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-20А

**Егорова Даниила Дмитриевича**

**Тема:** Разработка автоматизированной информационной системы по ведению базы данных «Слушатель» на базе платформы .NET

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

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

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

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

(дата)

г. Заволжье

2024 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

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 год

**задание**

**на выпускную квалификационную работу**

Студента IV курса группы ПС-20А

**Егорова Даниила Дмитриевича**

**Тема выпускной квалификационной работы:**

Разработка автоматизированной информационной системы по ведению базы данных «Слушатель» на базе платформы .NET

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 « 03» \_\_04\_\_\_\_ 2024 года

#### Дата окончания проекта « 15 » \_\_\_06\_\_ 2024 года

Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Хмелева Е.О./

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 15

2.4 Алгоритм представления задачи 19

2.4.1 Детальное проектирование программного обеспечения (прототип) 19

2.4.2 Описание основных программных модулей 27

2.5 Применяемые меры защиты информации 30

3 Тестирование 34

4 Экономическое обоснование 48

4.1 Расчет затрат на разработку программы 50

4.2 Стратегия продвижения программного продукта 52

4.3 Расчет цены разработанной программы 53

4.4 Технико-экономические показатели разработки программы 54

Заключение 55

Литература 56

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, подготовку договоров для печати и подготовку отчетов об указанной группе. 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 по ведению базы данных «Слушатель» на базе платформы .NEТ. 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; организациях; законных представителях.

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 Excel 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 Community, [Visual Studio Code](https://code.visualstudio.com/) и [Eclipse](https://www.eclipse.org/).

*Microsoft Visual Studio Community* – это линейка продуктов компании Microsoft, включающая интегрированную среду разработки программного обеспечения. Visual Studio Community предлагает мощные средства разработки, такие как интеллектуальная поддержка кода с помощью IntelliSense, инструменты для тестирования и отладки, встроенная система управления версиями и поддержка множества языков программирования, включая C#, C++, Python и другие. Visual Studio Community также интегрируется с популярными платформами, такими как Azure и GitHub, что упрощает разработку облачных приложений и совместную работу над проектами.

[*Visual Studio Code*](https://code.visualstudio.com/)– простой, бесплатный, полнофункциональный и расширяемый инструмент для разработчиков, создающих не корпоративные приложения. Этот редактор кода имеет встроенную поддержку отладки, контроль версий, подсветку синтаксиса и множество расширений, которые можно установить для добавления дополнительного функционала. Visual Studio Code особенно популярен среди веб-разработчиков благодаря своей легковесности и гибкости.

[*Eclipse*](https://www.eclipse.org/) – свободная интегрированная среда разработки модульных кроссплатформенных приложений. Содержит множество плагинов, отличный компилятор и отладчик. Однако у Eclipse есть некоторые недостатки, такие как сложность в настройке и более медленная производительность по сравнению с другими современными IDE.

Исходя из вышеперечисленных инструментальных средств, была выбрана среда разработки Microsoft Visual Studio Community, так как данная среда разработки имеет достаточный функционал для написания данного проекта и является бесплатной при разработке некоммерческих программных продуктов. Также имеется опыт работы при создании подобных проектов, что значительно сократит время разработки.

Данная среда разработки позволяет программировать на следующих языках программирования: С#, C++, Visual Basic, Java Script, Visual F# и другие. Если необходимого языка нет в стандартной библиотеке, его можно внедрить без особых трудностей.

*C# -*  является объектно-ориентированным языком, но поддерживает также и компонентно-ориентированное программирование. Этот язык разработки, созданный компанией Microsoft, позволяет создавать надежные и масштабируемые приложения для различных платформ. C# интегрируется с .NET, предоставляя разработчикам доступ к богатой библиотеке классов и API. Он поддерживает современные парадигмы программирования, такие как асинхронное программирование, лямбда-выражения и LINQ, что делает его мощным и гибким инструментом для разработки веб-приложений, настольных программ и мобильных приложений.

*Visual Basic* – язык программирования, который унаследовал стиль и синтаксис языка Basic, у которого есть немало диалектов. Visual Basic сочетает в себе процедуры и элементы объектно-ориентированных и компонентно-ориентированных языков программирования. Этот язык разработан для создания приложений с графическим пользовательским интерфейсом и является частью экосистемы .NET.

*C++* - компилируемый, статически типизированный язык программирования общего назначения. Он был разработан как расширение языка C, добавив к нему объектно-ориентированные возможности. C++ поддерживает как низкоуровневое программирование, позволяя работать с аппаратным обеспечением напрямую, так и высокоуровневые абстракции, обеспечивая гибкость и мощь в разработке сложных программных систем.

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 исползование C++ для выполнения выпускной квалификационной работы нерационально, а *Visual Basic* не соответствует постановке задачи.

2.2 Выбор способа организации входных и выходных данных

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

*PostgreSQL* – это мощная реляционная система управления базами данных с открытым исходным кодом. Она предназначена для хранения и управления большими и сложными базами данных, предлагая широкий спектр возможностей и функций для администрирования и разработки. PostgreSQL обладает высокой производительностью, масштабируемостью и расширяемостью. Благодаря поддержке стандартов ANSI SQL и SQL:2008, PostgreSQL совместим с другими реляционными базами данных и облегчает перенос приложений между различными системами. Также PostgreSQL предлагает безопасность, поддержку многоязычности, работу с JSON и NoSQL данными, а также активное сообщество разработчиков и пользователей, обеспечивающее поддержку и развитие проекта.

*MySQL -* свободная реляционная система управления базами данных, широко используемая в веб-разработке. MySQL поддерживает обработку больших объемов данных и высокую нагрузку. Клиент-серверная архитектура позволяет работать с базой данных в сетевом окружении, поддерживая множество одновременных подключений. MySQL интегрируется с различными языками программирования и обеспечивает высокий уровень безопасности данных и поддержку транзакций, что позволяет сохранять целостность данных

*SQLite* – компактная встраиваемая СУБД, не требующая отдельного сервера для работы. Она идеально подходит для небольших приложений и встраиваемых систем. SQLite занимает мало места, не требует установки и настройки сервера, а все данные хранятся в одном файле. Она работает быстро и эффективно с небольшими объемами данных. Однако SQLite не предназначена для работы с большими объемами данных и сложными запросами, что ограничивает её применение в более крупных проектах.

Исходя из выше перечисленных вариантов, принято решение использовать MySQL. Её функционала хватает для выполнения поставленной задачи, а также имеется практический навык работы с данной СУБД, что ускоряет процесс разработки. Кроме того, использование MySQL позволит в перспективе перейти к клиент-серверной версии приложения, что обеспечит лучшую масштабируемость и производительность.

База данных состоит из 21 таблицы, которые имеют связь в соответствии с рисунком 2.1

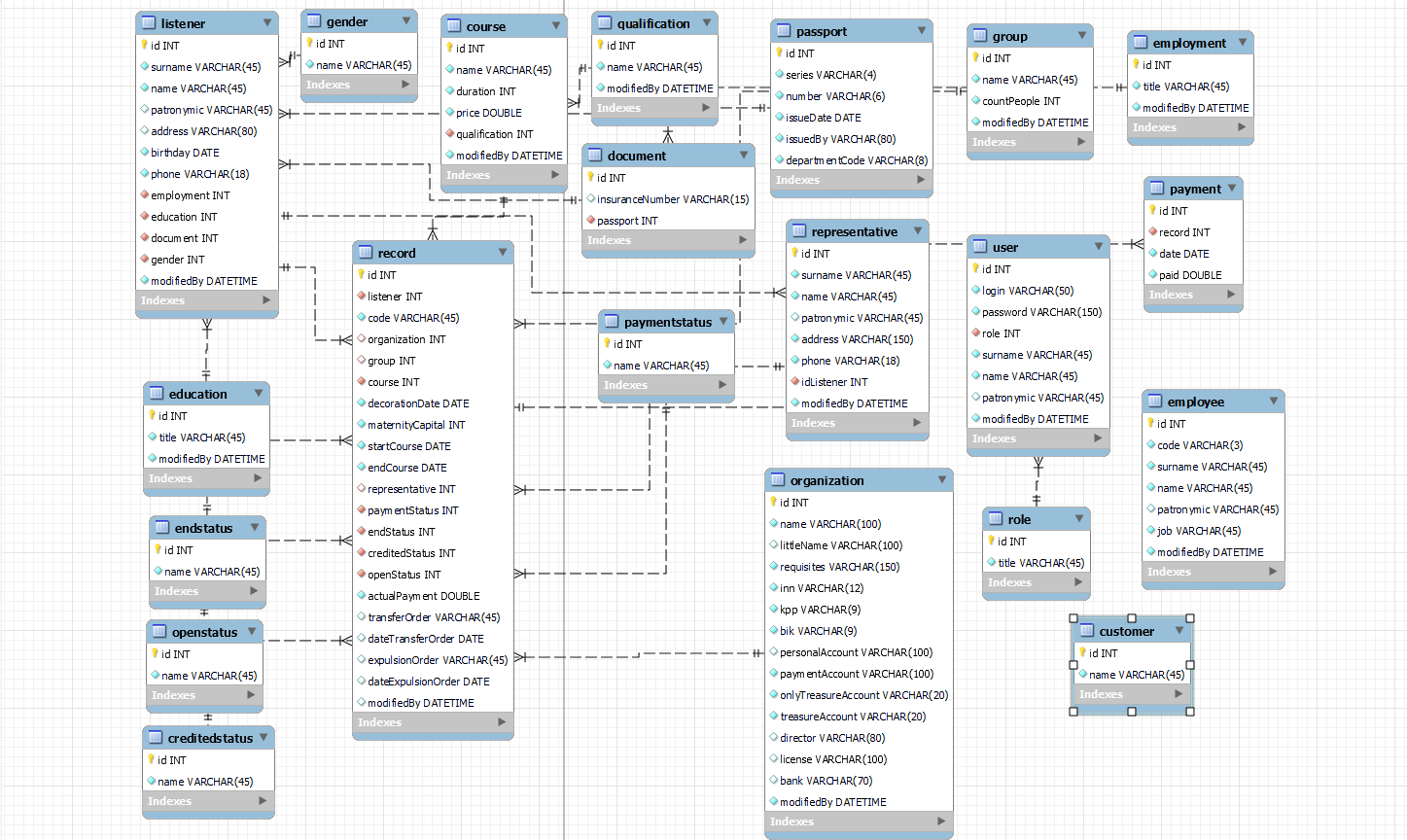


Рисунок 2.1 – Структура базы данных

Структура таблиц базы данных представлена таблицами 2.1-2.22

Таблица 2.2 – Структура таблицы «course» (курс)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца  (в БД) | Тип данных | Ограничения |
| 1 | 2 | 3 | 4 |
| Идентификатор курса | id | Счетчик | длинное целое число, заполняется автоматически |
| Название | name | Текстовый | ограничение в 45 знаков |
| Длительность | duration | Числовой | ограничение в 5 знаков |
| Цена | price | Числовой | ограничение в 10 знаков |
| Квалификация | qualification | Числовой | ограничение в 10 знаков |
| Изменено в | ModifiedBy | Дата и время | ограничение в 20 знаков |

