

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)

Политехнический институт: заочный
Кафедра «Системы автоматического управления»

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой

_____/ В.И. Ширяев

« ____ » _____ 2017 г.

Программное обеспечение системы контроля доступа персонала компании "РБТ"

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ – 09.03.01.2017.064.00 ПЗ ВКР

Руководитель работы

доц. каф. САУ, к.т.н.

_____/ Е.А. Алешин

« ____ » _____ 2017 г.

Автор работы

студент группы ПЗ-597

_____/ М.А. Герасимов

« ____ » _____ 2017 г.

Нормоконтролер

доц. каф. САУ, к.т.н.

_____/ Е.А. Алешин

« ____ » _____ 2017 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ.....	8
1.1 Описание предметной области.....	8
1.2 Обзор аналогов	13
1.3 Преимущества разрабатываемой программной части системы контроля и управления доступом	15
1.4 Постановка задачи	16
1.5 Выбор средств и технологий	18
1.6 Требования к интерфейсу приложения	27
2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА.....	29
2.1 Теоретическое обоснование.....	29
2.1.1 Работа автоматизированной проходной.....	31
2.1.2 Программное обеспечение автоматизированной проходной.....	32
2.2 Разработка базы данных.....	32
2.2.1 Логическая схема используемой базы данных	32
2.2.2 Описание таблиц базы данных Access.....	33
2.2.3 Создание файла базы данных	38
2.3 Модуль авторизации программного обеспечения.....	41
2.4 Модуль данных формы мониторинга	46
2.5 Алгоритм работы проходной	48
2.6 Главная форма	50
2.7 Модуль данных формы администрирования	52
2.8 Функционал вкладки для получения отчётов	53
3 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	57
3.1 Рабочее место оператора	57
3.2 Рабочая форма администратора	60
ЗАКЛЮЧЕНИЕ	75
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	76
ПРИЛОЖЕНИЕ А. Исходный код программы	78

									Лист
									5
Изм.	Лист	№ докум.	Подпись	Дата	09.03.01.2017.064.00 ПЗ				

ВВЕДЕНИЕ

Системы контроля и управления доступом с каждым годом получают всё более широкое распространение. Большой рост данной сферы объясняется увеличением грамотности людей в сфере безопасности, новыми разработками, а также возросшим количеством угроз в современном мире.

Необходимость в установке таких систем появляется на объектах, где постоянно требуется обеспечивать высокий уровень безопасности, контролировать попадание людей на территорию, их перемещение по ней, а также вести предотвращение хищений на предприятии.

Системы контроля и управления доступом – это важный компонент комплекса безопасности любого современного производства и установка подобных систем повышает обеспечиваемый уровень безопасности.

Установка подобных систем решает следующие задачи:

- Препятствует проникновению посторонних на территорию;
- Определяет права доступа для сотрудников в различные помещения;
- Ограничивает время входа на предприятие или в различные производственные помещения в определённые временные диапазоны;
- Предотвращает вероятность проникновения на территорию с чужим пропуском;
- Обеспечивает контроль перемещения сотрудников по предприятию и соблюдения ими трудовой дисциплины;
- Возможность оперативно заменить утерянные ключи доступа;
- Интеграцию с другими системами безопасности для обеспечения должного уровня безопасности в экстренных ситуациях.

Системы контроля и управления доступом также участвуют в решении немаловажных второстепенных задач таких как:

- Контроль рабочего времени сотрудников. Для снижения нагрузки на бухгалтерию при расчёте заработной платы;
- Управление инженерными коммуникациями;

					09.03.01.2017.064.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

- Улучшение процесса управления персоналом, заключающееся в быстром нахождении сотрудника на предприятии, а также помощь в оценке эффективности его работы;

- Оперативно реагировать на нештатные ситуации на территории организации;

- Снижение затрат на службу охраны;

- В случае возникновения чрезвычайной ситуации оперативно открывать все проходы для экстренной эвакуации людей.

Ранее перечисленные задачи, которые решает установка систем контроля и управления доступом делает актуальным разработку и внедрение программного обеспечения для проектируемой автоматизированной проходной, которая решит поставленные на данный момент задачи, такие как:

- Контроль за рабочим временем сотрудников;

- Контроль за посещением предприятия при помощи временных пропусков;

- Контроль за соответствием человека, предъявившего пропуск для прохода на предприятие.

Для выполнения поставленной задачи необходимо написать программу работы с базой данных в которую после выбора и покупки необходимого оборудования будут добавлены модули работы с оборудованием поставляемые поставщиками оборудования по требованию клиента, а также разработать базу данных, содержащую данные о правах доступа и фиксации посещений.

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		7

1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ

1.1 Описание предметной области

Система контроля и управления доступом - это совокупность программно-аппаратных технических средств безопасности и организационно-методических мероприятия, которые ставят перед собой цель ограничить и регистрировать вход-выход людей и транспортных средств на территории предприятия через определённые точки прохода: двери, ворота и контрольно-пропускные пункты. В основном СКУД решают задачу управления доступом на определённую территорию, а также ограничивают доступ на заданную территорию и проводят идентификацию лиц, имеющих определённые для них в системе права доступа. СКУД – это не просто программно-технический комплекс, а продуманная система управления передвижением персонала на территории охраняемого предприятия.

В настоящее время использование систем контроля и управления доступом, как средства контроля за человеческими ресурсами предприятия, а также обеспечения необходимого уровня безопасности становится одним из приоритетных направлений в обеспечении нормальной работы предприятия.

Средства систем контроля и управления доступом разделяют на классифицируют по следующим признакам:

– По функциональному назначению устройств контроля и управления доступом:

1) Управляемые преграждающие устройства (в составе преграждающих конструкций и исполнительных устройств систем контроля доступом):

а) По виду перекрытия проёма прохода устройства делят на:

- с частичным перекрытием (турникеты, шлагбаумы);
- с полным перекрытием (сплошные двери, ворота);
- с блокированием объекта в проеме (шлюзы, кабины-проходные).

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		8

б) По способу управления преграждающие устройства, входящие в состав СКУД, подразделяют на:

- с ручным управлением;
- с полуавтоматическим управлением,
- с автоматическим управлением.

2) Устройства ввода идентификационных признаков:

а) По виду используемых идентификационных признаков устройства могут быть:

- механические - идентификационные признаки представляют собой элементы конструкции идентификаторов;
- магнитные - идентификационные признаки представляют собой намагниченные участки поверхности или магнитные элементы идентификатора;
- оптические - идентификационные признаки представляют собой нанесенные на поверхности или внутри идентификатора метки, имеющие различные оптические характеристики в отраженном или проходящем оптическом излучении, помогающие осуществлять контроль доступа к охраняемому объекту;
- электронные - идентификационные признаки представляют собой электронный код, записанный в микросхеме идентификатора;
- акустические - идентификационные признаки представляют собой кодированный акустический сигнал;
- биометрические - идентификационные признаки для биометрических устройств представляют собой индивидуальные физические признаки человека;
- комбинированные - для идентификации используются одновременно несколько идентификационных признаков.

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		9

б) по способу считывания идентификационных признаков устройства ввода систем контроля доступом могут быть:

- с ручным вводом - ввод производится с помощью нажатия клавиш, поворотом переключателей или других подобных элементов систем управления доступом;
- контактные - ввод происходит при непосредственном, в том числе и при электрическом, контакте между считывателем и идентификатором;
- дистанционные - в системах контроля и управления доступом считывание кода происходит при поднесении идентификатора на определенное расстояние к считывателю;
- комбинированные.

3) Устройства управления:

- а) аппаратные;
- б) программные;
- в) программно-аппаратные средства.

– По устойчивости к разрушающим воздействиям средства контроля доступа классифицируются по:

- 1) устойчивости к взлому;
- 2) пулестойкости;
- 3) устойчивости к взрыву.

– По устойчивости к неразрушающим воздействиям средства контроля и управления доступом в зависимости от их назначения классифицируют по следующим показателям:

- 1) устойчивости к вскрытию - для управляемых преграждающих и исполнительных устройств систем контроля доступом (замков электромеханических, электромагнитных, электронных, запорных механизмов);
- 2) устойчивости к манипулированию;

- 3) устойчивости к наблюдению - для устройств, входящих в состав систем контроля и управления доступом и предназначенных для ввода идентификационных признаков с запоминаемым кодом (клавиатуры, кодовые переключатели и т.п.);
 - 4) устойчивости к копированию (для идентификаторов);
 - 5) устойчивости защиты устройств управления в системах, осуществляющих контроль доступа, от несанкционированного доступа к информации.
- По устойчивости к несанкционированному доступу:
- 1) нормальной;
 - 2) повышенной;
 - 3) высокой.

Классификацию систем контроля и управления доступом производят по следующим признакам:

- Способу управления системой контроля доступа:
- 1) автономные - для управления одним или несколькими управляемыми преграждающими устройствами без передачи информации на центральный пульт и без контроля со стороны оператора;
 - 2) централизованные - для управления преграждающими устройствами с обменом информацией с центральным пультом и контролем и управлением системой со стороны оператора;
 - 3) универсальные - включающие функции как автономных, так и сетевых систем, работающие в сетевом режиме под управлением центрального устройства управления и переходящие в автономный режим при возникновении отказов в сетевом оборудовании, в центральном устройстве или обрыве связи.
- Количеству контролируемых точек доступа:
- 1) малой емкости;
 - 2) средней емкости;
 - 3) большой емкости.

- Функциональным характеристикам:
 - 1) системы контроля и управления доступом с ограниченными функциями;
 - 2) системы контроля и управления доступом с расширенными функциями;
 - 3) многофункциональные СКУД.
- Виду объектов контроля, осуществляющие:
 - 1) контроль доступа физических объектов;
 - 2) контроль доступа к информации.
- Уровню защищенности системы от несанкционированного доступа к информации.
 - 1) нормальной;
 - 2) повышенной;
 - 3) высокой.
- По устойчивости к разрушающим воздействиям:
 - 1) устойчивости к взлому;
 - 2) пулестойкости;
 - 3) устойчивости к взрыву.
- По устойчивости к неразрушающим воздействиям системы контроля и управления доступом, также как средства контроля доступа, в зависимости от их функционального назначения классифицируют по следующим показателям:
 - 1) устойчивости к вскрытию - для управляемых преграждающих и исполнительных устройств;
 - 2) устойчивости к манипулированию;
 - 3) устойчивости к наблюдению - для устройств ввода идентификационных признаков с запоминаемым кодом;
 - 4) устойчивости к копированию;
 - 5) устойчивости защиты средств вычислительной техники устройств управления от несанкционированного доступа к информации.

На текущий момент на предприятии организована обычная проходная, состоящая из вахтёра, который сравнивает фотографию в пропуске с человеком, предъявившим его.

Преимущества текущей системы контроля доступом:

- В сравнении с отсутствием какой-либо системы контроля доступом, обеспечивается минимальный уровень контроля за персоналом и борьбой с хищениями на предприятии.

Недостатки обычной проходной:

- Присутствует человеческий фактор в процессе обеспечения контроля на проходной;
- Необходимо содержать большой штат вахтёров при необходимости контроля;
- Не обеспечивается необходимый уровень контроля за посетителями предприятия;
- Невозможно выполнять множество полезных функций, которые могут быть обеспечены добавлением дополнительных программных модулей к автоматизированному проходным.

1.2 Обзор аналогов

При проведённом анализе предметной области были выявлены, наиболее популярные в России системы контроля и управления доступом:

Системы контроля и управления доступом "Parsec" [2]:

Программно-аппаратные комплексы данного производителя позволяют, организовать системы контроля и управления доступом различного масштаба. В качестве оборудования, используются как обычные компьютеры, объединенные в локальную сеть, так и специализированные контроллеры, турникеты, считыватели, преобразователи интерфейсов и различные дополнительные модули различного назначения.

					09.03.01.2017.064.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

Программное обеспечение **ParsecNET** имеет множество функций для конфигурирования различных систем контроля и управления доступом, позволяющими достаточно эффективно обеспечивать контроль за территорией предприятия. Функциональные возможности программного обеспечения имеют следующие возможности:

- Организация многопользовательской системы с возможностью одновременного изменения данных с различных рабочих мест;
- Создание сложных конфигураций для территориально распределённых объектов;
- Автоматический поиск оборудования;
- Мониторинг и управление;
- Получение данных пользователей;
- Возможность настраивать систему для решения различных задач.

Система контроля и управления доступом "Perco" [3]:

С помощью систем «Perco» создают системы контроля и управления доступом, которые повышают обеспечиваемый уровень безопасности на предприятии и контроля за сотрудниками и посетителями организации. Кроме того, как и остальные аналоги подобных систем контроля и управления доступом на рынке подобных систем, позволяют решать множество второстепенных, но не менее важных для организации задач. Система позволяет значительно сократить расходы для обеспечения контроля и безопасности в долгосрочной перспективе за счёт необходимости содержать меньший штат сотрудников при наличии системы контроля доступом, чем при отсутствии такого комплекса.

«Perco» помогает обеспечивать следующие функции:

- Контроль на проходной;
- Контроль доступа автомобилей;
- Доступ в помещения;
- Организация центрального поста охраны;
- Учёт рабочего времени.

В данной системе реализовано множество функций, которые позволяют организовать различные конфигурации систем контроля доступом, подходящие под требования, предъявляемые к системе заказчиком.

Цифровой лингафонный кабинет КОДОС [4]:

Система контроля и управления доступом КОДОС, обеспечивает возможности построения системы, обладающей функциями контроля действий операторов и вахтёров, а также ведущей контроль за сотрудниками и посетителями организации.

Основные функции системы контроля доступа КОДОС:

- Разграничение доступа и контроля ситуации на объекте;
- Предотвращение несанкционированного доступа;
- Контроль рабочего времени.

Проведя анализ имеющихся аналогов, можно сделать вывод о том, что существующие системы контроля и управления доступом выполняют примерно одинаковый функционал, который подбирается под нужды предприятия в зависимости от выделенного на построение подобной системы бюджета и поставленной задачи по обеспечению контроля на предприятии.

1.3 Преимущества разрабатываемой программной части системы контроля и управления доступом

Главным преимуществом разрабатываемого программного обеспечения является простота внедрения и эксплуатации, а также значительная экономия на системе лицензирования в будущем. Необходимый функционал программного обеспечения для системы контроля доступа на проходной разрабатывается согласно требованиям службы безопасности.

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		15

Оснащение проходной предприятия автоматизированным турникетом и разработанным программным обеспечением позволит:

- Исключить вероятность прохода нескольких людей по одному пропуску;
- Фиксировать не соблюдение рабочего времени, тем самым повысить уровень соблюдения трудовой дисциплины;
- Фиксировать все входы и выходы сотрудников в рабочее время;
- Снизить расходы на содержании штата вахтёров.

1.4 Постановка задачи

Необходимо разработать программное обеспечение для аппаратно-программного комплекса автоматизированной проходной, позволяющее повысить уровень обеспечиваемого контроля на входе в организацию.

Основой для разработки послужила существующая проходная, которая в силу преобладания человеческого фактора в моменты большого прохода людей не могла обеспечивать должный уровень контроля за входящими на предприятие людьми, так как имелась вероятность проникновения людей, не работающих на предприятии.

Так как необходимо разработать программное обеспечение, аппаратная часть в виде турникета, считывателей и контроллера будет выбрана позднее после согласования и выбора оптимального оборудования, имеющегося в наличии у поставщиков. Проходная будет построена по аналогу, изображённого на рисунке 1.1.

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		16



Рисунок 1.1 – Автоматизированная проходная

В программном обеспечении должны быть реализованы следующие функции:

- Получение отчётов;
- Вывод фотографии сотрудника на экран при осуществлении прохода;
- Редактирование списка сотрудников в режиме администратора;
- Редактирование информации о доступе.

1.5 Выбор средств и технологий

Для разработки программного продукта была выбрана интегрированная среда разработки программного обеспечения «Delphi» и локальная реляционная система управления базами данных «Access» для хранения информации для системы контроля и управления доступом, а также для получения отчётов используется программа «Excel» для работы с таблицами из пакета Microsoft Office.

Delphi RAD Studio - среда разработки программного обеспечения, относящаяся к интегрированным средам разработки, включающим в себя текстовый редактор, компилятор и интерпретатор, средства автоматизации сборки и отладчик. Используемый язык программирования в этой среде разработки называется Delphi, который ранее носил название Object Pascal, но после добавления в него и в среду разработки новых возможностей был переименован. Данная среда разработки имеет в себе огромное количество программных инструментов для разработчиков программного обеспечения, которых хватает для разработки большинства программных продуктов.

Реляционная система управления базами данных «Access» выбрана в качестве используемой базы данных, так как требуется работа с небольшим объёмом данных и её использование происходит только на персональном компьютере проходной с планируемой в будущем синхронизацией с клиент-серверной системой управления базами данных.

Microsoft Excel в виду её возможности работы с таблицами выбрана для создания и вывода отчётов программы в удобном для пользователя виде.

Разработка с использованием ранее перечисленных программ позволяет:

- Быстро разработать программу и обеспечить её совместимость со всеми современными операционными системами Windows;
- В случае возникновения критических ошибок при разработке моментально их исправлять.

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		18

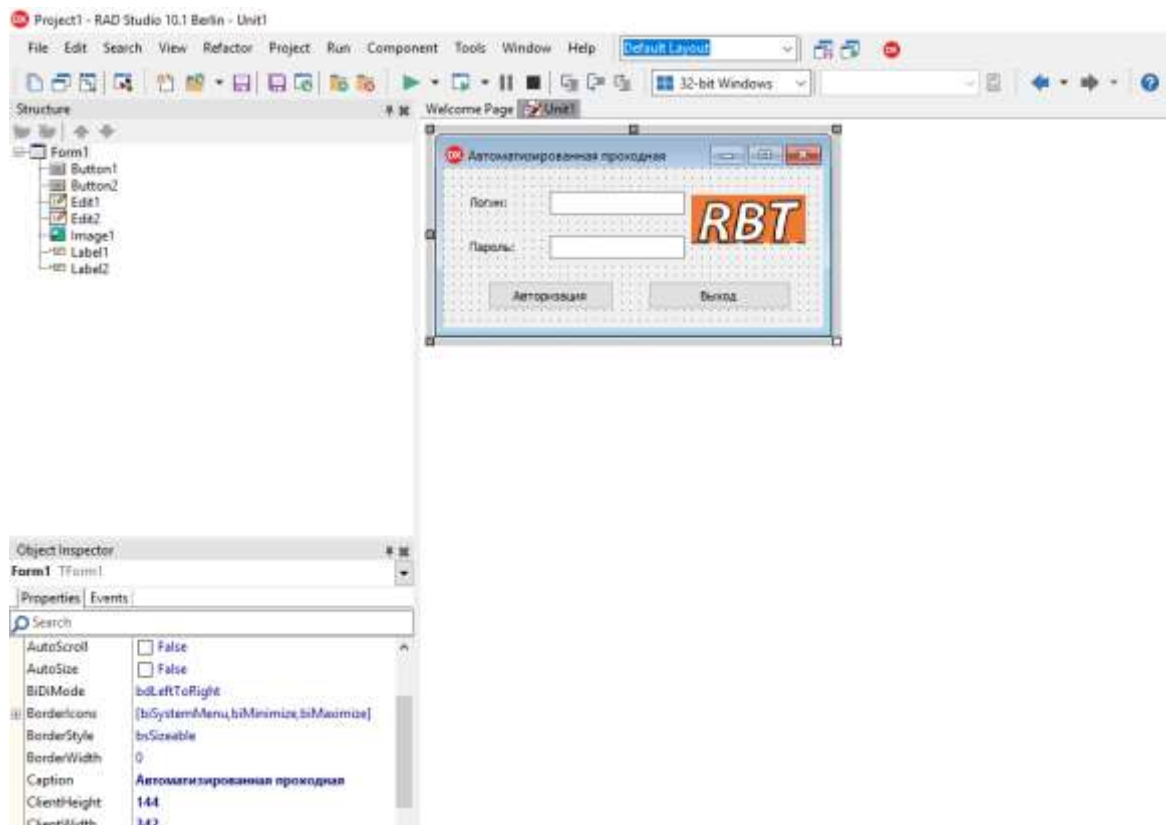


Рисунок 1.2 – Рабочая область среды разработки RAD Studio Delphi

Среда разработки Delphi позволяет разрабатывать приложения достаточно быстро за счёт наличия наборов визуальных компонентов для форм приложения. Без использования визуальных элементов процесс разработки программного обеспечения может требовать написание дополнительного кода и ведения разработки в модели программирования Windows Runtime или иных подобных моделях программирования для других операционных систем. Компоненты приложения при использовании такой модели программирования создаются в момент компиляции приложения. Подобная модель разработки влечёт за собой увеличение трудоёмкости поставленной задачи и времени необходимого для разработки готового продукта.

