Министерство образования и науки Нижегородской области

Государственное бюджетное профессиональное   
образовательное учреждение

«Заволжский автомоторный техникум»

(ГБПОУ ЗАМТ)

Специальность: 09.02.03 «Программирование в компьютерных системах»

**Выпускная квалификационная работа**

Студента IV курса группы ПС-19А

**Кузнецова Максима Денисовича**

**Тема:** Разработка автоматизированной информационной системы «Слушатели» отделения дополнительного образования техникума на базе платформы .NEТ

Число листов ВКР (прописью) – \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Число страниц пояснительной записки –\_\_\_\_\_\_ страниц

|  |  |
| --- | --- |
| **Допустить к защите:**  Зам. директора по УР  Мокрышева Л.Г.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | **Руководитель**  выпускной квалификационной работы:  Лапина О.И.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  **Рецензент:**  Хмелева Е.О.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Работа защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

г. Заволжье

2022 г.

Министерство образования и науки Нижегородской области

Государственное бюджетное профессиональное   
образовательное учреждение

«Заволжский автомоторный техникум»

(ГБПОУ ЗАМТ)

Специальность: 09.02.03 «Программирование в компьютерных системах»

Утверждаю

Зам. директора по УР

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мокрышева Л.Г.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 год

**задание**

**на выпускную квалификационную работу**

Студента IV курса группы ПС-19А

**Кузнецова Максима Денисовича**

**Тема выпускной квалификационной работы:**

Разработка автоматизированной информационной системы «Слушатели» отделения дополнительного образования техникума на базе платформы .NEТ

Выпускная квалификационная работа на указанную тему выполняется в следующем объеме:

### Пояснительная записка

Введение

1 Постановка задачи

2 Проектирование

2.1 Обоснование выбора среды разработки

2.2 Выбор способа организации входных и выходных данных

2.3 Проектирование моделей данных

2.4 Алгоритм представления задачи

2.4.1 Детальное проектирование программного обеспечения (прототип)

2.4.2 Описание основных программных модулей

2.5 Применяемые меры защиты информации

3 Тестирование

4 Экономическое обоснование

4.1 Расчет затрат на разработку программы

4.2 Стратегия продвижения программного продукта

4.3 Расчет цены разработанной программы

4.4 Технико-экономические показатели разработки программы

Заключение

Литература

Приложения

#### Дата выдачи задания « 06 » \_\_04\_\_\_\_ 2023 года

#### Дата окончания проекта « 15 » \_\_\_06\_\_ 2023 года

Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Лапина О.И./

Председатель ПЦК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Карпушева И.А./

Содержание

Введение 4

1 Постановка задачи 6

2 Проектирование 8

2.1 Обоснование выбора среды разработки 8

2.2 Выбор способа организации входных и выходных данных 10

2.3 Проектирование моделей данных 19

2.4 Алгоритм представления задачи 24

2.4.1 Детальное проектирование программного обеспечения (прототип) 24

2.4.2 Описание основных программных модулей 31

2.5 Применяемые меры защиты информации 40

3 Тестирование 46

4 Экономическое обоснование 63

4.1 Расчет затрат на разработку программы 65

4.2 Стратегия продвижения программного продукта 67

4.3 Расчет цены разработанной программы 68

4.4 Технико-экономические показатели разработки программы 69

Заключение 70

Литература 71

Приложения

Введение

В современном мире как никогда важны квалифицированные специалисты. Но чтобы такие специалисты были, нужно не менее квалифицированное образование. Таким образованием занимаются специальные организации, например, отделения дополнительного образования.

Проблема у таких отделений заключается в том, что, ведя учёт договоров и платежей, тратится большое количество времени на обработку информации, так как нет централизованной информационной системы. Создание такой информационной системы поможет отделению дополнительного образования тратить меньше времени на учёт различным данных и обработку отчетов, так как работа будет производиться в электронном виде.

Целью выпускной квалификационной работы является создание автоматизированной информационной системы для отделения дополнительного образования техникума на базе платформы .NEТ.

В процессе проектирования необходимо решить следующие задачи:

- сформулировать требования к программе;

- построить структуру базы данных;

- выбрать среду разработки;

- реализовать программный продукт;

- протестировать программный продукт;

- рассчитать технико-экономические показатели;

- разработать техническую и эксплуатационную документацию.

Автоматизированная информационная система позволит автоматизировать деятельность секретаря приемной комиссии в процессах: работы со слушателями, записей на курсы, учета договоров и учета оплат по определенным договорам. Позволит отслеживать статусы договоров, их оплат и освоения курсов слушателями, а также производить подготовку статистических отчетов по договорам и подготовку договоров для печати. Заведующий отделением сможет производить подготовку договоров для печати и подготовку статистических отчетов. Системный администратор сможет быстро подготовить систему к работе с помощью функции импорт для заполнения справочников или базы в целом, а также сможет производить экспорт данных из справочников или базы в целом, и производить работу с учетными записями пользователей и сотрудников.

1. **Постановка задачи**

Разработать автоматизированную информационную систему «Слушатели» отделения дополнительного образования на базе платформы .NET. Для разработки приложения использовать язык программирования высокого уровня. Для хранения информации использовать базу данных по усмотрению разработчика.

База должна содержать информацию: о курсах; слушателях; пользователях АИС; договорах и их содержимом; платежах; сотрудниках отделения; группах.

С АИС должны работать три пользователя: системный администратор, секретарь приемной комиссии и заведующий отделением дополнительного образования.

*Системный администратор* АИС имеет возможность:

1. разграничивать права доступа к разделам базы;
2. добавлять и удалять пользователей АИС;
3. заполнять справочники с помощью импорта данных;
4. восстанавливать базу данных.

*Секретарь приемной комиссии* имеет возможность:

1. заполнять справочную информацию с помощью форм ввода;
2. производить запись слушателей на курсы в определенную группу;
3. формировать договор слушателя и бланка квитанции об оплате для экспорта в приложения MS Word 2007;
4. формировать список группы для экспорта в приложения MS Word 2007;
5. вести учет платежей по договору;
6. вести мониторинг освоения курсов;
7. просматривать форму учета договоров, используя функции поиска, сортировки и фильтрации;
8. управлять статусами договора;
9. формировать статистические отчеты по учету курсов для экспорта в приложения MS Excel;

*Заведующий отделением* должен иметь возможность решать следующие задачи:

1. просматривать форму учета договоров, используя функции поиска, сортировки и фильтрации;
2. формировать различные формы отчетов для экспорта в приложение MS Excel/Word 2007.

Количество форм информационной системы определяет разработчик. В АИС предусмотреть выполнение всех функций, указанных в общих требованиях к программному продукту.

**2 Проектирование**

2.1 Обоснование выбора среды разработки

Для создания данного продукта можно использовать следующие среды разработки: Microsoft Visual Studio версия Professional, JetBrains [Rider](https://code.visualstudio.com/) и [Eclipse](https://www.eclipse.org/).

*Microsoft Visual Studio Professional* – линейка продуктов компании [Microsoft](https://ru.wikipedia.org/wiki/Microsoft), включающих [интегрированную среду разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как [консольные](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) [приложения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), так и приложения с [графическим интерфейсом](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), в том числе с поддержкой технологии [Windows Forms](https://ru.wikipedia.org/wiki/Windows_Forms). В Visual Studio множество качественных плагинов. С их помощью можно расширить функциональность приложения и подключить другие языки. Коммерческая редакция.

[*JetBrains*](https://code.visualstudio.com/) *Rider* – кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Имеет возможность рефакторинга кода и анализатор ошибок в коде, а также удобную навигацию среди файлов. Из недостатков – меньше количество возможностей по сравнению с Visual Studio, встречающиеся ошибки при работе с отладчиком.

[*Eclipse*](https://www.eclipse.org/) – свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation. По сути - это просто среда разработки и набор сервисов для построения приложений на основе встраиваемых компонентов (плагинов). Также содержит хороший компилятор. Недостатки: использование большого количества ресурсов ПК, несовместимость некоторых плагинов с новыми версиями Eclipse, проблемы среды разработки при работе с Git.

Исходя из вышеперечисленных инструментальных средств, была выбрана среда разработки Microsoft Visual Studio Professional, так как данная среда разработки имеет достаточный функционал для написания данного проекта и большее количество возможностей по сравнению с остальными средами разработки.

Данная среда разработки позволяет программировать на следующих языках программирования: С#, C++, Java Script, Visual F# и другие.

*C# -*  это объектно-ориентированный язык программирования, но поддерживает также и компонентно-ориентированное программирование. Он был создан командой инженеров Microsoft. Является популярным языком программирования за счет своей «простоты». Простота работы с данным языком обеспечивает возможность написания функциональных и производительных приложений в сжатые сроки.

*F#* – объектно-ориентированный язык программирования для Microsoft .NET Framework. F# сочетает простой четкий синтаксис со строгой статической типизацией и позволяет не только создавать сравнительно небольшие экспериментальные блоки кода, но и вести крупномасштабную разработку компонентов на основе .NET Framework в Visual Studio.

*C++* - компилируемый статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование, обеспечивает модульность, раздельную компиляцию, обработку исключений, абстракцию данных, объявление типов (классов) объектов, виртуальные функции.

Исходя из выше перечисленных языков программирования для разработки был выбран язык С#, так как это объектно-ориентированный высокоуровневый язык программирования, то есть он подходит для написания графических приложений. Работа с внутренностями .Net не требуется, сроки решения задачи не допускают этого, гораздо больше внимания следует уделить непосредственно задаче, а не техническим деталям. Исходя из этого, предпочтение было отдано в сторону С#, так как использование *C++ или F#* для выполнения выпускной квалификационной работы нерационально.

2.2 Выбор способа организации входных и выходных данных

Для организации входных и выходных данных необходима база данных. Visual Studio позволяет использовать базы данных: MS Access, SQLite, MуSQL и другие.

*MySQL* – свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle. Продукт распространяется как под [GNU General Public License](https://ru.wikipedia.org/wiki/GNU_General_Public_License), так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм [репликации](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)).

*MS Access* – это реляционная система управления базами данных, входящая в пакет MS Office от компании Microsoft. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Недостатки: слабые средства защиты и восстановления информации, ограничения на объем информации, низкая скорость при работе с большими объемами информации.

*SQLite* – это система управления базами данных, отличительной особенностью которой является ее встраиваемость в приложения. Это значит, что большинство СУБД являются самостоятельными приложениями, взаимодействие с которыми организовано по принципу клиент-сервер. Также отличается своей компактностью. Представляет собой небольшую библиотеку на языке C. Недостатки: отсутствие функции безопасности, плохо масштабируется, не подходит для большого количества данных.

Исходя из выше перечисленных вариантов, принято решение использовать реляционную систему управления базами данных *MySQL*, так как ее функционала хватает для выполнения поставленной задачи, в ней существует защита и управление пользователями, а также есть возможность работы с большим количеством данных.

База данных состоит из 13 таблиц, 11 из которых имеют связь в соответствии с рисунком 2.1

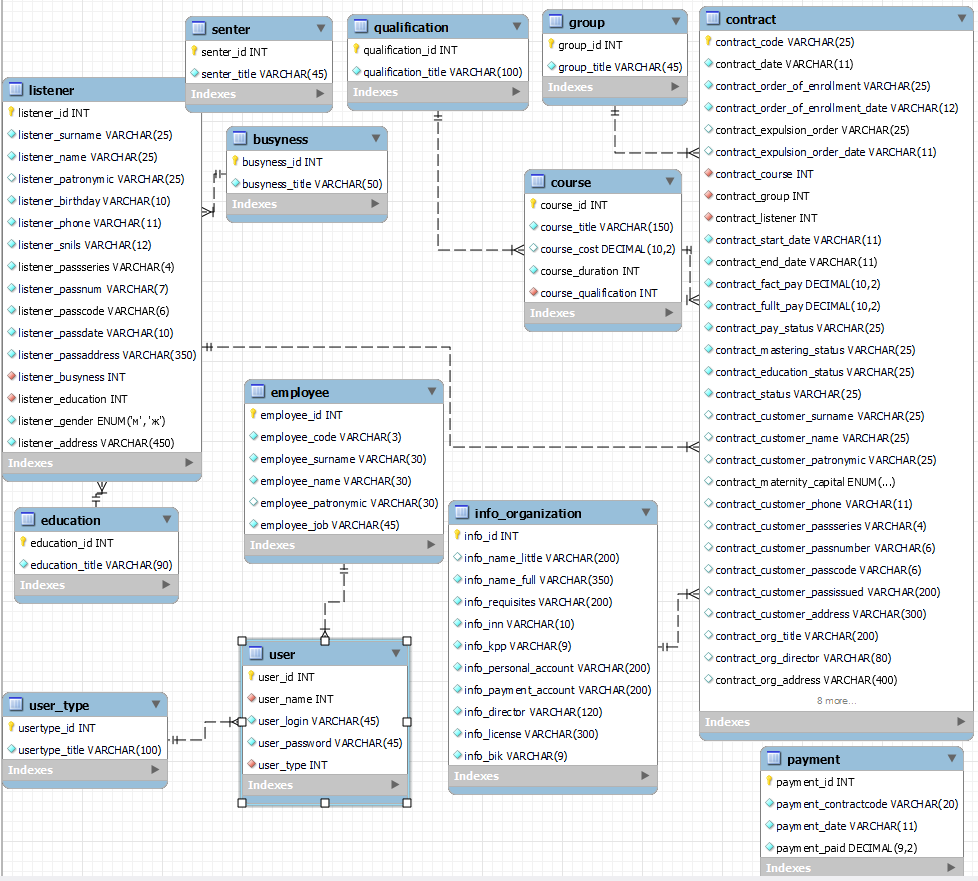


Рисунок 2.1 – Структура базы данных

Структура таблиц базы данных представлена таблицами 2.1-2.13

Таблица 2.1 – Структура таблицы «listener» (Слушатель)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца  (в БД) | Тип данных | Ограничения |
| 1 | 2 | 3 | 4 |
| Идентификатор слушателя | listener\_id | Счетчик | целое число без ограничений, заполняется автоматически |
| Фамилия | listener\_surname | Текстовый | ограничение в 25 знаков |
| Имя | listener\_name | Текстовый | ограничение в 25 знаков |
| Отчество | listener\_patronymic | Текстовый | ограничение в 25 знаков |
| Дата рождения | listener\_birthday | Текстовый | ограничение в 10 знаков |
| Номер телефона | listener\_phone | Текстовый | ограничение в 11 знаков |
| СНИЛС | listener\_snils | Текстовый | ограничение в 12 знаков |
| Серия паспорта | listener\_passseries | Текстовый | ограничение в 4 знаков |
| Номер паспорта | listener\_passnum | Текстовый | ограничение в 7 знаков |
| Код подразделения | listener\_passcode | Текстовый | ограничение в 6 знаков |
| Дата выдачи паспорта | listener\_passdate | Текстовый | ограничение в 10 знаков |
| Кем выдан паспорт | listener\_passaddress | Текстовый | ограничение в 350 знаков |
| Идентификатор занятости | listener\_busyness | Числовой | целое число |
| Идентификатор образования | listener\_education | Числовой | целое число |
| Пол | listener\_gender | Список | только 2 варианта данных (м, ж) |
| Адрес слушателя | listener\_address | Текстовый | ограничение в 450 знаков |

Таблица 2.2 – Структура таблицы «material» (Материалы)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Код контракта | contract\_code | Текстовый | ограничение в 25 знаков |
| Дата оформления договора | contract\_date | Текстовый | ограничение в 11 знаков |
| Номер приказа о зачислении | contract\_order\_of \_enrollment | Текстовый | ограничение в 25 знаков |
| Дата приказа о зачислении | contract\_order\_of \_enrollment\_date | Текстовый | ограничение в 12 знаков |
| Номер приказа об отчислении | contract\_expulsion \_order | Текстовый | ограничение в 25 знаков |
| Дата приказа об отчислении | contract\_expulsion \_order\_date | Текстовый | ограничение в 12 знаков |
| Идентификатор курса | contract\_course | Числовой | целое число |
| Идентификатор группы | contract\_group | Числовой | целое число |
| Идентификатор слушателя | contract\_listener | Числовой | целое число |
| Дата начала обучения | contract\_start\_date | Текстовый | ограничение в 11 знаков |
| Дата окончания обучения | contract\_end\_date | Текстовый | ограничение в 11 знаков |
| Фактическая оплата | contract\_fact\_pay | Десятичное число | ограничение в 10 знаков перед запятой и 2 знака после запятой |
| Полная оплата | contract\_fullt\_pay | Десятичное число | ограничение в 10 знаков перед запятой и 2 знака после запятой |
| Статус оплаты | contract\_pay\_status | Текстовый | ограничение в 25 знаков |
| Статус освоения | contract\_mastering \_status | Текстовый | ограничение в 25 знаков |
| Статус обучения | contract\_education \_status | Текстовый | ограничение в 25 знаков |
| Статус договора | contract\_status | Текстовый | ограничение в 25 знаков |

