: TObject; Shift: TShiftState;
    MousePos: TPoint; var Handled: Boolean);
  procedure ScrollBox1MouseWheelUp(Sender: TObject; Shift: TShiftState;
    MousePos: TPoint; var Handled: Boolean);
  procedure DBLookupComboBox1Enter(Sender: TObject);
  procedure N17Click(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure DBLookupComboBox1KeyPress(Sender: TObject; var Key: Char);
  procedure Edit9Change(Sender: TObject);
  procedure N18Click(Sender: TObject);
  procedure DateTimePicker2Enter(Sender: TObject);
  procedure N19Click(Sender: TObject);
  procedure DateTimePicker3Enter(Sender: TObject);
  procedure Edit6DbClick(Sender: TObject);
  procedure Edit5DbClick(Sender: TObject);
  procedure Edit4DbClick(Sender: TObject);
  procedure Edit12DbClick(Sender: TObject);
  procedure N20Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure N22Click(Sender: TObject);
  procedure N23Click(Sender: TObject);
  procedure BitBtn9Click(Sender: TObject);
  procedure DateTimePicker1Enter(Sender: TObject);
  procedure N28Click(Sender: TObject);
  procedure N30Click(Sender: TObject);
  procedure N31Click(Sender: TObject);
  procedure N15Click(Sender: TObject);
  procedure StatusBar1DrawPanel(StatusBar: TStatusBar; Panel: TStatusPanel;
    const Rect: TRect);
  procedure N32Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure N16Click(Sender: TObject);
  procedure N33Click(Sender: TObject);
  procedure N41Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form_GL: TForm_GL;
  inifile: TIniFile;
  w: integer;
  z: string;
implementation
{$R *.dfm}
uses Unit_login, Unit_f1, Unit_f2, Unit_pol, Unit_treb, Unit_park, Unit_img,
  Unit_otdel, Unit_otdel_spr, Unit_lic_spr, Unit_f3, Unit_copy, Unit_f4,
  Unit_f5;
procedure TForm_GL.BitBtn1Click(Sender: TObject);
var
  i, j: Integer;
  s: string;
begin
  if spec.Text='1' then
begin ShowMessage('Недостаточно прав!');
end else
if spec.Text='2' then
```

## Продолжение приложения А

```
begin
  if DBGrid1.SelectedRows.Count>0 then
    with DBGrid1.DataSource.DataSet do
      for i:=0 to DBGrid1.SelectedRows.Count-1 do
        begin
          GotoBookmark(DBGrid1.SelectedRows.Items[i]);
          Form_GL.id_z1_del.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id').AsString;
          Form_login.ADOQuery_ADD_z.Close;
          Form_login.ADOQuery_ADD_z.SQL.Clear;
          Form_login.ADOQuery_ADD_z.SQL.Add('update_z1 :STATUS, :IDI');
          Form_login.ADOQuery_ADD_z.Parameters.ParamByName('IDI').Value:=StrToInt(id_z1_del.text);
          Form_login.ADOQuery_ADD_z.Parameters.ParamByName('STATUS').Value:='На утверждении';
          Form_login.ADOQuery_ADD_z.ExecSQL;
        end;
        ShowMessage('Заявка утверждена. Таблица обновлена');
        Form_login.ADOQuery_FULL.Active:=false;
        Form_login.ADOQuery_FULL.Active:=true;
      end;
    end;
  procedure TForm_GL.BitBtn2Click(Sender: TObject);
  var
    i, j: Integer;
    s: string;
  begin
    if spec.Text='1' then
      begin ShowMessage('Недостаточно прав!');
      end else
      if spec.Text='2' then
        begin
          if DBGrid1.SelectedRows.Count>0 then
            with DBGrid1.DataSource.DataSet do
              for i:=0 to DBGrid1.SelectedRows.Count-1 do
                begin
                  GotoBookmark(DBGrid1.SelectedRows.Items[i]);
                  Form_GL.id_z1_del.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id').AsString;
                  Form_login.ADOQuery_ADD_z.Close;
                  Form_login.ADOQuery_ADD_z.SQL.Clear;
                  Form_login.ADOQuery_ADD_z.SQL.Add('update_z1 :STATUS, :IDI');
                  Form_login.ADOQuery_ADD_z.Parameters.ParamByName('IDI').Value:=StrToInt(id_z1_del.text);
                  Form_login.ADOQuery_ADD_z.Parameters.ParamByName('STATUS').Value:='В работе';
                  Form_login.ADOQuery_ADD_z.ExecSQL;
                end;
                ShowMessage('Заявка возвращена. Таблица обновлена');
                Form_login.ADOQuery_FULL.Active:=false;
                Form_login.ADOQuery_FULL.Active:=true;
              end;
            end;
          procedure TForm_GL.BitBtn3Click(Sender: TObject);
          var
            Data: TDate;
            td:TDateTime;
            i:integer;
            s:string;
          begin
            Memo1.Lines.Add(edit2.Text);
            while Edit13.text<>edit1.text do
              begin
                edit1.text:=DateToStr(DateTimePicker1.Date);
                data:=StrToDate(Edit13.text)+1;
```

## Продолжение приложения А

```
Memo1.Lines.Add(datetostr(data));
Edit13.Text:=DateToStr(data);
end;
if (Memo1.Lines.Count >0) then
begin
for i:=0 to Memo1.Lines.Count-1 do
begin
s:=Memo1.Lines.Strings[i];
td:=now;
BitBtn3.Visible:=false;
BitBtn8.Visible:=false;
Form_login.ADOQuery_ADD_z.Close;
Form_login.ADOQuery_ADD_z.SQL.Clear;
Form_login.ADOQuery_ADD_z.SQL.Add(' insert_z1 :dtz, :ido, :idl, :idp, :d1, :t1, :pr, :st, :t2, :d2, :cel, :m1, :km');
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('dtz').Value:=StrToDateTime(Form_GL.StatusBar1.Panels
[7].Text);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('ido').Value:=StrToInt(id_ot.text);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('idl').Value:=StrToInt(id_lic.text);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('idp').Value:=DBLookupComboBox1.KeyValue;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('d1').Value:=StrToDate(s);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('t1').Value:=MaskEdit1.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('pr').Value:=edit7.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('st').Value:='B работе';
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('t2').Value:=MaskEdit2.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('d2').Value:=StrToDate(s);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('cel').Value:=Edit8.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('m1').Value:=Edit3.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('km').Value:=StrToInt(Edit10.Text);
Form_login.ADOQuery_ADD_z.ExecSQL;
end;
memo1.Clear;
end;
Edit1.Text:="";
Edit2.Text:="";
Edit3.Text:="";
Edit7.Text:="";
Edit8.Text:="";
Edit10.Text:="";
MaskEdit1.Text:="";
MaskEdit2.Text:="";
DBLookupComboBox1.KeyValue:=0;
Form_login.ADOQuery_FULL.Close;
Form_login.ADOQuery_FULL.SQL.Clear;
Form_login.ADOQuery_FULL.SQL.Add('obsh_v :d');
Form_login.ADOQuery_FULL.Parameters.ParamByName('d').Value:=DateToStr(td);
Form_login.ADOQuery_FULL.Open;
ShowMessage('Добавлено выполнено. Таблица обновлена');
Form_login.ADOQuery_sum.Close;
Form_login.ADOQuery_sum.SQL.Clear;
Form_login.ADOQuery_sum.SQL.Add('sum :d1');
Form_login.ADOQuery_sum.Parameters.ParamByName('d1').Value:=DateToStr(td);
Form_login.ADOQuery_sum.Open;
GroupBox7.Caption:='Сводная информация';
GroupBox7.Caption:=GroupBox7.Caption+' '+IntToStr(Form_login.ADOQuery_full.RecordCount);
end;
procedure TForm_GL.BitBtn4Click(Sender: TObject);
var
td:Tdatetime;
begin
```

## Продолжение приложения А

```
td:=now;
BitBtn4.Visible:=false;
BitBtn7.Visible:=false;
Form_login.ADOQuery_ADD_z.Close;
Form_login.ADOQuery_ADD_z.SQL.Clear;
Form_login.ADOQuery_ADD_z.SQL.Add('update_z1_full :dtz, :idp, :d1, :t1, :pr, :t2, :d2, :cel, :m1, :km, :sti, :IDI');
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('dtz').Value:=StrToDateTime(Form_GL.StatusBar1.Panels
[7].Text);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('idp').Value:=DBLookupComboBox1.KeyValue;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('d1').Value:=Edit2.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('t1').Value:=MaskEdit1.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('pr').Value:=Edit7.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('t2').Value:=MaskEdit2.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('d2').Value:=StrToDate(Edit1.Text);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('cel').Value:=Edit8.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('m1').Value:=Edit3.Text;
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('km').Value:=StrToInt(Edit10.Text);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('sti').Value:='В работе';
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('IDI').Value:=StrToInt(id_z1_del.Text);
Form_login.ADOQuery_ADD_z.ExecSQL;
ShowMessage('Редактирование выполнено. Таблица обновлена');
id_lic_del.Text:='';
id_z1_del.Text:='';
Edit1.Text:='';
Edit2.Text:='';
Edit3.Text:='';
Edit7.Text:='';
Edit8.Text:='';
MaskEdit1.Text:='';
MaskEdit2.Text:='';
Edit10.Text:='';
LabeledEdit1.Text:='';
LabeledEdit2.Text:='';
LabeledEdit3.Text:='';
DBLookupComboBox1.KeyValue:=0;
Form_login.ADOQuery_FULL.Close;
Form_login.ADOQuery_FULL.SQL.Clear;
Form_login.ADOQuery_FULL.SQL.Add('obsh_v :d1, :d2');
Form_login.ADOQuery_FULL.Parameters.ParamByName('d1').Value:=DateToStr(td);
Form_login.ADOQuery_FULL.Parameters.ParamByName('d2').Value:=StrToDate('31.12.2099');
Form_login.ADOQuery_FULL.Open;
Form_login.ADOQuery_sum.Close;
Form_login.ADOQuery_sum.SQL.Clear;
Form_login.ADOQuery_sum.SQL.Add('sum :d1, :d2');
Form_login.ADOQuery_sum.Parameters.ParamByName('d1').Value:=DateToStr(td);
Form_login.ADOQuery_sum.Parameters.ParamByName('d2').Value:=StrToDate('31.12.2099');
Form_login.ADOQuery_sum.Open;
GroupBox7.Caption:='Сводная информация';
GroupBox7.Caption:=GroupBox7.Caption+' '+IntToStr(Form_login.ADOQuery_full.RecordCount);
end;
procedure TForm_GL.BitBtn5Click(Sender: TObject);
begin
GroupBox7.Caption:='Сводная информация';
if ComboBox1.Text<>'' then
begin
Form_login.ADOQuery_FULL.Close;
Form_login.ADOQuery_FULL.SQL.Clear;
Form_login.ADOQuery_FULL.SQL.Add('obsh_v_st :d1, :status');
Form_login.ADOQuery_FULL.Parameters.ParamByName('d1').Value:=strtodate(Edit9.Text);
```



## Продолжение приложения А

```
Form_login.ADOQuery_FULL.Parameters.ParamByName('status').Value:=ComboBox1.Text;
Form_login.ADOQuery_FULL.Open;
GroupBox7.Caption:='Сводная информация';
GroupBox7.Caption:=GroupBox7.Caption+' '+IntToStr(Form_login.ADOQuery_full.RecordCount);
Form_login.ADOQuery_sum.Close;
Form_login.ADOQuery_sum.SQL.Clear;
Form_login.ADOQuery_sum.SQL.Add('sum_st :d1, :status');
Form_login.ADOQuery_sum.Parameters.ParamByName('d1').Value:=strtodate(Edit9.Text);
Form_login.ADOQuery_sum.Parameters.ParamByName('status').Value:=ComboBox1.Text;
Form_login.ADOQuery_sum.Open;
end
else
if ComboBox1.Text="" then
begin
Form_login.ADOQuery_FULL.Close;
Form_login.ADOQuery_FULL.SQL.Clear;
Form_login.ADOQuery_FULL.SQL.Add('obsh_v_d :d1');
Form_login.ADOQuery_FULL.Parameters.ParamByName('d1').Value:=strtodate(Edit9.Text);
Form_login.ADOQuery_FULL.Open;
GroupBox7.Caption:='Сводная информация';
GroupBox7.Caption:=GroupBox7.Caption+' '+IntToStr(Form_login.ADOQuery_full.RecordCount);
Form_login.ADOQuery_sum.Close;
Form_login.ADOQuery_sum.SQL.Clear;
Form_login.ADOQuery_sum.SQL.Add('sum_d :d1');
Form_login.ADOQuery_sum.Parameters.ParamByName('d1').Value:=strtodate(Edit9.Text);
Form_login.ADOQuery_sum.Open;
end;
end;
procedure TForm_GL.BitBtn6Click(Sender: TObject);
var
td:TDateTime;
begin
GroupBox7.Caption:='Сводная информация';
td:=now;
Edit9.Text:="";
ComboBox1.Text:="";
Form_login.ADOQuery_FULL.Close;
Form_login.ADOQuery_FULL.SQL.Clear;
Form_login.ADOQuery_FULL.SQL.Add('obsh_v :d1, :d2');
Form_login.ADOQuery_FULL.Parameters.ParamByName('d1').Value:=DateToStr(td);
Form_login.ADOQuery_full.Parameters.ParamByName('d2').Value:=StrToDate('31.12.2099');
Form_login.ADOQuery_FULL.Open;
Form_login.ADOQuery_sum.Close;
Form_login.ADOQuery_sum.SQL.Clear;
Form_login.ADOQuery_sum.SQL.Add('sum :d1, :d2');
Form_login.ADOQuery_sum.Parameters.ParamByName('d1').Value:=DateToStr(td);
Form_login.ADOQuery_sum.Parameters.ParamByName('d2').Value:=StrToDate('31.12.2099');
Form_login.ADOQuery_sum.Open;
GroupBox7.Caption:=GroupBox7.Caption+' '+IntToStr(Form_login.ADOQuery_full.RecordCount);
end;
procedure TForm_GL.BitBtn7Click(Sender: TObject);
begin
BitBtn4.Visible:=false;
BitBtn7.Visible:=false;
ShowMessage('Редактирование заявки отменено!');
id_lic_del.Text:="";
id_z1_del.Text:="";
Edit1.Text:="";
Edit2.Text:="";
```

## Продолжение приложения А

```
Edit3.Text:="";
Edit7.Text:="";
Edit8.Text:="";
Edit10.Text:="";
MaskEdit1.Text:="";
MaskEdit2.Text:="";
LabeledEdit1.Text:="";
LabeledEdit2.Text:="";
LabeledEdit3.Text:="";
DBLookupComboBox1.KeyValue:=0;
end;
procedure TForm_GL.BitBtn8Click(Sender: TObject);
begin
ShowMessage('Добавление заявки отменено!');
id_lic_del.Text:="";
id_z1_del.Text:="";
Edit1.Text:="";
Edit2.Text:="";
Edit3.Text:="";
Edit7.Text:="";
Edit8.Text:="";
Edit10.Text:="";
MaskEdit1.Text:="";
MaskEdit2.Text:="";
LabeledEdit1.Text:="";
LabeledEdit2.Text:="";
LabeledEdit3.Text:="";
DBLookupComboBox1.KeyValue:=0;
BitBtn3.Visible:=false;
BitBtn8.Visible:=false;
end;
procedure TForm_GL.BitBtn9Click(Sender: TObject);
var
Data: TDate;
i:integer;
s:string;
begin
Memo1.Lines.Add(edit2.Text);
while Edit13.text<>edit1.text do
begin
edit1.text:=DateToStr(DateTimePicker1.Date);
data:=StrToDate(Edit13.text)+1;
Memo1.Lines.Add(datetostr(data));
Edit13.Text:=DateToStr(data);
end;
if Memo1.Lines.Count > 0 then
begin
for i:=0 to Memo1.Lines.Count-1 do
begin
s:=Memo1.Lines.Strings[i];
end;
memo1.Clear
end;
end;
procedure TForm_GL.Button1Click(Sender: TObject);
begin
GroupBox7.Caption:='Сводная информация';
Form_login.ADOQuery_FULL.Close;
Form_login.ADOQuery_FULL.SQL.Clear;
```

## Продолжение приложения А

```
Form_login.ADOQuery_FULL.SQL.Add('obsh ');
Form_login.ADOQuery_FULL.Open;
Form_login.ADOQuery_sum.Close;
Form_login.ADOQuery_sum.SQL.Clear;
Form_login.ADOQuery_sum.SQL.Add('sum_ob');
Form_login.ADOQuery_sum.Open;
GroupBox7.Caption:=GroupBox7.Caption+' '+IntToStr(Form_login.ADOQuery_full.RecordCount);
end;
procedure TForm_GL.Button2Click(Sender: TObject);
var
  td:Tdatetime;
begin
  td:=now;
  Form_login.ADOQuery_FULL.Close;
  Form_login.ADOQuery_FULL.SQL.Clear;
  Form_login.ADOQuery_FULL.SQL.Add('obsh_v :d');
  Form_login.ADOQuery_FULL.Parameters.ParamByName('d').Value:=DateToStr(td);
  Form_login.ADOQuery_FULL.Open;
  ShowMessage('Добавлено выполнено. Таблица обновлена');
end;
procedure TForm_GL.DateTimePicker1Change(Sender: TObject);
var
  a:integer;
begin
  Edit1.Text:=DateToStr(DateTimePicker1.Date);
  a:= abs( trunc(datetimerpicker2.date) - trunc(datetimerpicker1.date) );
  if
    StrToDate(Edit1.text)<StrToDate(Edit2.text)
  then
    begin
      ShowMessage('Дата приезда<Даты выезда');
      BitBtn3.Enabled:=false;
      BitBtn4.Enabled:=false;
    end
  else
    if
      a<w then
        begin
          BitBtn3.Enabled:=true;
          BitBtn4.Enabled:=true;
        end
      else
        ShowMessage(z+' '+IntToStr(w)+' дней');
    end;
end;
procedure TForm_GL.DateTimePicker1Enter(Sender: TObject);
var
  td:TDateTime;
begin
  td:=Now;
  DateTimePicker1.date:=now;
end;
procedure TForm_GL.DateTimePicker2Change(Sender: TObject);
begin
  Edit2.Text:=DateToStr(DateTimePicker2.Date);
  Edit13.Text:=DateToStr(DateTimePicker2.Date);
end;
procedure TForm_GL.DateTimePicker2Enter(Sender: TObject);
var
  td:TDateTime;
```

## Продолжение приложения А

