Введение

В наше время автоматизация различных процессов, протекающих на предприятии, является неотъемлемой частью повышения производительности труда и позволяет сэкономить массу времени и средств.

Слишком большие материальные затраты и затраты по трудовому времени на ведение и контроль документов, содержание их в неактуальном состоянии, а также наличие множества ошибок и сложность формирования отчетов – все это говорит о необходимости автоматизации.

В настоящее время специалисты разрабатывают и внедряют на предприятия для автоматизации процессов такие программы, как информационные системы.

Информационная система (ИС) – это система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы, которые обеспечивают и распространяют информацию.

ИС предназначена для своевременного обеспечения надлежащих людей надлежащей информацией, то есть для удовлетворения конкретных информационных потребностей в рамках определенной предметной области.

Регистрация в общежитии техникума является важным пунктом для выдачи студентам временной прописки. Обычно регистрация всех студентов проходит в один день, что значительно усложняет весь этот процесс, а заполнение документации в нескольких экземплярах еще и сильно затягивает его.

Заполняется и хранится вся документация в бумажном виде, что очень неудобно для хранения, поиска и выборки нужной сотрудникам информации. Также высока вероятность написания ошибочных данных, и как следствие, это приводит к большим проблемам.

Во избежание всех этих проблем было принято решение разработать и внедрить в Томской техникум информационных технологий свою информационную систему регистрации студентов в общежитии, которая бы ускорила и облегчила заполнение, редактирование и удаление данных, а также формирование различных отчетов.

ОГБПОУ «Томский техникум информационных технологий» – областное государственное бюджетное профессиональное образовательное учреждение, расположенное по адресу г. Томск, ул. Герцена 18, занимается предоставлением услуг обучения на базе 9 и 11 классов по разным IT-специальностям, а также подготовкой и повышением квалификации охранников 4-6 разрядов, предоставляет услуги тира. Численность сотрудников компании составляет порядка 70 человек. Из них большая часть преподавателей, которые занимаются самим процессом обучения, имеются также работники IT-отдела по разработке, который занимается сайтом техникума: поддержание его работоспособности, добавлением нового функционала, насыщением его контентом, разрабатывает различное дополнительное ПО; по обслуживанию оборудования: следит за внутри техникумовским кластером серверов, производит установку и настройку оборудования техникума, а также другими задачами, связанными с IT сферой. Ежедневный документа-оборот компании составляет примерно 500 документов. Из них примерно 1000 документов регистрируется и подтверждается в СЭД “ТЕЗИС” (СЭД – система электронного документооборота).

Основная задача сотрудников IT-отдела обеспечение бесперебойной работы определенного оборудования техникума, поддержание сайта техникума, обслуживание сервера, а также выполнение других задач, связанных с IT областью, занимается техподдержкой 1 и 2 уровней.

Цель дипломного проекта – разработать автоматизированную информационную систему для Томского техникума информационных технологий для регистрации студентов в общежитии.

В дипломном проекте представлены: эскизный проект, алгоритмы решения задач в виде блок схем, ERD диаграмма, листинг кода программы с комментариями и подробное руководство пользователя с иллюстрациями.

1 ОБЩАЯ ЧАСТЬ

1.1 Постановка задачи

Требуется разработать информационную систему для регистрации студентов в общежитии техникума. Система должна предусматривать хранение информации обо всех студентах, прошедших регистрацию (их паспортные и контактные данные, данные представителей и родственников, данные заключенного договора и др.), а также техническую информацию о самом общежитии (список комнат, список работников и пр.). Иметь возможность формирования ряда документации в различных форматах для дальнейшей их отправки на печать.

Система должна быть ориентирована как на обычного пользователя (сотрудника общежития), который будет работать со всеми имеющимися данными, не касающихся самих пользователей, а также и на администратора, в возможности которого входит управление списком пользователей, имеющим доступ к ПО. Иметь автоматизированный и ручной ввод данных. Должно быть организовано разграничение прав доступа к данным. Развитые механизмы поиска.

При добавлении данных, должна быть обеспечена целостность и непротиворечивость данных, а также предусмотрен контроль ввода некорректной информации. У системы должен быть простой, понятный и красивый интерфейс с анимацией. Нужно предусмотреть маски для ввода данных (телефон, паспортные данные и др.).

В системе должна быть предусмотрена авторизация пользователей с защитой от перебора паролей, сами пароли должны храниться в шифрованной виде, так как ПО хранит паспортные данные.

Обычный пользователь с помощью системы должен иметь возможность решать следующие задачи:

1. регистрировать студентов в общежитии;
2. просматривать, редактировать и удалять информацию по каждому студенту (его данные, представителей, родственников, данные договора);
3. просматривать список всех зарегистрированных студентов;
4. просматривать список всех заключенных договоров;
5. просматривать список родственников по каждому студенту;
6. экспортировать данные студентов в MS Word и MS Excel;
7. сортировать таблицы по любому полю;
8. производить поиск в таблицах;
9. просматривать список всех комнат и данных по ним, в том числе количество свободных мест на данный момент;
10. изменять список имеющихся комнат.