Таблица 2.3 – Структура таблицы «creditedStatus» (статус обучения)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор статуса | id | Счетчик | длинное целое число, заполняется автоматически |
| Наименование | name | Текстовый | ограничение в 45 знаков |

Таблица 2.4 – Структура таблицы «customer» (заказчик)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| 1 | 2 | 3 | 4 |
| Идентификатор заказчика | id | Счетчик | длинное целое число, заполняется автоматически |
| Наименование | name | Текстовый | ограничение в 45 знаков |

Таблица 2.5 – Структура таблицы «document» (документ)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор документа | id | Счетчик | целое число, заполняется автоматически |
| СНИЛС | insuranceNumber | Текстовый | ограничение в 15 знаков |
| Паспорт | passport | Числовой | ограничение в 10 знаков |

Таблица 2.6 – Структура таблицы «education» (образование)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор образования | id | Счетчик | длинное целое число, заполняется автоматически |
| Наименование | title | Текстовый | ограничение в 45 знаков |
| Изменено в | modifiedBy | Дата и время | ограничение в 20 знаков |

Таблица 2.7 – Структура таблицы «employee» (сотрудник)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор сотрудника | id | Счетчик | целое число, заполняется автоматически |
| Код | Code | Текстовый | ограничение в 3 знака |
| Фамилия | Surname | Текстовый | ограничение в 45 знаков |
| Имя | Name | Текстовый | ограничение в 45 знаков |
| Отчество | patronymic | Текстовый | ограничение в 45 знаков |
| Должность | job | Текстовый | ограничение в 80 знаков |
| Изменено в | modifiedBy | Дата и время | ограничение в 20 знаков |

Таблица 2.8 – Структура таблицы «employment» (занятость)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор занятости | id | Счетчик | длинное целое число, заполняется автоматически |
| Наименование | title | Текстовый | ограничение в 45 знаков |
| Изменено в | modifiedBy | Дата и время | ограничение в 20 знаков |

Таблица 2.9 – Структура таблицы «endStatus» (статус освоения)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор статуса | id | Счетчик | длинное целое число, заполняется автоматически |
| Наименование | name | Текстовый | ограничение в 45 знаков |

Таблица 2.10 – Структура таблицы «gender» (пол)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор пола | id | Счетчик | длинное целое число, заполняется автоматически |
| Наименование | name | Текстовый | ограничение в 10 знаков |

Таблица 2.11 – Структура таблицы «group» (группа)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор группы | id | Счетчик | длинное целое число, заполняется автоматически |
| Наименование | name | Текстовый | ограничение в 45 знаков |
| Кол-во людей | countPeople | Числовой | ограничение в 4 знака |
| Изменено в | modifiedBy | Дата и время | ограничение в 20 знаков |

Таблица 2.12 – Структура таблицы «listener» (слушатель)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор слушателя | id | Счетчик | целое число, заполняется автоматически |
| Фамилия | surname | Текстовый | ограничение в 45 знаков |
| Имя | name | Текстовый | ограничение в 45 знаков |
| Отчество | Patronymic | Текстовый | ограничение в 45 знаков |
| Адрес | address | Текстовый | ограничение в 150 знаков |
| Дата рождения | birthday | Дата | ограничение в 80 знаков |
| Телефон | Phone | Текстовый | ограничение в 20 знаков |
| Занятость | Employment | Числовой | ограничение в 20 знаков |
| Образование | Education | Числовой | ограничение в 20 знаков |
| Документ | Document | Числовой | ограничение в 20 знаков |
| Гендер | Gender | Числовой | ограничение в 3 знака |
| Изменено в | modifiedBy | Дата и время | ограничение в 20 знаков |

Таблица 2.13 – Структура таблицы «openStatus» (статус контракта)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор статуса | id | Счетчик | длинное целое число, заполняется автоматически |
| Наименование | name | Текстовый | ограничение в 45 знаков |

Таблица 2.14 – Структура таблицы «organization» (организация)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор организации | id | Счетчик | целое число, заполняется автоматически |
| Наименование | Name | Текстовый | ограничение в 100 знаков |
| Краткое наименование | littleName | Текстовый | ограничение в 100 знаков |
| Реквизиты | Requisites | Текстовый | ограничение в 150 знаков |
| ИНН | Inn | Текстовый | ограничение в 12 знаков |
| КПП | Kpp | Текстовый | ограничение в 9 знаков |
| БИК | Bik | Текстовый | ограничение в 9 знаков |
| Лицевой счет | personalAccount | Текстовой | ограничение в 100 знаков |
| Рассчетный счет | paymentAccount | Текстовый | ограничение в 100 знаков |
| Единый казначейский счет | onlyTreasureAccount | Текстовый | ограничение в 20 знаков |
| Казначейский счет | treasureAccount | Текстовый | ограничение в 20 знаков |
| Директор | Director | Текстовый | ограничение в 80 знаков |
| Лицензия | License | Текстовый | ограничение в 100 знаков |
| Банк | Bank | Текстовый | ограничение в 70 знаков |
| Изменено в | modifiedBy | Дата и время | ограничение в 20 знаков |

Таблица 2.15 – Структура таблицы «passport» (паспорт)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор паспорта | id | Счетчик | целое число, заполняется автоматически |
| Серия | series | Текстовый | ограничение в 4 знака |
| Номер | number | Текстовый | ограничение в 6 знаков |
| Дата выдачи | issueDate | Дата и время | ограничение в 20 знаков |
| Кем выдан | issuedBy | Текстовый | ограничение в 80 знаков |
| Код подразделения | departmentCode | Текстовый | ограничение в 8 знаков |

Таблица 2.16 – Структура таблицы «payment» (оплата)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор оплаты | id | Счетчик | целое число, заполняется автоматически |
| Запись | record | Числовой | ограничение в 20 знаков |
| Дата | date | Дата | ограничение в 10 знаков |
| Сумма | paid | Числовой | ограничение в 20 знаков |

Таблица 2.17 – Структура таблицы «paymentStatus» (статус оплаты)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор статуса | id | Счетчик | длинное целое число, заполняется автоматически |
| Наименование | name | Текстовый | ограничение в 45 знаков |

Таблица 2.18 – Структура таблицы «qualification» (квалификация)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор квалификации | id | Счетчик | целое число, заполняется автоматически |
| Наименование | name | Текстовый | ограничение в 45 знаков |
| Изменено в | modifiedBy | Дата и время | ограничение в 20 знаков |

Таблица 2.19 – Структура таблицы «record» (запись)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор записи | id | Счетчик | целое число, заполняется автоматически |
| Слушатель | Listener | Числовой | ограничение в 100 знаков |
| Код | Code | Текстовый | ограничение в 45 знаков |
| Организация | Organization | Числовой | ограничение в 20 знаков |
| Группа | Group | Числовой | ограничение в 20 знаков |
| Курс | course | Числовой | ограничение в 20 знаков |
| Дата оформления | decorationDate | Дата | ограничение в 10 знаков |
| Мат. капитал | maternityCapital | Числовой | ограничение в 1 знак |
| Дата начала курса | startCourse | Дата | ограничение в 10 знаков |
| Дата окончания курса | endCourse | Дата | ограничение в 10 знаков |
| Законный представитель | Representative | Числовой | ограничение в 20 знаков |

Продолжение таблицы 2.19

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Статус оплаты | paymentStatus | Числовой | ограничение в 20 знаков |
| Статус освоения | endStatus | Числовой | ограничение в 20 знаков |
| Статус обучения | creditedStatus | Числовой | ограничение в 20 знаков |
| Статус договора | openStatus | Числовой | ограничение в 20 знаков |
| Выплачено | actualPayment | Числовой | ограничение в 15 знаков |
| Приказ о зачислении | transferOrder | Текстовый | ограничение в 45 знаков |
| Дата приказа о зачислении | dateTransferOrder | Дата | ограничение в 10 знаков |
| Приказ об отчислении | expulsionOrder | Текстовый | ограничение в 45 знаков |
| Дата приказа об отчислении | dateExpulsionOrder | Дата | ограничение в 10 знаков |
| Изменено в | modifiedBy | Дата и время | ограничение в 20 знаков |

Таблица 2.20 – Структура таблицы «representative» (законный представитель)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор представителя | id | Счетчик | целое число, заполняется автоматически |
| Фамилия | surname | Текстовый | ограничение в 45 знаков |
| Имя | name | Текстовый | ограничение в 45 знаков |
| Отчество | Patronymic | Текстовый | ограничение в 45 знаков |

Продолжение таблицы 2.20

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Адрес | address | Текстовый | ограничение в 150 знаков |
| Телефон | phone | Текстовый | ограничение в 20 знаков |
| Слушатель | Listener | Числовой | ограничение в 20 знаков |
| Изменено в | modifiedBy | Дата и время | ограничение в 20 знаков |