Продолжение таблицы 2.2

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Фамилия родителя | contract\_customer \_surname | Текстовый | ограничение в 25 знаков |
| Имя родителя | contract\_customer \_name | Текстовый | ограничение в 25 знаков |
| Отчество родителя | contract\_customer \_patronymic | Текстовый | ограничение в 25 знаков |
| Материнский капитал | contract\_maternity \_capital | Список | только строка ‘да’ или ‘нет’ |
| Номер телефона родителя | contract\_customer \_phone | Текстовый | ограничение в 11 знаков |
| Серия паспорта родителя | contract\_customer \_passseries | Текстовый | ограничение в 4 знаков |
| Номер паспорта родителя | contract\_customer \_passnumber | Текстовый | ограничение в 6 знаков |
| Код подразделения паспорта родителя | contract\_customer \_passcode | Текстовый | ограничение в 6 знаков |
| Кем выдан паспорт родителя | contract\_customer \_passissued | Текстовый | ограничение в 200 знаков |
| Адрес родителя | contract\_customer \_address | Текстовый | ограничение в 300 знаков |
| Название организации | contract\_org\_title | Текстовый | ограничение в 200 знаков |
| Директор организации | contract\_org\_director | Текстовый | ограничение в 80 знаков |
| Юридический адрес организации | contract\_org\_address | Текстовый | ограничение в 400 знаков |
| ИНН организации | contract\_org\_inn | Текстовый | ограничение в 12 знаков |
| КПП организации | contract\_org\_kpp | Текстовый | ограничение в 10 знаков |
| БИК организации | contract\_org\_bik | Текстовый | ограничение в 10 знаков |
| Корреспондентский счет организации | contract\_org \_correspondent \_account | Текстовый | ограничение в 20 знаков |

Продолжение таблицы 2.2

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Лицевой счет организации | contract\_org \_payment\_account | Текстовый | ограничение в 20 знаков |
| Банк организации | contract\_org\_bank | Текстовый | ограничение в 250 знаков |
| Идентификатор информации об организации техникума | contrack\_org\_info | Числовой | целое число |
| Дата выдачи паспорта родителя | contract\_customer \_pastdate | Текстовый | ограничение в 11 знаков |

Таблица 2.3 – Структура таблицы «info\_organization» (Информация об организации)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца  (в БД) | Тип данных | Ограничения |
| Идентификатор информации об организации | info\_id | Счетчик | целое число без ограничений, заполняется автоматически |
| Краткое название организации | info\_name\_little | Текстовый | ограничение в 200 знаков |
| Полное название организации | info\_name\_full | Текстовый | ограничение в 350 знаков |
| Реквизиты | info\_requisites | Текстовый | ограничение в 200 знаков |
| ИНН | info\_inn | Текстовый | ограничение в 10 знаков |
| КПП | info\_kpp | Текстовый | ограничение в 9 знаков |
| Лицевой счёт | info\_personal\_account | Текстовый | ограничение в 200 знаков |
| Платежный счёт | info\_payment\_account | Текстовый | ограничение в 200 знаков |
| Директор | info\_director | Текстовый | ограничение в 120 знаков |
| Лицензия | info\_license | Текстовый | ограничение в 300 знаков |
| БИК | info\_bik | Текстовый | ограничение в 9 знаков |

Таблица 2.4 – Структура таблицы «busyness» (Занятость)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца  (в БД) | Тип данных | Ограничения |
| Идентификатор занятости | busyness\_id | Счетчик | целое число, заполняется автоматически |
| Наименование занятости | busyness\_title | Текстовый | ограничение в 50 знаков |

Таблица 2.5 – Структура таблицы «course» (Курсы)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца  (в БД) | Тип данных | Ограничения |
| Идентификатор курса | course\_id | Счетчик | целое число, заполняется автоматически |
| Название курса | course\_title | Текстовый | ограничение в 150 знаков |
| Стоимость курса | course\_cost | Десятичное число | ограничение в 10 знаков перед запятой и 2 знака после запятой |
| Длительность курса | course\_duration | Числовой | целое число |
| Идентификатор курса | course\_qualification | Числовой | целое число |

Таблица 2.6 – Структура таблицы «qualification» (Квалификация)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор квалификации | qualification\_id | Счетчик | целое число, заполняется автоматически |
| Название квалификации | qualification\_title | Текстовый | ограничение в 100 знаков |

Таблица 2.7 – Структура таблицы «group» (Группа)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор группы | group\_id | Счетчик | целое число, заполняется автоматически |
| Название группы | group\_title | Текстовый | ограничение в 45 знаков |

Таблица 2.8 – Структура таблицы «education» (Образование)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор образования | education\_id | Счетчик | целое число, заполняется автоматически |
| Название образования | education\_title | Текстовый | ограничение в 90 знаков |

Таблица 2.9 – Структура таблицы «payment» (Оплата)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца  (в БД) | Тип данных | Ограничения |
| Идентификатор оплаты | payment\_id | Счетчик | целое число, заполняется автоматически |
| Код договора | payment\_ contractcode | Текстовый | ограничение в 20 знаков |
| Дата оплаты | payment\_ date | Текстовый | ограничение в 11 знаков |
| Сумма оплаты | payment\_paid | Десятичное число | ограничение в 9 знаков перед запятой и 2 знака после запятой |

Таблица 2.10 – Структура таблицы «employee» (Сотрудники)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| 1 | 2 | 3 | 4 |
| Идентификатор сотрудника | employee\_id | Счетчик | целое число, заполняется автоматически |

Продолжение таблицы 2.10

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Табельный номер сотрудника | employee\_code | Текстовый | ограничение в 3 знаков |
| Фамилия | employee\_surname | Текстовый | ограничение в 30 знаков |
| Имя | employee\_name | Текстовый | ограничение в 30 знаков |
| Отчество | employee\_patronymic | Текстовый | ограничение в 30 знаков |
| Должность | employee\_job | Текстовый | ограничение в 45 знаков |

Таблица 2.11 – Структура таблицы «user» (Пользователи)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца  (в БД) | Тип данных | Ограничения |
| Идентификатор пользователя | user\_id | Счетчик | целое число, заполняется автоматически |
| Идентификатор сотрудника | user\_name | Числовой | целое число |
| Логин | user\_login | Текстовый | ограничение в 45 знаков |
| Пароль | user\_password | Текстовый | ограничение в 45 знаков |
| Идентификатор типа пользователя | user\_type | Числовой | целое число |

Таблица 2.12 – Структура таблицы «user\_type» (Тип пользователя)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца  (в БД) | Тип данных | Ограничения |
| Идентификатор типа пользователя | usertype\_id | Счетчик | целое число, заполняется автоматически |
| Название типа пользователя | usertype\_title | Текстовый | ограничение в 100 знаков |

Таблица 2.13 – Структура таблицы «senter» (Направитель)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца  (в БД) | Тип данных | Ограничения |
| Идентификатор направителя | senter\_id | Счетчик | целое число, заполняется автоматически |
| Название направителя | senter\_title | Текстовый | ограничение в 45 знаков |

2.3 Проектирование моделей данных

При проектировании моделей данных были разработаны диаграммы: вариантов использования и деятельности.

Диаграммы вариантов использования применяются при бизнес – анализе для моделирования видов работ, выполняемых организацией, и для моделирования функциональных требований к ПС при ее проектировании и разработке.

Диаграмма вариантов использования описывает функциональное назначение системы, т.е. что система будет делать в процессе своего функционирования. Является исходной концептуальной моделью системы в процессе ее проектирования и разработки. Диаграмма вариантов использования cсистемы автоматизированной информационной системы по учету заказов мебельного магазина представленная на рисунке 2.2

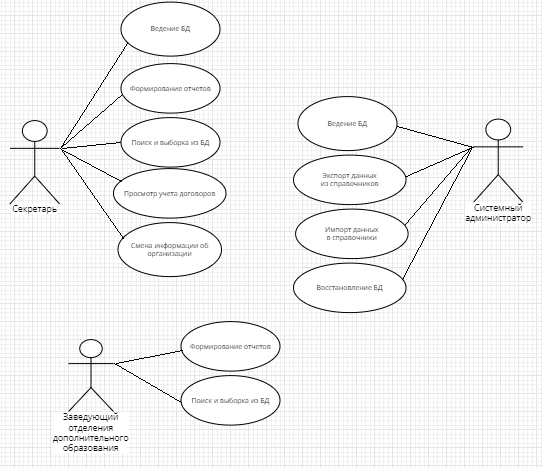


Рисунок 2.2 - Диаграмма вариантов использования

Сценарий использования – вариант использования, прецедент или же пользовательский сценарий в разработке программного обеспечения и системном проектировании. Это описание поведения системы, которым она отвечает на внешние запросы. Другими словами, сценарий использования описывает, «кто» и «что» может сделать с рассматриваемой системой. Методика сценариев использования применяется для выявления требований к поведению системы, известных также как функциональные требования. Сценарий диаграммы представлен в таблице 2.12

Таблица 2.12 Сценарий вариантов использования

|  |  |
| --- | --- |
| Действие актеров | Отклик |
| 1 | 2 |
| Варианты использования | Ведение базы данных и создание отчетов |
| Актеры | Секретарь, системный администратор, заведующий отделения |
| Краткое описание | Вход в АИС в режиме «Секретарь».  Открыть пункт «Слушатели» в главном меню, добавить слушателя, нажать кнопку «Запись на курс», заполнить информацию о договоре, выбрать группу и курс, записать на курс, перейти на форму учета договоров, Выбрать договор и нажать на кнопку «Просмотр договора», нажать на кнопку «Напечатать договор», «Напечатать платежку», «Напечатать соглашение». Сменить данные о датах обучения и данные о приказах и нажать на кнопку «Сменить статусы, данные о приказах и даты обучения». Вход в АИС в режиме «Заведующий». Нажать на «Отчеты», получить статистический отчёт или отчет по оплатам. Нажать на кнопку «Договоры». Выбрать договор и нажать на кнопку «Просмотр договора», нажать на кнопку «Напечатать договор», «Напечатать платежку», «Напечатать соглашение».  Вход в АИС в режиме «Администратор»: Нажать на кнопку «Сотрудники». Добавить сотрудника. Нажать на кнопку «Пользователи». Добавить пользователя. |
| Цель | Автоматизация создания и ведения договоров, слушателей, составления статистических отчётов, договоров, соглашений и платежных документов. |
| Тип | Базовый |

Актер это кто-то или что-то вне системы и влияющий на систему или находящийся под ее влиянием. Актер может быть человеком, устройством, другой системой или подсистемой, или временем. Человек в реальном мире может быть представлен несколькими актерами, если у них есть несколько различных ролей и целей по отношению к системе. Они взаимодействуют с системой и производят над ней некоторые действия.

Краткое описание. Варианты применения сценариев в процессе разработки зависят от используемой методологии разработки. В одних методологиях разработки все, что требуется - это краткий обзор сценария.

Цель – решение поставленных задач. Нет никакой необходимости в сценарии использования, когда нет никакой потребности ни в каком актере, чтобы достигнуть цели. Цель кратко описывает то, чего пользователь намеревается достигнуть с этим сценарием, в соответствии с таблицей 2.13

Таблица 2.13 Последовательность действий актеров- администратор

|  |  |
| --- | --- |
| Действия актеров | Отклик |
| 1. запускает программу | 1. Открытие формы авторизации | |
| 1. вводит логин и пароль и нажимает кнопку «Вход» | 1. Открытие главного меню | |
| 1. нажимает на кнопку «Сотрудники» | 1. Открытие формы сотрудников | |
| 1. заполняет данные о новом сотруднике и нажимает на кнопку «Добавить» | 1. Добавление сотрудника | |
| 1. выбирает сотрудника из таблицы | 1. Заполнение текстовых полей данными о сотруднике | |
| 1. изменяет данные о сотруднике и нажимает на кнопку «Редакитровать» | 1. Данные о сотруднике редактируются | |
| 1. выбирает сотрудника и нажимает на кнопку «Удалить» | 1. Вывод сообщение о предупреждении о удалении | |
| 1. соглашается с удалением | 1. Удаление сотрудника | |
| 1. нажимает на кнопку «В меню» | 1. Открытие главного меню | |
| 1. нажимает на кнопку «Пользователи» | 1. Открытие формы Пользователей | |
| 1. заполняет данные о новом пользователе и нажимает на кнопку «Добавить» | 1. Добавление нового пользователя | |
| 1. выбирает пользователя из таблицы | 1. Заполнение текстовых полей данными о пользователе | |

Диаграмма деятельности – UML-диаграмма, на которой показано разложение некоторой деятельности на её составные части. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчиненных элеемнтов – вложенных видов деятельности и отдельных действий, соединенных между собой потоками, которые идут от выходов одного узла к входам другого.

Диаграмма деятельности используется при моделированиии бизнес-процессов, технологических процессов, последовательных и параллельных вычислений. Диаграмма деятельности программного продукта продемонстрирована на рисунке 2.3

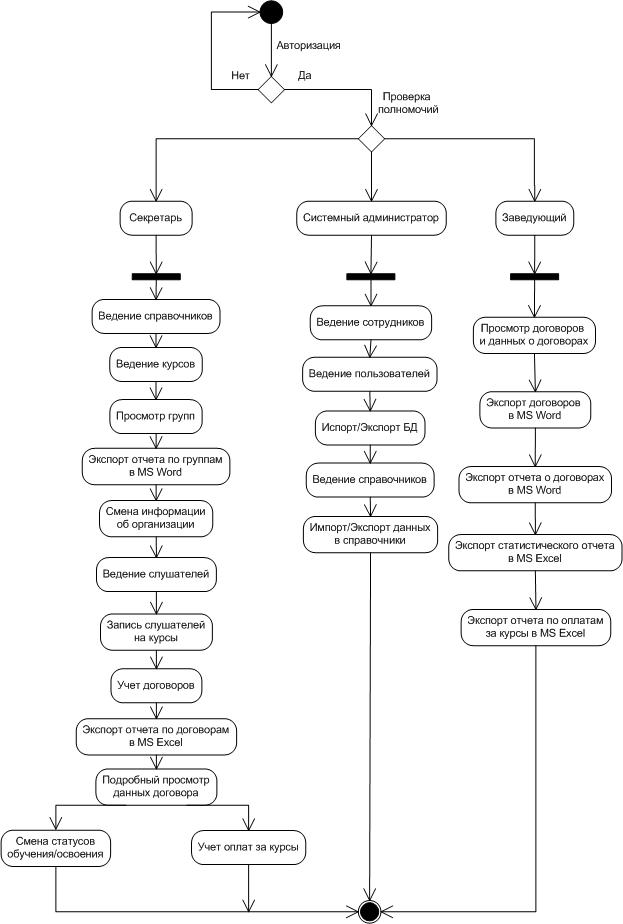


Рисунок 2.3 - Диаграмма деятельности

2.4 Алгоритм представления задачи

2.4.1 Детальное проектирование пользовательского интерфейса

Детальное проектирование ПО – это подробное описание компонентов программного обеспечения и интерфейсов между ними.

Прототипирование программного обеспечения – этап разработки программного обеспечения, процесс создания прототипа программы – макета (черновой, пробной версии) программы, обычно – с целью проверки пригодности предлагаемых для применения концепций, архитектурных или технологических решений, а также для представления программы заказчику на ранних стадиях процесса разработки.

