

Министерство науки и высшего образования Российской Федерации
Филиал Федерального государственного автономного образовательного учреждения
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
в г. Нижневартовске

Кафедра «Гуманитарные, естественно-научные и технические дисциплины»

ДОПУСТИТЬ К ЗАЩИТЕ

И.о.зав.кафедрой «ГЕНТД»
к.филол.н, доцент

/И.Г.Рябова

« ____ » _____ 2019 г.

Разработка менеджера паролей

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ ЮУрГУ-09.03.01.2019.793.ПЗ ВКР

Консультанты
Экономическая часть

к.э.н., доцент

/А.В.Прокопьев/

« ____ » _____ 2019г.

Безопасность жизнедеятельности

к.т.н., доцент

/В.В.Столяров/

« ____ » _____ 2019 г.

Руководитель работы

к.т.н., доцент

/В. А.Парасич/

« ____ » _____ 2019 г.

Автор работы

обучающийся группы НвФл-528

/ В.В.Мышковский/

« ____ » _____ 2019г.

Нормоконтролер

старший преподаватель

/Л.Н.Буйлушкина/

« ____ » _____ 2019г.

АННОТАЦИЯ

Мышковский В.В. Разработка менеджера паролей.
– Нижневартовск: ЮУрГУ, НвФл-528: 2019, 82 с.,
41 ил., 8 табл., библиогр. список – 20 наим., 2
прил.

В ходе выполнения работы исследована предметная область, проведен анализ существующих на рынке решений, выявлены недостатки этих решений для использования в организациях среднего и больших размеров.

В результате работы был разработан программный продукт, реализующий функционал хранителя паролей. В отличие от аналогов программа использует систему управления базой данных, чтобы обеспечить полнофункциональный многопользовательский доступ.

Были рассмотрены вопросы безопасности жизнедеятельности и проведено технико-экономическое обоснование проекта.

					ЮУрГУ-09.03.01.2019.793.ПЗ ВКР							
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>								
<i>Разраб.</i>		Мышковский В.В.			<i>Разработка менеджера паролей</i>			<i>Лит.</i>		<i>Лист</i>	<i>Листов</i>	
<i>Провер.</i>		Парасич В.А.						<i>В</i>	<i>К</i>	<i>Р</i>	5	82
<i>Н. Контр.</i>		Буйлушкина						<i>Филиал ФГАОУ ВО «ЮУрГУ (НИУ)» в г. Нижневартовске кафедра «ГЕНТД»</i>				
<i>Утверд.</i>		Рябова И.Г.										

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
1 АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ	9
1.1 Предметная область и решаемые задачи	9
1.2 Анализ существующих решений.....	10
1.2.1 KeePass..	11
1.2.2 Dashlane.	14
1.2.3 1Password	16
1.2.4 LastPass..	18
1.2.5 Sticky Password	19
2 ПРОЕКТНАЯ ЧАСТЬ	23
2.1 Выбор системы управление базой данных.....	23
2.1.1 Oracle 18c	23
2.1.2 MySQL.....	24
2.1.3 Microsoft SQL сервер	26
2.2 Выбор языка программирования	27
2.3 Разработка приложения.....	27
2.3.1 Установка и настройка СУБД.....	28
2.3.2 Проектирование БД	31
2.3.3 Разработка пользовательского интерфейса. Руководство пользователя.	33
3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ.....	41
3.1 Описание проектного решения.....	41
3.2 Постановка задачи.....	41
3.3 Расчет сметы затрат на разработку программного комплекса	42
3.4 Расчет сметы затрат на разработку программного комплекса	42
3.5 Расчет затрат на амортизацию оборудования	42
3.6 Расчет затрат на заработную плату	44

3.7	Расчет затрат на дополнительную заработную плату	44
3.8	Общая смета затрат на внедрение системы.....	45
4	БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ	47
4.1	Характеристика вредных факторов при работе с вычислительной техникой при написании выпускной квалификационной работы	47
4.2	Организация рабочего места.....	49
4.3	Санитарно-гигиенические параметры рабочих мест	50
4.3.1	Микроклиматические условия.....	50
4.3.2	Защита от шума	51
4.3.3	Освещение	54
4.3.4	Действие электромагнитного излучения.....	55
4.4	Требования к режиму труда и охраны	56
4.5	Влияние статических и зрительных нагрузок, меры защиты.....	57
4.5.1	Влияние статических нагрузок и их профилактика	57
4.5.2	Влияние статических нагрузок и их профилактика	58
4.6	Пожарная безопасность	61
	ЗАКЛЮЧЕНИЕ	65
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК	66
 ПРИЛОЖЕНИЯ		
	ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ПРОГРАММЫ	68
	ПРИЛОЖЕНИЕ Б. КОМПАКТ - ДИСК	82

ВВЕДЕНИЕ

Системные администраторы предприятий любого размера ежедневно сталкиваются с необходимостью в предоставлении, хранении и получении быстрого доступа к множеству комбинаций логинов и паролей.

Исходя из этого возникает необходимость в генерировании и хранении большого количества комбинаций логина и пароля. Кроме того, генерируемые пароли должны быть устойчивыми к взлому: иметь большое количество символов, состоять из букв верхнего и нижнего регистров, с использованием цифр и различных знаков. Все это способна предоставить такая программная система как менеджер паролей.

Анализ существующих решений в области программных систем хранения паролей выявил факт недостаточной функциональности для их использования в рамках организаций среднего либо крупного размеров, что является обоснованием актуальности темы выпускной квалификационной работы.

Целью работы является разработка программной системы хранения паролей.

Для выполнения поставленной цели необходимо решить следующие задачи:

- произвести анализ предметной области;
- произвести анализ существующих решений;
- произвести выбор системы управления базой данных;
- произвести выбор языка программирования;
- спроектировать базу данных;
- разработать пользовательский интерфейс;
- рассчитать затраты на внедрение системы;
- разработать меры обеспечения безопасности жизнедеятельности.

Объектом исследования являются программные системы хранения паролей.

Предметом исследования являются функциональные параметры таких систем, позволяющие внедрить собственное программное решение в организацию с большим количеством сотрудников.

1 АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

1.1 Предметная область и решаемые задачи

Менеджер паролей — программное обеспечение (далее – ПО), которое оказывает помощь пользователю при работе с паролями и PIN-кодами. У такого ПО, как правило, присутствует база данных (далее – БД) или файлы, в которых хранятся зашифрованные сведения пароля, она может быть расположена как локально на компьютере пользователя, так и удаленно на сервере. Многие системы хранения паролей также работают как заполнитель формы, то есть они заполняют поле логина и данные пароля автоматически. В большинстве случаев они представлены как расширение браузера.

Менеджеры паролей делятся на три основных категории:

Десктоп — хранят пароли к ПО, установленному на жестком диске компьютера.

Сетевые — менеджеры паролей онлайн, где пароли сохранены на веб-сайтах провайдеров.

Портативные — хранят пароли к программному обеспечению на мобильных устройствах, таких как планшет, смартфон или к портативным приложениям на USB-накопителе.

Системы хранения паролей могут также использоваться как защита от фишинга (вид интернет-мошенничества, основанного на принципах социальной инженерии, которым движет возможность украсть конфиденциальные данные пользователей — логины и пароли). В отличие от пользователей, менеджер паролей способен использовать автоматизированный скрипт логина не восприимчиво к визуальным подражаниям, которые имеют сходства с вебсайтами. С этим внедренным преимуществом использование систем хранения паролей благоприятно, даже если у пользователя есть всего несколько паролей, которые он «держит в голове».

Менеджеры паролей, как правило, применяют выбранный пользователем главный пароль, или же секретное слово, чтобы создать ключ, применяемый для зашифровки хранимых паролей. Этот основной пароль следует делать достаточно сложным, чтобы удержать атаки злоумышленников.

Если главный пароль будет украден, то, следовательно, будут рассекречены все хранимые в базе данных (далее – БД) программы пароли. Это демонстрирует обратную связь между удобством использования и безопасностью: единственный пароль может быть более практичен, но если он будет взломан, то под угрозой окажутся все хранимые пароли.

Основной пароль также может быть обнаружен и атакован при применении криптоанализа. Такая угроза может быть снижена путём использования виртуальной клавиатуры.

Так же существуют менеджеры паролей с одновременным однопользовательским и многопользовательским доступом, менеджеры с многопользовательским доступом дают возможность подключения к базе сразу нескольким пользователям одновременно.

Исходя из вышесказанного, определено, что:

Менеджер паролей должен решать за человека такой вопрос, как придумывание сложных и надежных паролей. ПО должно обеспечивать возможность одновременного многопользовательского подключения, т.к. как правило число пользователей программы равно числу сотрудников, обслуживающих информационную инфраструктуру организации.

1.2 Анализ существующих решений

На сегодняшний день представлен большой выбор различных систем хранения паролей. Проведем анализ наиболее популярных программ. Для анализа выделим параметры, которые позволят оценить функциональность этих менеджеров паролей наиболее полно:

- стоимость;
- удобство пользовательского интерфейса (где 1 – перегруженный интерфейс и неудобный интерфейс, 5 – простой и удобный интерфейс);
- кроссплатформенность;
- генератор паролей;
- перенос БД;
- поиск по записям;
- смена мастер-пароля;
- одновременный многопользовательский доступ.

1.2.1 KeePass

KeePass — одна из самых надежных и известных программ по хранению паролей. Распространяется по лицензии GPL v2, имеет открытый исходный код и поддерживает несколько языков, в том числе и русский. Официальная реализация есть только под Windows, но исходный код открыт, поэтому есть огромное количество реализаций, в том числе под Linux и под OS X. Также есть приложения и для мобильных ОС. БД паролей шифруется с помощью симметричного AES-256, а мастер-пароль хэшируется с SHA-256.

Программа позволяет пользователю хранить множество паролей в одном зашифрованном файле, доступ к которым предоставляется после ввода мастер-пароля т.е. используется один надежный пароль, который открывает доступ ко всем остальным.

Для KeePass есть большое количество плагинов и дополнительных инструментов: утилиты для импорта/экспорта паролей из БД, браузерные плагины, например такие как AutoType, позволяющий автоматически вводить пароли в браузерах и других программах.

Данный менеджер имеет свои минусы: сейчас используется две версии БД, несовместимых друг с другом. Первая написанная под Windows, и вторая

написанная на .NET и работающая в том числе под Mono в OS X и Linux. Так как в качестве БД используется отдельный файл у программы имеются проблемы с многопользовательским доступом.

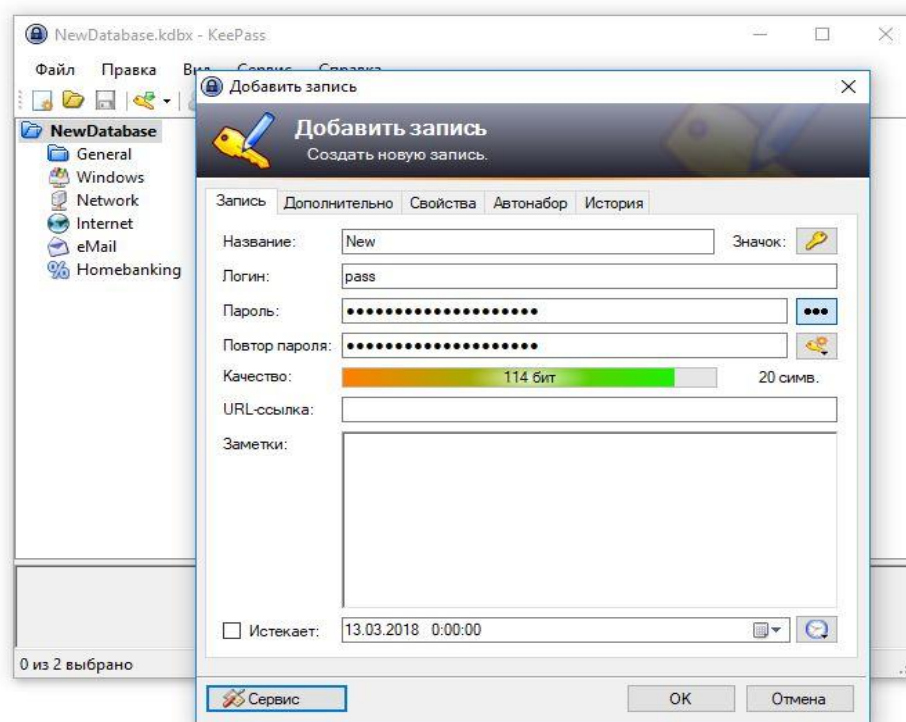


Рисунок 1.1 – Создание учетной записи в KeePass

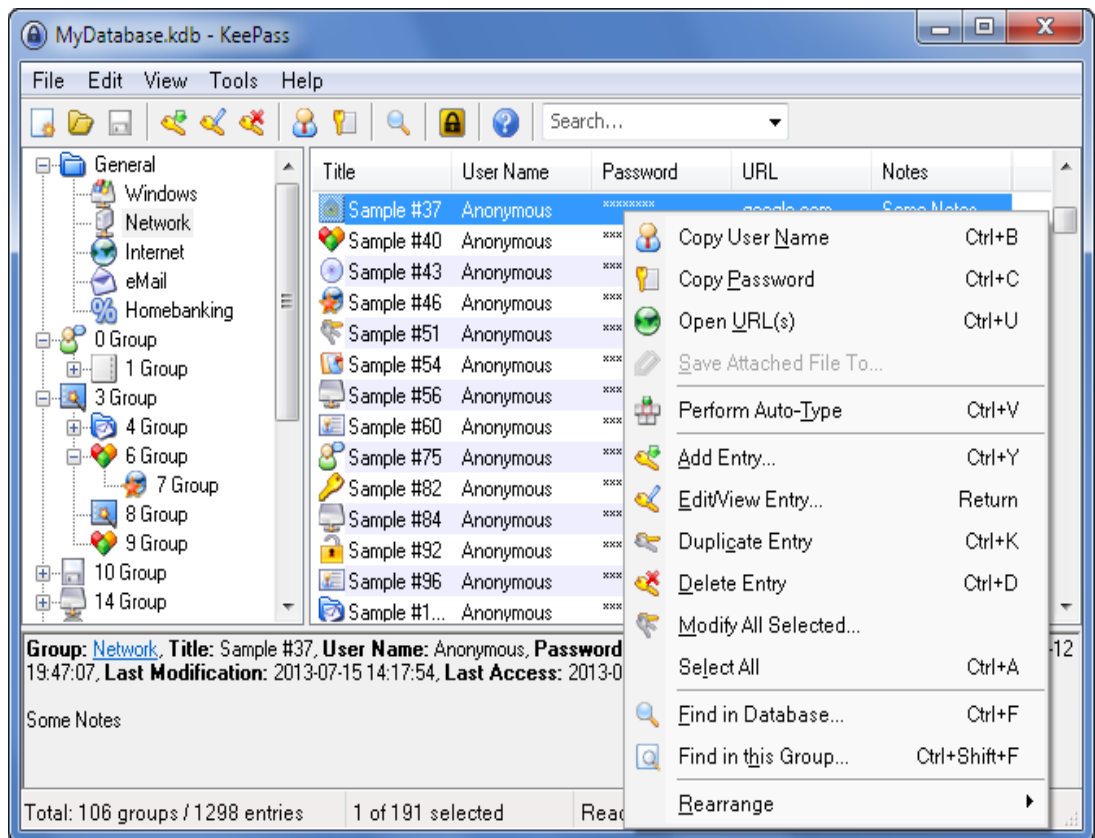


Рисунок 1.2 – Главная форма

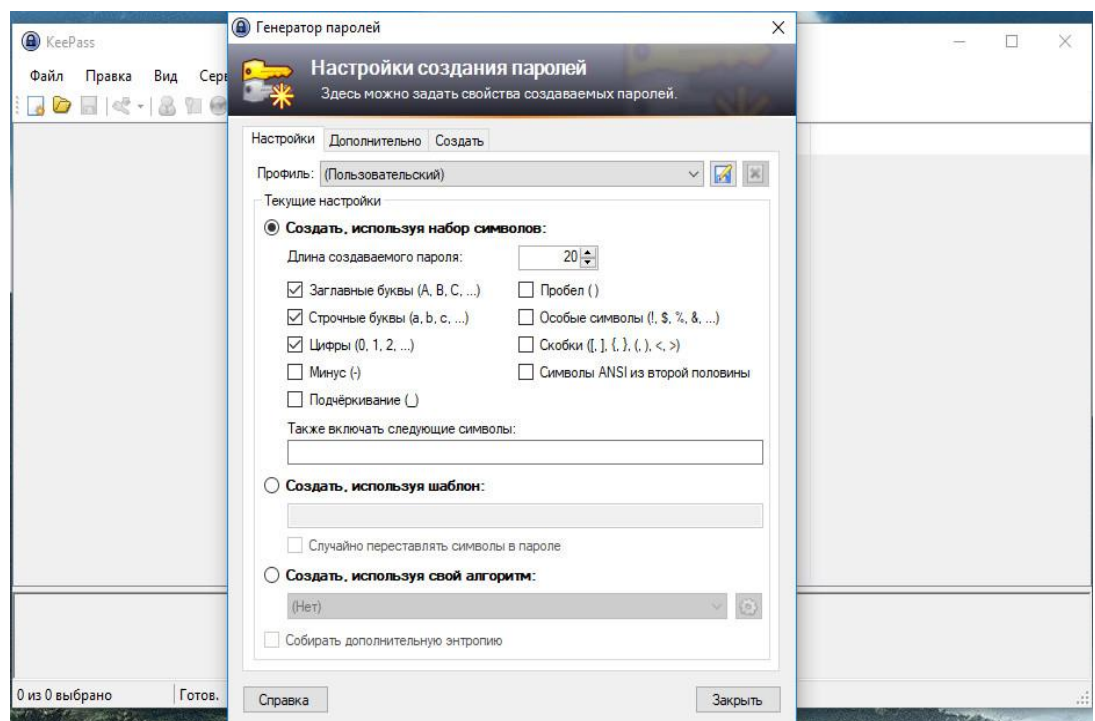


Рисунок 1.3 – Форма настроек генератора паролей

Анализ:

- стоимость – официальная версия для Windows бесплатна, но также имеются платные версии приложения;
- удобство пользовательского интерфейса – 4;
- кроссплатформенность – присутствует;
- генератор паролей – присутствует;
- перенос БД – присутствует;
- поиск по записям – присутствует;
- смена мастер пароля – присутствует;
- одновременный многопользовательский доступ – присутствует, но ограничен, пользователям необходимо синхронизироваться с БД, одновременная синхронизация влечет к потере изменений и ошибкам в БД.

1.2.2 Dashlane

Dashlane – менеджер паролей с двойной авторизацией, генератором паролей и системой предупреждения взлома аккаунтов. Программа поддерживается Windows, Mac, iOS, Android. Присутствует функция автоматизированной замены паролей на популярных сайтах. Также реализован импорт паролей, сохраненных в браузерах Chrome, Firefox и Internet Explorer. Кроме того, Dashlane позволяет импортировать пароли из LastPass и нескольких других конкурентных решений.

Менеджер паролей использует облачную синхронизацию между устройствами. Все пароли шифруются локально на устройстве пользователя, при шифровании используется мастер-пароль, который никогда не покидает устройство пользователя, и один из методов шифрования: Argon2d, PBKDF2 10,204 или PBKDF2 200,000. Дополнительно для авторизации возможно использовать двухфакторную аутентификацию, авторизацию через Windows Hello или физическим ключом U2F. На смартфонах доступна защита пин-кодом или отпечатком пальца.

Поставляется в двух версиях, на бесплатной версии существует ограничение не позволяющее создать более 50 паролей, а также отсутствует возможность использования облачной синхронизации между устройствами. Полноценная версия будет доступна после оформления подписки.

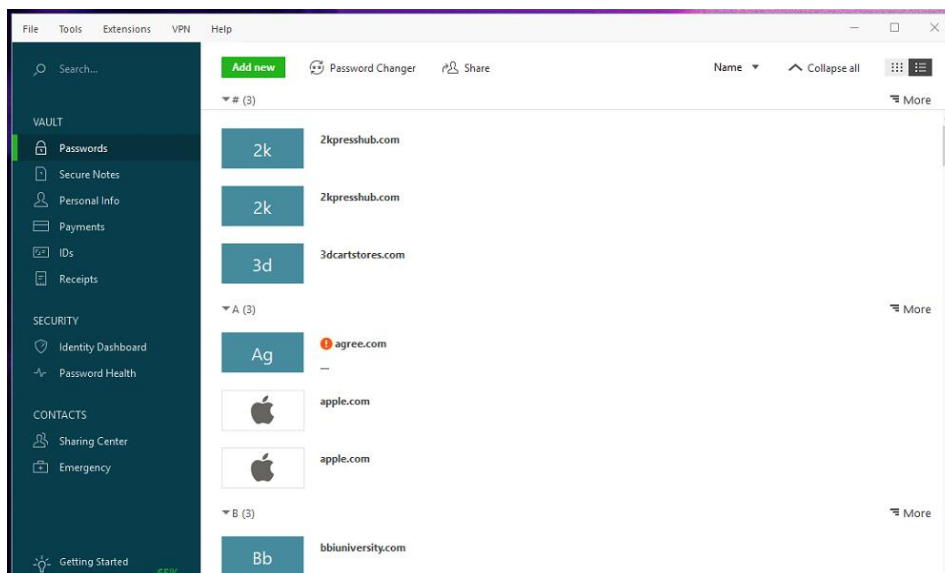


Рисунок 1.4 – Главная форма, вкладка Passwords

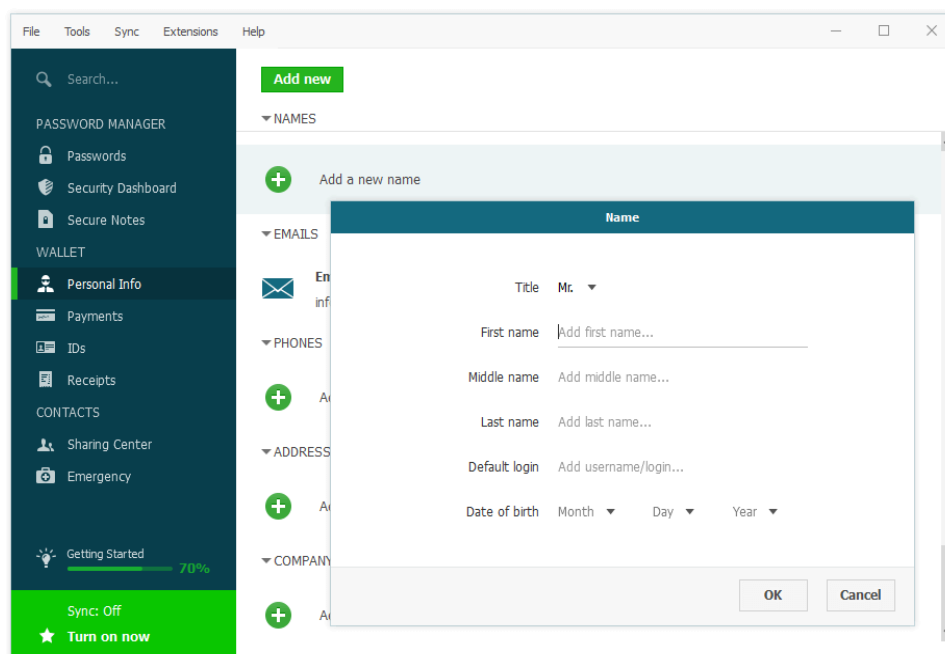


Рисунок 1.5 – Добавление пользователя на вкладке Personal Info

Анализ:

- стоимость – бесплатно (без синхронизации, бэкапов и веб-доступа), 4,99\$ в месяц за полную версию;
- удобство пользовательского интерфейса – 3;
- кроссплатформенность – присутствует;
- генератор паролей – присутствует;
- перенос БД – присутствует;
- поиск по записям – присутствует;
- смена мастер пароля – присутствует.
- одновременный многопользовательский доступ – присутствует в полной версии.

1.2.3 1Password

1Password — популярное на Mac OS X решение для хранения паролей, лицензий на ПО и другой персональной информации от компании AgileBits. Также есть версия под Windows.

Все версии 1Password обладают встроенной функцией синхронизации базы с помощью сервиса Dropbox. Эта функциональность не обязательна, БД по умолчанию хранится локально и зашифрована AES-128.

1Password для Mac интегрируется с Safari, Firefox, Chrome и Camino «из коробки». Версия для Windows интегрируется с Firefox, Chrome и IE. Также обе версии 1Password предлагают удобный интерфейс для использования хранимой информации в любых других приложениях.

Кроме интеграции с разными платформами, 1Password предоставляет еще один оригинальный способ доступа к своей базе. Хранилище паролей (agile keychain) представляет из себя набор файлов, один из которых — HTML-файл с полноценным интерфейсом для работы с базой, который может быть открыт любым браузером на практически любом устройстве.

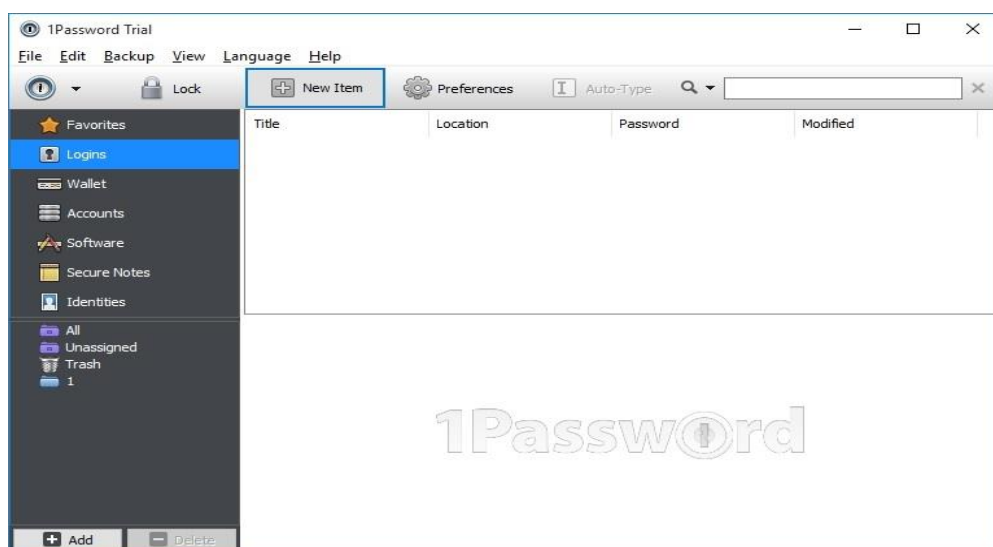


Рисунок 1.6 – Главная форма

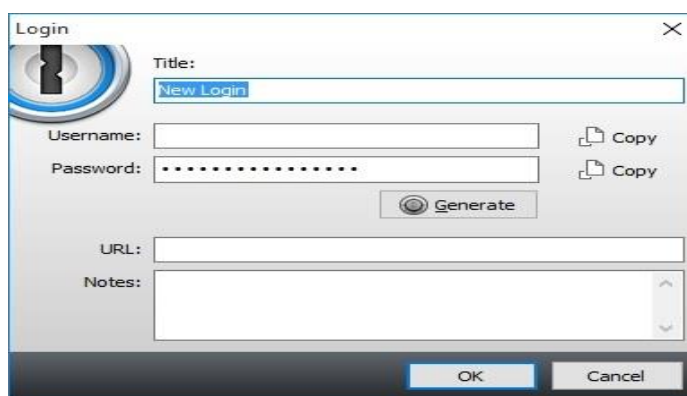


Рисунок 1.7 – Добавление пользователя

Анализ:

- стоимость – первый месяц использования бесплатна, далее 1 месяц 4,99\$;
- удобство пользовательского интерфейса – 4;
- кроссплатформенность – частично присутствует;
- генератор паролей – присутствует;
- перенос БД – присутствует;
- поиск по записям – присутствует;
- смена мастер пароля – присутствует.
- одновременный многопользовательский доступ – присутствует;

1.2.4 LastPass

LastPass – популярный менеджер паролей, разработанный американской компанией LastPass Corporate. Среди удобств программы обычно отмечают её простой интерфейс, кроссплатформенность, надёжное шифрование паролей, которое происходит на вашем устройстве, и только потом данные уже в зашифрованном виде передаются на серверы компании.

Приложение работает как на Windows, так и на Linux, Mac, а также на мобильных устройствах и планшетах под iOS и Android, на смартфонах под управлением Windows Phone, BlackBerry и даже FireFox Mobile. Кроме «домашней» версии, существует и корпоративная, имеющая функцию интеграции с локальным контроллером домена и настройки политик доступа. Также она хранит данные обо всех учётных записях сотрудников в корпоративных интернет-службах. Например, с помощью LastPass возможно настроить доступ работников организации к Office 365, Google Apps for Work и прочим бизнес-сервисам в Интернете, которые использует компания.

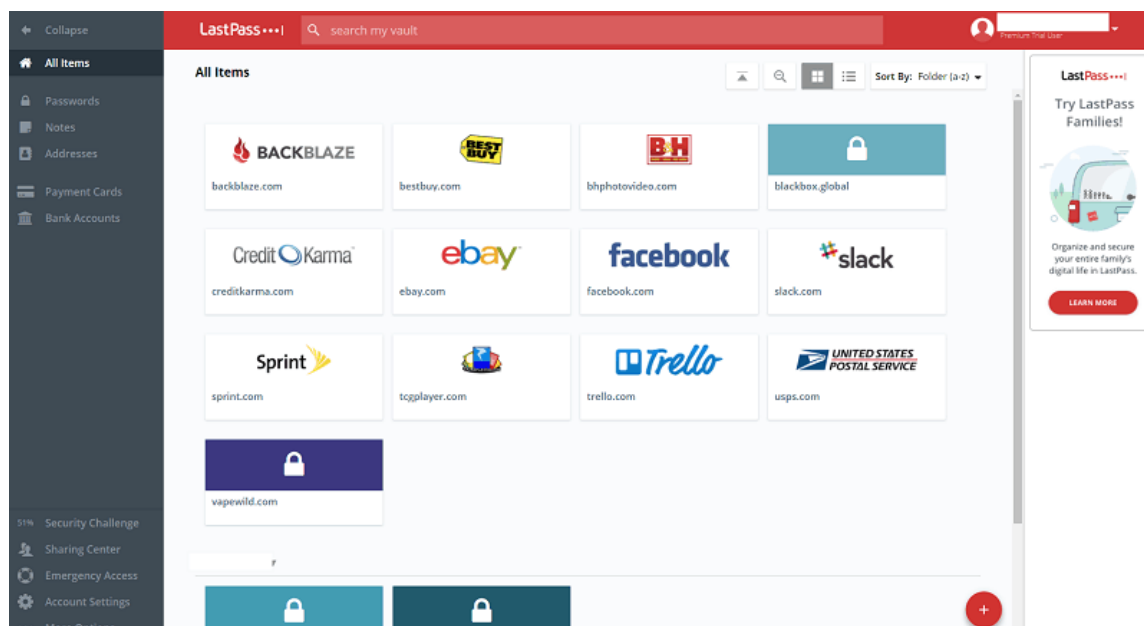


Рисунок 1.8 – Главная форма программы

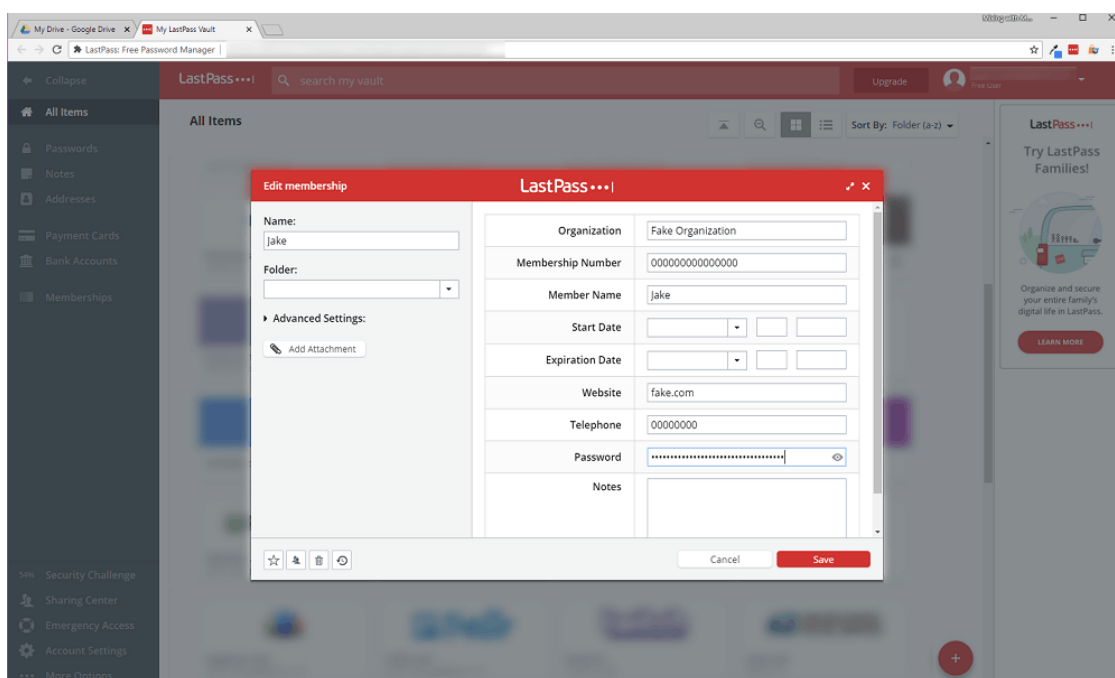


Рисунок 1.9 – Добавление пользователя

Анализ:

- стоимость – первый месяц использования бесплатна, далее в случае оформления подписки на год 3\$ месяц либо 5\$ в месяц, в зависимости от комплектации;
- удобство пользовательского интерфейса – 3;
- кроссплатформенность – присутствует;
- генератор паролей – присутствует;
- перенос БД – присутствует;
- поиск по записям – присутствует;
- смена мастер пароля – присутствует;
- одновременный многопользовательский доступ – присутствует.

1.2.5 Sticky Password

Приложение доступно на нескольких платформах и также умеет синхронизировать аккаунты с разных устройств. Согласно заявлениям разработчиков, Sticky Password умеет хранить пароли в зашифрованном

хранилище, а также автоматически заполнять формы и номера кредитных карт для быстрого использования. Защищается информация традиционным мастер-паролем, который и нужно запомнить. Как и в случае с LastPass, создатели системы сообщают, что не знают пользовательский мастер-пароль, поэтому его нужно заучить и не забывать. Из более интересных возможностей в программе имеется поддержка шифрования стандарта AES-256. Также авторы обещают корректную работу приложения в четырёх наиболее популярных операционных системах (далее – ОС) и шестнадцати браузерах, включая Google Chrome, Mozilla Firefox, Safari и Internet Explorer.

Sticky Password распространяется по подписке. При первой установке программы предлагается создать учётную запись, после чего пользователь получает 30 пробных дней премиума. Через месяц учётная запись Sticky Password автоматически становится базовой. Традиционно премиум-пользователи имеют ряд преимуществ перед обычными. Например, только премиум пользователям доступна возможность синхронизации (как через Wi-Fi, так и через облачные службы компании), а также сохранение резервных копий хранилища паролей в облаке.

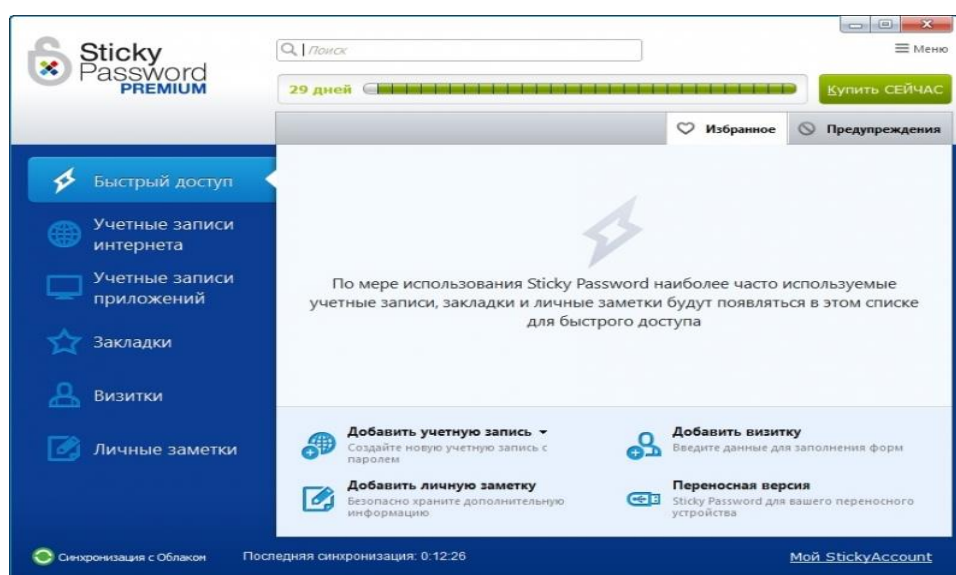


Рисунок 1.10 – Главная форма

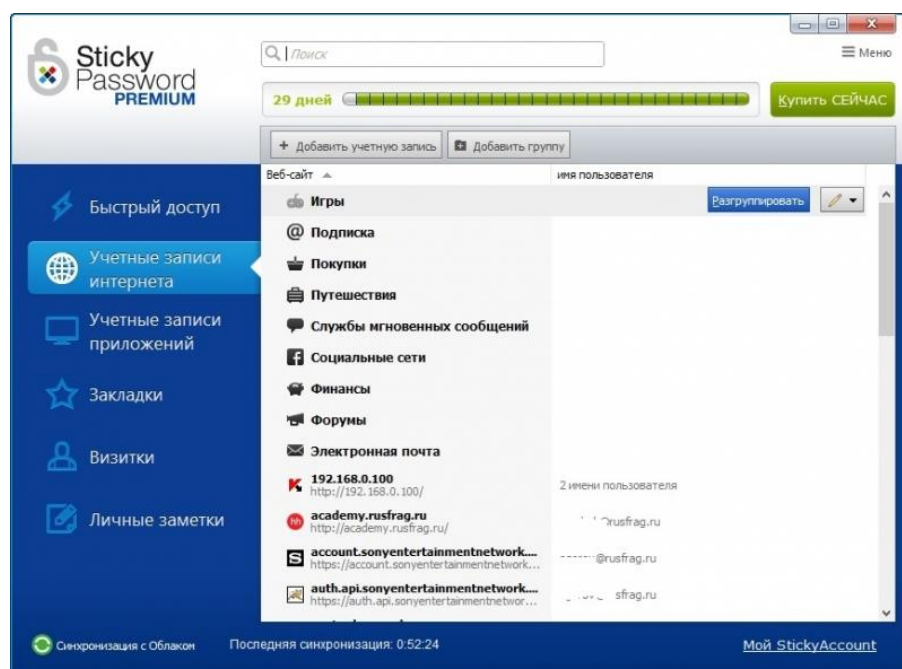


Рисунок 1.11 – Раздел учетных записей

Анализ:

- стоимость – премиум аккаунта на одного пользователя составляет 945 рублей за один год, за бессрочную премиум-лицензию 4845 рублей;
- удобство пользовательского интерфейса – 3;
- кроссплатформенность – присутствует;
- генератор паролей – присутствует;
- перенос БД – присутствует;
- поиск по записям – присутствует;
- смена мастер пароля – присутствует;
- одновременный многопользовательский доступ – присутствует.

Выводы по разделу один:

Проанализировав существующие решения, можно сказать, что ни одна из программ не подойдет в качестве решения, которое можно использовать в организациях средних и больших размеров. Каждая из рассмотренных

программных систем для хранения паролей имеет недостатки, которые сделают ее эксплуатацию крайне неудобной.

Практически все решения рассчитаны лишь для организация малого размера либо домашнего использования, где подразумевается один или несколько пользователей и хранение различных личных паролей. Часть решений осуществляет хранение информации на облачных серверах, что не допускается т.к. некоторые данные попадают под категорию персональных. Интерфейс всех рассмотренных решений, кроме KeePass, не позволяет удобно разбить пользователей на группы, а возможность полноценного многопользовательского доступа в KeePass отсутствует.

В рамках учета пользовательских данных организации следует выделить следующие наиболее важные функции:

- поиск по записям – является обязательным условием т.к. в программе будет большое количество полей;
- удобный пользовательский интерфейс – важный критерий, именно через него пользователи взаимодействует с программой. Необходимо, чтобы интерфейс был интуитивно понятен и прост;
- генератор паролей – функция, которая в случайном порядке формирует набор нелогичных символов большей или меньшей степени сложности, зависящей от заданных параметров. Эта функция нужна, чтобы пользователь не придумывал пароль самостоятельно;
- одновременный многопользовательский доступ – самый важный критерий, необходимо, чтобы работу в программе вели одновременно множество пользователей;

На основании выявленных недостатков рассмотренных систем принимается решение о необходимости реализации собственной разработки.

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Выбор системы управление базой данных

Несмотря на то, что все системы управления базами данных (далее – СУБД) выполняют одну и ту же основную задачу, выбор СУБД является сложной многопараметрической задачей и одним из важных этапов при разработке приложений использующих БД. Кроме того, функции и возможности каждой СУБД могут существенно отличаться. Выбранный программный продукт должен удовлетворять как текущим, так и будущим потребностям предприятия, при этом следует учитывать финансовые затраты на приобретение необходимого оборудования, самой системы и разработку необходимого программного обеспечения на ее основе. Различные СУБД документированы по разному: более или менее тщательно. По разному предоставляется и техническая поддержка.

Рассмотрим некоторые из них.

2.1.1 Oracle 18c

Oracle в настоящее время – БД номер один на рынке СУБД для корпоративных информационных систем. Высокая надежность и производительность являются качествами, которые оптимальным образом подходят для средних и крупных предприятий, где необходимо автоматизировать десятки и сотни рабочих мест и обеспечить обработку большого количества документов. Существуют различные поставки СУБД для обеспечения практически любых требований пользователя.

Отличительной чертой Oracle является наличие уникальных технологий в своих продуктах, например некоторые из функций новейшей версии Oracle включают в себя Automatic Indexing, Active Data Guard DML Redirect и другие.

Используя функцию Automatic Indexing БД самостоятельно определяет оптимальный набор индексов. При их создании учитываются структура данных, сами данные и запросы. Это означает, что БД можно создать и начать использовать вовсе без каких-либо индексов или с очень небольшим набором индексов. БД проанализирует ресурсоемкие запросы, чтобы понять, какие индексы потенциально нужны, проверит полезность отобранных кандидатов, а затем построит индексы, причем все это займет меньше времени, чем создание индексов вручную.

Многим приложениям при создании отчетов требуется не только читать данные, но и делать некоторые записи в БД. С помощью Active Data Guard DML Redirect операции записи немедленно и прозрачно перенаправляются в первичную БД и после занесения в нее реплицируются в резервную БД. С точки зрения пользователя это выглядит так, как будто он обновляет данные в резервной БД.

Достоинства:

- наличие уникальных технологий, таких как Oracle RAC, Total Recall и других;
- высокая производительность, наряду с надежностью.

Недостатки:

- очень большая стоимость;
- высокие требования к оборудованию т.к. СУБД может потребовать значительных ресурсов.

2.1.2 MySQL

MySQL одна из самых популярных БД. Фактически, является стандартом для веб-серверов, которые работают под управлением ОС Linux. MySQL — это свободная реляционная СУБД с открытым исходным кодом. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на

торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB.

Открытость исходного кода означает, что любой желающий имеет возможность использовать и модифицировать это программное обеспечение по своему усмотрению. Каждый пользователь, при желании, может изучить исходные тексты и изменить их в соответствии со своими потребностями. Программное обеспечение MySQL распространяется по лицензии GPL (GNU General Public License), которая регламентирует, что разрешено, а что нет в отношении программного обеспечения. Если по тем или иным причинам лицензия GPL не устраивает либо код MySQL требуется встраивать в коммерческие приложения, следует приобрести коммерческую лицензированную версию у компании MySQL AB

СУБД MySQL является клиент-серверной системой, включающей многопоточный SQL-сервер, прддерживающий различные платформы, несколько клиентских программ и библиотек, инструменты администрирования и широкий диапазон программных интерфейсов приложений.

Сервер MySQL существует также и в форме встраиваемое многопоточной библиотеки, которую можно связывать с разрабатываемыми приложениями, чтобы получить более компактные, быстрые и легкоуправляемые продукты.

Система невероятно надежна и не стремится подчинить себе все доступные аппаратные ресурсы.

Достоинства:

- имеет двойное лицензирование;
- открытый код;
- надежность;
- возможность работы с другими БД, например DB2 или Oracle.

Недостатки:

- медленная скорость чтения и записи файловой БД;
- отсутствует встроенная поддержка XML.

2.1.3 Microsoft SQL Server

Ещё одна популярная СУБД Microsoft SQL Server — это платформа для решения критически важных задач в масштабе предприятия, обладающая высокой доступностью, повышенной производительностью и безопасностью. Решение представляет собой хорошо масштабируемый, полностью реляционный, быстродействующий сервер, способный обрабатывать большие объемы данных для клиент-серверных приложений.

Рекордная производительность MS SQL Server обеспечивается новыми технологиями работы с памятью, что помогает предприятиям ускорить свой бизнес и реализовать новые сценарии работы. Кроме того, SQL Server позволяет использовать новые гибридные облачные решения и пользоваться новыми преимуществами облачных вычислений. Расширенные функции безопасности, в сочетании со встроенными, удобными для использования инструментами и управляемым доступом к данным, позволяют организации выполнить требования строгих политик соответствия нормам

Достоинства:

- простота в использовании;
- быстрая и стабильная работа;
- возможность отслеживания уровня производительности;
- отличное взаимодействие с другими продуктами Microsoft;
- полнофункциональные свободно распространяющиеся версии.

Недостатки:

- даже при тщательной настройке SQL Server способен занять все доступные ресурсы (при сильной нагрузке).

Именно на Microsoft SQL Server следует остановить выбор, продукт имеет множество преимуществ, в числе которых простота использования, наличие полнофункциональной свободно распространяющейся версии и отличное взаимодействие с другими продуктами Microsoft.

2.2 Выбор языка программирования

При выборе среды разработки необходимо сравнить программные продукты и их возможности.

Инструментальные средства позволяют:

- создавать интерфейс используя стандартные компоненты;
- создавать БД;
- разрабатывать более надежные программы.

Средства разработки характеризуются такими параметрами как:

- поддержка объектно-ориентированного программирования;
- поддержка БД;
- возможность использования CASE-технологий;
- использование визуальных компонентов для проектирования

интерфейса.

Свойствами, перечисленными выше обладают такие языки программирования как:

- C++;
- C#;
- Java.

Для решения поставленной задачи выбор пал на язык программирования C# так как выбранная ранее СУБД легко взаимодействует со средой разработки Microsoft Visual Studio.

2.3 Разработка приложения

Исходя из составленного списка требований к разрабатываемому менеджеру паролей основные функции, выделенные к реализации это:

- разработка простого и удобного пользовательского интерфейса;

- реализация функций генератора паролей;
- обеспечение одновременного многопользовательского доступа;

2.3.1 Установка и настройка СУБД

Для развертывания СУБД потребуется произвести установку компонентов Microsoft SQL server, после чего установить Microsoft SQL Server Management Studio который представляет собой графический набор средств, для работы и управления компонентами SQL сервера. Установка Microsoft SQL server (Рисунок 2.1)

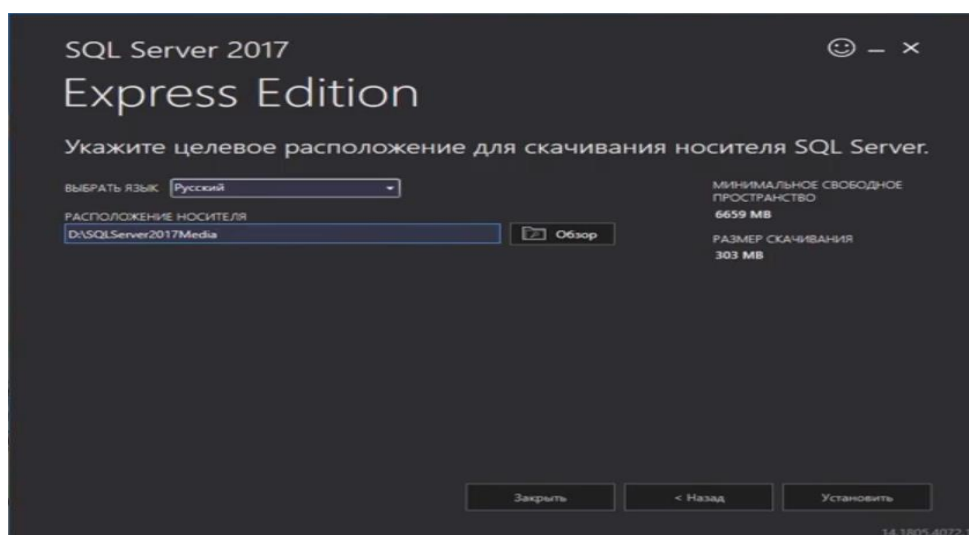


Рисунок 2.1 – Установка SQL Server

Во время установки никаких действий по настройке не требуется, необходимо лишь выбрать нужные компоненты (Рисунок 2.2), выбрать режим запуска служб (Рисунок 2.3), а затем включить режим проверки подлинности Windows (Рисунок 2.4).