```
begin
td:=Now;
DateTimePicker2.date:=now;
end;
procedure TForm_GL.DateTimePicker3Change(Sender: TObject);
begin
Edit9.Text:=DateToStr(DateTimePicker3.Date);
end;
procedure TForm_GL.DateTimePicker3Enter(Sender: TObject);
var
td:TDateTime;
begin
td:=Now;
DateTimePicker3.date:=now;
end;
procedure TForm_GL.DBGrid1CellClick(Column: TColumn);
begin
Form_GL.LabeledEdit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('dt_z').AsString;
Form_GL.LabeledEdit2.Text:=DBGrid1.DataSource.DataSet.fieldbyname('name_otd').AsString;
Form_GL.LabeledEdit3.Text:=DBGrid1.DataSource.DataSet.fieldbyname('km').AsString;
end;
procedure TForm_GL.DBGrid1DrawColumnCell(Sender: TObject; const Rect: TRect;
DataCol: Integer; Column: TColumn; State: TGridDrawState);
begin
if DBGrid1.DataSource.DataSet.fieldbyname('status').AsString='В работе' then {тут любое условие}
begin
DBGrid1.Canvas.Brush.Color := Edit6.Color;
DBGrid1.Canvas.Font.Color := clBlack;
DBGrid1.DefaultDrawColumnCell(Rect, DataCol, Column, State);
end
else
if DBGrid1.DataSource.DataSet.fieldbyname('status').AsString='Дубликат' then {тут любое условие}
begin
DBGrid1.Canvas.Brush.Color := Edit12.color;
DBGrid1.Canvas.Font.Color := clBlack;
DBGrid1.DefaultDrawColumnCell(Rect, DataCol, Column, State);
end
else
if DBGrid1.DataSource.DataSet.fieldbyname('status').AsString='Сформирована' then {тут любое условие}
begin
DBGrid1.Canvas.Brush.Color := Edit4.color;
DBGrid1.Canvas.Font.Color := clBlack;
DBGrid1.DefaultDrawColumnCell(Rect, DataCol, Column, State);
end
else
if DBGrid1.DataSource.DataSet.fieldbyname('status').AsString='На утверждении' then {тут любое условие}
begin
DBGrid1.Canvas.Brush.Color := Edit5.color;
DBGrid1.Canvas.Font.Color := clBlack;
DBGrid1.DefaultDrawColumnCell(Rect, DataCol, Column, State);
end;
end;
procedure TForm_GL.DBLookupComboBox1Enter(Sender: TObject);
begin
Form_login.ADOQuery_park.Close;
Form_login.ADOQuery_park.Open;
end;
procedure TForm_GL.DBLookupComboBox1KeyPress(Sender: TObject; var Key: Char);
begin
```

## Продолжение приложения А

```
if Ord(Key) = VK_RETURN then
  begin
    Form_login.ADOQuery_park_img.Close;
    Form_login.ADOQuery_park_img.SQL.Clear;
    Form_login.ADOQuery_park_img.SQL.Add('park_s :IDI');
    Form_login.ADOQuery_park_img.Parameters.ParamByName('idi').Value:=DBLookupComboBox1.KeyValue;
    Form_login.ADOQuery_park_img.Open;
    Form_GL.Edit_img.Text:=Form_login.ADOQuery_park_img.FieldByName('img').AsString;
    Form_GL.Edit_park.Text:=Form_login.ADOQuery_park_img.FieldByName('marka').AsString;
    Form_GL.Edit_g.Text:=Form_login.ADOQuery_park_img.FieldByName('gos_n').AsString;
    if Form_GL.Edit_img.Text<>" then
      begin
        Form_img.Caption:=Form_img.Caption+' '+Edit_park.Text+' / '+Form_GL.Edit_g.Text;
        Form_img.Image1.Picture.LoadFromFile(Form_GL.Edit_img.Text);
        Form_img.ShowModal;
      end else
        if Form_GL.Edit_img.Text="" then
          ShowMessage('Изображение не найдено');
          end;
        end;
        procedure TForm_GL.Edit12DbClick(Sender: TObject);
        begin
          if ColorDialog1.Execute then
            edit12.Color:=ColorDialog1.Color;
          end;
          procedure TForm_GL.Edit4DbClick(Sender: TObject);
          begin
            if ColorDialog1.Execute then
              edit4.Color:=ColorDialog1.Color;
            end;
            procedure TForm_GL.Edit5DbClick(Sender: TObject);
            begin
              if ColorDialog1.Execute then
                edit5.Color:=ColorDialog1.Color;
              end;
              procedure TForm_GL.Edit6DbClick(Sender: TObject);
              begin
                if ColorDialog1.Execute then
                  edit6.Color:=ColorDialog1.Color;
                end;
                procedure TForm_GL.Edit9Change(Sender: TObject);
                begin
                  if Edit9.Text="" then
                    BitBtn5.Enabled:=false
                  else
                    BitBtn5.Enabled:=true;
                  end;
                  procedure TForm_GL.FormClose(Sender: TObject; var Action: TCloseAction);
                  begin
                    Application.Terminate;
                  end;
                  procedure TForm_GL.FormCreate(Sender: TObject);
                  begin
                    if not FileExists(ExtractFileDir(Application.ExeName)+'\Color.ini') then begin
                      Form_GL.StatusBar1.Panels[8].Text:='Палитра: по умолчанию';
                      exit;
                    end;
                    iniFile:=TIniFile.Create(ExtractFileDir(Application.ExeName)+'\Color.ini');
                    form_GL.edit4.color:=StringToColor(iniFile.ReadString('text','Заявка сформирована',''));
                end;
            end;
        end;
    end;
  end;
```

## Продолжение приложения А

```
form_GL.edit5.Color:=StringToColor(inifile.ReadString('text','Заявка на утверждении',''));
form_GL.edit6.Color:=StringToColor(inifile.ReadString('text','Заявка в работе',''));
form_GL.edit12.Color:=StringToColor(inifile.ReadString('text','Дубликат',''));
inifile.Free;
Form_GL.StatusBar1.Panels[8].Text:='Палитра: загружена';
end;
procedure TForm_GL.N11Click(Sender: TObject);
var
td:TDateTime;
begin
td:=Now;
Form_f1.Edit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('d1').AsString;
if spec.Text='1' then
begin ShowMessage('Недостаточно прав!');
end else
if spec.Text='2' then
begin
Form_f1.ShowModal;
end;
end;
procedure TForm_GL.N12Click(Sender: TObject);
var
td:TDateTime;
begin
td:=Now;
Form_f2.Edit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('d1').AsString;
if spec.Text='1' then
begin ShowMessage('Недостаточно прав!');
end else
if spec.Text='2' then
begin
Form_f2.ShowModal;
end;
end;
procedure TForm_GL.N13Click(Sender: TObject);
begin
Form_pol.frxReport1.Variables['org']:='' + Form_GL.StatusBar1.Panels[10].Text+'';
Form_pol.frxReport1.Variables['v']:='' + Form_GL.StatusBar1.Panels[11].Text+'';
Form_pol.frxReport1.Variables['t']:='' + Form_GL.StatusBar1.Panels[7].Text+'';
Form_pol.frxPDFExport1.FileName:='Лицензионное соглашение';
Form_pol.frxReport1.ShowReport;
Form_pol.frxReport1.Export(Form_pol.frxPDFExport1);
end;
procedure TForm_GL.N14Click(Sender: TObject);
begin
Form_treb.ShowModal;
end;
procedure TForm_GL.N15Click(Sender: TObject);
begin
if spec.Text='1' then
begin ShowMessage('Недостаточно прав!');
end else
if spec.Text='2' then
begin
Form_lic_spr.ShowModal;
end;
end;
procedure TForm_GL.N16Click(Sender: TObject);
begin
```

## Продолжение приложения А

```
Form_copy.ShowModal;
end;
procedure TForm_GL.N17Click(Sender: TObject);
var
  i, j: Integer;
  s: string;
begin
  if spec.Text='1' then
  begin ShowMessage('Недостаточно прав!');
  end else
  if spec.Text='2' then
  begin
    if DBGrid1.SelectedRows.Count>0 then
      with DBGrid1.DataSource.DataSet do
        for i:=0 to DBGrid1.SelectedRows.Count-1 do
          begin
            GotoBookmark (DBGrid1.SelectedRows.Items[i]);
            Form_GL.id_z1_del.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id').AsString;
            Form_login.ADOQuery_ADD_z.Close;
            Form_login.ADOQuery_ADD_z.SQL.Clear;
            Form_login.ADOQuery_ADD_z.SQL.Add('update_z1 :STATUS,:IDI');
            Form_login.ADOQuery_ADD_z.Parameters.ParamByName('STATUS').Value:='На утверждении';
            Form_login.ADOQuery_ADD_z.Parameters.ParamByName('IDI').Value:=StrToInt(id_z1_del.text);
            Form_login.ADOQuery_ADD_z.ExecSQL;
          end;
          ShowMessage('Заявки утверждены. Таблица обновлена');
          Form_login.ADOQuery_FULL.Active:=false;
          Form_login.ADOQuery_FULL.Active:=true;
        end;
      end;
  procedure TForm_GL.N18Click(Sender: TObject);
  begin
    Form_GL.id_lic_del.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id_1').AsString;
    Form_GL.status.Text:=DBGrid1.DataSource.DataSet.fieldbyname('status').AsString;
    if (Form_GL.status.text<>'В работе') and (Form_GL.status.text<>'Дубликат') then //если не пустой
    begin
      try
        ShowMessage('Редактирование утвержденной заявки невозможно!');
      except
        MessageDlg('&#194;', mtError,[mbOK],0);
        exit;
      end;
    end else
    begin
      Form_GL.id_z1_del.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id').AsString;
      Form_GL.Edit2.Text:=DBGrid1.DataSource.DataSet.fieldbyname('d1').AsString;
      Form_GL.MaskEdit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('t1').AsString;
      Form_GL.Edit7.Text:=DBGrid1.DataSource.DataSet.fieldbyname('pr').AsString;
      Form_GL.DBLookupComboBox1.KeyValue:=DBGrid1.DataSource.DataSet.fieldbyname('id_p').AsInteger;
      Form_GL.Edit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('d2').AsString;
      Form_GL.MaskEdit2.Text:=DBGrid1.DataSource.DataSet.fieldbyname('t2').AsString;
      Form_GL.Edit8.Text:=DBGrid1.DataSource.DataSet.fieldbyname('cel').AsString;
      Form_GL.Edit3.Text:=DBGrid1.DataSource.DataSet.fieldbyname('m1').AsString;
      Form_GL.Edit10.Text:=DBGrid1.DataSource.DataSet.fieldbyname('km').AsString;
      BitBtn4.Visible:=true;
      BitBtn7.Visible:=true;
      BitBtn3.Visible:=false;
      BitBtn8.Visible:=false;
    end;
  end;
```

## Продолжение приложения А

```
end;
procedure TForm_GL.N19Click(Sender: TObject);
var
  TD:TDateTime;
begin
  td:=now;
  Form_GL.id_z1_del.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id').AsString;
  Form_login.ADOQuery_ADD_z.Close;
  Form_login.ADOQuery_ADD_z.SQL.Clear;
  Form_login.ADOQuery_ADD_z.SQL.Add('INSERT INTO z1(
dt_z,id_ot,id_lic,id_p,d1,t1,pr,status,d2,t2,cel,m1,km)SELECT z1.dt_z,
id_ot=:idt,id_lic=:idl,id_p=:d1,t1,pr,status=:st,d2=:d2,t2,cel,m1,km FROM z1 WHERE z1.id=:IDI');
  Form_login.ADOQuery_ADD_z.Parameters.ParamByName('idt').Value:=StrToInt(id_ot.text);
  Form_login.ADOQuery_ADD_z.Parameters.ParamByName('idl').Value:=StrToInt(Form_GL.id_lic.Text);
  Form_login.ADOQuery_ADD_z.Parameters.ParamByName('d1').Value:=now;
  Form_login.ADOQuery_ADD_z.Parameters.ParamByName('st').Value:='Дубликат';
  Form_login.ADOQuery_ADD_z.Parameters.ParamByName('d2').Value:=now;
  Form_login.ADOQuery_ADD_z.Parameters.ParamByName('idi').Value:=StrToInt(Form_GL.id_z1_del.Text);
  Form_login.ADOQuery_ADD_z.ExecSQL;
  Form_login.ADOQuery_FULL.Close;
  Form_login.ADOQuery_FULL.SQL.Clear;
  Form_login.ADOQuery_FULL.SQL.Add('obsh_v :d1, :d2');
  Form_login.ADOQuery_FULL.Parameters.ParamByName('d1').Value:=DateToStr(td);
  Form_login.ADOQuery_FULL.Parameters.ParamByName('d2').Value:=StrToDate('31.12.2099');
  Form_login.ADOQuery_FULL.Open;
  ShowMessage('Запись добавлена');
end;
procedure TForm_GL.N1Click(Sender: TObject);
begin
  BitBtn3.Visible:=true;
  BitBtn8.Visible:=true;
  BitBtn7.Visible:=false;
  BitBtn4.Visible:=false;
end;
procedure TForm_GL.N20Click(Sender: TObject);
var
  inifile:TIniFile;
begin
  inifile:=TIniFile.Create(ExtractFileDir(Application.ExeName)+'\Color.ini');
  inifile.WriteString('text','Заявка сформирована',ColorToString(Edit4.color));
  inifile.WriteString('text','Заявка на утверждении',ColorToString(Edit5.color));
  inifile.WriteString('text','Заявка в работе',ColorToString(Edit6.color));
  inifile.WriteString('text','Дубликат',ColorToString(Edit12.color));
  inifile.Free;
  ShowMessage('Палитра сохранена');
end;
procedure TForm_GL.N22Click(Sender: TObject);
begin
  edit4.Color:=clMoneyGreen;
  edit5.Color:=clGradientActiveCaption;
  edit6.Color:=clInfoBk;
  edit12.Color:=clOlive;
  ShowMessage('Палитра восстановлена');
  Form_GL.StatusBar1.Panels[8].Text:='Палитра: по умолчанию';
end;
procedure TForm_GL.N23Click(Sender: TObject);
begin
  if not FileExists(ExtractFileDir(Application.ExeName)+'\Color.ini') then begin
  Form_GL.StatusBar1.Panels[8].Text:='Палитра: по умолчанию';
```



## Продолжение приложения А

```
exit;
end;
inifile:=TIniFile.Create(ExtractFileDir(Application.ExeName)+'\Color.ini');
form_GL.edit4.color:=StringToColor(inifile.ReadString('text','Заявка сформирована,'));
form_GL.edit5.Color:=StringToColor(inifile.ReadString('text','Заявка на утверждении,'));
form_GL.edit6.Color:=StringToColor(inifile.ReadString('text','Заявка в работе,'));
form_GL.edit12.Color:=StringToColor(inifile.ReadString('text','Дубликат,'));
inifile.Free;
Form_GL.StatusBar1.Panels[8].Text:='Палитра: загружена';
end;
procedure TForm_GL.N28Click(Sender: TObject);
begin
Form_otdel_spr.ShowModal;
end;
procedure TForm_GL.N2Click(Sender: TObject);
begin
Form_GL.id_lic_del.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id_1').AsString;
Form_GL.status.Text:=DBGrid1.DataSource.DataSet.fieldbyname('status').AsString;
if (id_lic_del.Text<>id_lic.text) or (Form_GL.status.text<>'В работе') and (Form_GL.status.text<>'Дубликат') then
//если не пустой
begin
try
ShowMessage('Редактирование чужой/утвержденной заявки невозможно!');
except
MessageDlg('&#194;', mtError,[mbOK],0);
exit;
end;
end else
begin
Form_GL.id_z1_del.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id').AsString;
Form_GL.Edit2.Text:=DBGrid1.DataSource.DataSet.fieldbyname('d1').AsString;
Form_GL.MaskEdit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('t1').AsString;
Form_GL.Edit7.Text:=DBGrid1.DataSource.DataSet.fieldbyname('pr').AsString;
Form_GL.DBLookupComboBox1.KeyValue:=DBGrid1.DataSource.DataSet.fieldbyname('id_p').AsInteger;
Form_GL.Edit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('d2').AsString;
Form_GL.MaskEdit2.Text:=DBGrid1.DataSource.DataSet.fieldbyname('t2').AsString;
Form_GL.Edit8.Text:=DBGrid1.DataSource.DataSet.fieldbyname('cel').AsString;
Form_GL.Edit3.Text:=DBGrid1.DataSource.DataSet.fieldbyname('m1').AsString;
Form_GL.Edit10.Text:=DBGrid1.DataSource.DataSet.fieldbyname('km').AsString;
BitBtn4.Visible:=true;
BitBtn7.Visible:=true;
BitBtn3.Visible:=false;
BitBtn8.Visible:=false;
end;
end;
procedure TForm_GL.N30Click(Sender: TObject);
begin
if spec.Text='1' then
begin ShowMessage('Недостаточно прав!');
end else
if spec.Text='2' then
begin
Form_spr_park.ShowModal;
end;
end;
procedure TForm_GL.N31Click(Sender: TObject);
begin
if spec.Text='1' then
begin ShowMessage('Недостаточно прав!');
```

## Продолжение приложения А

```
end else
if spec.Text='2' then
begin
Form_otdel.ShowModal;
end;
end;
procedure TForm_GL.N32Click(Sender: TObject);
begin
if spec.Text='1' then
begin ShowMessage('Недостаточно прав!');
end else
if spec.Text='2' then
begin
Form_f3.ShowModal;
end;
end;
procedure TForm_GL.N33Click(Sender: TObject);
begin
Form_f4.ShowModal;
end;
procedure TForm_GL.N3Click(Sender: TObject);
var
td:TDateTime;
begin
td:=Now;
Form_GL.id_lic_del.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id_1').AsString;
Form_GL.id_z1_del.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id').AsString;
Form_GL.status.Text:=DBGrid1.DataSource.DataSet.fieldbyname('status').AsString;
if (id_lic_del.Text=id_lic.text) and (status.Text='В работе') then begin
Form_login.ADOQuery_del_z1.Close;
Form_login.ADOQuery_del_z1.SQL.Clear;
Form_login.ADOQuery_del_z1.SQL.Add('delete from z1 where ID=:IDI');
Form_login.ADOQuery_del_z1.Parameters.ParamByName('IDI').Value:=StrToInt(id_z1_del.Text);
Form_login.ADOQuery_del_z1.ExecSQL;
ShowMessage('Удаление выполнено. Таблица обновлена');
id_lic_del.Text:="";
id_z1_del.Text:="";
Form_login.ADOQuery_FULL.Active:=false;
Form_login.ADOQuery_FULL.Active:=true;
Form_login.ADOQuery_sum.Close;
Form_login.ADOQuery_sum.SQL.Clear;
Form_login.ADOQuery_sum.SQL.Add('sum_v :d1, :d2');
Form_login.ADOQuery_sum.Parameters.ParamByName('d1').Value:=DateToStr(td);
Form_login.ADOQuery_sum.Parameters.ParamByName('d2').Value:=StrToDate('31.12.2099');
Form_login.ADOQuery_sum.Open;
GroupBox7.Caption:='Сводная информация';
GroupBox7.Caption:=GroupBox7.Caption+' '+IntToStr(Form_login.ADOQuery_full.RecordCount);
end else
ShowMessage('Удаление чужой записи невозможно');
end;
procedure TForm_GL.N41Click(Sender: TObject);
var
td:TDateTime;
begin
td:=Now;
Form_f5.Edit1.Text:=DBGrid1.DataSource.DataSet.fieldbyname('d1').AsString;
if spec.Text='1' then
begin ShowMessage('Недостаточно прав!');
end else
```

## Продолжение приложения А

```
if spec.Text='2' then
begin
Form_f5.ShowModal;
end;
end;
procedure TForm_GL.N5Click(Sender: TObject);
begin
Form_login.Close;
end;
procedure TForm_GL.N7Click(Sender: TObject);
begin
Form_login.Edit1.text:='';
Form_login.DBLookupComboBox1.KeyValue:=0;
Form_GL.Visible:=false;
Form_login.ShowModal;
end;
procedure TForm_GL.ScrollBox1MouseDown(Sender: TObject; Shift: TShiftState;
  MousePos: TPoint; var Handled: Boolean);
begin
Scrollbox1.VertScrollBar.Position := Scrollbox1.VertScrollBar.Position + 4;
end;
procedure TForm_GL.ScrollBox1MouseWheelUp(Sender: TObject; Shift: TShiftState;
  MousePos: TPoint; var Handled: Boolean);
begin
Scrollbox1.VertScrollBar.Position:= Scrollbox1.VertScrollBar.Position-4;
end;
procedure TForm_GL.StatusBar1DrawPanel(StatusBar: TStatusBar;
  Panel: TStatusPanel; const Rect: TRect);
begin
if Panel = StatusBar.Panels[1] then
begin
StatusBar.Canvas.Font.Style:=fsBold;
StatusBar.Canvas.Font.Color := clTeal;
StatusBar.Canvas.TextOut(Rect.Left, Rect.Top, a)
end;
if Panel = StatusBar.Panels[3] then
begin
StatusBar.Canvas.Font.Style:=fsBold;
StatusBar.Canvas.Font.Color := clTeal;
StatusBar.Canvas.TextOut(Rect.Left, Rect.Top, b)
end;
if Panel = StatusBar.Panels[5] then
begin
StatusBar.Canvas.Font.Style:=fsBold;
StatusBar.Canvas.Font.Color := clTeal;
StatusBar.Canvas.TextOut(Rect.Left, Rect.Top, c)
end;
if Panel = StatusBar.Panels[7] then
begin
StatusBar.Canvas.Font.Style:=fsBold;
StatusBar.Canvas.Font.Color := clTeal;
StatusBar.Canvas.TextOut(Rect.Left, Rect.Top, i)
end;
end;
procedure TForm_GL.Timer1Timer(Sender: TObject);
var
td:TDateTime;
begin
td:=Now;
```

## Продолжение приложения А

```
Form_GL.StatusBar1.Panels[7].Text:=DateTimeToStr(td);
i:=Form_GL.StatusBar1.Panels[7].Text;
if (Form_login.Edit3.Text='ASU3\SQL EXPRESS') or (Form_login.Edit3.Text='ДОМАШНИЙ-ПК\SQL EXPRESS')
then begin
Form_GL.StatusBar1.Panels[10].Text:='Разработчик';
d:=Form_GL.StatusBar1.Panels[10].Text
end else
if (Form_login.Edit3.Text='kulpu-wsus-s01') or (Form_login.Edit3.Text='kulpu-s05') then begin
Form_GL.StatusBar1.Panels[10].Text:='ЛПУМГ' ;
d:=Form_GL.StatusBar1.Panels[10].Text;
end
else
if Form_login.Edit3.Text='mynout' then begin
Form_GL.StatusBar1.Panels[10].Text:='Домашка: Олег';
d:=Form_GL.StatusBar1.Panels[10].Text;
end;
end;
end.
unit Unit_f1;
interface
uses
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Buttons, Vcl.ComCtrls;
type
TForm_f1 = class(TForm)
Edit1: TEdit;
DateTimePicker1: TDateTimePicker;
BitBtn1: TBitBtn;
Label1: TLabel;
ComboBox1: TComboBox;
procedure DateTimePicker1Change(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure ComboBox1Change(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form_f1: TForm_f1;
implementation
{$R *.dfm}
uses Unit_login, Unit_GL, Unit_f1e;
procedure TForm_f1.BitBtn1Click(Sender: TObject);
begin
Form_login.ADOQuery_ADD_z.Close;
Form_login.ADOQuery_ADD_z.SQL.Clear;
Form_login.ADOQuery_ADD_z.SQL.Add('UPDATE z1 SET status=:STATUS where d1=:d1 and status=:st ');
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('d1').Value:=StrToDate(Edit1.text);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('st').Value:='На утверждении';
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('STATUS').Value:='Сформирована';
Form_login.ADOQuery_ADD_z.ExecSQL;
Form_login.ADOQuery_f1exp.Close;
Form_login.ADOQuery_f1exp.SQL.Clear;
Form_login.ADOQuery_f1exp.SQL.Add('select z1.id, z1.dt_z,z1.cel, z1.status, z1.d1, z1.m1, z1.t1,
park.marka,park.gos_n, otdel.name_otd, lic.fio, lic.id, z1.id_p from z1, park, otdel, lic where park.id=z1.id_p and
z1.id_ot=otdel.id and z1.id_lic=lic.id and d1=:d1+' and status=:status order by otdel.name_otd');
Form_login.ADOQuery_f1exp.Parameters.ParamByName('d1').Value:=StrToDate(edit1.Text);
Form_login.ADOQuery_f1exp.Parameters.ParamByName('status').Value:='Сформирована';
```

## Продолжение приложения А

```
Form_login.ADOQuery_f1exp.Open;
Form_login.ADOQuery_FULL.Active:=false;
Form_login.ADOQuery_FULL.Active:=true;
ShowMessage('Заявка сформирована. Таблица обновлена');
Form_f1_e.frxReport1.Variables['ot']:='' + Form_GL.StatusBar1.Panels[1].Text + '. Отдел: ' +
Form_GL.StatusBar1.Panels[3].Text + '';
Form_f1_e.frxReport1.Variables['d3']:='' + Form_GL.StatusBar1.Panels[7].Text + '';
Form_f1_e.frxReport1.Variables['da']:='' + Edit1.Text + '';
Form_f1_e.frxPDFExport1.FileName:='Ф2_Приказ-наряд'+'+Edit1.text;
Form_f1_e.frxRTFExport1.FileName:='Ф2_Приказ-наряд'+'+Edit1.text;
Form_f1_e.frxReport1.ShowReport;
if ComboBox1.Text='.pdf' then
Form_f1_e.frxReport1.Export(Form_f1_e.frxPDFExport1)
else
if ComboBox1.Text='.rtf'
then
Form_f1_e.frxReport1.Export(Form_f1_e.frxRTFExport1)
end;
procedure TForm_f1.ComboBox1Change(Sender: TObject);
begin
if ComboBox1.Text<>' ' then
BitBtn1.Enabled:=true
else
BitBtn1.Enabled:=false;
end;
procedure TForm_f1.DateTimePicker1Change(Sender: TObject);
begin
Edit1.Text:=DateToStr(DateTimePicker1.Date);
end;
end.
unit Unit_f2;
interface
uses
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ComCtrls, Vcl.StdCtrls, Vcl.Buttons,
Data.DB, Vcl.Grids, Vcl.DBGrids;
type
TForm_f2 = class(TForm)
BitBtn1: TBitBtn;
Edit1: TEdit;
DateTimePicker1: TDateTimePicker;
DBGrid1: TDBGrid;
ComboBox1: TComboBox;
Label1: TLabel;
procedure DateTimePicker1Change(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure ComboBox1Change(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form_f2: TForm_f2;
implementation
{$R *.dfm}
uses Unit_login, Unit_f2e, Unit_f1e, Unit_GL;
procedure TForm_f2.BitBtn1Click(Sender: TObject);
begin
```

## Продолжение приложения А

```
Form_login.ADOQuery_ADD_z.Close;
Form_login.ADOQuery_ADD_z.SQL.Clear;
Form_login.ADOQuery_ADD_z.SQL.Add('UPDATE z1 SET status=:STATUS where d1=:d1 and status=:st ');
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('d1').Value:=StrToDate(Edit1.text);
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('st').Value:='На утверждении';
Form_login.ADOQuery_ADD_z.Parameters.ParamByName('STATUS').Value:='Сформирована';
Form_login.ADOQuery_ADD_z.ExecSQL;
Form_login.ADOQuery_f2exp.SQL.Text:=Form_login.ADOQuery_f2exp.SQL.Text+'AND d1=:d1 and status=:st';
Form_login.ADOQuery_f2exp.Parameters.ParamByName('d1').Value:=StrToDate(edit1.Text);
Form_login.ADOQuery_f2exp.Parameters.ParamByName('st').Value:='Сформирована';
Form_login.ADOQuery_f2exp.Open;
Form_login.ADOQuery_FULL.Active:=false;
Form_login.ADOQuery_FULL.Active:=true;
//Form_login.ADOQuery_f2exp.SQL.Text:=Form_login.ADOQuery1.SQL.Text;
ShowMessage('Заявка сформирована. Таблица обновлена');
Form_f2_e.frxPDFExport1.FileName:='Ф1_Заявка на транспорт'+'+Edit1.text;
Form_f2_e.frxRTFExport1.FileName:='Ф1_Заявка на транспорт'+'+Edit1.text;
Form_f2_e.frxReport1.Variables['da']:='' + Edit1.Text + '';
Form_f2_e.frxReport1.Variables['ot']:='' + Form_GL.StatusBar1.Panels[1].Text +'. Отдел: '+
Form_GL.StatusBar1.Panels[3].Text + '';
Form_f2_e.frxReport1.Variables['d3']:='' + Form_GL.StatusBar1.Panels[7].Text + '';
Form_f2_e.frxReport1.ShowReport;
if ComboBox1.Text='.pdf' then
begin
Form_f2_e.frxReport1.Export(Form_f2_e.frxPDFExport1);
Form_login.ADOQuery_f2exp.SQL.Text:=Form_login.ADOQuery1.SQL.Text;
end
else
if ComboBox1.Text='.rtf'
then begin
Form_f2_e.frxReport1.Export(Form_f2_e.frxRTFExport1);
Form_login.ADOQuery_f2exp.SQL.Text:=Form_login.ADOQuery1.SQL.Text;
end;
end;
procedure TForm_f2.ComboBox1Change(Sender: TObject);
begin
if ComboBox1.Text<>" then
BitBtn1.Enabled:=true
else
BitBtn1.Enabled:=false;
end;
procedure TForm_f2.DateTimePicker1Change(Sender: TObject);
begin
Edit1.Text:=DateToStr(DateTimePicker1.Date);
end;
end.
unit Unit_f4;
interface
uses
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ComCtrls, Vcl.Buttons;
type
TForm_f4 = class(TForm)
Label1: TLabel;
BitBtn1: TBitBtn;
Edit1: TEdit;
DateTimePicker1: TDateTimePicker;
ComboBox1: TComboBox;
Edit2: TEdit;
```

## Продолжение приложения А

```
DateTimePicker2: TDateTimePicker;
  procedure DateTimePicker1Change(Sender: TObject);
  procedure DateTimePicker2Change(Sender: TObject);
  procedure DateTimePicker2Enter(Sender: TObject);
  procedure DateTimePicker1Enter(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form_f4: TForm_f4;
implementation
{$R *.dfm}
uses Unit_f3e, Unit_login, Unit_GL;
procedure TForm_f4.BitBtn1Click(Sender: TObject);
begin
  Form_login.ADOQuery_f3exp.Close;
  Form_login.ADOQuery_f3exp.SQL.Clear;
  Form_login.ADOQuery_f3exp.SQL.Add('F_4 :d1, :d2');
  Form_login.ADOQuery_f3exp.Parameters.ParamByName('d1').Value:=StrToDate(edit1.Text);
  Form_login.ADOQuery_f3exp.Parameters.ParamByName('d2').Value:=StrToDate(edit2.Text);
  Form_login.ADOQuery_f3exp.Open;
  Form_f4e.frxReport1.Variables['d2']:='' + Edit2.Text + '';
  Form_f4e.frxReport1.Variables['d1']:='' + Edit1.Text + '';
  Form_f4e.frxPDFExport1.FileName:='Ф3-Нагрузка на транспорт';
  Form_f4e.frxReport1.Variables['ot']:='' + Form_GL.StatusBar1.Panels[1].Text + '. Отдел: ' +
  Form_GL.StatusBar1.Panels[3].Text + '';
  Form_f4e.frxReport1.Variables['d3']:='' + Form_GL.StatusBar1.Panels[7].Text + '';
  Form_f4e.frxReport1.ShowReport;
  if (ComboBox1.Text='.pdf') and (StrToDate(Edit2.text)> StrToDate(Edit1.text)) then
    begin
      Form_f4e.frxReport1.Export(Form_f4e.frxPDFExport1);
      ShowMessage('Отчет выгружен') end
    else
      if (ComboBox1.Text='.rtf') and (StrToDate(Edit2.text)> StrToDate(Edit1.text))
      then
        begin
          Form_f4e.frxReport1.Export(Form_f4e.frxRTFExport1);
          ShowMessage('Отчет выгружен')
        end
        else
          if (StrToDate(Edit2.text)< StrToDate(Edit1.text)) then
            ShowMessage('Неправильная дата');
          end;
        procedure TForm_f4.DateTimePicker1Change(Sender: TObject);
        begin
          Edit1.Text:=DateToStr(DateTimePicker1.Date);
        end;
        procedure TForm_f4.DateTimePicker1Enter(Sender: TObject);
        var
          td:TDateTime;
        begin
          td:=Now;
          DateTimePicker1.date:=now;
        end;
        procedure TForm_f4.DateTimePicker2Change(Sender: TObject);
        begin
```

## Продолжение приложения А

```
Edit2.Text:=DateToStr(DateTimePicker2.Date);
end;
procedure TForm_f4.DateTimePicker2Enter(Sender: TObject);
var
td:TDateTime;
begin
td:=Now;
DateTimePicker2.date:=now;
end;
end.
unit Unit_f5;
interface
uses
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ComCtrls, Vcl.Buttons;
type
TForm_f5 = class(TForm)
Label1: TLabel;
BitBtn1: TBitBtn;
Edit1: TEdit;
DateTimePicker1: TDateTimePicker;
ComboBox1: TComboBox;
Edit2: TEdit;
DateTimePicker2: TDateTimePicker;
procedure BitBtn1Click(Sender: TObject);
procedure DateTimePicker2Change(Sender: TObject);
procedure DateTimePicker1Change(Sender: TObject);
procedure DateTimePicker2Enter(Sender: TObject);
procedure DateTimePicker1Enter(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form_f5: TForm_f5;
implementation
{$R *.dfm}
uses Unit_f5e, Unit_login, Unit_GL;
procedure TForm_f5.BitBtn1Click(Sender: TObject);
begin
Form_login.ADOQuery_f5.Close;
Form_login.ADOQuery_f5.SQL.Clear;
Form_login.ADOQuery_f5.SQL.Add('F_5 :d1, :d2');
Form_login.ADOQuery_f5.Parameters.ParamByName('d1').Value:=StrToDate(edit1.Text);
Form_login.ADOQuery_f5.Parameters.ParamByName('d2').Value:=StrToDate(edit2.Text);
Form_login.ADOQuery_f5.Open;
Form_f5e.frxReport1.Variables['d2']:='' + Edit2.Text + '';
Form_f5e.frxReport1.Variables['d1']:='' + Edit1.Text + '';
Form_f5e.frxReport1.Variables['ot']:='' + Form_GL.StatusBar1.Panels[1].Text + '. Отдел: ' +
Form_GL.StatusBar1.Panels[3].Text + '';
Form_f5e.frxReport1.Variables['d3']:='' + Form_GL.StatusBar1.Panels[7].Text + '';
Form_f5e.frxPDFExport1.FileName:='Ф4-Сводка по отделениям';
Form_f5e.frxReport1.ShowReport;
if (ComboBox1.Text='.pdf') and (StrToDate(Edit2.text)> StrToDate(Edit1.text)) then
begin
Form_f5e.frxReport1.Export(Form_f5e.frxPDFExport1);
ShowMessage('Отчет выгружен') end
else
```



```

if (ComboBox1.Text='.rtf') and (StrToDate(Edit2.text)> StrToDate(Edit1.text))
then
begin
Form_f5e.frxReport1.Export(Form_f5e.frxRTFExport1);
ShowMessage('Отчет выгружен')
end
else
if (StrToDate(Edit2.text)< StrToDate(Edit1.text)) then
ShowMessage('Неправильная дата');
end;
procedure TForm_f5.DateTimePicker1Change(Sender: TObject);
begin
Edit1.Text:=DateToStr(DateTimePicker1.Date);
end;
procedure TForm_f5.DateTimePicker1Enter(Sender: TObject);
var
td:TDateTime;
begin
td:=Now;
DateTimePicker1.date:=now;
end;
procedure TForm_f5.DateTimePicker2Change(Sender: TObject);
begin
Edit2.Text:=DateToStr(DateTimePicker2.Date);
end;
procedure TForm_f5.DateTimePicker2Enter(Sender: TObject);
var
td:TDateTime;
begin
td:=Now;
DateTimePicker2.date:=now;
end;
end.

```

## Приложение Б

### Список хранимых процедур

<p>Получение информации о пользователе который создал копию БД</p>	<pre>CREATE PROCEDURE dbo.copy_new @d1 AS DATETIME,  @id_lic AS INT,  @status AS VARCHAR(50),  @p1 AS VARCHAR(300) AS INSERT INTO copybase (d1, id_lic, status, p1) VALUES (  @d1, @id_lic, @status, @p1); SELECT otdel.name_otd,lic.fio,profil.name_p,copybase.p1,copybase.status,copybase.d1 FROM dbo.copybase INNER JOIN dbo.lic  ON copybase.id_lic = lic.id INNER JOIN dbo.profil  ON lic.spec = profil.id INNER JOIN dbo.otdel  ON lic.id_otd = otdel.id GO</pre>
<p>Получение информации о количестве километров за определенный период с группировкой по транспортному средству</p>	<pre>CREATE PROCEDURE dbo.F_4  @d1 AS DATE,  @d2 AS DATE AS SELECT * FROM ( SELECT CONCAT(park.marka,' / ', park.gos_n) AS tr,z1.status, km FROM z1, park WHERE z1.id_p=park.id AND d1 BETWEEN @d1 AND @d2 ) t PIVOT ( SUM(t.km)FOR t.status IN ([B работе],[Дубликат],[Сформирована], [На утверждении]) ) p ORDER BY p.tr GO</pre>
<p>Получение информации о количестве заявок за определенный период с группировкой по службам</p>	<pre>CREATE PROCEDURE dbo.F_5  @d1 AS DATE,  @d2 AS DATE AS SELECT * FROM (SELECT otdel.name_otd, z1.id, z1.status FROM z1, lic , otdel WHERE z1.id_lic=lic.id AND lic.id_otd= otdel.id AND z1.d1 BETWEEN @d1 AND @d2 ) t PIVOT (COUNT(t.id) FOR t.status IN ([B работе],[Дубликат],[Сформирована], [На утверждении]) ) p ORDER BY p.name_otd GO</pre>
<p>Добавление новой заявки в таблицу заявок</p>	<pre>CREATE PROC dbo.insert_z1 @dt_z AS NVARCHAR(30),@id_ot AS INT,@id_lic AS INT,@id_p AS INT,@d1 AS DATE,@t1 AS NVARCHAR(30),@pr AS NVARCHAR(100),@status AS NVARCHAR(30),@t2 AS NVARCHAR(30),@d2 AS DATE,@cel AS NVARCHAR(100),@m1 AS NVARCHAR(100),@km AS INT AS INSERT INTO z1 (dt_z,id_ot,id_lic,id_p,d1,t1,pr,status, t2,d2,cel, m1, km) VALUES (@dt_z ,@id_ot,@id_lic,@id_p,@d1,@t1,@pr,  @status,@t2,@d2,@cel,@m1,@km) GO</pre>
<p>Выбор данных для авторизации пользователя</p>	<pre>CREATE PROC dbo.lic_pas @fio AS NVARCHAR(50),  @pas AS NVARCHAR(20) AS select * from lic where id=@FIO and pas=@PAS GO</pre>

## Продолжение приложения Б

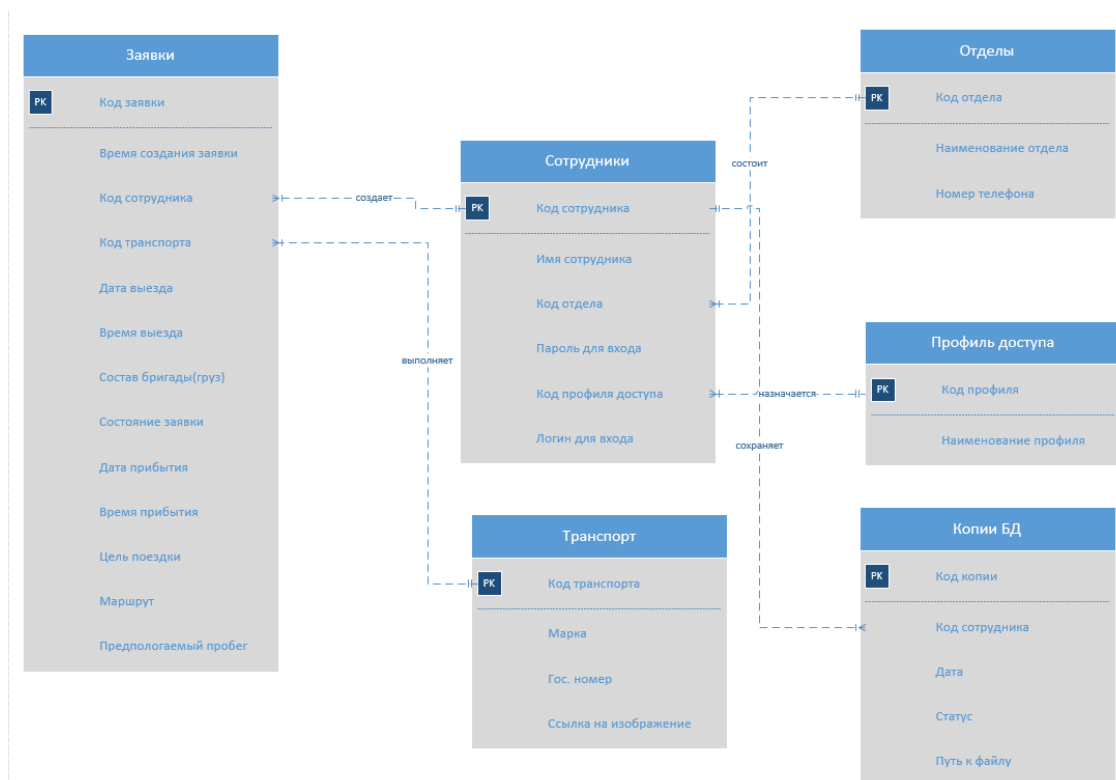
<p>Выбор данных о профиле авторизованного пользователя и службе</p>	<pre>CREATE PROC dbo.lic_v @fio AS NVARCHAR(50), @pas AS NVARCHAR(20) AS select * from lic, profil, otdel where lic.id=@FIO and pas=@PAS and lic.spec=profil.id and otdel.id=lic.id_otd GO</pre>
<p>Выбор списка заявок за определенный период</p>	<pre>CREATE PROC dbo.obsh_v @d1 AS DATE, @d2 AS DATE AS SELECT z1.id,z1.dt_z,z1.pr,z1.status, CONVERT(varchar,d1,104) as d1,z1.t1,park.marka,park.gos_n,otdel.name_otd,lic.fio,lic.id,z1.id_p, CONVERT(varchar,d1,104) as d2,t2,cel,m1,km FROM dbo.z1 INNER JOIN dbo.park ON z1.id_p = park.id INNER JOIN dbo.lic ON z1.id_lic = lic.id INNER JOIN dbo.otdel ON lic.id_otd = otdel.id WHERE d1 BETWEEN @d1 AND @d2 GO</pre>
<p>Выбор списка заявок за весь период работы АИС</p>	<pre>CREATE PROC dbo.obsh AS SELECT z1.id,z1.dt_z,z1.pr,z1.status, CONVERT(varchar,d1,104) as d1,z1.t1,park.marka,park.gos_n,otdel.name_otd,lic.fio,lic.id,z1.id_p, CONVERT(varchar,d1,104) as d2,t2,cel,m1,km FROM dbo.z1 INNER JOIN dbo.park ON z1.id_p = park.id INNER JOIN dbo.lic ON z1.id_lic = lic.id INNER JOIN dbo.otdel ON lic.id_otd = otdel.id GO</pre>
<p>Выбор списка заявок за определенный день с учетом статуса</p>	<pre>CREATE PROC dbo.obsh_v_st @d1 AS DATE, @status AS NVARCHAR(30) AS SELECT z1.id,z1.dt_z,z1.pr,z1.status, CONVERT(varchar,d1,104) as d1,z1.t1,park.marka,park.gos_n,otdel.name_otd,lic.fio,lic.id,z1.id_p, CONVERT(varchar,d1,104) as d2,t2,cel,m1,km FROM dbo.z1 INNER JOIN dbo.park ON z1.id_p = park.id INNER JOIN dbo.lic ON z1.id_lic = lic.id INNER JOIN dbo.otdel ON lic.id_otd = otdel.id WHERE z1.status = @status AND d1=@d1 GO</pre>
<p>Получение данных по выбранному транспортному средству</p>	<pre>CREATE PROC dbo.park_s @IDI AS INT AS select img, marka, gos_n from park where id=@IDI GO</pre>

## Продолжение приложения Б

Получение количества заявок за период с группировкой и сортировкой по дате	<pre>CREATE PROC dbo.sum @d1 AS DATE, @d2 AS DATE AS SELECT COUNT(id), d1 FROM z1 where d1 BETWEEN @d1 AND @d2 group by d1 ORDER BY d1 ASC GO</pre>
Получение количества заявок за определенный день	<pre>CREATE PROC dbo.sum_d @d1 AS DATE AS SELECT COUNT(id), CONVERT(varchar,d1,104) as d1 FROM z1 where d1=@d1 group by d1 GO</pre>