На рисунке 2.4 показаны формы аторизации

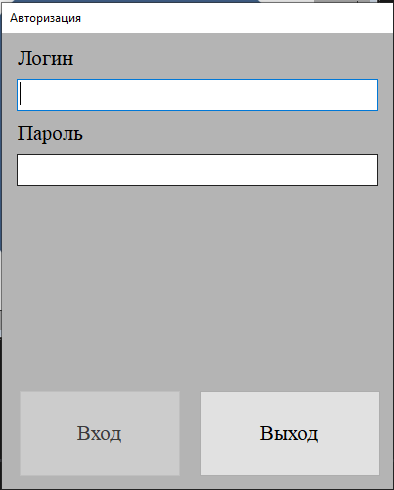
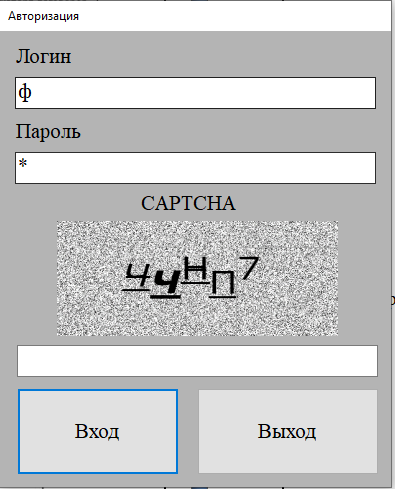
 

Рисунок 2.4 – Формы авторизации

На рисунке 2.5 показано главное меню (Системного администратора)

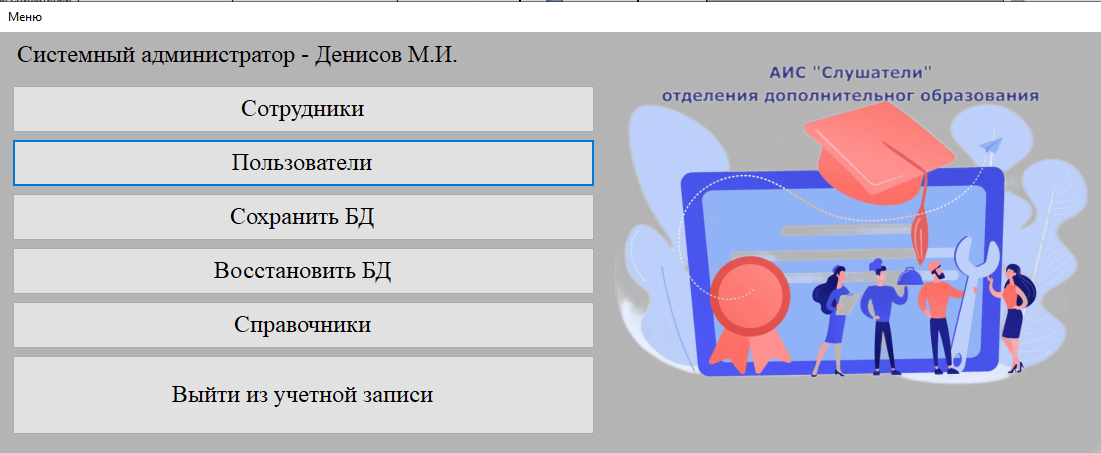


Рисунок 2.5 – Главное меню (Системного администратора)

На рисунке 2.6 показано главное меню (Секретаря)



Рисунок 2.6 – Главное меню (Секретаря)

На рисунке 2.7 показано главное меню (Заведующего)

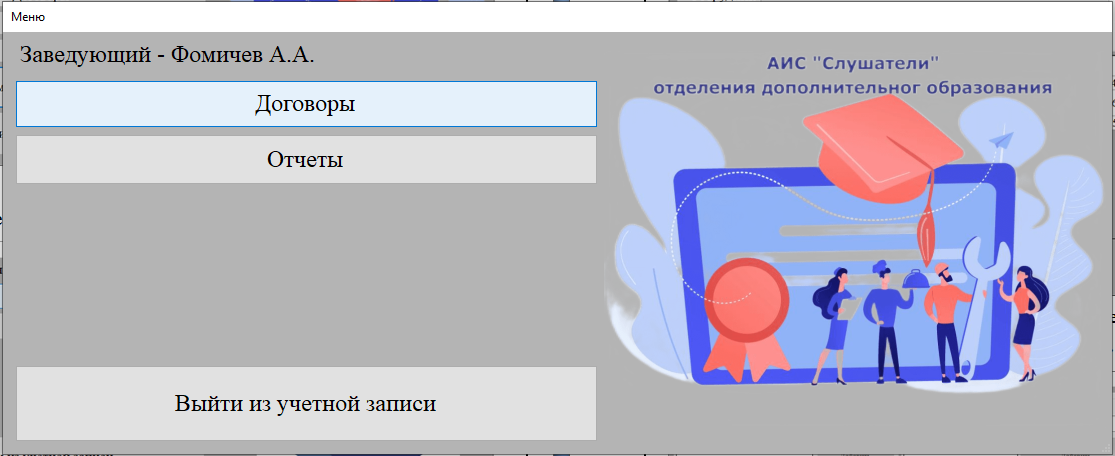


Рисунок 2.7 – Главное меню (Заведующего)

На рисунке 2.8 показана форма «Сотрудники»

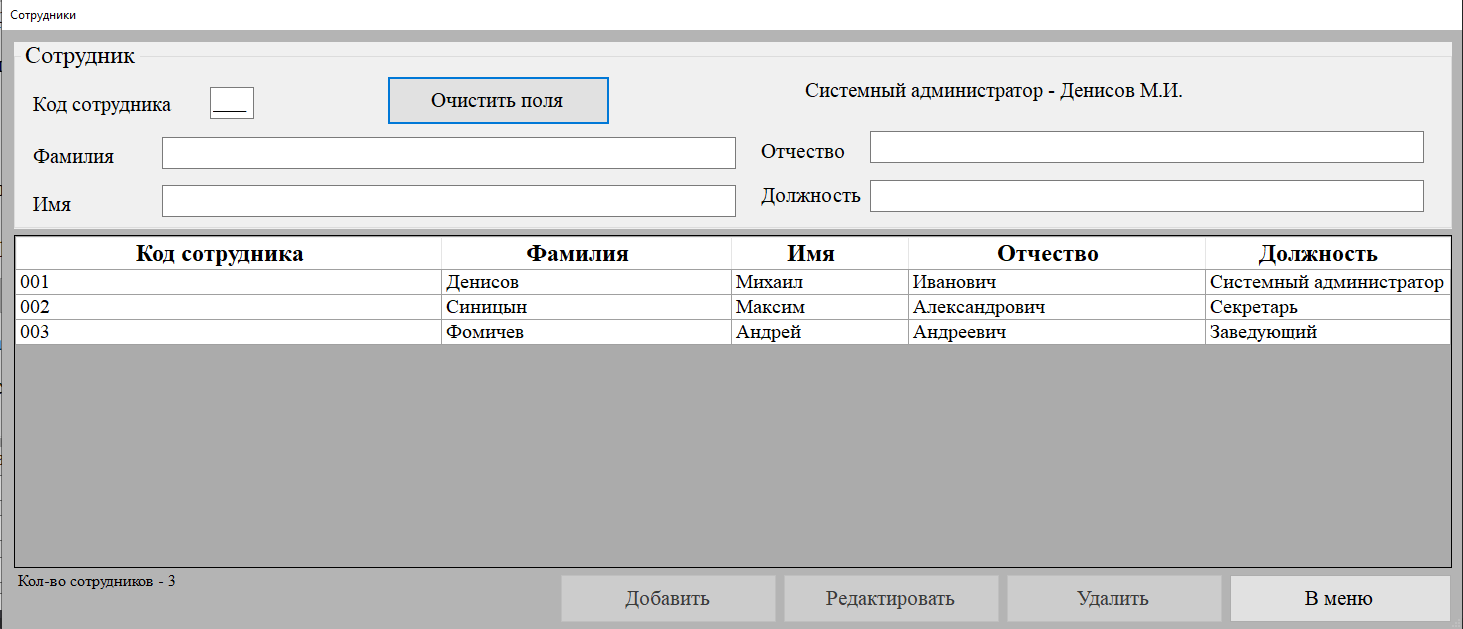


Рисунок 2.8 – Форма «Сотрудники»

На рисунке 2.9 показана форма «Пользователи»

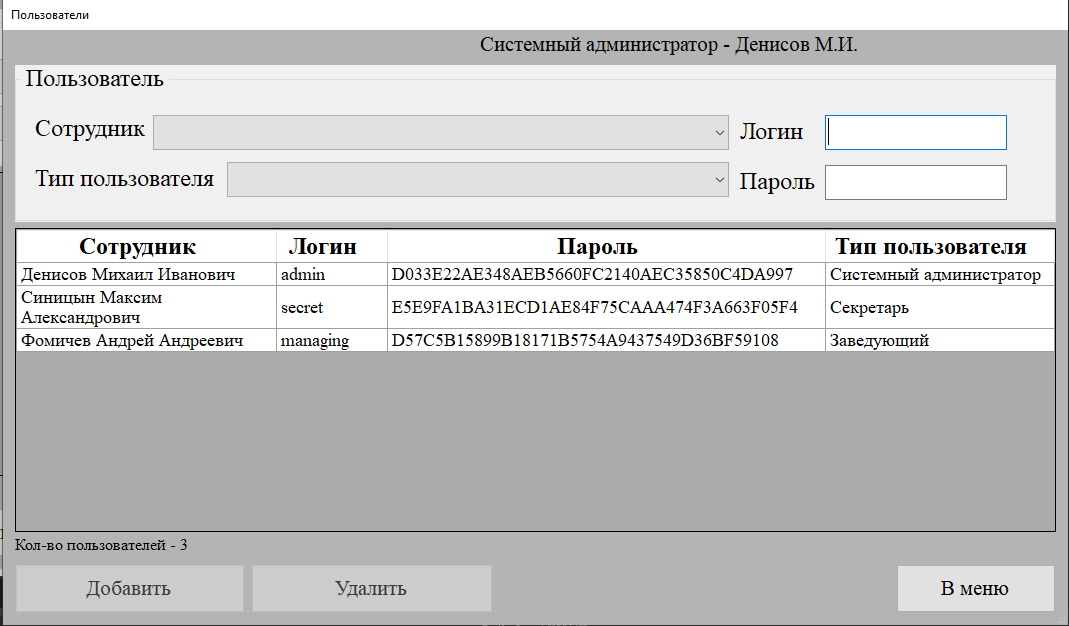


Рисунок 2.9 – Форма «Пользователи»

На рисунке 2.10 показана форма «Справочники»

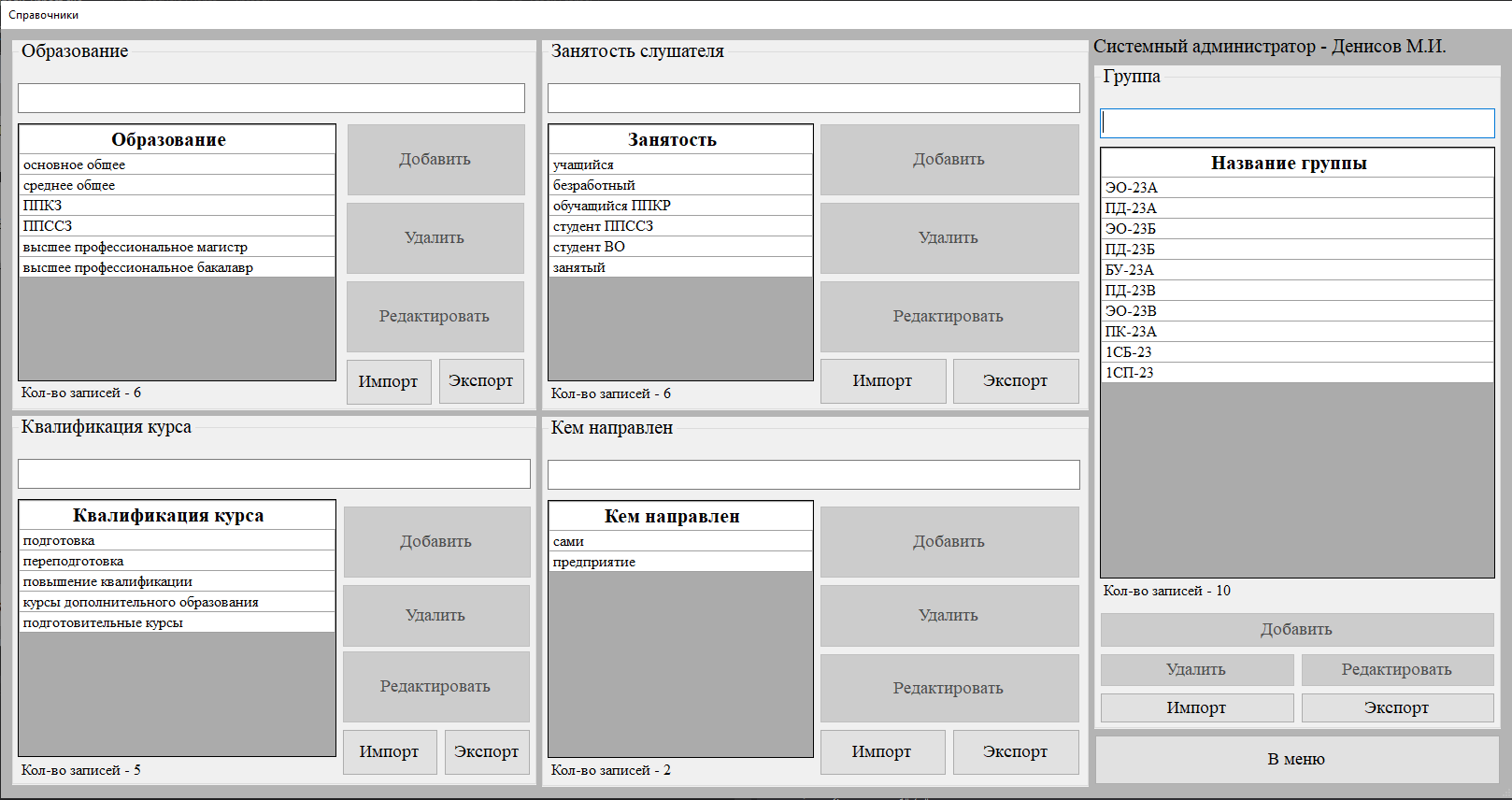


Рисунок 2.10 – Форма «Справочники»

На рисунке 2.11 показана форма «Слушатели»

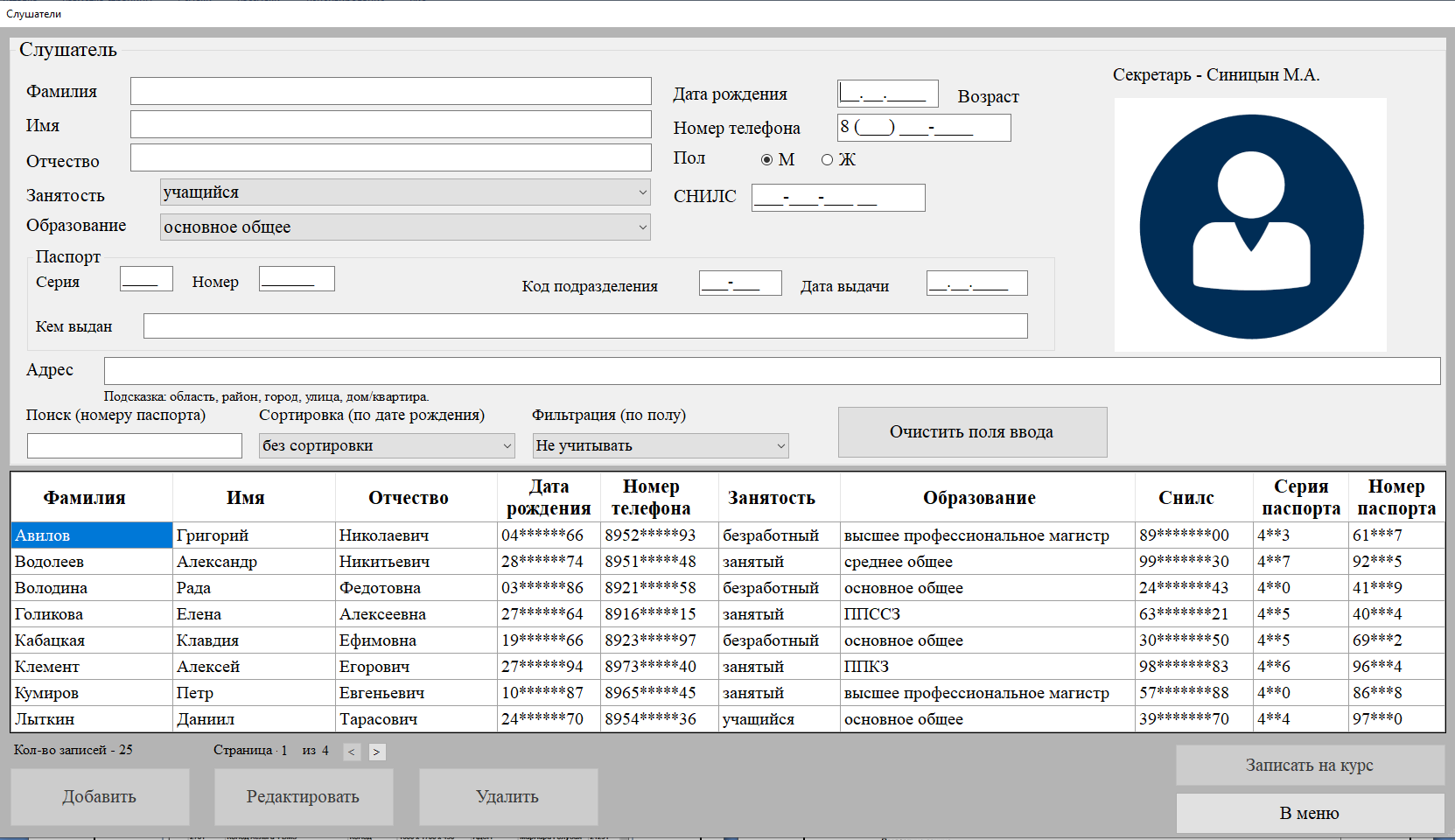


Рисунок 2.11 – Форма «Слушатели»

На рисунке 2.12 показана форма «Запись на курс»

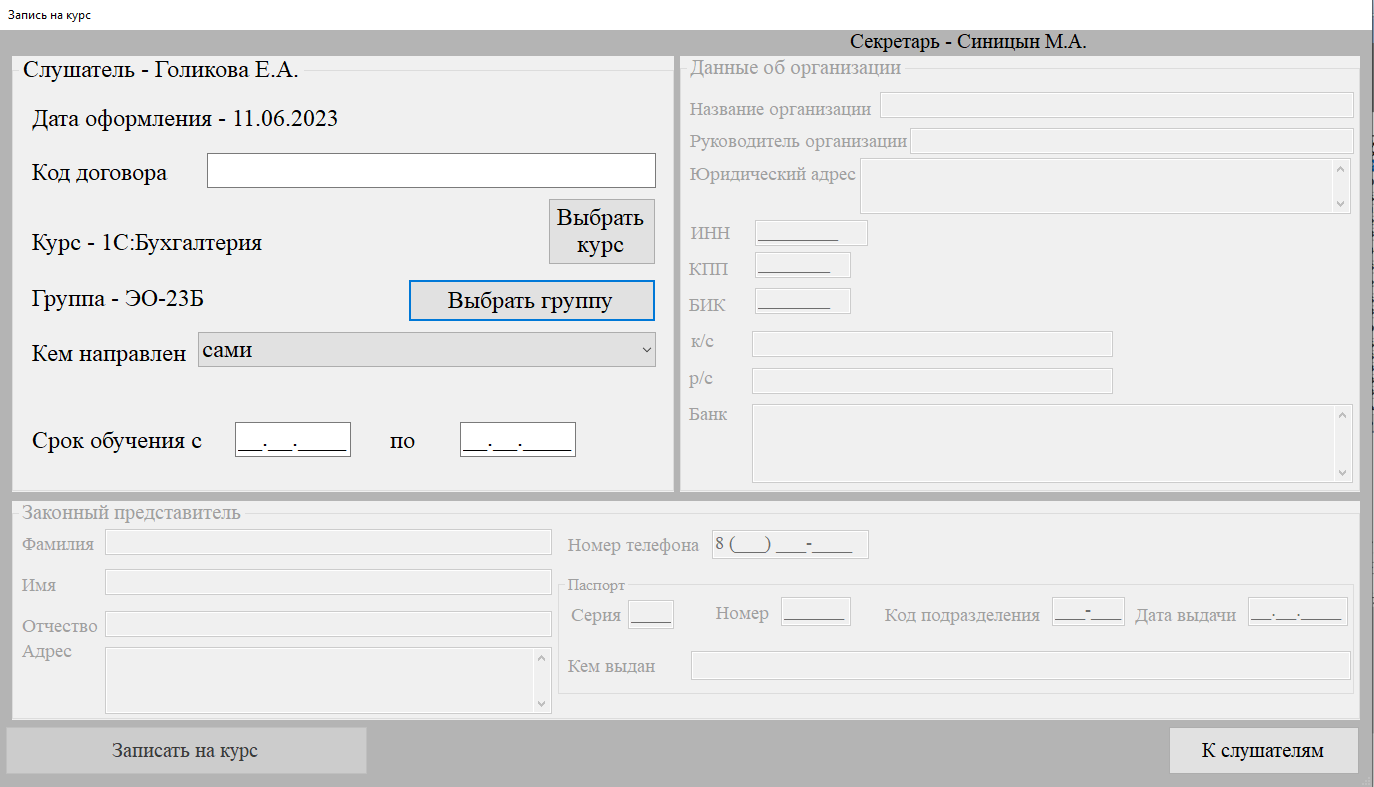


Рисунок 2.12 – Форма «Запись на курс»

На рисунке 2.13 показана форма «Учет договоров»

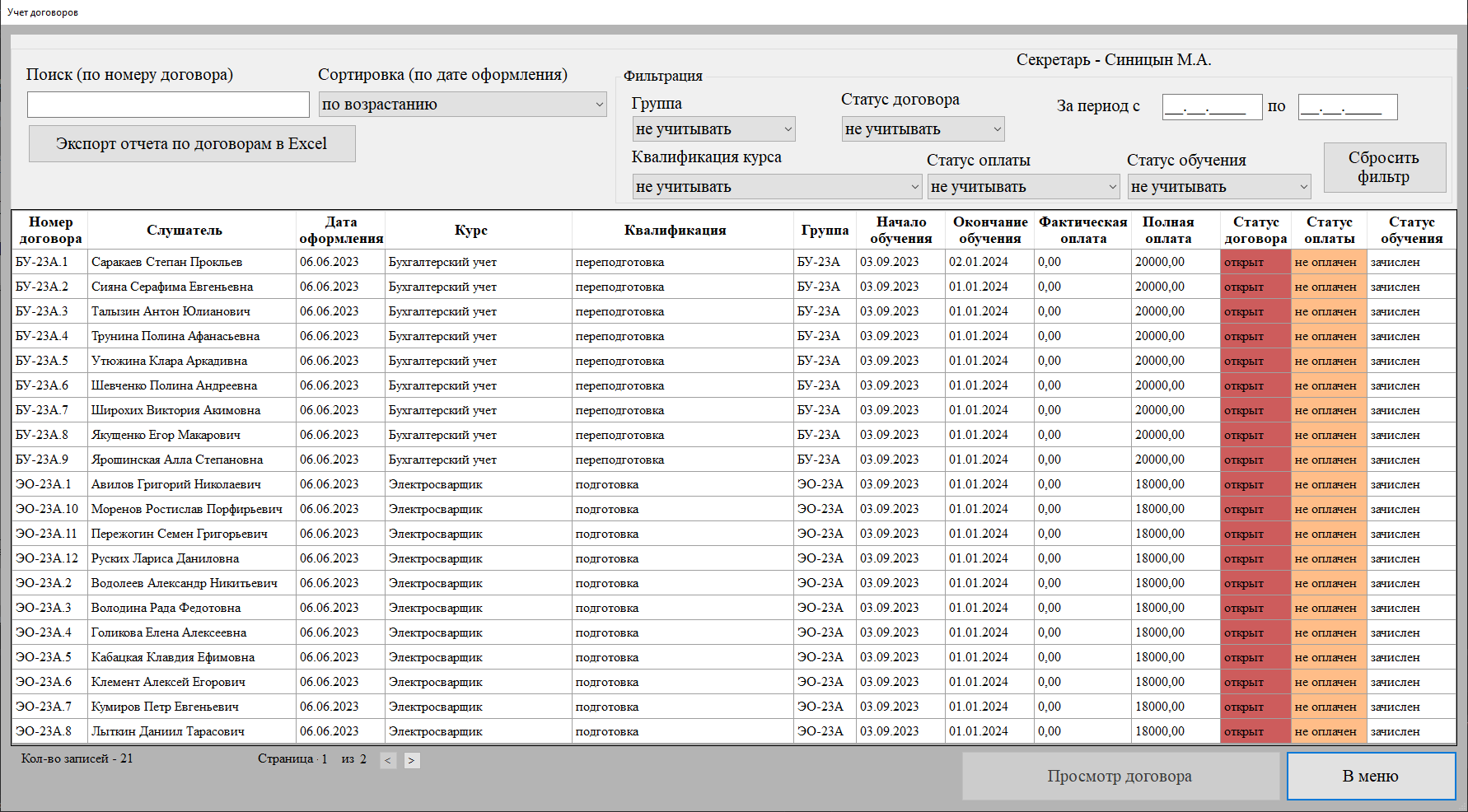


Рисунок 2.13 – Форма «Учет договоров»

На рисунке 2.14 показана форма «Просмотр договора»

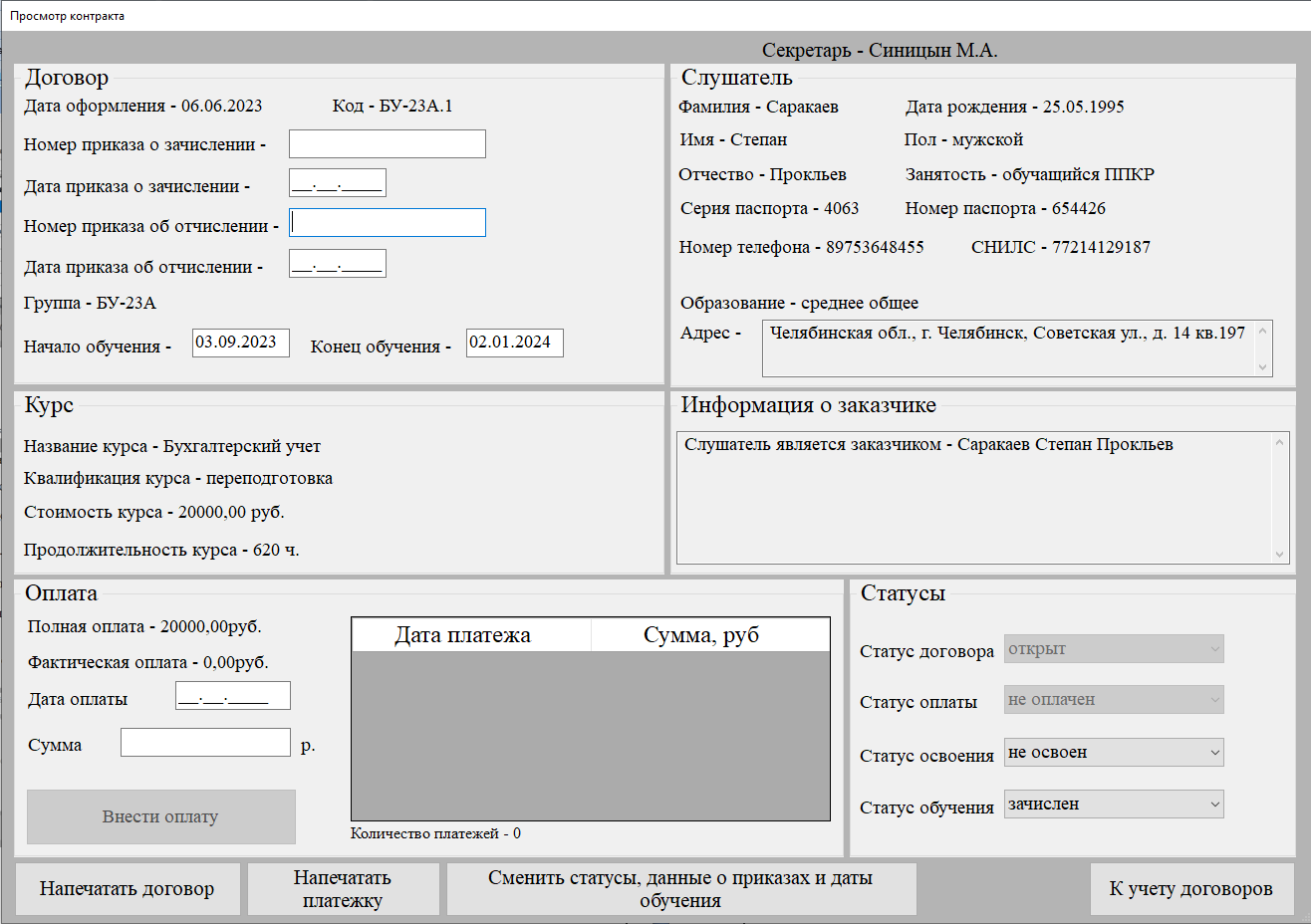


Рисунок 2.14 – Форма «Просмотр договора»

На рисунке 2.15 показана форма «Курсы»

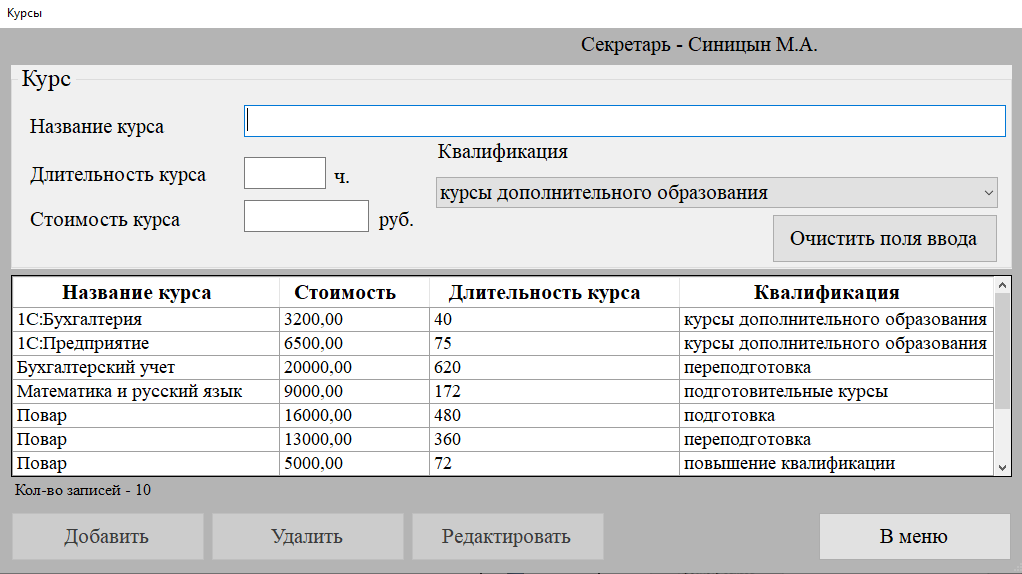


Рисунок 2.15 – Форма «Курсы»

На рисунке 2.16 показана форма «Группы»

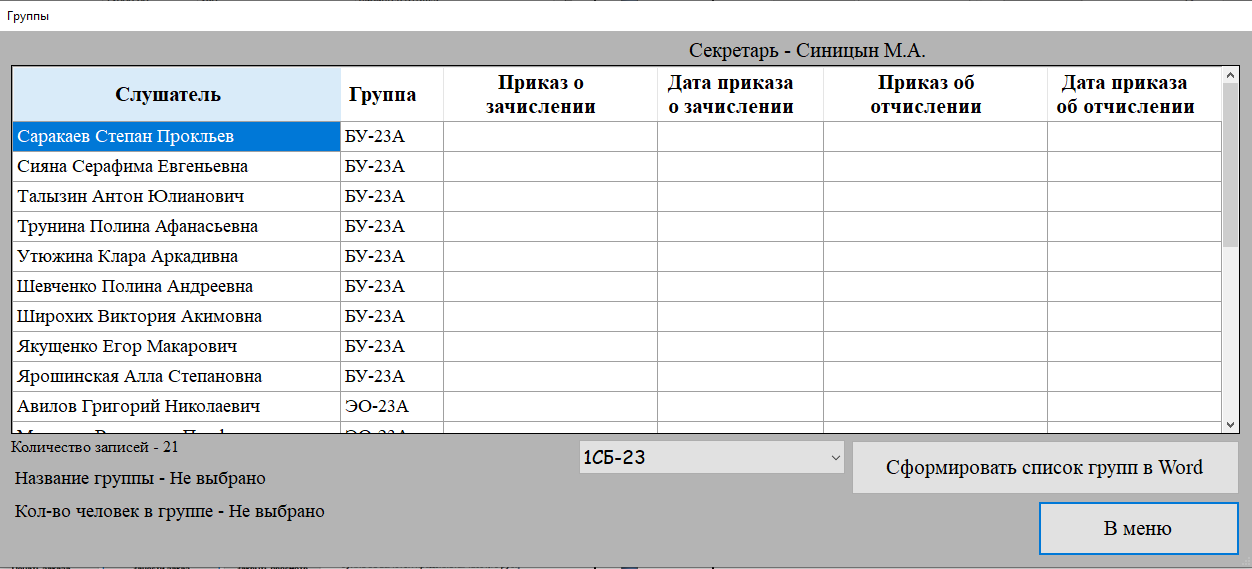


Рисунок 2.16 – Форма «Группы»

На рисунке 2.17 показана форма «Информация об организации»

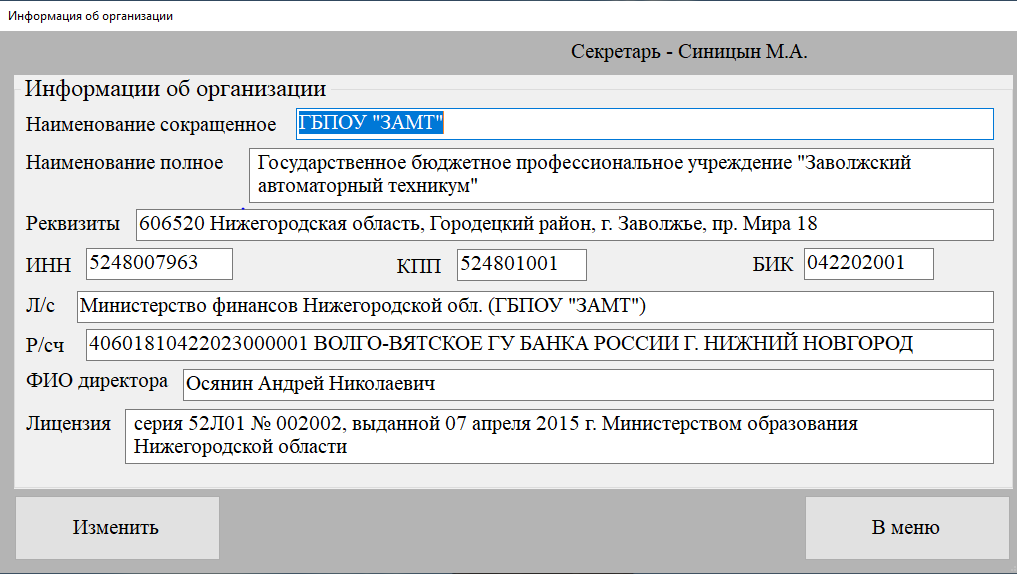


Рисунок 2.17 – Форма «Информация об организации»