Таблица 2.21 – Структура таблицы «role» (роль)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор роли | id | Счетчик | целое число, заполняется автоматически |
| Наименование | name | Текстовый | ограничение в 45 знаков |

Таблица 2.22 – Структура таблицы «user» (пользователь)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор пользователя | id | Счетчик | целое число, заполняется автоматически |
| Логин | login | Текстовый | ограничение в 100 знаков |
| Пароль | password | Текстовый | ограничение в 200 знаков |
| Роль | role | Числовой | ограничение в 20 знаков |
| Фамилия | surname | Текстовый | ограничение в 45 знаков |
| Имя | name | Текстовый | ограничение в 45 знаков |

Продолжение таблицы 2.22

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Отчество | patronymic | Текстовый | ограничение в 45 знаков |
| Изменено в | modifiedBy | Дата и время | ограничение в 20 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

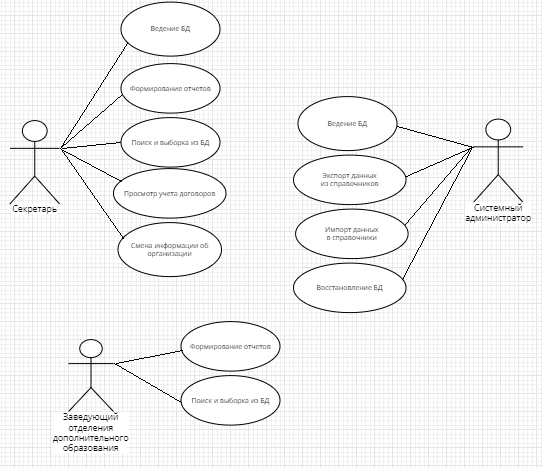


Рисунок 2.2 - Диаграмма вариантов использования

Сценарий использования – вариант использования, прецедент или же пользовательский сценарий в разратке 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 2.23

Таблица 2.23 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 2.24

Таблица 2.24 Последовательность действий актеров

|  |  |
| --- | --- |
| 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

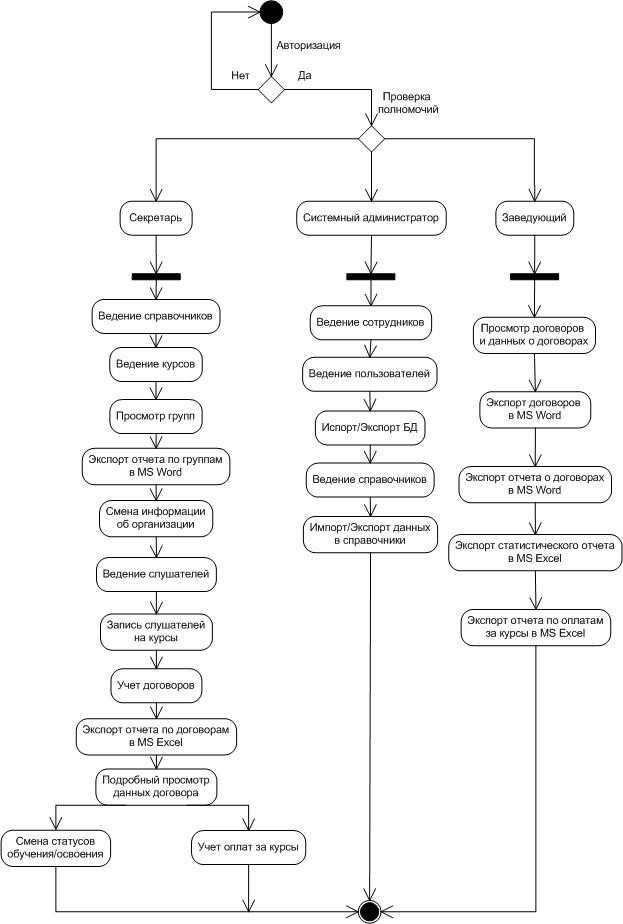


Рисунок 2.3 - Диаграмма деятельности

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

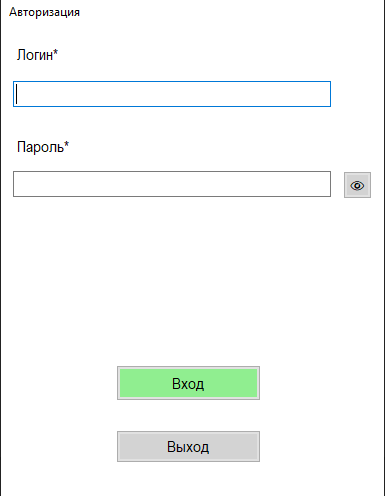
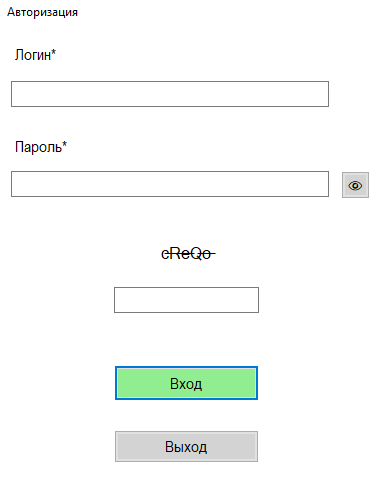
 

Рисунок 2.4 – Формы авторизации

На рисунке 2.5 показано главное меню (системного администратора)

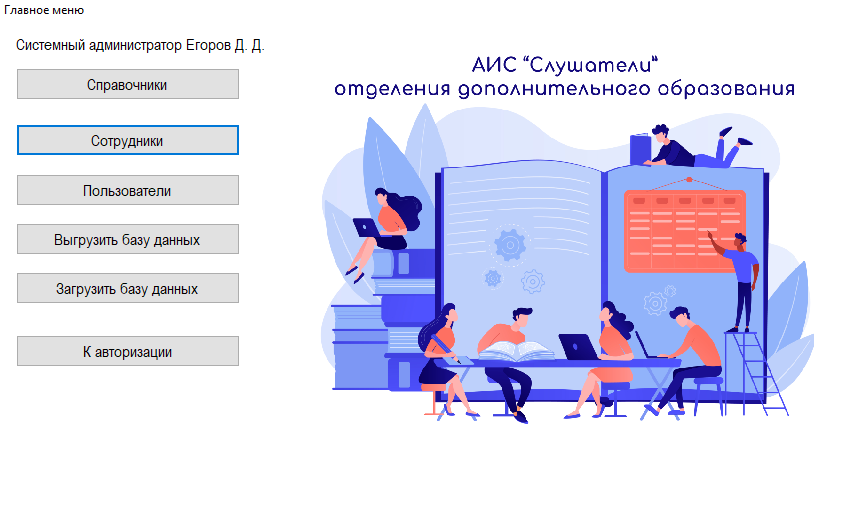


Рисунок 2.5 – Главное меню (системного администратора)

На рисунке 2.6 показано главное меню (заведующего)

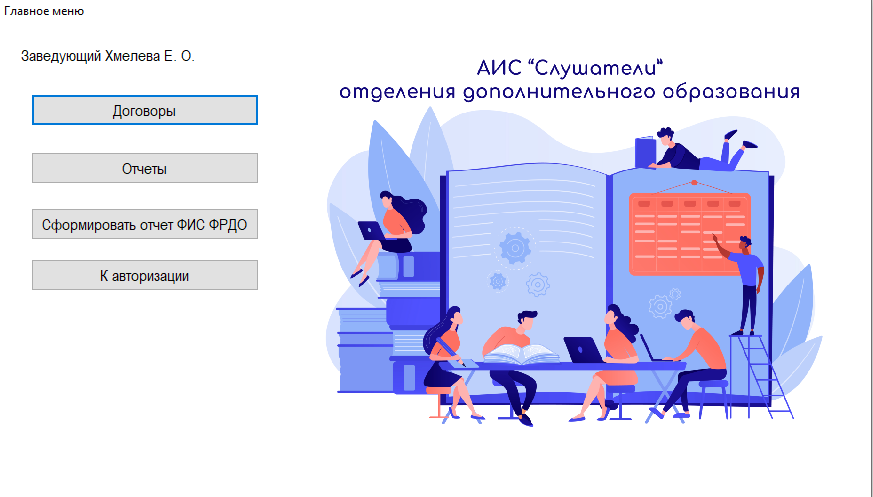


Рисунок 2.6 – Главное меню (заведующего)

На рисунке 2.7 показано главное меню (секретаря)

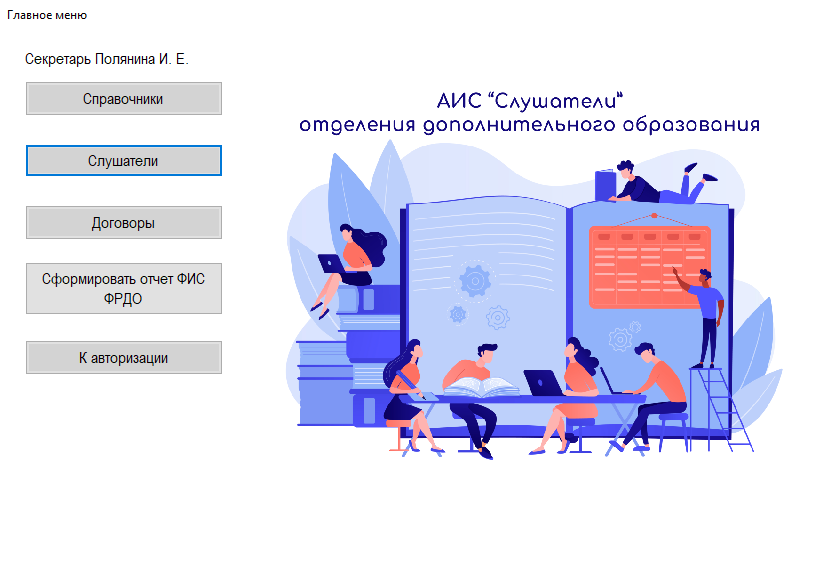


Рисунок 2.7 – Главное меню (секретаря)