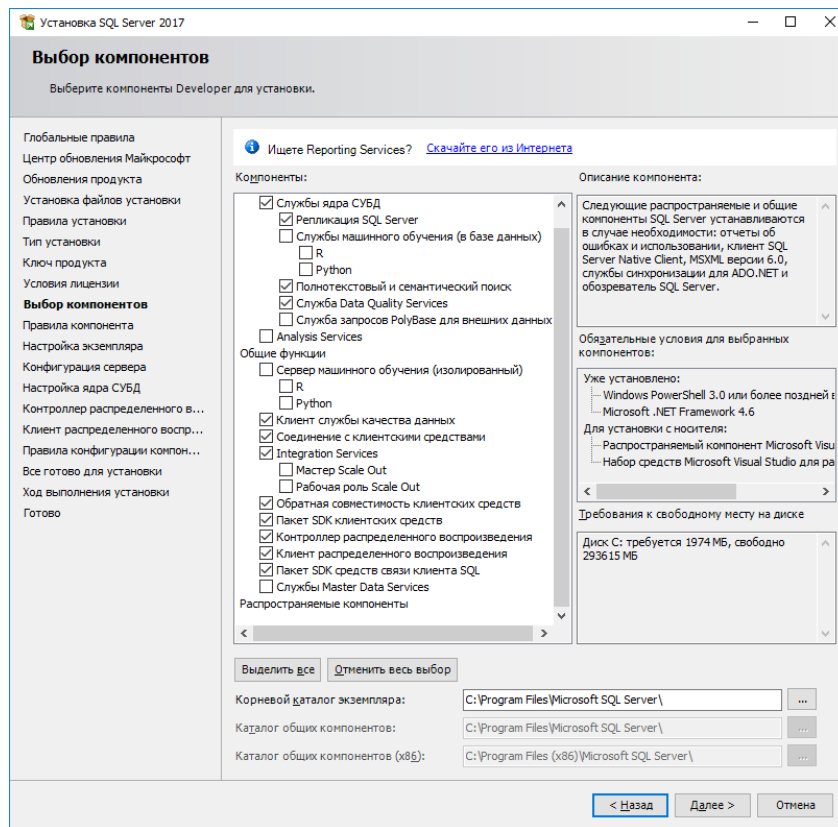


Рисунок 2.2 – Выбор компонентов

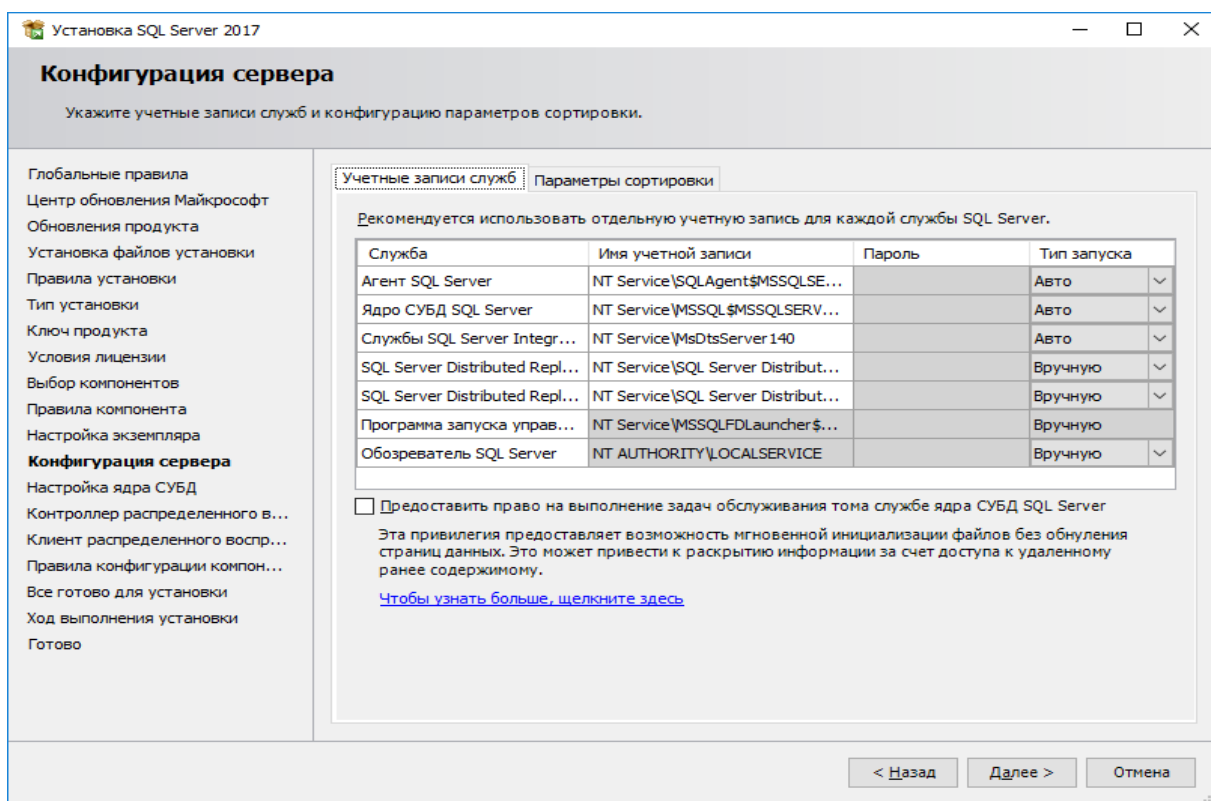


Рисунок 2.3 – Режим запуска служб

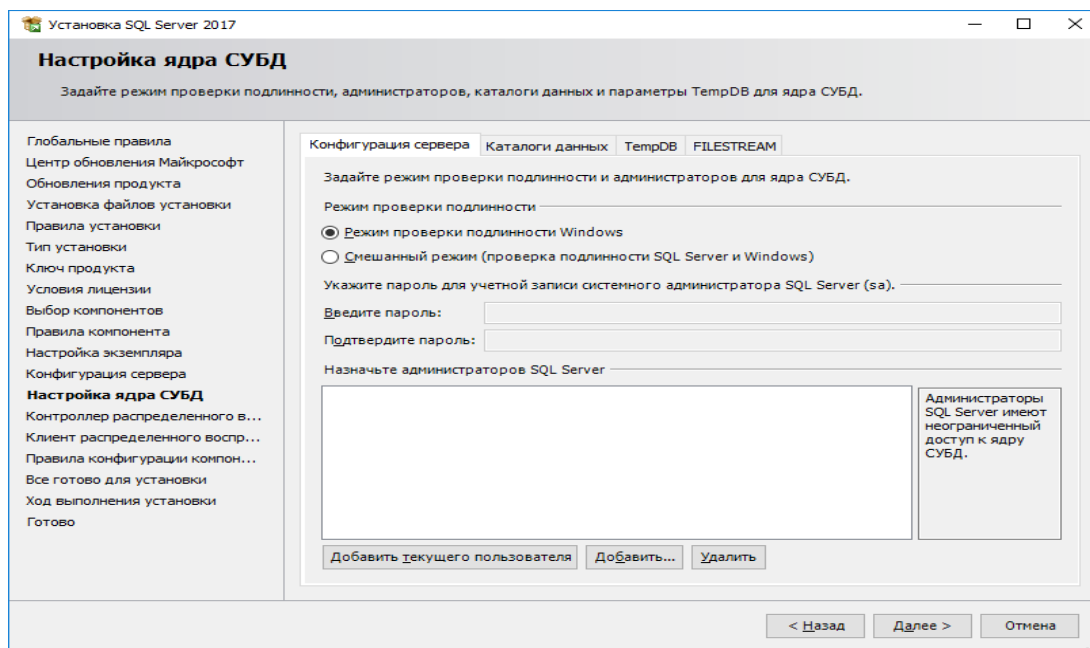


Рисунок 2.4 – Режим проверки подлинности Windows

После успешной установки Microsoft SQL Server необходимо установить Microsoft SQL Server Management Studio (Рисунок 2.5).

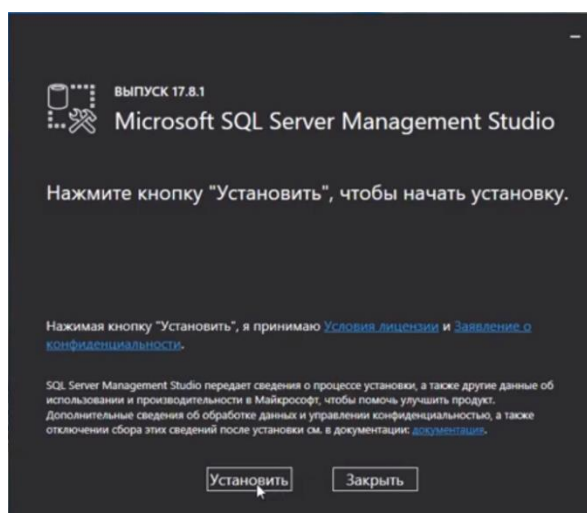


Рисунок 2.5 – Установка Microsoft SQL Server Management Studio

Когда установка будет завершена необходимо настроить сервер для того, чтобы другие пользователи могли подключаться к нему по имени или ip адресу машины, на котором он установлен. Для этого в диспетчере конфигурации SQL Server нужно включить соответствующие протоколы (Рисунок 2.6).

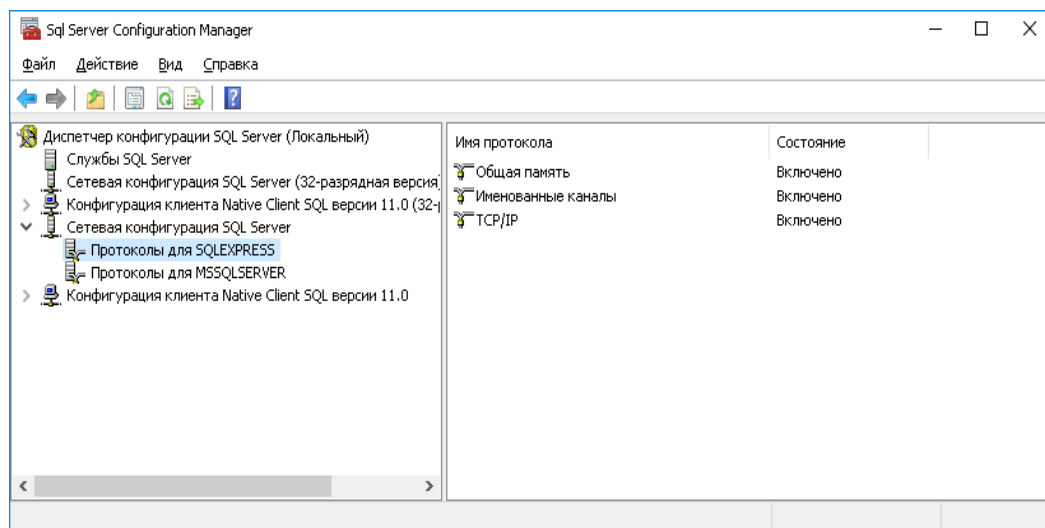


Рисунок 2.6 – Установка Microsoft SQL Server Management Studio

На этом установка и настройка SQL сервера завершена.

2.3.2 Проектирование БД

В БД необходимо хранить структурированную по группам информацию об учетных данных пользователей и различные административные учетные данные. В процессе работы получена схема БД (Рисунок 2.7).

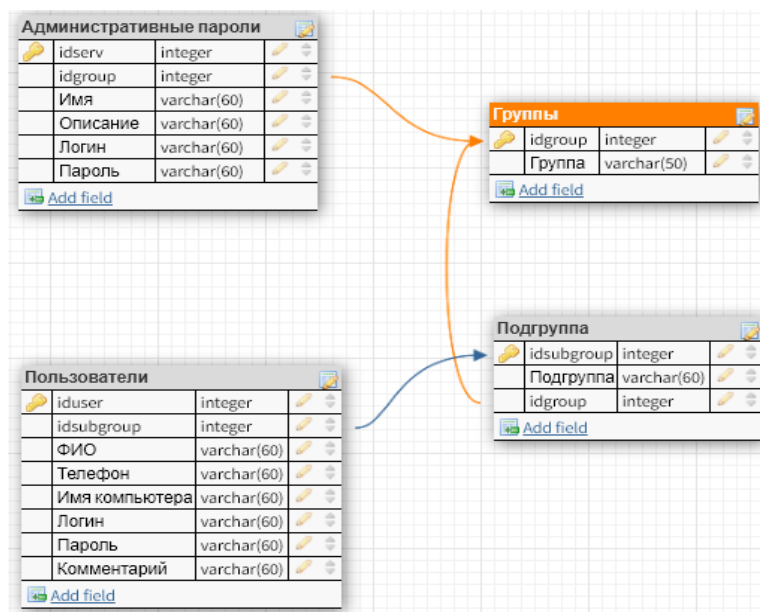


Рисунок 2.7 – Схема БД

В Microsoft SQL Server Management Studio проекты таблиц выглядят следующим образом:

- Таблица административных паролей (Рисунок 2.8).
- Таблица групп (Рисунок 2.9).
- Таблица подгрупп (Рисунок 2.10).
- Таблица пользователей (Рисунок 2.11).

Имя столбца	Тип данных	Разрешить значения NULL
idserver	int	<input type="checkbox"/>
Имя	varchar(50)	<input type="checkbox"/>
Логин	varchar(50)	<input checked="" type="checkbox"/>
Пароль	varchar(50)	<input checked="" type="checkbox"/>
Описание	varchar(50)	<input checked="" type="checkbox"/>
idgroup	int	<input type="checkbox"/>

Рисунок 2.8 – Проект таблицы административных паролей

Имя столбца	Тип данных	Разрешить значения NULL
idgroup	int	<input type="checkbox"/>
Группа	varchar(50)	<input type="checkbox"/>

Рисунок 2.9 – Проект таблицы групп

Имя столбца	Тип данных	Разрешить значения NULL
idsubgroup	int	<input type="checkbox"/>
Подгруппа	varchar(50)	<input type="checkbox"/>
idgroup	int	<input type="checkbox"/>

Рисунок 2.10 – Проект таблицы подгрупп

Имя столбца	Тип данных	Разрешить значения NULL
iduser	int	<input type="checkbox"/>
idsubgroup	int	<input type="checkbox"/>
ФИО	varchar(50)	<input type="checkbox"/>
Телефон	varchar(50)	<input checked="" type="checkbox"/>
[Имя компьютера]	varchar(50)	<input checked="" type="checkbox"/>
Логин	varchar(50)	<input checked="" type="checkbox"/>
Пароль	varchar(50)	<input checked="" type="checkbox"/>
Комментарий	varchar(50)	<input checked="" type="checkbox"/>

Рисунок 2.11 – Проект таблицы пользователей

2.3.3 Разработка пользовательского интерфейса. Руководство пользователя

Основным требованием к пользовательскому интерфейсу является простота и удобство использования. В качестве основы был выбран стиль оформления Windows Metro впервые появившийся в Windows 8.

При запуске программы открывается форма авторизации пользователя (Рисунок 2.12), где от пользователя требуется выбрать метод авторизации и сервер. При успешной авторизации пользователь перейдет к основной форме программы (Рисунок 2.13), а в случае недоступности сервера получит соответствующее сообщение (Рисунок 2.14).

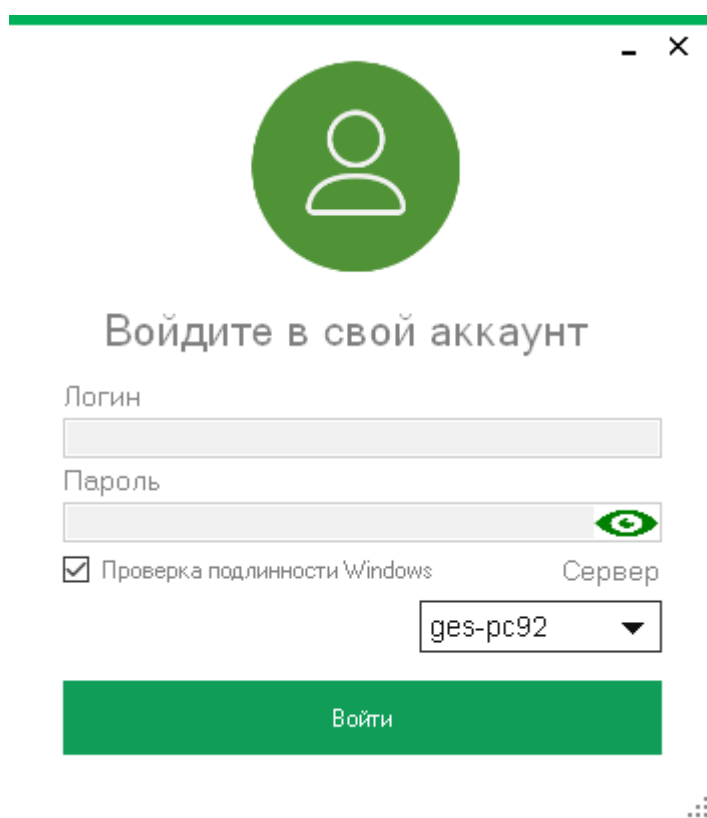


Рисунок 2.12 – Форма авторизации

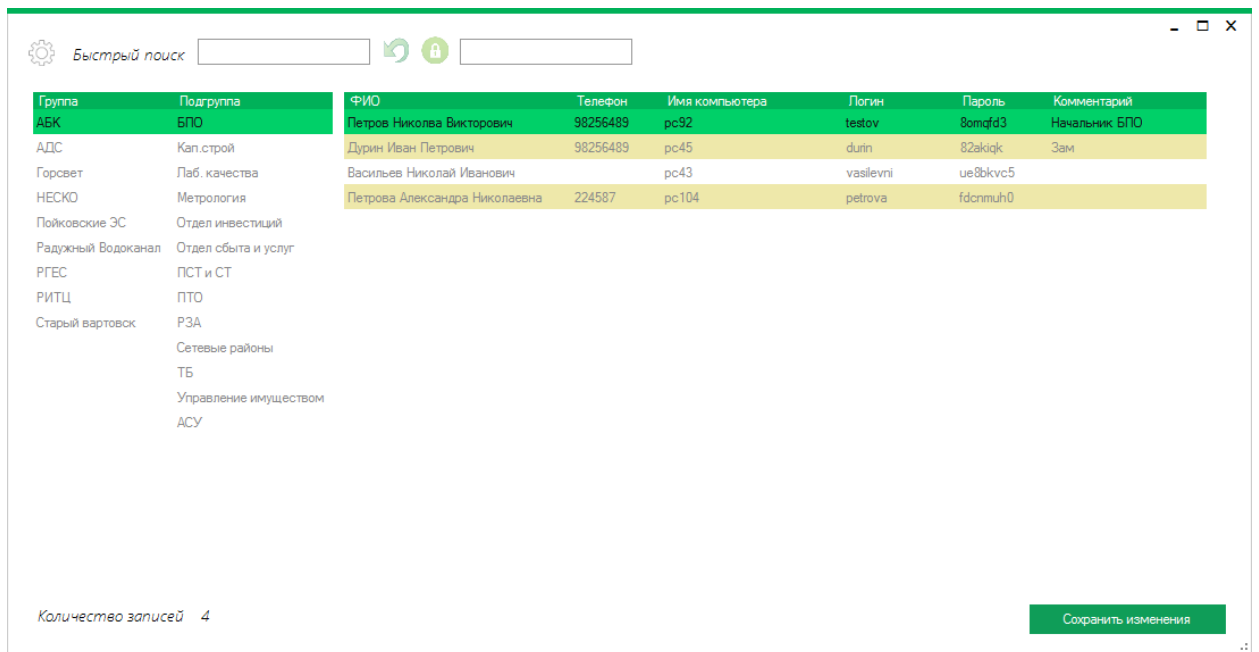


Рисунок 2.13 – Главная форма программы

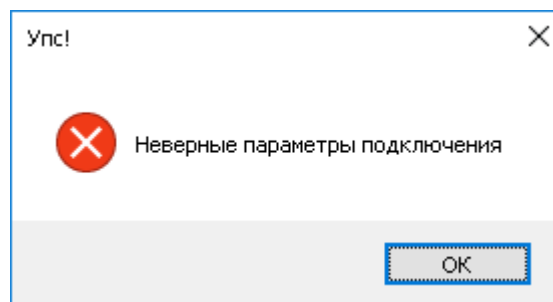


Рисунок 2.14 – Сообщение при ошибке подключения

Находясь на главной форме, пользователь имеет возможность производить поиск, редактировать, добавлять и удалять записи. Все записи строго группированы в соответствии с внутренними структурами, соответственно при клике на необходимую группу или подгруппу форма обновит отображаемые данные (Рисунок 2.15). В верхней части главной формы располагается функциональное меню (Рисунок 2.16) на котором распложены три кнопки и два текстовых поля.

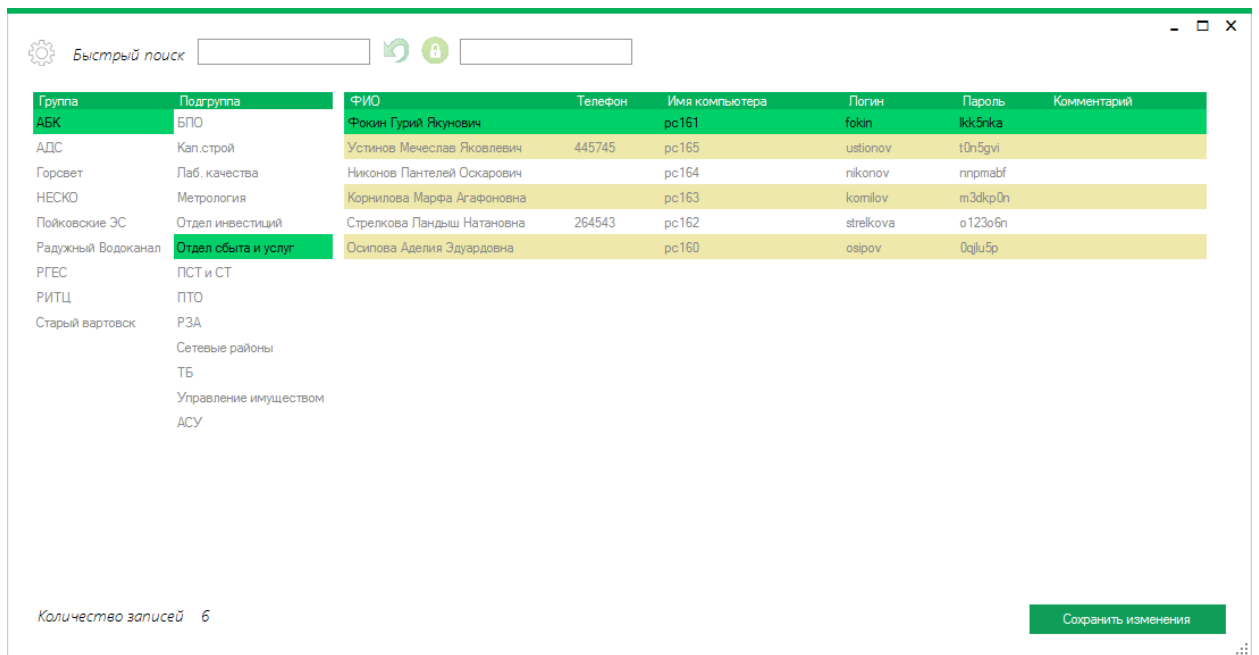


Рисунок 2.15 – Смена подгруппы



Рисунок 2.16 – Меню

При вводе текста в поле быстрого поиска форма обновит вид и произведет поиск по введенному пользователем тексту (Рисунок 2.17).

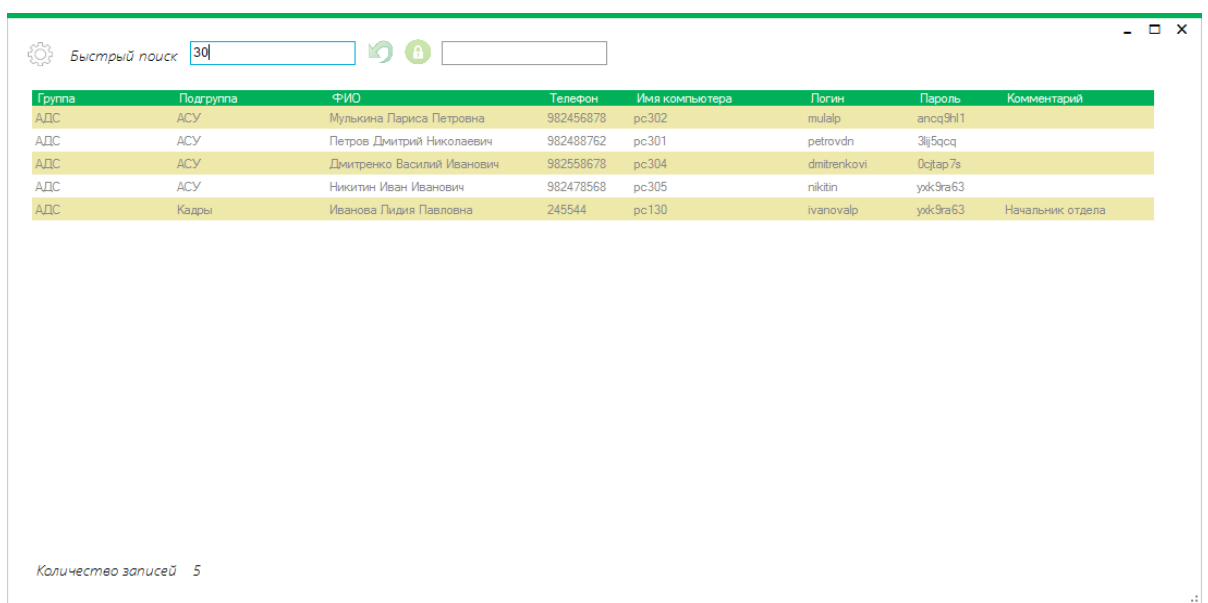


Рисунок 2.17 – Результат поиска

Для возврата к основному виду формы необходимо нажать на кнопку рядом с полем ввода (Рисунок 2.18)



Рисунок 2.18 – Кнопка возврата

Кнопка с замком (Рисунок 2.19) предназначена для быстрой генерации пароля (Рисунок 2.20), полученный пароль автоматически копируется в буфер обмена.



Рисунок 2.19 – Кнопка быстрой генерации пароля

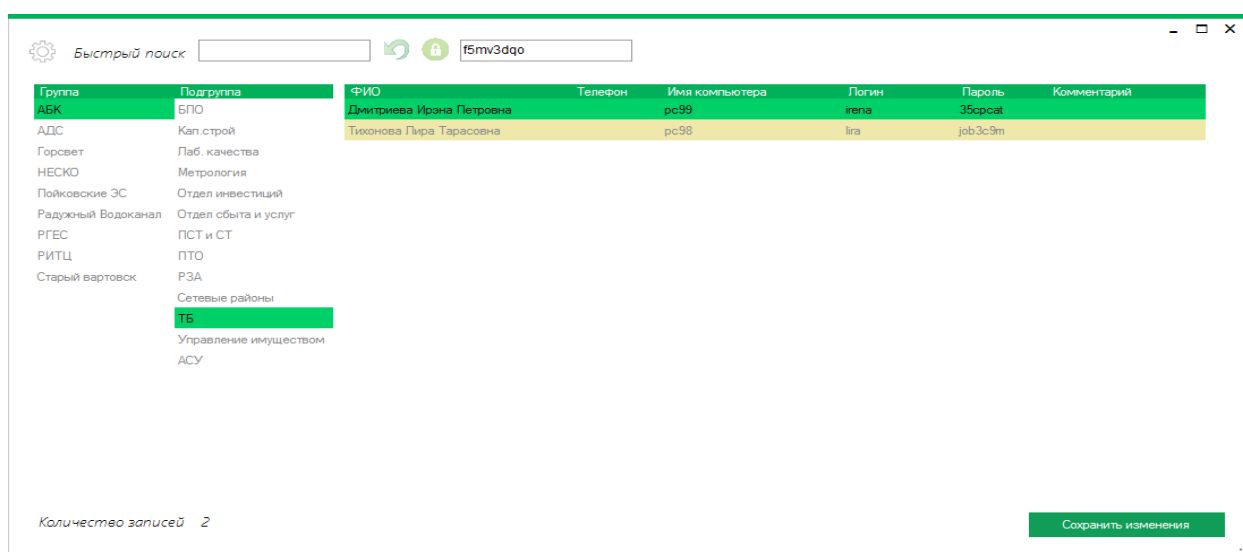


Рисунок 2.20 – Пример сгенерированного пароля

Для добавления записи достаточно кликнуть на новую строку и начать вводить текст (Рисунок 2.21), для редактирования нужно выбрать необходимое поле и начать менять текст, а для удаления нажать на правую кнопку мыши и выбрать соответствующий пункт контекстного меню (Рисунок 2.22).

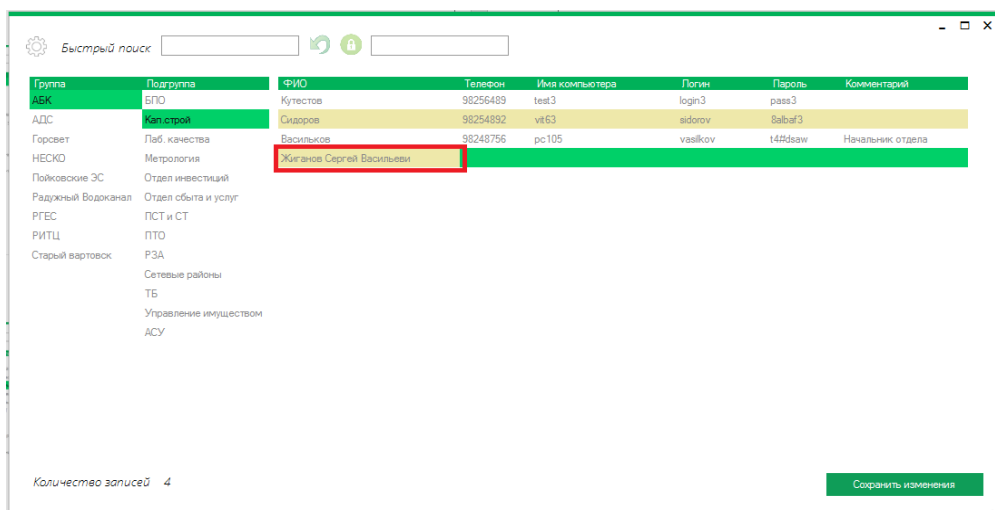


Рисунок 2.21 – Добавление записи

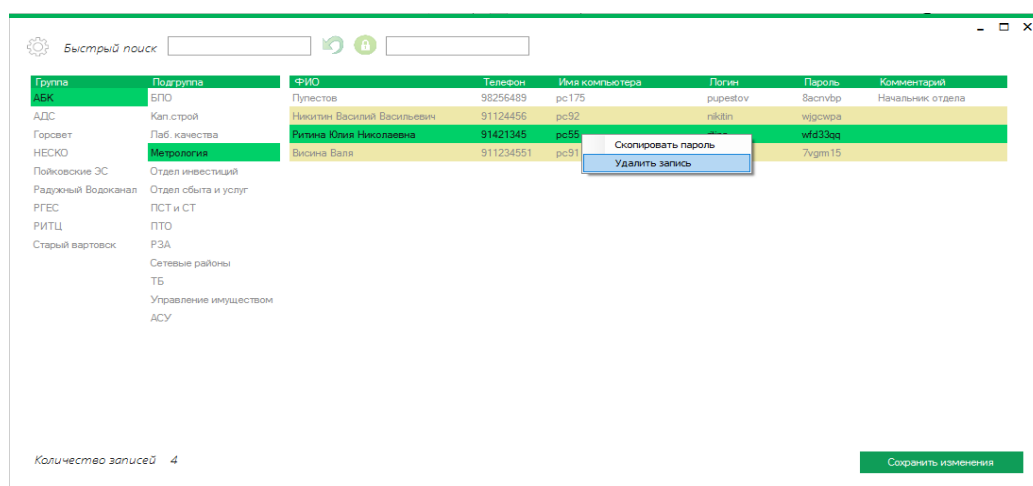


Рисунок 2.22 – Удаление записи

Для применения любых изменений необходимо нажать кнопку сохранения (Рисунок 2.23).

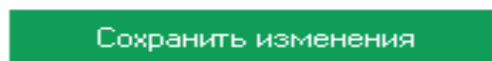


Рисунок 2.23 – Сохранение изменений

Для переключения к отображению административных паролей либо для открытия формы генератора паролей необходимо вызвать меню (Рисунок 2.24) нажав на соответствующую кнопку (Рисунок 2.25).



Рисунок 2.24 – Кнопка вызова меню

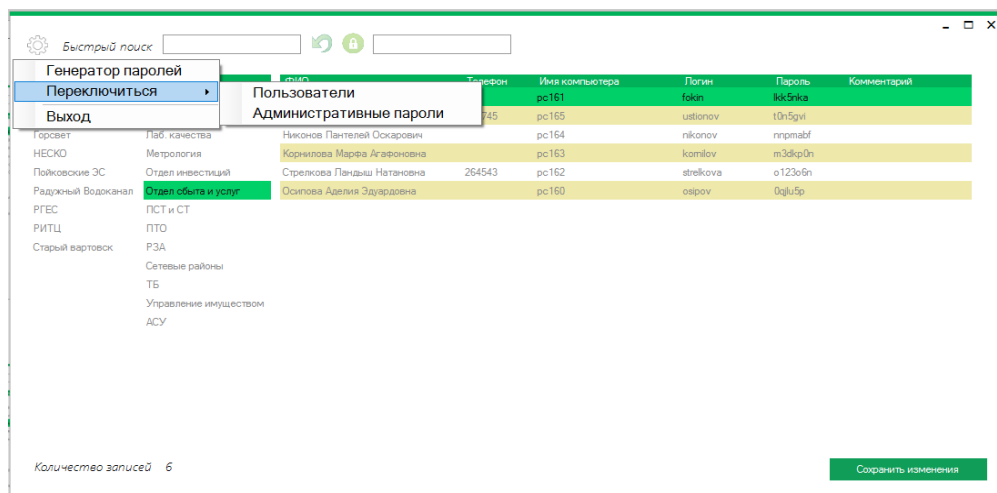


Рисунок 2.25 – Меню переключения

При вызове формы генератора паролей откроется форма (Рисунок 2.26) в которой можно сгенерировать пароли (Рисунок 2.28) заданной сложности нажав соответствующую кнопку (Рисунок 2.27).

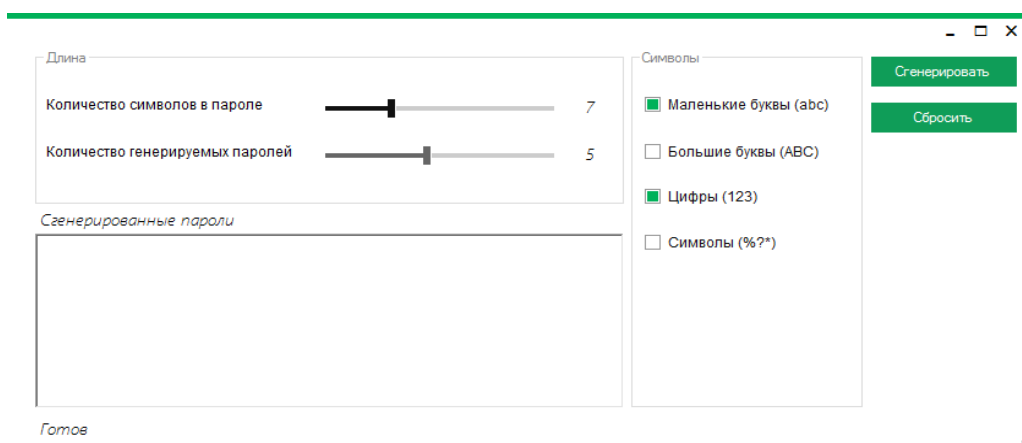


Рисунок 2.26 – Форма генератора паролей



Рисунок 2.27 – Кнопка для генерации паролей

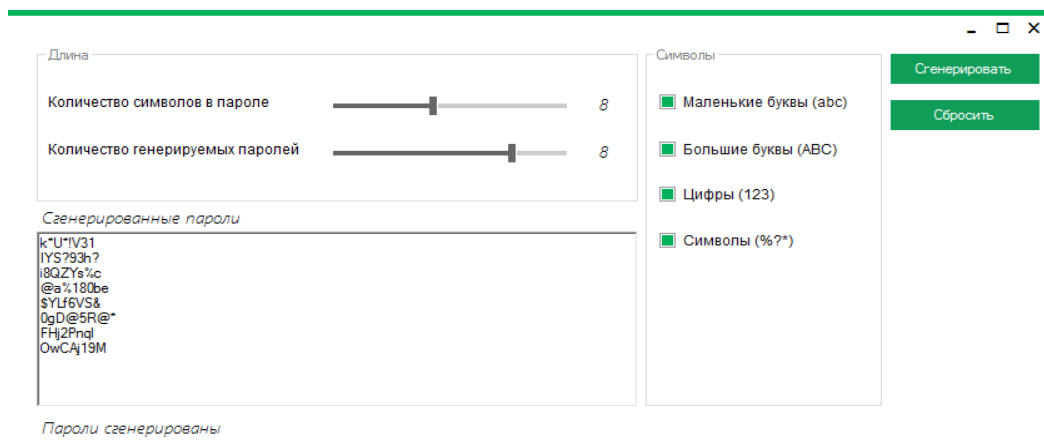


Рисунок 2.28 – Пример сгенерированных паролей

При переключении к отображению административных паролей главная форма меняет вид (Рисунок 2.29).

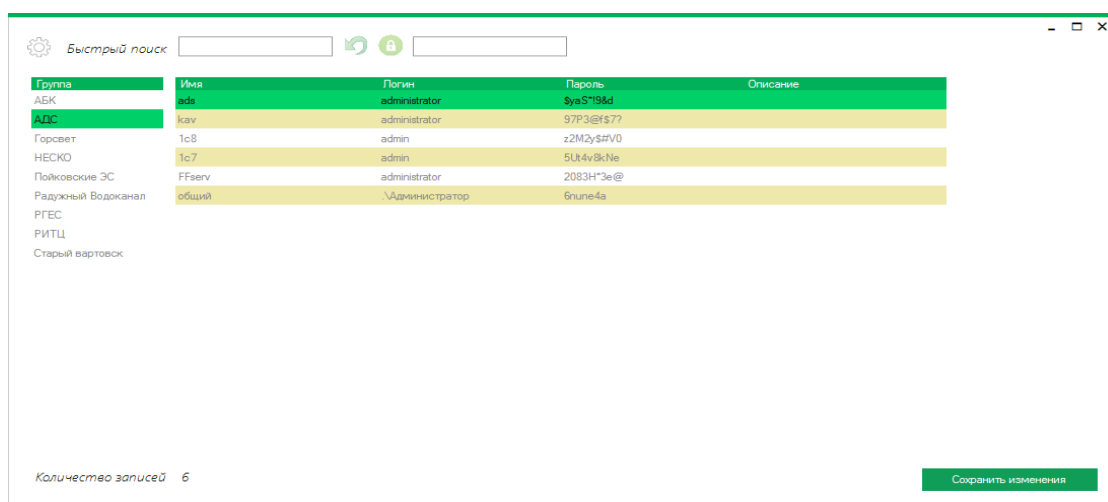


Рисунок 2.29 – Административные пароли

Добавление, редактирование и удаление записей работает аналогично обычному пользовательскому режиму. Независимо от режима отображения внизу формы всегда можно увидеть информацию по количеству найденных записей (Рисунок 2.30)

Количество записей 6

Рисунок 2.30 – Количество записей

Выводы по разделу два:

Разработанный менеджер паролей выполняет все выделенные в первом разделе функции. СУБД, расположенная на сервере, решает сразу несколько проблем присущих аналогичным программным системам, рассмотренным ранее.

Для разработанной программы не будет существовать проблемы с одновременным многопользовательским доступом так как БД не находится в отдельном локальном файле. Так же не требуется использовать протоколы шифрования данных как в случае, если бы данные хранились на предоставляемом стороннем сервере, потому что доступ к СУБД осуществляется по безопасной внутренней сети организации.

Поставленные задачи выполнены в полном объёме.

3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

3.1 Описание проектного решения

Разработано приложение для хранения учетных данных пользователей и прочих административных данных, обеспечивающее многопользовательский доступ благодаря использованию СУБД.

3.2 Постановка задачи

Основной задачей раздела является определение величины затрат на проведение исследований, себестоимость для определения экономического эффекта от использования в общественном производстве основных и сопутствующих результатов, получаемых при решении поставленной технической задачи в данной выпускной квалификационной работе. Оценка эффективности принятого научно-технического решения должна учитывать все необходимые расходы и затраты, для этого требуется провести ряд необходимых расчетов по определенной схеме.

Основной частью показателей экономической эффективности является расчет сметы затрат на разработку программного комплекса, включающую в себя:

- Затраты на аппаратное и программное обеспечение.
- Расчет затрат на электроэнергию.
- Расчет затрат на амортизацию оборудования.
- Расчет затрат на заработную плату.
- Расчет затрат на дополнительную заработную плату.
- Отчисления на социальные нужды.
- Общая смета затрат на внедрение системы.

3.3 Расчет сметы затрат на разработку программного комплекса

Таблица 3.1 – Стоимость аппаратного и программного обеспечения

Наименование	Количество	Цена рублей
Компьютер	1	34700
Microsoft Visual Studio	1	0
Microsoft SQL Server 2017 Express	1	0
Microsoft SQL Server Management Studio	1	0
ИТОГО:		34700

3.4 Расчет сметы затрат на разработку программного комплекса

Затраты на электроэнергию находятся исходя из продолжительности периода разработки ПО, количества кВт/ч, затраченных на проектирование программного обеспечения и тарифа за 1 кВт/ч Тариф по городу Нижневартовску составляет 2,87 руб. за кВт/ч. [1]

Таблица 3.2 – Расчет затрат на электроэнергию

Наименование	Установленная мощность, кВт	Стоимость 1кВт в час (руб)	Количество часов работы	Общая стоимость в руб
Компьютер	0,22	2,87	135	85,2
Освещение	0,1			38,7
ИТОГО:				123,9

3.5 Расчет затрат на амортизацию оборудования

Затраты на амортизацию оборудования проводятся за период их использования, т.е. за период внедрения и создания программного обеспечения.

Денежное выражение амортизации является амортизационным отчислением, которое входит в текущие затраты.

Величина амортизационных отчислений определяется на основе норм амортизации.

Норма амортизации – это установленный размер амортизационных отчислений на полное восстановление, выраженное в %. Норма амортизации устанавливается на основе экономически целесообразного срока службы и должна обеспечить возмещение износа основных средств к моменту возможного их морального и физического износа и создать экономическую основу для замены.

Амортизационные отчисления, приходящиеся на 1 час работы системы, рассчитываются по формуле (1):

$$A_{\text{ч}} = \Phi_{\text{перв}} \times a / F_{\text{д}}, \quad (1)$$

где $\Phi_{\text{перв}}$ – первоначальная стоимость системы или отдельных элементов;

a – норма амортизации (0.2);

$F_{\text{д}}$ – фонд времени работы за год (1970 часов).

Таблица 3.3 – Расчет амортизационных отчислений

Элемент КТС	Φ перв	$F_{\text{д}}$	$A_{\text{ч}}$	Количество часов работы	Общая стоимость (руб.)
Компьютер	34700	1970	2,152	135	475,58
ИТОГО:					475,58

Просуммировав расчет, мы получили расходы на приобретение, содержание и эксплуатацию программного и аппаратного обеспечения равные 35299,48 рублей

3.6 Расчет затрат на заработную плату

Для расчета расходов на заработанную платы необходимо умножить среднюю часовую ставку программиста на трудоемкость работы, чел/час по каждому из этапов разработки системы.

Средняя часовая ставка взята по формуле (2):

$$З_{ч} = \frac{З_{м}}{168}, \quad (2)$$

где $З_{ч}$ – средняя часовая ставка программиста;

$З_{м}$ – средняя месячная ставка техника-программиста (43000 рублей).

$$З_{ч} = 43000 / 168 = 255,95 \text{ рублей.}$$

Таблица 3.4 – Расчет затрат на заработную плату

Содержание работы	Трудоемкость работы, чел /час	Основная заработная плата (руб.)
Анализ предметной области	4	1023,8
Анализ существующих решений	8	2047,6
Выбор СУБД	8	2047,6
Выбор языка программирования	3	767,85
Проектирование	28	7166,6
Разработка	60	15357
Тестирование (отладка)	24	6142,8
ИТОГО	135	34553,25

3.7 Расчет затрат на дополнительную заработную плату

Дополнительную заработную плату разработчиков определяют в процентах от итоговой суммы основной заработной платы (15 %).

$$ЗП \text{ доп.} = 34553,25 \times 0,15 = 5182,98 \text{ (руб.)}$$

3.7 Отчисления на социальные нужды

Единый социальный налог рассчитывают в процентах от суммы основной и дополнительной заработных плат, в пенсионный фонд, в фонд социального и мед. страхования. На 2019 год данный процент составляет 30%, рассчитывается по формуле (3):

$$ЕСН=30\% \times (ЗП_{\text{осн}} + ЗП_{\text{доп}}), \quad (3)$$

где ЕСН – единый социальный налог;

ЗП_{осн} – основная заработная плата;

ЗП_{доп} – дополнительная заработная плата.

$$ЕСН = 0,30 \times (34553,25+5182,98) =11920,86 \text{ (руб.)}$$

3.8 Общая смета затрат на внедрение системы

Таблица 3.5 – Смета затрат

п/п	Элементы затрат	Сумма, руб.
1	Приобретение, содержание и эксплуатация программного и аппаратного обеспечения	35299,48
2	Затраты на основную заработную плату	34553,25
3	Затраты на дополнительную заработную плату	5182,98
4	Отчисления на социальные нужды	11920,86
ИТОГО:		86956,57

Выводы по разделу три:

В данном разделе было проведено технико-экономическое обоснование данной системы. В ходе обоснования были рассчитаны следующие затраты:

- Затраты на аппаратное и программное обеспечение.
- Затраты на амортизацию оборудования.
- Затраты на заработную плату.
- Затраты на дополнительную заработную плату.
- Отчисления на социальные нужды.
- Общая смета затрат на внедрение системы.

Общие затраты на внедрение системы составляют 86956,57 руб.

4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

4.1 Характеристика вредных факторов при работе с вычислительной техникой при написании выпускной квалификационной работы

Работа с вычислительной техникой (далее – ВТ) относится к категории работ, связанных с опасными и вредными условиями труда. На оператора может воздействовать ряд опасных и вредных производственных факторов, наиболее значимые из которых следующие:

1. Повышенный уровень напряжения в электрических цепях питания и управления персональным компьютером (далее – ПК), который может привести к электротравме оператора при отсутствии заземления оборудования (источник – переменный ток промышленной частоты 50Гц напряжением 220В, служащий для питания ПК, а также токи высокой частоты напряжением до 1200В систем питания отдельных схем и узлов дисплея).

2. Не соответствующие нормам параметры микроклимата: повышенная температура из-за постоянного нагрева деталей ПК, пониженная влажность.

3. Нарушение норм по аэроионному составу воздуха, особенно в помещениях с разной системой приточно-вытяжной вентиляции и (или) с кондиционерами, при этом концентрация полезных для организма отрицательно заряженных легких ионов кислорода воздуха (аэроионов) может быть в 10-50 раз ниже нормы, а концентрация вредных положительных ионов значительно превышать норму.

4. Пониженный или повышенный уровень освещенности в помещении; не соответствующие санитарным нормам визуальные параметры дисплея.

5. Повышенный уровень шума в системном блоке компьютера являются:

- вентилятор процессора;
- вентилятор видеокарты;
- вентилятор блока питания;
- вентиляторы системного блока;

- жесткий диск;
- оптические приводы;

6. Повышенный уровень загазованности воздуха (в первую очередь по углекислому газу и аммиаку, которые образуются при выдыхании); повышенное содержание в воздухе патогенной (вызывающей заболевания) микрофлоры (прежде всего – стафилококка) особенно зимой при повышенной температуре в помещении, плохом проветривании, пониженной влажности, нарушении аэроионного состава воздуха.

7. Психофизиологическая напряженность труда, включающая:

- монотонность труда;
- повышенное умственное напряжение из-за большого объема перерабатываемой и усваиваемой информации;
- повышенное нервно-эмоциональное напряжение (при этом ускоряется вывод из организма многих жизненно важных и необходимых витаминов и микроэлементов);
- наличие в воздухе рабочей зоны вредных веществ (окиси углерода, озона, аммиака, окислов азота, серы и т.п.), солей тяжелых металлов и органических соединений (фенола, бензопорена, формальдегида, полихлорированных бензенилов, свободных радикалов и др.);
- резкое ухудшение качества воздуха по аэроионному составу, увеличения в нем содержания различных аллергенов, грибков, вирусов, бактерий, микроорганизмов; рост информационных нагрузок извне (причем не только во время работы на ПК).

Большинство проблем, связанных с пагубным влиянием передовых технологий решаются при правильной организации рабочего места, соблюдении правил техники безопасности и разумном распределении рабочего времени.

4.2 Организация рабочего места

В соответствии с СанПиН 2.2.2/2.4.1340-03 [2].

Рабочее место – часть пространства, предназначенная для выполнения работником производственной деятельности, оснащенная ВТ.

Рабочее помещение должно хорошо проветриваться и быть достаточно просторным для организации рабочего процесса. Минимально допустимая площадь, выделенная на одно рабочее место, должна быть не менее 6 м^2 для мониторов с ЭЛТ и $4,5\text{ м}^2$ для жидко кристаллических мониторов, минимальный объем – 20 м^3 (для учебных заведений – 15 м^3). Расстояние между рабочими местами не менее 2 м, а расстояние между боковыми поверхностями мониторов – не менее 1,2 м.

Не менее важную роль играет планировка на рабочем месте, необходимо учитывать зоны досягаемости рук при расположении дисплеев, клавиатуры ЭВМ. Эти зоны устанавливаются на основании антропологических данных человеческого тела.

Экран видеомонитора не должен находиться на расстоянии ближе, чем 500 мм от глаз пользователя, рекомендуется 600 - 700 мм.

Конструкция рабочего кресла (стула) должна гарантировать сохранение рациональной рабочей позы при работе на персональных электронно-вычислительных машинах (далее – ПЭВМ), а также позволять менять позу для снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления.

Рабочее кресло (стул) должно быть регулируемым по высоте и углам наклона спинки, подъемно-поворотным, регулировка каждого параметра должна быть независимой и надежно фиксируемой.

Поверхности элементов кресла (стула) должны быть полумягкими, с нескользящими, слабо электризующимися и воздухопроницаемыми покрытиями, обеспечивающими легкую очистку от загрязнений.

4.3 Санитарно-гигиенические параметры рабочих мест

4.3.1 Микроклиматические условия

В помещениях, где работа на ПЭВМ является основной, необходимо обеспечивать оптимальные параметры микроклимата, приведенные в таблице 4.1

Микроклимат помещения оказывает как положительное, так и отрицательное влияние на работников. Расхождение в отдельных параметрах микроклимата от рекомендованных значений значительно сказывается на снижении работоспособности сотрудника, ухудшают его личное самочувствие и могут спровоцировать проявление заболеваний профессионального характера.

При пониженной влажности воздуха (менее 25-30%), в верхних дыхательных путях у человека появляется ощущение сухости слизистых оболочек, что так же значительно сказывается на ухудшении его самочувствия и снижении работоспособности. В создание микроклимата рабочего места скорость движения воздуха играет значительную роль. Человек начинает ощущать движение воздуха при скорости примерно 0,15 м/с. При этом действие воздушного потока зависит от его температуры. При температуре мене 36°C поток оказывает на человека освежающее действие, а при температуре более 40°C – неблагоприятное.

Для того, чтобы повысить влажность воздуха в помещениях необходимо использовать увлажнители воздуха, ежедневно наполняемые питьевой или дистиллированной водой.

Таблица 4.1 – Оптимальные нормы микроклимата для помещений с ПЭВМ

Период года	Категория работы	Температура воздуха, С0, не более	Относительная влажность воздуха %	Скорость движения воздуха, м/с
Холодный	Легкая – 1а	22-24	40-60	0,1
	Легкая – 1б	21-23	40-60	0,1
Теплый	Легкая – 1а	23-25	40-60	0,1
	Легкая – 1б	22-24	40-60	0,2

К категории 1а относятся работы, выполняемые в положении сидя, не

требующие физического напряжения и при которых расход энергии будет составлять до 120 ккал/ч.

К категории 1б относятся работы, выполняемые в положении сидя, стоя или связанные с ходьбой и сопровождающиеся некоторым физическим напряжением, при которых расход энергии будет составлять от 120 до 150 ккал/ч.

Под оптимальными параметрами микроклимата принято понимать такие, которые при длительном и систематическом воздействии на человека обеспечивают сохранение нормального функционального и теплового состояния организма без напряжения реакций терморегуляции, создают ощущение теплового комфорта и являются предпосылкой высокого уровня работоспособности.

Допускается влагосодержание не ниже 6 г/м, оптимальное влагосодержание составляет 10%.

4.3.2 Защита от шума

Шум – любой неблагоприятно воспринимаемый звук. Чрезмерный шум, уровень которого противоречит существующим нормам санитарно-эпидемиологических правил, оказывает вредное влияние на организм человека: способствует развитию глухоты и тугоухости, является чрезмерным возбудителем нервной системы, вследствие чего повышается пульс, учащается дыхание, и изменяется кровяного давление, что способствует появлению головных болей, мигрени, а также может вызывать бессонницу.

Источниками шума при работе с ПЭВМ являются: кулеры, охлаждающие процессор, блок питания и графическую карту; приводы оптических и жестких дисков. В результате генерируется широкий спектр звуков (включая ультразвук), причем каждый компьютер отличается в этом смысле своей индивидуальностью. Корпус компьютера при этом играет роль резонатора: он привносит в общую картину шума низкочастотные составляющие.

Предельный допустимый уровень звука в соответствии с выполняемой работой не должен превышать 50 дБА (СанПиН 2.2.4.3359-16 [3]).

В помещениях, где работают инженерно-технические работники, осуществляется лабораторный, аналитический или измерительный контроль, уровень шума не должен превышать 60 дБА.

В помещениях операторов ЭВМ (без дисплеев) уровень шума не должен превышать 65 дБА.

На рабочих местах в помещениях для размещения шумных агрегатов вычислительных машин уровень шума не должен превышать 75 дБА. Снизить уровень шума в помещениях можно использованием звукопоглощающих материалов. Дополнительным звукопоглощением могут служить однотонные занавеси из плотной ткани, гармонирующие с окраской стен и подвешенные в складку на расстоянии 15-20 см от ограждения. Ширина занавеси должна быть в два раза больше ширины окна. А снизить уровень шума самого компьютера можно при правильном и качественном выборе комплектующих – вентиляторов, кулера, блока питания, звукоизолирующего бокса для подавления шума винчестера и так далее.

Уровень шума, возникающий от нескольких, одновременно работающих, некогерентных источников рассчитывается путем суммирования излучений отдельных источников, по формуле (4):

$$L_{\Sigma} = 10 \times \lg \sum_{t=1}^{i=n} 10^{0,1L_i} \quad (4)$$

где L_i – уровень звукового давления i -го источника шума;

n – количество источников шума.

Результаты расчета проходят сравнение с допустимым значением уровня шума для рабочего места. В том случае если результат превысил допустимое значение необходимо применить меры по снижению шума, которые были описаны ранее в данном разделе.

Для расчетов возьмем конфигурацию компьютера, на котором выполнялась выпускная квалификационная работа:

- корпус DEERCOOL MatreX 30;
- один вентилятора 120 мм DEERCOOL UF 120;
- процессор intel core i3 8100;
- кулер для процессора DEERCOOL GAMMAXX 200T;
- материнская плата Gigabyte B360 M H;
- модуль оперативной памяти Kingston HyperX Fury DDR4 DIMM, 4GB, 2400MHz, CL15, HX424C15FR2/8 (2 шт.);
- твердотельный накопитель 256 Гб A-Data XPG GAMMIX S11;
- блок питания KCAS PLUS 400W;

Из данной конфигурации выделим компоненты способные создавать шум и запишем их уровни звукового давления в таблицу 4.2. Стоит отметить, что используются дБА (акустический децибел, как правило, ниже фактического значения на 10 дБ), а не дБ, в силу того, что именно этот параметр дается для аппаратуры. Поскольку не учитывается расстояние от программиста до системного блока, то более низкое значение дБА будет оптимальным для расчетов.

Таблица 4.2 – Уровни звукового давления источников шума на рабочем месте

Источник шума	Уровень шума, дБА
Вентилятор блока питания	36
Вентилятор корпуса	26
Вентилятор процессора	36

Подставив значения уровня звукового давления для каждого вида оборудования в формулу, получим:

$$L_{\Sigma}=10 \times \lg(10^{3,6}+10^{2,6}+10^{3,6}) \approx 39,22 \text{ дБА}$$

Так же следует понимать, что для расчета были взяты предельно возможные значения, которые способны произвести конкретные вентиляторы. ПЭВМ

регулирует обороты вращения вентиляторов автоматически в зависимости от нагрузки поэтому действительное значение уровня звукового давления будет ниже так как они никогда не работают в пиковой нагрузке.

Полученное значение не превышает допустимый уровень шума для рабочего места программиста (50 дБ) (ГОСТ 12.1.003-83 [4]).

4.3.3 Освещение

Освещение рабочего места – один из важнейших факторов, участвующих в создании комфортных и безопасных условий труда. Правильно спроектированное и выполненное освещение может обеспечить высокий уровень работоспособности сотрудников, оказывать благоприятное воздействие на психику работающего, способствовать повышению производительности труда. Первостепенной важностью освещения является тот факт, что деятельность операторов связана с преобладанием зрительной информации – свыше 80% от общего объема информации. Освещение в учреждениях – 300-500 лк., является допустимым значением освещенности. Исходя из требований отсутствия бликов и равномерности освещения всего помещения, в организации преобладает искусственное освещение общего типа. Система общего искусственного освещения в организации представляет собой потолочные или подвесные лампы, размещенные параллельно друг другу и равномерно по потолку.

В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы и компактные люминесцентные лампы. При устройстве отраженного освещения в производственных и административно-общественных помещениях допускается применение металлогалогенных ламп. В светильниках местного освещения допускается применение ламп накаливания, в т.ч. галогенных.

Для освещения помещений с ПЭВМ следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными

пускорегулирующими аппаратами (далее – ЭПРА). Допускается использование многоламповых светильников с ЭПРА, состоящими из равного числа опережающих и отстающих ветвей.

Для обеспечения нормируемых значений освещенности в помещениях для использования ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

4.3.4 Действие электромагнитного излучения

В настоящее время широкое распространение получили беспроводные сети доступа. Беспроводные точки доступа имеются практически в каждом крупном офисе, торговом центре, а также в квартирах людей. Одним из главных преимуществ беспроводных сетей является отсутствие необходимости прокладывать проводные линии связи.

