

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Лесосибирский педагогический институт –  
филиал Сибирского федерального университета

Кафедра высшей математики, информатики и естествознания

УТВЕРЖДАЮ  
Заведующий кафедрой  
Н.Ф. Романцова  
подпись  
« 8 » июня 2018 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.02 Информационные системы и технологии

Разработка информационной системы для АТП г.Лесосибирска

Руководитель	<u>Л.Н. Храмова</u> подпись, дата 04.06.18	доцент, канд.экон.наук	Л.Н.Храмова
Руководитель	<u>А.В. Рубцов</u> подпись, дата 04.06.18	доцент, канд.экон.наук	А.В.Рубцов
Выпускник	<u>А.С. Сахаров</u> подпись, дата 01.06.18		А.С.Сахаров

Лесосибирск 2018

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка информационной системы для АТП г. Лесосибирска».

Выпускная квалификационная работа состоит из введения, 2 глав, выводов, списка использованных источников, состоящего из 48 источников. Работа изложена на 78 листах печатного текста, содержит 33 рисунка и 2 таблицы.

Целью данной работы является разработка и внедрение информационной системы на автотранспортном предприятии ООО «Автобусное содружество» для повышения эффективности работы предприятия.

В соответствии с данной целью в работе решаются следующие задачи:

- проанализировать научную и специальную литературу по теме исследования;
- раскрыть теоретические основы информационных систем;
- спроектировать организационную и процессную модели АТП;
- разработать информационную систему предприятия с использованием Borland Delphi.

С использованием СУБД Microsoft Office Access и ООП Delphi 7.0 был создан комплекс программных приложений объединенных сетью клиент-сервер для автотранспортного предприятия.

Разработанная информационная система выполняет следующие функции: кадровый учёт, кассовый учёт доходов и расходов, складской учёт, хранение информации о прохождении ежедневного медицинского осмотра сотрудниками и технического осмотра автотранспорта.

Разработанное приложение позволяет улучшить эффективность работы предприятия и позволяет вывести автотранспортное предприятие на новый информационный уровень.

## СОДЕРЖАНИЕ

Введение.....	5
1 Теоретические основы информационных систем .....	6
1.1 Информационные системы.....	6
1.1.1 Данные .....	7
1.1.2 База данных.....	9
1.1.3 База знаний.....	12
1.1.4 Программное обеспечение (пакет прикладных программ).....	12
1.1.5 Экспертные системы.....	13
1.1.6 Компьютерные сети.....	15
1.1.7 Защита информации и информационная безопасность.....	17
1.2 Информационная система предприятия АТП.....	18
1.3 Модель автотранспортного предприятия.....	19
1.3.1 Организационная модель.....	21
1.3.2 Процессная модель.....	23
2 Разработка программного продукта .....	28
2.1 Техническое задание на разработку.....	28
2.2 Обоснования выбора Borland Delphi 7.0.....	31
2.3 Обоснования выбора Microsoft Office Access.....	33
2.4 Интерфейс конечного пользователя.....	33
2.5 Финансовый расчёт по внедрению технических средств информационной системы на предприятии.....	49
Заключение.....	52
Список использованных источников.....	53
Приложение А Коды программы.....	56
Приложение Б Очерк научной статьи.....	73

## ВВЕДЕНИЕ

Выпускная квалификационная работа направлена на разработку информационной системы для автотранспортного предприятия. Организации, функционирующие в течение продолжительного времени, сталкиваются с ростом объёма хранимых данных и соответственно увеличением трудовых затрат на их обработку. Так же существуют требования контролирующих органов, и налоговых служб, которые устанавливают требования наличия информационных систем.

Информатизация становится основой дальнейшего развития транспортных и логистических систем. Все большее применение находят информационные технологии для обслуживания пассажиров на транспорте общего пользования.

Цель работы – разработать информационную систему для автотранспортного предприятия.

Объектом исследования является информационная система.

Предмет исследования – особенности создания информационной системы

Задачи:

- проанализировать научную и специальную литературу по теме исследования;
- раскрыть теоретические основы информационных систем;
- спроектировать организационную и процессную модели АТП;
- разработать информационную систему предприятия с использованием Borland Delphi.

Работа состоит из введения, двух глав, заключения, списка использованных источников и приложений.

# 1 Теоретические основы информационных систем

## 1.1 Информационные системы

Информационная система – программно-аппаратная система, предназначенная для автоматизации целенаправленной деятельности конечных пользователей, обеспечивающая в соответствии с заложенной в нее логикой обработки возможность получения, модификации и хранения информации. Информационные системы разрабатываются для быстроты обработки и доступности хранимых данных. Если используются технические средства ЭВМ, то такие информационные системы называют автоматизированными.

По целевой функции информационные системы можно условно разделить на следующие основные категории: управляющие, информационно-справочные, поддержки принятия решений [40].

На информационные системы, возможно, задать такие задачи как: кадровый учет, кассовый учет доходов и расходов, хранение данных о выпуске на линию авто, складской учет и многое другое...

Принципы построения информационных систем:

– интеграция заключается в том, что данные, используются многократно;

– системность заключается в обработке данных в различных аспектах с целью получения информации, необходимой для принятия решений на всех уровнях управления;

– комплексность заключается в механизации и автоматизации преобразования данных на всех этапах работы.

Информационная система предприятия включает в себя:

- данные;
- информацию;

- знания;
- базы данных;
- базы знаний;
- программное обеспечение (пакет прикладных программ);
- экспертные системы;
- компьютерные сети (корпоративные сети);
- защиту информации.

В этом параграфе работы требовалось изучить понятия информационных систем. В процессе изучения были сделаны выводы, что внедрение информационных систем позволит улучшить эффективность работы на предприятии.

### **1.1.1 Данные**

Данные – это совокупность сведений зафиксированных на определенном носителе в форме, пригодной для постоянного хранения, передачи и обработки [44].

К определению данные относятся фиксированная информация; информация, извлеченная из данных, подвергнутая обработке, а после зафиксированная как новые данные; а также информация в виде наблюдений, которые не используются, а пока хранятся.

Данные подвергают обработке, преобразуя их из одного вида в другой с целью повышения качества информации. Обработка данных может осуществляться посредством следующих операций:

- ввод данных – накопленные данных, с целью наличия их достаточного объёма для принятия решения;
- формализация данных – преобразования данных, поступающих из разных источников, к единой форме, для повышения их доступности;

- фильтрация данных – это отбор данных, в соответствии с заданными требованиями, с целью получения достоверной и удобной информации;
- сортировка данных – распределение данных по определенному признаку, для удобства в использовании;
- архивация – способ хранения и скоростной передачи данных в простой и легкодоступной форме;
- защита данных – меры, предпринимаемые во избежание искажений или потери данных, а также от несанкционированного использования;
- транспортировка данных – организованный прием и передача данных между пользователями информационного процесса;
- преобразование данных – перевод данных из одной формы в другую или из одной организованной структуры в другую.

Преобразование информации позволяет получить данные. В базе хранятся различные данные, а система управления базой выдаёт по определенному запросу требуемую информацию.

Классификации информации:

По форме представления:

- Текстовая – в виде символов лексемы языка;
- Числовая – в виде цифр и знаков, математического действия;
- Графическая – в виде изображений, объектов, предметов, графиков, диаграмм;
- Звуковая – устная или в виде записи, аудиальным путем.

По назначению:

- Массовая – передающаяся большей части людей;
- Специальная – содержит специфический набор данных, понятна узкосоциальной группе лиц;
- Секретная – передаваемая узкому кругу лиц по защищенным каналам;
- Личная – набор сведений о какой-либо личности.

По назначению:

- Актуальная – ценная в данный момент времени;
- Достоверная – информация без искажений;
- Понятная – выражена на языке понятном и доступном получателю;
- Полная – достаточная для понимания и принятия решения;
- Полезная – полезность определяется получателем зависимости от

возможности ее применения.

По истинности:

- Истинная;
- Ложная.

В рассмотренном разделе проанализированы определения и операции по обработке данных.

### **1.1.2 База данных**

База данных (БД) – совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объектов и их взаимосвязей в рассматриваемой предметной области [42].

Существуют следующие типы организации данных и связей между ними: иерархический (в виде дерева), сетевой и реляционный.

В иерархической базе данных (рисунок 1) наблюдается упорядоченность элементов, один элемент – главный, а остальные – подчиненные. Данные распределены в некоторой последовательности. Поиск данных в такой системе трудоемкий, так как необходимо последовательно проходить предшествующие иерархические уровни.

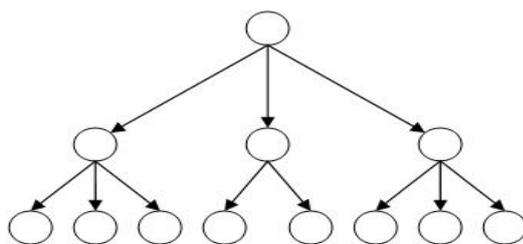


Рисунок 1 – Иерархическая БД

Сетевая база данных (рисунок 2) – это база данных, в которой помимо вертикальных связей, имеется возможность установки дополнительных горизонтальных связей. Это ускоряет процесс поиска запрашиваемых данных, так как уже не требует обязательного прохождения всех предшествующих ступеней.

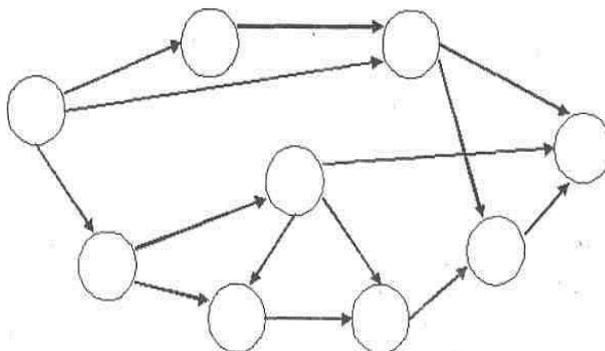


Рисунок 2 – Сетевая БД

Реляционная база данных – самый распространенный способ организации данных (рисунок 3), к нему можно отнести как иерархическую, так и сетевую базы данных. В реляционной базе данных под записью понимается строка таблицы. Столбцы – элементы файлов. Все элементы в столбце имеют одинаковый тип (числовой, символьный), а каждый столбец — неповторяющееся имя. В таблице отсутствуют одинаковые строки. Плюсами таких БД являются наглядность организации данных, скорость поиска нужной информации. Пример реляционной БД – таблица.

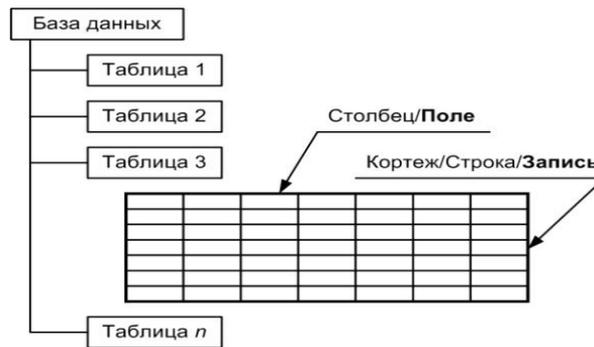


Рисунок 3 – Реляционная БД

По технологии обработки данных базы делятся на: централизованные и распределенные. Централизованная база данных хранится в памяти одной вычислительной системы. В том случае если эта вычислительная система является компонентом сети ЭВМ, возможен распределенный доступ к такой базе. Данный способ использования баз данных применяют в локальных сетях ПК. База данных, состоящая из нескольких, возможно, пересекающихся или дублирующих друг друга частей, хранимых в вычислительной сети, называются распределенными. Работа с такой базой осуществляется с использованием систем управления распределенными базами данных.

Система управления базами данных (СУБД) – это комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования БД многими пользователями [42]. Обычно СУБД различают по используемой модели данных. СУБД, использующие реляционную модель данных, называют реляционными СУБД.

Приложения, обеспечивающие автоматизацию обработки информации для прикладных задач, могут создаваться в среде или вне среды СУБД, так например Delphi, используют средства доступа к БД. Приложения, разработанные в среде СУБД, часто называют приложениями СУБД, а приложения, разработанные вне СУБД – внешними приложениями.

В данном разделе раскрыты особенности построения базы данных, основные его составляющие, понятия и классификации.

### **1.1.3 База знаний**

База знаний – это особого рода база данных, разработанная для оперирования знаниями (метаданными). База знаний содержит структурированную информацию, покрывающую некоторую область знаний, для использования кибернетическим устройством (или человеком) с конкретной целью. Современные базы знаний работают совместно с системами поиска информации, имеют классификационную структуру и формат представления знаний [37].

Любая база знаний содержит в себе базу данных в качестве составляющей, но не сводится к ней. Полноценные базы знаний содержат в себе не только фактическую информацию, но и правила ее вывода, допускающие автоматические умозаключения о вновь вводимых фактах, и как следствие, осмысленную обработку информации.

В данном разделе было рассмотрено основное понятие. Данное изучение позволило сделать вывод, что вводимые знание в базу данных позволит ускорить процесс работы программы.

### **1.1.4 Программное обеспечение (пакет прикладных программ)**

Прикладное программное обеспечение является процессом и конечным результатом создания программ, предназначенных для выполнения определенных задач (приложений), отвечающих интересам пользователя. По функциональному назначению его можно классифицировать в зависимости от поставленной задачи [48].

Прикладное программное обеспечение состоит из:

- различных пакетов прикладных программ;
- рабочих программ пользователя информационной системы.

В комплекс пакета входят задачи, направленные на реализацию целей пользователя. Каждое приложение обладает рядом способов обработки

данных, видами их представления, полнотой диагностики, все это позволяет пользователю выбрать вариант, подходящий для его конкретных целей.

Прикладное программное обеспечение бывает следующих типов:

- общего назначения;
- методо-ориентированное;
- проблемно-ориентированное;
- программное обеспечение для администрирования вычислительных процессов организации.

В качестве дополнительных модулей в пакет приложений могут включаться такие компоненты, как система экспорта-импорта файлов, калькулятор, календарь, система программирования. Информационная связь между компонентами может осуществляться с использованием компьютерной сети.

В данном разделе изучено понятие программного обеспечения, особенности прикладных программных комплексов.

### **1.1.5 Экспертные системы**

Экспертная система – программа способная заменить специалиста-эксперта в разрешении проблемной ситуации [38]. В основе экспертной системы лежит база знаний, которая накапливается в процессе эксплуатации экспертной системы. Накопление и организация знаний, самая важная составляющая часть всех экспертных систем.

Преимущества экспертных систем перед человеком-экспертом:

- отсутствие предубеждений;
- исключение поспешных выводов;
- вероятность выбора наилучшей альтернативы (оптимальное решение) из всех возможных;
- устойчивость к «помехам»;

– возможность накопления большой базы знаний, исключение потери данных.

Достоинства экспертных систем:

– возможность заменить собой экспертов, уменьшить количество людей, занятых в бизнес-процессе;

– сокращение штата сотрудников, экономия времени, снижение уровня бюрократии;

– возможность хранения данных, а также правил их вывода, касающихся той или иной предметной области, в памяти компьютера и застрахованность от чьего-либо влияния;

– экономия финансовых ресурсов и эксплуатационных расходов.

С точки зрения обработки и представления информации, основными компонентами в экспертных системах являются: интерфейс пользователя, база знаний, интерпретатор, модуль создания системы.

Специалист использует интерфейс для ввода информации и команд в экспертную систему и получает выходную из нее информацию. Ввод информации можно выполнять с помощью меню, команд, естественного языка и собственного интерфейса, а в качестве выходной информации выступает не только решение, но и необходимые объяснения. Система должна пояснить каждый шаг своих рассуждений, ведущих к решению задачи.

База знаний содержит факты, описывающие проблемную область, а также логическую взаимосвязь этих фактов и их обработку по правилам. Правило определяет, что следует делать в данной конкретной ситуации, и состоит из двух частей: условия, которое может выполняться или нет, и действия, которое следует произвести, если условие выполняется.

Обработку знаний в определенном порядке выполняет интерпретатор. Если условие, содержащееся в правиле, соблюдается, выполняется определенное действие, и пользователю предоставляется вариант решения проблемы.

Модуль создания системы необходим для формирования набора правил. Существует два подхода, положенных в основу модуля создания системы: использование алгоритмических языков программирования, или оболочек экспертных систем.

В данном разделе были рассмотрены достоинства и принцип работы экспертной системы.

### **1.1.6 Компьютерные сети**

Компьютерная сеть – это совокупность компьютеров, взаимосвязанных через каналы передачи данных для обеспечения обмена информацией и коллективного доступа пользователей к аппаратным, программным и информационным ресурсам сети [43].

Для создания компьютерной сети требуется наличие линии связи между компьютерами, специального аппаратного обеспечения, и специального программного обеспечения.

Основное назначение сети – обмен данными между пользователями ПК

Выделяют следующие виды компьютерных сетей:

- локальные, использующиеся для обмена информацией между ПК, расположенными на ограниченной территории в пределах одного здания или в рамках одной организации. При этом чаще всего используются проводные линии связи;
- корпоративные сети, функционирующие в рамках одного с предприятия, владеющего данной сетью;
- региональные, связывающие компьютеры отдельной страны или экономического региона;
- глобальные, соединяющие компьютеры разных стран.

Цель использования компьютерных сетей на предприятии, повышение эффективности в работе, защита информации от общего доступа в пределах фирмы.

Локальная сеть – объединение нескольких компьютеров, расположенных на небольшом расстоянии друг от друга (обычно в пределах одного здания) для совместного решения информационных, вычислительных и других задач [46].

Существует множество способов объединения компьютеров, но наиболее распространенные для локальной сети являются одноранговая сеть и сеть клиент-сервер (рисунок 4).

Одноранговая сеть – это объединение равноправных компьютеров. Используется чаще всего в домашних условиях, или небольших офисах.

Сеть клиент-сервер, как правило, встречается в организациях, а не в домашних условиях. В подобном типе сетей (рисунок 4) один компьютер, называемый сервером, является сердцем сети. Он хранит информацию и ресурсы и делает их доступными другим компьютерам данной сети. Остальные компьютеры, использующие сеть для получения этой информации называются клиентами.

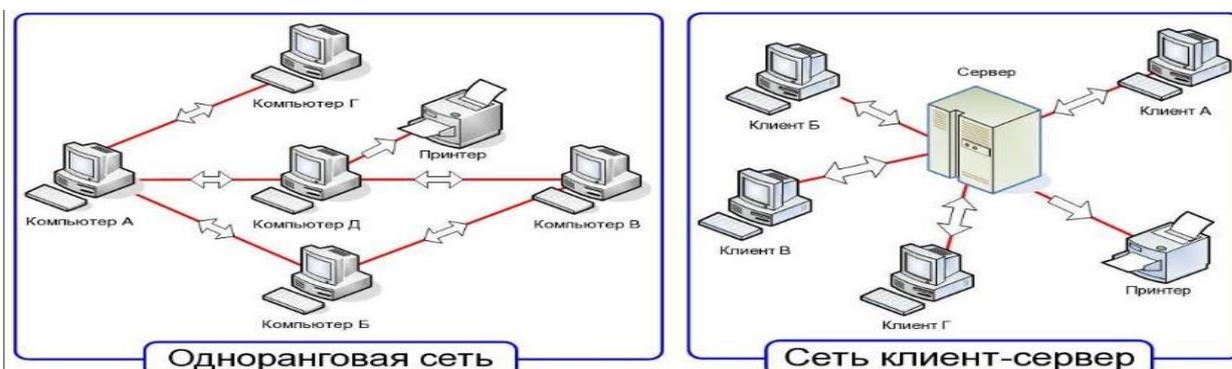


Рисунок 4 – Типы локальных компьютерных сетей

Сети клиент-сервер – наилучший вариант объединения в сеть более десяти компьютеров. Несмотря на высокую стоимость, в случаях, когда необходимо хранить большой объем информации, это самый лучший выбор.

Проанализировав литературные источники по компьютерным сетям, мы выявили их виды и изучили особенности построения, выявив тем самым наиболее удобный тип сети для использования на предприятии.

### **1.1.7 Защита информации и информационная безопасность**

Защита информации – это комплекс мероприятий, направленных на обеспечение информационной безопасности [41].

Системный подход защиты информации состоит в том, что все средства, используемые для обеспечения информационной безопасности должны работать как единый комплекс взаимосвязанных мер. Главный принцип защиты информации – принцип «разумной достаточности», заключающийся в том что: стопроцентной защиты не существует ни при каких условиях, и стремиться стоит не к максимально достижимому уровню защиты информации, а к минимально необходимому в существующих условиях и при данном уровне возможной угрозы.

В защиту информации входят: обеспечение доступности, целостности и конфиденциальности. Доступность – это возможность получить требуемую информацию. Под целостностью понимается актуальность и непротиворечивость информации, ее защищенность от разрушения и несанкционированного изменения. Конфиденциальность – это защита от несанкционированного доступа к информации.

Информационная безопасность – защищенность информации и поддерживающей инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера, которые могут нанести неприемлемый ущерб субъектам информационных отношений, в том числе владельцам и пользователям информации и поддерживающей инфраструктуры [41].

Самая распространенная защита компьютерной информации – защита на основе пароля. При реализации такой защиты запуск приложения, запрос доступа к данным сопровождается запросом пароля и последующим его сравнением с оригиналом. Пароль представлен в виде последовательность символов алфавита и специальных знаков. Последовательность должна удовлетворять ограничению на наименьшую и наибольшую длину.

Парольная защита рекомендуется при защите информации, предназначенной для узкого круга пользователей. При широком использовании программ, защищенных подобным образом, очень велика вероятность того, что пароль узнает злоумышленник, этого будет достаточно, чтобы сделать защищенное приложение общедоступным.

Хранить оригинальный пароль лучше всего в месте, доступном защищенному приложению и малодоступном для злоумышленника. Следовательно, основная задача пользователя, использующего парольную защиту, как можно лучше «спрятать» оригинальный пароль.

Способы хранения эталонного пароля:

- а) эталон хранится в самой защищенной программе;
- б) эталон хранится в отдельном специально предназначенном файле;
- в) эталон хранится в системных областях.

Например, при защите Windows – приложений разработчики хранят оригинальный пароль в системной базе данных.

Таким образом, были рассмотрены способы защиты данных, и раскрыт подробно один из наиболее распространенных методов, на основе пароля, планируемый использовать в разработке приложений.

## **1.2 Информационная система предприятия АТП**

Изучив теоретические основы информационных систем и их составляющих, приходим к выводу, что при разработке проекта для автотранспортного предприятия, нам необходимо учесть тот факт, что придется работать с большим потоком информации, которая ежедневно должна быть фиксирована и сохранена в архиве в виде данных. Все данные для удобного доступа и наглядности будут храниться в виде таблицы – базы данных.

К некоторым данным будут добавлены знания (правила) при выполнении которых, программа будет принимать самостоятельные

решения. Так, например, при медицинском осмотре, будут вводиться показатели человеческого состояния, а программа, пользуясь заданными знаниями, принимать решения о допуске или не допуске сотрудника до рабочего места. Также обстоит дело и с транспортом, контролер – эксперт при осмотре транспортного средства, будет фиксировать данные, а программа на основе этих данных выдавать разрешение о выводе авто на линию.

Проект, с целью обхвата предприятия и всех наиболее важных сторон его функционирования, будет состоять из нескольких приложений – прикладного программного комплекса, элементы которого будут связаны внутренней корпоративной сетью, с возможностью предоставления информации контролирующему органу.

Об аппаратной и организационной безопасности решения будут приниматься руководящими кадрами предприятия, но и в предлагаемом программном продукте, можно предусмотреть некоторые аспекты организации защиты данных. Возможно установление паролей на каждом из рабочих мест (в каждом приложении) или сокрытие некоторых данных, от наблюдателей. Так, например, избежать выведения на общий экран (общую базу данных) отдела кадров, информацию о документах сотрудника. И сделать возможным просмотр данных паспорта лишь по требованию специалиста.

Таким образом, проанализировав предоставленный выше материал, разработали комплекс требований и условий соответствия для разработки информационной системы автотранспортного предприятия.

### **1.3 Модель автотранспортного предприятия**

Модель – это результат отображения одной структуры изученной на другую малоизученную [48].

Строится и исследуется модель при определенных допущениях, гипотезах. Модель формируется так, чтобы наиболее полно воспроизводить те качества объекта, которые необходимо изучить в соответствии с поставленной целью. Модель для изучения должна быть проще объекта и удобнее его. Вследствие чего, для одного и того же объекта возможно существование различных моделей, классов моделей, соответствующих различным целям изучения. Необходимое условие моделирования – подобие объекта и его модели. В этом случае говорят об адекватности модели объекту-оригиналу.

Если результаты моделирования подтверждаются и служат основой для прогнозирования процессов, протекающих в исследуемых объектах, то можно говорить, что модель адекватна объекту. Адекватность зависит от цели моделирования и принятых критериев.

Любая модель обладает следующими свойствами:

- конечностью: модель отображает оригинал лишь в конечном числе его отношений;
- упрощенностью: модель отображает только существенные стороны объекта;
- приближенностью: модель отображает действительность грубо или приблизительно;
- адекватностью: модель успешно описывает моделируемую систему;
- информативностью: содержание модели достаточной информации о системе в рамках гипотез, принятых при построении модели.

Система является совокупностью взаимосвязанных объектов, которые называются элементами системы.

Состояние системы характеризуется ее структурой, то есть составом и свойствами элементов, их отношениями и связями между собой. При воздействии различных внешних и внутренних изменений система сохраняет свою целостность, пока сохраняет не изменой свою структуру. Если

структура системы меняется, то система может перестать функционировать как целое.

Системный подход применяется в любой области знания, особенно при изучении сложных систем. Суть системного подхода при построении модели отображается в виде этапов анализа:

- сложный объект система рассматривается в виде набора более простых элементов объектов;
- для каждого элемента определяется роль, выполняемая в системе;
- определяются отношения между элементами;
- устанавливается влияние параметров каждого элемента объекта на поведение системы в целом.

Полученная модель тем окажется точнее, чем тщательней проводился анализ системы. Но модель должна отображать характеристики системы лишь с необходимой степенью точности. Точность определяется целью и подразумевает осмысленный отбор необходимых черт, отображающих модель.

В данном разделе раскрыто понятие модели, и важность системного подхода при ее создании. Это поможет определить структуру предприятия и организовать организационную и процессную модели. Ниже показана разработка организационной и процессной модели автотранспортного предприятия.

### **1.3.1 Организационная модель**

Организационная модель – это принципы формирования подразделений, указания полномочий и наделения функциональностью [39].

Структуры управления на многих современных предприятиях были построены в соответствии с принципами управления:

- иерархичность уровней управления, при котором каждый вышестоящий уровень подчиняет;

- соответствие полномочий и соответствие работников управления месту в иерархии;
- разделение труда на заданные функции и классификацию работников по выполняемым обязанностям;
- формализация и стандартизация деятельности, обеспечивающая однородность выполнения работниками своих прямых обязанностей и системность различных задач;
- квалификационный отбор.

Организационная структура (рисунок 5), разработанная в соответствии с этими принципами называется иерархической.

Преимущества линейных структур:

- взаимно связь внутри функций и в соответствующих им подразделениях;
- система единоначалия – сосредоточения в одних руках всё управление;
- выраженная ответственность;
- реакция исполнительных подразделений на прямые указания руководство.

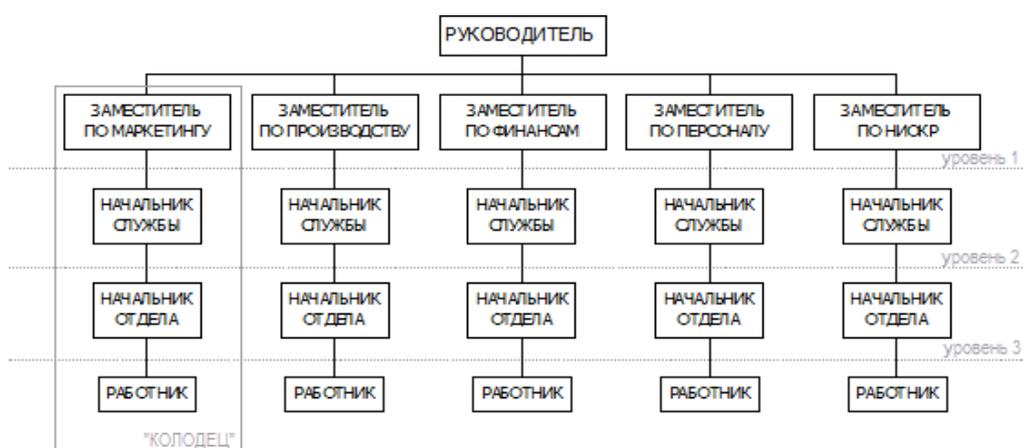


Рисунок 5 – Организационная модель

На малом автотранспортном предприятии численность персонала сокращено, и каждый сотрудник выполняет по несколько функций, так организационная модель (рисунок 6) офисной ступени автотранспортного предприятия будет выглядеть следующим образом:



Рисунок 6 – Организационная модель АТП

Во главе стоит директор, ему подчиняется заместитель директора, далее инспектор по кадрам, медицинский работник он же выполняющий функции диспетчера-нарядчика, кассир-финансист, и механик по совместительству заведующий складом запчастей.

На основе созданной модели, ярко выделены связи отчетностей. Для разработки информационной системы требуется создание четырех прикладных приложений формирующих отчет и отправляющих его по внутренней корпоративной сети заместителю директора.

### 1.3.2 Процессная модель

Любая современная компания является сложной организационной структурой с большим объемом документации, проходящих как внутри предприятия, так и между предприятием и другими физическими и юридическими лицами.

Деловой процесс — последовательность действий, называемых подпроцессами, направленная на получение заданного результата, представляющего ценность для потребителя [45].

Результат (выход) делового процесса — то, ради чего осуществляется деловой процесс — некоторый результат, удовлетворяющий заданным требованиям [45].

Входы делового процесса — ресурсы (материальные, информационные, человеческие), необходимые для выполнения и получения результата процесса, которые потребляются или преобразовываются в процессе его выполнения [45].

Процессная модель позволяет просмотреть деятельность предприятия как единую систему деловых процессов, каждый из которых протекает во взаимосвязи с другими процессами или внешней средой.

При таком подходе любую компанию в целом можно рассматривать как систему, потребляющую некоторые ресурсы на входе, затем преобразующую их внутри себя и выдающую определенный результат на выходе. Полная система подразумевает собой комбинацию и последовательность процессов, разработка которых даёт получение результата, позволяющего достичь поставленных задач.

В рисунке 7 продемонстрирована простейшая процессная модель организации:



Рисунок 7 – Процессная модель

В каждой организации ежедневно протекает множество процессов. Чтобы систематизировать и упорядочить бизнес-процессы, протекающие на предприятии, их принято делить на следующие основные группы:

а) Основные – генерируют доходы компании. Примерами процессов закупки, продажи и производства продукта; ее складирования и транспортировки.

б) Обеспечивающие – поддерживают инфраструктуру компании. В эту категорию входит техническое, информационное и административно – хозяйственное обеспечение; обеспечение безопасности, социальной сферы и финансовых операций; консультационное и аудиторское обеспечение.

в) Управления – управляют компанией. К этой категории относят такие процессы как процессы управления финансами, маркетингом, персоналом и логистикой; экономическое, стратегическое и организационное управление, а также управленческий учет.

г) Развития – развивают компанию. Представляют инвестиционные виды деятельности, нацеленные на прибыль от сегодняшних вложений в долгосрочной перспективе.

На основе рассмотренного материала, создадим процессную модель (рисунок 8) автотранспортного предприятия.

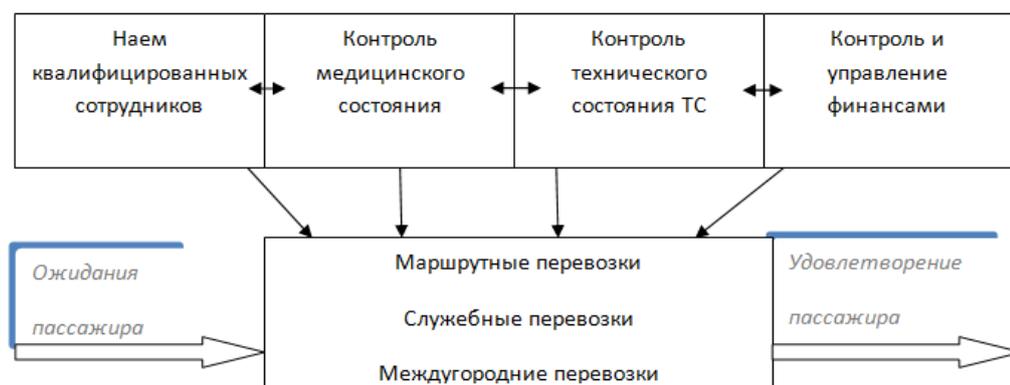


Рисунок 8 – Процессная модель АТП

Есть пассажир, он же заказчик с определенными требованиями и ожиданиями:

- все автобусы должны быть на линии, простой пассажира на остановке не более двух минут;

- пассажирский транспорт в хорошем техническом состоянии, соответствует условиям максимальной безопасности. Чистый и комфортабельный;

- перевозку осуществляет опытный водитель, со стажем, вменяемый (не находящийся в алкогольном опьянении или под воздействием психотропных средств), с хорошими показателями самочувствия (т.е. прошедший предрейсовый медицинский осмотр);

- обилечивает и следит за порядком в салоне честный и вежливый кондуктор.

То, какой в действительности результат получит пассажир на выходе, зависит от системности и качества проведения внутренних процессов.

- наем квалифицированных сотрудников со стажем, с хорошими рекомендациями (ответственный инспектор по кадрам);

- проведение ежедневного предрейсового осмотра транспортного средства. Осуществление своевременного периодического ремонта и замена всех расходников авто (ответственный механик);

- проведение ежедневного предрейсового медицинского осмотра водителей, а также послерейсового для оценки посильности условий труда для сотрудника (ответственный медицинский работник);

- контроль и управление финансами, (ответственный кассир-финансист).

Для того чтобы результат на выходе максимально соответствовал ожиданиям пассажира, предприятие нуждается в слаженной организованной информационной системе, которая бы помогала персоналу в осуществлении процессов, и могла хранить большие объемы данных.

В данной главе были рассмотрены теоретические основы информационной системы автотранспортного предприятия. Данное изучение позволило сделать ряд выводов:

1) внедрение информационных систем позволит улучшить эффективность работы на предприятии,

2) выявить удобный тип локальной сети для использования на автотранспортном предприятии,

3) выявили наиболее эффективный способ защиты – закрытие доступа к программе с помощью пароля,

4) сопоставив теоретические основы информационных систем на предприятии, создали организационную и процессную модели для лучшего представления организации процессов на автотранспортном предприятии.

## **2 Разработка программного продукта**

### **2.1 Техническое задание на разработку**

Наименование программного продукта: «Разработка информационной системы для АТП г. Лесосибирска»

Область применения:

– Программный продукт разработан для персонала предприятия заместитель директора, заведующий отделом кадров, диспетчер-нарядчик, медицинский работник, механик-кладовщик, кассир.

– Разработчик данного продукта – студент ЛПИ – филиала СФУ, Сахаров А. С.

– Заказчик программного продукта – ООО «АВТОБУСНОЕ СОТРУЖЕСТВО» Красноярский край, г. Лесосибирск, улица Горького 116, здание 3 Помещение 7. Генеральный директор Шефер Владимир Александрович.

Факторы эффективности и качества программы:

1) Социальный фактор. Программная разработка проста в обучении и предназначена не только для опытных, но и на начинающих пользователей, работающих под Windows. Просто интуитивно понятный интерфейс даёт возможность работать с программным продуктом без специальной и предварительной подготовки.

2) Аналоговое или превосходящее состояние на рынке программного оборудования данного профиля. Технология разработки прикладных программ в визуальных средах программирования ставит ее интерфейс универсальным и совместимым с ОС Windows.

3) Экономические факторы. Программный продукт соответствует наилучшему соотношению цены и предлагаемых ею характеристик и займет

свою нишу на рынке бюджетных программ. Основными заказчиками станут системные инженеры или пользователи, у которых нет возможности закупить дорогой программный продукт.

Цели разработки программы:

- разработка программного продукта преследует ряд экономическо-технических целей;
- разработка программного продукта, необходимого для упрощения работы персонала предприятия;
- разработка интуитивно понятного интерфейса программы с удобным и универсальным для экономии времени;
- разработка дешевой альтернативы существующим в настоящее время дорогим программам данного типа.

Требование к функционалу:

- а) Отображать подробную информацию о выполняемой работе;
- б) Сохранять таблицу;
- в) Предоставлять отчёт.

Организация входных и выходных данных является отображение на монитор полной информации в виде таблиц.

Требования к надежному функционированию:

- а) Бесперебойная работа программ при нормальной работе ЭВМ. Если во время функционирования программы произошел сбой в работе ЭВМ, восстановление нормальной работы программы должно производиться после:
  - 1) запуска исполняемого файла программы;
  - 2) перезагрузки операционной системы;
  - 3) повторного выполнения действий, потерянных до последнего сохранения информации в файл на магнитном диске.
- б) Требование надежности программы должно соответствовать технологии программирования, предусматривающей:
  - проверку исходных текстов программы;

- автоматическое тестирование модулей программы;
- тестирование сопряжений модулей программы;
- комплексное тестирование программы.

Требование к эксплуатации и хранению:

- Программный продукт должен храниться в виде копий на лазерном компакт диске и обязательно эталонная версия программы.
- Минимальные требования к техническим устройствам:
  - а) Pentium 3 – 800 МГц;
  - б) 64Мб RAM;
  - в) клавиатура;
  - г) мышь;
  - д) монитор;
  - е) принтер.

Для решения задачи использовать частично автоматизированный метод.

В ходе разработки программного продукта использовать следующие программные средства:

- Borland Delphi v7.0 и выше;
- Microsoft Office Access.

Виды испытаний:

Проверка документации к разработанному продукту проводится руководителем автотранспортного предприятия ООО «Автобусное содружество».

В процессе разработки программы испытания и тестирование должны проводиться самим разработчиком:

1) С использованием контрольных тестов, которые позволяют добиться проверки правильности работоспособности и взаимной совместимости максимального числа функций и операторов программы или модуля при минимальных затратах временных и финансовых ресурсов.

2) Путем пошагового исполнения программы или модуля в соответствии с набором тестовых примеров и сравнения полученных в процессе тестирования значений с контрольными значениями тестовых примеров.

Общие требования к приёмке:

Приёмка программы и ввод в тестовый режим будет, осуществляется руководителем автотранспортного предприятия ООО «Автобусное содружество».

Таким образом, определены требования к предстоящей разработки программного продукта его характеристики и функционал. В соответствии предъявляемых требований разрабатывается информационная система для автотранспортного предприятия.

## **2.2 Обоснования выбора Borland Delphi 7.0**

Модуль формирования структуры разработан в среде визуального программирования Delphi 7.0 и предназначен для функционирования в операционной системе Windows 7.

На этапе подготовки разработки программного продукта проанализировано несколько возможных вариантов реализации данного модуля.

Сформулируем основные требования, по которым производился подбор среды программирования для разработки данного модуля.

а) Разработка интуитивно понятного интерфейса для создания удобства при работе с программой.

б) Работа модуля должна выполняться с максимально возможной скоростью. Не приемлемы ситуации, в которых пользователю продолжительное время придется ожидать окончания работы модуля.

в) Поддержка длинных имен файлов.

- г) Простота в установке и использовании модуля.
- д) Дешевизна на разработку модуля.

В процессе анализа имеющихся средств программирования, исходя из перечисленных требований был выбран один вариант разработки данного модуля с использованием системы визуального программирования Borland Delphi 7.0 для операционной системы Windows 7. Данное заключение основывалось на следующем.

Среда визуального программирования Delphi 7.0 эффективно функционирует в операционной системе Windows 7 и даёт программисту возможность реализовать все достоинства графического интерфейса этой системы. С учётом использования подавляющим большинством пользователей персональных компьютеров, работающих сегодня в среде операционных систем семейства Windows, то этот интерфейс позволяет проводить более удобную и эффективную работу с программой.

Многие среды разработки приложений для Windows генерируют код-полуфабрикат, который не может быть выполнен процессором без дополнительной трансляции во время работы самой программы, что снижает производительность компьютера.

Delphi использует собственный компилятор и компоновщик и генерирует в полном объёме машинный код. Такая реализация лишена лишней работы, что делает программы, разработанные на Delphi 7.0, наиболее эффективными.

Delphi – мощная современная система программирования, имеющая многочисленные приложения везде, где сегодня применяются компьютеры – от инженерных и научных расчетов до автоматизации управленческой деятельности.

Таким образом, выбранная платформа, удовлетворяет поставленным требованиям, поэтому выбор был остановлен на данной системе программирования.

## **2.3 Обоснования выбора Microsoft Office Access**

Приложение Microsoft Access является высокопроизводительной 32-разрядной системой управления реляционной базой данных (далее СУБД).

Достоинством СУБД является ввод новых данных в таблицы. Существует три критерия:

1) СУБД разрабатываются с целью обеспечения эффективной обработки больших объёмов информации.

2) СУБД может легко связывать две таблицы так, что для пользователя они будут представляться одной таблицей.

3) СУБД минимизируют общий объём базы данных. Для этого таблицы, содержащие повторяющиеся данные, разбиваются на несколько связанных таблиц.

Преимущество Access является в его эффективном использовании непрофессиональными пользователями.

Таблицы. В базе данных информация хранится в виде двумерных таблиц. Одновременно могут быть открыты 1024 таблицы.

В состав Access входит множество мастеров, строителей и надстроек, которые позволяют упростить процесс разработки объектов базы данных.

Исходя из анализа, выбранная СУБД, удовлетворяет поставленным требованиям, поэтому выбор был остановлен на данной системе программирования.

## **2.4 Интерфейс конечного пользователя**

Приступая к разработке программы, предусмотрен доступ с запрашиванием пароля (рисунок 9) для каждого создаваемого приложения.

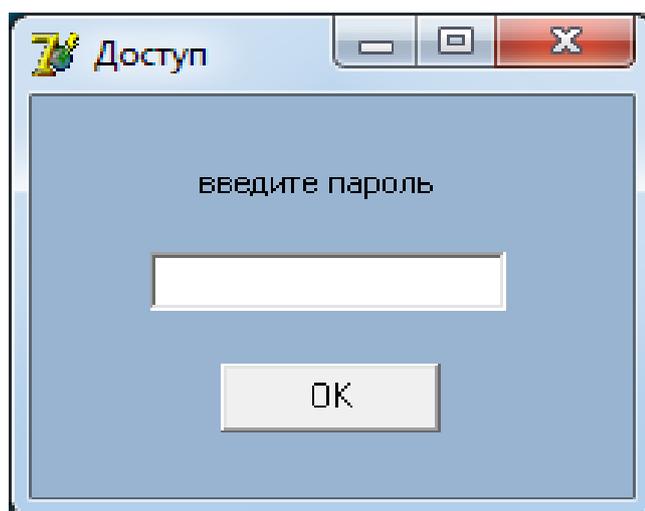


Рисунок 9 – Доступ к приложениям

Если пароль введен верно, далее следует переход непосредственно к приложению, в противном случае на экране будет выведено (рисунок 10) уведомляющее сообщение о неправильности пароля.

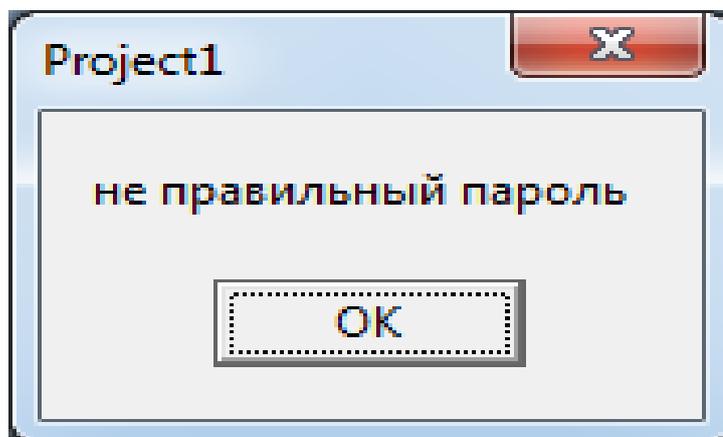


Рисунок 10 – Сообщение о не верности пароля

Программа для отдела кадров (рисунок 11) включает в себя базу данных о сотрудниках с краткой информацией о них, и возможность поиска работника по фамилии. Во время поиска если сотрудник с данной фамилией имеется в базе, то программа просто переведет на него курсор и выделит строку.

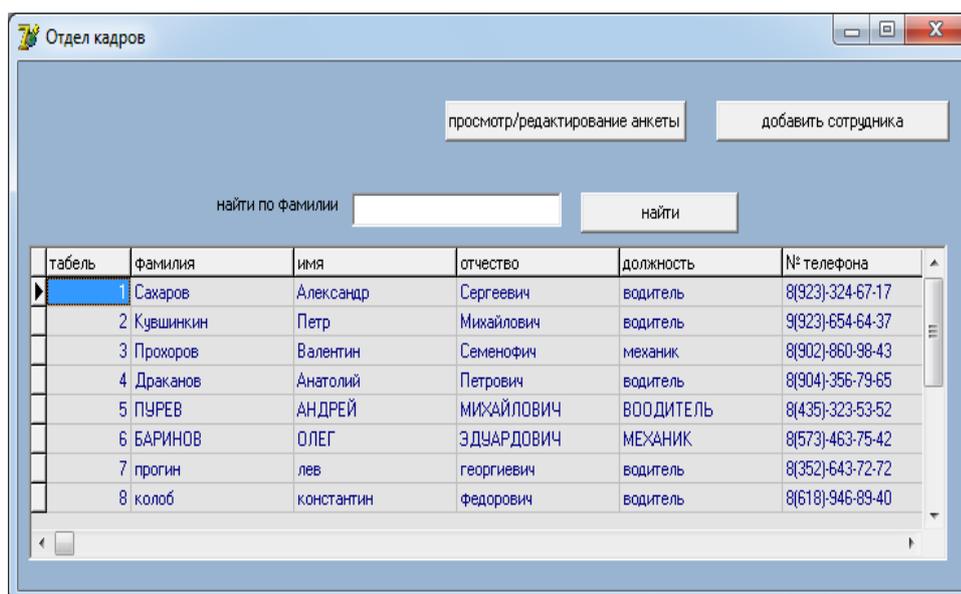


Рисунок 11 – Интерфейс приложения Отдел кадров

Во время нажатия кнопки (рисунок 12) «просмотра/редактирования анкеты», инспектор по кадрам может просмотреть более подробную информацию по конкретному сотруднику. В расширенной базе предусмотрены следующие пункты:

- табель;
- ФИО;
- дата рождения;
- количество детей;
- адрес проживания;
- телефон;
- табельный номер;
- занимаемая должность;
- дата приема на работу;
- паспортные данные;
- ИНН;
- номер медицинского страхования;
- номер водительского удостоверения;

- военнообязанный.

Кнопка «добавить сотрудника», запускает эту же анкету, но с чистым бланком для заполнения и последующего его сохранения в базе данных предприятия.

Анкета сотрудника

**личные данные**

фамилия Кучешкин  
имя Петр  
отчество Михайлович  
дата рождения 03.05.1976  
адрес проживания Белинского 7 кв 45  
номер телефона 9(923)-654-64-37  
количество детей 3

**должность**

табельный номер 2  
должность водитель  
дата приема на работу 30.07.2001

**документы**

серия паспорта 25 52  
ИНН 134545656576  
номер водительского удостоверения 16 58 90-955  
номер паспорта 155464  
СНИЛС 134-665-476-80  
военнообязанный

сохранить

Рисунок 12 – Интерфейс окна анкеты сотрудника

Программа для кассира (рисунок 13) будет представлять собой две таблицы, первая по приходу с колонками:

- сумма;
- источник доходов.

Вторая по расходу с колонками:

- сумма;
- направление расходов.

Данные таблицы заполняются с учетом даты, и возможностью просмотра данных за тот или иной день. Одним из важных аспектов является итоговый подсчет сумм.

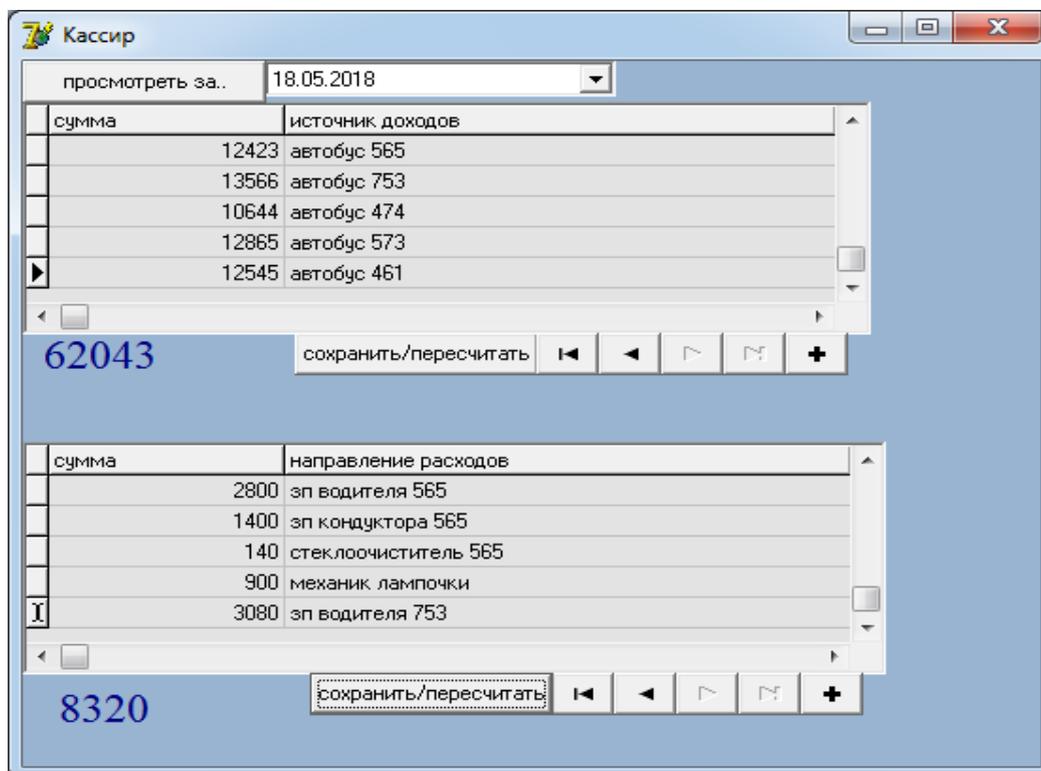


Рисунок 13 – Интерфейс приложения Кассир

Приложение для медицинского работника/диспетчера-нарядчика (рисунок 14) будет состоять из нескольких пунктов:

- верхнего меню предрейсовый медицинский осмотр;
- присвоение наряда диспетчером-нарядчиком;
- послерейсовый медицинский осмотр.

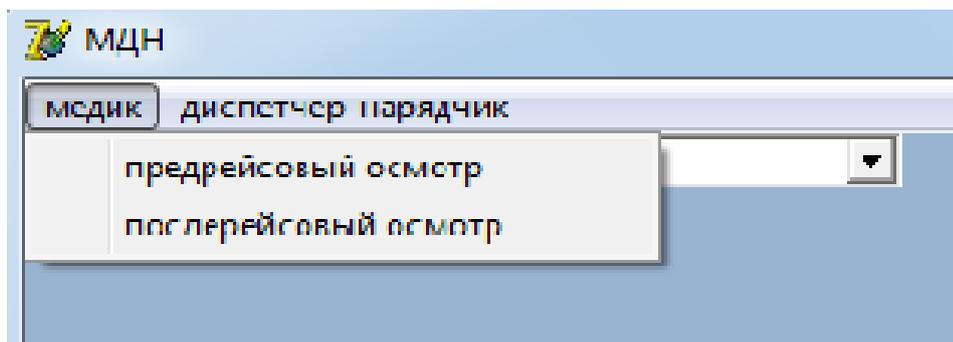


Рисунок 14 – Верхнее меню приложения МДН

Предрейсовый осмотр (рисунок 15) представлен базой данных медицинских показателей сотрудника, прошедшего медицинский осмотр, с присвоением ему номера путевого листа.

 The image shows the 'МДН' application window with the 'предрейсовый осмотр' option selected. Below the menu, there is a date selector set to '04.06.2018' and a 'начать осмотр' button. A table displays the following data:
 

time	№	Фамилия водителя	визуальное состояние	t тела	давление верхнее	давление нижнее	пульс	тест на алкоголь
04.06.2018 7:44:57	101	евсеев	удовлетворительное	36,6	110	70	77	отрицательный
04.06.2018 7:45:21	102	кортышев	удовлетворительное	36,6	120	80	80	отрицательный
04.06.2018 7:45:49	103	пырев	удовлетворительное	36,6	110	70	85	отрицательный

Рисунок 15 – Интерфейс приложения МДН

При нажатии кнопки «начать осмотр» запускается анкета (рисунок 16), куда вносятся основные исследуемые показатели состояния здоровья работника. Осмотр проводится по категориям:

- визуальное состояние;
- температура;
- давление;
- пульс;
- тестирование на наличие алкоголя.

В том случае, если данные показатели не соответствуют норме, программа не вносит данного сотрудника в базу, присвоение номер путевого листа также не осуществляется.

Рисунок 16 – Анкета медицинского осмотра

Если допуск получен, в базы данных диспетчера нарядчика и после рейсового осмотра будут перенесены данные (рисунок 17) о фамилии и номере путевого листа.

time	№	Фамилия водителя	№ автобуса	№ карточки маршрута	фамилия кондуктора	серия билетов	№ билетов
04.06.2018 7:44:57	101	есеев					
04.06.2018 7:45:21	102	кортышев					
04.06.2018 7:45:49	103	пырев					

Рисунок 17 – Интерфейс вкладки диспетчер-нарядчик

На рисунке 18 показаны оставшиеся параметры, которые заполняются непосредственно в базе данных:

- закрепление транспортного средства;
- присвоение номера карточки маршрута;
- снабжение экипажем (кондуктором);
- серия и номер билетов (по необходимости).

time	№	Фамилия водителя	№ автобуса	№ карточки маршрута	фамилия кондуктора	серия билетов	№ билетов
04.06.2018 7:44:57	101	евсеев	с 794 ва	24	выбина	ав 456	763822
04.06.2018 7:45:21	102	кортышев	ы 658 рп	46			
04.06.2018 7:45:49	103	пырев	е 805 ь	41	лакова	лп 865	593765

Рисунок 18 – Заполненная база данных «диспетчера-нарядчика»

В конце рабочего дня осуществляется медицинский осмотр и дается заключение о состоянии работника (рисунок 19). Последовательность выезда на линию, не совпадает со сходом, поэтому целесообразно снабжение базы данных поиском по фамилии. После выбора испытуемого водителя, и нажатия на кнопку осмотра открывается анкета аналогичная предрейсовому досмотру.

time	№ путевого/билетно учетного	Фамилия водителя	визуальное состояние	t тела	давление верхнее	давление нижнее	пульс	тест на алкоголь
		101 евсеев	.					
		102 кортышев	.					
		103 пырев	.					

Рисунок 19 – Интерфейс вкладки «послерейсовый осмотр»

Заполненная база данных (рисунок 20) в конце рабочего дня будет выглядеть следующим образом.

time	№ путевого/билетно учетного	Фамилия водителя	визуальное состояние	t тела	давление верхнее	давление нижнее	пульс	тест на алкоголь
04.06.2018 8:06:18		101 евсеев	удовлетворительное	36,6	120	80	80	отрицательный
04.06.2018 8:04:55		102 кортышев	удовлетворительное	36,6	130	80	83	отрицательный
04.06.2018 8:07:57		103 пырев	удовлетворительное	37,0	135	80	90	отрицательный

Рисунок 20 – Заполненная база данных послерейсовый осмотр

Приложение механика представлено (рисунок 21) двумя вкладками выпуск на линию и склад.

База данных выпуск на линию содержит информацию о транспортных средствах, допущенных программой к выезду и закрепленных за ними водителях.



№ путевого листа	дата	гос № ТС	Фамилия водителя	пробег	прохождение контроля	фамилия контролера
101	04.06.2018	с 794 в	евсеев	345375	допущен	зюев
102	04.06.2018	ы 658 р	кортышев	650850	допущен	зюев
103	04.06.2018	е 805 ь	пырев	439837	допущен	зюев

Рисунок 21 – Интерфейс вкладки выпуск на линию

Осмотр авто (рисунок 22) осуществляется по категориям: визуальное состояние, исправность тормозной системы, исправность рулевого управления, исправность световых приборов, состояние колес/шин, исправность двигателя, состояние тахографа, исправность звуковых сигналов.

По результатам осмотра программа принимает решение о допуске или не разрешении выезда на линию транспортного средства.

Предрейсовый осмотр ТС

**осмотр ТС**

- визуальное состояние
- исправность тормозной системы
- исправность рулевого управления
- исправность световых приборов
- состояние колес/шин
- исправность двигателя
- наличие рабочего тахографа
- исправность звуковых сигналов

пробег

ФИО контролера

**допуск**

№ путевого листа

Гос №ТС

ФИО водителя

сохранить

Рисунок 22 – Анкета осмотра транспортного средств

Вкладка склад представлена четырьмя таблицами (рисунок 23). Главная база – склад, верхний левый угол, содержит информацию обо всех имеющихся у механика запчастях их количестве, и основных характеристиках. Формируется данная таблица посредством приходов и расходов, две таблицы слева. Нижний правый угол – таблица поиска по названию запчастей.

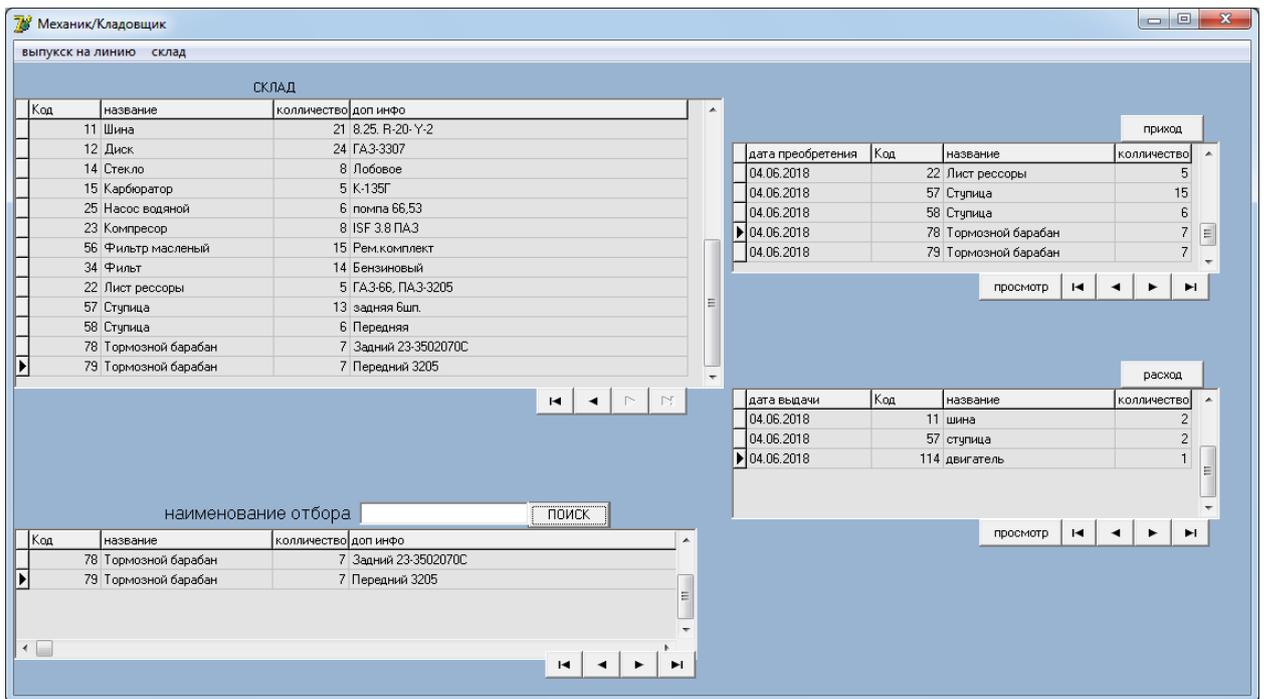


Рисунок 23 – Интерфейс приложения «Механик/кладовщик»

Информация о приходе – база данных с краткой информацией:

- дата прихода;
- код товара;
- название товара;
- количество.

Расширенный бланк приходов, заполняемый по каждому товару, содержит колонки:

- код;
- название;
- количество;
- дополнительная информация.

Цена за единицу, с возможностью просмотра его в последующем. (рисунок 24). Если товар с данным кодом на складе уже имеется, то новой записи не создается, а просто суммируется количество.

Приход

04.06.2018

код товара 11

название Шина

количество 23

доп инфо 8.25. R-20-Y-2

цена за ед. 6700

сохранить

Рисунок 24 – Бланк прихода запчастей

На рисунке 25 отображена база данных расход, также содержит краткую информацию по пунктам: дата, код, название, количество,

Расширенный бланк с возможностью просмотра по колонкам: код название, наименование, количество, выдано кому, для транспортного средства с номером.

Расход

04.06.2018

код товара 11

название шина

количество 2

выдано КОМУ евсееву

для ТС гос № 725

сохранить

Рисунок 25 – Бланк расхода запчастей

В том случае если количество товара требуемого для расхода меньше имеющихся на складе, их количество будет отнято, если количество совпадает, то товар со склада будет удален. Если же запрашиваемое

количество превышает имеющийся на складе резерв, будет выведено уведомляющее (рисунок 26) об этом инциденте и отобразится сообщение – «товар с данным кодом, в указанном количестве на складе отсутствует».

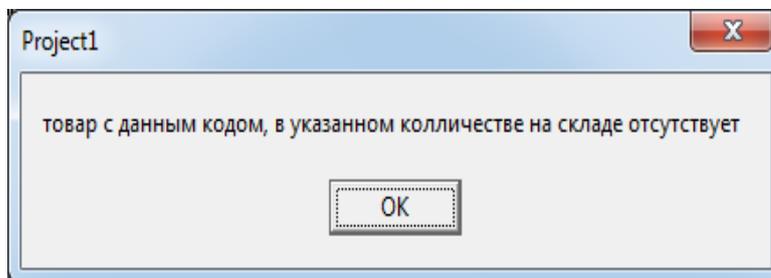


Рисунок 26 – Сообщение о не соответствии

Приложение заместителя директора включает в себя получение некоторых общих сведений, отчетов по работе от нижестоящих сотрудников, просмотреть которые возможно при переходе по соответствующей кнопке. Главная форма приложения заместителя директора представлена на рисунке 27.

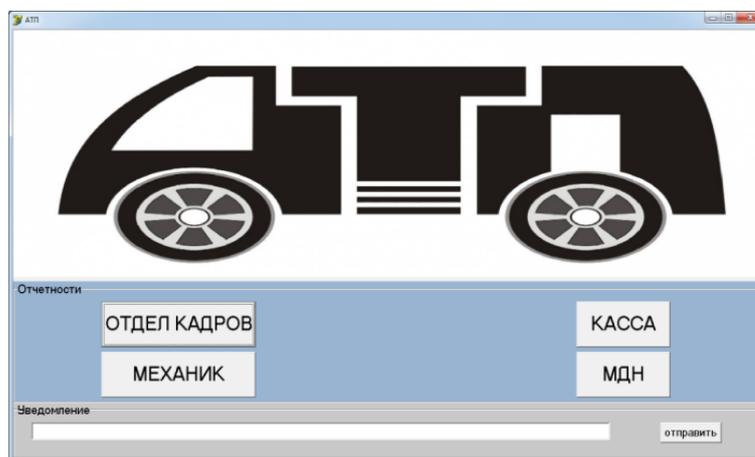


Рисунок 27 – Интерфейс главного окна приложения для заместителя директора

Также предусмотрена возможность отправки уведомлений (рисунок 28) всем и каждому в отдельности.

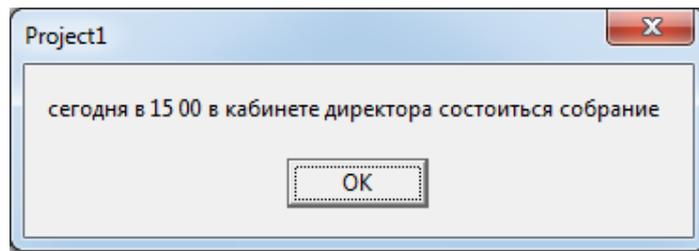


Рисунок 28 – Уведомление для сотрудников предприятия

При переходе на вкладку отдел кадров, доступной является:

- база данных со списком сотрудников;
- информация о занимаемой ими должности;
- номер телефона для связи.

Организована возможность поиска по фамилии (рисунок 29). Если инспектором по кадрам был нанят новый сотрудник, то заместителю директора автоматически приходит уведомление об этом факте.

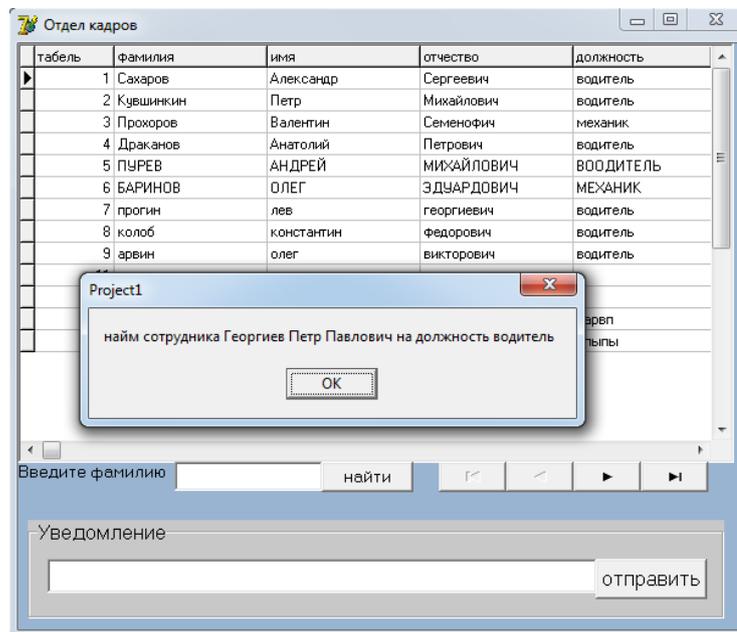


Рисунок 29– Окно отдела кадров, с уведомляющим сообщением

На вкладке касса (рисунок 30), имеется вся информация по приходам и расходам, доступная для детального просмотра заместителю директора.

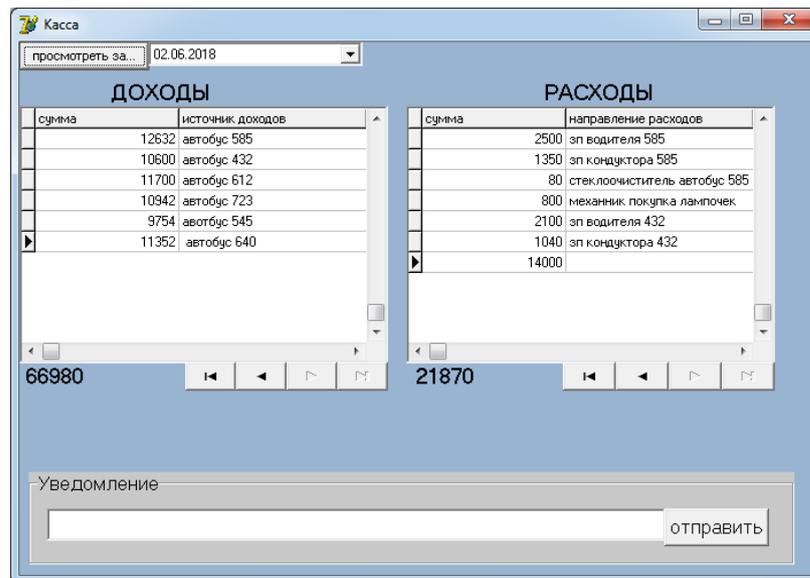


Рисунок 30 – Окно касса в приложении для заместителя директора

Диспетчер-нарядчик (рисунок 31), после завершения выдачи нарядов и выпуска авто на линию, нажимает кнопку отчет, и тогда у заместителя директора появляется таблица с информацией о том, какие автобусы вышли на линию, кто водитель и кондуктор, а также уведомляющее сообщение о количестве таких экипажей.

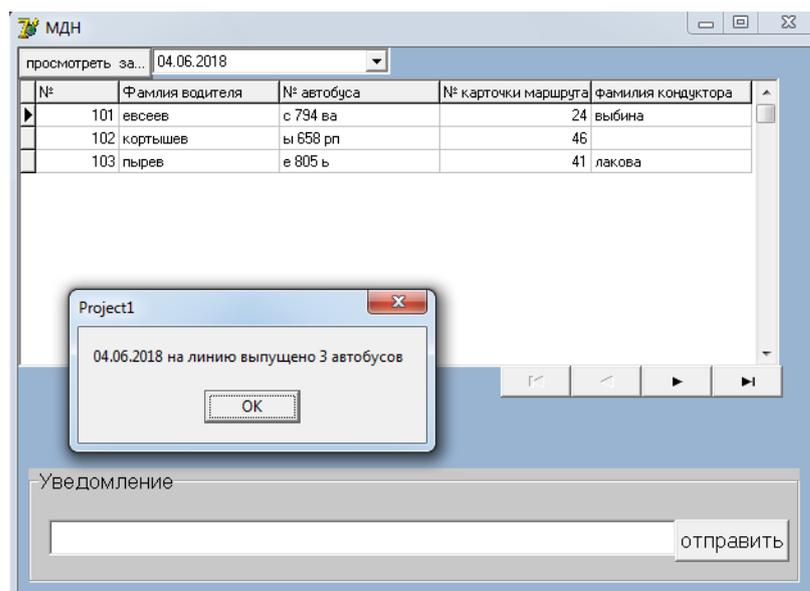


Рисунок 31 – Окно МДН с уведомляющим от диспетчера сообщением

Перейдя по кнопке «механик», можно просмотреть имеющиеся на складе запчасти и их количество (рисунок 32), а также просмотреть информацию о предрейсовом осмотре контролером-экспертом, о вышедшем автотранспорта на рейс.

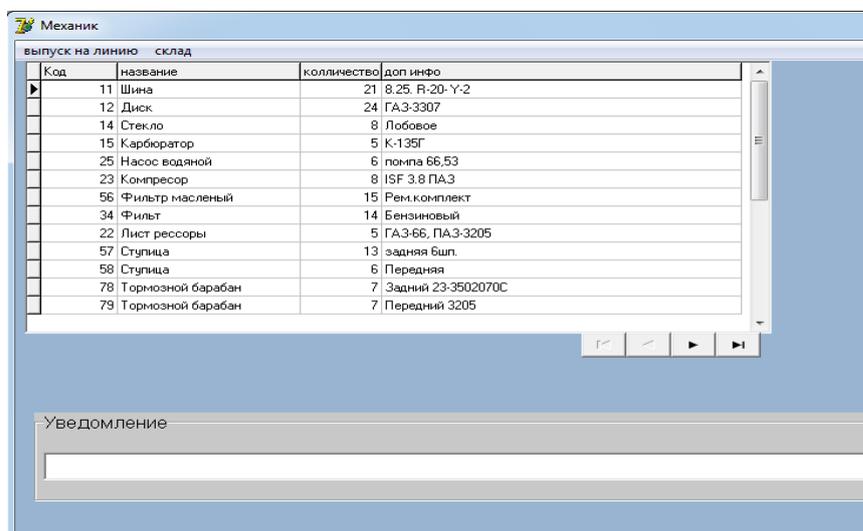


Рисунок 32 – Вкладка «склад» в разделе механик

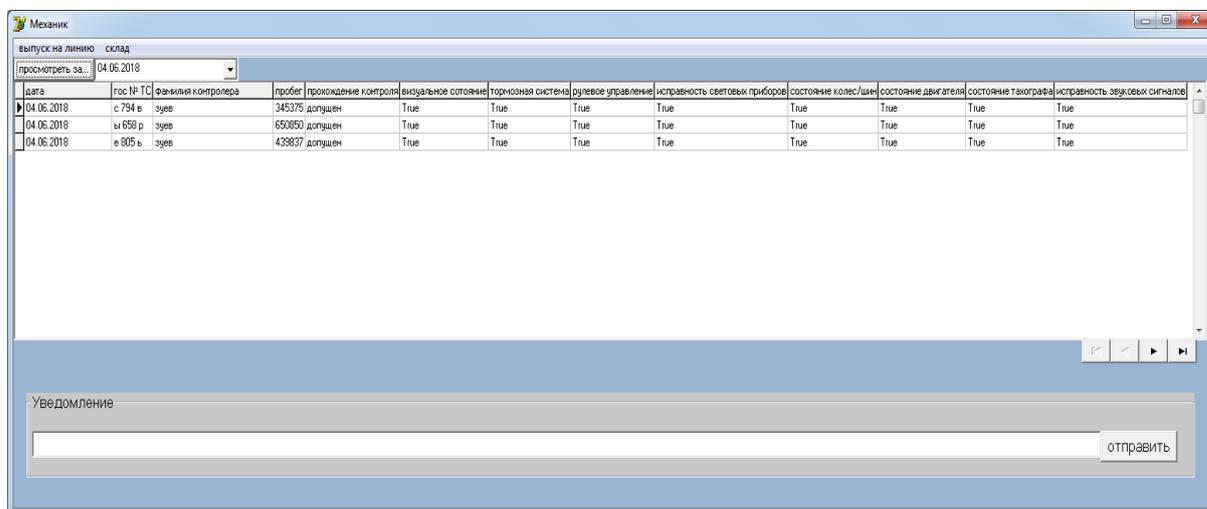


Рисунок 33 – Вкладка «предрейсовый осмотр авто» в разделе механик

В данном разделе рассмотрен готовый интерфейс с его полным описанием функционала. Все технические требования к визуализации программы выполнены.

## 2.5 Финансовый расчёт по внедрению технических средств информационной системы на предприятии

Заданное количество персональных компьютеров:

- для заместителя директора автотранспортного предприятия – 1 шт.;
- для механик/кладовщик – 1 шт.;
- для отдела кадров – 1 шт.;
- для диспетчера/медицинского работника – 1 шт.;
- для кассира – 1 шт.

Итоговое количество ПК – 5 шт.

В таблице 1 отображен перечень товара для руководства и других работников АТП.

Таблица 1 – Перечень товара

№	Наименование товара	Характеристики товара	Количество в шт.
1	Процессор	Intel Pentium 4 №531 3.0GHz /1Mb/800/socket-775 EMT64 Box	5
2	Системная плата	Intel D945PAWLK(i945P, LGA775, dual DDRII667, PCIX16, HD Audio, 2PCIX1, GbtLan)	5
3	Корпус	Midi Tower ATX P4 400W Proxima Prestige V-159 silver-blue(2USB, audio, 1394) TAC2.0 Tube 20+4	5
4	Оперативная память	DiMM 512Mb DDRII SDRAM (PC5300) 667Mhz 240-pin Hynix	5
5	Жесткий диск	160Gb Seagate SATA-300 (7200) (8Mб)	5
6	Видеокарта	PCI-E, 256Mb GeCude Radeon X1600Pro DDR2 DVi TV-out	5
7	Флоппи дисководы	FDD 3.5" silver	5
8	DVD-RW	DVD-RW Pioneer DVR-111 (dual) (DVD-RAM, 8xDL/R9, 16x DVD-R, 6x DVD-RW, 8x DVD+RW	5
9	Монитор	ЖК LG 1752TR silver (1280x1024@75Гц;300кд/м;1600:1: 2мс;160°/160°;DVI;)	5
10	Клавиатура	Oklick 510S PS/2 black	5
11	Мышь	NetScrol 110 оптическая (USB), 800dpi, Black	5
12	Факс-модем	Acorp Sprinter@ADSL W422G(ADSL2+WiFi)	5
13	колонки	Genius SP-F120	5

В таблицы 2 отображен расчет затрат на персональные компьютеры.

Таблица 2 – Расчёт затрат

№	Наименование товара	Стоимость 1 ед. товара, руб.
1	Процессор	3658
2	Системная плата	3528
3	Корпус	1563
4	Оперативная память	1896
5	Жесткий диск	1918
6	Видеокарта	2231
7	Флоппи дисководы	212
8	DVD-RW	1200
9	Монитор	7079
10	Клавиатура	495
11	Мышь	127
12	Факс-модем	2213
13	колонки	289
Итого		132045

Расчет затрат на принтеры: Samsung SCX-4200 цена одного аппарата 6295 рублей, стоимость пяти принтеров Canon LBP 3000 равняется 22175 рублей.

Затраты на приобретение оптоволоконного определяем, учитывая, что автоматические рабочие места располагаются в одном здании на небольшой удаленности. Затраты на покупку и монтаж кабеля, а также на приобретение стыковочного приспособления будут составлять 15000 рублей.

Общая сумма предполагаемого перечня прикладных программ составляет 15000 рублей.

В итоге реализация информационной системы в рассматриваемом предприятии, с учетом введения должности программиста с окладом 6000 рублей будет стоить 190220 рублей.

В данной главе была рассмотрена разработка программного продукта «Разработка информационной системы для АТП г. Лесосибирска» для автотранспортного предприятия. Так же получены следующие решения:

- а) утверждение технических требований;
- б) выбор и обоснование программных сред;

в) разработка интерфейса программы;

г) создание финансового отчёта.

В ходе полученного результата работы формируется вывод, что разработка программного продукта соответствует требованиям заказчика.

## ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе по теме «Разработка информационной системы для АТП г. Лесосибирска» была спроектирована, запрограммирована и протестирована программа информационная система АТП г. Лесосибирска. В ходе проведения анализа предметной области, на основании которого был сделан вывод, что лучшим способом хранения данных в данной работе является – база данных, затем был разработан алгоритм действия программы. Выбрана среда визуального программирования Delphi 7.0 и система управления базами данных Microsoft Office Access.

После разработан пользовательский интерфейс, позволивший пользователям эффективно работать с данной программой.

Успешное внедрение информационной системы на предприятии обеспечивает принципиально новые возможности во взаимодействии сотрудников с заместителем директора посредством локальной сети. Для сотрудника нет необходимости заполнения обычного бумажного журнала, а для заместителя директора нет необходимости обходить офис, для сбора информации. Теперь программа выполняет оперативный сбор и вывод информации. Так же информационная система предприятия позволяет выводить запрос и отчёты за требуемый период, что ускоряет и упрощает работу для предприятия. Всё это обеспечивает улучшение логистики работы в данной сфере бизнеса.

Данная программа снабжена всей необходимой для установки работы документацией.