На рисунке 2.8 показана форма меню «Записи»

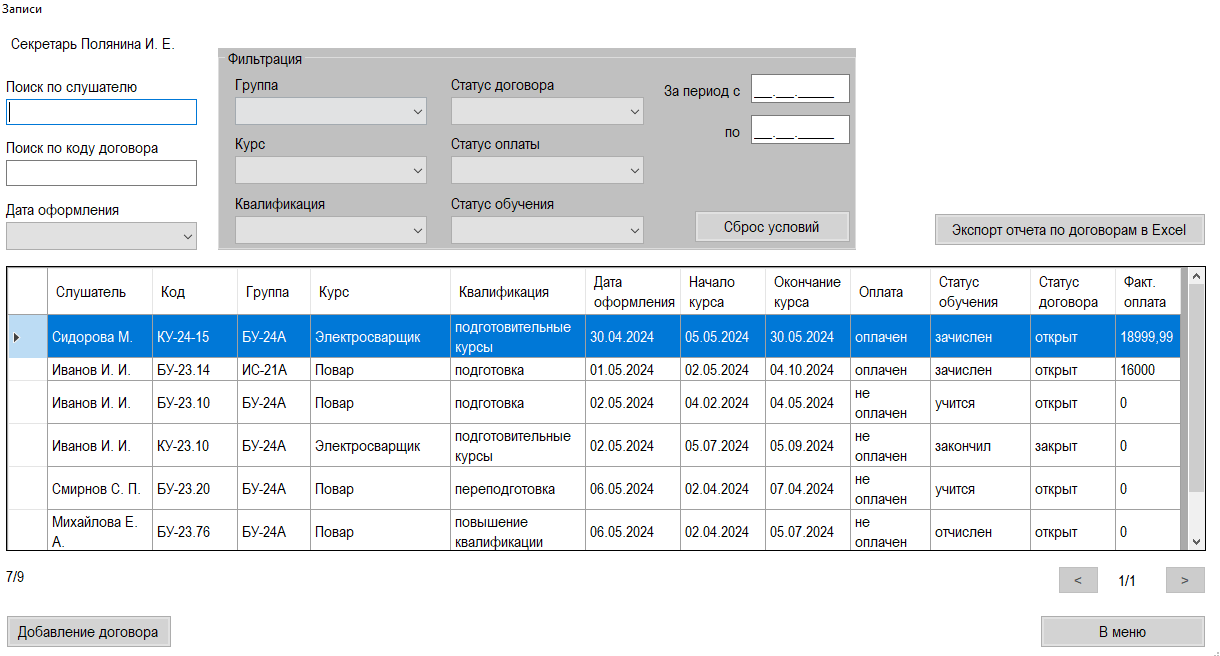


Рисунок 2.8 – Форма «Записи»

На рисунке 2.9 показана форма «Изменение договора»

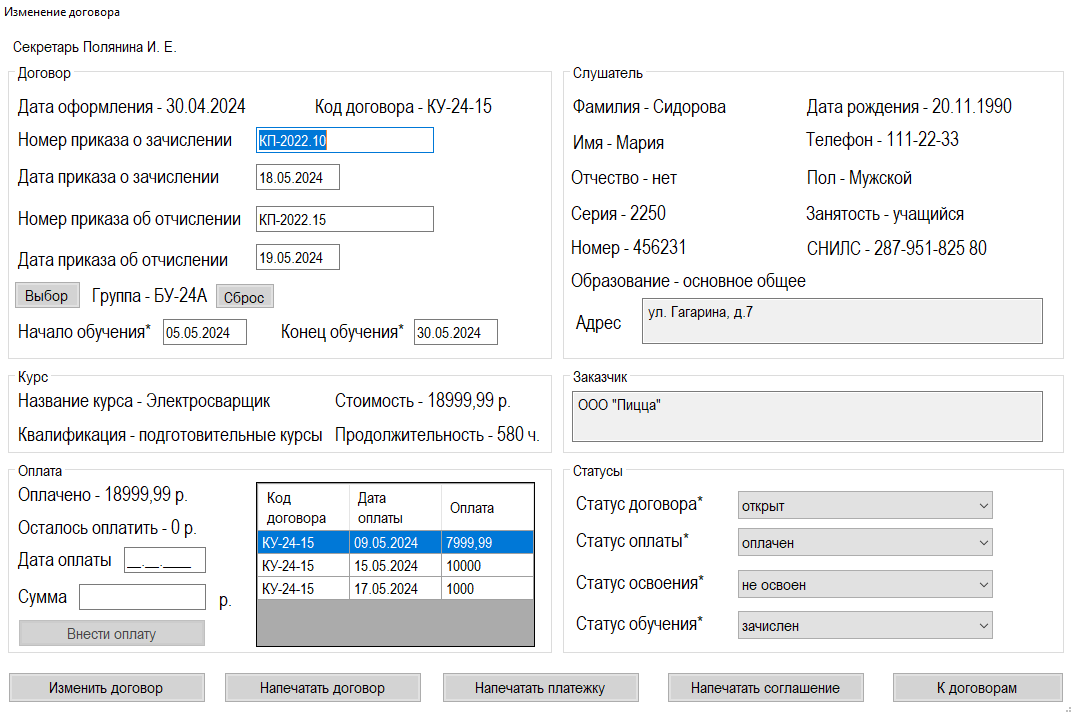


Рисунок 2.9 – Форма «Изменение договора»

На рисунке 2.10 показана форма «Курсы»

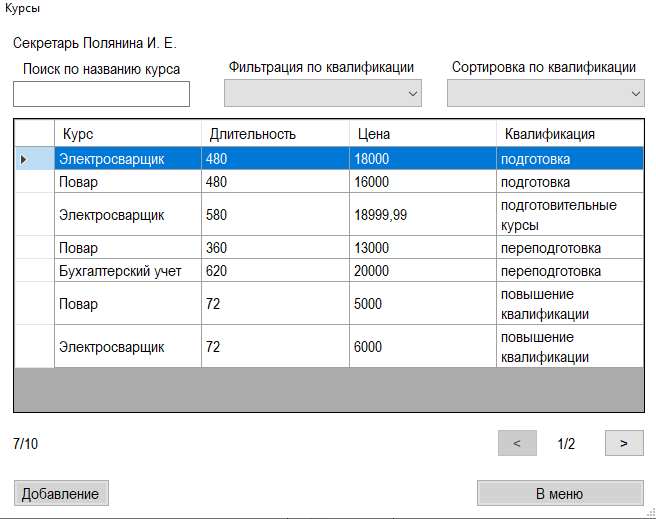


Рисунок 2.10 – Форма «Курсы»

На рисунке 2.11 показана форма «Законные представители»

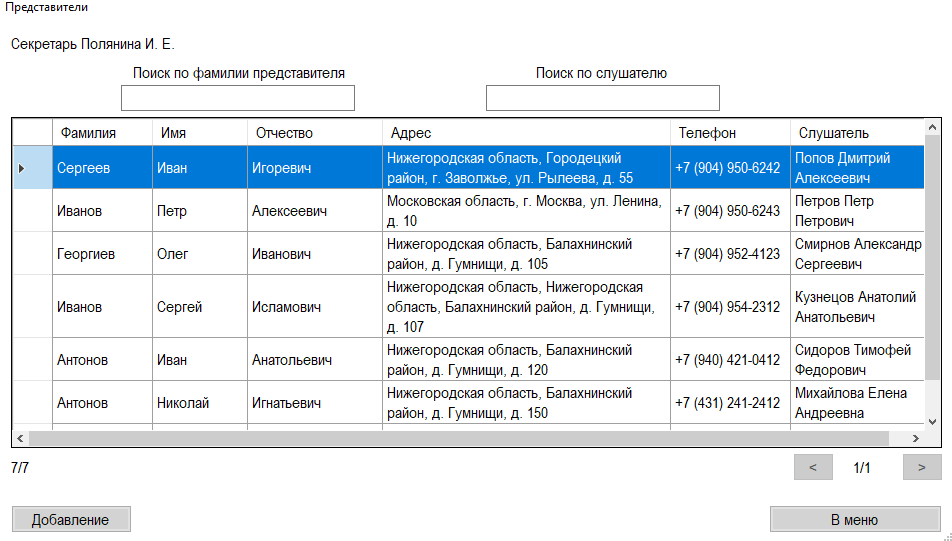


Рисунок 2.11 – Форма «Законные представители»

На рисунке 2.12 показана форма «Организации»

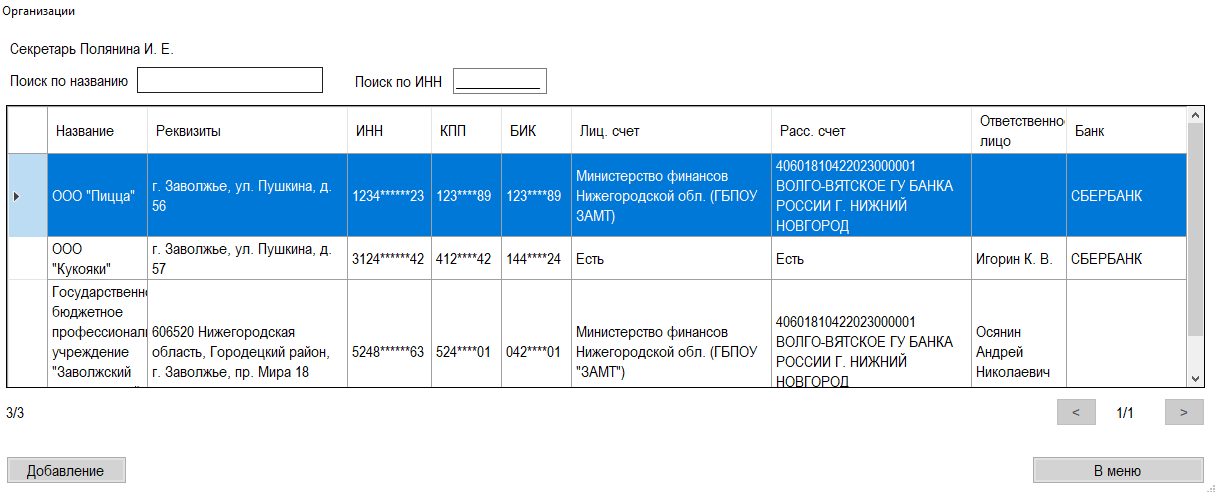


Рисунок 2.12 – Форма «Организации»