Однако беспроводные сети работают в диапазоне СВЧ 2,4 ГГц; 5 ГГц. На частоте 2,4 ГГц работают микроволновые печи. Влияние на окружающую среду офисных точек доступа минимальны, так как мощность передатчиков в беспроводной сети небольшая порядка 100 мВт. Для сетей, подлежащих лицензированию мощность передатчиков несколько больше.

Так же необходимо учитывать тот факт, что мощность электромагнитного излучения падает обратно пропорционально квадрату расстояния до источника. Стоит принять в расчёт также время воздействия излучения. Для объективной оценки вреда электромагнитного излучения для здоровья, нужно опираться на все четыре параметра: частота, мощность, расстояние и время.

Воздействию электромагнитных излучений (далее - ЭМИ), создаваемых системами беспроводной связи, особо подвержены группы лиц, чья работа непосредственно связана с источниками ЭМИ, население, проживающее в непосредственной близости от вышек связи.

Для обеспечения безопасности работ с источниками ЭМИ должен производиться систематический контроль фактических значений нормативных параметров на рабочих местах и в местах возможного нахождения рабочего персонала.

4.4 Требования к режиму труда и охраны

При работе с ПК необходимо соблюдать режим труда и отдыха в зависимости от вида и категории трудовой деятельности, во избежание накопления усталости при неподвижной работе и постоянном напряжении глаз.

Виды трудовой деятельности разделяются на 3 группы:

- группа А – считывание информации с экранов дисплеев;
- группа Б – ввод информации;
- группа В – творческая работа в режиме диалога с ПК.

Для каждого вида устанавливается три категории тяжести и напряженности работы с ПК. Для каждой категории устанавливаются свои нормы на уровень нагрузки и время перерывов.

Уровень нагрузки и время рекомендуемых регламентированных перерывов для каждой группы и каждой категории приведены в таблице 4.3.

Таблица 4.3 – Режим работы оператора ПК

Категория работы с ПК	Уровень нагрузки за рабочую смену			Суммарное время перерывов, мин	
	Группа А, кол. знаков	Группа Б, кол. знаков	Группа В, час	При 8-час. смене	При 12-час. смене
I	20 тыс.	15 тыс.	2	50	80
II	40 тыс.	30 тыс.	4	70	110
III	60 тыс.	40 тыс.	6	90	140

4.5 Влияние статических и зрительных нагрузок, меры защиты

4.5.1 Влияние статических нагрузок и их профилактика

Характерной особенностью работой за ПК является статический режим: большой объем работы приходится выполнять в сидячем положении. При этом большинство групп мышц находятся в напряжении, что приводит к быстрой утомляемости, способствует развитию профессиональных патологических изгибов позвоночника. Неправильное расположение дисплеев по высоте – слишком низкое, под неправильным углом является основной причиной появления сутулости; слишком высокое положение дисплея приводит к длительному напряжению шейного отдела позвоночника может привести к развитию остеохондроза.

Интенсивное и продолжительное задействование клавиатуры при работе на компьютере может стать источником тяжелых профессиональных заболеваний рук. Комплекс этих заболеваний, получивших общее название «травмы повторяющихся нагрузок», включает такие заболевания, как тендинит, травматический эпикондилит, синдром канала запястья.

Травмы повторяющихся нагрузок представляют собой травму накапливающихся недомоганий. Легкая боль в руке, если ее не вылечить, сможет привести к полной инвалидности. Симптом заболевания – становится трудно самому закинуть небольшой груз на полку, уровень которой ненамного превышает ваш рост. Обычно начинает болеть правая рука, поскольку на нее ложится большая нагрузка, в конце концов заболевает и левая.

Вообще использование манипулятора типа «мышь» опасно для здоровья, запястья, предплечья, плечи – все это из-за «мышь» – может серьезно пострадать. О «мышинных проблемах» начали говорить с конца 90-х годов. Самым известным заболеванием, связанным с использованием мыши, клавиатуры является карпальный туннельный синдром или его еще называют синдромом запястного

канала.

Жалобы при тунельном синдроме:

- появление неприятных ощущений в области запястья, ладони и пальцев рук (в проекции расхождения ветвей срединного нерва);
- со временем появляется ослабление пальцев и слабость пораженной ладони, онемение, боль и тяжесть в руке;
- болезненность и онемение ладоней заставляют просыпаться, т.к. во время сна человек не контролирует положение рук;
- неловкость ладони и пальцев создает затруднение при письме, а попытка поднять любой более-менее тяжелый предмет приводит к возникновению жгучей боли в запястном суставе.

Вообще человеческий организм всегда болезненно реагирует на длительные постоянные нагрузки. Давно известно, что у тех, кто ранее работал за пишущей машинкой, вручную занимается ковровой росписью или же расшивает узорами ткани, развиваются те же симптомы, что и у операторов, работающих с мышью.

Для снижения нагрузок на позвоночник необходимо правильно организовать рабочее место и, в первую очередь, подобрать качественную мебель.

4.5.2 Влияние зрительных нагрузок и их профилактика

В процессе работы оператору ПК приходится иметь дело с изображениями на экране монитора. Считывание текста, таблиц, графиков с экрана отличается от чтения той же информации с листа бумаги. Отличие в следующем:

- При работе с дисплеем пользователь целиком зависит от положения дисплея, тогда как при чтении печатной информации можно найти положения листа для наиболее комфортного восприятия информации.
- Экран, являясь источником света, считается прибором активного контраста, в то время как при чтении с листа бумаги мы имеем дело с отраженным светом, т.е. с пассивным контрастом, который в малой степени зависит от

интенсивности освещения и угла падения светового потока на бумагу.

– Текст на бумаге является неизменным, а текст на экране периодически обновляется в процессе сканирования электронного луча по поверхности экрана. Достаточно низкая частота обновления вызывает мерцание изображений.

– Монитор надолго приковывает к себе внимание оператора, что является причиной длительной неподвижности глазных и внутриглазных мышц, что приводит к их ослаблению.

– Длительная работа с ПК требует повышенной сосредоточенности, что приводит к большим нагрузкам на зрительную систему пользователя. Развивается зрительное утомление (зрительная астенопия), которое способствует возникновению близорукости, головной боли, раздражительности, нервного напряжения и стресса.

Приведенные выше особенности изображений на экране дисплея в значительной степени влияют на степень утомляемости зрительного аппарата. На утомляемость влияет также четкость и резкость изображения. Эти параметры зависят от разрешающей способности монитора, которая определяется числом дискретных элементов изображения, воспроизводимых монитором по горизонтали и вертикали. Чем выше разрешающая способность, тем четче и точнее изображение на экране.

Изображение на экране монитора обновляется с определенной частотой, называемой частотой смены кадров (или частотой вертикальной развертки). Если эта частота достаточно высока, изображение кажется неизменным несмотря на то, что фактическая яркость любой точки изображения – переменная величина. Эффект стабильного изображения создается как результат взаимодействия двух факторов: инерционности зрения человека и инерционности монитора.

Критическая частота, при которой изображение начинает восприниматься как неизменное, не является постоянной величиной. Эти колебания воспринимаются, как мерцание экрана.

Крайняя нижняя граница частоты смены кадров, определяется по методике

MPR (Manufacturing Resource Planning), равняется 76 Гц, для позитивного изображения (светлый фон, темный текст) и 67 Гц для негативного изображения (смешанный фон, разноконтрастный текст).

На утомление зрения влияет и яркость изображения на экране монитора. Необходимо учитывать также, что внешнее освещение в рабочем помещении в некотором смысле «конкурирует» со светом, излучаемый монитором, снижая контраст изображения и ухудшая условия работы. Хорошие показатели яркости могут помочь компенсировать этот фактор.

К факторам, очень сильно мешающим воспринимать информацию с экрана, следует отнести блики. Они заставляют неосознанно менять положение головы и корпуса, напрягать зрение, чтобы прочесть нужную информацию. При этом глаза испытывают дополнительную нагрузку, происходит увеличение нагрузки на шею, спину, плечи и руки, что приводит к быстрой утомляемости всего организма.

Блики создает любой пучок света, отраженный экраном дисплея и попавший на оболочку глаза. Их источниками могут быть расположенные напротив монитора яркие поверхности, светлое оборудование, осветительные приборы, не зашторенные окна, часто – светлая одежда оператора. Блики тем заметнее и тем сильнее снижают контрастность изображения, чем выше коэффициент зеркального отражения экрана. В современных мониторах, чтобы уменьшить отражения применяют: темное или тонированное стекло; цилиндрические (или вертикально-плоские экраны – ЭЛТ TRINITRON и DIAMOND-NRON) и плоские прямоугольные экраны (обладают лучшими антибликовыми свойствами в силу действия обычных законов отражения), а также используют защитные фильтры.

Проблемы снижения зрительного утомления решают с помощью применения специальных защитных средств, правильной организации рабочего места, режимов труда и отдыха, специальных упражнений для снятия утомления.

4.6 Пожарная безопасность

Пожар, возникший в организациях, может привести к выходу из строя аппаратуры.

Основными причинами возникновения пожара являются:

- небрежное обращение с открытым огнем;
- неисправность электрооборудования, электропроводки;
- самовозгорание горючих веществ;
- небрежное или неумелое обращение со сварочным аппаратом,

паяльником.

К инженерно-техническим мероприятиям по пожарной профилактике относятся:

- применение негорючих и огнестойких материалов;
- правильная планировка зданий, необходимое количество входов и

выходов.

Помещения организации должны быть оборудованы всем необходимым противопожарным инвентарем, углекислотными огнетушителями. Весь пожарный инвентарь должен находиться на видном месте и содержаться в исправном состоянии, к нему должен быть обеспечен беспрепятственный доступ.

Легковоспламеняющиеся жидкости должны находиться в закрытой металлической таре и храниться в специальном помещении. Организации должны оснащаться автоматической пожарной сигнализацией. Во всех пожароопасных помещениях должна быть предусмотрена пожарная сигнализация, построенная по лучевой схеме, с использованием извещателей «на дым».

В организациях должны быть размещены планы эвакуации людей при пожаре. Для тушения пожара применяются огнетушители, песок, вода и другие средства. Горючие жидкости следует тушить пенообразными составами, засыпать песком или землёй.

Пожарная безопасность должна обеспечиваться следующими проектными решениями:

- соблюдение требуемых пределов огнестойкости в соответствии со 2 степенью огнестойкости;
- облицовкой поверхностей стен несгораемыми или трудно-сгораемыми материалами в соответствии со СНиП 31-03-2001 [5];
- установкой противопожарных дверей в помещениях с категорией В, обеспечением габаритов проходов и дверей, в соответствии с противопожарными требованиями СНиП 31-03-2001, СНиП 2.01.02-85 и другими действующими нормативными документами [5, 6].

Противопожарные мероприятия в помещении ЭПУ обеспечиваются:

- выбором устройств защиты электрооборудования и токораспределительных участков;
- выбором соответствующих марок и сечений кабелей и способов их прокладки;
- применением несгораемых материалов для монтажа электрооборудования;
- наличие первичных средств пожаротушения;
- герметизация проходов.

Предотвращение пожара в организации достигается полным исключением образования опасной горючей среды и появления источников возгорания. В случае возникновения пожароопасной ситуации, для ее устранения в организации применяются следующие средства пожаротушений: огнетушители ручные и передвижные, пожарные гидранты, сухой песок. В организации применяются пенные и углекислотные ручные огнетушители.

Ручные огнетушители установлены в помещении из расчета 1 огнетушитель на 40-50 м² площади.

Обязанности и действия персонала при возникновении пожара.

При обнаружении пожара немедленно сообщить в пожарную охрану по телефону «01»:

- сообщить о пожаре по телефону 01;
- подготовить к эвакуации людей;
- эвакуировать людей;
- принять меры по тушению пожара или загорания до прибытия

пожарных частей, а также по возможности произвести эвакуацию материальных ценностей. При этом использовать средства противопожарной защиты, а также предварительно обесточить помещение;

При эвакуации из задымленных помещений, обезопасить органы дыхания (закрывать влажным полотенцем или платком рот и нос) и, пригнувшись ниже к полу, покинуть задымленное помещение.

По возможности, приступить к ликвидации пожара (огнетушителями и другими, имеющимися средствами пожаротушения) и эвакуации материальных ценностей.

Встретить прибывшие пожарные подразделения и проводить их к месту пожара, предупредить об особенностях тушения пожара в данном помещении.

Каждый работник предприятия обязан изучить и знать схему эвакуации при пожаре, уметь пользоваться огнетушителями.

Выводы по разделу четыре:

Во время написания выпускной квалификационной работы в данном разделе были рассмотрены характеристики вредных факторов при работе с вычислительной техникой:

- действие электромагнитного излучения;
- повышенный уровень шума в системном блоке компьютера;
- пониженный или повышенный уровень освещенности в помещении;
- микроклиматические условия;

Рассмотрены нормы режима труда, рассмотрено влияние статических и зрительных нагрузок и меры защиты от них. Произведен расчет уровня звукового давления на рабочем месте написания работы, а также рассмотрены вопросы пожарной безопасности.

ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе были проанализированы программные системы хранения паролей, рассмотренные системы не подходят для использования в организациях среднего или большего размеров, так как имеют существенные недостатки. Проанализировав их функционал, были выявлены наиболее важные критерии, таким образом, была обоснована необходимость разработки собственного решения создания удобного менеджера паролей рассчитанного на хранения большого количества учетных данных и одновременную многопользовательскую работу.

В результате было спроектировано и создано приложение, включающее в себя следующие компоненты:

- удобный и простой интерфейс;
- быстрый поиск информации;
- возможность генерации паролей;
- одновременный многопользовательский доступ;

Рассмотрены вопросы безопасности жизнедеятельности, связанные с характеристикой вредных факторов при работе с ПК, организацией рабочего места, требования к санитарно-гигиеническим параметрам рабочих мест, требования к режиму труда и отдыха, пожарная безопасность в аудиториях. Произведен расчет уровня звукового давления на рабочем месте.

Проведено технико-экономическое обоснование проекта, согласно которому общие затраты на внедрение системы составляют 86956,57 руб.

Цели выпускной квалификационной работы достигнуты, поставленные задачи выполнены в полном объеме.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Официальный сайт компании ООО «Нижевартовская Энергосбытовая Компания». – <http://www.nesko-nv.ru/clients/physical-persons/tariffs-for-electricity/> [дата обращения – 25.03.2019]
2. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы, М.: Изд-во стандартов, 2003.– 56 с.
3. СанПиН 2.2.4.3359-16 Санитарно-эпидемиологические требования к физическим факторам на рабочих места, М.: Норматика, 2018.– 68 с.
4. ГОСТ 12.1.003-83 Система стандартов безопасности труда. Шум. Общие требования безопасности, М.: Изд-во стандартов, 1998.– 11 с.
5. СНиП 31-03-2001 Производственные здания, М.: Изд-во стандартов, 2001.– 13 с
6. СНиП 2.01.02-85 Противопожарные нормы, М.: Изд-во стандартов, 1991.– 16 с.
7. Иванова, Г.С. Технология программирования: учебник/ Г.С. Иванова – М.: КНОРУС, 2011. – 336 с.
8. Методические рекомендации по подготовке и оформлению выпускной квалификационной работы (проекта) для технических направлений подготовки 09.03.01 Информатика и вычислительная техника, 09.03.04 Программная инженерия, 12.03.01 Приборостроение, 23.03.01 Технология транспортных процессов / сост. Л.Н.Буйлушкина. - Нижевартовск, 2017. - 35с.
9. Библиотека Microsoft SQL Server. – <https://msdn.microsoft.com/ru-ru/enen/library/bb545450.aspx> [дата обращения – 28.03.2019]
10. Руководство по языку C#. – <https://docs.microsoft.com/ru-ru/dotnet/csharp/> [дата обращения – 29.03.2019]
11. Официальный сайт менеджера паролей «Dashlane». – <https://www.dashlane.com/> [дата обращения – 03.04.2019]

12. Официальный сайт менеджера паролей «1Password». – <https://1password.com/ru/> [дата обращения – 05.04.2019]
13. Официальный сайт менеджера паролей «StickyPassword». – <https://www.stickypassword.com/ru/> [дата обращения – 07.04.2019]
14. Официальный сайт менеджера паролей «LastPass». – <https://www.lastpass.com/ru> [дата обращения – 13.04.2019]
15. Официальный сайт менеджера паролей «KeePass». – <https://keepass.info/> [дата обращения – 15.04.2019]
16. Справочник по Transact-SQL. – <https://docs.microsoft.com/ru-ru/sql/t-sql/language-reference?view=sql-server-2017> [дата обращения – 15.04.2019]
17. Мельников, В.В. Защита информации в компьютерных системах: Учеб. пособие / В. В. Мельников. – М.: Финансы и статистика, 2011. – 368 с.: ил. – 2000 экз. – ISBN 5-279-01631-4.
18. Орлов, С.А. Технологии разработки программного обеспечения: учебник для вузов. / С. А. Орлов. -М: Питер, 2012. - 606 с.
19. Романец, Ю. В. Защита информации в компьютерных системах и сетях / Ю. В. Романец. – М.: Радио и связь, 2012. – 349 с.
20. Мартин, Р.С. Принципы, паттерны и методики гибкой разработки на языке С# / Р.С. Мартин. – М.: Символ-Плюс, 2011. – 768 с.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ПРОГРАММЫ

1. Исходный код файла App.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <add name="pwdMan.Properties.Settings.gesnvConnectionString"
      connectionString="Data Source=ges-pc92;Initial Catalog=gesnv;Integrated
Security=True"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
  </startup>
</configuration>
```

2. Исходный код файла login.cs

```
using System;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace pwdMan
{
  public partial class login : MetroFramework.Forms.MetroForm
  {
    public login()
    {
      InitializeComponent();
      textBox2.PasswordChar = '\u25CF';
      metroComboBox1.SelectedIndex = 0;
    }
    private void login_Load(object sender, EventArgs e)
    {

```