Пользователю с правами администратора требуется предоставить следующий функционал:

1. возможность создавать новых пользователей;
2. просматривать и редактировать список пользователей;
3. иметь возможность задать новый логин или пароль выбранному пользователю;
4. изменять данные своего профиля.

### Основные функции программы:

1. Функция блокировки кнопки входа на 1 минуту после трех неудачных попыток авторизации с последующим увеличением времени на 1 минуту после каждой следующей попытки;
2. Функция авторизации пользователя;
3. Функция добавления студента (имя, фамилия, отчество, телефон, дата рождения, группа, и возможность загрузить фото);
4. Функция добавления паспорта студента (серия, номер, кем выдан, дата выдачи, адрес прописки, код подразделения);
5. Функция добавления представителей студента (имя, фамилия, отчество, телефон, кем приходится, место работы, место жительства, домашний телефон);
6. Функция добавления паспорта представителя (серия, номер, кем выдан, дата выдачи, адрес прописки, код подразделения);
7. Функция добавления списка родственников студента (имя, фамилия, отчество, телефон, степень родства, адрес);
8. Функция добавления договора;
9. Функция вывода списка всех зарегистрированных студентов;
10. Функция мягкого поиска студента в списке студентов;
11. Функция удаления студента;
12. Функция редактирования данных студента
13. Функция редактирования паспортных данных студента;
14. Функция редактирования данных представителя студента;
15. Функция редактирования паспортных данных представителя;
16. Функция редактирования списка родственников студента;
17. Функция редактирования договора студента;
18. Функция удаления родственника у студента;
19. Функция вывода списка родственников каждого студента;
20. Функция мягкого поиска нужного списка родственников;
21. Функция вывода списка всех договоров;
22. Функция мягкого поиска нужного договора в списке договоров;
23. Функция удаления договора студента;
24. Функция вывода списка всех комнат;
25. Функция удаления комнаты;
26. Функция добавления комнаты;
27. Функция формирования документа «Договор о найме жилого помещения в специализированном жилом фонде»;
28. Функция формирования документа «Дополнительное соглашение к договору о найме жилого помещения в специализированной жилом фонде»;
29. Функция формирования документа «Анкета студента»;
30. Функция изменения данных текущего профиля;
31. Функция вывода списка всех пользователей программы;
32. Функция изменения логина и/или пароля для выбранного пользователя;
33. Функция добавления нового пользователя;
34. Функция сортировки таблицы.

1.2 Структура данных

База данных информационной системы будет состоять из нескольких таблиц. Взаимосвязь таблиц представлена на рисунке 1. В таблицах 1–11 представлено подробное описание данных.