Визуальные компоненты позволяют тратить на разработку форм меньшее количество времени и в случае необходимости исправлений на форме оперативно вносить их. Все визуальные компоненты среды разработки находятся на панели инструментов (рисунок 1.3).

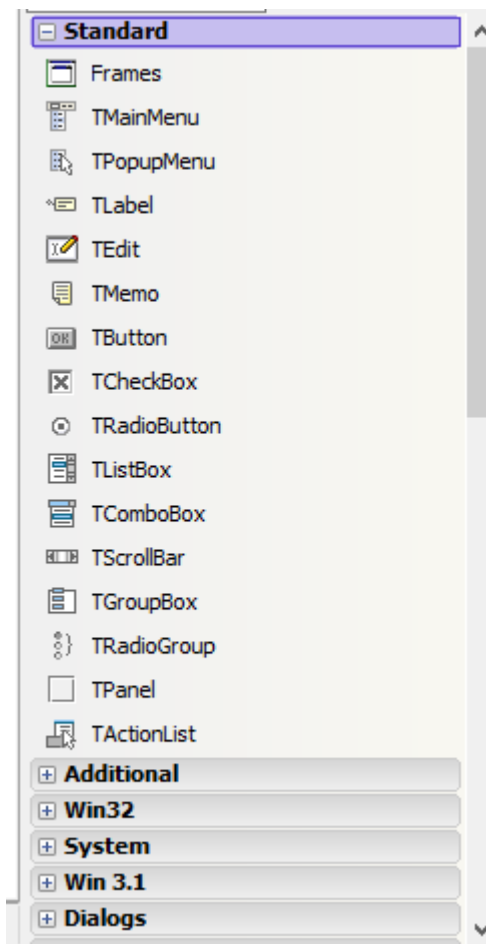


Рисунок 1.3 – Панель визуальных инструментов Delphi

Через панель с названием «Инспектор объектов» (рисунок 1.4) можно изменять доступные свойства выбранного для редактирования в текущий момент объекта интерфейса.

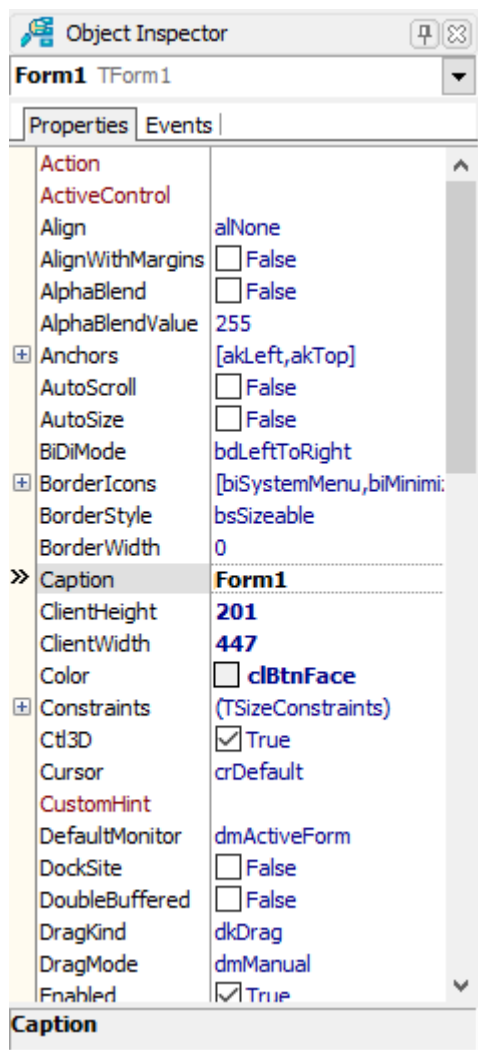


Рисунок 1.4 – Панель свойств элементов интерфейса «Инспектор объектов»

Изменение свойств любого объекта через «Инспектор объектов» позволяет в большинстве случаев увидеть визуальные изменения ещё до этапа компиляции программы.

Преимуществами среды разработки Delphi в сравнении с аналогами от других разработчиков являются:

- Быстрая разработка программного обеспечения;
- Обеспечение достаточной производительности приложений;
- Наличие большого количества дополнительных функций, обеспечиваемых встраиванием новых компонент и инструментов в среду Delphi от сторонних разработчиков;

- Возможность собственной разработки дополнительных модулей для расширения функционала среды разработки.

Выбор системы управления базами данных осуществлялся на основе простоты работы с ней на выбранном языке программирования и доступности данной базы данных для небольших проектов и в случае необходимости есть возможность синхронизации, разработанной в ней базы с более крупной системой управления базами данных Microsoft SQL Server. Также выбранная среда разработки содержит набор инструментов, позволяющих быстро подключать написанные приложения к различным системам управления базами данных включая «Access». Реляционная система управления базами данных «Access» (рисунок 1.5) имеет необходимые для разработки программного обеспечения характеристики:

- Позволяет создавать локальные базы данных небольшого размера, что обеспечивает быструю скорость обработки SQL-запросов;
- Отсутствует необходимость устанавливать на рабочее место пользователя SQL-сервер;
- Access поддерживает весь необходимый функционал для работы с таблицами локальной базы данных.

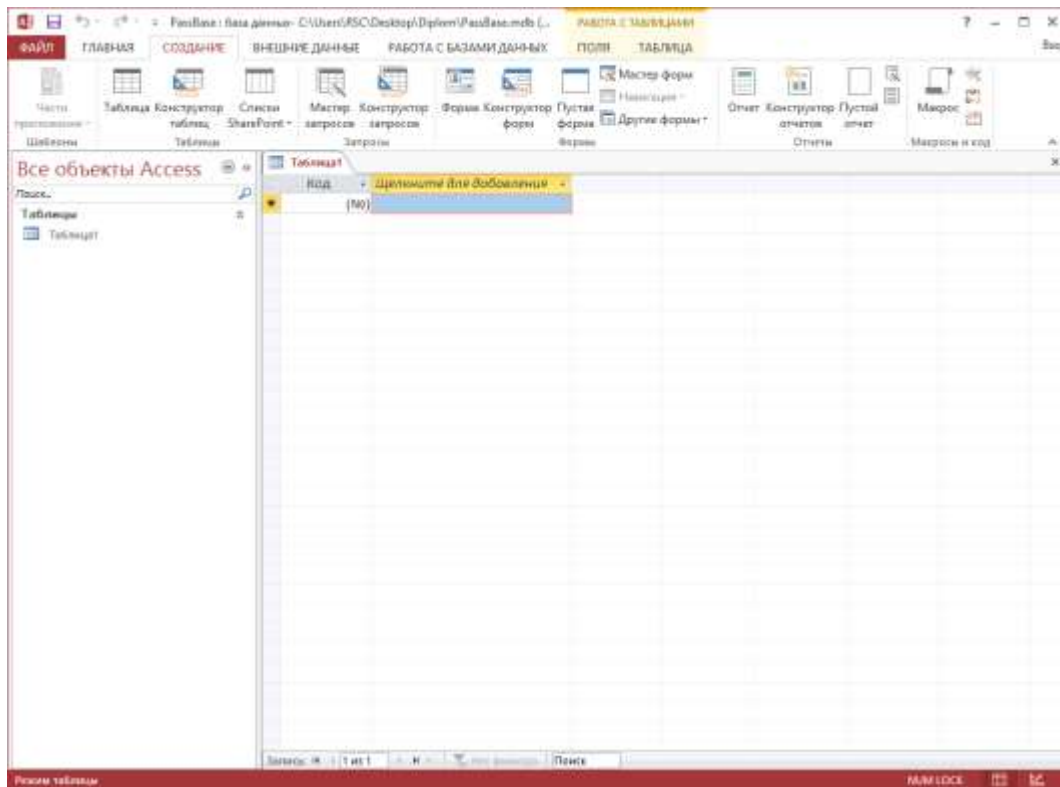


Рисунок 1.5 – Интерфейс системы управления базами данных «Access»

В режиме таблице на рисунке выше можно просматривать содержимое выбранной в текущий момент таблицы.

Также данная система управления базами данных, имеет множество удобных инструментов, которые могут пригодится при разработке небольшой базы данных. Например, конструктор схемы данных (рисунок 1.6), необходимый для быстрого построения правильных связей между таблицами разрабатываемой базы данных.

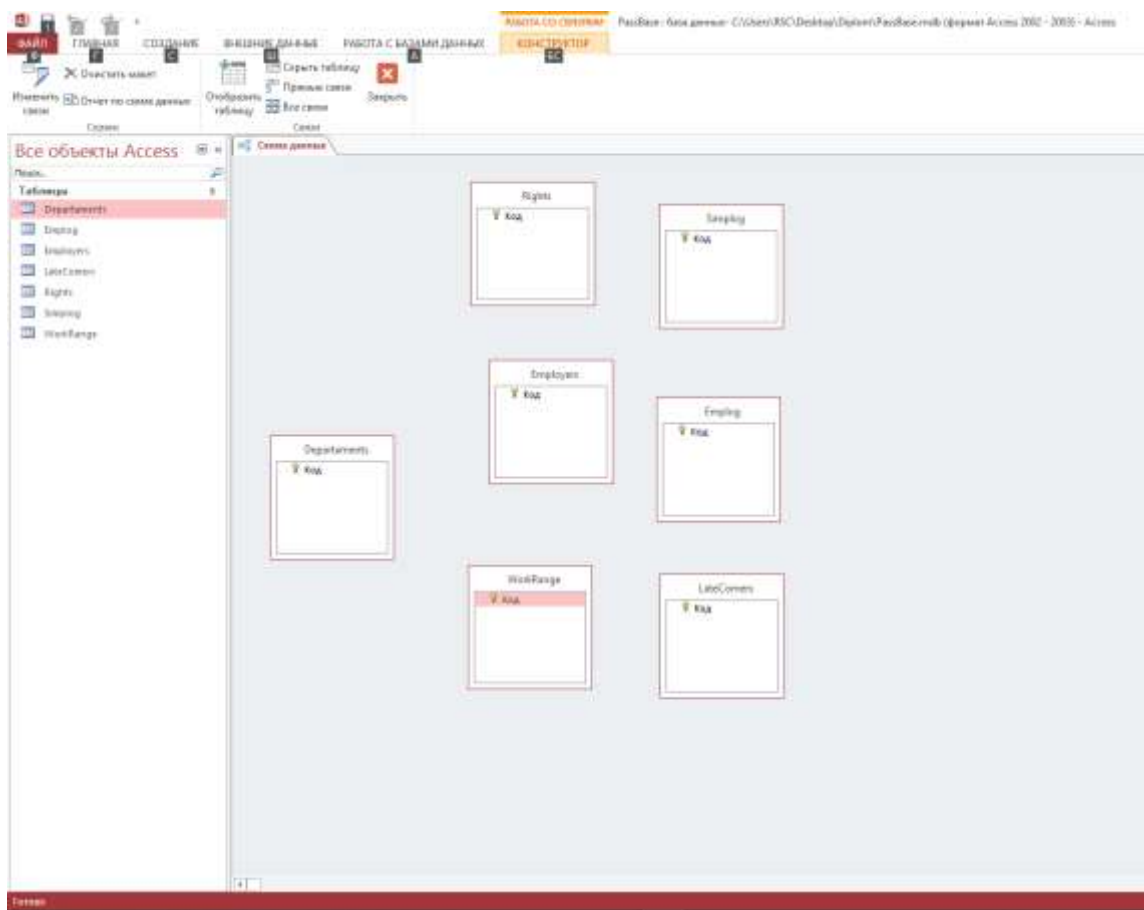


Рисунок 1.6 – Конструктор схемы данных

Конструктор позволяет создавать связи между таблицами базы данных следующих типов:

- Связь «один ко многим»;
- Связь «многие ко многим»;
- Связь «один к одному».

Конструктор таблиц (рисунок 1.7), содержащийся в программе позволяет ускорить разработку базы данных благодаря удобному и простому интерфейсу, а также содержащему информацию о всех необходимых типах данных.

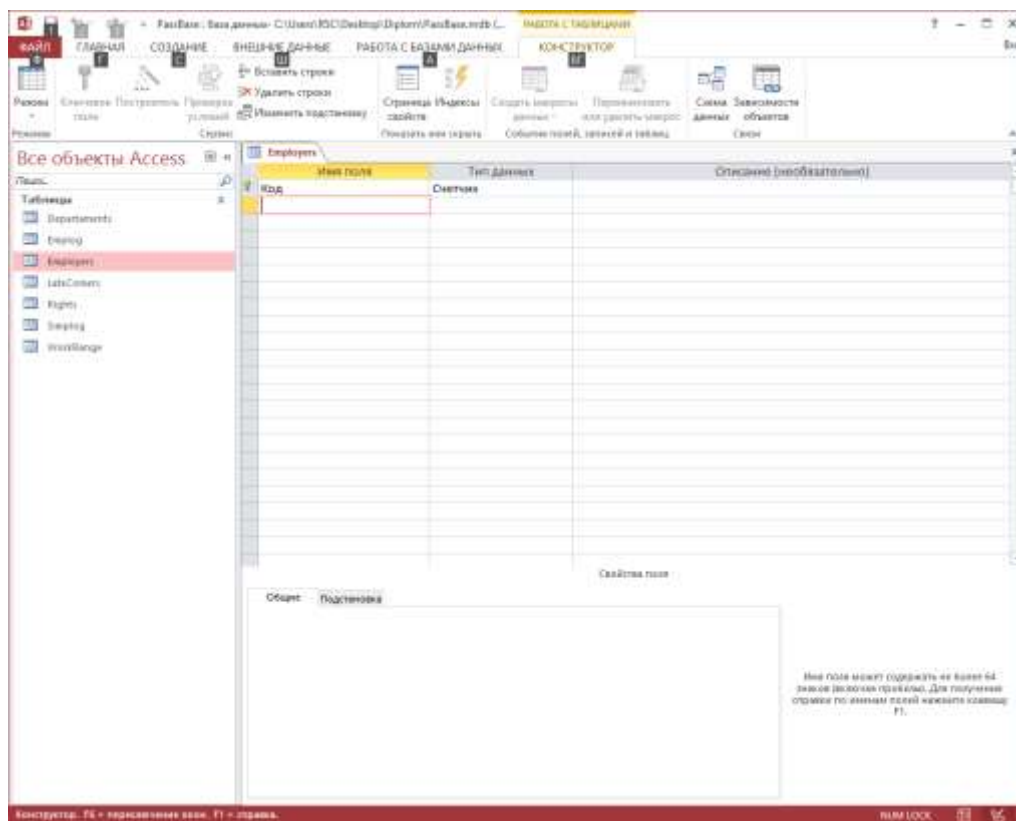


Рисунок 1.7 – Конструктор схемы данных

Изм.	Лист	№ докум.	Подпись	Дата

09.03.01.2017.064.00 ПЗ

Лист

25

Программа для работы с электронными таблицами Excel (рисунок 1.8) была выбрана для создания отчётов из разрабатываемого программного обеспечения, так как она включена в приобретённый пакет стороннего программного обеспечения необходимого для получения нужного функционала, а также удобного получения отчётов с помощью языка разработки Delphi.

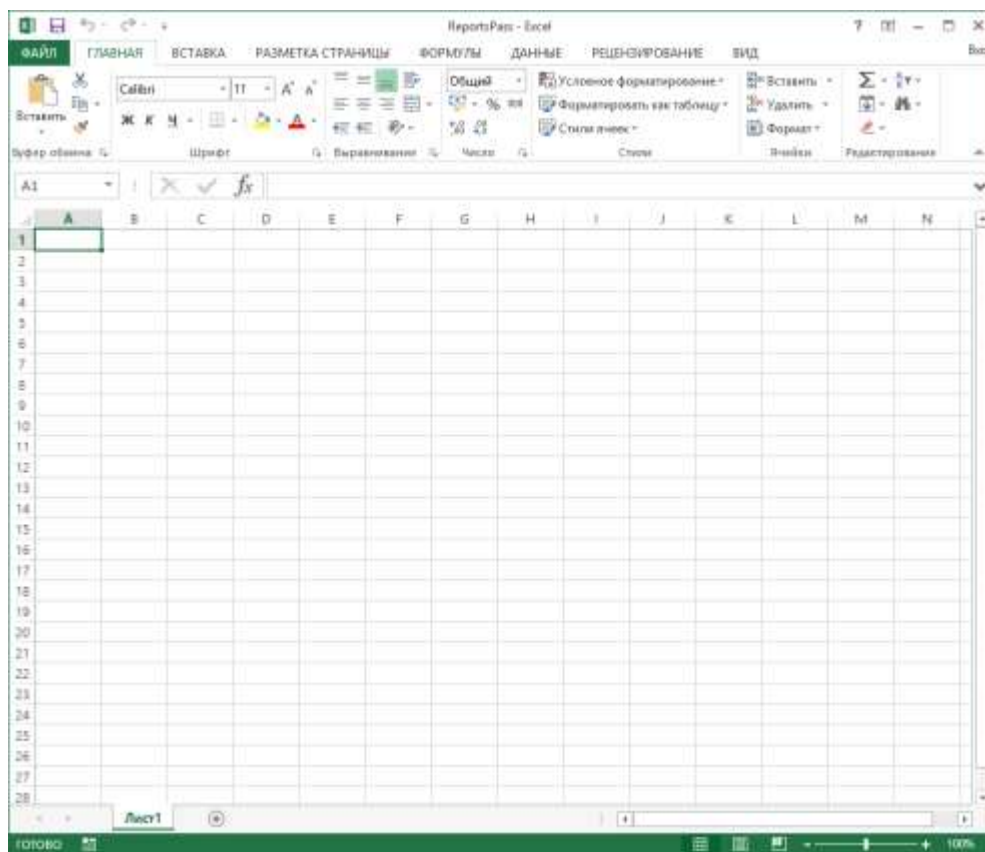


Рисунок 1.8 – Окно программы для работы с электронными таблицами «Excel»

Также получение отчётов из программы в Excel не требует заполнения шаблона в самом офисном приложении, так как шаблон прописывается в коде разрабатываемого программного обеспечения.

1.6 Требования к интерфейсу приложения

Перед разработчиками программного обеспечения сейчас стоит проблема выбора нужного средства разработки графического интерфейса из-за их большого количества. Это затрудняет оценку обеспечиваемых конкретным средством разработки возможностей необходимых для выполнения поставленных задач.

Средство разработки графического интерфейса в основном должно сокращать время разработки программного обеспечения и снижать количество доработок в будущем.

Пользовательский интерфейс – это часть программы, которая постоянно находится в поле зрения пользователя, но многие разработчики оставляют дизайн интерфейса на потом, что часто вызывает недовольство пользователя разработанным программным обеспечением. Конечный пользователь не видит программного кода и поэтому расположение элементов интерфейса для него намного важнее того, что происходит в исполняемом коде.

Для создания хорошего интерфейса необходимо понимать его назначение и как связаны между собой различные окна программы, поэтому пользовательский интерфейс должен обеспечивать функционал, чтобы трудностей при его использовании возникало как можно у меньшего количества людей, так как все имеют различные навыки работы с персональными компьютерами.

Разработка форм ввода должна вестись так, чтобы они обеспечивали быструю скорость своей работы. Для выполнения этой задачи необходимо соблюдать следующие требования:

- Обеспечить клавиатурные эквиваленты команд, которые должны сочетаться с другими устройствами ввода;
- Расположение элементов интерфейса должно быть продумано, чтобы обеспечивать быстрое выполнение пользователем задач;
- Использование заметной ненавязчивой обратной связи;

					09.03.01.2017.064.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

- Добавление и редактирование записей желательно выполнять на одной рабочей форме.

Существует два вида интерфейсов: однодокументный (SDI) и многодокументный (MDI). Независимо от используемого в разработке приложения вида интерфейса, программное обеспечение должно быть написано так, чтобы у пользователя не было возможностей нарушить функционирование программы.