Получение количества заявок за весь период работы АИС	<pre>CREATE PROC dbo.sum_ob AS SELECT COUNT(id), d1 FROM z1 group by d1 ORDER BY d1 ASC GO</pre>
Получение количества заявок за определенный день с учетом статуса	<pre>CREATE PROC dbo.sum_st @d1 AS DATE, @status AS NVARCHAR(30) AS SELECT COUNT(id), CONVERT(varchar,d1,104) as d1 FROM z1 where d1=@d1 and status=@status group by d1 GO</pre>
Обновление статуса выбранной заявки	<pre>CREATE PROC dbo.update_z1 @status AS NVARCHAR(30), @idi AS INT AS UPDATE z1 SET status=@status where ID=@IDI GO</pre>
Обновление данных по выбранной заявке	<pre>CREATE PROC dbo.update_z1_full @dtz AS NVARCHAR(30),@idp AS INT,@d1 AS DATE,@t1 AS NVARCHAR(30),@pr AS NVARCHAR(100),@t2 AS NVARCHAR(30), @d2 AS DATE,@cel AS NVARCHAR(100),@m1 AS NVARCHAR(100),@km AS INT,@sti AS NVARCHAR(30),@idi AS INT AS UPDATE z1 SET dt_z=@dtz, id_p=@idp, d1=@d1, t1=@t1,pr=@pr, t2=@t2, d2=@d2, cel=@cel, m1=@m1, km=@km, status=@sti where ID=@idi GO</pre>

## Приложение В

### Диаграмма «сущность-связь»



# Приложение Г

## Отчет № 1 «Ф-1 Заявка на транспорт»

Сформирован: 02.01.2020 13:47:31    Кем сформирован: Перевртов К.С., Отдел: АУП    Версия: 2.0.0.4

### ЗАЯВКА

На транспорт на 05.06.2018

№ п/п	Дата, время выезда	Маршрут движения	Состав бригады (груз)	Дата, время возвращения	Пробег	Старший бригады	Марка автомобиля	Цель
1	05.06.2018 08:30	КС-Южный поток-КС	3 чел	05.06.2018 16:30	250	Каркищенко А.А.	ГАЗ Борт.(ЭТВС) М*1*СН1*	ТО эл.оборудования
2	05.06.2018 08:30	КС-ст.Ще*ская-КС	1 чел	05.06.2018 16:30	25	Каркищенко А.А.	ГАЗ асинез. Х *41ХХ *3	вывоз сточных вод
3	05.06.2018 08:00	по *ще*скому району	2 человека	05.06.2018 16:30	150	Марченко Е.Н.	УАЗ ав/клуб Х **1 *у *3	Объезд трасс КЛС
4	05.06.2018 08:30	КС-2-я Пятилетка-КС	2 чел	05.06.2018 16:00	160	Скрябин В.	ГАЗ Борт.(ЭГРС) В*3*ОТ1*	ТО ЗРА
5	05.06.2018 08:30	КС-Николае*ка-КС	2 чел	05.06.2018 16:00	180	Скрябин В.	УАЗ легк. А *52 ОА *1	ТО АКГВ
6	05.06.2018 08:30	КС-Сладкий Лиман-КС	2 чел	05.06.2018 16:00	220	Скрябин В.	УАЗ ав/клуб * 52* *Р *3	ТО ШРП
7	05.06.2018 08:30	КС-Красное-КС	2 чел	05.06.2018 16:00	50	Скрябин В.	УАЗ Груз. * Б*3 *Р *3	ТО ПГА
8	05.06.2018 08:00	Егорлыкская	2 спеца ИТЦ	05.06.2018 17:00	400	Рогочий М.В.	Мерседес Эл. Лаб. У 3** ХХ *	Командировка согласно письма ИТЦ № 162 от 29.05.2018
9	05.06.2018 08:00	* Кане*ского и Приморско-Ахтарского районов* (командиро*ка)	2 чел. (командировка)	05.06.2018 17:00	200	Гулько Д.С.	ГАЗ Лаб.КИП О 1* ОА *3	Проверка вычислителей * ГРС (командировка)
10	05.06.2018 08:30	* Ленинградского района	1 чел.	05.06.2018 16:00	120	Гулько Д.С.	УАЗ легк. О *54 ЕН *	ТО пожарной сигнализации
11	05.06.2018 08:30	* Ейского и Староминого района*	1 чел.	05.06.2018 16:00	180	Гулько Д.С.	ГАЗ Борт. В 4*3 ОХ *3	Снятие показаний, ТО *

Приложение Д  
Отчет № 2 «Ф-2 Приказ наряд»

Сформирован:	02.01.2020 13:48:47	Кем сформирован:	Перевертов К.С., Отдел: АУП	Версия: 2.0.0.4
--------------	---------------------	------------------	-----------------------------	-----------------

Приказ-наряд на 05.06.2018

№ п/п	Служба	Время выезда	Ответственный	Вид транспорта	Гос. Номер	Маршрут
1	АТЦ	08:00	Рогочий М.В.	Мерседес Эл. Лаб.	У 3** ХХ *	Егорлыкская
2	АТЦ	07:00	Рогочий М.В.	ПАЗ	Т *** *Х *З	*ще*ская
3	АТЦ	07:00	Рогочий М.В.	ПАЗ	О *4* УА *	*ще*ская
4	АТЦ	07:00	Рогочий М.В.	ПАЗ	Т *** *Х *З	*ще*ская
5	АТЦ	08:00	Рогочий М.В.	МАЗ АТЗ	* 5*1 *Р *З	КС
6	АТЦ	08:00	Рогочий М.В.	МАЗ АТЗ	М44* УН 3*	КС

## Приложение Е

### Отчет № 3 «Ф-3 Анализ нагрузки на транспортное средство»

Сформирован:	31.12.2019 14:37:41	Кем сформирован:	Перевертов К.С., Отдел: АУП	Версия: 2.0.0.4
--------------	---------------------	------------------	-----------------------------	-----------------

#### Анализ нагрузки на транспортные средства

Критерии анализа	
Дата выезда начальная	01.01.2018
Дата выезда конечная	31.12.2019

Сводная информация	
Состояние	Всего км.
В работе	137112
На утверждении	4440
Дубликат	5250
Сформирована	1227318

Детальная информация				
Транспорт	Состояние заявки			
	В работе	Дубликат	Сформирована	На утверждении
VAO-33 Автопогрузчик / Б/Н	452		751	
Б-170 Бульдозер / *1-34 УТ *	17		509	
ГАЗ автобус / А*550А*1	2670		41295	
ГАЗ автобус / В 4*3 ОХ *3	4220	560	28310	
ГАЗ автобус / С 4** НМ *3	560		15530	100
ГАЗ асинез. / Х *41ХХ *3	390		4890	25
ГАЗ Борт. / В 4*3 ОХ *3	2850		31800	320
ГАЗ Борт. / Т *12 *Х *3	850		4099	



Приложение Ё  
Отчет № 4 «Ф-4 Заявки по отделениям»

Сформирован:	02.01.2020 13:41:03	Кем сформирован:	Перевертов К.С., Отдел: АУП	Версия: 2.0.0.4
--------------	---------------------	------------------	-----------------------------	-----------------

**Анализ нагрузки по отделениям**

Критерии анализа	
Дата выезда начальная	01.01.2018
Дата выезда конечная	31.12.2019

Сводная информация	
Состояние	Всего заявок
В работе	817
На утверждении	30
Дубликат	28
Сформирована	8977

Детальная информация				
Отделение	Состояние заявки			
	В работе	Дубликат	Сформирована	На утверждении
АТЦ	241	3	2217	9
АУП	13	2	184	0
ГКС	8	2	198	0
КАДРЫ	12	0	116	0
ЛЭС	133	3	2849	9
МТС	7	0	107	0
ОТ и ПБ	2	0	27	0
ОТ и ТЭП	0	0	1	0
САиМО	57	2	654	2
Связь	31	0	215	1
СЗК	37	0	561	0
УКГ	0	0	23	0
УТРЗ	31	1	220	0
ЭГРС	185	1	1141	7
ЭТВС	60	14	464	2