На рисунке 2.18 показана форма «Отчеты»

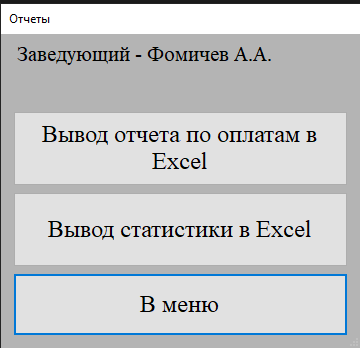


Рисунок 2.18 – Форма «Отчеты»

2.4.2 Описание основных программных модулей

private void button1\_Click(object sender, EventArgs e) – функция проверка логина, пароля и CAPTCHA, а также функция входа;

static string Hash(string input) – функция хэширования пароля;

private void clear() – функция очистки данных на форме авторизации;

private void captchaupd() –функция обновления фото в тесте CAPTCHA;

private void button2\_Click(object sender, EventArgs e) – функция выхода с формы авторизации;

private void authorization\_Load(object sender, EventArgs e) – загрузка формы авторизации;

private void wait() – функция блокировки формы и запуск таймера;

private void Counter(Object myObject, EventArgs myEventArgs) – функция таймера на блокировку формы (10 секунд);

private void button6\_Click(object sender, EventArgs e) – экспорт БД в файл .sql;

private void button5\_Click(object sender, EventArgs e) – импорт БД в файл .sql;

private void button4\_Click(object sender, EventArgs e) – функция выхода с главного меню администратора;

private void Counter1(Object myObject, EventArgs myEventArgs) – функция таймера на бездействие;

private void menu\_adm\_Load(object sender, EventArgs e) – загрузка главного меню администратора;

private void button3\_Click(object sender, EventArgs e) – функция перехода на форму сотрудников;

private void button1\_Click(object sender, EventArgs e) – функция перехода на форму пользователей;

private void button2\_Click(object sender, EventArgs e) - функция перехода на форму справочников;

private void button2\_Click(object sender, EventArgs e) – функция выхода с главного меню заведующего;

private void menu\_head\_Load(object sender, EventArgs e) – загрузка главного меню заведующего;

private void button7\_Click(object sender, EventArgs e) – функция перехода на форму учета договоров;

private void button1\_Click(object sender, EventArgs e) – функция перехода на форму отчетов;

private void button6\_Click(object sender, EventArgs e) – функция перехода на форму учета договоров;

private void button2\_Click(object sender, EventArgs e) – функция выхода с главного меню секретаря;

private void menu\_secretary\_Load(object sender, EventArgs e) – загрузка главного меню секретаря;

private void button3\_Click(object sender, EventArgs e) – функция перехода на форму слушателей;

private void button5\_Click(object sender, EventArgs e) – функция перехода на форму справочников;

private void button7\_Click(object sender, EventArgs e) – функция перехода на форму курсов;

private void button4\_Click(object sender, EventArgs e) – функция перехода на форму групп;

private void button1\_Click(object sender, EventArgs e) – функция перехода на форму информации об организации;

private void contracts\_Load(object sender, EventArgs e) – загрузка формы учета договоров;

private void upddg() – функция обновления таблицы с договорами с учетом фильтрации и сортировки;

private void maskedTextBox3\_TextChanged(object sender, EventArgs e) – функция запуска фильтрации;

private void maskedTextBox1\_TextChanged(object sender, EventArgs e) – функция запуска фильтрации;

private void button3\_Click(object sender, EventArgs e) – выход с формы учета договоров;

private void search2() – функция поиска по номеру договора;

private void comboBox5\_SelectedIndexChanged(object sender, EventArgs e) – функция запуска сортировки по дате оформления;

private void button6\_Click(object sender, EventArgs e) – функция сброса фильтрации;

private void comboBox7\_SelectedIndexChanged(object sender, EventArgs e) – функция запуска фильтрации;

private void comboBox6\_SelectedIndexChanged(object sender, EventArgs e) - функция запуска фильтрации;

private void comboBox8\_SelectedIndexChanged(object sender, EventArgs e) - функция запуска фильтрации;

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e) - функция запуска фильтрации;

private void button7\_Click(object sender, EventArgs e) – функция перехода на форму подробного просмотра договора;

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e) – функция выделения договора;

private void button4\_Click(object sender, EventArgs e) – функция перехода на предыдущую страницу в учете договоров;

private void button2\_Click(object sender, EventArgs e) – функция перехода на следующую страницу в учете договоров;

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e) - функция запуска фильтрации;

private void button1\_Click(object sender, EventArgs e) – функция вывода отчета по договорам в MS EXCEL 2007;

private void courses\_Load(object sender, EventArgs e) – загрузка формы курсов;

private void button4\_Click(object sender, EventArgs e) – функция выхода с формы курсов;

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e) – функция выделения курса;

private void upddg() – функция заполнения таблицы курсов;

private void button3\_Click(object sender, EventArgs e) – функция удаления курса;

private void button1\_Click(object sender, EventArgs e) – функция добавления курса;

private void button2\_Click(object sender, EventArgs e) – функция редактирования курса;

private void button5\_Click(object sender, EventArgs e) – функция выхода с формы курсов;

private void button8\_Click(object sender, EventArgs e) – функция очистки данных на форме курсы;

private void directory\_Load(object sender, EventArgs e) – загрузка формы справочников;

private void filldg1() – функция заполнения таблицы групп;

private void filldg2() – функция заполнения таблицы образования;

private void filldg3() – функция заполнения таблицы занятости;

private void filldg4() – функция заполнения таблицы квалификации;

private void filldg5() – функция заполнения таблицы отправителя;

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e) – функция выделения группы;

private void dataGridView2\_CellClick(object sender, DataGridViewCellEventArgs e) – функция выделения образования;

private void dataGridView3\_CellClick(object sender, DataGridViewCellEventArgs e) – функция выделения занятости слушателя;

private void dataGridView4\_CellClick(object sender, DataGridViewCellEventArgs e) – функция выделения квалификации;

private void dataGridView5\_CellClick(object sender, DataGridViewCellEventArgs e) – функция выделения отправителя;

private void button21\_Click(object sender, EventArgs e) – функция выхода с формы справочников;

private void button1\_Click(object sender, EventArgs e) – функция добавления группы;

private void button10\_Click(object sender, EventArgs e) – функция добавления образования;

private void button15\_Click(object sender, EventArgs e) – функция добавления занятости;

private void button20\_Click(object sender, EventArgs e) – функция добавления квалификации;

private void button26\_Click(object sender, EventArgs e) – функция добавления отправителя;

private void button3\_Click(object sender, EventArgs e) – функция удаления группы;

private void button8\_Click(object sender, EventArgs e) – функция удаления образования;

private void button13\_Click(object sender, EventArgs e) – функция удаления занятости;

private void button18\_Click(object sender, EventArgs e) – функция удаления квалификации;

private void button24\_Click(object sender, EventArgs e) – функция удаления отправителя;

private void button2\_Click(object sender, EventArgs e) – функция редактирования группы;

private void button9\_Click(object sender, EventArgs e) – функция редактирования образования;

private void button14\_Click(object sender, EventArgs e) – функция редактирования занятости;

private void button19\_Click(object sender, EventArgs e) – функция редактирования квалификации;

private void button25\_Click(object sender, EventArgs e) – функция редактирования отправителя;

private void button17\_Click(object sender, EventArgs e) – функция импорта квалификации из файла .csv;

private void button16\_Click(object sender, EventArgs e) – функция экспорта квалификации в файл .csv;

private void button7\_Click(object sender, EventArgs e) – функция импорта образования из файла .csv;

private void button6\_Click(object sender, EventArgs e) – функция экспорта образования в файл .csv;

private void ExportF(string Name) – функция экспорта в файл .csv:

private void button11\_Click(object sender, EventArgs e) – функция экспорта занятости в файл .csv;

private void button5\_Click(object sender, EventArgs e) – функция экспорта групп в файл .csv;

private void button22\_Click(object sender, EventArgs e) – функция экспорта отправителя в файл .csv;

private void ImportF(string Name) – функция импорта из файла .csv;

private void button12\_Click(object sender, EventArgs e) – функция импорта занятости из файла .csv;

private void button23\_Click(object sender, EventArgs e) – функция импорта отправителя из файла .csv;

private void button4\_Click(object sender, EventArgs e) – функция импорта групп из файла .csv;

private void employee\_Load(object sender, EventArgs e) – загрузка формы сотрудников;

private void upddg() – функция обновления таблицы сотрудников;

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e) – функция выделения сотрудника;

private void clear() – функция очистки данных на форме сотрудников;

private void button1\_Click(object sender, EventArgs e) – функция добавления сотрудника;

private void button3\_Click(object sender, EventArgs e) – функция удаления сотрудника;

private void button2\_Click(object sender, EventArgs e) – функция редактирования сотрудника;

private void button4\_Click(object sender, EventArgs e) – функция выхода с формы сотрудников;

private void button5\_Click(object sender, EventArgs e) – очистка данных на форме сотрудников;

private void group\_Load(object sender, EventArgs e) – загрузка формы групп;

private void button2\_Click(object sender, EventArgs e) – выход с формы групп;

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e) – функция выделения группы;

private void button3\_Click(object sender, EventArgs e) – функция выхода с формы групп;

private void button1\_Click(object sender, EventArgs e) – функция вывода списка групп в MS WORD 2007;

private void information\_org\_Load(object sender, EventArgs e) – загрузка формы информации об организации;

private void button1\_Click(object sender, EventArgs e) – функция изменения информации об организации;

private void button3\_Click(object sender, EventArgs e) – функция выхода с формы информации об организации;

private void listener\_Load(object sender, EventArgs e) – загрузка формы слушателей;

private void upddg() – заполнение таблицы слушателей с учетом фильтрации и сортировки;

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e) – функция сортировки по дате рождения;

private void comboBox4\_SelectedIndexChanged(object sender, EventArgs e) – функция фильтрации по полу;

private void sear() – функция поиска по номеру паспорта;

private void button4\_Click(object sender, EventArgs e) – функция выхода с формы слушателей;

private void button3\_Click(object sender, EventArgs e) – функция удаления слушателя;

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e) – функция выделения слушателя;

private void button1\_Click(object sender, EventArgs e) – функция добавления слушателя;

private void button2\_Click(object sender, EventArgs e) – функция редактирования слушателя;

private void button5\_Click(object sender, EventArgs e) – функция перехода на форму записи на курс;

private void button6\_Click(object sender, EventArgs e) – функция перехода на предыдущую страницу;

private void button7\_Click(object sender, EventArgs e) – функция перехода на следующую страницу;

private void button8\_Click(object sender, EventArgs e) – функция очистки данных на форме слушателей;

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e) – функция определения отправителя слушателя;

private void new\_contract\_Load(object sender, EventArgs e) – загрузка формы записи на курс;

private void button4\_Click(object sender, EventArgs e) – функция выхода с формы записи на курс;

private void button2\_Click(object sender, EventArgs e) – функция выбора курса;

private void button3\_Click(object sender, EventArgs e) – функция выбора группы;

private void button1\_Click(object sender, EventArgs e) – функция записи на курс;

private void report\_Load(object sender, EventArgs e) – загрузка формы отчетов:

private void button2\_Click(object sender, EventArgs e) – функция выхода с формы отчетов;

private void button3\_Click(object sender, EventArgs e) – функция вывода статистического отчета;

private void button1\_Click(object sender, EventArgs e) – функция вывода отчета по оплатам;

private void user\_Load(object sender, EventArgs e) – загрузка формы пользователей:

static string Hash(string input) – функция хэширования паролей;

private void upddg() – функция обновления таблицы пользователей;

private void clear() – функция очистки данных на форме пользователей;

private void button3\_Click(object sender, EventArgs e) – функция удаление пользователя:

private void button1\_Click(object sender, EventArgs e) – функция добавления пользователя;

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e) – функция выделения пользователя;

private void button4\_Click(object sender, EventArgs e) – функция выхода с формы пользователей;

private void view\_contract\_Load(object sender, EventArgs e) – загрузка формы подробного просмотра договора;

public void upddate() – функция обновления данных о договоре;

private void button1\_Click(object sender, EventArgs e) – функция внесения оплаты;

private void button6\_Click(object sender, EventArgs e) – функция выхода с формы подробного просмотра договора;

private void button3\_Click(object sender, EventArgs e) - функция смены статусов, данных о приказах;

private void button4\_Click(object sender, EventArgs e) – функция вывода договора в MS WORD 2007;

private void button2\_Click(object sender, EventArgs e) – функция вывода платежного документа;

private void ReplaceWord(string stubSTR, string text, Word.Document wordDocument) – функция забора данных в отчеты;

private void button5\_Click(object sender, EventArgs e) – функция вывода соглашения в MS WORD 2007.

2.5 Применяемые меры защиты информации

Защита данных – это предупреждение несанкционированного (случайного или умышленного) доступа к данным, их изменения или разрушения данных при сбоях аппаратных или программных средств и ошибках в работе сотрудников группы эксплуатации.

Функции:

* обеспечение безопасности данных;
* обеспечение секретности данных.

В процессе разработки информационной системы были реализованы следующие меры защиты информации:

*1. парольная защита базы данных.*

Парольная защита необходима для обеспечения ее защиты от несанкционированного доступа в соответствии с рисунком 2.19.

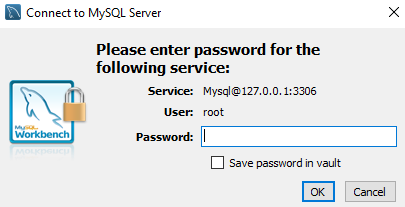


Рисунок 2.19 Парольная защита базы данных

2. *установки прав доступа,* для того чтобы разграничить доступ пользователям к ресурсам базы. Для этого необходимо поддерживать два фундаментальных принципа: проверку полномочий и проверку подлинности (аутентификацию). Добавлены дополнительные степени защиты: наличие capcha при неудачном вводе логина и пароля и блокировка на 10 секунд при повторной неудачной попытке, в соответствии с рисунками 2.20

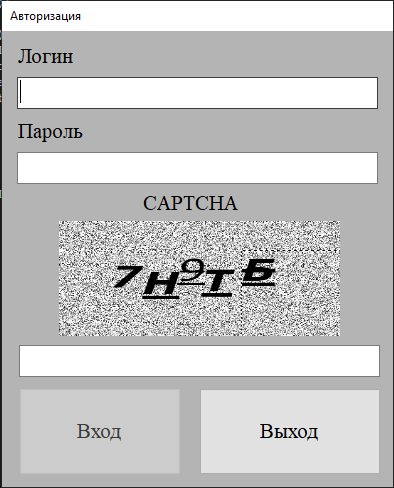


Рисунок 2.20 – Парольная защита программы

Проверка полномочий основана на том, что каждому пользователю или процессу информационной системы соответствует набор действий, которые он может выполнять по отношению к определенным объектам. Проверка подлинности означает достоверное подтверждение того, что пользователь или процесс, пытающийся выполнить санкционированное действие, действительно тот, за кого он себя выдает.

3. *скрытие персональных данных на форме –* одно из направлений, на которое обращает внимание закон «О защите персональных данных» и обращение президента РФ. Применено скрытие данных в таблице и полях ввода на форме Клиенты в соответствии с рисунком 2.21.