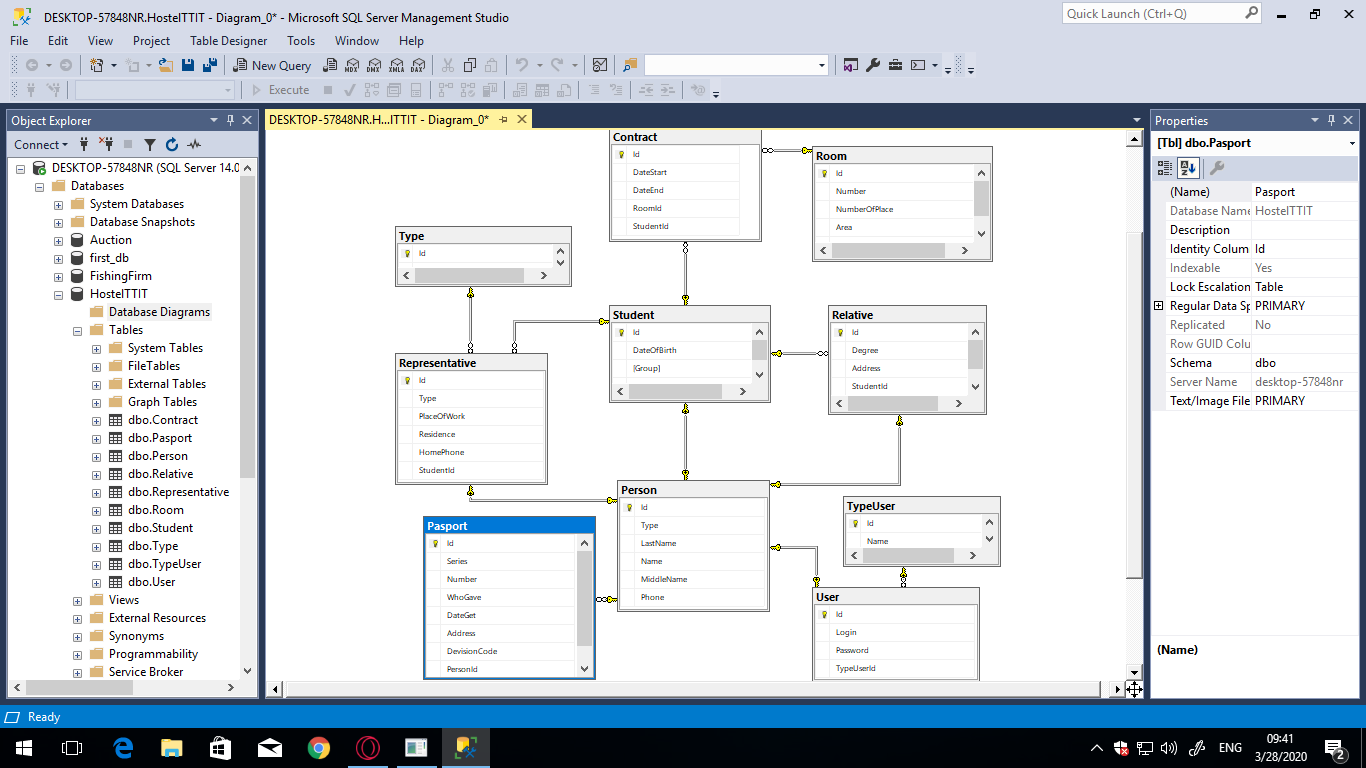


Рисунок 1 – Структура базы данных.

Таблица 1 – Person

|  |  |  |  |
| --- | --- | --- | --- |
| №  п-п | Название поля | Тип данных | Описание |
| 1 | Id | INT (IDENTITY) | Идентификатор персоны |
| 2 | TypePersonId | INT | Идентификатор типа персоны |
| 3 | LastName | NVARCHAR(50) | Фамилия |
| 4 | Name | NVARCHAR(50) | Имя |
| 5 | MiddleName | NVARCHAR(50) | Отчество |
| 6 | Phone | NVARCHAR(50) | Номер телефона |

Таблица 2 – Type person

|  |  |  |  |
| --- | --- | --- | --- |
| №  п-п | Название поля | Тип данных | Описание |
| 1 | Id | INT | Идентификатор типа персоны |
| 2 | Name | NVARCHAR(50) | Название |

Таблица 3 - Representative

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Название поля | Тип данных | Описание |
| 1 | Id | INT | Идентификатор представителя |
| 2 | TypeId | INT | Идентификатор типа представителя |
| 3 | WorkOfPlace | NVARCHAR(50) | Место работы |
| 4 | Residence | NVARCHAR(50) | Место жительства |
| 5 | HomePhone | NVARCHAR(50) | Домашний телефон |
| 6 | StudentId | INT | Идентификатор студента |

Таблица 4 – Relative

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Название поля | Тип данных | Описание |
| 1 | Id | INT | Идентификатор родственника |
| 2 | Degree | NVARCHAR(50) | Степень родства |
| 3 | Address | NVARCHAR(50) | Адрес |
| 4 | StudentId | INT | Идентификатор студента |

Таблица 5 – Passport

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Название поля | Тип данных | Описание |
| 1 | Id | INT (IDENTITY) | Идентификатор паспорта |
| 2 | PersonId | INT | Идентификатор персоны |
| 3 | Series | NVARCHAR(50) | Серия |
| 4 | Number | INT | Номер |
| 5 | WhoGave | NVARCHAR(50) | Кем выдан |
| 6 | DateOfIssue | DATE | Дата выдачи |
| 7 | Address | NVARCHAR(MAX) | Адрес |
| 8 | DivisionCode | NVARCHAR(50) | Код подразделения |

Таблица 6 – Type representative

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Название поля | Тип данных | Описание |
| 1 | Id | INT (IDENTITY) | Идентификатор  типа представителя |
| 2 | Name | NVARCHAR(50) | Название |

Таблица 7 – Student

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Название поля | Тип данных | Описание |
| 1 | Id | INT | Идентификатор  студента |
| 2 | DateOfBirth | DATE | Дата рождения |
| 3 | Group | NVARCHAR(50) | Номер группы |
| 4 | Photo | IMG | Фото студента |

Таблица 8 – Contract

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Название поля | Тип данных | Описание |
| 1 | Id | INT (IDENTITY) | Идентификатор договора |
| 2 | DateStart | DATE | Дата заселения |
| 3 | DateEnd | DATE | Дата выселения |
| 4 | RoomId | INT | Идентификатор комнаты |
| 5 | StudentId | INT | Идентификатор студента |

Таблица 9 – Type user

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Название поля | