Министерство образования и науки Самарской области

ГБПОУ «ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ»

**КУРСОВОЙ ПРОЕКТ**

**Разработка программы по автоматизации и поддержки управления персоналом для Военкомата.**

ПМ.11 РАЗРАБОТКА, АДМИНИСТРИРОВАНИЕ И ЗАЩИТА БАЗ ДАННЫХ

09.02.07 Информационные системы и программирование

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обучающийся | | | | | |  |  |  | **Мурзин К.А.** |
|  | | | | | |  | *подпись* |  | *И.О. Фамилия* |
|  |  |  | 20 | 23 |  |  |  |  |  |
|  | | | | | |  |  |  |  |
| Оценка выполнения и защиты курсового проекта | | | | | | | | |  |
|  | | | | | | | | |  |
|  | | | | | | | | |  |
| Руководитель курсового проекта | | | | | |  |  |  | Х.М Матьякубов |
|  | | | | | |  | *подпись* |  | *И.О. Фамилия* |
|  |  |  | 20 | 23 |  |  |  |  |  |
|  | | | | | |  |  |  |  |
|  | | | | | |  |  |  |  |
|  | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | | | | | |  |  |  |  |

г. Самара, 2023

**ГБПОУ «ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ»**

**КАЛЕНДАРНЫЙ ПЛАН  
выполнения курсового проекта**

**ПМ.11 РАЗРАБОТКА, АДМИНИСТРИРОВАНИЕ И ЗАЩИТА БАЗ ДАННЫХ**

**09.02.07 Информационные системы и программирование**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обучающемуся | | | 3 | курса | ИСП-336 | группы | Мурзин К.А. | | | | |
|  | | |  |  |  |  | ФИО | | | | |
| По теме | | Разработка программы по автоматизации и поддержки управления персоналом для Военкомата. | | | | | | | | | |
| **№**  **этапа**  **работы** | **Содержание этапов работы** | | | | | | | **Плановый срок выполнения этапа** | **Отметка о выполнении**  **этапа (подпись руководителя)** | |
|  | Написано введение курсового проекта. | | | | | | | 01.02.23 |  | |
|  | Проведен анализ:объекта, процесса, предметной области как объекта информатизации;информационного обеспечения объекта информатизации. | | | | | | | 03.02.23 |  | |
|  | Описаны:организационная и функциональные схемы объектов информатизации;категории пользователей и потоков данных объекта информатизации. | | | | | | | 05.02.23 |  | |
|  | Проведен анализ функциональных требований программного продукта и входных-выходных данных. | | | | | | | 09.02.23 |  | |
|  | Описана структура данных и проведена ее оптимизация. Разработана схема данных. | | | | | | | 12.02.23 |  | |
|  | Описано клиент-серверное взаимодействие. | | | | | | | 15.02.23 |  | |
|  | Проведен анализ и выбор программного обеспечения для разработки программы и СУБД. | | | | | | | 16.02.23 |  | |
|  | Разработана удаленная база данных для программного продукта. | | | | | | | 20.02.23 |  | |
|  | Разработаны и реализованы запросы к БД и функциональные требования к программному продукту. | | | | | | | 25.02.23 |  | |
|  | Разработан интерфейс программы, проведена интеграция всех модулей программы и СУБД. | | | | | | | 05.03.23 |  | |
|  | Проведено тестирование и отладка программного продукта. Программа готова к эксплуатации. | | | | | | | 15.03.23 |  | |
|  | Написано заключение курсового проекта. | | | | | | | 20.03.23 |  | |
|  | Подготовлена презентация и речь для защиты. | | | | | | | 25.03.23 |  | |
| Студент | | | |  | | | Мурзин К.А. | | |
| Руководитель | | | |  | | | Х.М Матьякубов | | |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc58177990)

[ГЛАВА 1 АНАЛИТИЧЕСКАЯ ЧАСТЬ 7](#_Toc58177991)

[1.1 Анализ объекта, процесса, предметной области 7](#_Toc58177992)

[1.2 Анализ информационного обеспечения объекта информатизации 8](#_Toc58177993)

[1.3 Организационная и функциональная схемы объектов информатизации 8](#_Toc58177994)

[1.4 Описание категорий пользователей и потоков данных](#_Toc58177995) [10](#_Toc58177995)

[ГЛАВА 2 ПРЕДПРОЕКТНАЯ ЧАСТЬ 12](#_Toc58177996)

[2.1 Анализ функциональных требований 12](#_Toc58177997)

[2.2Анализ входных и выходных данных 12](#_Toc58177998)

[2.3Анализ структуры данных 13](#_Toc58177999)

[2.4 Разработка и оптимизация схемы данных 14](#_Toc58178000)

[2.5 Описание клиент-серверного взаимодействия 15](#_Toc58178001)

[2.6 Анализ и обоснование выбора программного обеспечения 15](#_Toc58178002)

[2.7 Анализ выбора СУБД 16](#_Toc58178003)

[2.8 Разработка тестового сценария и методики испытаний программного продукта 17](#_Toc58178004)

[ГЛАВА 3 ПРОЕКТНАЯ ЧАСТЬ 18](#_Toc58178005)

[3.1 Разработка удаленной базы данных 18](#_Toc58178006)

[3.2 Разработка интерфейса программного продукта 20](#_Toc58178007)

[3.4 Интеграция программных модулей 25](#_Toc58178008)

[3.5 Тестирование и отладка программного продукта 25](#_Toc58178009)

[ЗАКЛЮЧЕНИЕ 28](#_Toc58178010)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 29](#_Toc58178011)

[ПРИЛОЖЕНИЕ А Структурная схема программного продукта 30](#_Toc58178012)

[ПРИЛОЖЕНИЕ Б Схема базы данных 31](#_Toc58178013)

[ПРИЛОЖЕНИЕ В Листинг кода программного продукта 32](#_Toc58178014)

# ВВЕДЕНИЕ

ВОЕННЫЙ КОМИССАРИАТ (военкомат), территориальный орган Министерства обороны РФ, предназначенный для организации и проведения военномобилизационной и учётнопризывной работы, мероприятий по социальноправовой защите военнослужащих, ветеранов, др. граждан, находившихся или находящихся на военной службе, и членов их семей. Создаются В. к. в соответствии с адм.терр. делением: республиканские, окружные, краевые, областные, городские и районные, в отд. случаях – объединённые (для обслуживания нескольких административных рай­онов или городов). Входят в состав военных округов, на территории которых они находятся. В. к. возглавляется воен. комиссаром, который имеет пра­ва заместителя главы органа исполнит. власти субъекта РФ или заместителя главы органа местного самоуправления по вопросам обороны.  
Защита баз данных для военкоматов является крайне важной, поскольку эти данные содержат конфиденциальную информацию о гражданах, которые обязаны проходить военную службу. Базы данных военкоматов содержат информацию о личности, медицинских данных, адресе проживания и других личных данных граждан.

Актуальность выбранной мной темы в том, что моя автоматизированная база данных для воен.кома упростит работу для сотрудников в выполнении их множества не легких задач, и поможет в дальнейшем развитии такой замечательной гос.структуре, как военный комиссариат.

**Цель курсового проекта:** Разработка удаленной и простой базы данных для работы военкомата

**Задачи:**

* выполнить анализ объекта, процесса, предметной области как объекта информатизации;
* описать организационную и функциональную схемы объектов информатизации;
* описать категории пользователей и потоков данных объекта информатизации;
* описать структуру данных и процесс проведения ее оптимизации;
* описать клиент-серверное взаимодействие;
* провести анализ и выбрать программное обеспечение для разработки программы и БД;
* разработать удаленную базу данных и таблицы БД;
* разработать и реализовать запросы к БД;
* произвести разработку интерфейса, работающего с удаленной базой данных;
* провести отладку и тестирование программного продукта;

**Объект исследования**: управление автоматизированной поддержки управления персоналом.

**Предмет исследования:** управление БД для автоматической поддержки управлением персоналом.

КП основан на гипотезе**,** согласно которой автоматизируемый процесс в рассматриваемой организации будет максимально эффективен если:

* Правильно описаны функциональные требования программного продукта:
* Разработаны таблицы базы данных.
* Разработана схема данных.
* Проведен анализ.
* Разработана база данных.
* Разработан оптимальный, удобный и понятный интерфейс и процесс взаимодействия программного продукта с удаленной базой данных
* Написано приложение для администрирования данных.
* Разработана форма аутентификации.
* Программа должна быть понятна для восприятия.
* Реализованы и протестированы все функции программного продукта.

# 

# ГЛАВА 1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

## 1.1 Анализ объекта, процесса, предметной области

Военный комиссариат создается в субъекте Российской Федерации в целях обеспечения исполнения гражданами воинской обязанности, организации и проведения мобилизационной подготовки и мобилизации, реализации права граждан, уволенных с военной службы, и членов их семей.

Отдел имеет в своем распоряжении множество документов, в которых может храниться различная информация о сотрудниках, призывниках, их родителей и внутренней бухгалтерии. В большинстве случаях вся документация хранится в бумажном виде. Из-за большого количества документов, сортировка, обработка, редактирование и поиск информации занимает значительное время, что существенно снижает производительность работы.

В результате анкетирования и интервьюирования всех заинтересованных лиц были сформулированы потребности заказчика относительно разрабатываемого ПО.

Требуется создать программный продукт, который будет выполнять следующие операции:

* Хранить данные о сотрудниках;
* Хранить информацию о призывниках ;
* Хранить логины и пароли:

## 

## 1.2 Анализ информационного обеспечения объекта информатизации

Непосредственно в эксплуатационно-техническом отделе ВКОР имеются компьютеры с характеристиками, предоставленными в таблице 1.

*Таблица 1*

Характеристики компьютеров эксплуатационно-технического отдела

|  |  |  |
| --- | --- | --- |
| **Характеристика** | **Компьютер №1** | **Компьютер №2** |
| Центральный процессор | 1x Intel CORE i5 | 1x AMD A4 |
| Объем оперативной памяти | 2x 4000Mb DDR3 | 1x 8000Mb DDR4 |
| Доступный объем жесткого диска | 500Gb | 1Tb |
| Операционная система | Windows 7 | Windows 11 |

Имеется подключение к сети Интернет, что необходимо для работы с удаленной базой данных.

Таким образом, проведя анализ информационного обеспечения эксплуатационно-технического отдела, выяснилось, что оно подходит для работы с АИС (Автоматизированной информационной системой) и скорость работы будет оптимальной при стабильном подключении к сети Интернет (при условии работы с удаленной базой данных).

## 1.3 Организационная и функциональная схемы объектов информатизации

Организационная схема ВОЕННОГО комиссариата Самарской области представлена на рисунке 1.

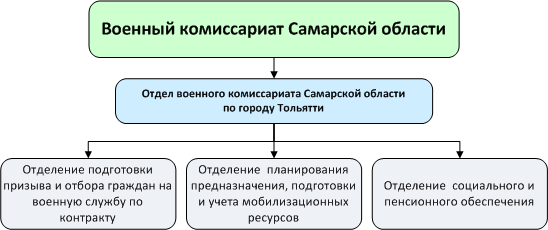


Рисунок 1 – Структурная схема ВОЕННОМ комиссариате.

Объектом информатизации является диспетчерская служба эксплуатационно**-**техническогоотделаВКОР. Структура управления данным структурным подразделением представлена на рисунке 2.

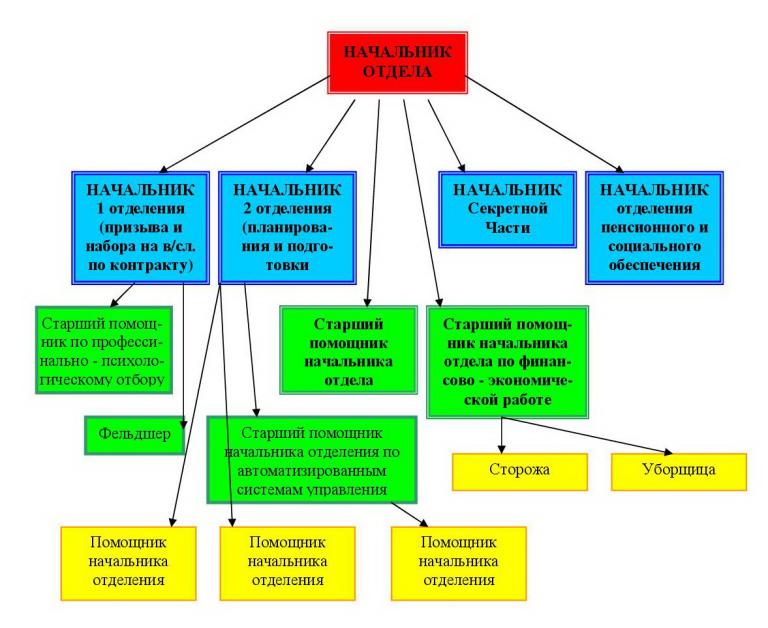


Рисунок 2 – Структура управления ВОЕННОГО комиссариата.

Функциональная схема разрабатываемого программного продукта представлена на рисунке 3.

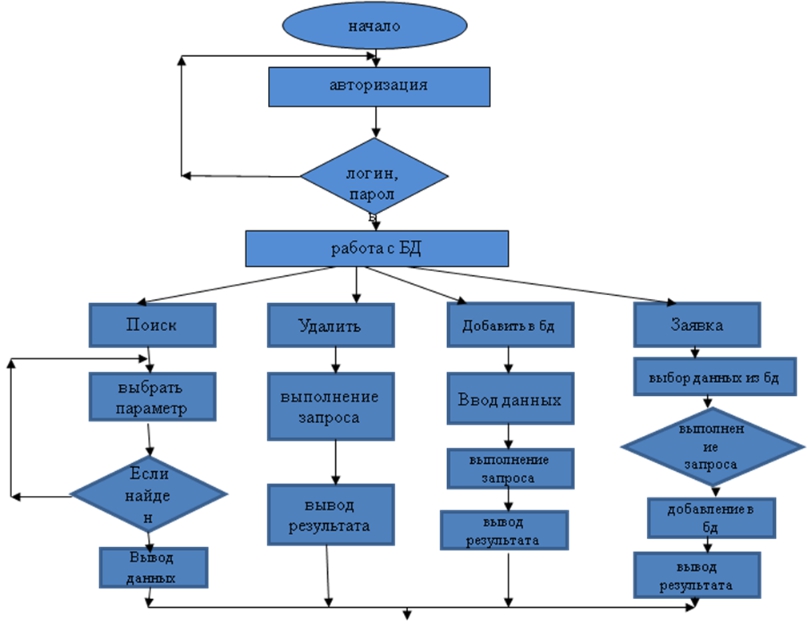


Рисунок 3 – Функциональная схема ПП

## 

## 1.4 Описание категорий пользователей и потоков данных

В системе есть один тип пользователя: старший помощник начальника отдела.

Старшие помощники начального отдела могут просматривать, добавлять и редактировать любые данные.

Описание категорий пользователей и потоков данных объекта информатизации расположено в таблице 2.

*Таблица 2*

Категории пользователей

| **Категория пользователей** | **Функциональные возможности** |
| --- | --- |
| Старший помощник начальника отдела | Удаление данных  Добавление данных  Редактирование данных  Экспорт в Word  Поиск данных |
| Секретарь | Просмотр данных и их редактирование |

# ГЛАВА 2 ПРЕДПРОЕКТНАЯ ЧАСТЬ

## 2.1 Анализ функциональных требований

В результате анкетирования и интервьюирования всех заинтересованных лиц были сформулированы потребности заказчика относительно разрабатываемого ПО и сформулированы основные функции:

* аутентификация пользователя – программный продукт должен обеспечивать конфиденциальность хранимых в программе данных путем аутентификации с использованием логина и пароля;
* добавление и удаление информации о сотрудниках, призывниках;
* поиск информации о сотрудниках, призывниках;
* функция генерации заявок в формате электронного документа.

В случае, если пользователь не прошел авторизацию, доступ к основному окну программы предоставлен не будет, и пользователь увидит сообщение об ошибке авторизации.

При успешной аутентификации пользователя открывается главная форма программы, предназначенная для этой категории. Главная форма содержит в себе весь основной функционал программного продукта, а именно:

* просмотр таблиц;
* добавление информации;
* редактирование информации;
* формирование таблиц.

## 2.2Анализ входных и выходных данных

Входные данные:

* информация о логинах и паролях;
* информация о сотрудниках;
* информация о призывнике;

Выходные данные:

* результаты внесенных изменений;
* сгенерированные таблицы.

## 

## 2.3Анализ структуры данных

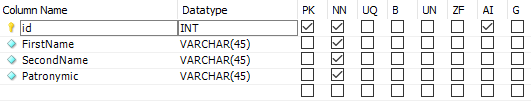


Рисунок 4 – Таблица «Employee»

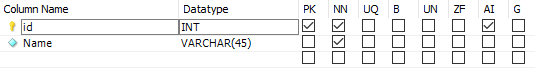


Рисунок 5 – Таблица «Fit\_categories»

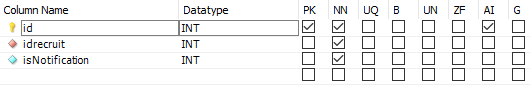


Рисунок 6 – Таблица «Notification»

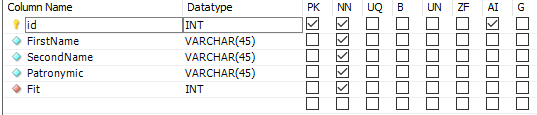


Рисунок 7 – Таблица «recruit»

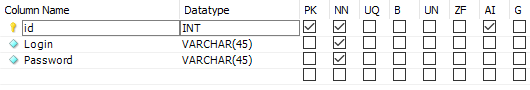


Рисунок 8 – Таблица «User»

## 2.4 Разработка и оптимизация схемы данных

Для предметной области была разработана реляционная схема БД ( ER – диаграмма):

1. Выделяем сущности сотрудники, пользователи.
2. Выделяем сотрудники сущности. Это – сотрудники, призывник ,пользователь.
3. Выделяем дочерние сущности. Это – призывник .
4. В каждой сущности определяем атрибуты.

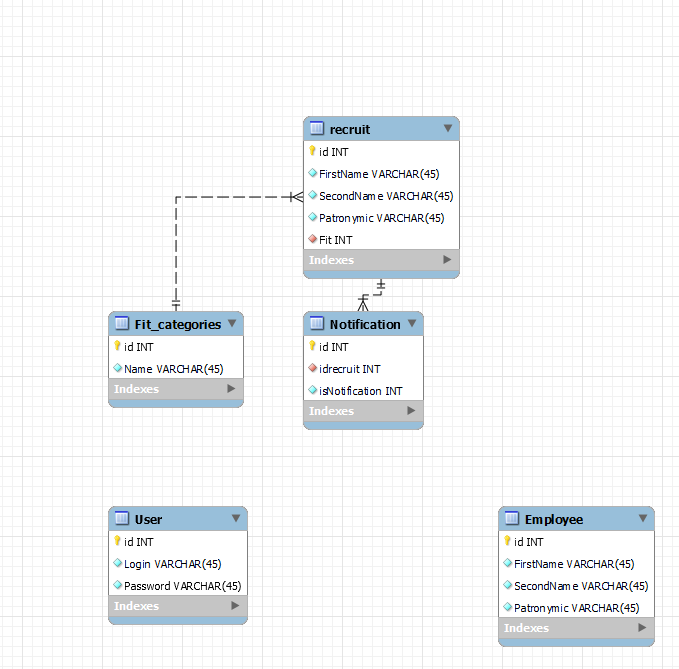


Рисунок 9 – ER - диаграмма

## 

## 2.5 Описание клиент-серверного взаимодействия

С помощью специальной библиотеки производится подключение к базе данных, с помощью которой осуществляется работа с программой. В самой программе пишется код подключения к этой библиотеке.

Клиент и сервер взаимодействуют друг с другом с помощью специального протокола. Протокол описывает правила, по которым осуществляется взаимодействие с сервером, а само взаимодействие производит клиент. Данная концепция взаимодействия разработана для того чтобы разделить нагрузку между участниками процесса.

## 2.6 Анализ и обоснование выбора программного обеспечения

## При выборе программного обеспечения сравнивались три системы Visual Studio, Project Rider , Eclipse.

## Project Rider - не самый лучший вариант, тк. часть функций все еще находится в разработке и среда разработки содержит много багов, в добавок к этому является платным приложением.

## Eclipse - хоть это и бесплатная среда разработки, но она работает очень нестабильно, из-за чего часто вылетает и имеет много глюков.

## − Visual Studio- является наиболее подходящей тк она является бесплатной и содержит большое количество различных инструментов и дополнений которые могут помочь в разработке. Результаты сравнения языков программирования представлены ниже:

## C# - кроссплатформенный язык постоянно совершенствуется, имеет огромное количество дополнительных библиотек и плагинов, также быстро загружает проект.

## На основании вышесказанного был выбран язык программирования C# и система Visual studio для разработки информационной системы

## 2.7 Анализ выбора СУБД

Выбирая систему управления базами данных (далее – СУБД) для разработки базы данных сравнивались три системы: SQLite, MySQL, SQL. Результаты представлены ниже:

* SQL - имеет возможность отмены запроса, но не является кроссплатформенной СУБД, является платной и может использовать только один механизм хранения.
* SQLite - является кроссплатформенной, бесплатной и маловесящей СУБД, но не подходит для создания многопользовательских приложений, поддерживает ограниченное количество типов данных, также имеет ограничение на количество записанных данных из-за чего при создании больших СУБД будут ошибки.
* MySQL - тоже является кроссплатформенной и бесплатной СУБД , в отличии от SQLite подходит для создания многопользовательской базы данных, поддерживает большое количество типов данных, может обрабатывать большой объем данных в кратчайшие сроки, также имеет большой функционал.

В результате сравнения, наиболее подходящим вариантом является СУБД MySQL.

Для создания базы данных был выбран сервер MySQL. Данная СУБД является одной из самых быстрых и высокопроизводительных, её достаточно просто администрировать. Благодаря данной СУБД обеспечивается высокий уровень безопасности.

## 

## 2.8 Разработка тестового сценария и методики испытаний программного продукта

Тестовый сценарий, представленный в таблице 3, предназначен для проверки корректности работы программного продукта.

*Таблица 3*

Тестовый сценарий работы

| **Функция** | **Действия** | **Ожидаемый результат** |
| --- | --- | --- |
| Авторизация | Запуск программы | Аутентификация администратора |
| Меню программы | Реализация  функций программы | Показ выбранной формы |
| employee | Открывается таблица о работниках | Открытие соответствующего окна |
| Fit\_categories | Открывается таблица о котегории | Открытие соответствующего окна |
| Notification | Открывается таблица о повестках | Открытие соответствующего окна |
| Rectuit | Отркрывается таблица о призывнике | Открытие соответствующего окна |
| user | Отцрывается таблица о пользователе | Открытие соответствующего окна |
| рекдактирование | Нажатие кнопки «рекдактирование » на формах | Данные отредактированны |
| Удаление | Нажатие кнопки «Удаление» на формах | Данные удалены |
| Добавление | Нажатие кнопки «Добавить» на форме заявки | В БД добавлены данные |

# ГЛАВА 3 ПРОЕКТНАЯ ЧАСТЬ

## 3.1 Разработка удаленной базы данных

На сервере MySql с помощью программы Workbench разрабатываем базу данных.

После создания база данных имеет структуру, показанную на рисунке 10.

* id: уникальный идентификатор (формат int);
* first name: имя (обязателен к заполнению формат varchar);
* second name : фамилия (обязателен к заполнению формат varchar);
* patromic (обязателен к заполнению формат varchar);
* fir: подход (формат int).

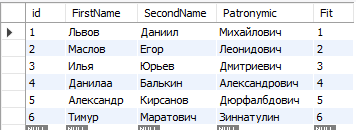


Рисунок 10 ­ Структура БД

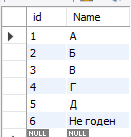


Рисунок 11 ­ Таблица «Fit\_categories»

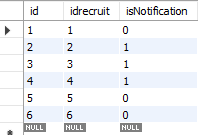


Рисунок 12 ­ Таблица «Notification»

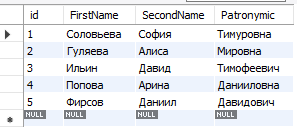


Рисунок 13 ­ Таблица «Employee»

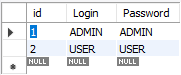


Рисунок 14 ­ Таблица «user»

## 

## 3.2 Разработка интерфейса программного продукта

Разработка интерфейса программного продукта началась с предпроектного анализа. Было представлено первичное видение интерфейса приложения относительно его функционала. Интерфейс формы авторизации был выполнен в виде типичного шаблона, который прост и понятен каждому пользователю (рисунок 15).

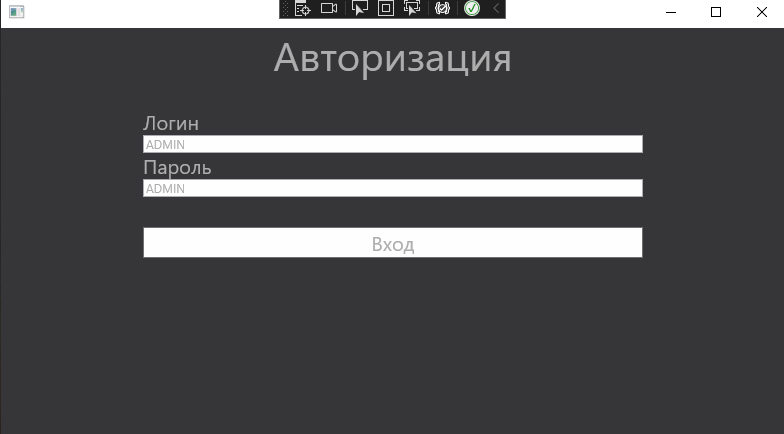


Рисунок 15 – Форма авторизации

Если авторизация прошла успешно, появляется форма МЕНЮ (рисунок 18)

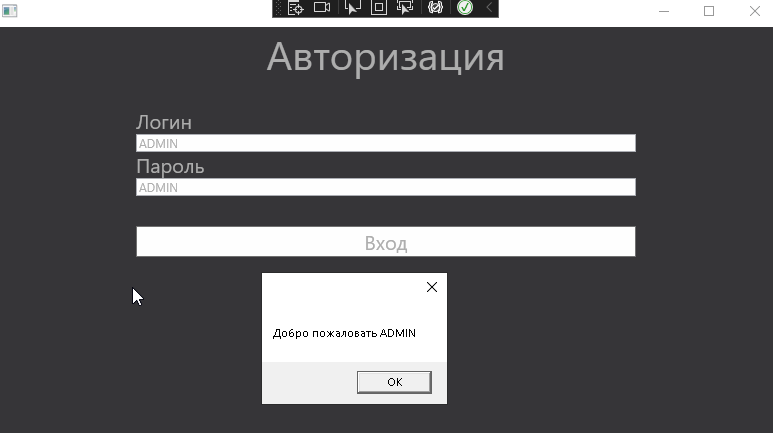


Рисунок 17 –Главное меню

**3.3 Реализация** **функциональных требований**

После успешной авторизации, пользователь видит главное меню с доступными функциями (рисунок 18)

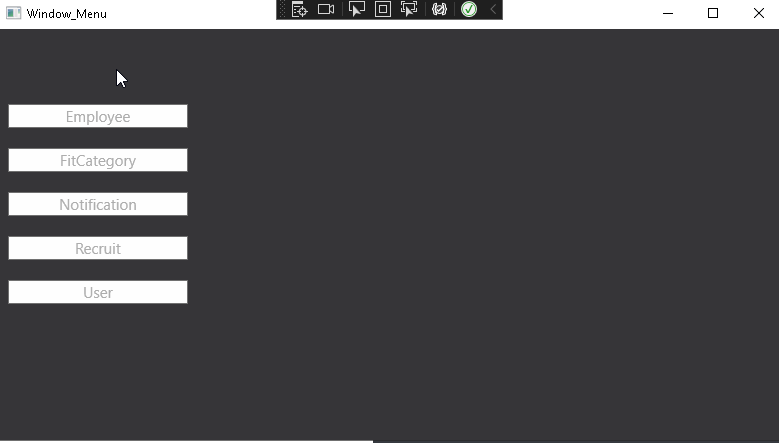


Рисунок 18- Главное меню

При нажатии кнопки «Расходники» пользователь видит окно работы с расходными материалами, в котором можно добавить, удалить, а также найти нужный материал (рисунок 19).

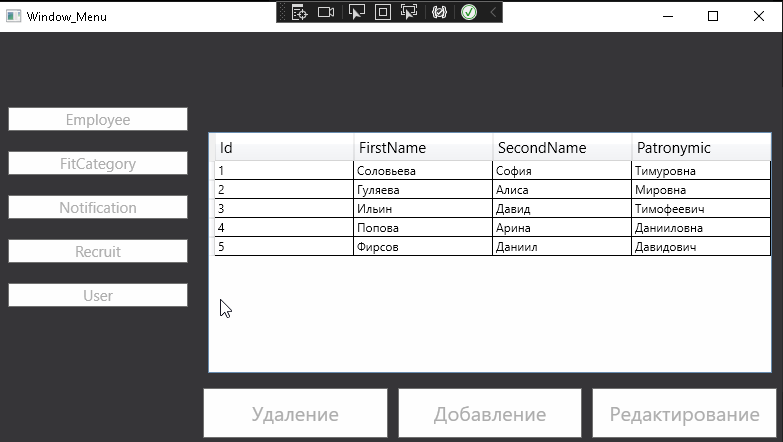


Рисунок 19- Форма «работники»

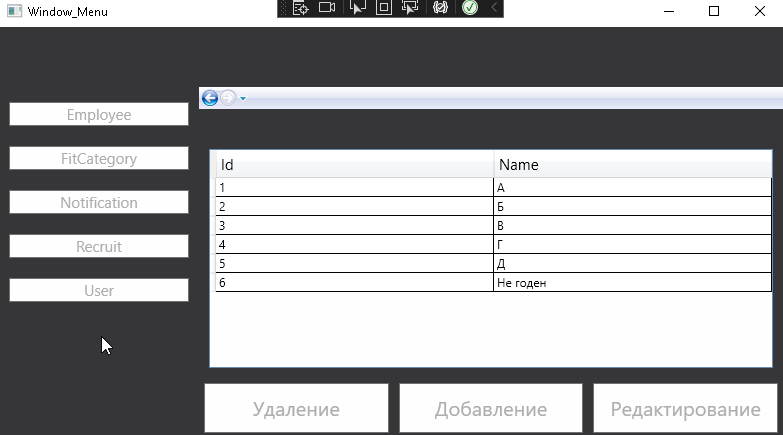


Рисунок 20- Форма «категория»