Таким образом, предъявляемые требования заказчиком к программному продукту полностью выполнены и соответствуют техническому заданию.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Алексахин, С. В. Автоматизированные системы обработки информации и управления в экономике: Учебник / С. В. Алексахин, А. Б. Николаев, И. А. Кузнецов, В. Ю. Строганов.; Под ред. А. Б. Николаева / Москва : Издательский центр «Академия», 2013. – 224 с.
2. Алешин, Л. И. Информационные технологии: Учебное пособие / Л. И. Алешин. – Москва : Маркет ДС, 2011. – 384 с.
3. Алиев, В. С. Информационные технологии и системы финансового менеджмента: Учебное пособие / В. С. Оглы Алиев. – Москва : Форум, ИНФРА-М, 2011. – 320 с.
4. Андерсен, В. Базы данных Microsoft Access. Проблемы и решения: Прост, пособ. / В. Андерсен; Пер. с англ. – Москва : Издательство ЭКОМ, 2001. – 384 с.
5. Архангельский, А. Я. Delphi 2006. Язык Delphi, классы, функции Win32 и .NET. Справочное пособие / А. Я. Архангельский. – Москва : Бинوم-Пресс, 2011. – 661 с.
6. Васильев, А. VBA в Office 2000: Учеб. курс / А. Васильев, А. Андреев. – Санкт-Петербург : Питер, 2001. – 432 с.
7. Ветитнев, А. М. Информационные технологии в социально-культурном сервисе и туризме. Оргтехника: Учебное пособие / А. М. Ветитнев. – Москва .: Форум, 2010. – 400 с.
8. Вдовин, В. М. Информационные технологии в налогообложении: Практикум / В. М. Вдовин, Л. Е. Суркова. – Москва : Дашков и К, 2012. – 248 с.
9. Вдовин, В. М. Информационные технологии в финансово-банковской сфере: Практикум / В. М. Вдовин. – Москва : Дашков и К, 2012. – 248 с.

10. Виллариал, Б. Программирование Access 2002 в примерах: Пер. с англ / Б. Виллариал. – Москва : КУДИЦ-ОБРАЗ, 2003. – 496 с.
11. Гаврилов, М. В. Информатика и информационные технологии: Учебник для бакалавров / М. В. Гаврилов, В. А. Климов. – Москва : Юрайт, 2013. – 378 с.
12. Гарнаев, А. Ю. Самоучитель VBA / А. Ю. Гарнаев. – Санкт-Петербург : БХВ – Петербург, 2001. – 512 с.
13. Гвоздева, В. А. Информатика, автоматизированные информационные технологии и системы: Учебник / В. А. Гвоздева. – Москва : ИД ФОРУМ, НИЦ ИНФРА-М, 2013. – 544 с.
14. Голицына, О. Л. Информационные технологии: Учебник / О. Л. Голицына, Н. В. Максимов, Т. Л. Партыка, И. И. Попов. – Москва : Форум, ИНФРА-М, 2013. 608 с.
15. Голубенко, Н. Б. Информационные технологии в библиотечном деле / Н. Б. Голубенко. – Ростов-на-Дону : Феникс, 2012. – 282 с.
16. Горев, А., Ахаян Р., Эффективная работа с СУБД / А. Горев, Р. Ахаян, С. Макашарипов. – Санкт-Петербург : Питер, 1997. – 704 с.
17. Гофман, В. Э. Работа с базами данных в Delphi / В. Э. Гофман, А. Д. Хомоненко. – Санкт-петербург : БХВ-Петербург, 2006. – 656 с.
18. Грэм, М. Программирование для Microsoft SQL SERVER 2000 с использованием XML / М. Грэм; Пер. с англ. – Москва : НИЦ ИНФРА-М, 2013. – 238 с.
19. Гуськова, Л. Б. О построении автоматизированного рабочего места менеджера / Л. Б. Гуськова // Успехи современного естествознания. – 2012. – №6. – С. 106.
20. Диго, С. М. Проектирование и использование баз данных / С. М. Диго. – Москва : Финансы и статистика, 1995. – 487 с.
21. Есенин, С. DirectX и Delphi. Разработка графических и мультимедийных приложений / С. Есенин. – Санкт-Петербург : БХВ-Петербург, 2006. – 512 с.

22. Жданова, М. М. Вопросы формирования профессионально важных качеств инженера / М. М. Жданова // Вестник Таджикского технического университета. – 2011. – Т. 4. – № 4. – С. 122-124.

23. Желонкин, А. Основы программирования в интегрированной среде DELPHI / А. Желонкин. – Санкт-Петербург : Бином. Лаборатория знаний, 2004. – 240 с.

24. Киммел, П. Освой самостоятельно программирование для Microsoft Access 2002 за 24 часа / П. Киммел ; Пер. с англ. – Москва : Издательский дом «Вильямс», 2003. – 480 с.

25. Когловский, М. Р. Технология баз данных на персональных ЭВМ / М. Р. Когловский. – Москва Финансы и статистика, 1992. – 315 с.

26. Козырев, А. А. Самоучитель работы на персональном компьютере. Учебное пособие / А. А. Козырев; Изд. 2-е, перераб. и доп. – Санкт-Петербург : Изд-во Михайлова В. А., 2006. – 304 с.

27. Культин, Н. Основы программирования в Turbo Delphi / Н. Культин. – Санкт-Петербург : БХВ-Петербург, 2012. – 38 с.

28. Львович, И. Я. Разработка системы поддержки принятия, управленческих решений деятельности предприятия / И. Я. Львович // Вестник Воронежского государственного технического университета. – 2007. – Т. 3. – № 8. – С. 191-192.

29. Львович, Я. Е. Принятие проектных решений на основе формирования экспертно-виртуальной среды САПР / Я. Е. Львович, Д. С. Яковлев // Вестник Воронежского института высоких технологий. – 2013. – С. 128-130.

30. Магда, Ю. Разработка приложений Microsoft Office 2007 в Delphi / Ю. Магда. – Санкт-Петербург : БХВ-Петербург, 2009. – 160 с.

31. Преображенский, А. П. Некоторые проблемы при управлении предприятием / А. П. Преображенский // Современные тенденции в экономике и управлении: новый взгляд. – 2010. – С. 168-169.

32. Рыбин, Н. Н. Организационно- производственные структуры и управления технической службой предприятий автотранспортного комплекса / Н. Н. Рыбин. – Курган : Издательство Курганского государственного университета, 2013. – 180с.

33. Фаронов, В. Delphi 2005. Язык, среда, разработка приложений / В. Фаронов. – Санкт-Петербург : Питер, 2007. – 560 с.

34. Хомоненко, А. Delphi 7 / А. Хомоненко, В. Гофман, Е. Мещеряков. – Санкт-Петербург : БХВ-Петербург, 2010. – 1120 с.

35. Хомоненко, А. – Москва : Книга по Требованию, 2006. – 454 с.

36. Шпак, Ю. А. Разработка приложений в Delphi 2005/2006 / Ю. А. Шпак. – Москва : МК-Пресс, 2006. – 544 с.

37. <http://izron.ru/articles/aktualnye-voprosy-i-perspektivy-razvitiya-matematicheskikh-i-estestvennykh-nauk-sbornik-nauchnykh-tr/sektsiya-87-informatsionnye-tehnologii-v-sovremennom-mire/bazy-znaniy-v-iskusstvennom-intellekte>

38. [http://studbooks.net/1860206/meditsina/naznachenie\\_osobennosti\\_eksper\\_tnyh\\_sistem](http://studbooks.net/1860206/meditsina/naznachenie_osobennosti_eksper_tnyh_sistem)

39. <https://studfiles.net/preview/1810935/>

40. <https://studfiles.net/preview/2928690/page:2/>

41. <https://studfiles.net/preview/2012615/>

42. <https://studfiles.net/preview/5640506/page:5/>

43. <https://studfiles.net/preview/5714063/page:2/>

44. <https://studfiles.net/preview/5879150/>

45. <https://studfiles.net/preview/6810637/page:3/>

46. [https://studopedia.ru/19\\_98595\\_novie-elementi-mezhdunarodnoy-pravovoy-sistemi.html](https://studopedia.ru/19_98595_novie-elementi-mezhdunarodnoy-pravovoy-sistemi.html)

47. [https://studopedia.ru/4\\_8114\\_ponyatie-bazi-i-banka-dannih.html](https://studopedia.ru/4_8114_ponyatie-bazi-i-banka-dannih.html)

48. [https://studwood.ru/1788626/informatika/modelirovanie\\_sistemy](https://studwood.ru/1788626/informatika/modelirovanie_sistemy)

## ПРИЛОЖЕНИЕ А

### Коды программ

#### Код приложения «отдел кадров»

##### **F1 (главная)**

```
procedure TForm1.Button2Click(Sender: TObject); // просмотр и редактирование записи
begin
  Form2.ShowModal;
end;

procedure TForm1.Button1Click(Sender: TObject); // создание новой записи
begin
  Form1.ADOTable1.Append;
  Form2.ShowModal;
end;

procedure TForm1.Button3Click(Sender: TObject); // поиск по фамилии
begin
  if Form1.Edit1.Text="" then Showmessage('ВВЕДИТЕ ФАМИЛИЮ') ELSE begin
    Form1.ADOTable1.First;
    while not Form1.ADOTable1.Eof do begin
      if Form1.ADOTable1['фамилия']=Form1.Edit1.Text then break else Form1.ADOTable1.Next;
    end; end; end;

procedure TForm1.ClientSocket1Read(Sender: TObject; Socket: TCustomWinSocket); // прием
уведомлений от сервера
begin
  Showmessage(Socket.ReceiveText);
end;
```

##### **F2 (анкета сотрудника)**

```
procedure TForm2.Button1Click(Sender: TObject); // сохранение в базе нового сотрудника
begin
  if Form1.ADOTable1.Modified then begin

    Form1.ADOTable1.Post;
    //отправка уведомления о найме заместителю директора
    Form1.ClientSocket1.Socket.SendText('найм сотрудника '+Form1.ADOTable1['фамилия']+
'+Form1.ADOTable1['имя']+ '+Form1.ADOTable1['отчество']+ на должность
'+Form1.ADOTable1['должность']);

    end;
  Close;
end;
```

##### **F3 (пароль)**

```
//пароль
procedure TForm3.Button1Click(Sender: TObject);
```

```
begin
if form3.Edit1.text='444' then begin form1.ShowModal; form3.Close; end else
showmessage('не правильный пароль');
end;
```

## Код приложения «механик»

### **F1 (главная)**

```
procedure TForm1.FormCreate(Sender: TObject);
begin
//настройка видимости по запуску
form1.DBGrid1.Visible:=false;
form1.Label2.Visible:=false;
form1.DBGrid2.Visible:=false;
form1.DBGrid3.Visible:=false;
form1.DBGrid4.Visible:=false;
form1.Label1.Visible:=false;
form1.DBNavigator1.Visible:=false;
form1.DBNavigator2.Visible:=false;
form1.DBNavigator3.Visible:=false;
form1.DBNavigator4.Visible:=false;
form1.DBNavigator5.Visible:=false;
form1.Edit1.Visible:=false;
form1.Button1.Visible:=false;
form1.Button2.Visible:=false;
form1.Button3.Visible:=false;
form1.Button4.Visible:=false;
form1.Button5.Visible:=false;
form1.DBGrid5.Visible:=false;
form1.DateTimePicker1.Visible:=false;
form1.Button7.Visible:=false;
form1.Button6.Visible:=false;
form1.Button8.Visible:=false;

// очистка таблицы запроса
Form1.ADOTable4.First;
while not Form1.ADOTable4.Eof do
begin
Form1.ADOTable4.Delete;
if Form1.ADOTable4.Modified then Form1.ADOTable4.Post;
Form1.ADOTable4.Next;
end;
end;

procedure TForm1.Button2Click(Sender: TObject); // активация формы 2 приход
begin
Form1.ADOTable2.Append;
Form2.ShowModal;
end;

procedure TForm1.Button3Click(Sender: TObject); // активация формы 3 расход
begin
```

```

Form1.ADOTable3.Append;
Form3.ShowModal;
end;

procedure TForm1.Button4Click(Sender: TObject); //просмотр прихода
begin
  Form2.ShowModal;
end;

procedure TForm1.Button5Click(Sender: TObject); // просмотр расхода
begin
  Form3.ShowModal;
end;

//поиск по названию, формирование таблицы запрос
procedure TForm1.Button1Click(Sender: TObject);
begin
  Form1.ADOTable4.First;
if Form1.Edit1.text="" then Showmessage('ВВЕДИТЕ название') ELSE
begin
  //очистить таблицу от предыдущего запроса
  while not Form1.ADOTable4.Eof do
  begin
    Form1.ADOTable4.Delete;
    if Form1.ADOTable4.Modified then Form1.ADOTable4.Post;
    Form1.ADOTable4.Next;
  end;
  //формирование таблицы по запросу
  Form1.ADOTable1.First;
  while not Form1.ADOTable1.Eof do
  begin
    if Form1.ADOTable1['название']=Form1.edit1.Text then
      begin
        Form1.ADOTable4.insert;
        Form1.ADOTable4.FieldName('Код').AsInteger:=Form1.ADOTable1.FieldName('Код').AsInteger;
        Form1.ADOTable4.FieldName('название').AsString:=Form1.ADOTable1.FieldName('название').AsString;
        Form1.ADOTable4.FieldName('количество').AsInteger:=Form1.ADOTable1.FieldName('количество').AsInteger;
        Form1.ADOTable4.FieldName('доп инфо').AsString:=Form1.ADOTable1.FieldName('доп инфо').AsString;
        if Form1.ADOTable4.Modified then Form1.ADOTable4.Post;
        Form1.ADOTable1.Next;
      end
    else Form1.ADOTable1.Next;
  end;
  Form1.Edit1.Text:="";
end;

end;

```

```

procedure TForm1.N1Click(Sender: TObject);
begin
//фильтр таблица выпуск на линию по дате при запуске
DateTimePicker1.Date:=Date();
Form1.ADOtable5.Filtered:=false;
Form1.ADOtable5.Filter:='дата='+DateToStr(DateTimePicker1.Date);
Form1.ADOtable5.Filtered:=true;

//настройка видимости по меню 1
form1.DBGrid1.Visible:=false;
form1.DBGrid2.Visible:=false;
form1.DBGrid3.Visible:=false;
form1.DBGrid4.Visible:=false;
form1.DBGrid5.Visible:=true;
form1.Label1.Visible:=false;
form1.Label2.Visible:=false;
form1.DBNavigator1.Visible:=false;
form1.DBNavigator2.Visible:=false;
form1.DBNavigator3.Visible:=false;
form1.DBNavigator4.Visible:=false;
form1.DBNavigator5.Visible:=true;
form1.Edit1.Visible:=false;
form1.Button1.Visible:=false;
form1.Button2.Visible:=false;
form1.Button3.Visible:=false;
form1.Button4.Visible:=false;
form1.Button5.Visible:=false;
form1.Button6.Visible:=true;
form1.Button7.Visible:=true;
form1.Button8.Visible:=true;
form1.DateTimePicker1.Visible:=true;
end;

```

```

procedure TForm1.N2Click(Sender: TObject);
begin
//настройка видимости по меню 2
form1.DBGrid1.Visible:=true;
form1.Label2.Visible:=true;
form1.DBGrid2.Visible:=true;
form1.DBGrid3.Visible:=true;
form1.DBGrid4.Visible:=true;
form1.Label1.Visible:=true;
form1.DBNavigator1.Visible:=true;
form1.DBNavigator2.Visible:=true;
form1.DBNavigator3.Visible:=true;
form1.DBNavigator4.Visible:=true;
form1.Edit1.Visible:=true;
form1.Button1.Visible:=true;
form1.Button2.Visible:=true;
form1.Button3.Visible:=true;
form1.Button4.Visible:=true;
form1.Button5.Visible:=true;

```

```

form1.DBGrid5.Visible:=false;
form1.DateTimePicker1.Visible:=false;
form1.Button7.Visible:=false;
form1.DBNavigator5.Visible:=false;
form1.Button8.Visible:=false;
form1.Button6.Visible:=false;
end;

```

```

procedure TForm1.Button6Click(Sender: TObject); //предрейсовый осмотр
begin
Form1.ADOtable5.Append;
Form4.ShowModal;
end;

```

```

procedure TForm1.Button7Click(Sender: TObject); // «просмотреть за», фильтр по дате
begin
Form1.ADOtable5.Filtered:=false;
Form1.ADOtable5.Filter:='дата='+DateToStr(DateTimePicker1.Date);
Form1.ADOtable5.Filtered:=true;
end;

```

```

procedure TForm1.Button8Click(Sender: TObject); //просмотреть предрейсовый осмотр
begin
Form4.ShowModal;
end;

```

```

procedure TForm1.ClientSocket1Read(Sender: TObject; Socket: TCustomWinSocket); //прием
уведомлений от сервера
begin
Showmessage(socket.ReceiveText);
end;

```

## F2 (приход)

```

procedure TForm2.FormCreate(Sender: TObject);
begin
form2.DateTimePicker1.Date:=Date(); //получить дату
end;

```

```

procedure TForm2.Button1Click(Sender: TObject); var x,y:integer;
begin
// формирование таблицы по приходу
if Form1.ADOtable2.Modified then begin
Form1.ADOtable2.FieldName('дата
преобращения').Value:=trunc(form2.DateTimePicker1.date);
Form1.ADOtable2.Post; end;
//формирование таблицы склад
// если код прихода совпадает с имеющимся на складе + к количеству
Form1.ADOtable1.First;
while not Form1.ADOtable1.Eof do begin
if form1.ADOtable1['Код']= form1.ADOtable2['Код'] then
begin
x:= strtoint(Form1.ADOtable1['количество']);

```

```

y:= strtoint(Form1.ADOTable2['количество']);
x:=x+y;
form1.adotable1.edit;
form1.ADOTable1['количество']:=inttostr(x);
if Form1.ADOTable1.Modified then Form1.ADOTable1.Post; break;
end
else Form1.ADOTable1.Next; end;
//если достигли конца ТСклад и условие совпадения кода не выполнилось, то создание
новой записи с копированием данных из ТПриход
if (Form1.ADOTable1.eof) and
(Form1.ADOTable1.FieldByName('Код').AsInteger<>Form1.ADOTable2.FieldByName('Код').
AsInteger) then
begin
Form1.ADOTable1.insert;
Form1.ADOTable1.FieldByName('Код').AsInteger:=Form1.ADOTable2.FieldByName('Код').A
sInteger;
Form1.ADOTable1.FieldByName('название').AsString:=Form1.ADOTable2.FieldByName('наз
вание').AsString;
Form1.ADOTable1.FieldByName('количество').AsInteger:=Form1.ADOTable2.FieldByName
('количество').AsInteger;
Form1.ADOTable1.FieldByName('доп
инфо').AsString:=Form1.ADOTable2.FieldByName('доп инфо').AsString;
end;
Close;
end;

```

### **Ф3 (расход)**

```

procedure TForm3.FormCreate(Sender: TObject);
begin
form3.DateTimePicker1.Date:=Date(); //получить дату
end;

procedure TForm3.Button1Click(Sender: TObject); var x,y,z:integer;
begin
// формирование таблицы по расходу
if Form1.ADOTable3.Modified then
begin
Form1.ADOTable1.First;
// если код прихода совпадает с имеющимся на складе
while not Form1.ADOTable1.Eof do
begin
if form1.ADOTable1['Код']= form1.ADOTable3['Код'] then
begin
x:= strtoint(Form1.ADOTable1['количество']);
y:= strtoint(Form1.ADOTable3['количество']);
//если кол-во таб3 не превышает кол-во таб1. Форируем таб3, изменяем кол-во в таб 1
if x>y then
begin
Form1.ADOTable3.FieldByName('дата
выдачи').Value:=trunc(form3.DateTimePicker1.date);
Form1.ADOTable3.Post;
z:=x-y;

```

```

        form1.adotable1.edit;
        form1.ADOTable1['количество']:=inttostr(z);
        if Form1.ADOTable1.Modified then Form1.ADOTable1.Post; break;
    end;
//если кол-во Т3=Т1, формируем Т3, удаляем строку из Т1
    if x=y then
        begin
            Form1.ADOTable3.FieldByName('дата
выдачи').Value:=trunc(form3.DateTimePicker1.date);
            Form1.ADOTable3.Post;
            Form1.ADOTable1.Delete;
            if Form1.ADOTable1.Modified then Form1.ADOTable1.Post; break;
        end;
//если кол-во E
        if x<y then
            begin
                Form1.ADOTable3.Delete;
                Showmessage('товар с данным кодом, в указанном количестве на
складе отсутствует'); break;
            end;
        end
        else Form1.ADOTable1.Next;
    end;
end;
Close;
end;

```

#### **F4 (осмотр)**

```

procedure TForm4.Button1Click(Sender: TObject);
begin
    form1.DateTimePicker1.Date:=Date();
    if (DBCheckBox1.Checked=true) and (DBCheckBox2.Checked=true) and
(DBCheckBox3.Checked=true) and (DBCheckBox4.Checked=true) and
    (DBCheckBox5.Checked=true) and (DBCheckBox6.Checked=true) and
(DBCheckBox7.Checked=true) and (DBCheckBox8.Checked=true)
    then begin
        Form1.ADOTable5.FieldByName('прохождение контроля').AsString:='допущен';
        Form1.ADOTable5.FieldByName('дата').Value:=trunc(form1.DateTimePicker1.Date);
        Form1.ADOTable5.Post;
    end
    else Form1.ADOTable5.delete;
close;
end;

```

#### **F5 (пароль)**

```

//пароль
procedure TForm5.Button1Click(Sender: TObject);
begin
    if form5.Edit1.text='333' then begin form1.ShowModal; form5.Close; end else
showmessage('не правильный пароль');
end;

```

## Код приложения «МДН»

### Ф1 (главная)

```
procedure TForm1.FormCreate(Sender: TObject); //фильтр по дате при запуске формы
begin
DateTimePicker1.Date:=Date();
DateTimePicker1.Time:=Time();

Form1.ADOtable1.Filtered:=false;           //1
Form1.ADOtable1.Filter:='date='+DateToStr(DateTimePicker1.Date);
Form1.ADOtable1.Filtered:=true;

Form1.ADOtable2.Filtered:=false;           //2
Form1.ADOtable2.Filter:='date='+DateToStr(DateTimePicker1.Date);
Form1.ADOtable2.Filtered:=true;

Form1.ADOtable3.Filtered:=false;           //3
Form1.ADOtable3.Filter:='date='+DateToStr(DateTimePicker1.Date);
Form1.ADOtable3.Filtered:=true;
end;

procedure TForm1.N2Click(Sender: TObject); //меню настройка видимости объектов 1
begin
Form1.DBGrid1.Visible:=true;
Form1.DBGrid2.Visible:=false;
Form1.DBGrid3.Visible:=false;
button1.Visible:=true;
DBNavigator1.visible:=false;
label1.Visible:=false;
edit1.Visible:=false;
button3.Visible:=false;
button4.Visible:=false;
button2.Visible:=true;
button5.Visible:=False;
end;

procedure TForm1.N3Click(Sender: TObject); // меню настройка видимости объектов 2
begin
Form1.DBGrid1.Visible:=false;
Form1.DBGrid2.Visible:=true;
Form1.DBGrid3.Visible:=false;
button1.Visible:=false;
DBNavigator1.visible:=false;
label1.Visible:=true;
edit1.Visible:=true;
button3.Visible:=true;
button4.Visible:=true;
button2.Visible:=true;
button5.Visible:=False;
end;
```

```

procedure TForm1.N4Click(Sender: TObject); // меню настройка видимости объектов 3
begin
Form1.DBGrid1.Visible:=false;
Form1.DBGrid2.Visible:=false;
Form1.DBGrid3.Visible:=true;
button1.Visible:=false;
DBNavigator1.visible:=true;
label1.Visible:=false;
edit1.Visible:=false;
button3.Visible:=false;
button4.Visible:=false;
button2.Visible:=true;
button5.Visible:=true;
end;

```

```

Procedure TForm1.Button1Click(Sender: TObject); // начало осмотра
begin
Form1.ADOtable1.Append;
Form1.ADOtable1.FieldByName('date').Value:=trunc(DateTimePicker1.Date);
DateTimePicker1.Time:=Time();
Form1.ADOtable1.FieldByName('time').Value:=DateTimePicker1.Time;
Form2.ShowModal;
end ;

```

```

procedure TForm1.Button2Click(Sender: TObject); // фильтр по дате при изменении даты
begin
Form1.ADOtable1.Filtered:=false; //1
Form1.ADOtable1.Filter:='date='+DateToStr(DateTimePicker1.Date);
Form1.ADOtable1.Filtered:=true;

```

```

Form1.ADOtable2.Filtered:=false; //2
Form1.ADOtable2.Filter:='date='+DateToStr(DateTimePicker1.Date);
Form1.ADOtable2.Filtered:=true;

```

```

Form1.ADOtable3.Filtered:=false; //3
Form1.ADOtable3.Filter:='date='+DateToStr(DateTimePicker1.Date);
Form1.ADOtable3.Filtered:=true;
end;

```

```

procedure TForm1.Button4Click(Sender: TObject); //поиск по фамилии т2
begin
if Form1.Edit1.text="" then Showmessage('ВВЕДИТЕ ФАМИЛИЮ') ELSE begin
Form1.ADOtable2.First;
while not Form1.ADOtable2.Eof do begin
if Form1.ADOtable2['фамлия водителя']=Form1.edit1.Text then break else
Form1.ADOtable2.Next; end;end;
end;

```

```

procedure TForm1.Button3Click(Sender: TObject); //заполнение т2
begin

```

```

Form3.ShowModal;
end;
procedure TForm1.ClientSocket1Read(Sender: TObject; Socket: TCustomWinSocket); //прием
уведомлений от сервера
begin
  Showmessage(socket.ReceiveText);
end;

//уведомление сервера о выпуске на линию
procedure TForm1.Button5Click(Sender: TObject); var s:integer;
begin
s:=0;
while not Form1.ADOTable1.Eof do begin
  if Form1.ADOTable1['date']= DateToStr(DateTimePicker1.Date) then s:=s+1;
Form1.ADOTable1.Next; end;
  label2.Caption:=IntToStr(s);
  FORM1.ClientSocket1.Socket.SendText(DateToStr(DateTimePicker1.Date)+' на линию
выпущено '+IntToStr(s)+' автобусов');
end;
end.

```

## **F2 (анкета предрейсового мед осмотра)**

```

procedure TForm2.Button1Click(Sender: TObject);
begin
//условия мед допуска
if (DBCheckBox1.Checked=true then) and (form1.adotable1.
FieldByName('температура').AsString='36,6') and (form1.adotable1.
FieldByName('пульс').AsInteger<80) and (form1.adotable1. FieldByName('давление
верхнее').AsInteger<131) and (form1.adotable1. FieldByName('давление
верхнее').AsIntege>99) and (DBCheckBox2.Checked<>true then) then

begin
Form1.ADOTable1.Post; //медосмотр пройден, запись данных в таблицу
//копирование данных в т3
Form1.ADOTable3.insert;
Form1.ADOTable3.FieldByName('№').AsInteger:=Form1.ADOTable1.FieldByName('№').AsInt
eger;
Form1.ADOTable3.FieldByName('фамлия
водителя').AsString:=Form1.ADOTable1.FieldByName('фамлия водителя').AsString;
  Form1.ADOTable3.FieldByName('date').Value:=trunc(form1.DateTimePicker1.Date);
  Form1.ADOTable3.FieldByName('time').Value:=form1.DateTimePicker1.Time;
//копирование данных в т2
Form1.ADOTable2.insert;
Form1.ADOTable2.FieldByName('№ путевого/билетно
учетного').AsInteger:=Form1.ADOTable1.FieldByName('№').AsInteger;
Form1.ADOTable2.FieldByName('фамлия
водителя').AsString:=Form1.ADOTable1.FieldByName('фамлия водителя').AsString;
Form1.ADOTable2.FieldByName('date').Value:=trunc(form1.DateTimePicker1.Date);
end
//иначе данные удалить
else Form1.ADOTable1.Delete;
Close;

```

end;

### **F3 (анкета послерейсового осмотра)**

```
procedure TForm3.Button1Click(Sender: TObject);
begin
form1.DateTimePicker1.Time:=Time();//запись времени осмотра
Form1.ADOTable2.FieldByName('time').Value:=form1.DateTimePicker1.Time;
Form1.ADOTable2.Post;//запись данных осмотра
Close;
end;
```

### **F4 (пароль)**

```
//пароль
procedure TForm4.Button1Click(Sender: TObject);
begin
if form4.Edit1.text='222' then begin form1.ShowModal; form4.Close; end else
showmessage('не правильный пароль');
end;
```

## **Код приложения «касса»**

### **F1 (главная)**

```
procedure TForm1.FormCreate(Sender: TObject);
begin
DateTimePicker1.Date:=Date();
// фильтр по дате при запуске формы
Form1.ADOTable1.Filtered:=false;
Form1.ADOTable1.Filter:='date='+DateToStr(DateTimePicker1.Date);
Form1.ADOTable1.Filtered:=true;

Form1.ADOTable2.Filtered:=false;
Form1.ADOTable2.Filter:='date='+DateToStr(DateTimePicker1.Date);
Form1.ADOTable2.Filtered:=true;

//подсчет итога при запуске формы
s:=0;
Form1.ADOTable1.First;
while not Form1.ADOTable1.Eof do begin
S:=S+strtoint(Form1.ADOTable1['сумма']); Form1.ADOTable1.Next; end;
label1.Caption:=inttostr(s);
d:=0;
Form1.ADOTable2.First;
while not Form1.ADOTable2.Eof do begin
d:=d+strtoint(Form1.ADOTable2['сумма']); Form1.ADOTable2.Next; end;
label2.Caption:=inttostr(d);
end;

procedure TForm1.Button1Click(Sender: TObject); //сохранить/пересчитать 1
begin
if Form1.ADOTable1.Modified then begin
Form1.ADOTable1['date']:=trunc(DateTimePicker1.Date); Form1.ADOTable1.Post; end;
s:=0;
```

```

Form1.ADOtable1.First;
  while not Form1.ADOtable1.Eof do begin
    S:=S+strtoint(Form1.ADOtable1['сумма']); Form1.ADOtable1.Next; end;
  label1.Caption:=inttostr(s);
Form1.ADOtable1.edit;
end;

procedure TForm1.Button2Click(Sender: TObject); //фильтр по дате просмотреть за...
begin
  Form1.ADOtable1.Filtered:=false; //1
  Form1.ADOtable1.Filter:='date='+DateToStr(DateTimePicker1.Date);
  Form1.ADOtable1.Filtered:=true;
  Form1.ADOtable2.Filtered:=false; //2
  Form1.ADOtable2.Filter:='date='+DateToStr(DateTimePicker1.Date);
  Form1.ADOtable2.Filtered:=true;
  //пересчет при изменении даты
  s:=0; //1
  Form1.ADOtable1.First;
  while not Form1.ADOtable1.Eof do begin
    S:=S+strtoint(Form1.ADOtable1['сумма']); Form1.ADOtable1.Next; end;
    label1.Caption:=inttostr(s);
    d:=0; //2
  Form1.ADOtable2.First;
  while not Form1.ADOtable2.Eof do begin
    d:=d+strtoint(Form1.ADOtable2['сумма']); Form1.ADOtable2.Next; end;
    label2.Caption:=inttostr(d);
  end;

  procedure TForm1.Button4Click(Sender: TObject); //сохранить/пересчитать 2
  begin
    if Form1.ADOtable2.Modified then begin
      Form1.ADOtable2['date']:=trunc(DateTimePicker1.Date); Form1.ADOtable2.Post; end;
      d:=0;
      Form1.ADOtable2.First;
      while not Form1.ADOtable2.Eof do begin
        d:=d+strtoint(Form1.ADOtable2['сумма']); Form1.ADOtable2.Next; end;

        label2.Caption:=inttostr(d);
      Form1.ADOtable2.edit;
    end;

  procedure TForm1.ClientSocket1Read(Sender: TObject; Socket: TCustomWinSocket); //прием
  уведомлений от сервера
  begin
    Showmessage(socket.ReceiveText);
  end;

end.

F2 (пароль)

//пароль
procedure TForm2.Button1Click(Sender: TObject);
begin

```

```
if form2.Edit1.text='222' then begin form1.ShowModal; form2.Close; end else
showmessage('не правильный пароль');
end;
```

## Код приложения «заместитель директора»

### **F1 (главная)**

```
procedure TForm1.Button2Click(Sender: TObject); //отдел кадров
begin
form2.ShowModal;
end;
```

```
procedure TForm1.Button4Click(Sender: TObject); //касса
begin
form3.ShowModal;
end;
```

```
procedure TForm1.Button5Click(Sender: TObject); //МДН
begin
form4.ShowModal;
end;
```

```
procedure TForm1.Button3Click(Sender: TObject); //механик
begin
form5.ShowModal;
end;
```

```
//отправка уведомления всем клиентам
procedure TForm1.Button1Click(Sender: TObject); var i:integer;
begin
For i:=0 to SeverSocket.ActiveConnections-1 do
serversocket1.Socket.Connections[i].SendText(edit1.Text);
end;
```

### **F2 (отдел кадров)**

```
procedure TForm2.Button1Click(Sender: TObject); //поиск по фамилии
begin
if Form2.Edit1.text="" then Showmessage('ВВЕДИТЕ ФАМИЛИЮ') ELSE begin
Form2.ADOTable1.First;
while not Form2.ADOTable1.Eof do begin
if Form2.ADOTable1['фамилия']=Form2.edit1.Text then break else Form2.ADOTable1.Next;
end; end;
end;
```

```
procedure TForm2.ServerSocket1ClientRead(Sender: TObject; // прием уведомления
Socket: TCustomWinSocket);
begin
SHOWMESSAGE(SOCKET.ReceiveText); end;
procedure TForm2.Button2Click(Sender: TObject); //отправка уведомления
begin
serversocket1.Socket.Connections[0].SendText(edit2.Text);
end;
```

### F3 (касса)

```
procedure TForm3.FormCreate(Sender: TObject); var s,d:integer;
begin
  DateTimePicker1.Date:=Date();
    // фильтр по дате при запуске формы
  Form3.ADOTable1.Filtered:=false;
  Form3.ADOTable1.Filter:='date='+DateToStr(DateTimePicker1.Date);
  Form3.ADOTable1.Filtered:=true;

  Form3.ADOTable2.Filtered:=false;
  Form3.ADOTable2.Filter:='date='+DateToStr(DateTimePicker1.Date);
  Form3.ADOTable2.Filtered:=true;
    //подсчет итога при запуске формы
  s:=0;
  Form3.ADOTable1.First;
  while not Form3.ADOTable1.Eof do begin
    S:=S+strtoint(Form3.ADOTable1['сумма']); Form3.ADOTable1.Next; end;
  label3.Caption:=inttostr(s);
  d:=0;
  Form3.ADOTable2.First;
  while not Form3.ADOTable2.Eof do begin
    d:=d+strtoint(Form3.ADOTable2['сумма']); Form3.ADOTable2.Next; end;
  label4.Caption:=inttostr(d);
end;

procedure TForm3.Button1Click(Sender: TObject); var s,d:integer; //просмотреть за...
begin
  Form3.ADOtable1.Filtered:=false; //1
  Form3.ADOtable1.Filter:='date='+DateToStr(DateTimePicker1.Date);
  Form3.ADOtable1.Filtered:=true;
  Form3.ADOtable2.Filtered:=false; //2
  Form3.ADOtable2.Filter:='date='+DateToStr(DateTimePicker1.Date);
  Form3.ADOtable2.Filtered:=true;
    //пересчет при изменении даты
  s:=0; //1
  Form3.ADOTable1.First;
  while not Form3.ADOTable1.Eof do begin
    S:=S+strtoint(Form3.ADOTable1['сумма']); Form3.ADOTable1.Next; end;
  label3.Caption:=inttostr(s);
  d:=0; //2
  Form3.ADOTable2.First;
  while not Form3.ADOTable2.Eof do begin
    d:=d+strtoint(Form3.ADOTable2['сумма']); Form3.ADOTable2.Next; end;
  label4.Caption:=inttostr(d);
end;
procedure TForm3.Button2Click(Sender: TObject);//отправка уведомления
begin
  serversocket1.Socket.Connections[1].SendText(edit1.Text);
end;
end.
```

#### **F4 (МДН)**

```
Procedure TForm4.FormCreate(Sender: TObject);//фильтр по дате при запуске
begin
DateTimePicker1.Date:=Date();
Form4.ADOtable1.Filtered:=false;
Form4.ADOtable1.Filter:='date='+DateToStr(DateTimePicker1.Date);
Form4.ADOtable1.Filtered:=true;
end;

procedure TForm4.Button1Click(Sender: TObject);//Фильтр по дате при ее изменении
begin
Form4.ADOtable1.Filtered:=false;
Form4.ADOtable1.Filter:='date='+DateToStr(DateTimePicker1.Date);
Form4.ADOtable1.Filtered:=true;
end;

//прием уведомления
procedure TForm4.ServerSocket1ClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  SHOWMESSAGE(SOCKET.ReceiveText);
end;

//отправка уведомления
procedure TForm4.Button2Click(Sender: TObject);
begin
serversocket1.Socket.Connections[2].SendText(edit1.Text);
end;
```

#### **F5 (Механик)**

```
//настройка видимости объектов при запуске
procedure TForm5.FormCreate(Sender: TObject);
begin
Form5.DBGrid1.Visible:=false;
Form5.DBGrid2.Visible:=false;
Form5.DBNavigator1.Visible:=false;
Form5.DBNavigator2.Visible:=false;
Form5.Button1.Visible:=false;
Form5.DateTimePicker1.Visible:=false;
end;

procedure TForm5.N1Click(Sender: TObject);//видимость меню 1
begin
Form5.DBGrid1.Visible:=true;
Form5.DBGrid2.Visible:=false;
Form5.DBNavigator1.Visible:=true;
Form5.DBNavigator2.Visible:=false;
Form5.Button1.Visible:=true;
Form5.DateTimePicker1.Visible:=true;

DateTimePicker1.Date:=Date();
```

```
//фильтр по дате
Form5.ADOtable1.Filtered:=false;
Form5.ADOtable1.Filter:='дата='+DateToStr(DateTimePicker1.Date);
Form5.ADOtable1.Filtered:=true;
end;
```

```
procedure TForm5.N2Click(Sender: TObject);//видимость меню 2
begin
Form5.DBGrid1.Visible:=false;
Form5.DBGrid2.Visible:=true;
Form5.DBNavigator1.Visible:=false;
Form5.DBNavigator2.Visible:=true;
Form5.Button1.Visible:=false;
Form5.DateTimePicker1.Visible:=false;
end;
```

```
procedure TForm5.Button1Click(Sender: TObject);//фильтр по дате при ее изменении
begin
Form5.ADOtable1.Filtered:=false;
Form5.ADOtable1.Filter:='дата='+DateToStr(DateTimePicker1.Date);
Form5.ADOtable1.Filtered:=true;
end;
```

```
procedure TForm5.Button2Click(Sender: TObject); //отправка уведомления
begin
serversocket1.Socket.Connections[3].SendText(edit1.Text);
end;
end.
```

### **Ф6 (пароль)**

```
//пароль
procedure TForm6.Button1Click(Sender: TObject);
begin
if form6.Edit1.text='222' then begin form1.ShowModal; form6.Close; end else
showmessage('не правильный пароль');
end;
```

## ПРИЛОЖЕНИЕ Б

### Очерк научной статьи

#### **Информационная система автотранспортного предприятия**

***Сахаров Александр Сергеевич***

*студент, ЛПИ – филиал СФУ, РФ, г.Лесосибирск*

***Рубцов Александр Владимирович***

*научный руководитель, кандидат экономических наук, ЛПИ - СФУ, РФ, г. Лесосибирск*

***Храмова Людмила Николаевна***

*научный руководитель, кандидат экономических наук, доцент, РФ, ЛПИ – СФУ, г. Лесосибирск*

Автотранспортные предприятия, имеют широкое применение и высокоорганизованную автоматизированную структуру управления в больших городах. В малых же городах, таких как Лесосибирск, подобные предприятия, хоть и являются многолетними и востребованными организациями, в целях хранения информации, пользуются исключительно бумажными носителями. Но вопрос о разработке программных приложений, поднимался, ни единожды. Даже частичная автоматизация данной сферы бизнеса, значительно бы улучшила качество работы предприятия и во многом бы упростила работу сотрудников.

С этой целью появилась потребность разработки информационной системы автотранспортного предприятия.

ИС – система по обработки информации, включающая в свой состав следующие компоненты:

- Данные
- Информация

- Знания
- Базы данных
- Базы знаний
- Программное обеспечение (пакет прикладных программ)
- Экспертные системы
- Компьютерные сети (корпоративные сети)
- Защита информации

Следовательно, при разработке проекта необходимо учесть тот факт, что придется работать с потоком информации, которая ежедневно должна быть фиксирована и сохранена в архиве в виде данных. Все данные для удобного доступа и наглядности будут храниться в виде таблицы – базы данных.

К некоторым данным будут добавлены знания (правила) при выполнении которых, программа будет принимать самостоятельные решения. Так, например, при медицинском осмотре, будут вводиться показатели человеческого состояния, а программа, пользуясь заданными параметрами, будет принимать решения о допуске или не допуске сотрудника до рабочего места. Также обстоит дело и с транспортом, механик при осмотре транспортного средства, будет фиксировать данные, а программа на основе этих данных выдавать разрешение о выводе авто на линию.

Проект, с целью обхвата предприятия и всех наиболее важных сторон его функционирования, будет состоять из нескольких приложений – прикладного программного комплекса, элементы которого будут связаны внутренней корпоративной сетью, с возможностью предоставления информации контролирующему органу.

Об аппаратной и организационной безопасности решения будут приниматься руководящими кадрами предприятия, но и в предлагаемом программном продукте, можно предусмотреть некоторые аспекты

организации защиты данных. Возможно установление паролей на каждом из рабочих мест (в каждом приложении) или сокрытие некоторых данных, от посторонних наблюдателей. Так, например, избежать выведения на общий экран (общую базу данных) отдела кадров, информацию о документах сотрудника. И сделать возможным просмотр данных паспорта лишь по требованию специалиста.

Разработка должностной модели, должна наглядно показать, сколько и для кого необходимо разработать программных приложений, и каким образом они должны быть соединены в корпоративную сеть. На Рисунке 1 представлена структура офиса предприятия в соответствии с уровнем должностных обязательств:

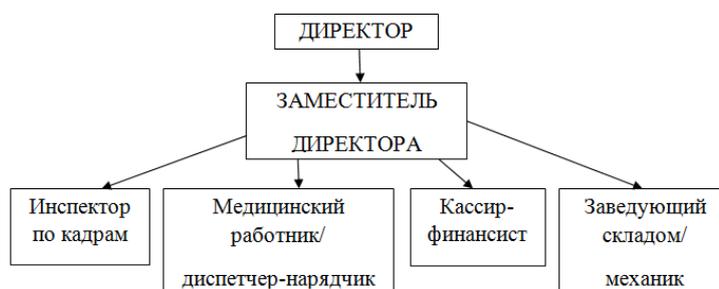


Рисунок Б1 – Должностная модель предприятия АТП

Во главе стоит директор, ему подчиняется заместитель директора, ответ перед которым держат инспектор по кадрам, медицинский работник он же выполняющий функции диспетчера-нарядчика, кассир-финансист, и механик по совместительству заведующий складом запчастей.

Для разработки информационной системы требуется создание четырех прикладных приложений формирующих отчет и отправляющих его по внутренней сети заместителю директора.

Процессная модель, разрабатывается с целью рассмотрения выполняемых внутри организацией процессов, и их ожиданий Рисунок 2.

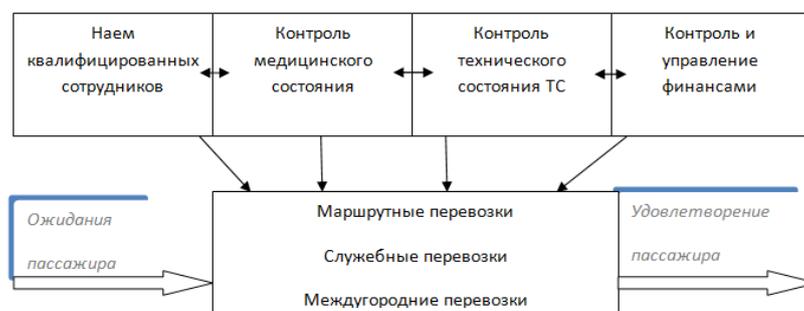


Рисунок Б2 – Процессная модель

Есть пассажир, он же заказчик с определенными требованиями и ожиданиями:

- Все автобусы должны быть на линии, простой пассажира на остановке не более двух минут
- Пассажирский транспорт в хорошем техническом состоянии, соответствует условиям максимальной безопасности. Чистый и комфортабельный
- Перевозку осуществляет опытный водитель, со стажем, вменяемый (не находящийся в алкогольном опьянении или под воздействием психотропных средств), с хорошими показателями самочувствия (т.е. прошедший предрейсовый медицинский осмотр)
- Обилечивает и следит за порядком в салоне честный и вежливый кондуктор

То какой в действительности результат получит пассажир на выходе, зависит от системности и качества проведения внутренних процессов.

- Наем квалифицированных сотрудников со стажем, с хорошими рекомендациями (ответственный инспектор по кадрам)
- Проведение ежедневного предрейсового осмотра транспортного средства. Осуществление своевременного периодического ремонта и замена всех расходников авто (ответственный механик)

– Проведение ежедневного предрейсового медицинского осмотра водителей, а также послерейсового для оценки посильности условий труда для сотрудника (ответственный медицинский работник)

– Контроль и управление финансами, (ответственный кассир-финансист)

Для того чтобы результат на выходе максимально соответствовал ожиданиям пассажира, предприятие нуждается в слаженной организованной системе, которая бы помогала персоналу в осуществлении процессов, и могла хранить большие объемы данных.

Программа для отдела кадров включает в себя базу данных о сотрудниках с краткой информацией о них, и возможность поиска работника по фамилии. Также необходимо наличие анкеты приема сотрудника на работу, доступной в дальнейшем для просмотра, с содержанием более полной информации по каждому кадру (табель, ФИО, дата рождения, количество детей, адрес проживания, образование, занимаемая должность, дата приема на работу, паспортные данные, ИНН, номер медицинского страхования, номер водительского удостоверения).

Программа для кассира будет представлять собой две таблицы, одна по приходу с колонками: сумма, источник доходов. Другая по расходу с колонками: сумма, направление расходов. А также, что самое главное возможность подсчета итогов по таблицам.

Приложение для медицинского работника/ диспетчера-нарядчика будет состоять из нескольких пунктов верхнего меню предрейсовый медицинский осмотр, присвоение наряда диспетчером-нарядчиком и послерейсовый медицинский осмотр. Осмотр медицинским работником проводится по категориям: визуальное состояние, температура, давление, пульс, тестирование на наличия алкоголя. В случае удовлетворения по всем медицинским показателям и допуска сотрудника к работе, диспетчером-нарядчиком будет присвоен ему номер путевого листа, закреплено транспортное средство, снабжение экипажем (кондуктор). В конце рабочего

дня также осуществляется медицинский осмотр и дается заключение о состоянии работника.

Механик/кладовщик нуждается в приложении, которое бы фиксировало все имеющиеся в его распоряжении запчасти, информацию по приходу и расходу, также важна организация поиска по списку, с целью определения их наличия. Помимо информации о состоянии склада, в обязанности механика входит проведение предрейсового осмотра ТС по пунктам: визуальное состояние, исправность тормозной системы, исправность рулевого управления, исправность сетевых приборов, состояние колес/шин, исправность двигателя, состояние тахографа, исправность звуковых сигналов.

Приложение заместителя директора включает в себя получение некоторых общих сведений, отчетов по работе от нижестоящих сотрудников.

Создание данного набора программ – проект по автоматизации хранения и обработки данных, значительно облегчит работу всего персонала, даст хороший толчок автотранспортному предприятию, и поможет ему выйти на новый информационный уровень.

#### **Список литературы:**

1. Михеев Е.В. Информационные технологии в профессиональной деятельности // Издательский центр «Академия», 2008-384с.
2. Рыбин Н.Н. Организационно- производственные структуры и управления технической службой предприятий автотранспортного комплекса // Издательство Курганского государственного университета, 2013-180с.

#### **Выходные данные**

Сахаров А.С., Рубцов А.В., Храмова Л.Н. Информационная система автотранспортного предприятия // Студенческий форум: электрон. научн. журн. 2018. № 9(30). URL: <https://nauchforum.ru/journal/stud/30/35469>