Меню в программном интерфейсе также являются важной частью разработки, поэтому для проектирования эффективных меню существуют следующие рекомендации:

- Желательно располагать уже ставшими стандартными следующие названия пунктов меню в Windows: Файл, Редактировать, Просмотр и т.д.;
- При группировке пунктов необходимо соблюдать логический порядок, а также обращать внимание на содержание;
- Нежелательно перегружать меню большим количеством подменю и пунктов;
- Обязательно некоторым пунктам меню назначать клавиатурные эквиваленты;
- Панель инструментов должна содержать часто используемые команды меню.

Используя визуальную среду программирования, разработчик получает доступ ко множеству инструментов, которые при правильном подходе к выполнению работы должны свести недочёты при проектировании интерфейса к минимуму, а также уменьшить количество исправлений в программе в будущем.

Принципиальным преимуществом работы с приложением является интуитивно понятный интерфейс, но для упрощения и ускорения процесса освоения пользователем процесса работы с программой в программе должен присутствовать раздел справки, чтобы облегчить процесс изучения сложного программного обеспечения.

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
						28
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА

2.1 Теоретическое обоснование

Для разработки программного обеспечения системы контроля управления доступом необходимо определить то как будет выглядеть автоматизированная проходная, так как функционал программного обеспечения будет напрямую зависеть от используемых аппаратных модулей.

Разработка программного обеспечения разделена на следующие части:

- Создание приблизительной схемы простой автоматизированной проходной;
- Разработка программного обеспечения.

Для выполнения поставленной задачи были использованы некоторые наработки многих фирм производителей систем контроля и управления доступом.

Аппаратная часть системы с которой будет работать разрабатываемое программное обеспечение будет выбрана позднее, поэтому будет приведён обобщённый список требуемого оборудования:

- Персональный компьютер;
- Контроллер;
- Физический пульт управления турникетом;
- Считыватели пластиковых карт доступа;
- Электромеханические турникеты.

Для автоматизации проходной подходит любой персональный компьютер нижнего ценового сегмента, так как обеспечиваемой ими производительности достаточно для выполнения подобной задачи. Компьютер необходим для хранения результатов работы в течение определённого срока.

Контроллер необходим для связи со считывателями, турникетом и пультом управления.

Физический пульт требуется для управления турникетом в экстренных ситуациях, а также в некоторых исключительных случаях. Например, для

										Лист
										29
Изм.	Лист	№ докум.	Подпись	Дата	09.03.01.2017.064.00 ПЗ					

обеспечения прохода сотрудников на предприятие в сопровождении начальника отдела, когда сотрудник забыл пропуск.

Считыватели требуются для получения идентификаторов карт доступа по которому определяется имеет ли человек право доступа на предприятие.

Общая схема автоматизированной проходной, устанавливаемой на пропускной пункт организации показан далее на рисунке 2.1.



Рисунок 2.1 – Схема автоматизированной проходной

2.1.1 Работа автоматизированной проходной

Поднесение карты доступа к считывателю передаёт с помощью него уникальный идентификатор контроллеру, который передаёт его через выбранный аппаратный интерфейс в установленное на компьютере программное обеспечение. В качестве аппаратного интерфейса был выбран Ethernet, так как при расширении системы в будущем он позволяет легко расширить систему контроля доступа с помощью существующей сети предприятия без проведения большого количества дополнительных кабельных сетей, а также уменьшает количество монтажных работ по их прокладке. Приложение опрашивает текущее состояние контроллера и при получении кода готовности для передачи считанного идентификатора карты доступа, отправляет запрос на его получение. Происходит поиск полученного идентификатора в локальной базе данных и после нахождения, определяются права доступа для сотрудника с соответствующим идентификатором.

Далее произошедшее событие фиксируется в журналы проходной. Журналы должны иметь удобный вид, чтобы обеспечивать возможность получения основных типов отчётов:

- Отчёт о входах и выходах, который показывает все попытки входа или выхода, в том числе и неудачные;
- Отчёт рабочего времени персонала, показывающий сколько времени сотрудник провёл на рабочем месте;
- Отчёт об отсутствии, содержащий список отсутствующих работников за определённый интервал времени;
- Отчёт опозданий, показывающий список сотрудников, опоздавших на работу за определённый временной интервал.

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		31

2.1.2 Программное обеспечение автоматизированной проходной

Разработанная программа устанавливается на компьютер находящийся на проходной и включает в себя функционал мониторинга, который доступен обычному работнику, а также функционал работы с локальной базой данных для учётной записи администратора. Разграничение функционала обеспечивается благодаря содержащейся в базе данных информации о доступе к тому или иному функционалу.

2.2 Разработка базы данных

Файл базы данных поставляемый с данным программным обеспечением находится непосредственно в одном каталоге с исполняемым файлом, что обеспечивает максимально быструю работу программного обеспечения, а также гарантирует автономность работы проходной. В базе данных хранятся сведения о сотрудниках, которые имеют доступ на предприятие. Для обеспечения работы программы с базой данных не требуется установка Microsoft Access.

База данных располагается в одном единственном файле с расширением mdb. В качестве пути расположения к файлу базы данных определён каталог с исполняемым файлом программы.

2.2.1 Логическая схема используемой базы данных

Схема базы данных разработана для хранения информации об отделах предприятия, сотрудниках, правах доступа, а также для хранения отчётности о зафиксированных событиях, произошедших во время работы проходной. Схема показана на рисунке 2.2.

									Лист
									32
Изм.	Лист	№ докум.	Подпись	Дата	09.03.01.2017.064.00 ПЗ				

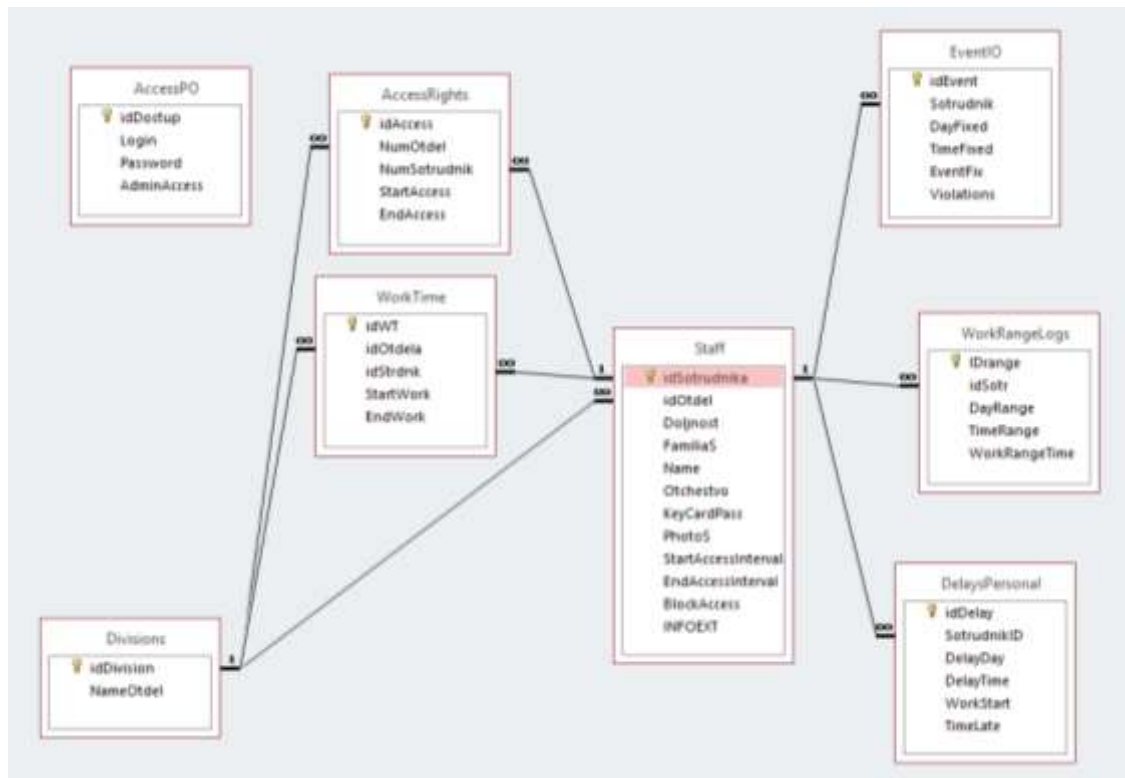


Рисунок 2.2 – Схема локальной базы данных

Связи между таблицами установлены так, чтобы информация о сотрудниках и зафиксированных событиях обеспечивали получение достоверной информации о любом сотруднике предприятия.

2.2.2 Описание таблиц базы данных Access

Таблица «Divisions» содержит список отделов организации. Её поля описаны в таблице 2.1.

Таблица 2.1 – Описание полей таблицы «Division»

Название столбца в таблице	Описание
idDivision	Индекс отдела в таблице базы данных.
NameOtdel	Название отдела в организации.

Таблица «Staff» обеспечивает хранение информации о сотрудниках организации. Описание полей в таблице 2.2.

Таблица 2.2 – Описание полей таблицы «Staff»

Название столбца в таблице	Описание
idSotrudnika	Индекс сотрудника в таблице.
idOtdel	Номер отдела в котором числится сотрудник.
Doljnost	Название занимаемой должности.
FamiliaS	Фамилия сотрудника.
Name	Имя сотрудника.
Otchestvo	Отчество сотрудника.
KeyCardPass	Идентификатор пропуска.
PhotoS	Содержит название фотографии для сотрудника.
StartAccessInterval	Начало интервала доступа по пропуску.
EndAccessInterval	Конец интервала доступа по пропуску.
BlockAccess	Информация о блокировке пропуска.
INFOEXT	Заметки о сотруднике.

Таблица «AccessRights» определяет время доступа сотрудников на предприятие. Поля таблицы описаны в таблице 2.3.

Таблица 2.3 – Описание полей таблицы «AccessRights»

Название столбца в таблице	Описание
idAccess	Индекс интервала доступа.
NumOtdel	Номер отдела.
NumSotrudnik	Номер сотрудника.
StartAccess	Начало доступа.
EndAccess	Конец доступа.

Таблица «WorkTime» обеспечивает хранение информации о времени нахождения сотрудника на рабочем месте. Поля таблицы описаны в таблице 2.4.

Таблица 2.4 – Описание полей таблицы «WorkTime»

Название столбца в таблице	Описание
idWT	Индекс рабочего диапазона.
idOtdela	Номер отдела.
idStrdnk	Номер сотрудника.
StartWork	Начало рабочего времени.
EndWork	Конец рабочего времени.

Таблица «EventIO» обеспечивает хранение информации о произошедших событиях на проходной. Описание полей таблицы произведено в таблице 2.5.

Таблица 2.5 – Описание полей таблицы «EventIO»

Название столбца в таблице	Описание
idEvent	Индекс события.
Sotrudnik	Номер сотрудника.
DayFixed	Зафиксированная дата события.
TimeFixed	Зафиксированное время события.
EventFix	Событие проходной.
Violations	Дополнительная информация в случае нарушения.

Таблица «WorkRangeLogs» хранит информацию о времени проведенным сотрудниками на рабочем месте. Поля таблицы описаны в таблице 2.6.

Таблица 2.6 – Описание полей таблицы «WorkRangeLogs»

Название столбца в таблице	Описание
iDrange	Индекс временного диапазона рабочего времени.
idSotr	Номер сотрудника.
DayRange	Дата фиксации диапазона рабочего времени.
TimeRange	Время фиксации диапазона рабочего времени.
WorkRangeTime	Время, проведенное сотрудником на работе.

Таблица «DelaysPersonal» является журналом, содержащем записи об опозданиях сотрудников. Описание её полей приведено в таблице 2.7.

Таблица 2.7 – Описание полей таблицы «DelaysPersonal»

Название столбца в таблице	Описание
idDelay	Индекс события «Опоздание».
SotrudnikID	Номер сотрудника.
DelayDay	Дата опоздания.
DelayTime	Время опоздания.
WorkStart	Требуемое время прихода сотрудника.
TimeLate	Количество времени опоздания.

Таблица «AccessPO» содержит информацию о доступе в программное обеспечение для сотрудников службы безопасности. Описание полей таблицы приведено в таблице 2.8.

Таблица 2.8 – Описание полей таблицы «AccessPO»

Название столбца в таблице	Описание
idDostup	Индекс учётной записи.
Login	Название учётной записи для авторизации в программе.
Password	Пароль для авторизации.
AdminAccess	Свойство для отличия учётных записей администраторов и операторов проходной.

2.2.3 Создание файла базы данных

Создать ранее описанную локальную базу данных можно с помощью программного обеспечения «Microsoft Access», которое входит в состав пакета программ «Microsoft Office». Инструкцию для начала работы с этой системой управления базами данных можно найти на сайте компании производителя [6].

После создания базы данных в среде разработки Delphi необходимо подключить её через компонент ADOConnection. Для этого необходимо настроить свойство компонента «ConnectionString» и свойство Connected установить в положение «True».

Для настройки подключения к базе у свойства «ConnectionString» нажмём кнопку «...» и появившемся окне выберем строку «Use Connection String». Данное программное окно показано на рисунке 2.3.

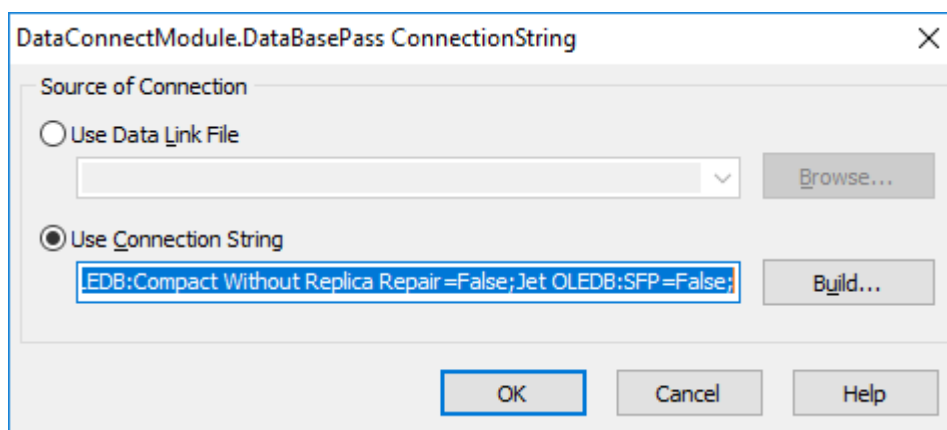


Рисунок 2.3 – Первый этап подключения к базе данных

Далее необходимо нажать кнопку «Build» и получить окно в котором надо выбрать драйвер подключения к базе данных «Microsoft Jet 4.0 OLE DB Provider». Этот момент показан на рисунке 2.4.

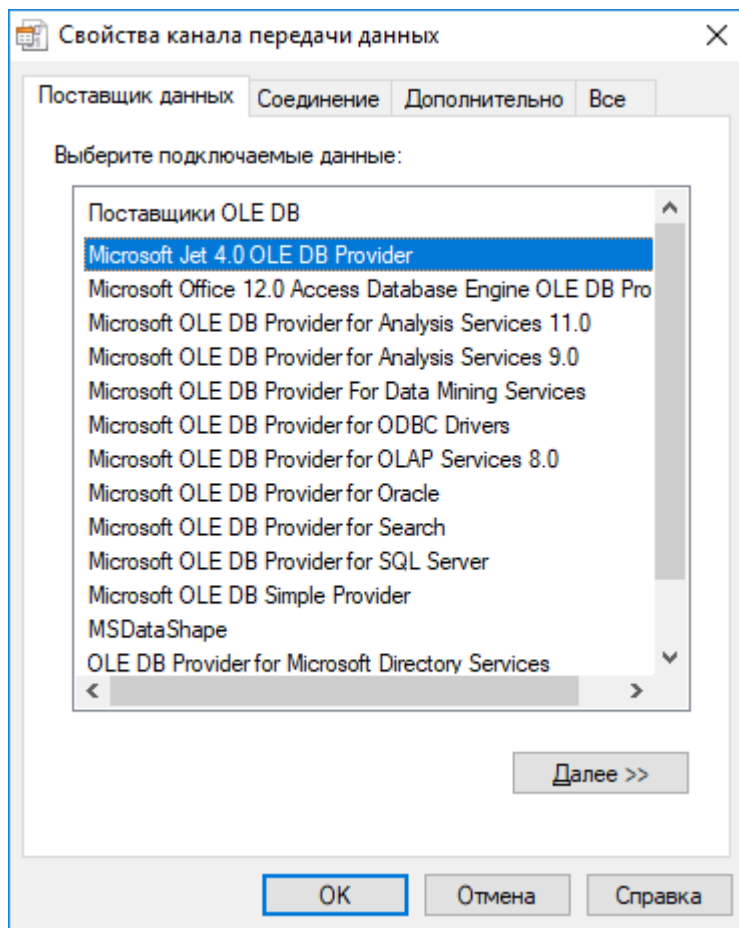


Рисунок 2.4 – Второй этап подключения базы данных

Затем на вкладке «Соединение» указать путь к базе данных. При необходимости также данные для подключения к базе данных. Чтобы приложение обращалось к файлу, находящемуся в каталоге с исполняемым файлом необходимо после указания пути к файлу стереть весь путь и оставить только название файла и нажать кнопку «Проверить соединение». Последний этап подключения базы данных к проекту показан на рисунке 2.5.

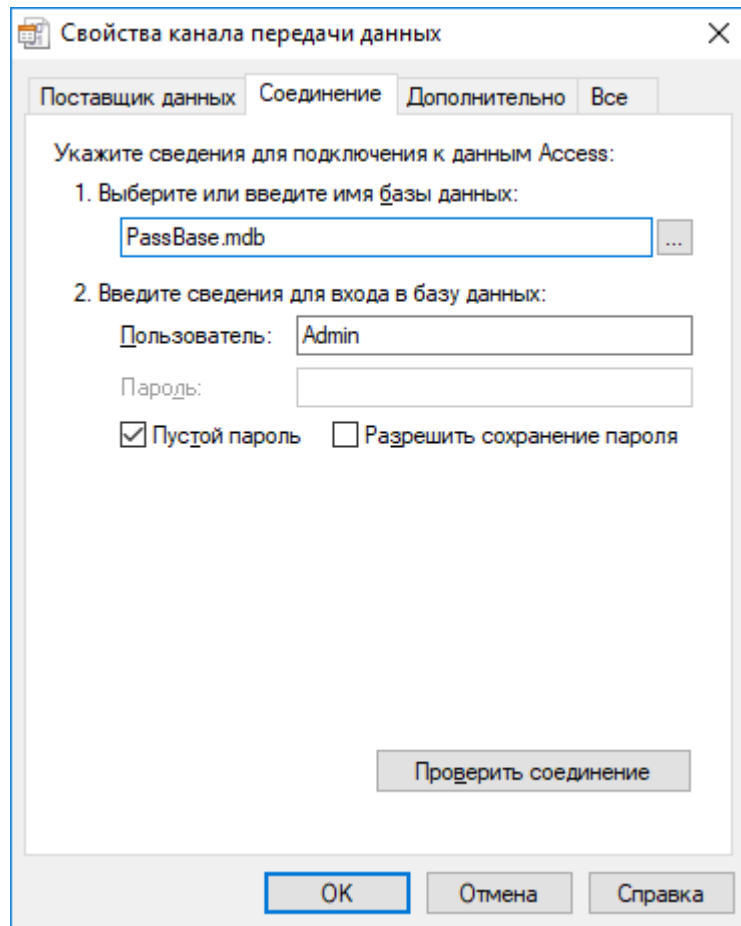


Рисунок 2.5 – Заключительный этап подключения базы данных

Изм.	Лист	№ докум.	Подпись	Дата

09.03.01.2017.064.00 ПЗ

Лист

40

2.3 Модуль авторизации программного обеспечения

Функционал этого модуля сводится к тому, чтобы предотвратить несанкционированный доступ посторонних лиц к работе программы или части её функций. При запуске приложения появляется форма, которая запрашивает данные для авторизации пользователя. По умолчанию присутствуют две учётные записи с ролями: «Администратор» и «Оператор». Администратор имеет доступ к управлению информацией в базе данных и настройке прав доступа в программном обеспечении. Оператору доступна лишь форма мониторинга и наблюдения за правильностью работы программы, а также разрешения возникающих проблем в различных внештатных ситуациях.