```

    }
    public void button1_Click(object sender, EventArgs e)
    {
        SqlConnectionStringBuilder connect1 = new
SqlConnectionStringBuilder();
        connect1.DataSource = metroComboBox1.Text; // имя сервера
        connect1.InitialCatalog = "gesnv"; // имя базы данных
        if (checkBox1.Checked == true)
        {
            connect1.IntegratedSecurity = true; //проверка подлинности через
авторизацию Windows
        }
        else
        {
            connect1.IntegratedSecurity = false;
            connect1.UserID = textBox1.Text;
            connect1.Password = textBox2.Text;
        }
        int j = 1;
        try
        {
            SqlConnection testconnect = new
SqlConnection(connect1.ConnectionString);
            testconnect.Open();
            j++;
            testconnect.Close();
        }
        catch
        {
            MessageBox.Show("Неверные параметры подключения", "Упс!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        if (j != 1)
        {
            string conf_name =
"pwdMan.Properties.Settings.gesnvConnectionString"; // название строки подключения
из app.config
            ConnectionStringSettings cs =
ConfigurationManager.ConnectionStrings[conf_name];
            cs = new ConnectionStringSettings(conf_name,
connect1.ConnectionString, "System.Data.SqlClient");

            System.Configuration.Configuration config =
                ConfigurationManager.OpenExeConfiguration(
                    ConfigurationUserLevel.None);
            // Получаем доступ к строкам подключения
            ConnectionStringsSection csSection =
                config.ConnectionStrings;
            // заменяем строку подключения
            csSection.ConnectionStrings.Remove(cs.Name);
            csSection.ConnectionStrings.Add(cs);
            // сохранение файла конфигурации
            config.Save(ConfigurationSaveMode.Modified);
            this.Hide();
            // MessageBox.Show("Подключение установлено");
        }
    }
}

```

```

    }
}
private void login_FormClosing(object sender, FormClosingEventArgs e)
{
    if (e.CloseReason == CloseReason.UserClosing) Application.Exit();
}
private void button2_Click(object sender, EventArgs e)
{
    if (textBox2.PasswordChar == '\u25CF') textBox2.PasswordChar = '\0';
    else textBox2.PasswordChar = '\u25CF';
}
private void button2_Leave(object sender, EventArgs e)
{
}
private void checkBox1_EnabledChanged(object sender, EventArgs e)
{
}
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    if (checkBox1.Checked)
    {
        textBox1.Enabled = false;
        textBox2.Enabled = false;
    }
    else
    {
        textBox1.Enabled = true;
        textBox2.Enabled = true;
    }
}
}
}
}

```

3. Исходный код файла pwGen.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace pwdMan
{
    public partial class pwGen : MetroFramework.Forms.MetroForm
    {

```

```

public pwdGen()
{
    InitializeComponent();
    metroLabel3.Text = metroTrackBar1.Value.ToString();
    metroLabel4.Text = metroTrackBar2.Value.ToString();
}
private void metroTrackBar1_ValueChanged(object sender, EventArgs e)
{
    metroLabel3.Text = metroTrackBar1.Value.ToString();
}
private void metroTrackBar2_ValueChanged(object sender, EventArgs e)
{
    metroLabel4.Text = metroTrackBar2.Value.ToString();
}
private string GetCharacter()
{
    string[] Lower = new string[26] { "a", "b", "c", "d", "e", "f", "g",
    "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w",
    "x", "y", "z" };
    string[] Upper = new string[26] { "A", "B", "C", "D", "E", "F", "G",
    "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W",
    "X", "Y", "Z" };
    string[] Number = new string[10] { "1", "2", "3", "4", "5", "6", "7",
    "8", "9", "0" };
    string[] Symbol = new string[] { "!", "@", "#", "$", "%", "&", "*", "?" };
};

Random rand = new System.Random(Guid.NewGuid().GetHashCode());
int rplace = rand.Next(0, 4);
int rplace2 = 0;
while (!IsChecked(rplace))
{
    rplace = rand.Next(0, 4);
}
switch (rplace)
{
    case 0:
        rplace2 = rand.Next(0, 26);
        return Lower[rplace2];
    case 1:
        rplace2 = rand.Next(0, 26);
        return Upper[rplace2];
    case 2:
        rplace2 = rand.Next(0, 10);
        return Number[rplace2];
    case 3:
        rplace2 = rand.Next(0, 8);
        return Symbol[rplace2];
    default:
        return "";
}
}
private bool IsChecked(int r)

```

```

{
    switch (r)
    {
        case 0:
            metroLabel6.Text = "Пароли сгенерированы";
            return metroCheckBox1.Checked;
        case 1:
            metroLabel6.Text = "Пароли сгенерированы";
            return metroCheckBox2.Checked;
        case 2:
            metroLabel6.Text = "Пароли сгенерированы";
            return metroCheckBox3.Checked;
        case 3:
            metroLabel6.Text = "Пароли сгенерированы";
            return metroCheckBox4.Checked;
        default:
            metroLabel6.Text = "Пароли сгенерированы";
            return true;
    }
}

private void button1_Click(object sender, EventArgs e)
{
    if (metroCheckBox1.Checked || metroCheckBox2.Checked ||
metroCheckBox3.Checked || metroCheckBox4.Checked )
    {
        richTextBox1.Text = "";

        //Change Numeric Up/Down boxes value into an int

        int PasswordLength = Convert.ToInt32(metroTrackBar1.Value);
        int NumberOfPasswords = Convert.ToInt32(metroTrackBar2.Value);
        // metroLabel6.Text = "Generating Password/s";
        for (int i = 0; i < NumberOfPasswords; i++) //generate each password
        {
            for (int g = 0; g < PasswordLength; g++) //generate each character
value
            {
                richTextBox1.Text += GetCharacter();
            }
            richTextBox1.Text += "\r\n"; //seperate the passwords to multiple
lines
        }
    }
    else
    {
        MessageBox.Show("Необходимо выбрать хотя бы один параметр из группы
'СИМВОЛЫ'", "УПС!",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void button2_Click(object sender, EventArgs e)
{

```

```

metroTrackBar1.Value = 7;
metroTrackBar2.Value = 5;
metroCheckBox1.Checked = true;
metroCheckBox2.Checked = false;
metroCheckBox3.Checked = true;
metroCheckBox4.Checked = false;
richTextBox1.Text = " ";
metroLabel6.Text = "ГОТОВ";
    }
}

```

4. Исходный код файла userform.cs

```

using System;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace pwdMan
{
    public partial class userform : MetroFramework.Forms.MetroForm
    {
        public userform()
        {
            InitializeComponent();
        }
        public int UserOrAdmin = 1;
        private void userform_Load(object sender, EventArgs e)
        {
            this.metroGrid4.Visible = false;
            this.metroGrid5.Visible = false;
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "gesnvDataSet.АдминистративныеПароли". При необходимости она может быть перемещена
            или удалена.

            this.административныеПаролиTableAdapter.Fill(this.gesnvDataSet.АдминистративныеПаро
            ли);

            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "gesnvDataSet.Пользователи". При необходимости она может быть перемещена или
            удалена.

            this.пользователиTableAdapter.Fill(this.gesnvDataSet.Пользователи);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "gesnvDataSet.Подгруппа". При необходимости она может быть перемещена или удалена.
            this.подгруппаTableAdapter.Fill(this.gesnvDataSet.Подгруппа);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "gesnvDataSet.Группа". При необходимости она может быть перемещена или удалена.
        }
    }
}

```

```

        this.группаTableAdapter.Fill(this.gesnvDataSet.Группа);
        // TODO: данная строка кода позволяет загрузить данные в таблицу
"gesnvDataSet1.Пользователи". При необходимости она может быть перемещена или
удалена.
        this.пользователиTableAdapter1.Fill(this.gesnvDataSet1.Пользователи);
        //this.Hide();
        login loginform = new login();
        loginform.ShowDialog();
        // SqlConnection connection = new
SqlConnection(DataSource.ConnectionString);
        // connection.Open();

        // pwdMan.Properties.Settings.Default.gesnvConnectionString =
DataSource.ConnectionString;

        /* // TODO: данная строка кода позволяет загрузить данные в таблицу
"gesnvDataSet1.Пользователи". При необходимости она может быть перемещена или
удалена.
this.пользователиTableAdapter1.Fill(this.gesnvDataSet1.Пользователи);
        // TODO: данная строка кода позволяет загрузить данные в таблицу
"gesnvDataSet.Пользователи". При необходимости она может быть перемещена или
удалена.
        this.пользователиTableAdapter.Fill(this.gesnvDataSet.Пользователи);
        // TODO: данная строка кода позволяет загрузить данные в таблицу
"gesnvDataSet.Подгруппа". При необходимости она может быть перемещена или удалена.
        this.подгруппаTableAdapter.Fill(this.gesnvDataSet.Подгруппа);
        // TODO: данная строка кода позволяет загрузить данные в таблицу
"gesnvDataSet.Группа". При необходимости она может быть перемещена или удалена.
        this.группаTableAdapter.Fill(this.gesnvDataSet.Группа);*/
    }

    private void userform_FormClosed(object sender, FormClosedEventArgs e)
    {

    }

    private void metroLink1_Click(object sender, EventArgs e)
    {
        metroContextMenu1.Show(metroLink1, 0, metroLink1.Height);
    }

    private void выходToolStripMenuItem_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void metroGrid3_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {

    }

    private void fillByToolStripButton_Click(object sender, EventArgs e)
    {

    }

```

```

private void metroTextBox1_TextChanged(object sender, EventArgs e)
{
    пользователиBindingSource.SuspendBinding();
    if (metroTextBox1.Text != null)
    {
        for (int i = 0; i < metroGrid4.RowCount; i++)
        {
            metroGrid4.Rows[i].Visible = false;
        }

        for (int i = 0; i < metroGrid4.RowCount; i++)
        {
            metroGrid4.Rows[i].Selected = false;
            for (int j = 0; j < metroGrid4.ColumnCount; j++)
                if (metroGrid4.Rows[i].Cells[j].Value != null)
                    if
(metroGrid4.Rows[i].Cells[j].Value.ToString().ToLower().Contains(metroTextBox1.Text
.ToLower()))
                        {
                            metroGrid4.Rows[i].Visible = true;
                            break;
                        }

        }
    }

    int Rall=0;
    for (int i = 0; i < metroGrid4.RowCount; i++)
    {
        if (metroGrid4.Rows[i].Visible == true) Rall++;
        metroLabel3.Text = Rall.ToString();
    }
    пользователиBindingSource.ResumeBinding();
}

private void генераторПаролейToolStripMenuItem_Click(object sender,
EventArgs e)
{
    pwdGen passgenform = new pwdGen();
    passgenform.Show();
}
private void metroLink3_Click(object sender, EventArgs e)
{
    metroTextBox2.Text = "";
    string abc = "abcdefghijklmnopqrstuvwxy1234567890";
    Random rnd = new Random();
    for (int i = 0; i < 8; i++)
        metroTextBox2.Text += abc[rnd.Next(abc.Length)];
    Clipboard.SetText(metroTextBox2.Text);
}

```



```

private void button1_Click(object sender, EventArgs e)
{
    this.пользователиTableAdapter.Update(this.gesnvDataSet);
    this.группаTableAdapter.Update(this.gesnvDataSet);
    this.подгруппаTableAdapter.Update(this.gesnvDataSet);
    this.административныеПаролиTableAdapter.Update(this.gesnvDataSet);
}

private void metroTextBox1_Enter(object sender, EventArgs e)
{
    if (UserOrAdmin == 1)
    {
        metroGrid1.Visible = false;
        metroGrid2.Visible = false;
        metroGrid3.Visible = false;
        button1.Visible = false;
        metroGrid4.Visible = true;
        metroLabel3.Text = metroGrid4.RowCount.ToString();
    }
}

private void metroLink2_Click(object sender, EventArgs e)
{
    if (UserOrAdmin == 1)
    {
        metroTextBox1.Text = "";
        metroGrid1.Visible = true;
        metroGrid2.Visible = true;
        metroGrid3.Visible = true;
        metroGrid4.Visible = false;
        button1.Visible = true;
        int R = metroGrid3.Rows.Count;
        R--;
        metroLabel3.Text = R.ToString();
    }
}

private void metroGrid3_Layout(object sender, LayoutEventArgs e)
{
    if (metroGrid3.Visible == true)
    {
        int R = metroGrid3.Rows.Count;
        R--;
        metroLabel3.Text = R.ToString();
    }
}

private void metroGrid4_CellMouseDown(object sender,
DataGridViewCellMouseEventArgs e)
{
    // Ignore if a column or row header is clicked
    if (e.RowIndex != -1 && e.ColumnIndex != -1)
    {

```

```

        if (e.Button == MouseButton.Right)
        {
            DataGridViewCell clickedCell = (sender as
DataGridView).Rows[e.RowIndex].Cells[e.ColumnIndex];

            // Here you can do whatever you want with the cell
            this.metroGrid4.CurrentCell = clickedCell; // Select the
clicked cell, for instance

            // Get mouse position relative to the vehicles grid
            var relativeMousePosition =
metroGrid4.PointToClient(Cursor.Position);

            // Show the context menu
            this.metroContextMenu3.Show(metroGrid4, relativeMousePosition);
        }
    }

    private void metroGrid3_CellMouseDown(object sender,
DataGridViewCellMouseEventArgs e)
    {
        // Ignore if a column or row header is clicked
        if (e.RowIndex != -1 && e.ColumnIndex != -1)
        {
            if (e.Button == MouseButton.Right)
            {
                DataGridViewCell clickedCell = (sender as
DataGridView).Rows[e.RowIndex].Cells[e.ColumnIndex];

                // Here you can do whatever you want with the cell
                this.metroGrid3.CurrentCell = clickedCell; // Select the
clicked cell, for instance

                // Get mouse position relative to the vehicles grid
                var relativeMousePosition =
metroGrid3.PointToClient(Cursor.Position);

                // Show the context menu
                this.metroContextMenu2.Show(metroGrid3, relativeMousePosition);
            }
        }
    }

    private void удалитьЗаписьToolStripMenuItem_Click(object sender, EventArgs
e)
    {
        try
        {
            int del = metroGrid3.SelectedCells[0].RowIndex;
            metroGrid3.Rows.RemoveAt(del);
        }
        catch (Exception ex)
        {

```

```

        MessageBox.Show("Кажется что-то пошло не так", "Упс!",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

    }
    private void добавитьЗаписьToolStripMenuItem_Click(object sender, EventArgs
e)
    {
    }

    private void скопироватьПарольToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        Clipboard.SetText(metroGrid3.SelectedCells[4].Value.ToString());
    }

    private void toolStripMenuItem2_Click(object sender, EventArgs e)
    {
        Clipboard.SetText(metroGrid4.SelectedCells[6].Value.ToString());
    }

    private void metroGrid4_CellFormatting(object sender,
DataGridViewCellFormattingEventArgs e)
    {
        /* if (e.ColumnIndex == 6 && e.Value != null && checkBox1.Checked==true)
        {
            metroGrid4.Rows[e.RowIndex].Tag = e.Value;
            e.Value = new String('\u25CF', e.Value.ToString().Length);
        }*/
    }
    private void пользователиToolStripMenuItem_Click(object sender, EventArgs
e)
    {
        metroTextBox1.Enabled = true;
        UserOrAdmin = 1;
        metroGrid2.Visible = true;
        metroGrid3.Visible = true;
        metroGrid1.Visible = true;
        metroGrid4.Visible = false;
        metroGrid5.Visible = false;
        int R = metroGrid3.Rows.Count;
        R--;
        metroLabel3.Text = R.ToString();
    }
    private void административныеПаролиToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        button1.Visible = true;
        metroTextBox1.Enabled = false;
        UserOrAdmin = 2;
        metroGrid2.Visible = false;
    }

```

```

metroGrid3.Visible = false;
metroGrid1.Visible = true;
metroGrid4.Visible = false;
metroGrid5.Visible = true;
int R = metroGrid5.Rows.Count;
R--;
metroLabel3.Text = R.ToString();

}

private void metroGrid5_Layout(object sender, LayoutEventArgs e)
{
    int R = metroGrid5.Rows.Count;
    R--;
    metroLabel3.Text = R.ToString();
}

private void metroLink4_Click(object sender, EventArgs e)
{
    if (metroGrid3.Visible == true)
    {
        Microsoft.Office.Interop.Excel.Application XlApp = new
Microsoft.Office.Interop.Excel.Application();
        Microsoft.Office.Interop.Excel.Workbook XlWorkBook =
XlApp.Workbooks.Add(); //создать новый файл
        Microsoft.Office.Interop.Excel.Worksheet XlWorkSheet =
(Microsoft.Office.Interop.Excel.Worksheet)XlWorkBook.Worksheets.get_Item(1); //1-й
лист по порядку

        for (int i = 0; i < metroGrid3.Rows.Count; i++)
        {
            for (int j = 0; j < metroGrid3.ColumnCount; j++)
            {
                XlWorkSheet.Cells[i + 1, j + 1] =
metroGrid3.Rows[i].Cells[j].Value;
            }
        }
        XlApp.Visible = true;
    }
    if (metroGrid4.Visible == true)
    {
        Microsoft.Office.Interop.Excel.Application XlApp = new
Microsoft.Office.Interop.Excel.Application();
        Microsoft.Office.Interop.Excel.Workbook XlWorkBook =
XlApp.Workbooks.Add(); //создать новый файл
        Microsoft.Office.Interop.Excel.Worksheet XlWorkSheet =
(Microsoft.Office.Interop.Excel.Worksheet)XlWorkBook.Worksheets.get_Item(1); //1-й
лист по порядку
        for (int i = 0; i < metroGrid4.Rows.Count; i++)
        {
            for (int j = 0; j < metroGrid4.ColumnCount; j++)
            {
                XlWorkSheet.Cells[i + 1, j + 1] =
metroGrid4.Rows[i].Cells[j].Value;
            }
        }
    }
}

```



```
{
    private static string _ConnectionString;
    public static string ConnectionString
    {
        get
        {
            _ConnectionString = FunctionCreateConnectionString();
            return _ConnectionString;
        }
    }
    public static string whereDB;
    public static string Login;
    public static string Password;
    private static string FunctionCreateConnectionString()
    {
        /* ConnectionStringSettings settings =
ConfigurationManager.ConnectionStrings["pwdMan.Properties.Settings.gesnvConnections
tring"];
        string connectString = settings.ConnectionString;
        SqlConnectionStringBuilder builder = new
SqlConnectionStringBuilder(connectString);
        builder.DataSource = whereDB;
        builder.UserID = Login;
        builder.Password = Password;
        // return builder.ConnectionString;
    }
} */
}
```

ПРИЛОЖЕНИЕ Б. КОМПАКТ - ДИСК

Содержание:

1. Пояснительная записка.
2. Презентация.
3. Приложение «pwdMan».
4. Исходный код программы.