На рисунке 2.13 показана форма «Справочники»

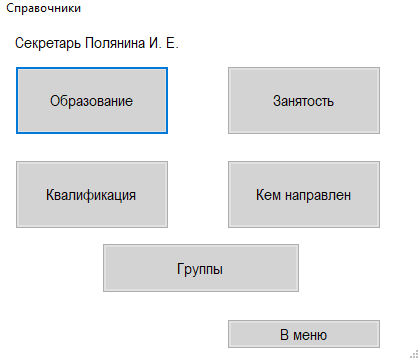


Рисунок 2.13 – Форма «Справочники»

На рисунке 2.14 показана форма «Образование»

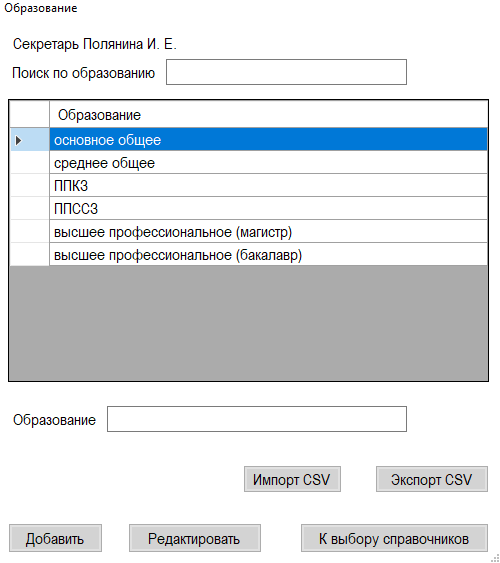


Рисунок 2.14 – Форма «Образование»

На рисунке 2.15 показана форма «Занятость»

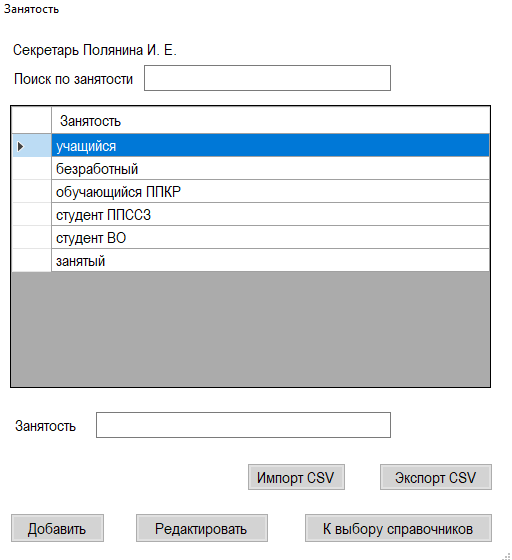


Рисунок 2.15 – Форма «Занятость»

На рисунке 2.16 показана форма «Квалификации»

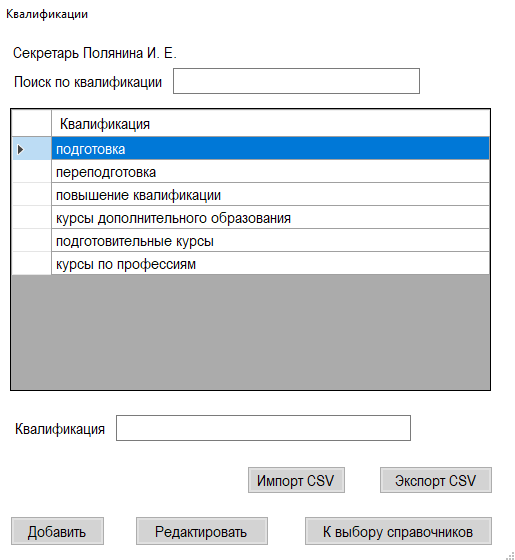


Рисунок 2.16 – Форма «Квалификации»

На рисунке 2.17 показана форма «Кем направлен»

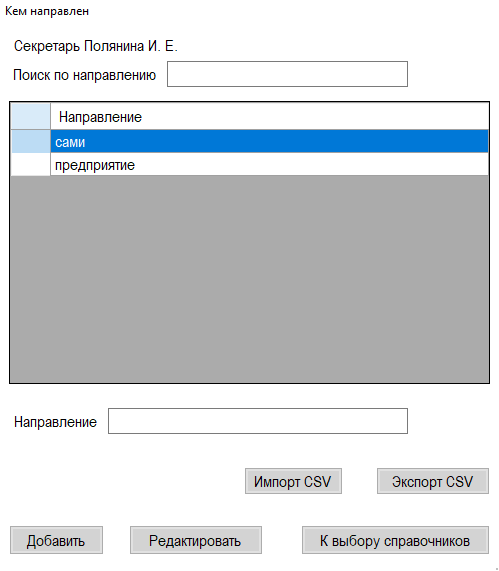


Рисунок 2.17 – Форма «Кем направлен»

На рисунке 2.18 показана форма «Группа»

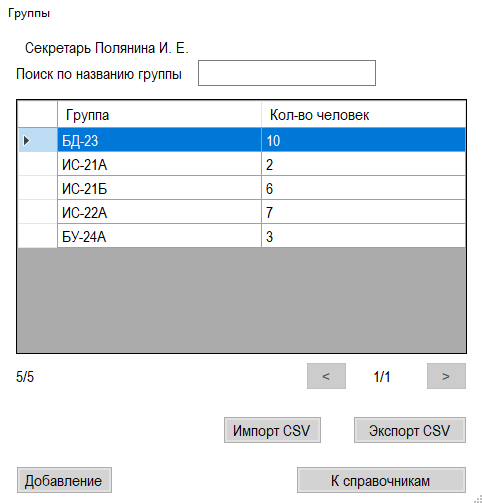


Рисунок 2.18 – Форма «Группы»

На рисунке 2.19 показана форма «Об организации»

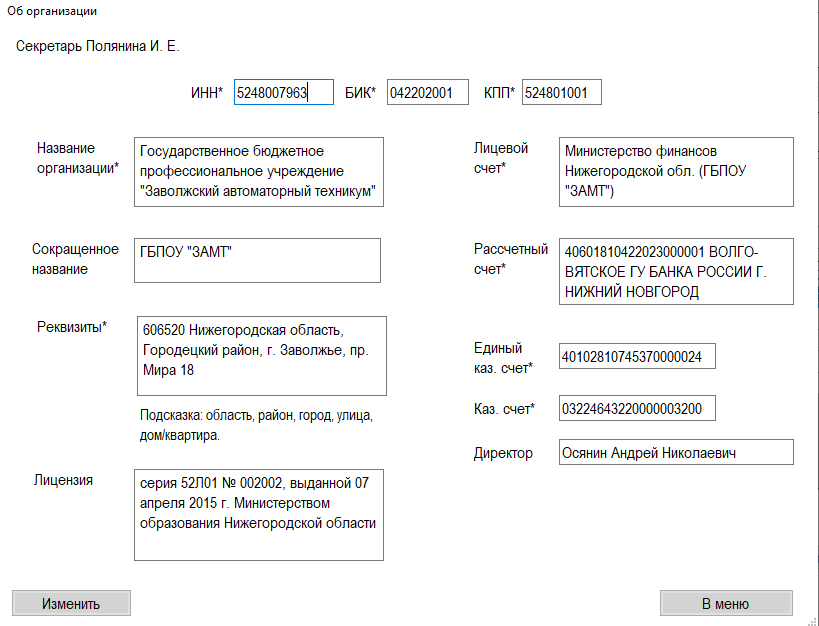


Рисунок 2.19 – Форма «Об организации»

На рисунке 2.20 показана форма «Отчеты»

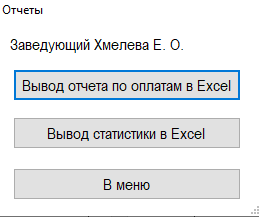


Рисунок 2.20 – Форма «Отчеты»

На рисунке 2.21 показана форма «Сотрудники»



Рисунок 2.21 – Форма «Сотрудники»

На рисунке 2.22 показана форма «Пользователи»

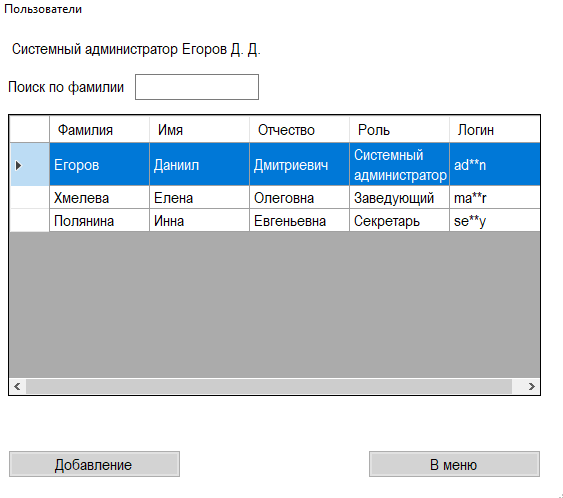


Рисунок 2.22 – Форма «Пользователи»

2.4.2 Описание основных программных модулей

public DataTable GetDataTable(string query) – получение информации из базы данных;

public DataTable GetListeners(string query) – получение слушателей из базы данных;

public bool AddRecord(int idListener, string code, string organization, string group, int idCourse, DateTime now, int capital, DateTime start, DateTime end, string representative) – запись на курс;

public List<IFirstPattern> GetInformationForPatternReport(string group) – получение информации для отчета;

public bool ExecuteRequest(string query) – выполнение запроса на обновление данных;

public bool ExecuteTransaction(List<string> queries) – выполнение транзакции;

public string[] FillComboBox(string query) – заполнение выпадающих списков;

public string GetValue(string query) – получение одного значения из базы данных;

public bool ExportDatabase(string path, string fileName) – выгрузка базы данных в виде скрипта;

public bool ImportDatabase(string path) – загрузка скрипта базы данных;

public bool ImportEducationData(string fileName) – импорт CSV;

public bool ExportData(string path, string fileName) – экспорт CSV

public bool CreatePatternDpo(string group) – создание отчета по группе;

public AuthorizationResponse Login(AuthorizationRequest candidate) – процесс авторизации;

private void ShowUser() – отображение пользователя;

public void ActivityControl(DateTime lastActivityTime) –

public void StartLoginForm() – удаление позиции из заказа;

private bool RecordIsExist() – контроль уникальности записи;

private void ClearFields() – очищение полей;

private void UpdateCountInGroup(string group) – изменение количества человек в группе;

private bool ControlRepresentativeNotExistForListener() – проверка выбранного законного представителя;

private void StartCreateFormAdd() – запуск формы добавления;

private bool ControlFillFields() – проверка заполненных полей;

private void DisplayChanged() – отслеживание изменений;

private void UpdateUserInterface() – обновление пользовательского интерфейса;

private void SwitchToRussianKeyboardLayout() – изменение языка на русский;

private void FillPictureBox(string imagePath) – выбор изображения;

private void ResetConditions() – обнуление условий;

private void OnDestroyForm() – заполнение законного представителя при закрытии формы;

private bool CanDelete(string primaryKey) – проверка на возможность удаления записи;

private void FillSort() – заполнение сортировки;

private void LoadData() – отображение данных в таблицу;

private bool SearchHaveResult() – проверка результатов поиска ;

private void UpdateAfterChanging() – обновление при закрытии формы;

public void ShowMessage(string message, string title, MessageBoxIcon icon) – отображение сообщения;

public void ControlBirthdayDate(string inputDate) – контроль даты рождения;

public string[] GetAllEmployment() – получение всех видов занятости;

public void ControlNewListener() – проверка на существование слушателя;

public void DeleteRow(string primaryKey) – удаление записи из базы данных

private void ControlPagination(Pagination pagination) – отображение пагинации;

private bool ValidatePrimaryKey(string primaryKey) – проверка первичного ключа на существование.

2.5 Применяемые меры защиты информации

Защита данных – это предупреждение несанкцианированного(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.*

В соответствии с рисунком 2.23, парольная защита необходима для обеспечения ее защиты от несанкционированного доступа в соответствии с рисунком 2.31.

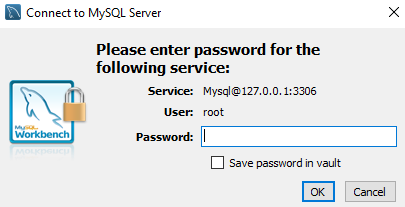


Рисунок 2.23 - Парольная защита базы данных

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 рисунком 2.24.



Рисунок 2.24 – 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 Слушатели в соотвествии с рисунком 2.25.

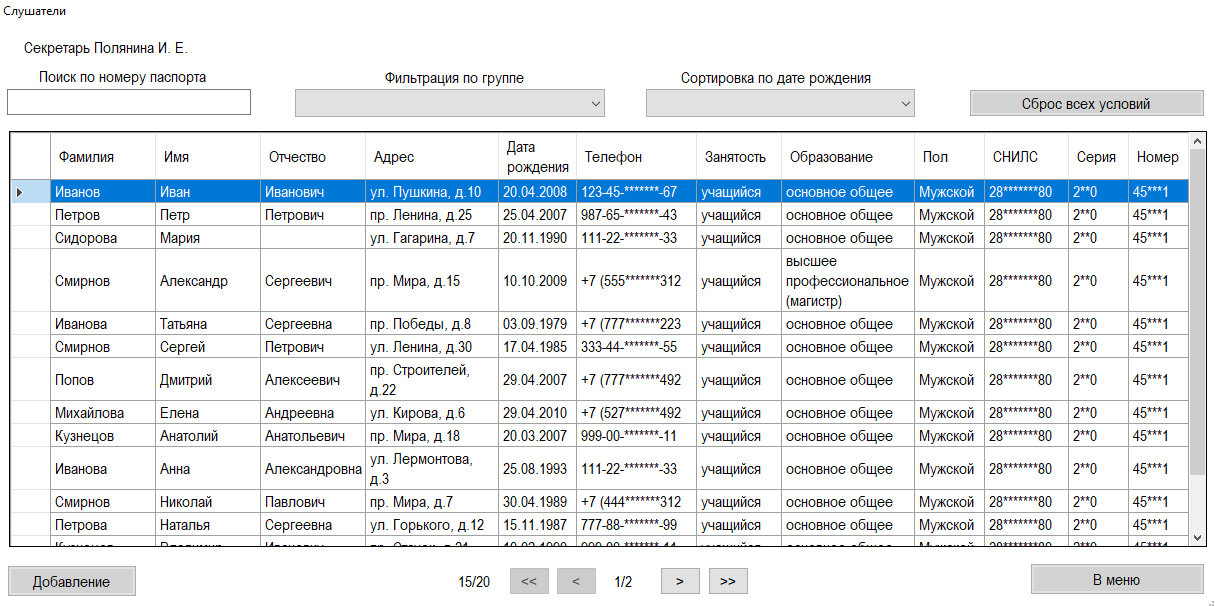


Рисунок 2.25 – Форма «Слушатели»

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 соотвествии с рисунком 2. 26.

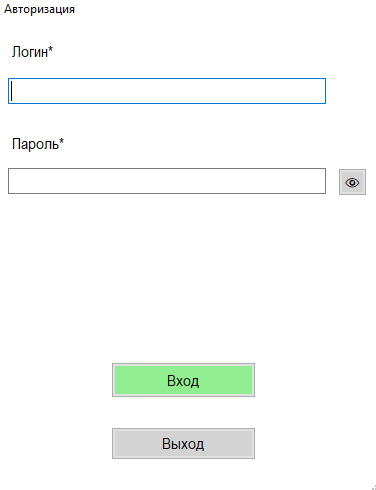


Рисунок 2.26 – Блокировка системы при отсутствии активности

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 2.27 резервное копирования осуществляется в меню администратора

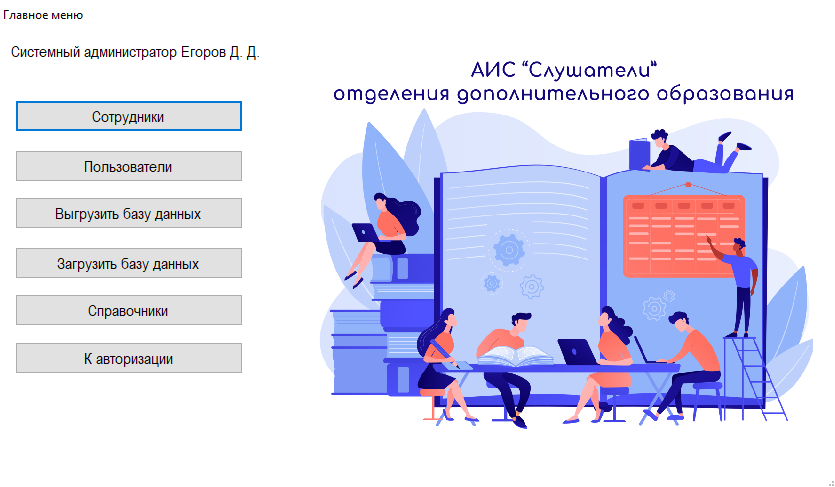


Рисунок 2.27 – Меню администратора

Диалоговое окно восстановлении базы данных, в соответствии с рисунком 2.28

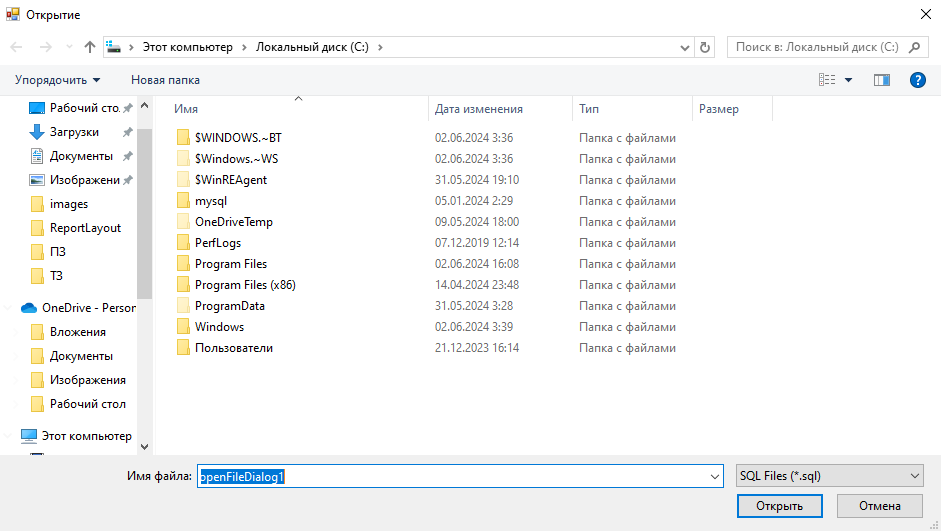


Рисунок 2.28 – Диалоговое окно восстановления базы

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

Восстановление после сбоев осуществляется при нажатии кнопки «Восстановить базу», 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** | | |
| Заполнение поля логин | Блокировка ввода русского алфавита | Корректное выполнение команды |
| Заполнение поля пароль | Блокировка ввода русского алфавита | Корректное выполнение команды |
| Кнопка «Вход» | При правильно веденных логине и пароле – переход в главное меню | Корректное выполнение команды |
| Кнопка «Вход» | При ошибке логина или пароля отображение капчи | Корректное выполнение команды |
| Кнопка «Вход» | При ошибке логина или пароля на форме авторизации с captcha  задержка на 10 секунд | Корректное выполнение команды |
| Кнопка «Выход» | Завершение работы приложения | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | | 3 | |
| Ввод в поле «Логин» | | Возможность ввода только английской раскладки и цифр | | Корректное выполнение команды |
| Ввод в поле «Пароль» | | Запрет нажатия клавиши Space | | Корректное выполнение команды |
| **Форма главного меню администратора** | | | | |
| Кнопка «Справочники» | | Переход на форму справочники | | Корректное выполнение команды |
| Кнопка «Пользователи» | | Переход на форму пользователи | | Корректное выполнение команды |
| Кнопка «Сотрудники» | | Переход на форму сотрудники | | Корректное выполнение команды |
| Кнопка «Выгрузить базу данных» | | Экспорт базы данных | | Корректное выполнение команды |
| Кнопка «Загрузить базу данных» | | Импорт базы данных | | Корректное выполнение команды |
| **Форма главного меню секретаря** | | | | |
| Кнопка «Справочники» | | Переход на форму справочники | | Корректное выполнение команды |
| Кнопка «Слушатели» | | Переход на форму слушатели | | Корректное выполнение команды |
| Кнопка «Договоры» | | Переход на форму договоры | | Корректное выполнение команды |
| Кнопка «Сформировать отчет ФИС ФРДО» | | Открытие окна с выбором группы | | Корректное выполнение команды |
| **Форма главного меню заведующего отделением** | | | | |
| Кнопка «Договоры» | | Переход на форму договоры | | Корректное выполнение команды |
| Кнопка «Отчеты» | | Переход на форму отчеты | | Корректное выполнение команды |
| Кнопка «Сформировать отчет ФИС ФРДО» | | Открытие окна с выбором группы | | Корректное выполнение команды |
| **Форма Справочники** | | | | |
| Кнопка «Образование» | | Переход на форму образование | | Корректное выполнение команды |
| Кнопка «Квалификация» | | Переход на форму квалификация | | Корректное выполнение команды |
| Кнопка «занятость» | | Переход на форму занятость | | Корректное выполнение команды |
| Кнопка «Кем направлен» | | Переход на форму кем направлен | | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | | 2 | 3 |
| Кнопка «Группы» | | Переход на форму группы | Корректное выполнение команды |
| Кнопка «Организации» | | Переход на форму организации | Корректное выполнение команды |
| Кнопка «Курсы» | | Переход на форму курсы | Корректное выполнение команды |
| Кнопка «Законные представители» | | Переход на форму законные представители | Корректное выполнение команды |
| Кнопка «Об организации» | | Переход на форму об организации | Корректное выполнение команды |
| Кнопка «В меню» | | Выход в меню | Корректное выполнение команды |
| **Форма Образование** | | | |
| Ввод в поле «Поиск по образованию» | Возможность ввода только русской раскладки, без цифр, без пробелов | | Корректное выполнение команды |
| Ввод в поле «Образование» | Возможность ввода только русской раскладки, без цифр, без пробелов | | Корректное выполнение команды |
| Кнопка «Добавить» | При заполнении всех полей и отсутствии в базе данных идентичной записи происходит добавление образования, очищается поле | | Корректное выполнение команды |
| Кнопка «Добавить» | При заполнении не всех полей отображается сообщение «Заполните поле вида образования» | | Корректное выполнение команды |
| Кнопка «Добавить» | При присутствии идентичной записи в базе данных отображается сообщение «Данный вид образования уже существует» | | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | |
| Кнопка «Редактировать» | При заполнении всех полей и отсутствии в базе данных идентичной записи происходит редактирование образования, очищается поле | | Корректное выполнение команды |
| Кнопка «Редактировать» | При заполнении не всех полей отображается сообщение «Заполните поле вида образования» | | Корректное выполнение команды |
| Кнопка «Редактировать» | При присутствии идентичной записи в базе данных отображается сообщение «Данный вид образования уже существует» | | Корректное выполнение команды |
| Нажатие на таблицу правой кнопкой мыши | Открытие контекстного меню с пунктом «Удаление» | | Корректное выполнение команды |
| Нажатие на пункт меню «Удаление» | При использовании данной записи в других таблицах отображается сообщение «Нельзя удалить данную запись» | | Корректное выполнение команды |
| Нажатие на пункт меню «Удаление» | При отсутствии данной записи в других таблицах отображается сообщение «Вы действительно желаете удалить данную запись?» При нажатии «Да» произойдет удаление записи | | Корректное выполнение команды |
| Кнопка «Экспорт CSV» | Открытие модального окна с выбором папки | | Корректное выполнение команды |
| Выбор папки и нажатие кнопки «ОК» | Выполнится экспорт таблицы в файл csv и отобразится сообщение «Данные успешно экспортированы» | | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | | 2 | 3 |
| Кнопка «Импорт CSV» | | Открытие модального окна с выбором файла | Корректное выполнение команды |
| Выбор файла и нажатие кнопки «ОК» | | Выполнится импорт в таблицу из файла csv и отобразится сообщение «Данные успешно импортированы» | Корректное выполнение команды |
| **Форма Квалификация** | | | |
| Ввод в поле «Поиск по квалификации» | Возможность ввода только русской раскладки, без цифр, без пробелов | | Корректное выполнение команды |
| Ввод в поле «Квалификация» | Возможность ввода только русской раскладки, без цифр, без пробелов | | Корректное выполнение команды |
| Кнопка «Добавить» | При заполнении всех полей и отсутствии в базе данных идентичной записи происходит добавление квалификации, очищается поле | | Корректное выполнение команды |
| Кнопка «Добавить» | При заполнении не всех полей отображается сообщение «Заполните поле вида квалификации» | | Корректное выполнение команды |
| Кнопка «Добавить» | При присутствии идентичной записи в базе данных отображается сообщение «Данный вид квалификации уже существует» | | Корректное выполнение команды |
| Кнопка «Редактировать» | При заполнении всех полей и отсутствии в базе данных идентичной записи происходит редактирование квалификации, очищается поле | | Корректное выполнение команды |
| Кнопка «Редактировать» | При заполнении не всех полей отображается сообщение «Заполните поле вида квалификации» | | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Кнопка «Редактировать» | При присутствии идентичной записи в базе данных отображается сообщение «Данный вид образования уже существует» | Корректное выполнение команды |
| Нажатие на таблицу правой кнопкой мыши | Открытие контекстного меню с пунктом «Удаление» | Корректное выполнение команды |
| Нажатие на пункт меню «Удаление» | При использовании данной записи в других таблицах отображается сообщение «Нельзя удалить данную запись» | Корректное выполнение команды |
| Нажатие на пункт меню «Удаление» | При отсутствии данной записи в других таблицах отображается сообщение «Вы действительно желаете удалить данную запись?» При нажатии «Да» произойдет удаление записи | Корректное выполнение команды |
| Кнопка «Экспорт CSV» | Открытие модального окна с выбором папки | Корректное выполнение команды |
| Выбор папки и нажатие кнопки «ОК» | Выполнится экспорт таблицы в файл csv и отобразится сообщение «Данные успешно экспортированы» | Корректное выполнение команды |
| Кнопка «Импорт CSV» | Открытие модального окна с выбором файла | Корректное выполнение команды |
| Выбор файла и нажатие кнопки «ОК» | Выполнится импорт в таблицу из файла csv и отобразится сообщение «Данные успешно импортированы» | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| **Форма Занятость** | | |
| Ввод в поле «Поиск по занятости» | Возможность ввода только русской раскладки, без цифр, без пробелов | Корректное выполнение команды |
| Ввод в поле «Занятость» | Возможность ввода только русской раскладки, без цифр, без пробелов | Корректное выполнение команды |
| Кнопка «Добавить» | При заполнении всех полей и отсутствии в базе данных идентичной записи происходит добавление занятости, очищается поле | Корректное выполнение команды |
| Кнопка «Добавить» | При заполнении не всех полей отображается сообщение «Заполните поле вида занятости» | Корректное выполнение команды |
| Кнопка «Добавить» | При присутствии идентичной записи в базе данных отображается сообщение «Данный вид занятости уже существует» | Корректное выполнение команды |
| Кнопка «Редактировать» | При заполнении всех полей и отсутствии в базе данных идентичной записи происходит редактирование занятости, очищается поле | Корректное выполнение команды |
| Кнопка «Редактировать» | При заполнении не всех полей отображается сообщение «Заполните поле вида занятости» | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Кнопка «Редактировать» | При присутствии идентичной записи в базе данных отображается сообщение «Данный вид занятости уже существует» | Корректное выполнение команды |
| Нажатие на таблицу правой кнопкой мыши | Открытие контекстного меню с пунктом «Удаление» | Корректное выполнение команды |
| Нажатие на пункт меню «Удаление» | При использовании данной записи в других таблицах отображается сообщение «Нельзя удалить данную запись» | Корректное выполнение команды |
| Нажатие на пункт меню «Удаление» | При отсутствии данной записи в других таблицах отображается сообщение «Вы действительно желаете удалить данную запись?» При нажатии «Да» произойдет удаление записи | Корректное выполнение команды |
| Кнопка «Экспорт CSV» | Открытие модального окна с выбором папки | Корректное выполнение команды |
| Выбор папки и нажатие кнопки «ОК» | Выполнится экспорт таблицы в файл csv и отобразится сообщение «Данные успешно экспортированы» | Корректное выполнение команды |
| Кнопка «Импорт CSV» | Открытие модального окна с выбором файла | Корректное выполнение команды |
| Выбор файла и нажатие кнопки «ОК» | Выполнится импорт в таблицу из файла csv и отобразится сообщение «Данные успешно импортированы» | Корректное выполнение команды |
| **Форма «Кем направлен»** | | |
| Ввод в поле «Поиск по квалификации» | Возможность ввода только русской раскладки, без цифр, без пробелов | Корректное выполнение команды |
| Ввод в поле «Занятость» | Возможность ввода только русской раскладки, без цифр, без пробелов | Корректное выполнение команды |
| Кнопка «Добавить» | При заполнении всех полей и отсутствии в базе данных идентичной записи происходит добавление занятости, очищается поле | Корректное выполнение команды |
| Кнопка «Добавить» | При заполнении не всех полей отображается сообщение «Заполните поле вида занятости» | Корректное выполнение команды |
| Кнопка «Добавить» | При присутствии идентичной записи в базе данных отображается сообщение «Данный вид занятости уже существует» | Корректное выполнение команды |
| Кнопка «Редактировать» | При заполнении всех полей и отсутствии в базе данных идентичной записи происходит редактирование занятости, очищается поле | Корректное выполнение команды |
| Кнопка «Редактировать» | При заполнении не всех полей отображается сообщение «Заполните поле вида занятости» | Корректное выполнение команды |
| Кнопка «Редактировать» | При присутствии идентичной записи в базе данных отображается сообщение «Данный вид занятости уже существует» | Корректное выполнение команды |
| Нажатие на таблицу правой кнопкой мыши | Открытие контекстного меню с пунктом «Удаление» | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Нажатие на пункт меню «Удаление» | При использовании данной записи в других таблицах отображается сообщение «Нельзя удалить данную запись» | Корректное выполнение команды |
| Нажатие на пункт меню «Удаление» | При отсутствии данной записи в других таблицах отображается сообщение «Вы действительно желаете удалить данную запись?» При нажатии «Да» произойдет удаление записи | Корректное выполнение команды |
| Кнопка «Экспорт CSV» | Открытие модального окна с выбором папки | Корректное выполнение команды |
| Выбор папки и нажатие кнопки «ОК» | Выполнится экспорт таблицы в файл csv и отобразится сообщение «Данные успешно экспортированы» | Корректное выполнение команды |
| Кнопка «Импорт CSV» | Открытие модального окна с выбором файла | Корректное выполнение команды |
| Выбор файла и нажатие кнопки «ОК» | Выполнится импорт в таблицу из файла csv и отобразится сообщение «Данные успешно импортированы» | Корректное выполнение команды |
| **Форма Группы** | | |
| Ввод в поле «Поиск по названию группы» | Возможность ввода только русской раскладки, без цифр, без пробелов | Корректное выполнение команды |
| Кнопка «Экспорт CSV» | Открытие модального окна с выбором папки | Корректное выполнение команды |
| Выбор папки и нажатие кнопки «ОК» | Выполнится экспорт таблицы в файл csv и отобразится сообщение «Данные успешно экспортированы» | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | | 3 |
| Кнопка «Импорт CSV» | Открытие модального окна с выбором файла | | Корректное выполнение команды |
| Выбор файла и нажатие кнопки «ОК» | Выполнится импорт в таблицу из файла csv и отобразится сообщение «Данные успешно импортированы» | | Корректное выполнение команды |
| Нажатие на таблицу правой кнопкой мыши | Открытие контекстного меню с пунктом «Удаление» | | Корректное выполнение команды |
| Нажатие на пункт меню «Удаление» | При использовании данной записи в других таблицах отображается сообщение «Нельзя удалить данную запись» | | Корректное выполнение команды |
| Нажатие на пункт меню «Удаление» | При отсутствии данной записи в других таблицах отображается сообщение «Вы действительно желаете удалить данную запись?» При нажатии «Да» произойдет удаление записи | | Корректное выполнение команды |
| Кнопка «Добавление» | Переход на форму добавления группы | | Корректное выполнение команды |
| **Форма Группы (Добавление)** | | | |
| Ввод в поле «Название группы» | | Разрешен ввод русских букв, символа тире | Корректное выполнение команды |
| Ввод в поле «Кол-во людей» | | Разрешен ввод только цифр, не более 2 цифр | Корректное выполнение команды |
| Кнопка «Добавить» | | При заполнении всех полей произойдет добавление и очистка полей | Корректное выполнение команды |
| Кнопка «Добавить» | | При заполнении не всех полей отобразится сообщение «Должны быть заполнены все обязательные поля» | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Кнопка «К группам» | Переход на форму «Группы» (просмотр) | Корректное выполнение команды |
| **Форма Организации** | | |
| Ввод в поле «Поиск по названию» | Разрешен ввод только русских символов, без цифр, без специальных символов, с заглавной буквы | Корректное выполнение команды |
| Ввод в поле «Поиск по ИНН» | Разрешен ввод только цифр | Корректное выполнение команды |
| Нажатие на таблицу правой кнопкой мыши | Открытие контекстного меню с пунктом «Удаление» | Корректное выполнение команды |
| Нажатие на пункт меню «Удаление» | При использовании данной записи в других таблицах отображается сообщение «Невозможно удалить организацию, которая содержится в договорах» | Корректное выполнение команды |
| Нажатие на пункт меню «Удаление» | При отсутствии данной записи в других таблицах отображается сообщение «Вы действительно желаете удалить данную запись?» При нажатии «Да» произойдет удаление записи | Корректное выполнение команды |
| Кнопка «Добавление» | Переход на форму добавления организации | Корректное выполнение команды |
| Форма Организации (добавление) | | |
| Ввод в поле ИНН | Разрешен ввод только цифр | Корректное выполнение команды |
| Ввод в поле БИК | Разрешен ввод только цифр | Корректное выполнение команды |
| Ввод в поле КПП | Разрешен ввод только цифр | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Ввод в поле «Каз. Счет» | Разрешен ввод только цифр | Корректное выполнение команды |
| Ввод в поле «Единый каз. Счет» | Разрешен ввод только цифр | Корректное выполнение команды |
| Ввод в поле «Директор» | Разрешен ввод только русских символов с заглавной буквы | Корректное выполнение команды |
| Кнопка «Добавить» | При заполнении всех полей произойдет добавление и очистка полей | Корректное выполнение команды |
| Кнопка «Добавить» | При заполнении не всех полей отобразится сообщение «Должны быть заполнены все обязательные поля» | Корректное выполнение команды |
| Кнопка «К организациям» | Переход на форму организации | Корректное выполнение команды |
| **Форма «Законные представители"** | | |
| Ввод в поле «Поиск по фамилии представителя» | Разрешен ввод только русских символов, без цифр, без специальных символов, с заглавной буквы | Корректное выполнение команды |
| Ввод в поле «Поиск по слушателю» | Разрешен ввод только русских символов, без цифр, без специальных символов, с заглавной буквы | Корректное выполнение команды |
| Кнопка «Добавление» | Переход на форму добавления законных представителей | Корректное выполнение команды |
| **Форма «Законные представители» (Добавление)** | | |
| Ввод в поле «Фамилия» | Разрешен ввод только русских символов, без цифр, без специальных символов, с заглавной буквы | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Ввод в поле «Имя» | Разрешен ввод только русских символов, без цифр, без специальных символов, с заглавной буквы | Корректное выполнение команды |
| Ввод в поле «Отчество» | Разрешен ввод только русских символов, без цифр, без специальных символов, с заглавной буквы | Корректное выполнение команды |
| Ввод в поле «Телефон» | Разрешен ввод только цифр | Корректное выполнение команды |
| Кнопка «Выбор» | Открывает форму просмотра слушателей | Корректное выполнение команды |
| Кнопка «Добавить» | При заполнении всех полей добавляет нового законного представителя | Корректное выполнение команды |
| Кнопка «Добавить» | При заполнении не всех полей отображается сообщение «Заполните все обязательные поля» | Корректное выполнение команды |
| Кнопка «К представителям» | Переход на форму просмотра законных представителей | 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 потрачен 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. | 50000 |
| ДМ | Среднее количество рабочих дней в месяце | дни | 24 |
| Счас | Стоимость час работы программиста | руб. | 150 |
| ЦЭЛ | Тариф за 1 кВт/час | руб. | 4,48 |
| Тчас | Количество часов работы над разработкой в день | час | 5 |
| P | Мощность, потребляемая ЭВМ | кВт | 0,75 |
| 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 **=** 150 **×** 5**×** 24 = 18000 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 4,24 рублей, а потребляемая мощность ПК составляет 0,7 кВт в час.