Пароли хранятся в таблице «AccessPO» базы данных «PassControl.mdb». Пароли хешируются алгоритмом MD5. Далее приведён фрагмент функции хеширования паролей:

```
Function md5(s:string):string;
  var a:array[0..15] of byte;
      i:integer;

  LenHi, LenLo: longword;
  Index: DWord;
  HashBuffer: array[0..63] of byte;
  CurrentHash: array[0..3] of DWord;

  procedure Burn;
  begin
    LenHi:= 0; LenLo:= 0;
    Index:= 0;
    FillChar(HashBuffer,Sizeof(HashBuffer),0);
    FillChar(CurrentHash,Sizeof(CurrentHash),0);
  end;

  procedure Init;
  begin
    Burn;
    CurrentHash[0]:= $67452301;
    CurrentHash[1]:= $efcdab89;
    CurrentHash[2]:= $98badcfe;
    CurrentHash[3]:= $10325476;
  end;
```

```
function LRot32(a, b: longword): longword;
begin
  Result:= (a shl b) or (a shr (32-b));
end;
```

```
procedure Compress;
```

```
var
```

```
  Data: array[0..15] of dword;
```

```
  A, B, C, D: dword;
```

```
begin
```

```
  Move(HashBuffer,Data,Sizeof(Data));
```

```
  A:= CurrentHash[0];
```

```
  B:= CurrentHash[1];
```

```
  C:= CurrentHash[2];
```

```
  D:= CurrentHash[3];
```

```
  A:= B + LRot32(A + (D xor (B and (C xor D))) + Data[ 0] + $d76aa478,7);
```

```
  D:= A + LRot32(D + (C xor (A and (B xor C))) + Data[ 1] + $e8c7b756,12);
```

```
  C:= D + LRot32(C + (B xor (D and (A xor B))) + Data[ 2] + $242070db,17);
```

```
  B:= C + LRot32(B + (A xor (C and (D xor A))) + Data[ 3] + $c1bdceee,22);
```

```
  A:= B + LRot32(A + (D xor (B and (C xor D))) + Data[ 4] + $f57c0faf,7);
```

```
  D:= A + LRot32(D + (C xor (A and (B xor C))) + Data[ 5] + $4787c62a,12);
```

```
  C:= D + LRot32(C + (B xor (D and (A xor B))) + Data[ 6] + $a8304613,17);
```

```
  B:= C + LRot32(B + (A xor (C and (D xor A))) + Data[ 7] + $fd469501,22);
```

```
  A:= B + LRot32(A + (D xor (B and (C xor D))) + Data[ 8] + $698098d8,7);
```

```
  D:= A + LRot32(D + (C xor (A and (B xor C))) + Data[ 9] + $8b44f7af,12);
```

```
  C:= D + LRot32(C + (B xor (D and (A xor B))) + Data[10] + $ffff5bb1,17);
```

```
  B:= C + LRot32(B + (A xor (C and (D xor A))) + Data[11] + $895cd7be,22);
```

```
  A:= B + LRot32(A + (D xor (B and (C xor D))) + Data[12] + $6b901122,7);
```

```
  D:= A + LRot32(D + (C xor (A and (B xor C))) + Data[13] + $fd987193,12);
```

```
  C:= D + LRot32(C + (B xor (D and (A xor B))) + Data[14] + $a679438e,17);
```

```
  B:= C + LRot32(B + (A xor (C and (D xor A))) + Data[15] + $49b40821,22);
```

```
  A:= B + LRot32(A + (C xor (D and (B xor C))) + Data[ 1] + $f61e2562,5);
```

```
  D:= A + LRot32(D + (B xor (C and (A xor B))) + Data[ 6] + $c040b340,9);
```

```
  C:= D + LRot32(C + (A xor (B and (D xor A))) + Data[11] + $265e5a51,14);
```

```
  B:= C + LRot32(B + (D xor (A and (C xor D))) + Data[ 0] + $e9b6c7aa,20);
```

```
  A:= B + LRot32(A + (C xor (D and (B xor C))) + Data[ 5] + $d62f105d,5);
```

```
  D:= A + LRot32(D + (B xor (C and (A xor B))) + Data[10] + $02441453,9);
```

```
  C:= D + LRot32(C + (A xor (B and (D xor A))) + Data[15] + $d8a1e681,14);
```

```
  B:= C + LRot32(B + (D xor (A and (C xor D))) + Data[ 4] + $e7d3fbc8,20);
```

```
  A:= B + LRot32(A + (C xor (D and (B xor C))) + Data[ 9] + $21e1cde6,5);
```

```
  D:= A + LRot32(D + (B xor (C and (A xor B))) + Data[14] + $c33707d6,9);
```

```
  C:= D + LRot32(C + (A xor (B and (D xor A))) + Data[ 3] + $f4d50d87,14);
```

Изм.	Лист	№ докум.	Подпись	Дата

B:= C + LRot32(B + (D xor (A and (C xor D))) + Data[8] + \$455a14ed,20);
 A:= B + LRot32(A + (C xor (D and (B xor C))) + Data[13] + \$a9e3e905,5);
 D:= A + LRot32(D + (B xor (C and (A xor B))) + Data[2] + \$fcefa3f8,9);
 C:= D + LRot32(C + (A xor (B and (D xor A))) + Data[7] + \$676f02d9,14);
 B:= C + LRot32(B + (D xor (A and (C xor D))) + Data[12] + \$8d2a4c8a,20);

A:= B + LRot32(A + (B xor C xor D) + Data[5] + \$fffa3942,4);
 D:= A + LRot32(D + (A xor B xor C) + Data[8] + \$8771f681,11);
 C:= D + LRot32(C + (D xor A xor B) + Data[11] + \$6d9d6122,16);
 B:= C + LRot32(B + (C xor D xor A) + Data[14] + \$fde5380c,23);
 A:= B + LRot32(A + (B xor C xor D) + Data[1] + \$a4beea44,4);
 D:= A + LRot32(D + (A xor B xor C) + Data[4] + \$4bdecfa9,11);
 C:= D + LRot32(C + (D xor A xor B) + Data[7] + \$f6bb4b60,16);
 B:= C + LRot32(B + (C xor D xor A) + Data[10] + \$bebfbc70,23);
 A:= B + LRot32(A + (B xor C xor D) + Data[13] + \$289b7ec6,4);
 D:= A + LRot32(D + (A xor B xor C) + Data[0] + \$eaa127fa,11);
 C:= D + LRot32(C + (D xor A xor B) + Data[3] + \$d4ef3085,16);
 B:= C + LRot32(B + (C xor D xor A) + Data[6] + \$04881d05,23);
 A:= B + LRot32(A + (B xor C xor D) + Data[9] + \$d9d4d039,4);
 D:= A + LRot32(D + (A xor B xor C) + Data[12] + \$e6db99e5,11);
 C:= D + LRot32(C + (D xor A xor B) + Data[15] + \$1fa27cf8,16);
 B:= C + LRot32(B + (C xor D xor A) + Data[2] + \$c4ac5665,23);

A:= B + LRot32(A + (C xor (B or (not D))) + Data[0] + \$f4292244,6);
 D:= A + LRot32(D + (B xor (A or (not C))) + Data[7] + \$432aff97,10);
 C:= D + LRot32(C + (A xor (D or (not B))) + Data[14] + \$ab9423a7,15);
 B:= C + LRot32(B + (D xor (C or (not A))) + Data[5] + \$fc93a039,21);
 A:= B + LRot32(A + (C xor (B or (not D))) + Data[12] + \$655b59c3,6);
 D:= A + LRot32(D + (B xor (A or (not C))) + Data[3] + \$8f0ccc92,10);
 C:= D + LRot32(C + (A xor (D or (not B))) + Data[10] + \$ffeff47d,15);
 B:= C + LRot32(B + (D xor (C or (not A))) + Data[1] + \$85845dd1,21);
 A:= B + LRot32(A + (C xor (B or (not D))) + Data[8] + \$6fa87e4f,6);
 D:= A + LRot32(D + (B xor (A or (not C))) + Data[15] + \$fe2ce6e0,10);
 C:= D + LRot32(C + (A xor (D or (not B))) + Data[6] + \$a3014314,15);
 B:= C + LRot32(B + (D xor (C or (not A))) + Data[13] + \$4e0811a1,21);
 A:= B + LRot32(A + (C xor (B or (not D))) + Data[4] + \$f7537e82,6);
 D:= A + LRot32(D + (B xor (A or (not C))) + Data[11] + \$bd3af235,10);
 C:= D + LRot32(C + (A xor (D or (not B))) + Data[2] + \$2ad7d2bb,15);
 B:= C + LRot32(B + (D xor (C or (not A))) + Data[9] + \$eb86d391,21);

Inc(CurrentHash[0],A);
 Inc(CurrentHash[1],B);
 Inc(CurrentHash[2],C);
 Inc(CurrentHash[3],D);
 Index:= 0;


```

FillChar(HashBuffer,Sizeof(HashBuffer),0);
end;

procedure Update(const Buffer; Size: longword);
var
  PBuf: ^byte;
begin
  Inc(LenHi,Size shr 29);
  Inc(LenLo,Size*8);
  if LenLo< (Size*8) then
    Inc(LenHi);

  PBuf:= @Buffer;
  while Size> 0 do
  begin
    if (Sizeof(HashBuffer)-Index)<= DWord(Size) then
      begin
        Move(PBuf^,HashBuffer[Index],Sizeof(HashBuffer)-Index);
        Dec(Size,Sizeof(HashBuffer)-Index);
        Inc(PBuf,Sizeof(HashBuffer)-Index);
        Compress;
      end
    else
      begin
        Move(PBuf^,HashBuffer[Index],Size);
        Inc(Index,Size);
        Size:= 0;
      end;
    end;
  end;
end;

procedure Final(var Digest);
begin
  HashBuffer[Index]:= $80;
  if Index>= 56 then Compress;
  PDWord(@HashBuffer[56])^:= LenLo;
  PDWord(@HashBuffer[60])^:= LenHi;
  Compress;
  Move(CurrentHash,Digest,Sizeof(CurrentHash));
  Burn;
end;

begin

```

					09.03.01.2017.064.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		44

```
Init;  
Update(s[1],Length(s));  
Final(a);  
result:="";  
for i:=0 to 15 do  
  result:=result+IntToHex(a[i],2);  
Burn;  
end;
```

					09.03.01.2017.064.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		45

2.4 Модуль данных формы мониторинга

Все элементы связи с базой данных располагаются в специальном модуле среды разработки Delphi называемом TDataModule.

Форма модуля представляет из себя поле для размещения компонентов для работы с базами данных, а также множества других не визуальных компонентов. Далее приведён список компонентов, которые необходимы для работы с базой данных:

- RightsQuery:ADOQuery – модуль для работы с набором данных, связанным с правами доступа сотрудников по присвоенному им идентификатору пропуска;
- AutoDeleteQuery:ADOQuery – обеспечивает удаление устаревших данных в журналах проходной;
- StatQuery:ADOQuery – обеспечивает фиксацию событий, возникающих при входе и выходе с предприятия.

Для установления связи этих компонентов с используемой базой данных в свойстве «Connection» каждого компонента надо указать основной модуль «DataBasePass:ADOConnection», используемый для соединения с базой данных. Установленное свойство показано на рисунке 2.6.

Active	<input checked="" type="checkbox"/> True
AutoCalcFields	<input checked="" type="checkbox"/> True
CacheSize	1
CommandTimeout	30
Connection	DataBasePass
ConnectionString	
CursorLocation	clUseClient
CursorType	ctStatic
DataSource	
EnableBCD	<input checked="" type="checkbox"/> True
ExecuteOptions	[]
Filter	
Filtered	<input type="checkbox"/> False
LiveBindings Designer	LiveBindings Designer
LockType	ltOptimistic
MarshalOptions	moMarshalAll
MaxRecords	0
Name	Autorizations
ParamCheck	<input checked="" type="checkbox"/> True
Parameters	(Parameters)
Prepared	<input type="checkbox"/> False

Рисунок 2.6 – Свойство «Connection» компонента ADOQuery

2.5 Алгоритм работы проходной

Блок-схема работы алгоритма программного обеспечения, отвечающего за фиксацию событий и работу проходной показана на рисунке 2.7.

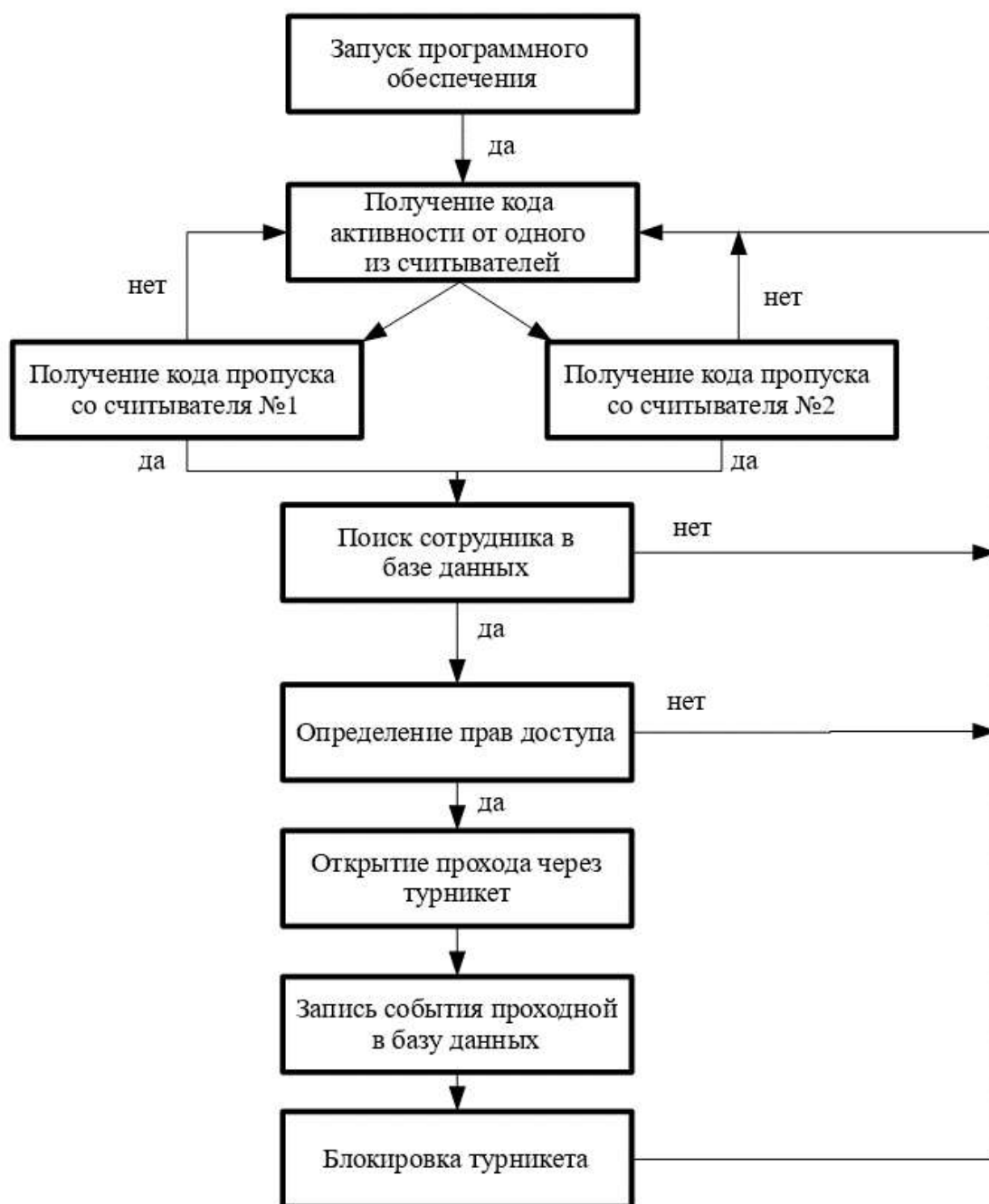


Рисунок 2.7 – Блок-схема алгоритма работы модуля «Мониторинг»

Опишем подробно шаг алгоритма при записи произошедшего события в журнал. В таблицу «EventIO» записывается: дата, время, идентификатор сотрудника, а также тип произошедшего события, чтобы в потом можно было получить отчётность по работе проходной.

В таблице «WorkRangeLogs» при входе сотрудников ведётся запись следующих данных: дата, время, идентификатор сотрудника. При осуществлении сотрудниками выхода сначала находится последняя запись входа сотрудника на предприятие, а потом происходит подсчёт разницы времени выхода и входа с учётом временных интервалов рабочего времени из таблицы «WorkTime», и затем происходит обновление найденной записи в следующем порядке данных: дата, время, сотрудник, проведённое рабочее время.

В таблице «DelaysPersonal» в случае входа сотрудника сначала определяется, а входил ли он уже сегодня? При отрицательном ответе время входа сотрудника сравнивается со временем прихода на работу из таблицы «WorkTime» и при необходимости запишется опоздание в формате: дата, время, сотрудник, время прихода, на сколько опоздал.

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		49

2.6 Главная форма

Главная форма приложения включает в себя основные элементы интерфейса для авторизации пользователя.

При запуске программы показываются поля для авторизации в программном обеспечении. На кнопке «Авторизация» в событии «OnClick» написан код, который проверяет наличие введённых данных авторизации в базе данных, а также обеспечивает проверки на пустое поле и неверные данные:

```
procedure TAuthorizationForms.LogOnClick(Sender: TObject);
begin
//Авторизация через параметры
DataBaseUnit.DataConnectModule.Authorizations.Close;
//Текст запроса в свойстве компонента
DataBaseUnit.DataConnectModule.Authorizations.SQL.Text;
if (Trim(Login.Text) <> ") and (Trim>Password.Text) <> ")
then
begin

DataBaseUnit.DataConnectModule.Authorizations.Parameters.ParamByName('p_login'
).Value:=Login.Text;

DataBaseUnit.DataConnectModule.Authorizations.Parameters.ParamByName('p_passw
').Value:=Password.Text;
try
DataBaseUnit.DataConnectModule.Authorizations.Open;
Except
Application.Title:='Автоматизированный КПП';
ShowMessage('Не удалось открыть таблицу пользователей');
Exit;
end;
if DataBaseUnit.DataConnectModule.Authorizations.RecordCount > 0
then
begin
//Проверка типа учётной записи
if
DataBaseUnit.DataConnectModule.Authorizations.FieldName('AdminAccess').AsBo
olean = True then
begin
//Вход для учётных записей администраторов
Application.Title:='Автоматизированный КПП';
ShowMessage('Вход успешно выполнен');
```

```

AutorizationForms.Visible := False;
Administrator.Visible := True;
Administrator.Show;
Login.Text := "";
Password.Text := "";
end
else
//Вход для операторов
begin
Application.Title:='Автоматизированный КПП';
ShowMessage('Вход успешно выполнен');
AutorizationForms.Visible := False;
OperatorView.Visible := True;
OperatorView.Show;
Login.Text := "";
Password.Text := "";
end;
end
else
begin
//Действия при неверных данных
Application.Title:='Автоматизированный КПП';
ShowMessage('Неверный логин или пароль.Повторите ввод');
Login.Text := "";
Password.Text := "";
AutorizationForms.ActiveControl:=Login;
end;
end
else
begin
//Действия, если хотя бы одно поле пустое
Application.Title:='Автоматизированный КПП';
ShowMessage('Пустые поля логин и пароль не допускаются');
Login.Text := "";
Password.Text := "";
AutorizationForms.ActiveControl:=Login;
end;
end;

```

После удачной авторизации появляется нужна форма для работы в зависимости от типа учётной записи. Так как программное обеспечение включает в себя функционал мониторинга и работы с базой данных и

						Лист
					09.03.01.2017.064.00 ПЗ	51
Изм.	Лист	№ докум.	Подпись	Дата		

представлено одним исполняемым файлом, то форма авторизации в системе общая. Форма авторизации показана на рисунке 2.8.

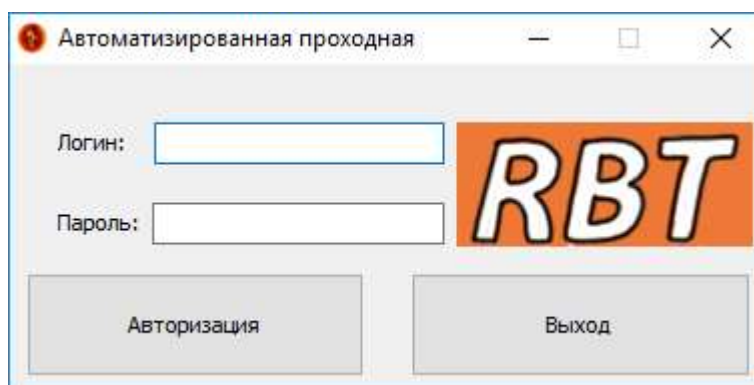


Рисунок 2.8 – Форма авторизации

2.7 Модуль данных формы администрирования

Для работы этой формы программного обеспечения подключаемся к тому же компоненту «DataBasePass:ADODConnection» для соединения с базой данных, так все модули программы между собой связаны, то для подключения новых компонентов достаточно в свойстве «Connection» компонентов ADOQuery выбрать единственный модуль подключения к базе данных.

В этом TDataModule расположены следующие элементы:

- DataBasePass:TADODConnection – модуль для соединения с файлом базы данных;
- QStaff:ADOQuery – обеспечивает доступ к записям о сотрудниках;
- ReportQuery:ADOQuery – обеспечивает доступ к данным для получения отчётов;
- Autorizations:ADOQuery – доступ к данным для авторизации в программном обеспечении;
- QAccountPO:ADOQuery – необходимо для редактирования учётных данных по умолчанию и добавлению новых учётных записей в программное обеспечение;

- ReportDivision:ADOQuery – для редактирования списка подразделений организации;
- ReportQuery:ADOQuery – обеспечивает доступ к данным для получения отчётов;
- QueryTMP:ADOQuery – для обеспечения остального функционала. Текст запросов для доступа к нужному набору данных переписывается в коде программы в зависимости от активной вкладки на форме «Администратор».

2.8 Функционал вкладки для получения отчётов

После того, как база данных накопит достаточное количество статистических данных можно начать получать отчёты по ним.

Для этого у нас есть компонент ReportQuery:TADOQuery, позволяющий с помощью SQL-запросов получать нужные записи из базы данных, удовлетворяющие определённым требованиям. SQL запросы записываются в свойство SQL компонента ReportQuery, которое имеет текстовый тип. Для получения различных отчётов код SQL-запросов будет отличаться.

- Отчёт проходной. Требуется выбрать всех сотрудников, которые отметились на проходной за определённый временной интервал:

```
ReportQuery.SQL.Clear; // Очистка запроса
```

```
ReportQuery.SQL.Add('select DayFixed, TimeFixed, EventFix, Name, NameOtdel  
from EventIO, Staff, Divisions where simplog.today>=:datestart and  
simplog.today<=:dateend');
```

```
ReportQuery.Open;// Выполнить запрос
```

Где datestart и dateend – это начало и конец периода отчёта типа TDateTime.

- Отчёт о рабочем времени. Подсчитывает суммарное рабочее время для конкретного сотрудника, проведённое на работе, либо для всего отдела в целом. Одним SQL-запросом сделать это невозможно. Поэтому далее будет представлен код следующего вида:

```

ReportQuery.SQL.Clear;
Case type of
//Получение списка всех сотрудников
-1:SQL.Add('Select * from Staff);
//Получение списка всех сотрудников определённого отдела
0:SQL.Add('Select * from Staff where idOtdel='+idDivisions);
//Получение для одного сотрудника
1:SQL.Add('Select * from Staff where Name='+Name);
End;
ReportQuery.Open;

```

Где переменная Type – подтип отчёта, idDivisions и Name – переменные содержащие название отдела и имя определённого сотрудника.

Далее необходимо перебрать полученных сотрудников с подсчётом для каждого интервала рабочего времени из таблицы «WorkRangeLogs»:

```

ReportQuery.First;//Установка указателя на первую найденную запись
While not ReportQuery.EOF do// Перебор полученного набора данных до конца
Begin
QueryTMP.SQL.Clear;//Очистка SQL временного компонента данных
QueryTMP.SQL.Add('Select WorkRangeTime from WorkRangeLogs where
idSotr=:idSotr and DayRange>=:DateStart and DayRange<=:Dateend and
WorkRangeTime is not null');
QueryTMP.ParamByName('idSotr').asinteger:=ReportQuery.FieldName
(' idSotrudnika').asinteger;
QueryTMP.Open;
//если обнаружены интервалы рабочего времени
If QueryTMP.RecordCount>0 then
Begin
QueryTMP.First;//Берём первый интервал

```

While not QueryTMP.EOF do// Проходом до конца набора суммируем с
ОСТАЛЬНЫМИ

Begin

Decodetime(fieldbyname('WorkRangeTime').asDatetime, hour,min,sec,msec);

hours:=hours+hour;

mins:=mins+min;

if mins>=60 then

begin hours:=hours+(mins div 60);

mins:=mins mod 60;

end;

QueryTMP.Next;

end;

end;

ReportQuery.Next;//Переход к следующему сотруднику

end;

Таким способом подсчитывается полное рабочее время за определённый
временной интервал.

- Отчёт об отсутствующих. Этот отчёт требует проверки регистрации
прохода сотрудника в журнале проходной в определённый период. SQL-запрос
будет иметь следующий вид:

ReportQuery.Clear;

ReportQuery.SQL.Add('Select NameOtdel, Name from Divisions, Staff where
idSotr not in (Select distinct idSotr from EventIO where today>=:DateStart and
today<=:Dateend)');

ReportQuery.Open;//Выполнить запрос

- Отчёт об опозданиях. Получение списка сотрудников с опозданиями за
определённый срок. Получается SQL-запросом вида:

ReportQuery.Clear;

ReportQuery.SQL.Add('Select NameOtdel, Name, WorkStart, TimeLate from
Divisions, Staff, DelaysPersonal where

										Лист
										55
Изм.	Лист	№ докум.	Подпись	Дата	09.03.01.2017.064.00 ПЗ					

Staff.idSotrudnilka=DelaysPersonal.SotrudnikID and today>=:DateStart and today<=:Dateend');

ReportQuery.Open;//Выполнить запрос

После получения и вывода данных в таблицу на форме можно сразу вывести их в Excel с помощью заранее прописанного в коде шаблона.

					09.03.01.2017.064.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		56

3 Описание программного обеспечения

3.1 Рабочее место оператора

После авторизации в качестве оператора вы попадаете на основную рабочую форму программного обеспечения автоматизированной проходной. В состав функционала программного обеспечения для оператора входит форма «Режим Оператор».

Данная форма выполняет функционал мониторинга с выводом основной информации о человеке, осуществляющем проход на предприятие. Форма мониторинга представлена на рисунке 3.1.

Фамилия	Власенко
Имя	Павел
Отчество	Геннадьевич
Отдел	Бухгалтерия
Должность	Бухгалтер
Номер пропуска	1

Сотрудник	Организация	Отдел	Событие	Время
Власенко	РБТ	Бухгалтерия	Вход	8:30:45
Власенко	РБТ	Бухгалтерия	Выход	18:38:09

Рисунок 3.1 – Рабочее окно режима «Оператор»

В элементы на форме заносится информация о последнем сотруднике, который совершил вход или выход через проходную.

В левой части сверху находится кнопка меню «Операции», которая содержит в себе следующие пункты меню:

- Сменить учётную запись – пункт меню позволяет снова выйти на форму авторизации для смены учётной записи пользователя;
- Выход – которая обеспечивает завершение работы приложения.

Структура пункта меню приведена на рисунке 3.2.

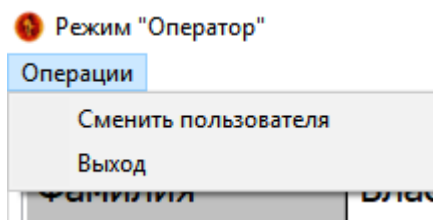


Рисунок 3.2 – Меню «Операции»

Изображение текущего состояния имеет три состояния, которые показывают направление перемещения человека через проходную (стрелки с указанием направления), а также изображение креста, который обозначает, что турникет находится в режиме ожидания прохода через него и попытки неудачного попытки прохода. В качестве дополнительной информации к некоторым типам нарушений выводятся всплывающие сообщения. Пример окна такого события приведён на рисунке 3.3.

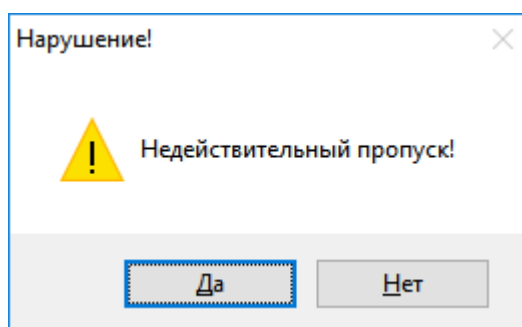


Рисунок 3.3 – Всплывающее сообщение о нарушении

На форме мониторинга находится кнопка отключения турникета и перевода его в разблокированное состояние в случае экстренных ситуаций, а для исключительных ситуаций предполагается использование ручного пульта управления. Также для удобства оператора на форму выведена текущие дата и время. Часть формы с этими элементами показаны на рисунке 3.4.

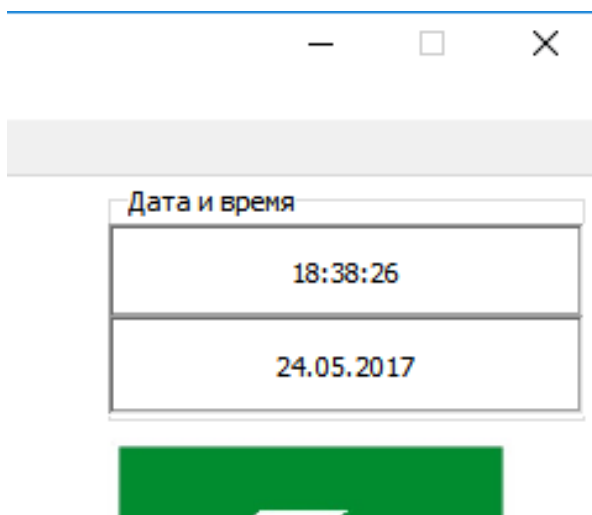


Рисунок 3.4 – Кнопка отключения и блок «Дата и Время»

В нижней части формы присутствует журнал проходной, который может понадобиться оператору для принятия решения в некоторых проблемных ситуациях, которые могут возникнуть в процессе работы проходной. Например, при попытке повторного входа, если человек не выходил с предприятия или осуществляется попытка прохода по чужому пропуску другого человека, когда владелец уже находится на предприятии.

3.2 Рабочая форма администратора

После успешной авторизации под учётной записью администратора сотрудник получает доступ к работе с базой данных проходной. По умолчанию активной оказывается вкладка с редактированием информации о сотрудниках. Форма администратора изображена на рисунке 3.5.

Название отдела	Должность	Фамилия	Имя	Отчество	Начало доступа	Конец доступа	Блокировка
Экономический	Бухгалтер	Власенко	Павел	Геннадьевич	01.02.2005	01.02.2018	False
Бухгалтерия	Бухгалтер	Миллер	Светлана	Алексеевна	25.10.2007	30.06.2017	True
Склад	Кладовщик	Ваханцев	Пётр	Васильевич	01.01.2010	28.05.2017	True
Тех.отдел	Мл. администратор	Клецовская	Анастасия	Егорова	21.05.2017	28.05.2020	False
Экономический	Экономист	Барашков	Ярослав	Валерьевич	21.05.2017	28.05.2020	True
Склад	Старший кладовщик	Лавлинская	Виктория	Александровна	01.05.2017	31.05.2020	False
Тех.отдел	Администратор	Кривовид	Сергей	Андреевич	08.05.2017	30.05.2020	False
Бухгалтерия	Главный бухгалтер	Луговой	Роман	Петрович	08.05.2017	21.05.2017	True
Тех.отдел	Помощник администратора	Буслов	Иван	Романович	01.02.2005	01.02.2018	False
Склад	Водитель погрузчика	Пленчук	Сергей	Андреевич	01.01.2010	21.05.2017	False

Рисунок 3.5 – Рабочее окно администратора

Для начала заполнения информации о сотрудниках изначально надо внести в базу данных список отделов организации. Сделать это можно на вкладке «Информация о подразделениях». Названная вкладка формы администрирования представлена на рисунке 3.6.

Рисунок 3.6 – Вкладка «Информация о подразделениях»

На форме имеется поле для внесения информации о подразделениях организации в таблицу-справочник базы данных. Также имеется поле поиска отделов по названию, которое по мере ввода названия отдела устанавливает указатель в табличном компоненте на нужной записи.

После первоначального заполнения информации о подразделениях организации можно перейти к редактированию, либо заполнению информации о сотрудниках. Вкладка «Редактирование информации о сотрудниках» представлена на рисунке 3.7.

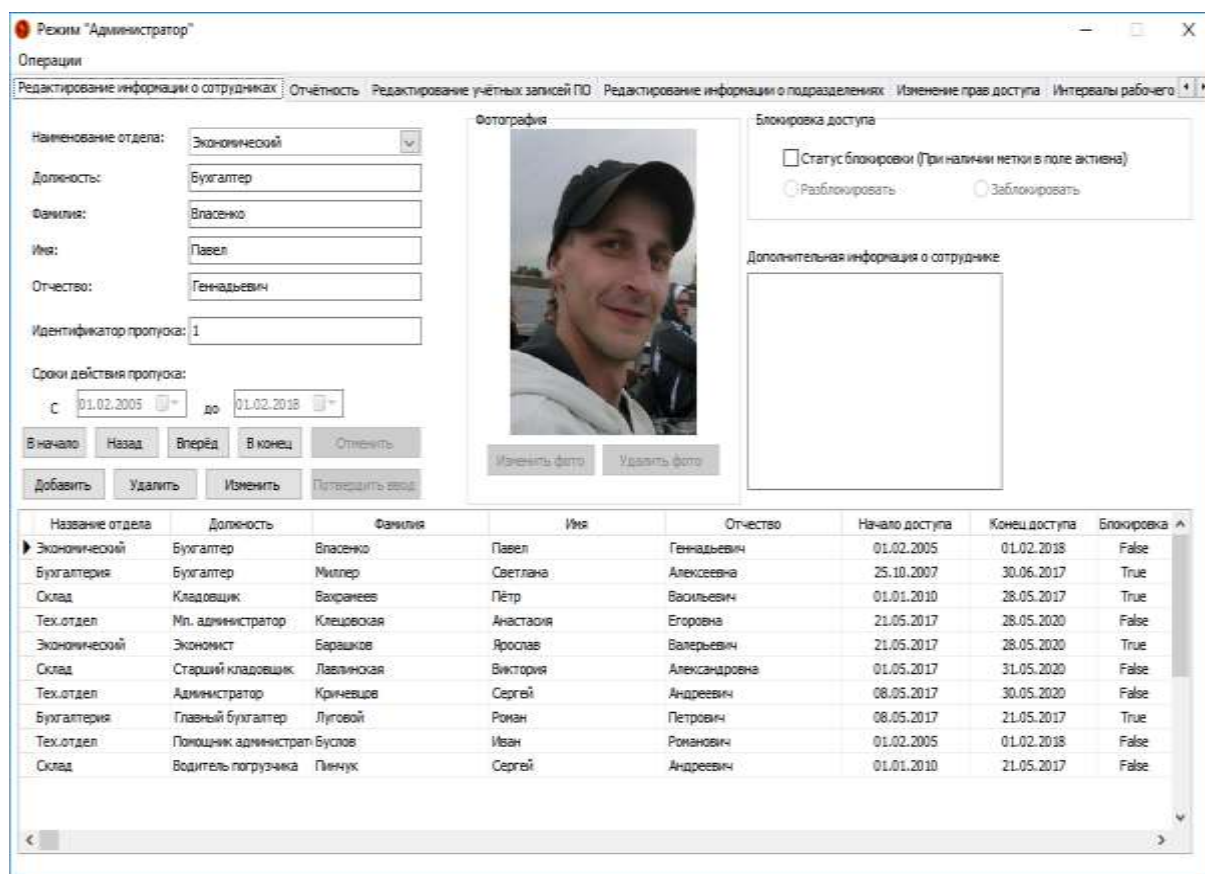


Рисунок 3.7 – Вкладка «Редактирование информации о сотрудниках»

Для того чтобы работать с информацией о сотрудниках на вкладке присутствует набор кнопок. На этой вкладке можно заполнять о сотрудниках следующую информацию: фамилия, имя, отчество, должность, идентификатор пропуска, установить сроки действия пропуска, заблокировать пропуск, внести дополнительную информацию в случае необходимости, осуществить перевод сотрудника из одного подразделения в другое, добавить или обновить фотографию. Изображение кнопок, обеспечивающих этот функционал приведено на рисунке 3.8.

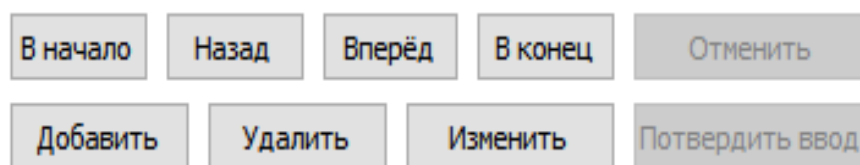


Рисунок 3.8 – Кнопки управления для работы с информацией о сотрудниках

Далее приведено название и описание функционала кнопок:

- «В начало» - кнопка позволяет вернуться к самой первой записи о сотрудниках в базе с текущей на данный момент;
- «Назад» - позволяет перемещаться по записям в направлении от последней записи к первой с шагом в одну запись;
- «Вперёд» - обладает функционалом аналогичным кнопке «Назад», только в противоположном направлении;
- «В конец» - кнопка позволяет переместить указатель на последнюю запись в таблице;
- «Отменить» - кнопка позволяет отменить операцию добавления или редактирования информации;
- «Добавить» - открывает все доступные поля ввода для добавления информации о сотрудниках;
- «Удалить» - кнопка удаляет из базы данных запись о сотруднике на которой в данный момент стоит указатель;
- «Изменить» - открывает для редактирования информацию в таблице, содержащей информацию о персонале;
- «Подтвердить ввод» - кнопка подтверждает внесение изменений в текущий набор данных, а также подтверждает добавление новой записи.

Для добавления или удаления фотографий сотрудников на форме присутствуют две кнопки с которых блокировка снимает на момент внесения изменений или добавления новых записей. Блок работы с фотографиями сотрудников приведён на рисунке 3.9.



Рисунок 3.9 – Блок для работы с фотографиями

Кнопка «Удалить фото» удаляет фотографию у записи в базе данных и из каталога с фотографиями. При нажатии на кнопку «Изменить фото» для последующего добавления или изменения текущей фотографии открывается диалоговое окно, в котором необходимо указать нужный файл с расширением «jrg» или «jpeg», так как для исключения выбора файла с неверным расширением установлен фильтр. Диалоговое окно для выбора файла фотографии изображено на рисунке 3.10.

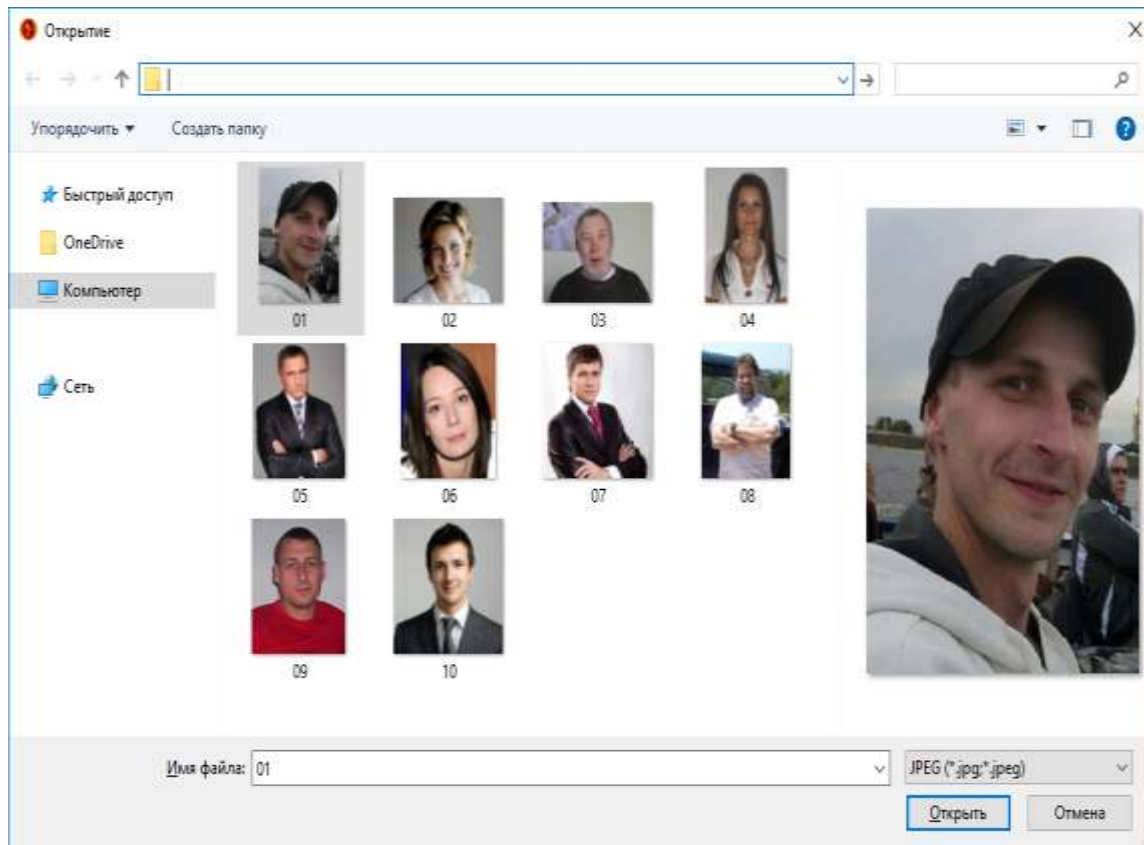


Рисунок 3.10 – Диалоговое окно добавления фотографий

Для получения идентификатора пропуска, необходимо поднести карту, выданную сотруднику к считывателю, когда курсор ввода установлен в поле «Идентификатор пропуска». В случае успешного считывания код пропуска будет занесён в это поле и можно продолжить заполнение информации о сотруднике. Также потому, что поле «Идентификатор пропуска» является доступным к ручному редактированию, то его можно обнулить в случае замены утерянного пропуска или если пропуск забирают при увольнении сотрудника с целью вторичного использования.

Изм.	Лист	№ докум.	Подпись	Дата

09.03.01.2017.064.00 ПЗ

Лист

65

Для установки времени доступа присутствует вкладка «Изменение прав доступа», которая позволяет устанавливать временные интервалы для каждого конкретного сотрудника, либо отдела во время которых сотрудник имеет доступ на работу.

Для установки интервала доступа необходимо из выпадающего списка выбрать необходимый пункт в списке, заполнить поля с началом и окончанием времени доступа, также добавить описание и с помощью панели навигации под таблицей подтвердить добавление или иное действие. Для упрощения выбора нужного варианта в выпадающем меню по мере ввода нужного наименования остаются варианты содержащие набранный набор символов, что существенно уменьшает список при большом количестве записей. Вкладка «Изменение прав доступа» показана на рисунке 3.11.

Режим "Администратор"

Операции

Редактирование информации о сотрудниках | Отчётность | Редактирование учётных записей

Выберите сотрудника или отдел:

Доступ с: до: Описание:

Доступ с	Доступ до	Описание
08:00:00	17:30:00	Рабочее время
17:30:00	20:00:00	Внеурочное время

Панель навигации: < << >> > + - > >> >>>

Рисунок 3.11 – Вкладка «Изменение прав доступа»

В случае, когда сотрудник приходит после окончания времени доступа, то программное обеспечение выдаёт уведомление оператору, что доступ для сотрудника в текущий момент запрещён и турникет остаётся в заблокированном состоянии.

Диапазоны рабочего времени устанавливаются аналогично диапазонам времени доступа, только элементы интерфейса, которые позволяют их устанавливать находятся на вкладке «Интервалы рабочего времени». Вкладка показана на рисунке 3.12.

Режим "Администратор"

Операции

Редактирование информации о сотрудниках Отчётность Редактирование учётных записей ПО

Выберите сотрудника или отдел:

Начинает с: и заканчивает в:

Пояснение к временному диапазону:

Начинает работу	Заканчивает работу	Краткое описание
08:30:00	12:30:00	До обеденное время
13:30:00	17:30:00	Послеобеденное время
18:30:00	20:00:00	Переработка

Рисунок 3.12 – Вкладка «Изменение рабочего времени»

Подсчёт рабочего времени начинает работать после того как установлены временные интервалы рабочего для отдела, либо конкретного сотрудника. Это необходимо для получения отчёта, связанного с рабочим временем, который работает по следующим принципам:

- Рабочее время подсчитывается для сотрудника только в период, когда он присутствовал на рабочем месте в установленных в программе временных диапазонах рабочего времени;
- Подсчёт рабочего времени не ведётся для сотрудника, если он находился на рабочем месте вне всех заданных диапазонов рабочего времени.

Приведём разъяснения на следующем далее примере.

Например, диапазоны рабочего времени установлены с 8:00 до 12:00 и с 13:00 до 17:30.

Сотрудник приходит на работу в 8:45, находится на рабочем месте до 12:30, затем уходит на обеденный перерыв и возвращается в 13:15, а уходит с работы в 18:00. В таком случае программным обеспечением будет зафиксировано следующее: опоздание сотрудником на 45 минут (так как время, к которому необходимо прийти установлено на 8:00), утренние часы работы будут посчитаны, как 12:00 минус 8:45 и равным 3 часа 15 минут (Хотя сотрудник ушел в 12:30, все равно считается, что в обеденное время он не работает), запишет опоздание на 15 минут (приходит к 13:00, опоздал на 15 минут). Послеобеденные часы подсчитает, как 17:00 минус 13:15 и равным 3 часа 45 минут (сотрудник ушел в 18:00, но рабочий день закончился в 17:00, лишний час переработок не учитывается).

Итого, получим в отчете «Рабочее время» за день для сотрудника – 7 часов, а в отчёте «Опоздавшие» - опоздал на 45 минут после 8:00, и опоздал на 15 минут после 13:00.

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		68

Если внутренний распорядок организации такой, что важно только то, сколько времени вообще человек провёл на работе (учитывая все его задержки после окончания рабочего времени на рабочем месте после окончания рабочего дня), то можно задать один интервал рабочего времени (для описанного выше примера будет: с 8:00 до 23:59) и программа подсчитает: опоздание для сотрудника (т.е. приход после 8:00), рабочее время - сколько всего времени за день сотрудник присутствовал на работе.

Программное обеспечение, при поиске в базе данных интервалов времени («Время доступа» или «Рабочее время») для конкретного сотрудника, ищет сначала интервалы непосредственно для конкретного сотрудника, в случае, если сотруднику ничего не приписано, то поиск продолжается для отдела, где работает этот сотрудник. Если вообще не найдены интервалы времени доступа, то сотрудник не сможет пройти турникет.

Если вообще не найдены интервалы рабочего времени, то рабочее время будет подсчитано полностью.

Также на форме администратора присутствует вкладка «Редактирование учётных записей ПО». По умолчанию в базе данных для программного обеспечения автоматизированной проходной хранятся данные для авторизации для двух учётных записей: «Admin» и «User», с паролями аналогичными названиям учётных записей.

Поэтому администратор программного обеспечения после первой авторизации обязан осуществить смену пароля. Для редактирования данных для авторизации присутствует панель навигации. Вкладка «Редактирование учётных записей ПО» изображена на рисунке 3.13.

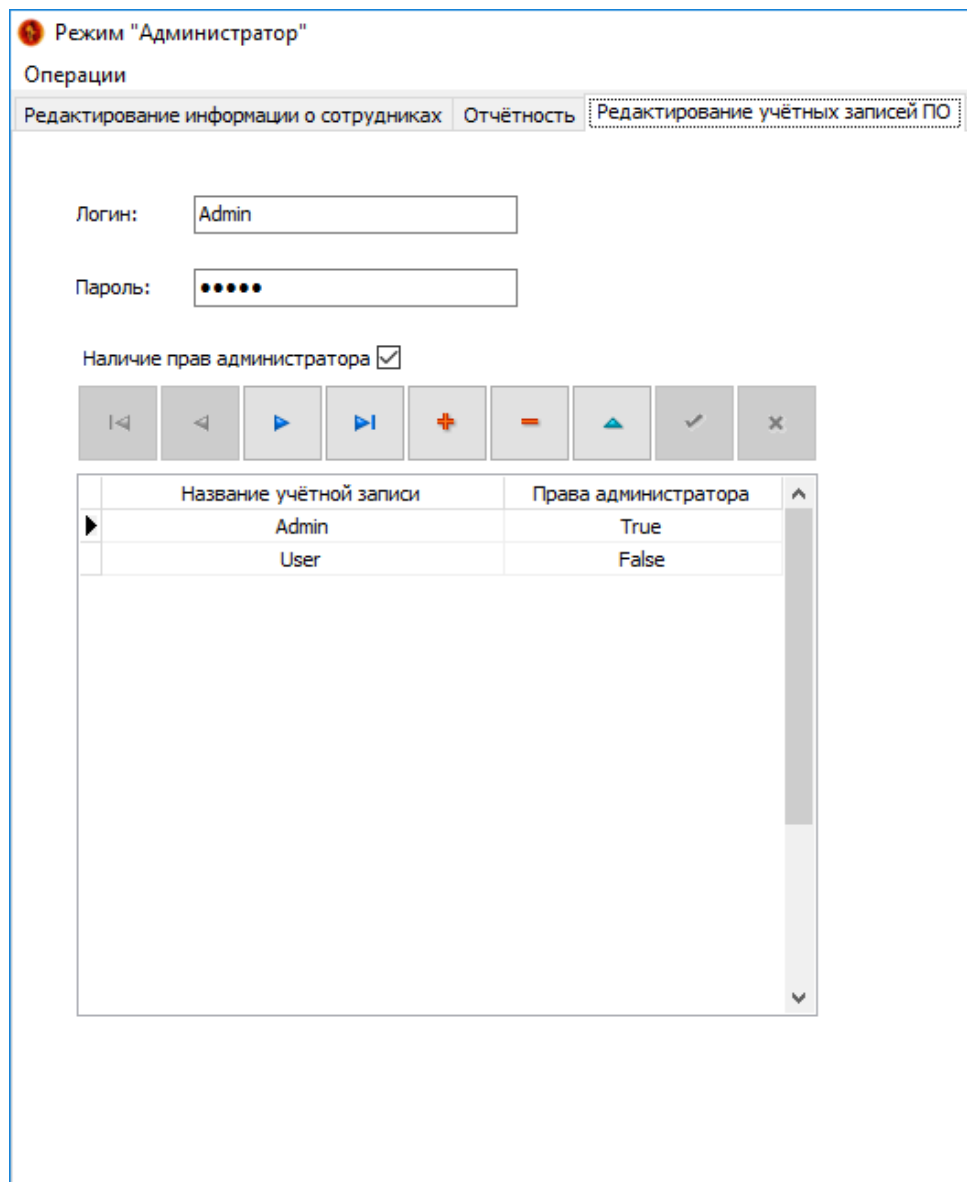


Рисунок 3.13 – Вкладка «Изменение рабочего времени»

Таблица, расположенная под элементами работы с учётными данными пользователей, показывает список всех учётных записей, имеющих доступ к различному функционалу программного обеспечения.

Для получения отчётов в программе предусмотрена вкладка «Отчётность». Вкладка позволяет получить с помощью различных фильтров несколько видов отчётов. Внешний вид её показан на рисунке 3.14.

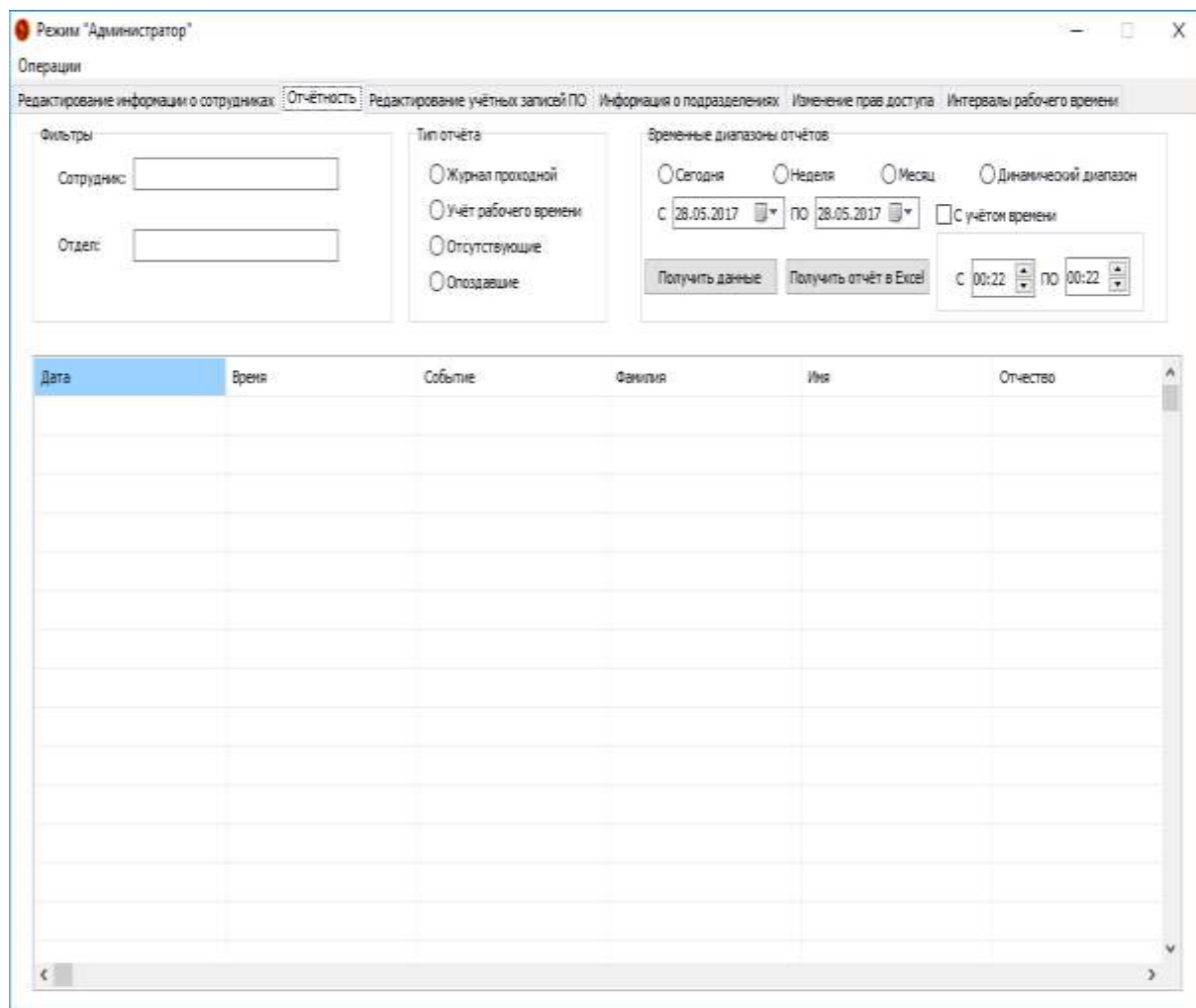


Рисунок 3.14 – Вкладка «Отчётность»

Теперь опишем подробно элементы интерфейса, находящиеся на этой форме.

Группа элементов «Фильтры» позволяет отфильтровать полученный в таблицу набор данных для получения более подробных видов отчётов для конкретных отделов или сотрудников предприятия. Изображение группы элементов подробно приведено на рисунке 3.15.

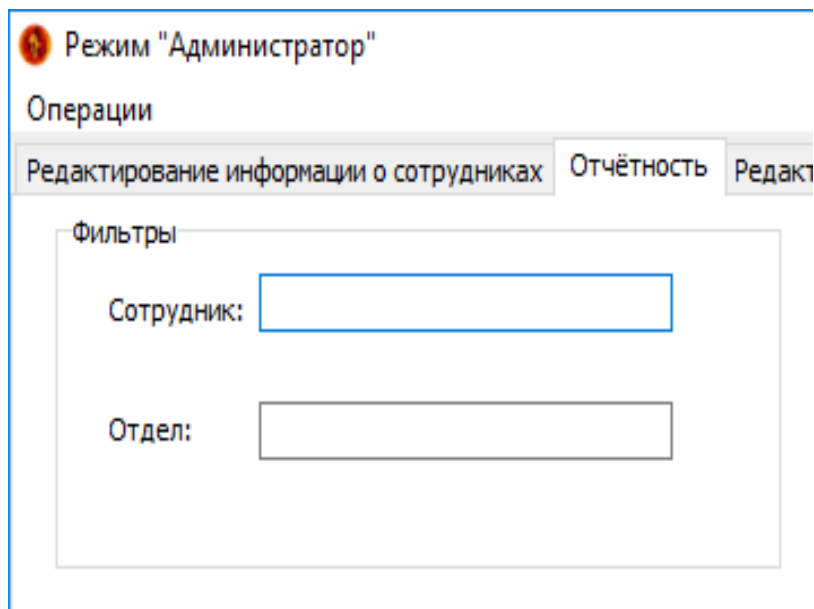


Рисунок 3.15 – Группа фильтров

Получение разных типов отчётов обеспечивает блок «Тип отчёта» с пунктами, имеющими названия, соответствующие содержанию отчёта. Также для получения отчётности по временным диапазонам имеется блок «Временные диапазоны отчётов». Названные блоки приведены на рисунке 3.16.

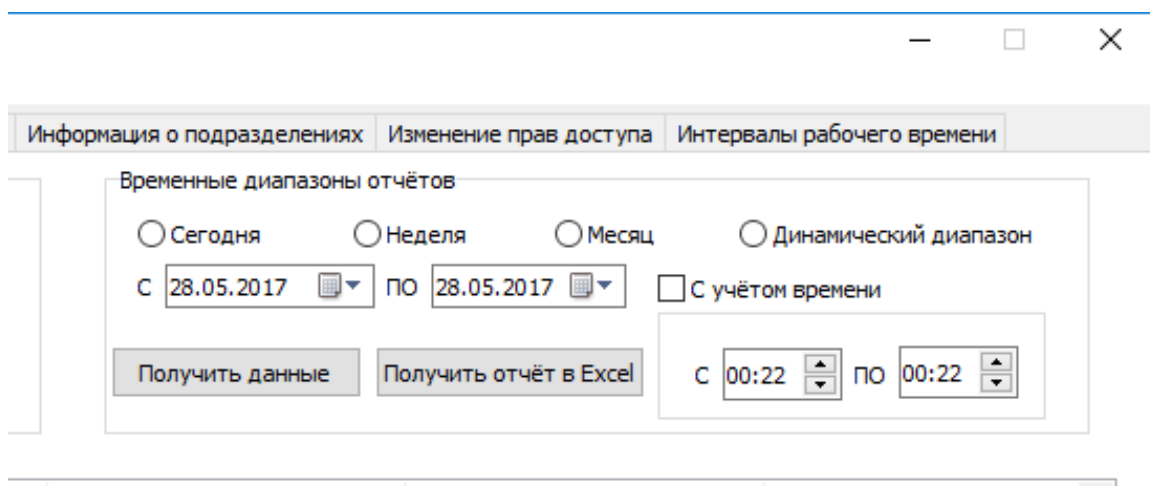


Рисунок 3.16 – Блоки для установки условий для отчётов

Приведённые ранее на рисунке элементы интерфейса позволяют получить четыре основных типа отчётов, которые предполагаются в большинстве случаев использования подобных систем.

Далее приведено описание форматов содержимого отчётов:

- Отчёт проходной.

Содержит информацию о всех проходах сотрудников через проходную и имеет следующую структуру:

Дата | Время | Событие | Ф.И.О. | Отдел | Нарушение;

- Отчёт о рабочем времени сотрудников предприятия.

Показывает суммарное время, проведённое сотрудником на работе за определённый диапазон времени, и имеет следующую структуру:

Отдел | Ф.И.О. | Рабочее время;

- Отчёт об отсутствующих.

Показывает список сотрудников, которые отсутствовали в определённое время на работе. Имеет формат структуры:

Отдел | Ф.И.О.;

- Отчёт об опозданиях.