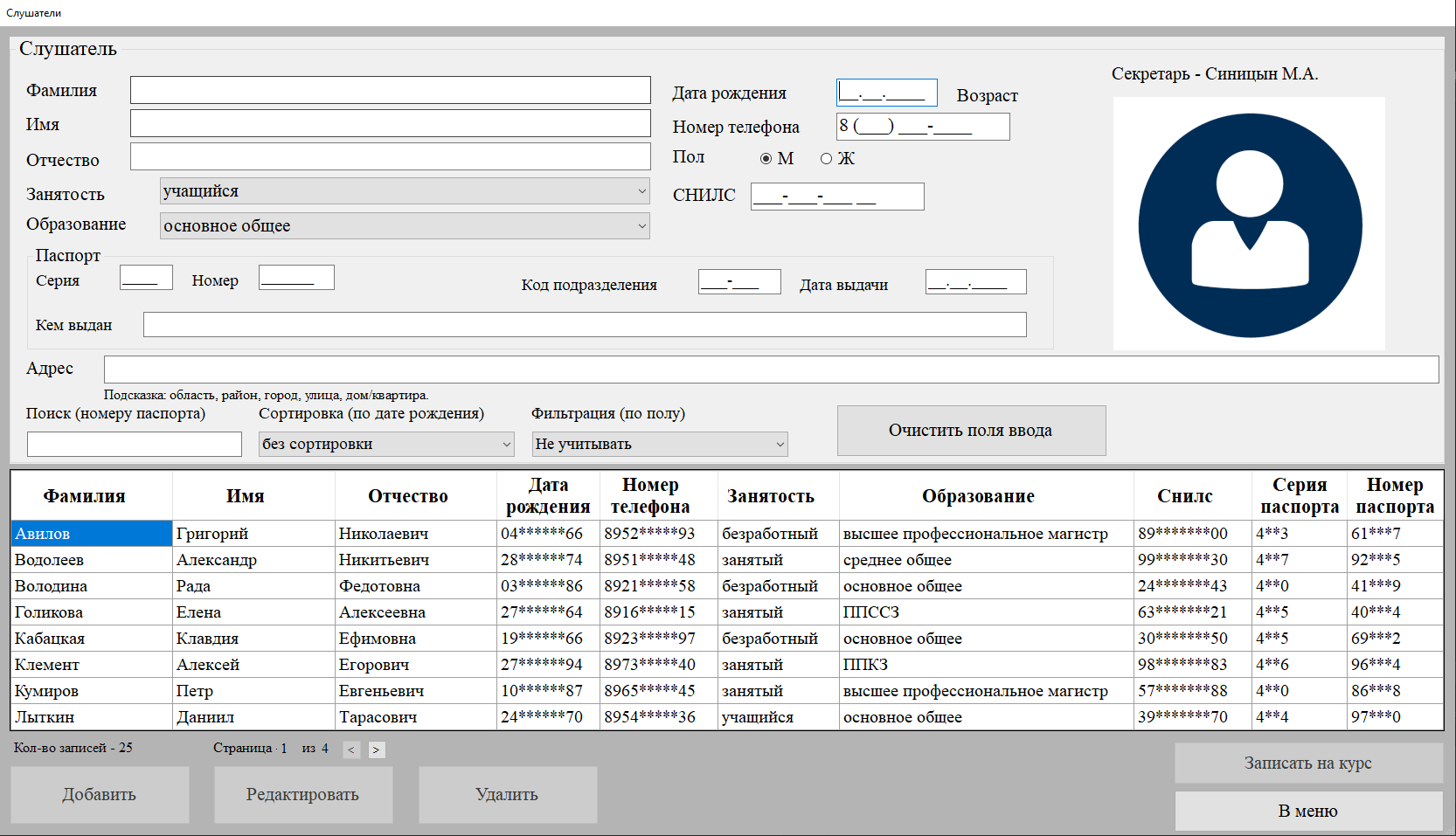


Рисунок 2.21 – Форма «Слушатели»

4. *скрытие форм* при отсутствии активности работы с АИС в течении 3-х минут. Этот способ обезопасит от несанкционированного доступа при оставлении АИС без присмотра в соответствии с рисунком 2.22.

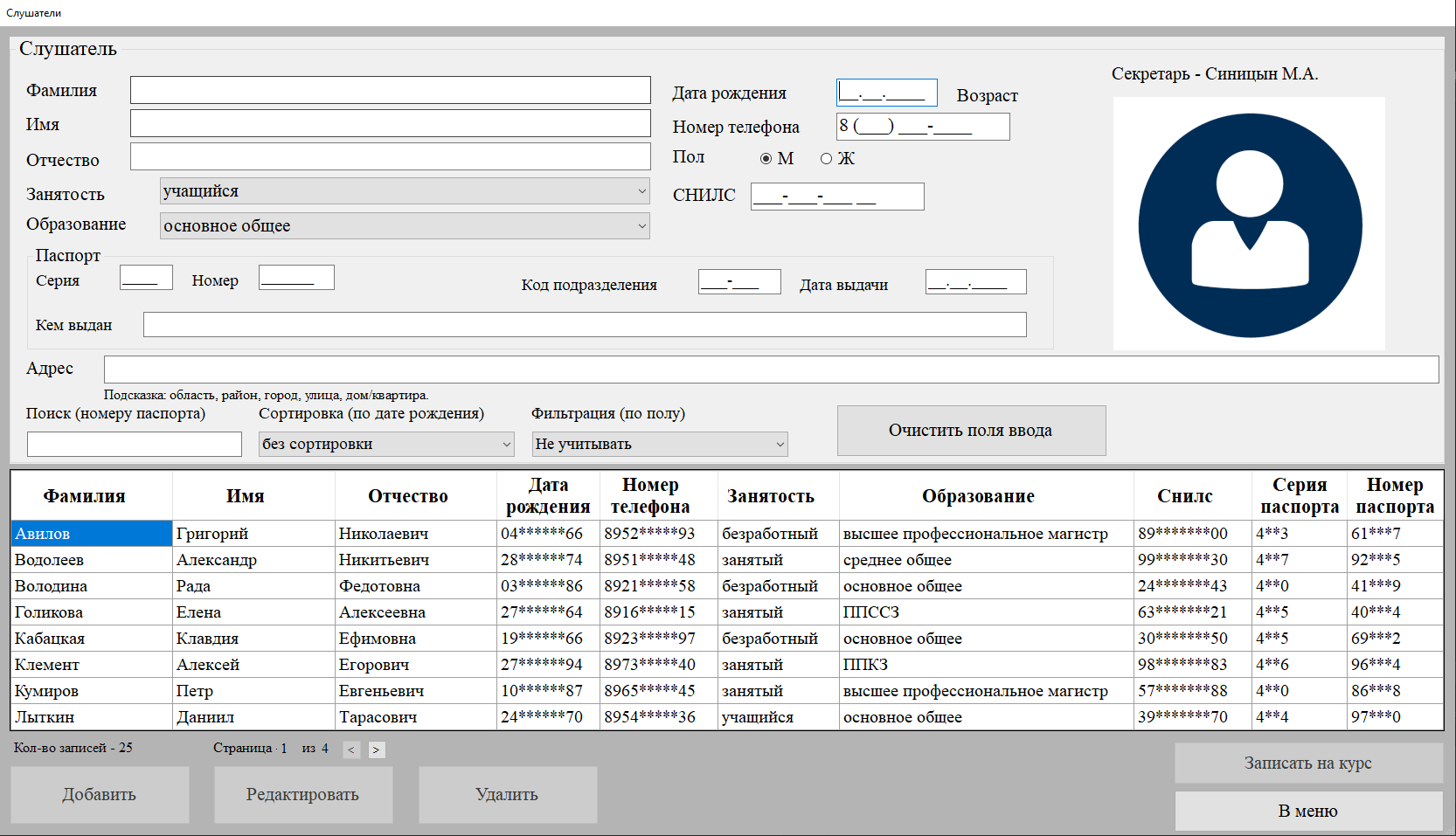
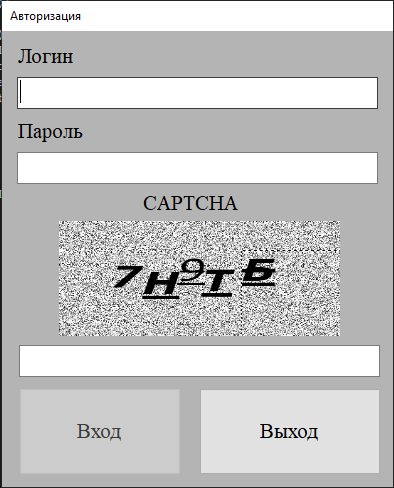
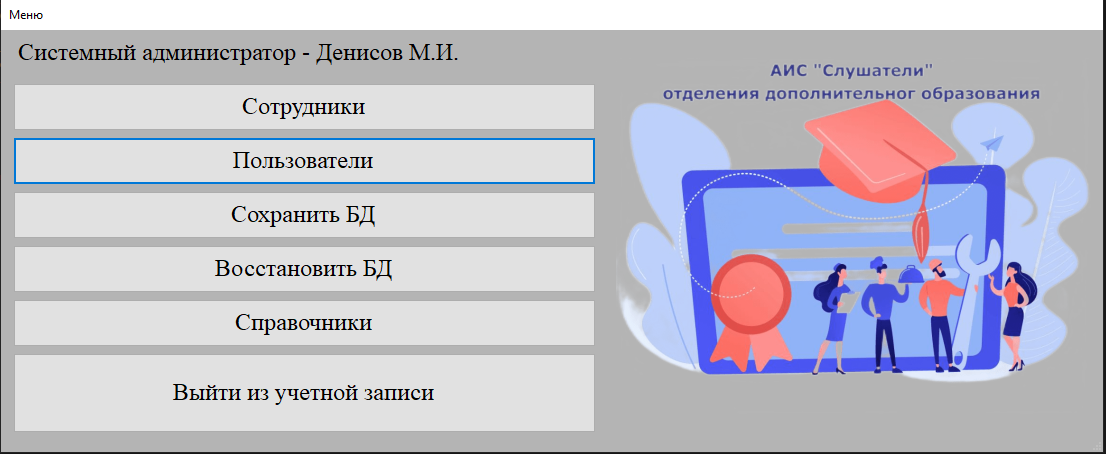


Рисунок 2.22 – Блокировка системы при отсутствии активности

5. *резервное копирование БД -* необходимо для зашиты базы данных от системных сбоев, ошибок и для возможности восстановления продолжения работы с базой данных. Резервное копирование осуществляется автоматически при выходе из программы.

Согласно рисунку 2.23 резервное копирования осуществлялется в главном меню администратора и на форме «Справочники».



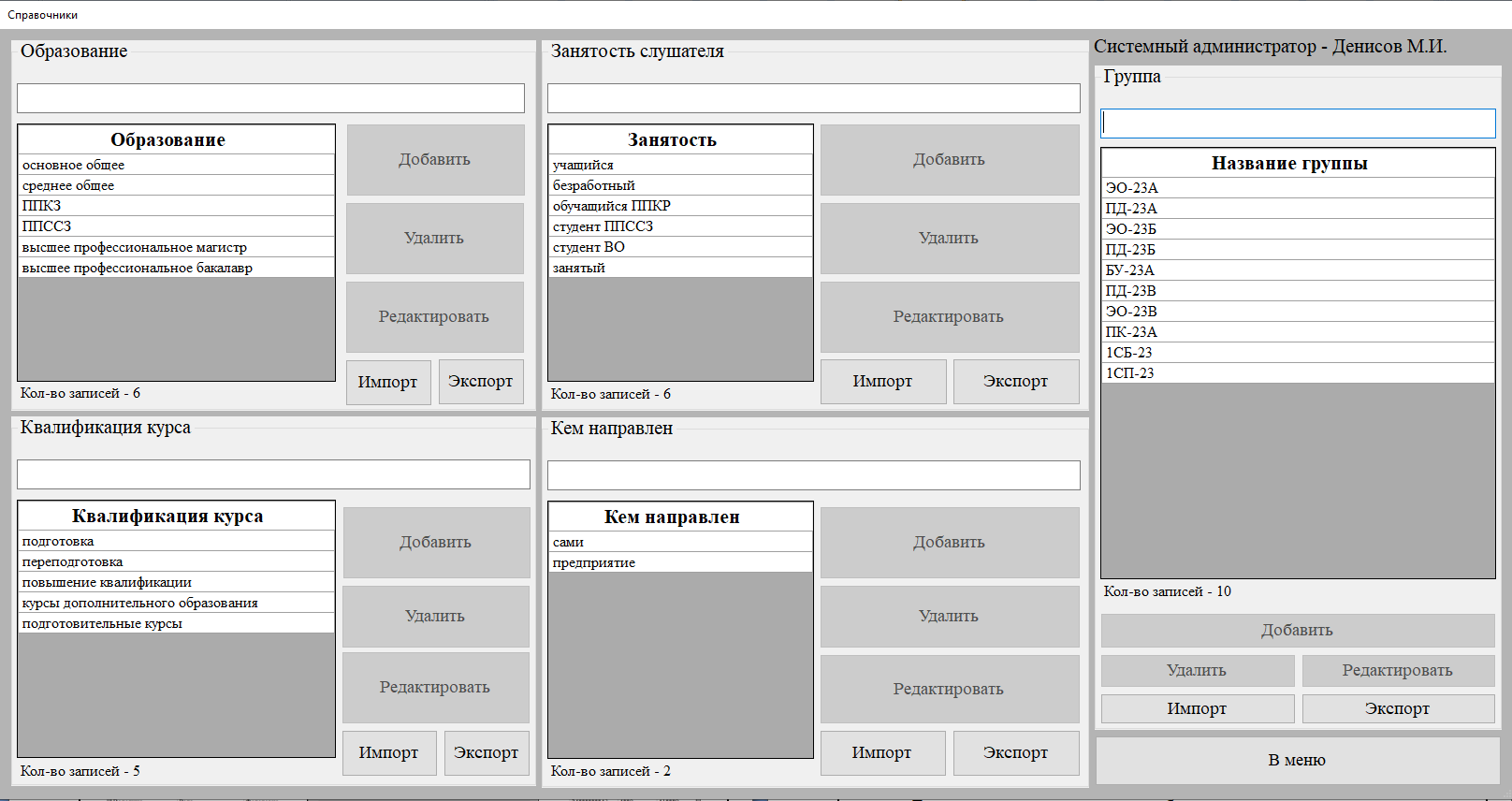
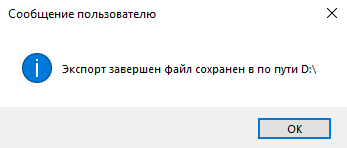


Рисунок 2.23 – Главное меню администратора и форма «Справочники»

При успешном выполнении резервного копирования появляется окно сообщения в соответствии с рисунком 2.24



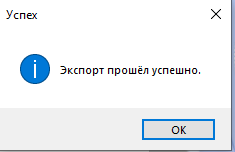


Рисунок 2.24 – Сообщения об успешном резервном копировани

Диалоговое окно восстановлении базы данных, в соответствии с рисунком 2.25

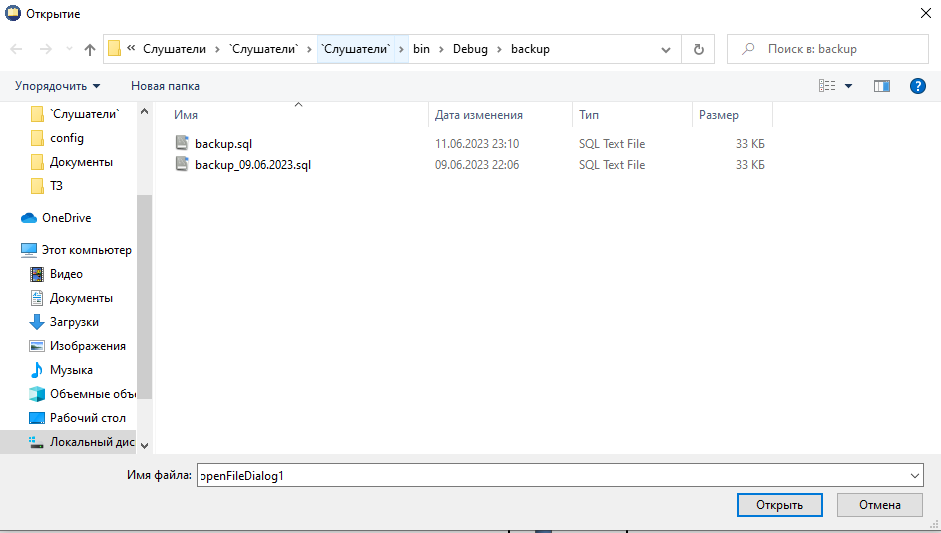


Рисунок 2.25 – Диалоговое окно восстановления базы

Существует несколько причин, при которых база данных может оказаться поврежденной.

Восстановление после сбоев осуществляется при нажатии кнопки «Восстановить БД», откроется диалоговое окно, где нужно выбрать необходимую резервную копию.

**3 Тестирование**

Тестирование – ряд мероприятий, связанных с различного рода испытаниями объекта тестирования с целью установления соответствия или несоответствия его характеристик определенным требованиям и выявления дефектов. Дефектами, в свою очередь, могут быть как ошибки в работе, так и неприемлемое качество функционирования в определенных условиях эксплуатации.

Результаты тестирования- это список выявленных несоответствий и дефектов, но, как правило, без указания их причин.

Тестирование программного продукта проводят как на этапах его создания, так и на этапах технического сопровождения и дальнейшего развития. Оно является одним из наиболее устоявшихся способов обеспечения качества разработки программного обеспечения и входит в набор эффективных средств современной системы обеспечения качества программного продукта.

Таблица 3.1 – Тестирование программы

|  |  |  |  |
| --- | --- | --- | --- |
| Название элемента формы | Ожидаемый результат | Полученный результат | |
| 1 | 2 | 3 | |
| **Форма авторизации** | | | |
| Кнопка «Войти» | При правильно веденных логине и пароле – переход в главное меню | | Корректное выполнение команды |
| Кнопка «Войти» | При ошибке логина или пароля отображение captcha | | Корректное выполнение команды |
| Кнопка «Войти» | При повторной ошибке логина, пароля или captcha блокировка формы на 10 секунд и обновление captcha | | Корректное выполнение команды |
| Ввод в поле captcha | Ввод только больших русских букв | | Корректное выполнение команды |
| Кнопка «Выйти» | Завершение работы приложения и резервное копирование БД | | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| **Форма главного меню системного администратора** | | |
| Кнопка «Сотрудники» | Переход на форму сотрудники | Корректное выполнение команды |
| Кнопка «Пользователи» | Переход на форму пользователи | Корректное выполнение команды |
| Кнопка «Сохранить БД» | Вывод окна выбора папки, при выборе папки сохранения файла БД формата .sql | Корректное выполнение команды |
| Кнопка «Восстановить БД» | Выбор файла восстановления формата .sql и восстановление БД | Корректное выполнение команды |
| Кнопка «Справочники» | Переход на форму справочников | Корректное выполнение команды |
| Кнопка «Выйти из учетной записи» | Вывод сообщения о подтверждении выхода и при положительном ответе переход на форму авторизации | Корректное выполнение команды |
| Неактивность пользователя | Переход на форму авторизации | Корректное выполнение команды |
| **Форма главного меню секретаря** | | |
| Кнопка «Слушатели» | Переход на форму слушателей | Корректное выполнение команды |
| Кнопка «Справочники» | Переход на форму справочников | Корректное выполнение команды |
| Кнопка «Договоры» | Переход на форму учета договоров | Корректное выполнение команды |
| Кнопка «Курсы» | Переход на форму курсов | Корректное выполнение команды |
| Кнопка «Группы» | Переход на форму групп | Корректное выполнение команды |
| Кнопка «Информация об организации» | Переход на форму смены информации об организации | Корректное выполнение команды |
| Кнопка «Выйти из учетной записи» | Вывод сообщения о подтверждении выхода и при положительном ответе переход на форму авторизации | Корректное выполнение команды |
| Неактивность пользователя | Переход на форму авторизации | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| **Форма главного меню заведующего** | | |
| Кнопка «Договоры» | Переход на форму учета договоров | Корректное выполнение команды |
| Кнопка «Отчеты» | Переход на форму отчетов | Корректное выполнение команды |
| Кнопка «Выйти из учетной записи» | Вывод сообщения о подтверждении выхода и при положительном ответе переход на форму авторизации | Корректное выполнение команды |
| Неактивность пользователя | Переход на форму авторизации | Корректное выполнение команды |
| **Форма Сотрудники** | | |
| Ввод в поле Фамилия | Ввод только русских букв, автоматическая смена раскладки на русскую, первая буква верхнего регистра | Корректное выполнение команды |
| Ввод в поле Имя | Ввод только русских букв, автоматическая смена раскладки на русскую, первая буква верхнего регистра | Корректное выполнение команды |
| Ввод в поле Отчество | Ввод только русских букв, автоматическая смена раскладки на русскую, первая буква верхнего регистра | Корректное выполнение команды |
| Подсчет количества записей сотрудников | Корректный подсчет сотрудников | Корректное выполнение команды |
| Ввод в поле Код сотрудника | При вводе данных установка курсора на 1 букву, ввод только цифр | Корректное выполнение команды |
| Кнопка «Очистить поля» | Очистка всех полей ввода на форме, снятие выделения с таблицы, блокировка кнопок на форме | Корректное выполнение команды |
| Выделение записи в таблице сотрудников | Заполнение полей ввода данными из выбранной записи, разблокировка кнопок на форме | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Доступ к кнопке «Добавить» | Разблокировка кнопки при вводе всех данных | Корректное выполнение команды |
| Доступ к кнопке «Редактировать» | Разблокировка кнопки при вводе всех данных и выбора записи для редактирования | Корректное выполнение команды |
| Доступ к кнопке «Удалить» | Разблокировка кнопки при выборе записи для удаления | Корректное выполнение команды |
| Кнопка «Добавить» | Добавление записи в таблицу сотрудников | Корректное выполнение команды |
| Кнопка «Редактировать» | Редактирование данных выбранной записи | Корректное выполнение команды |
| Кнопка «Удалить» | Отображение окна вопроса об удалении, при положительном ответе удаление записи | Корректное выполнение команды |
| Попытка добавления сотрудника с неуникальным кодом | Отображение ошибки неуникальности кода | Корректное выполнение команды |
| Попытка удаления сотрудника, который является пользователем | Отображение ошибки, что сотрудник является пользователем | Корректное выполнение команды |
| Кнопка «В меню» | Переход на главное меню | Корректное выполнение команды |
| Неактивность пользователя | Переход на форму авторизации | Корректное выполнение команды |
| **Форма Пользователи** | | |
| Ввод в поле «Сотрудник» | Выпадающий список с сотрудниками из таблицы сотрудников | Корректное выполнение команды |
| Ввод в поле «Тип пользователя» | Выпадающий список с типами | Корректное выполнение команды |
| Выделение записи в таблице пользователи | Разблокировка кнопки удаления | Корректное выполнение команды |
| Подсчет количества записей пользователей | Корректный подсчет пользователей | Корректное выполнение команды |
| Доступ к кнопке «Добавить» | Ввод всех данных в поля ввода | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Доступ к кнопке «Удалить» | Выделение записи для удаления | Корректное выполнение команды |
| Кнопка «Добавить» | Добавление пользователя в таблицу пользователей | Корректное выполнение команды |
| Кнопка «Удалить» | Отображение окна вопроса об удалении, при положительном ответе удаление записи | Корректное выполнение команды |
| Кнопка «В меню» | Переход на главное меню | Корректное выполнение команды |
| Неактивность пользователя | Переход на форму авторизации | Корректное выполнение команды |
| **Форма Справочники** | | |
| Кнопка «Импорт» | Открытие выбора файла для импорта формата .csv, при выборе файла импорт данных в справочник | Корректное выполнение команды |
| Кнопка «Экспорт» | Открытие выбора папки для экспорта файла .csv, при выборе папки экспорт файла | Корректное выполнение команды |
| Ввод в поля справочников | Автоматическое переключение на русскую раскладку | Корректное выполнение команды |
| Доступ к кнопке «Добавить» | Ввод данных для добавления в поле ввода справочника | Корректное выполнение команды |
| Доступ к кнопке «Редактировать» | Ввод данных и выбор записи для редактирования | Корректное выполнение команды |
| Доступ к кнопке «Удалить» | Выбор записи для удаления | Корректное выполнение команды |
| Кнопка «Добавить» | Добавление записи в справочник | Корректное выполнение команды |
| Кнопка «Редактировать» | Редактирование выбранной записи | Корректное выполнение команды |
| Кнопка «Удалить» | Появление сообщение об удалении записи, при положительном ответе пользователя удаление записи из справочника | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Подсчет записей справочников | Корректный подсчет записей справочников | Корректное выполнение команды |
| Попытка удаления использующихся данных из справочника | Отображение ошибки, что данное используется | Корректное выполнение команды |
| Попытка добавления одинаковых данных | Отображение ошибки, что такая запись существует | Корректное выполнение команды |
| Кнопка «В меню» | Переход на главное меню | Корректное выполнение команды |
| Неактивность пользователя | Переход на форму авторизации | Корректное выполнение команды |
| **Форма Слушатели** | | |
| Ввод в поле Фамилия | Ввод только русских букв, автоматическая смена раскладки на русскую, первая буква верхнего регистра | Корректное выполнение команды |
| Ввод в поле Имя | Ввод только русских букв, автоматическая смена раскладки на русскую, первая буква верхнего регистра | Корректное выполнение команды |
| Ввод в поле Отчество | Ввод только русских букв, автоматическая смена раскладки на русскую, первая буква верхнего регистра | Корректное выполнение команды |
| Подсчет количества записей слушателей | Корректный подсчет слушателей | Корректное выполнение команды |
| Скрытие персональных данных | Скрытие персональных данных в таблице слушателей | Корректное выполнение команды |
| Ввод Занятости | Выпадающий список из справочника | Корректное выполнение команды |
| Ввод Образования | Выпадающий список из справочника | Корректное выполнение команды |
| Ввод даты рождения | Проверка на правильность введения даты (вывод окна ошибки), при входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Ввод номера телефона | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод СНИЛС | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод серии паспорта | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод номера паспорта | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод кода подразделения | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод даты выдачи паспорта | Проверка на правильность введения даты (вывод окна ошибки), при входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Кнопка «Очистить поля ввода» | Очистка всех полей ввода на форме, снятие выделения с таблицы, блокировка кнопок на форме | Корректное выполнение команды |
| Пагинация | Динамический вывод 8 записей на странице, и переход на следующую и предыдущую страницу | Корректное выполнение команды |
| Ввод в поле поиска | Выделение найденной строки с указанным номером паспорта | Корректное выполнение команды |
| Смена сортировки на «без сортировки» | Сортировка по фамилии слушателя | Корректное выполнение команды |
| Смена сортировки на «по убыванию» | Сортировка по дате рождения по убыванию | Корректное выполнение команды |
| Смена сортировки на «по возрастанию» | Сортировка по дате рождения по возрастанию | Корректное выполнение команды |
| Смена фильтрации на «Не учитывать» | Снятие фильтрации | Корректное выполнение команды |
| Смена фильтрации на «Мужской» | Отображение слушателей с мужским полом | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Смена фильтрации на «Женский» | Отображение слушателей с женским полом | Корректное выполнение команды |
| Доступ к кнопке «Добавить» | Ввод данных для добавления в поля ввода справочника | Корректное выполнение команды |
| Доступ к кнопке «Редактировать» | Ввод данных и выбор записи для редактирования | Корректное выполнение команды |
| Доступ к кнопке «Удалить» | Выбор записи для удаления | Корректное выполнение команды |
| Кнопка «Добавить» | Добавление слушателя в таблицу | Корректное выполнение команды |
| Кнопка «Редактировать» | Редактирование выбранного слушателя | Корректное выполнение команды |
| Кнопка «Удалить» | Появление сообщение об удалении слушателя, при положительном ответе пользователя удаление слушателя из таблицы | Корректное выполнение команды |
| Попытка удаления слушателя, заключившего договор | Отображение ошибки, что со слушателем заключен договор | Корректное выполнение команды |
| Попытка добавления слушателя с одинаковыми документами | Отображение ошибки, что такой слушатель существует | Корректное выполнение команды |
| Кнопка «В меню» | Переход на главное меню | Корректное выполнение команды |
| Неактивность пользователя | Переход на форму авторизации | Корректное выполнение команды |
| Подсчет возраста | Автоматический подсчет возраста при указании даты рождения слушателя | Корректное выполнение команды |
| Кнопка «Запись на курс» | Переход на форму записи на курс | Корректное выполнение команды |
| Выделение слушателя из таблицы | Разблокировка кнопок и вывод данных выбранного слушателя в поля ввода | Корректное выполнение команды |
| **Форма Запись на курс** | | |
| Отображение фамилии выбранного слушателя | Отображение фамилии выбранного слушателя | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Дата оформления договора | Автоматический забор даты оформления договора с системной даты | Корректное выполнение команды |
| Нажатие на кнопку «Выбрать курс» | Переход на форму выбора курса | Корректное выполнение команды |
| Выбор курса | На форме выбора курса нажатие на курс и кнопку «Выбрать курс» | Корректное выполнение команды |
| Нажатие на кнопку «Выбрать группу» | Переход на форму выбора группы | Корректное выполнение команды |
| Выбор группы | На форме выбора группы нажатие на курс и кнопку «Выбрать группу» | Корректное выполнение команды |
| Ввод направителя | Выпадающий список | Корректное выполнение команды |
| Ввод направителя «предприятие» | Разблокировка полей ввода информации о предприятии | Корректное выполнение команды |
| Проверка на совершеннолетие слушателя | При несовершеннолетии слушателя разблокируется поля ввода законного представителя | Корректное выполнение команды |
| Ввод срока обучения | Проверка на правильность введения даты (вывод окна ошибки), при входе на поле ввода установка курсора на первую букву, проверка чтобы дата «от» была меньше чем дата «до» | Корректное выполнение команды |
| Ввод кода договора | Смена раскладки на русский язык, ввод только символом верхнего регистра | Корректное выполнение команды |
| Ввод фамилии законного представителя | Ввод только русских букв, автоматическая смена раскладки на русскую, первая буква верхнего регистра | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Ввод имени законного представителя | Ввод только русских букв, автоматическая смена раскладки на русскую, первая буква верхнего регистра | Корректное выполнение команды |
| Ввод отчества законного представителя | Ввод только русских букв, автоматическая смена раскладки на русскую, первая буква верхнего регистра | Корректное выполнение команды |
| Ввод адреса законного представителя | Автоматическая смена раскладки на русскую | Корректное выполнение команды |
| Ввод серии паспорта законного представителя | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод номера паспорта законного представителя | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод кода подразделения законного представителя | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод даты выдачи паспорта законного представителя | Проверка на правильность введения даты (вывод окна ошибки), при входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод директора предприятия | Автоматическая смена раскладки на русскую, автоматическая смена первой буквы на заглвную | Корректное выполнение команды |
| Ввод ИНН организации | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод КПП организации | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод БИК организации | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Проверка на уникальность договора | Если такой код договора уже существует, то происходит отображение ошибки | Корректное выполнение команды |
| Кпопка «К слушателям» | Переход на форму Слушатели | Корректное выполнение команды |
| Кнопка «Запись на курс» | Создание новго договора, вывод сообщения-вопроса о переходе на форму учета договора, при положительном ответе происходит переход на форму Учет договоров, при отрицательном ответе происходит переход на форму Слушатели | Корректное выполнение команды |
| Неактивность пользователя | Переход на форму авторизации | Корректное выполнение команды |
| **Форма Курсы** | | |
| Ввод в поле «Длительность курса» | Ввод только цифр | Корректное выполнение команды |
| Ввод в поле «Стоимость курса» | Ввод целого числа или десятичного | Корректное выполнение команды |
| Выделение курса из таблицы | Разблокировка кнопок и вывод данных выбранного курса в поля ввода | Корректное выполнение команды |
| Ввод в поле «Квалификация» | Выпадающий список | Корректное выполнение команды |
| Кнопка «Очистить поля ввода» | Очистка всех полей ввода на форме, снятие выделения с таблицы, блокировка кнопок на форме | Корректное выполнение команды |
| Проверка на уникальность курса | Если совпадают название курса и квалификация выводится сообщение пользователю | Корректное выполнение команды |
| Доступ к кнопке «Добавить» | Ввод данных для добавления слушателя | Корректное выполнение команды |
| Доступ к кнопке «Редактировать» | Ввод данных и выбор записи для редактирования | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Доступ к кнопке «Удалить» | Выбор записи для удаления | Корректное выполнение команды |
| Кнопка «Добавить» | Добавление курса в таблицу | Корректное выполнение команды |
| Кнопка «Редактировать» | Редактирование выбранного курса | Корректное выполнение команды |
| Кнопка «Удалить» | Появление сообщение об удалении курса, при положительном ответе пользователя удаление курса из таблицы | Корректное выполнение команды |
| Попытка удаления курса, который используется в договоре | Вывод ошибки пользователю | Корректное выполнение команды |
| Подсчет количества курсов в таблице | Подсчет количества курсов в таблице | Корректное выполнение команды |
| Кнопка «В меню» | Переход в главное меню | Корректное выполнение команды |
| Неактивность пользователя | Переход на форму авторизации | Корректное выполнение команды |
| **Форма Группы** | | |
| Выделение группы из таблицы | Вывод названия группы и количества человек в ней | Корректное выполнение команды |
| Подсчет количества записей в таблице | Подсчет количества записей в таблице | Корректное выполнение команды |
| Кнопка «В меню» | Переход в главное меню | Корректное выполнение команды |
| Кнопка «Сформировать список групп в Word» | Вывод списка группы в Word в зависимости от выбранной группы из выпадающего списка | Корректное выполнение команды |
| Неактивность пользователя | Переход на форму авторизации | Корректное выполнение команды |
| **Форма Информация об организации** | | |
| Ввод ИНН | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод КПП | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | | 2 | 3 |
| Ввод БИК | | При входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Кнопка «В меню» | | Переход в главное меню | Корректное выполнение команды |
| Доступ к кнопке «Изменить» | | Ввод всех данных в поля ввода | Корректное выполнение команды |
| Кнопка «Изменить» | | Добавляет новые данные об организации | Корректное выполнение команды |
| Неактивность пользователя | | Переход на форму авторизации | Корректное выполнение команды |
| **Форма Отчеты** | | | |
| Кнопка «В меню» | | Переход в главное меню | Корректное выполнение команды |
| Кнопка «Вывод отчета по оплатам в Excel» | | Выводи отчета по оплатам всех договоров | Корректное выполнение команды |
| Кнопка «Вывод статистики в Excel» | | Вывод статистического отчета обучившихся слушателей | Корректное выполнение команды |
| Неактивность пользователя | | Переход на форму авторизации | Корректное выполнение команды |
| **Форма Учет договоров** | | | |
| Подсчет количества записей договоров | Корректный подсчет договоров | | Корректное выполнение команды |
| Смена фильтра группа | Если поставлено «Не учитывать», то фильтр не учитывается, в противном случае отображаются договоры выбранной группы | | Корректное выполнение команды |
| Смена фильтра статуса договора | Если поставлено «Не учитывать», то фильтр не учитывается, в противном случае отображаются договоры с выбранным статусом договора | | Корректное выполнение команды |
| Смена фильтра квалификации курса | Если поставлено «Не учитывать», то фильтр не учитывается, в противном случае отображаются договоры с выбранной квалификацией курса | | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Смена фильтра статуса оплаты | Если поставлено «Не учитывать», то фильтр не учитывается, в противном случае отображаются договоры с выбранным статусом оплаты | Корректное выполнение команды |
| Смена фильтра статуса обучения | Если поставлено «Не учитывать», то фильтр не учитывается, в противном случае отображаются договоры с выбранным статусом обучения | Корректное выполнение команды |
| Ввод дат оформления договора | Проверка на правильность введения дат (вывод окна ошибки), при входе на поле ввода установка курсора на первую букву, проверка порядочность двух дат | Корректное выполнение команды |
| Фильтрация дат оформления договора | Отображение договоров оформленным в указанном периоде | Корректное выполнение команды |
| Кнопка «Сбросить фильтр» | Сброс фильтров | Корректное выполнение команды |
| Кнопка «Экспорт отчета по договорам в Excel» | Вывод отчета по договорам в Excel | Корректное выполнение команды |
| Кнопка «В меню» | Переход в главное меню | Корректное выполнение команды |
| Пагинация | Динамический вывод 20 записей на странице, и переход на следующую и предыдующую страницу | Корректное выполнение команды |
| Ввод в поле поиска | Выделение найденной строки с указанным номером договора | Корректное выполнение команды |
| Смена сортировки на «по убыванию» | Сортировка по дате оформления по убыванию | Корректное выполнение команды |
| Смена сортировки на «по возрастанию» | Сортировка по дате оформления по возрастанию | Корректное выполнение команды |
| Условное форматирование статуса договора | Статус договора «закрыт» подсвечивается красным, а «открыт» зеленым | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Условное форматирование статуса оплаты | Статус оплаты «не оплачен» подсвечивается оранжевым, а «оплачен» белым | Корректное выполнение команды |
| Выделение договора из таблицы | Доступ к кнопке «Просмотр договора» | Корректное выполнение команды |
| Кнопка «Просмотр договора» | Переход на форму «Просмотр договора» | Корректное выполнение команды |
| Неактивность пользователя | Переход на форму авторизации | Корректное выполнение команды |
| **Форма Подробного просмотра договора** | | |
| Отображение информации о выбранном договоре | Отображение информации о выбранном договоре на форме | Корректное выполнение команды |
| Ввод даты приказа о зачислении | Проверка на правильность введения дат (вывод окна ошибки), при входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод даты приказа об отчислении | Проверка на правильность введения дат (вывод окна ошибки), при входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод даты начала обучения | Проверка на правильность введения дат (вывод окна ошибки), при входе на поле ввода установка курсора на первую букву, проверка порядочность двух дат | Корректное выполнение команды |
| Ввод даты окончания обучения | Проверка на правильность введения дат (вывод окна ошибки), при входе на поле ввода установка курсора на первую букву, проверка порядочность двух дат | Корректное выполнение команды |
| Ввод даты оплаты | Проверка на правильность введения дат (вывод окна ошибки), при входе на поле ввода установка курсора на первую букву | Корректное выполнение команды |
| Ввод суммы оплаты | Ввод только целых и десятичных | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Ввод статуса обучения | Выпадающий список | Корректное выполнение команды |
| Ввод статуса освоения | Выпадающий список | Корректное выполнение команды |
| Выбор статуса обучения «отчислен» | Вывод предупреждения пользователю | Корректное выполнение команды |
| Автоматическая смена статуса договора | Статус договора изменяется на «закрыт» при статусе оплаты «оплачен», статусе обучения «закончил», статусе освоения «освоен» или статусе обучения «отчислен» | Корректное выполнение команды |
| Автоматическая смена статуса оплаты | Автоматическая смена статуса оплаты на «оплачен» при оплате полной стоимости курсы | Корректное выполнение команды |
| Доступность кнопки «Напечатать соглашение» | Доступ только при оплате материнским капиталом | Корректное выполнение команды |
| Доступность кнопки «Напечатать платежку» | Не доступна, если заказчиком является предприятие | Корректное выполнение команды |
| Различные виды договора | Виды договоров в зависимости от совершеннолетия и заказчика | Корректное выполнение команды |
| Кнопка «Напечатать договор» | Экспорт договора в MS Word 2007 | Корректное выполнение команды |
| Кнопка «Напечатать платежку» | Экспорт платежного документа в MS Word 2007 | Корректное выполнение команды |
| Кнопка «Напечатать соглашение» | Экспорт соглашения в MS Word 2007 | Корректное выполнение команды |
| Кнопка «Сменить статусы, данные о приказах и даты обучения» | Смена статусов обучения и освоения, данных о приказах и дат начала и окончания обучения | Корректное выполнение команды |
| Кнопка «Внести оплату» | Внесение оплаты за курс и отображение её в таблице оплат | Корректное выполнение команды |
| Попытка ввода большей суммы оплаты чем полная | Вывод ошибки пользователю | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Кнопка «К учету договоров» | Переход на форму учета договоров | Корректное выполнение команды |
| Неактивность пользователя | Переход на форму авторизации | Корректное выполнение команды |

**4 Экономическое обоснование**

Разрабатываемое программное приложение предназначено для организации учета договоров отделения дополнительного образования техникума. Существующие аналоги, например 1С Управление учебными курсами, построены на 1С имеют огромный функционал: учёт договоров, учащихся, курсов и оплат. Все программные продукты платные, требуют предварительной настройки и сопровождения, которые являются платными услугами.

В рамках выпускной квалификационной работы было выбрано одно направление – «Учет договоров и оплат курсов». Реализации данного направления достаточно для демонстрации профессиональных навыков программирования в области автоматизации деятельности сотрудников отделения системы при работе со слушателями на платформе .NET.

Разрабатываемая АИС позволит автоматизировать процессы:

- занесения слушателя и запись его на курсы;

- просмотр слушателей и договоров с учетом поиска, фильтрации и сортировки;

- вести учёт и оплат договоров;

- обновление статусов договоров;

- вывод договоров, соглашений и платежных документов;

- получение отчетов по группам, по оплатам, статистических, по договорам;

- выполнять импорт и экспорт данных, резервное копирование и восстановление БД.

Программное приложение позволит:

- оперативно осуществлять выборку данных из базы;

- редактировать базу данных.

Стадии разработки программного продукта представлены в таблице 4.1

Таблица 4.1 – Стадии и этапы разработки

| № п/п | Название этапа | Содержание | Кол-во дней |
| --- | --- | --- | --- |
| 1 | Анализ | Анализ предметной области | 2 |
| 2 | Проектирование | Составление технического задания | 3 |
| 3 | Разработка | Разработка структуры БД | 26 |
| Разработка форм программного приложения |
| Разработка программного кода |
| 4 | Тестирование | Проверка работоспособности всех функций программного продукта | 1 |
| 5 | Документирование | Составление руководства пользователя | 3 |
| Составление листинга | 1 |
| Составление пояснительной записки | 5 |

На изготовление программного продукта будет потрачено 40 дней.

Приведенные в данном разделе выпускной квалификационной дипломной работы результирующие таблицы, перечни данных, позволяют сопоставить результаты разработки и затраты на нее, чтобы сделать вывод об эффективности проекта. Исходные данные для расчета экономических показателей приведены в таблице 4.2.

Таблица 4.2 Исходные данные для расчета экономических показателей

|  |  |  |  |
| --- | --- | --- | --- |
| Обозначение | Наименование показателя | Единицы измерения | Значение показателя |
| СЭВМ | Стоимость ЭВМ | руб. | 35000 |
| ДМ | Среднее количество рабочих дней в месяце | дни | 24 |
| Счас | Стоимость час работы программиста | руб. | 100 |
| ЦЭЛ | Тариф за 1 кВт/час | руб. | 4,48 |
| Тчас | Количество часов работы над разработкой в день | час | 5 |
| P | Мощность, потребляемая ЭВМ | кВт | 0,5 |
| Тинтер | Тариф за услугу Интернет, месяц | руб. | 600 |

4.1 Расчет себестоимости программного продукта

Себестоимость разработки программного продукта рассчитываются по следующей формуле:

|  |  |
| --- | --- |
| Спп = ЗП + Зэл + А + Зпр | (1) |

где ЗП– затрата программиста;

Зэл – затраты на электроэнергию;

А – амортизация оборудования;

Зпр – прочие расходы.

Зарплата программиста рассчитывается по формуле:

|  |  |
| --- | --- |
| ЗП **=** Сч **×** Тчас **×** Д | (2) |

где Сч – стоимость час работы программиста (руб.);

Тчас – количество часов работы над разработкой в день;

Д– количество дней затраченных на разработку программного продукта.

Согласно таблице 4.1 на разработку программного продукта ушло 24 рабочих дня. Среднее количество часов работы в день составило 5 часов. Так как навыков разработки программных продуктов мало, то часовая ставка начинающего программиста составит 100 рублей, что соответствует данным таблицы 4.2

|  |  |
| --- | --- |
| ЗП **=** 100 **×** 5**×** 24 = 12000 руб. |  |

Затраты на электроэнергию рассчитывается по формуле:

|  |  |
| --- | --- |
| Зэл **=** Чотраб **×** Ц1кВт **×** Р | (3) |

где Чотраб – количество часов работы оборудования (ПК), освещения;

Ц1кВт – стоимость 1кВт электроэнергии, (руб.);

Р – мощность, потребляемая ЭВМ, (кВт).

Количество часов работы оборудования (ПК), освещения рассчитывается по формуле:

|  |  |
| --- | --- |
| Чотраб **=** Тчас **×** Д | (4) |

где Тчас – количество часов работы над разработкой в день;

Д– количество дней затраченных на разработку программного продукта.

Количество дней на разработку составило 40, что соответствует данным таблицы 4.1. Количество часов работы в день 5, в соответствии с таблицей 4.2

|  |  |
| --- | --- |
| Чотраб **=** 5**×** 40 = 200 часа |  |

В соответствии с данными таблицы 4.2 стоимости 1кВт электроэнергии равна 4,48 рублей, а потребляемая мощность ПК составляет 0,5 кВт в час.

|  |  |
| --- | --- |
| Зэл **=** 200 **×** 4,48 **×** 0,5= 448 руб. |  |

Амортизационные отчисления рассчитываются с учетом нормы амортизации (Нам **=** 5-10%):

|  |  |
| --- | --- |
| А **=** Нам **×** СЭВМ | (5) |

где СЭВМ – стоимость оборудования, согласно таблице 4.2.

|  |  |
| --- | --- |
| А **=** 0,05 **×** 35000 = 1750 руб. |  |

Для разработки программного продукта необходимо использовать Интернет, следовательно определим затраты на Интернет, как прочие.

|  |  |
| --- | --- |
| З пр **=** Ц инт (1ч) **×** Тчас **×** Д | (6) |

Цена одного часа работы в Интернет рассчитывается по формуле:

|  |  |
| --- | --- |
| Ц инт (1ч) = Цтар : Дмес : 24 | (7) |

Согласно таблице 4.2 стоимость услуги Интернет в месяц составил 500 рублей. Исходя из стоимости тарифа определим стоимость часа работ в Интернет.

|  |  |
| --- | --- |
| Ц инт (1ч) = 600 : 24 : 24 = 1,04 руб.  З пр **=** 1,04 **×** 5**×** 24 = 124,80 руб. |  |

Выполнив все расчеты, определим себестоимость программного продукта.

|  |  |
| --- | --- |
| Спп = 12000 + 448 + 1750 + 124,80 = 14322,80 рублей  14500 рублей |  |

4.2 Стратегия продвижения программного продукта на рынок

Сегодня самым эффективным способом продажи программного продукта является Интернет.

Наиболее распространёнными средствами реализации программного продукта в сети Интернет являются: демоверсия, реклама, размещение на различных хостингах.

- демоверсия продукта — предварительная (ограниченная и, как правило, бесплатная) версия продукта. Демоверсии дают представление пользователям о готовящемся продукте и создаются в качестве рекламы и привлечения вниманию к будущему продукту. В этом смысле они приравниваются к трейлерам для кинофильмов. Обычно они являются в некотором смысле не готовой к продаже версией полного продукта. По этим причинам в большинстве случаев они распространяются бесплатно;

- реклама (баннеры) - один из преобладающих форматов [Интернет-рекламы](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D1%80%D0%B5%D0%BA%D0%BB%D0%B0%D0%BC%D0%B0). Представляет собой графическое изображение, аналогичное рекламному модулю в прессе, но способное содержать анимированные (редко видео-) элементы, а также может являться [гиперссылкой](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%81%D1%81%D1%8B%D0%BB%D0%BA%D0%B0) на сайт рекламодателя или страницу с дополнительной информацией;

- размещение на специализированных хостингах – также вариантом распространения программного продукта является размещение на специализированных хостингах, которые являются своего рода специализированными магазинами программного обеспечения.

Из выше перечисленных вариантов распространения программного продукта, было выбрано размещение дипломного проекта за сайте курсар.ру, который взимает плату 30% от стоимости работы. Дипломные работы вирируются в размере от 3000 до 12000 рублей.

4.3 Расчет цены разработанной программы

Цена разработанной программы определяется по следующей формуле:

|  |  |
| --- | --- |
| Ц пп = Спп | (8) |

где Спп – себестоимость программного продукта (руб.);

Зпродв – суммарные затраты на продвижение программы (руб.);

Ц пп = 15000 руб

Выручка от продаж при условии  – количество пользователей, желающих прибрести программу, составит:

|  |  |
| --- | --- |
|  | (10) |

Было принято решение продавать информационную систему по ведению базы данных за 5000 рублей. При продаже каждой копии выручка составит 2800 рублей.

Срок окупаемости инвестиционного проекта () – это период времени, который потребуется для возмещения инвестиций. – определяется с учетом дисконтирования, путем суммирования ежегодных поступлений до определенного периода, в котором они превзойдут первоначальные расходы денежных средств. Расчёт срока окупаемости представлен в таблице 4.3

Таблица 4.3 Расчет срока окупаемости разработанной программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Период | Цена ПП,  руб. | Кол-во покупок  шт. | Выручка,  руб. | Накапливаемая выручка,  руб. |
| 1 год | 5000 | 3 | 10500 | 10500 |
| 2 год | 5000 | 2 | 7000 | 17500 |
| Итого | 5000 | 5 | 17500 |  |

4.4 Технико-экономические показатели разработки программы

Обобщенные технико-экономические показатели разработки программы сведены в таблицу 4.4.

Таблица 4.4 – Технико-экономические показатели разработки программы

|  |  |
| --- | --- |
| Показатель | Значение |
| Себестоимость программного продукта (руб.) | 15000 |
| Цена разработки ПП (руб.) | 15000 |
| Отпускная цена (руб.) | 5000 |
| Срок окупаемости (года) | 2 |

Данные произведенные расчеты показывают, что разработанный программный продукт является эффективным с экономической точки зрения.

**Заключение**

В ходе выполнения выпускной квалификационной работы была создана автоматизированная информационная система «Слушатели» отделения дополнительного образовании, отвечающую поставленным требованиям.

В процессе написания программного кода, использовался язык программирования высокого уровня. Были улучшены навыки по работе с платформой .Net, БД MY SQL, а так же приобретен навык по разработке подобных проектов.

Программное приложение обеспечивается выполнение следующих функций:

- аутентификация пользователя;

- автоматизации ввода исходных данных;

- проверка на корректность ввода данных;

- просмотр, редактирование, добавление и удаление записей;

- подсчет фактической и внесенной оплаты курсов;

- управление статусами договора;

- поиск, сортировка и фильтрация записей по различным критериям;

- подсчет статистики по курсовой подготовки;

- формирование отчетов: договор, доп. соглашение, список группы, квитанция, отчет по договорам, статистический отчет для экспорта в приложение MS Word/Excel 2007;

- резервное копирование и восстановление БД в режиме администратора.

Программный продукт имеет архитектуру клиент – сервера и позволяет в дальнейшем наращивать функционал, переходя на другой уровень архитектуры, например разработка и подключение сайта отделения дополнительного образования техникума. Программа может использоваться в качестве портфолио. После внедрения возможно расширение и доработка функционала.

**Литература**

1. Эндрю Троелсен., Язык программирования C# и платформа .NET., 2015г.
2. Алексей Васильев., Программирование на C# для начинающий. Основные сведения, 2018г.
3. Троелсен Эндрю, Джепикс Филипп, Язык программирования C#7 и платформы .NET и .NET Core, 2018г.
4. Евдокимов П. В.,C# на примерах, 2019г.
5. Вагнер Билл., Наиболее эффективное программирование на C#. 50 способов улучшения кода, 2017г.
6. Мюллер Джон Поль, Семпф Билл ., C# для чайников ,2019г.