|  |  |
| --- | --- |
| Зэл **=** 200 **×** 4,48 **×** 0,75 = 672 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 **×** 50000 = 2500 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 = 124,8 руб. |  |

Выполнив все расчеты, определим себестоимость программного продукта.

|  |  |
| --- | --- |
| Спп = 18000 + 672 + 2500 + 124,8 = 21296 рублей ≈ 21300 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 = 21300 руб

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

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

Было принято решение продавать информационную систему по ведению базы данных за 4000 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 | 4000 | 4 | 11200 | 11200 |
| 2 год | 4000 | 4 | 11200 | 21400 |
| Итого | 4000 | 8 | 21400 |  |

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.) | 21300 |
| Цена разработки ПП (руб.) | 21300 |
| Отпускная цена (руб.) | 4000 |
| 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 «Слушатель», отвечающая поставленным требованиям.

В процессе написания программного кода, использовался язык программирования высокого уровня. Были улучшены навыки по работе с платформой .NET, БД MySQL, 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, 2019г.
3. Троелсен Эндрю, Джепикс Филипп, Язык программирования C#7 и платформы .NET и .NET Core, 2019г.
4. Евдокимов П. В.,C# на примерах, 2019г.
5. Вагнер Билл., Наиболее эффективное программирование на C#. 50 способов улучшения кода, 2019г.
6. Мюллер Джон Поль, Семпф Билл ., C# для чайников ,2019г.