Выдаёт список сотрудников, которые опоздали на рабочее место в определённые временные диапазоны. Структура отчёта имеет вид:

Отдел | Ф.И.О. | Дата | Время прихода | Опоздание.

Процесс получения разных отчётов отличается лишь выбором нужных пунктов и будет описан далее в общем виде.

В блоке элементов «Тип отчёта» мышкой необходимо выбрать нужный тип отчёта. В блоке «Временные диапазоны» укажите интервал времени: «Сегодня», «Неделя», «Месяц», «Динамический диапазон». Под пунктом «Динамический диапазон» подразумевается произвольное указание начальной и конечно даты для получения отчета, которые задаются в расположенных в блоке полях «С» и «по». Также, если вам необходимо определить не только даты начала и конца срока отчета, а еще и время суток для этих дат, то установив метку в поле с названием «С учётом времени» в запросе будут учтены часовые временные

									Лист
									73
Изм.	Лист	№ докум.	Подпись	Дата	09.03.01.2017.064.00 ПЗ				

диапазоны «с» и «по» - которые должны быть заполнены необходимым вам временем суток. Для получение индивидуальных отчётов необходимо заполнить любое из полей с фильтрами в зависимости от того, для кого вы хотите получить отчёт. Далее нажмите на кнопку «Получить данные». Таблица на форме заполнится необходимым набором данных. После этого надо нажать кнопку «Получить отчёт в Excel», а далее с учётом типа отчёта будет применён необходимый шаблон отчёта, прописанный в коде программы для вывода в Excel.

					09.03.01.2017.064.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		74

ЗАКЛЮЧЕНИЕ

В результате выполнения дипломного проекта было разработано и реализовано программное обеспечение, для комплекса автоматизированной проходной, которая ведёт контроль доступа и учёта рабочего времени сотрудников предприятия. С её помощью можно облегчить труд сотрудников охраны, а также сократить расходы на содержании персонала. Программное обеспечение сокращает объём рутинной работы и допускает меньше ошибок при допуске сотрудников на предприятие.

Установка системы планируется на проходной компании «РБТ», где система будет смонтирована в ближайшем будущем. С внедрением комплекса ожидается повышение уровня дисциплины персонала, так как исключается возможность прийти на рабочее место позже или уйти раньше, чем указано в рабочем расписании. Также если позволяет специфика предприятия можно ввести диапазоны времени для отработки пропущенных часов.

Созданное программное обеспечение имеет перспективы развития. В планах находится разработка сети автоматизированных проходов в различные отделы предприятия, что позволит отслеживать перемещение сотрудников по территории организации.

Внедрение описанного программного обеспечения позволит повысить безопасность путём уменьшения влияния человеческого фактора при контроле за входом на предприятие, а также улучшить соблюдение трудовой дисциплины на предприятии.

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		75

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. СТО ЮУрГУ 04 – 2008: Курсовое и дипломное проектирование: Общие требования к оформлению – Издательство «ЮУрГУ», 2008. – 56 с.
2. Электронный ресурс: Системы контроля доступа и идентификации «Parsec» - <https://www.parsec.ru/> (Internet)
3. Электронный ресурс: Системы безопасности «Perco» - <https://www.perco.ru/> (Internet)
4. Электронный ресурс: Российские современные системы безопасности «КОДОС» - <http://kodos.ru/> (Internet)
5. Архангельский А.Я. Delphi 7 Справочное пособие. – М., Бином-Пресс. - 2004. – 1024с.
6. Электронный ресурс: Справка по Access – <https://support.office.com/ru-ru/article/Создание-базы-данных-в-Access-32a1ea1c-a155-43d6-aa00-f08cd1a8f01e> (Internet)
7. Архангельский А.Я. Программирование в Delphi 7, Бином, 2005. – 1152с.
8. Вигерс К. Разработка требований к программному обеспечению. /Пер, с англ. – М.: Издательско-торговый дом «Русская Редакция», 2004. - 576 с.
9. Глушаков С.В., Клевцов А.Л., Программирование в среде Delphi 7.0, Фолио 2003. – 520с.
10. Леонтьев Б.К., Мультимедия Microsoft Windows без страха, Новый издательский дом, 2005. – 173с.
11. Мандел Т. Дизайн интерфейсов, ДМК, 2005. – 410с.
12. Могилев А.В., Листрова Л.В., Методы программирования. Компьютерные вычисления, СПб.: БВХ-Петербург, 2008. – 320с.
13. Музыченко Е.В., Фролов И.Б., Мультимедия для Windows, 2003. – 192с.
14. Пайс А., Гении науки. – М.: Институт компьютерных исследований, 2002. – 448с.
15. Архангельский А.А., Программирование в Delphi. – М.: Бином, 2003. – 1231 с.

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		76

16. Техническая документация лингафонного класса «Lingvo 4.0»
17. Аппак М.А., Автоматизированные рабочие места на основе персональных ЭВМ, М.:Радио и связь, 1989. – 176с.
18. Барановская Т.П. и др. Архитектура компьютерных систем и сетей. Учеб. пособие. - М.: Финансы и статистика, 2003. – 256с.
19. Благодатских В.А. и др. Стандартизация разработки программных средств: Учеб. пособие / Благодатских В. А., Волнин В. А., Посакалов К. Ф.; Под ред. Разумова О. С. – М.: Финансы и статистика, 2003. – 288с.

					09.03.01.2017.064.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		77

ПРИЛОЖЕНИЕ А. Исходный код программы

Листинг файла KPPControlPass.pas

```
unit KPPControlPass;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
  System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.StdCtrls,
  Vcl.Imaging.pngimage, Jpeg, DateUtils;

type
  TAuthorizationForms = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Login: TEdit;
    Password: TEdit;
    Image1: TImage;
    LogOn: TButton;
    LogOff: TButton;
    procedure FormCreate(Sender: TObject);
    procedure LogOnClick(Sender: TObject);
    procedure LoginKeyPress(Sender: TObject; var Key: Char);
    procedure PasswordKeyPress(Sender: TObject; var Key: Char);
    procedure LogOffClick(Sender: TObject);
    procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
  private
    { Private declarations }
  public
    { Public declarations }
  //Для запрета перемещения формы за границы экрана
  procedure WMMoving(var Msg: TWMMoving); message WM_MOVING;
  end;

var
  AuthorizationForms: TAuthorizationForms;

implementation
  //Обеспечение связи между формами и модулями
  uses DataBaseUnit, AdminUnit, ConnectN, WorkUnit;
  {$R *.dfm}

  procedure TAuthorizationForms.LogOnClick(Sender: TObject);
```

									Лист
									78
Изм.	Лист	№ докум.	Подпись	Дата	09.03.01.2017.064.00 ПЗ				

```

begin
//Авторизация через параметры
DataBaseUnit.DataConnectModule.Autorizations.Close;
//Текст запроса в свойстве компонента
DataBaseUnit.DataConnectModule.Autorizations.SQL.Text;
if (Trim(Login.Text) <> ") and (Trim>Password.Text) <> ")
then
begin

DataBaseUnit.DataConnectModule.Autorizations.Parameters.ParamByName('p_login').Value:=Login.Text;

DataBaseUnit.DataConnectModule.Autorizations.Parameters.ParamByName('p_passw').Value:=Password.Text;
try
DataBaseUnit.DataConnectModule.Autorizations.Open;
Except
Application.Title:='Автоматизированный КПП';
ShowMessage('Не удалось открыть таблицу пользователей');
Exit;
end;
if DataBaseUnit.DataConnectModule.Autorizations.RecordCount > 0
then
begin
//Проверка типа учётной записи
if DataBaseUnit.DataConnectModule.Autorizations.FieldByName('AdminAccess').AsBoolean =
True then
begin
//Вход для учётных записей администраторов
Application.Title:='Автоматизированный КПП';
ShowMessage('Вход успешно выполнен');
AuthorizationForms.Visible := False;
Administrator.Visible := True;
Administrator.Show;
Login.Text := ";
Password.Text := ";
end
else
//Вход для операторов
begin
Application.Title:='Автоматизированный КПП';
ShowMessage('Вход успешно выполнен');
AuthorizationForms.Visible := False;
OperatorView.Visible := True;
OperatorView.Show;
Login.Text := ";

```

						Лист
					09.03.01.2017.064.00 ПЗ	79
Изм.	Лист	№ докум.	Подпись	Дата		

```

    Password.Text := ";
end;
end
else
begin
    //Действия при неверных данных
    Application.Title:='Автоматизированный КПП';
    ShowMessage('Неверный логин или пароль.Повторите ввод');
    Login.Text := ";
    Password.Text := ";
    AuthorizationForms.ActiveControl:=Login;
end;
end
else
begin
    //Действия, если хотя бы одно поле пустое
    Application.Title:='Автоматизированный КПП';
    ShowMessage('Пустые поля логин и пароль не допускаются');
    Login.Text := ";
    Password.Text := ";
    AuthorizationForms.ActiveControl:=Login;
end;
end;

procedure TAuthorizationForms.LogOffClick(Sender: TObject);
{Предохранитель на закрытие программы}
begin
If messageBox(Handle,'Вы уверены, что хотите закрыть программу?','Выход',
mb_YesNo or MB_ICONWARNING+MB_DEFBUTTON1)=idYes then
Application.Terminate
else AuthorizationForms.Show;
end;

procedure TAuthorizationForms.FormCloseQuery(Sender: TObject;
var CanClose: Boolean);
begin
{Предохранитель закрытия программы на стандартной кнопке}
If messageBox(Handle,'Вы уверены, что хотите закрыть программу?','Выход',
mb_YesNo or MB_ICONWARNING+MB_DEFBUTTON1)=idYes then
CanClose:=True else
CanClose:=False;
end;

procedure TAuthorizationForms.FormCreate(Sender: TObject);
{Функция запрещающая многократный запуск программы}
Var ExtEndedStyle:dword;

```

```

Wnd:hWnd;
buff:Array[0.. 127] of Char;
Begin
Wnd:= GetWindow(Handle, gw_HWndFirst);
While Wnd <> 0
DO
  Begin
  If (Wnd <> Application.Handle)
  and (GetWindow(Wnd, gw_Owner) = 0) Then
    Begin GetWindowText (Wnd, buff, sizeof (buff));
    If StrPas (buff) = Application.Title Then
      Begin Halt;
      End;
    End;
  Wnd:= GetWindow (Wnd, gw_hWndNext);
  End;
End;

procedure TAuthorizationForms.LoginKeyPress(Sender: TObject; var Key: Char);
begin
//Перевод фокуса на следующее поле
if Key = #13 then Password.SetFocus;
end;

procedure TAuthorizationForms.PasswordKeyPress(Sender: TObject; var Key: Char);
begin
//Активация кнопки авторизации
if Key = #13 then LogOn.Click;
end;

procedure TAuthorizationForms.WMMoving(var Msg: TWMMoving);
{Процедура делающая невозможным перемещение формы за пределы экрана}
var
workArea: TRect;
begin
workArea := Screen.WorkareaRect;
with Msg.DragRect^ do
begin
if Left < workArea.Left then
OffsetRect(Msg.DragRect^, workArea.Left - Left, 0) ;
if Top < workArea.Top then
OffsetRect(Msg.DragRect^, 0, workArea.Top - Top) ;
if Right > workArea.Right then
OffsetRect(Msg.DragRect^, workArea.Right - Right, 0) ;
if Bottom > workArea.Bottom then
OffsetRect(Msg.DragRect^, 0, workArea.Bottom - Bottom) ;

```

```
end;  
inherited;  
end;  
end.
```

Листинг файла AdminUnit.pas

```
unit AdminUnit;  
  
interface  
  
uses  
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,  
  Vcl.Graphics,  
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.Grids, Vcl.DBGrids,  
  Vcl.ExtCtrls, Vcl.DBCtrls, Vcl.StdCtrls, Vcl.Mask, Vcl.Menus, Vcl.ComCtrls,  
  Jpeg, DateUtils, Vcl.ExtDlgs;  
  
type  
  TAdministrator = class(TForm)  
    PageControl1: TPageControl;  
    TabSheet1: TTabSheet;  
    TabSheet2: TTabSheet;  
    TabSheet3: TTabSheet;  
    TabSheet4: TTabSheet;  
    TabSheet5: TTabSheet;  
    MainMenu1: TMainMenu;  
    MenuFILE: TMenuItem;  
    DBEdit2: TDBEdit;  
    DBEdit1: TDBEdit;  
    Label1: TLabel;  
    Label2: TLabel;  
    AdminCheck: TDBCheckBox;  
    DBNavigator1: TDBNavigator;  
    DBGrid1: TDBGrid;  
    GroupBox1: TGroupBox;  
    GroupBox2: TGroupBox;  
    RadioButton1: TRadioButton;  
    RadioButton2: TRadioButton;  
    RadioButton3: TRadioButton;  
    RadioButton4: TRadioButton;  
    RadioButton5: TRadioButton;  
    RadioButton6: TRadioButton;  
    RadioButton7: TRadioButton;  
    RadioButton8: TRadioButton;  
    Label3: TLabel;  
    DateTimePicker1: TDateTimePicker;  
    Label4: TLabel;
```

						Лист
						82
Изм.	Лист	№ докум.	Подпись	Дата	09.03.01.2017.064.00 ПЗ	

DateTimePicker2: TDateTimePicker;
 CheckBox1: TCheckBox;
 Button1: TButton;
 DateTimePicker3: TDateTimePicker;
 DateTimePicker4: TDateTimePicker;
 Label5: TLabel;
 Label6: TLabel;
 GroupBox3: TGroupBox;
 DBGrid3: TDBGrid;
 Label7: TLabel;
 DBEdit3: TDBEdit;
 DBNavigator2: TDBNavigator;
 GroupBox4: TGroupBox;
 LabeledEdit1: TLabeledEdit;
 DBLookupComboBox1: TDBLookupComboBox;
 Label8: TLabel;
 DBEdit4: TDBEdit;
 DBEdit5: TDBEdit;
 DBEdit6: TDBEdit;
 DBEdit7: TDBEdit;
 DateTimePicker5: TDateTimePicker;
 DateTimePicker6: TDateTimePicker;
 Label9: TLabel;
 Label10: TLabel;
 Label11: TLabel;
 Label12: TLabel;
 Label13: TLabel;
 DBEdit8: TDBEdit;
 Label14: TLabel;
 Label15: TLabel;
 Label16: TLabel;
 DBEdit9: TDBEdit;
 DBEdit10: TDBEdit;
 Image1: TImage;
 Button2: TButton;
 OpenDialog1: TOpenDialog;
 Button3: TButton;
 DBEdit11: TDBEdit;
 GroupBox5: TGroupBox;
 GroupBox6: TGroupBox;
 DBMemo1: TDBMemo;
 Label17: TLabel;
 RadioButton9: TRadioButton;
 RadioButton10: TRadioButton;
 DBGrid4: TDBGrid;
 DBCheckBox2: TDBCheckBox;

						Лист
					09.03.01.2017.064.00 ПЗ	83
Изм.	Лист	№ докум.	Подпись	Дата		

Button4: TButton;
 Button5: TButton;
 Button6: TButton;
 Button7: TButton;
 Button8: TButton;
 Button9: TButton;
 Button10: TButton;
 Button11: TButton;
 Button12: TButton;
 DBEdit12: TDBEdit;
 N1: TMenuItem;
 TabSheet6: TTabSheet;
 N2: TMenuItem;
 ComboBox1: TComboBox;
 Label18: TLabel;
 Label19: TLabel;
 Label20: TLabel;
 Label21: TLabel;
 MaskEdit1: TMaskEdit;
 MaskEdit2: TMaskEdit;
 StringGrid1: TStringGrid;
 DBNavigator3: TDBNavigator;
 Edit1: TEdit;
 ComboBox2: TComboBox;
 Label22: TLabel;
 Edit2: TEdit;
 Label23: TLabel;
 MaskEdit3: TMaskEdit;
 Label24: TLabel;
 MaskEdit4: TMaskEdit;
 Label25: TLabel;
 StringGrid2: TStringGrid;
 DBNavigator4: TDBNavigator;
 GroupBox7: TGroupBox;
 Label26: TLabel;
 Label27: TLabel;
 Edit3: TEdit;
 Edit4: TEdit;
 StringGrid3: TStringGrid;
 Button13: TButton;
 procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
 procedure RadioButton5Click(Sender: TObject);
 procedure RadioButton6Click(Sender: TObject);
 procedure RadioButton6Enter(Sender: TObject);
 procedure RadioButton5Enter(Sender: TObject);
 procedure RadioButton7Click(Sender: TObject);

					09.03.01.2017.064.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		84

```

procedure FormCreate(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure DateTimePicker6Change(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure DateTimePicker5Change(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure RadioButton10Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure Button12Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure DBGrid4ColumnMoved(Sender: TObject; FromIndex, ToIndex: Integer);
procedure RadioButton9DbClick(Sender: TObject);
procedure RadioButton9Enter(Sender: TObject);
procedure RadioButton10Enter(Sender: TObject);
procedure RadioButton7Enter(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }

protected
  //Процедура для панели задач
  procedure CreateParams(var Params: TCreateParams);
  override;
  end;

var
  Administrator: TAdministrator;
  fullpath:string;//Переменная для изображений
  FormatSettings: TFormatSettings; //Для краткого формата даты
implementation
  //Обеспечение связи между формами и модулями
  uses DataBaseUnit, ConnectN, KPPControlPass, WorkUnit;
  {$R *.dfm}

  procedure TAdministrator.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
  {Предохранитель закрытия программы на стандартной кнопке}
  begin
  If messageBox(Handle,'Вы уверены, что хотите закрыть программу?','Выход',

```

						<i>Лист</i>
					09.03.01.2017.064.00 ПЗ	85
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

```

mb_YesNo or MB_ICONWARNING+MB_DEFBUTTON1)=idYes then
begin
  CanClose:=True;
  Application.Terminate;
end
else
CanClose:=False;
end;

```

```

procedure TAdministrator.FormCreate(Sender: TObject);
begin
  //Текущая дата при старте ПО
  DateTimePicker1.Date:=Now;
  DateTimePicker2.Date:=Now;
  //Лимиты даты для datetimerpicker
  DateTimePicker1.MaxDate:=Date;
  DateTimePicker2.MaxDate:=Date;
  DateTimePicker1.MinDate:=IncYear(Date,-20);
  DateTimePicker2.MinDate:=IncYear(Date,-20);
  DateTimePicker5.MaxDate:=Date;
  DateTimePicker6.MaxDate:=IncYear(Date,20);
  DateTimePicker5.MinDate:=IncYear(Date,-20);
  DateTimePicker6.MinDate:=IncYear(Date,-20);

end;

```

```

procedure TAdministrator.FormShow(Sender: TObject);
begin
  //Подгрузка первой фотографии при показе формы
  fullpath:= Pchar(ExtractFilePath(Application.ExeName))+PhotoSot\'+DBEdit11.txt;
  if fileexists(fullpath)=True then
begin
Image1.picture.loadfromFile(fullpath);
end
else
begin

Image1.picture.loadfromFile(PChar(ExtractFilePath(Application.ExeName))+PhotoSot\NoFoto.jpg');
end;

  //Установка даты первой записи в datetimerpicker
  FormatSettings.ShortDateFormat:='dd.MM.yyyy';
  FormatSettings.DateSeparator:='.';
  if (DBEdit9.Text<>") and (DBEdit10.Text<>") then
begin

```

					<i>Лист</i>
					09.03.01.2017.064.00 ПЗ
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	86

```

DateTimePicker5.Date:= StrToDate(DBEdit9.Text, FormatSettings);
DateTimePicker6.Date:= StrToDate(DBEdit10.Text, FormatSettings);
end
else
begin
DBEdit9.Text:=DateToStr(Date);
DBEdit10.Text:=DateToStr(Date);
end;
end;

procedure TAdministrator.RadioButton10Click(Sender: TObject);
begin
DataBaseUnit.DataConnectModule.QStaff.FieldByName('BlockAccess').AsVariant:=True;
end;

procedure TAdministrator.RadioButton10Enter(Sender: TObject);
begin
DataBaseUnit.DataConnectModule.QStaff.FieldByName('BlockAccess').AsVariant:=True;
end;

procedure TAdministrator.RadioButton5Click(Sender: TObject);
begin
//Установка даты на текущую для datetimepicker
DateTimePicker1.Date:=Date;
DateTimePicker2.Date:=Date;
end;

procedure TAdministrator.RadioButton5Enter(Sender: TObject);
begin
//Установка даты на текущую для datetimepicker
DateTimePicker1.Date:=Date;
DateTimePicker2.Date:=Date;
end;

procedure TAdministrator.RadioButton6Click(Sender: TObject);
begin
//Установка недельного диапазона для фильтра
DateTimePicker2.Date:=IncDay(DateTimePicker1.Date,7)
end;

procedure TAdministrator.RadioButton6Enter(Sender: TObject);
begin
//Установка недельного диапазона для фильтра
DateTimePicker2.Date:=IncDay(DateTimePicker1.Date,7);
end;

```

					<i>Лист</i>
					09.03.01.2017.064.00 ПЗ
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	87

```

procedure TAdministrator.RadioButton7Click(Sender: TObject);
begin
//Установка месячного диапазона для фильтра
DateTimePicker2.Date:=IncMonth(DateTimePicker1.Date, 1);
end;

procedure TAdministrator.RadioButton7Enter(Sender: TObject);
begin
//Установка месячного диапазона для фильтра
DateTimePicker2.Date:=IncMonth(DateTimePicker1.Date, 1);
end;

procedure TAdministrator.RadioButton9DbClick(Sender: TObject);
begin
DataBaseUnit.DataConnectModule.QStaff.FieldByName('BlockAccess').AsVariant:=False;
end;

procedure TAdministrator.RadioButton9Enter(Sender: TObject);
begin
DataBaseUnit.DataConnectModule.QStaff.FieldByName('BlockAccess').AsVariant:=False;
end;

procedure TAdministrator.Button10Click(Sender: TObject);
begin
//Добавить
DataConnectModule.QStaff.Append;
DBEdit11.Text:='NoFoto.jpg';
//Защита от пустых полей для конвертации формата даты
if (DBEdit9.Text='') and (DBEdit10.Text='') then
begin
FormatSettings.ShortDateFormat:='dd.MM.yyyy';
FormatSettings.DateSeparator:='.';
DBEdit9.Text:=DateToStr(DateTimePicker5.Date, FormatSettings);
DBEdit10.Text:=DateToStr(DateTimePicker6.Date, FormatSettings);
end;
//Подгрузка фото
fullpath:= Pchar(ExtractFilePath(Application.ExeName))+'\PhotoSot'+DBEdit11.text;
if fileexists(fullpath)=True then
begin
Image1.picture.loadfromFile(fullpath);
end
else
begin
Image1.picture.loadfromFile(PChar(ExtractFilePath(Application.ExeName))+'\PhotoSot\NoFoto.jpg');
end;
end;

```

					<i>Лист</i>
					88
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	09.03.01.2017.064.00 ПЗ

```

end;

Button4.Enabled:=False;
Button5.Enabled:=False;
Button6.Enabled:=False;
Button7.Enabled:=False;
Button11.Enabled:=False;
Button8.Enabled:=False;

Button9.Enabled:=True;
Button12.Enabled:=True;

DBLookupComboBox1.ReadOnly:=False;
DBEdit4.ReadOnly:=False;
DBEdit5.ReadOnly:=False;
DBEdit6.ReadOnly:=False;
DBEdit7.ReadOnly:=False;
DBEdit8.ReadOnly:=False;

Button2.Enabled:=True;
Button3.Enabled:=True;

DateTimePicker5.Enabled:=True;
DateTimePicker6.Enabled:=True;

DBMemo1.ReadOnly:=False;

RadioButton9.Enabled:=True;
RadioButton10.Enabled:=True;
end;

procedure TAdministrator.Button11Click(Sender: TObject);
begin
//Удалить
if DBEdit12.Text<>" then
begin
if (MessageDlg('Вы действительно хотите удалить запись?', mtConfirmation, [mbYes,
mbNo], 0) = mrYes) then
begin
DataConnectModule.QStaff.Delete;
end
end
else
begin
Showmessage('FUCK YOU!!!');
DataConnectModule.QStaff.Cancel;

```

					09.03.01.2017.064.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		89

```

    end;
end;

procedure TAdministrator.Button12Click(Sender: TObject);
begin
//ОТМЕНА
DataConnectModule.QStaff.Cancel;
    Button2.Enabled:=False;
    Button3.Enabled:=False;
    Button4.Enabled:=True;
    Button5.Enabled:=True;
    Button6.Enabled:=True;
    Button7.Enabled:=True;
    Button11.Enabled:=True;
    Button8.Enabled:=True;
    Button10.Enabled:=True;
    Button12.Enabled:=False;
    Button9.Enabled:=False;
    RadioButton9.Enabled:=False;
    RadioButton10.Enabled:=False;
    DBEdit4.ReadOnly:=True;
    DBEdit5.ReadOnly:=True;
    DBEdit6.ReadOnly:=True;
    DBEdit7.ReadOnly:=True;
    DBEdit8.ReadOnly:=True;
    DBMemo1.ReadOnly:=True;
    DateTimePicker5.Enabled:=False;
    DateTimePicker6.Enabled:=False;
end;

procedure TAdministrator.Button2Click(Sender: TObject);
//Загрузка фото для добавления
begin
try
if OpenFileDialog1.Execute then
begin
DataConnectModule.QStaff.FieldName('PhotoS').AsVariant:=ExtractFileName(OpenDialog1.FileName);
CopyFile(Pchar(OpenDialog1.FileName),Pchar(ExtractFilePath(Application.ExeName)+'\PhotoSot'+ExtractFileName(OpenDialog1.FileName)),true);
Image1.picture.loadfromFile(PChar(ExtractFilePath(Application.ExeName))+'\PhotoSot'+DBEdit11.text);
end;
except
on e:Exception do
end;
end;

```

					<i>Лист</i>
					09.03.01.2017.064.00 ПЗ
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	90

```

end;

procedure TAdministrator.Button3Click(Sender: TObject);
begin
//Удаление фото
if DataConnectModule.QStaff.FieldByName('PhotoS').Text<>'NoFoto.jpg' then
begin
DeleteFile(PChar(ExtractFilePath(Application.ExeName))+PhotoSot\'+DBEdit11.Text);
DBEdit11.Text:='NoFoto.jpg';
end
else
begin
DBEdit11.Text:='NoFoto.jpg';
end;
end;

procedure TAdministrator.Button4Click(Sender: TObject);
var i:integer;
begin
//Кнопка вперёд

if not DataConnectModule.QStaff.EOF then
begin
DataConnectModule.QStaff.Next;
Button4.Enabled:=True;
Button7.Enabled:=True;
Button5.Enabled:=True;
Button6.Enabled:=True;
end
else
begin
Button4.Enabled:=False ;
Button7.Enabled:=False;
Button5.Enabled:=True;
Button6.Enabled:=True;
end;

//Защита от пустых полей для конвертации формата даты
if (DBEdit9.EditText<>'') and (DBEdit10.Text<>'') then
begin
FormatSettings.ShortDateFormat:='dd.MM.yyyy';
FormatSettings.DateSeparator:='.';
DateTimePicker5.Date:= StrToDate(DBEdit9.Text, FormatSettings);
DateTimePicker6.Date:= StrToDate(DBEdit10.Text, FormatSettings);
end;

//Подгрузка фото

```

					<i>Лист</i>
					09.03.01.2017.064.00 ПЗ
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	91


```

fullpath:= Pchar(ExtractFilePath(Application.ExeName))+'PhotoSot\'+DBEdit11.text;
if fileexists(fullpath)=True then
begin
Image1.picture.loadfromFile(fullpath);
end
else
begin

Image1.picture.loadfromFile(PChar(ExtractFilePath(Application.ExeName))+'PhotoSot\NoFoto.jpg');
end;

end;

procedure TAdministrator.Button5Click(Sender: TObject);
begin
//Кнопка назад
if not DataConnectModule.QStaff.BOF then
begin
DataConnectModule.QStaff.Prior;
Button4.Enabled:=True;
Button7.Enabled:=True;
Button5.Enabled:=True;
Button6.Enabled:=True;
end
else
begin
Button4.Enabled:=True;
Button7.Enabled:=True;
Button5.Enabled:=False;
Button6.Enabled:=False;
end;

FormatSettings.ShortDateFormat:='dd.MM.yyyy';
FormatSettings.DateSeparator:='.';
DateTimePicker5.Date:= StrToDate(DBEdit9.Text, FormatSettings);
DateTimePicker6.Date:= StrToDate(DBEdit10.Text, FormatSettings);
//Защита от пустых полей для конвертации формата даты
if (DBEdit9.EditText<>") and (DBEdit10.Text<>") then
begin
DateTimePicker5.Date:= StrToDate(DBEdit9.Text, FormatSettings);
DateTimePicker6.Date:= StrToDate(DBEdit10.Text, FormatSettings);
end;
//Подгрузка фото
fullpath:= Pchar(ExtractFilePath(Application.ExeName))+'PhotoSot\'+DBEdit11.text;
if fileexists(fullpath)=True then

```

						Лист
					09.03.01.2017.064.00 ПЗ	92
Изм.	Лист	№ докум.	Подпись	Дата		

```

begin
Image1.picture.loadfromFile(fullpath);
end
else
begin

Image1.picture.loadfromFile(PChar(ExtractFilePath(Application.ExeName))+'PhotoSot\NoF
oto.jpg');
end;
end;

procedure TAdministrator.Button6Click(Sender: TObject);
begin
//Возврат к первой записи
DataConnectModule.QStaff.First;
Button5.Enabled:=False;
Button6.Enabled:=False;
Button4.Enabled:=True;
Button7.Enabled:=True;

//Защита от пустых полей для конвертации формата даты
if (DBEdit9.EditText<>") and (DBEdit10.Text<>") then
begin
FormatSettings.ShortDateFormat:='dd.MM.yyyy';
FormatSettings.DateSeparator:='.';
DateTimePicker5.Date:= StrToDate(DBEdit9.Text, FormatSettings);
DateTimePicker6.Date:= StrToDate(DBEdit10.Text, FormatSettings);
end;
//Подгрузка фото
fullpath:= Pchar(ExtractFilePath(Application.ExeName))+'PhotoSot\''+DBEdit11.text;
if fileexists(fullpath)=True then
begin
Image1.picture.loadfromFile(fullpath);
end
else
begin

Image1.picture.loadfromFile(PChar(ExtractFilePath(Application.ExeName))+'PhotoSot\NoF
oto.jpg');
end;

end;

procedure TAdministrator.Button7Click(Sender: TObject);
begin

```

					<i>Лист</i>
					09.03.01.2017.064.00 ПЗ
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	93

```

//К последней записи
DataConnectModule.QStaff.Last;
Button4.Enabled:=False;
Button7.Enabled:=False;
Button5.Enabled:=True;
Button6.Enabled:=True;

//Защита от пустых полей для конвертации формата даты
if (DBEdit9.EditText<>") and (DBEdit10.Text<>") then
begin
FormatSettings.ShortDateFormat:='dd.MM.yyyy';
FormatSettings.DateSeparator:='.';
DateTimePicker5.Date:= StrToDate(DBEdit9.Text, FormatSettings);
DateTimePicker6.Date:= StrToDate(DBEdit10.Text, FormatSettings);
end;
//Подгрузка фото
fullpath:= Pchar(ExtractFilePath(Application.ExeName))+'\PhotoSot'+DBEdit11.Text;
if fileexists(fullpath)=True then
begin
Image1.picture.loadfromFile(fullpath);
end
else
begin
Image1.picture.loadfromFile(PChar(ExtractFilePath(Application.ExeName))+'\PhotoSot\NoFoto.jpg');
end;

end;

procedure TAdministrator.Button8Click(Sender: TObject);
begin
//ИЗМЕНИТЬ ДАННЫЕ
DataConnectModule.QStaff.Edit;
Button2.Enabled:=True;
Button3.Enabled:=True;
Button12.Enabled:=True;
Button9.Enabled:=True;
DateTimePicker5.Enabled:=True;
DateTimePicker6.Enabled:=True;
Button4.Enabled:=False;
Button5.Enabled:=False;
Button6.Enabled:=False;
Button7.Enabled:=False;
Button11.Enabled:=False;
Button10.Enabled:=False;

```

```

RadioButton9.Enabled:=True;
RadioButton10.Enabled:=True;
DBMemo1.ReadOnly:=False;
DBLookupComboBox1.ReadOnly:=False;
DBEdit4.ReadOnly:=False;
DBEdit5.ReadOnly:=False;
DBEdit6.ReadOnly:=False;
DBEdit7.ReadOnly:=False;
DBEdit8.ReadOnly:=False;
end;

```

```

procedure TAdministrator.Button9Click(Sender: TObject);

```

```

begin

```

```

//Подтверждение ввода

```

```

if DBLookupComboBox1.KeyValue<>null then

```

```

begin

```

```

DataConnectModule.QStaff.Post;

```

```

Button4.Enabled:=True;

```

```

Button5.Enabled:=True;

```

```

Button6.Enabled:=True;

```

```

Button7.Enabled:=True;

```

```

Button8.Enabled:=True;

```

```

Button10.Enabled:=True;

```

```

Button11.Enabled:=True;

```

```

Button8.Enabled:=True;

```

```

Button12.Enabled:=False;

```

```

Button9.Enabled:=False;

```

```

Button2.Enabled:=False;

```

```

Button3.Enabled:=False;

```

```

DBMemo1.ReadOnly:=True;

```

```

DBLookupComboBox1.ReadOnly:=True;

```

```

RadioButton9.Enabled:=False;

```

```

RadioButton10.Enabled:=False;

```

```

DBEdit4.ReadOnly:=True;

```

```

DBEdit5.ReadOnly:=True;

```

```

DBEdit6.ReadOnly:=True;

```

```

DBEdit7.ReadOnly:=True;

```

```

DBEdit8.ReadOnly:=True;

```

```

DateTimePicker5.Enabled:=False;

```

```

DateTimePicker6.Enabled:=False;

```

```

end

```

```

else

```

```

begin

```

```

DataConnectModule.QStaff.Cancel;

```

```

Button4.Enabled:=True;

```

```

Button5.Enabled:=True;

```

					<i>Лист</i>
					09.03.01.2017.064.00 ПЗ
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	95

```

Button6.Enabled:=True;
Button7.Enabled:=True;
Button11.Enabled:=True;
Button8.Enabled:=True;
Button12.Enabled:=False;
Button10.Enabled:=True;
Button9.Enabled:=False;
Button2.Enabled:=False;
Button3.Enabled:=False;
DBMemo1.ReadOnly:=True;
DBLookupComboBox1.ReadOnly:=True;
RadioButton9.Enabled:=False;
RadioButton10.Enabled:=False;
DBEdit4.ReadOnly:=True;
DBEdit5.ReadOnly:=True;
DBEdit6.ReadOnly:=True;
DBEdit7.ReadOnly:=True;
DBEdit8.ReadOnly:=True;
DateTimePicker5.Enabled:=False;
DateTimePicker6.Enabled:=False;
end;
end;

procedure TAdministrator.CheckBox1Click(Sender: TObject);
begin
//Скрыть-показать задание времени для отчётов
if CheckBox1.Checked=True then
  GroupBox3.Visible:=True
else
  GroupBox3.Visible:=False;
end;

procedure TAdministrator.CreateParams(var Params: TCreateParams) ;
//Показ в панели задач
begin
  inherited;
  Params.ExStyle := Params.ExStyle or WS_EX_APPWINDOW;
  Params.WndParent := 0;
end;

procedure TAdministrator.DateTimePicker5Change(Sender: TObject);
begin
//Выбор даты начала действия пропуска
FormatSettings.ShortDateFormat:='dd.MM.yyyy';
FormatSettings.DateSeparator:='.';
DBEdit9.Text:=DateToStr(DateTimePicker5.Date,FormatSettings);

```

```

end;

procedure TAdministrator.DateTimePicker6Change(Sender: TObject);
begin
//Выбор даты окончания пропуска
FormatSettings.ShortDateFormat:='dd.MM.yyyy';
FormatSettings.DateSeparator:='.';
DBEdit10.Text:=DateToStr(DateTimePicker6.Date,FormatSettings);
end;

procedure TAdministrator.DBGrid4ColumnMoved(Sender: TObject; FromIndex,
  ToIndex: Integer);
begin
//Фиксация столбцов
DBGrid1.Columns.Items[ToIndex].Index:=FromIndex;
end;

end.

procedure TForm4.N6Click(Sender: TObject);
{Распечатка отчета}
var a,b,c,f:variant;
index,i:integer;
begin
a:=createOleObject('excel.application');
b:=a.workBooks.open(extractFilePath(application.exename)+'Catalogic.xls');
c:=b.sheets[3];
  c.activate;
  c.range['b1:k'+inttostr(DataModule1.qsotr.RecordCount+3)].horizontalAlignment:=-4108;
  c.rows[1].font.color:= clblack;
  c.rows[1].font.bold:=true;
  c.rows[2].font.bold:=true;
  c.cells[2,1].interior.color:=$004FAC68;
  c.cells[2,2].interior.color:=$004FAC68;
  c.cells[2,3].interior.color:=$004FAC68;
  c.cells[2,4].interior.color:=$004FAC68;
  c.cells[2,5].interior.color:=$004FAC68;
  c.cells[2,6].interior.color:=$004FAC68;
  c.cells[2,7].interior.color:=$004FAC68;
  c.cells[2,8].interior.color:=$004FAC68;
  c.cells[2,9].interior.color:=$004FAC68;
  c.cells[2,10].interior.color:=$004FAC68;
  c.cells[2,11].interior.color:=$004FAC68;
  c.columns[1].columnwidth:=12;
  c.columns[2].columnwidth:=10;
  c.columns[3].columnwidth:=15;
  c.columns[4].columnwidth:=22;

```

					<i>Лист</i>
					09.03.01.2017.064.00 ПЗ
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	97

```

c.columns[5].columnwidth:=26;
c.columns[6].columnwidth:=33;
c.columns[7].columnwidth:=12;
c.columns[8].columnwidth:=16;
c.columns[9].columnwidth:=14;
c.columns[10].columnwidth:=22;
c.columns[11].columnwidth:=8;
f:=c.range['A3:k'+inttostr(DataModule1.qsofr.RecordCount+3)];
f.interior.color:=clskyblue;
c.cells[1,1].interior.color:=clskyblue;
c.cells[1,3].interior.color:=clskyblue;
c.cells[1,2].interior.color:=clskyblue;
c.cells[1,4].interior.color:=clskyblue;
c.cells[1,5].interior.color:=clskyblue;
c.cells[1,6].interior.color:=clskyblue;
c.cells[1,7].interior.color:=clskyblue;
c.cells[1,8].interior.color:=clskyblue;
c.cells[1,9].interior.color:=clskyblue;
c.cells[1,10].interior.color:=clskyblue;
c.cells[1,11].interior.color:=clskyblue;
c.cells[1,4].font.size:=20;
c.cells[1,4]:='Отчёт';
c.rows[2+DataModule1.qsofr.RecordCount+1].font.bold:=true ;
c.cells[2,1]:=' Дата ';
c.cells[2,2]:=' Время ';
c.cells[2,3]:=' Событие ';
c.cells[2,4]:='Фамилия';
c.cells[2,5]:='Имя';
c.cells[2,6]:='Отчество';
c.cells[2,7]:='Отдел';
c.cells[2,8]:='Нарушение';
index:=3;
with DataBaseConnet.TMPQuery do begin
first; for i:=1 to recordcount do begin
  c.cells[index,1]:=fieldbyname('Date').AsString;
  c.cells[index,2]:=fieldbyname('Time').AsString;
  c.cells[index,3]:=fieldbyname('EventIO').AsString;
  c.cells[index,4]:=fieldbyname('Familiya').AsString;
  c.cells[index,5]:=fieldbyname('Name').AsString;
  c.cells[index,6]:=fieldbyname('Otchestvo').AsString;
  c.cells[index,7]:=fieldbyname('Otdel').AsString;
  c.cells[index,7]:=fieldbyname('Violations').AsString;
  inc(index);
  next;
end;
end;

```

```
c.cells[1,4].interior.color:=clskyblue;;  
c.cells[1,5].font.size:=20;  
c.cells[1,5]:=' от: '+FormatDateTime('dd/mm/yyyy', Now);  
c.cells[1,5].interior.color:=clskyblue;  
a.visible:=true;  
end;
```

					09.03.01.2017.064.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		99