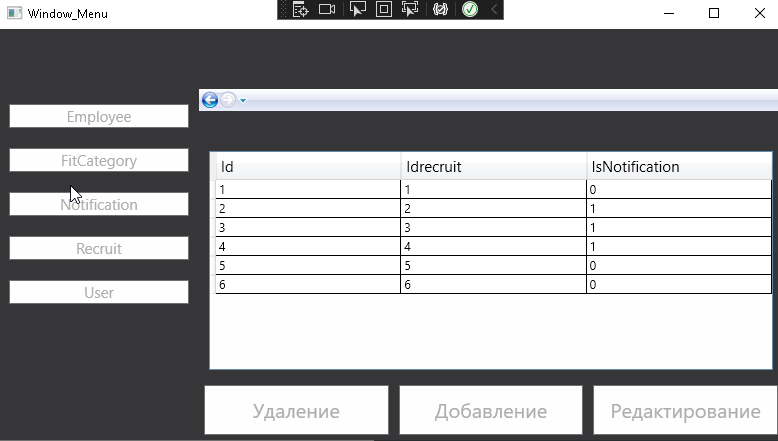


Рисунок 21- Форма «нотификация»

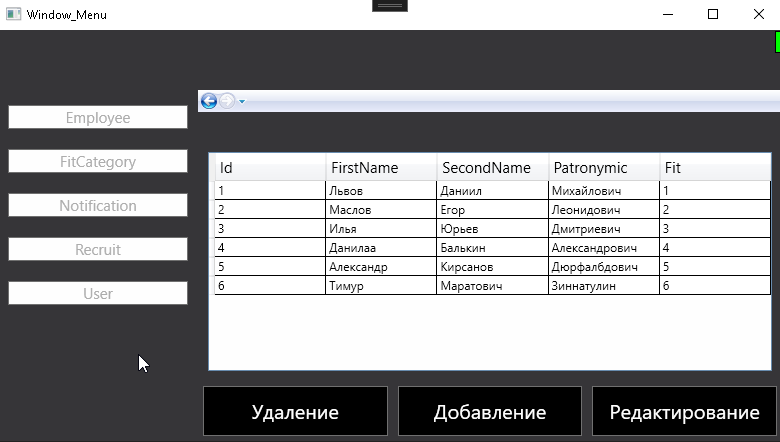
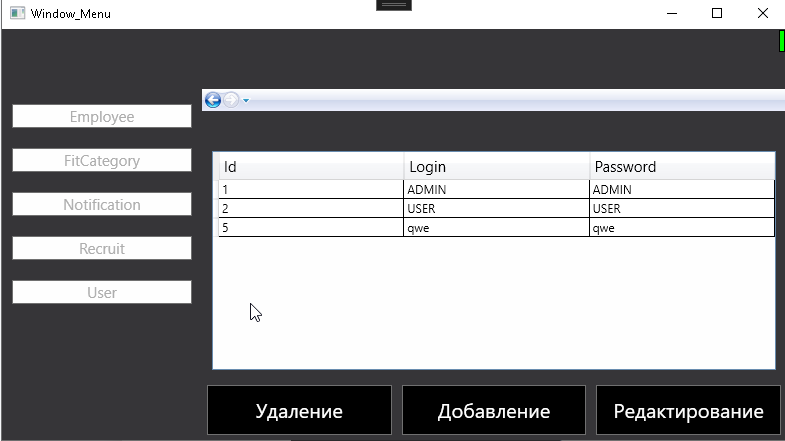


Рисунок 22 – Реализация функции «призывник»

 Рисунок 23 – Реализация функции «пользователь»

## 

## 3.4 Интеграция программных модулей

Подключение удаленной базы данных к программе происходит с помощью драйвера MySQL Connector ODBC.

Строка подключения расположена в коде программы, а ее параметры расположены в текстовых файлах, расположенных в папке с программой.

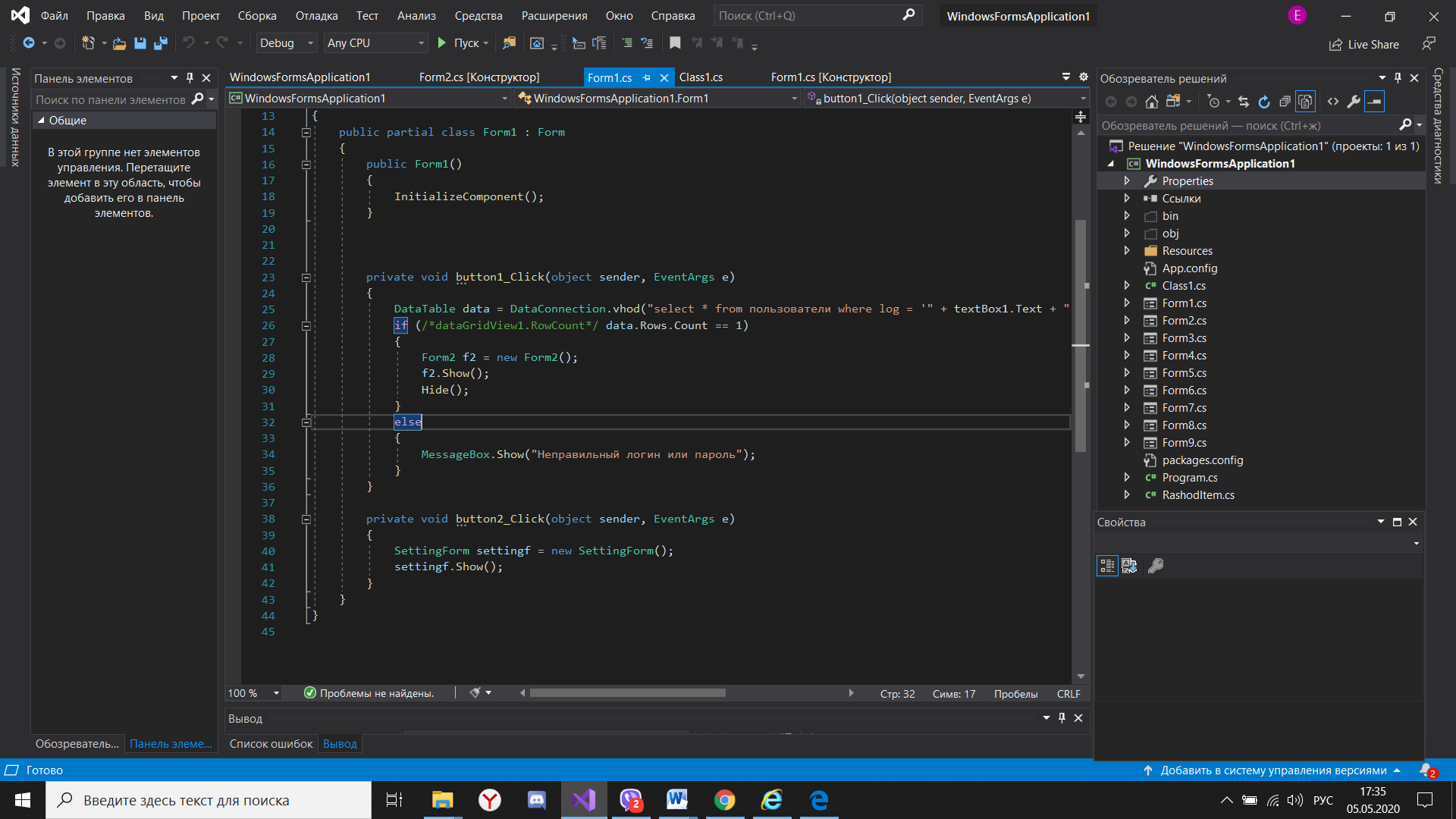


Рисунок 24 – Строка подключения к базе данных

Параметры подключения передаются в форму проверки соединения (рисунок 25).

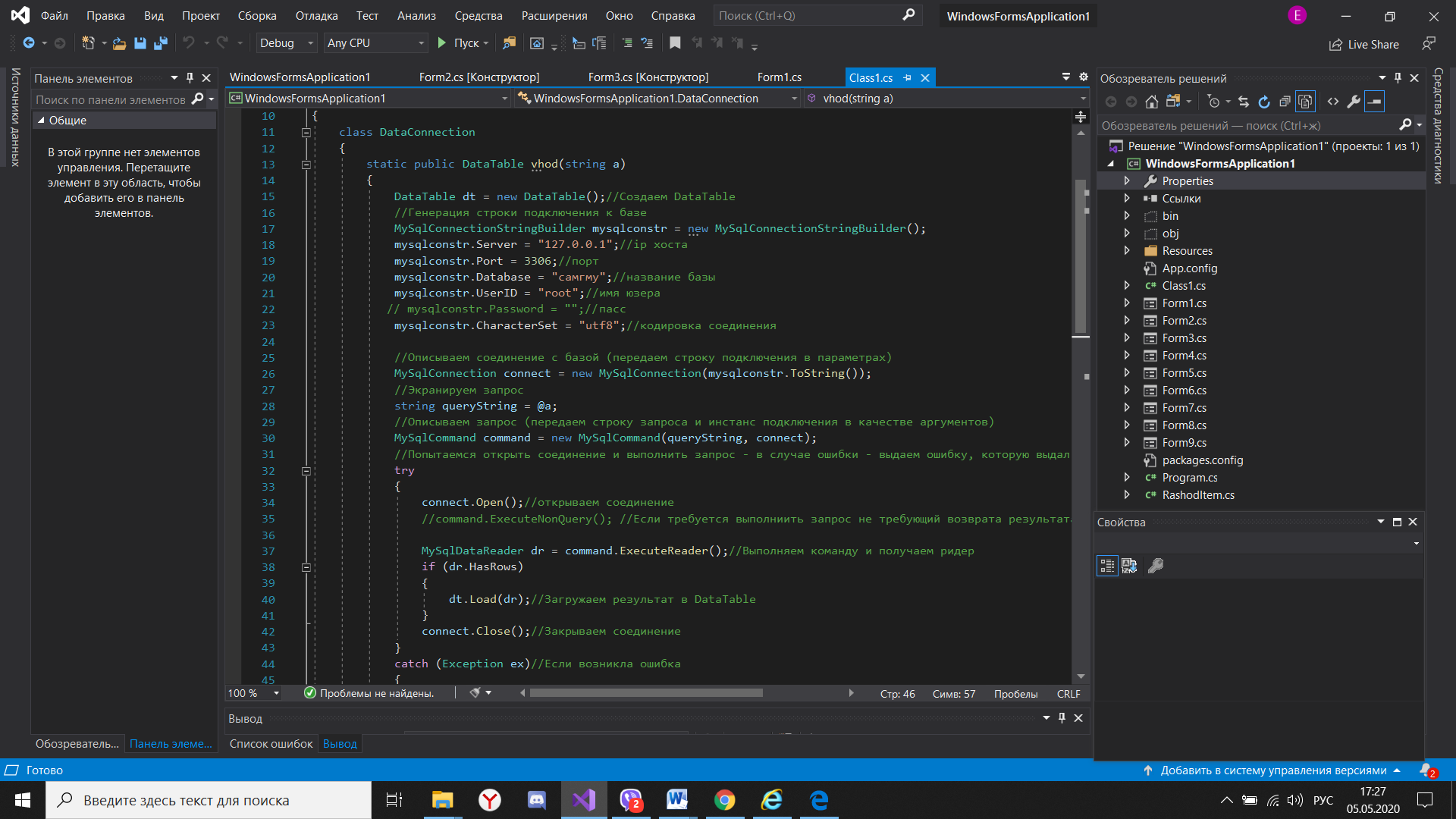


Рисунок 25 – Параметры подключения

## 3.5 Тестирование и отладка программного продукта

Процесс тестирования программы представлен в таблице 3.

*Таблица 3*

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Действия** | **Ожидаемый результат** | **Полученный результат** |
| Авторизация | Запуск программы | Аутентификация администратора | Аутентификация администратора |
| Меню программы | Реализация  функций программы | Показ выбранной формы | Показ выбранной формы |
| employee | Открывается таблица о работниках | Открытие соответствующего окна | Открытие соответствующего окна |
| Fit\_categories | Открывается таблица о котегории | Открытие соответствующего окна | Открытие соответствующего окна |
| Notification | Открывается таблица о повестках | Открытие соответствующего окна | Открытие соответствующего окна |
| Rectuit | Отркрывается таблица о призывнике | Открытие соответствующего окна | Открытие соответствующего окна |
| user | Отцрывается таблица о пользователе | Открытие соответствующего окна | Открытие соответствующего окна |
| рекдактирование | Нажатие кнопки «рекдактирование » на формах | Данные отредактированны | Данные отредактированны |
| Удаление | Нажатие кнопки «Удаление» на формах | Данные удалены | Данные удалены |
| Добавление | Нажатие кнопки «Добавить» на форме заявки | В БД добавлены данные | В БД добавлены данные |

# ЗАКЛЮЧЕНИЕ

В процессе работы над курсовым проектом все поставленные задачи и цель были достигнуты.

Выполнена цель курсового проекта:

Разработка удалённой базы данных ВОЕННОГО комиссариата

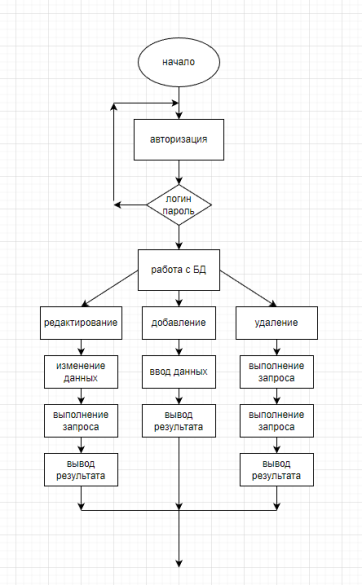
Выполнены поставленные задачи:

* выполнен анализ объекта, процесса, предметной области как объекта информатизации;
* описаны организационную и функциональную схемы объектов информатизации;
* описаны категории пользователей и потоков данных объекта информатизации;
* описана структура данных и процесс проведения ее оптимизации;
* описано клиент-серверное взаимодействие;
* проведен анализ и выбрано программное обеспечение для разработки программы и БД;
* разработана удаленная база данных и таблицы БД;
* разработаны и реализованы запросы к БД;
* произведена разработка интерфейса, работающего с удаленной базой данных;
* проведена отладка и тестирование программного продукта;

Таким образом, гипотеза, на которой был основан курсовой проект, согласно которой автоматизируемый процесс на рассматриваемом предприятии будет максимально эффективен, подтвердилась.

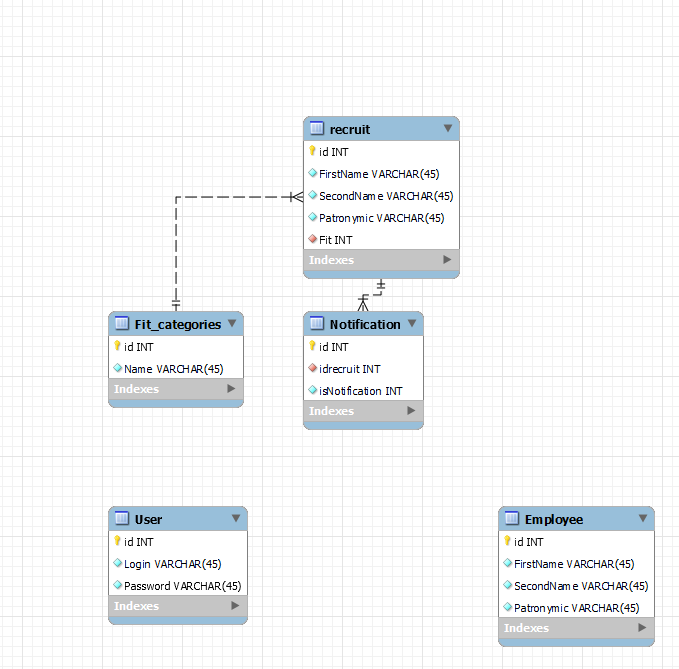
**ПРИЛОЖЕНИЕ А**Структурная схема программного продукта

**Структурная схема программного продукта**



**ПРИЛОЖЕНИЕ Б**Схема базы данных

**Схема базы данных**



**ПРИЛОЖЕНИЕ В**Листинг кода программного продукта

**Листинг кода программного продукта**

using \_336.BdModels;

using \_336.Windows;

using MySqlConnector;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MySqlConnectionStringBuilder build = new MySqlConnectionStringBuilder()

}

private void Author\_Click(object sender, RoutedEventArgs e)

{

if(LoginTB.Text==null&&PassTB.Text==null)

{

MessageBox.Show("Заполните все поля");

}

else

{

User user = CoreModel.init().Users.FirstOrDefault(u=>u.Login==LoginTB.Text&&u.Pass.ToString()==PassTB.Text);

if(user!=null)

{

if(user.Roli=="recruit")

{

MessageBox.Show("You recruit ");

}

else if(user.Roli == "administrator")

{

MessageBox.Show("🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨\r\n🚨Добро пожаловать Администратор🚨\r\n🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨🚨");

}

}

}

Window\_Menu wind = new Window\_Menu();

wind.Show();

this.Close();

}

}

}

using KursovayaISP\_336.Pages;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace KursovayaISP\_336.Windows

{

/// <summary>

/// Interaction logic for Window\_Menu.xaml

/// </summary>

public partial class Window\_Menu : Window

{

public Window\_Menu()

{

InitializeComponent();

}

private void Show\_Tupe(object sender, RoutedEventArgs e)

{

FrameNav.Navigate(new Tupe());

}

private void Show\_Location(object sender, RoutedEventArgs e)

{

FrameNav.Navigate(new Location());

}

private void Show\_Employee(object sender, RoutedEventArgs e)

{

FrameNav.Navigate(new Employee());

}

private void Show\_Tovar(object sender, RoutedEventArgs e)

{

FrameNav.Navigate(new Tovar());

}

private void Show\_User(object sender, RoutedEventArgs e)

{

FrameNav.Navigate(new User());

}

}

}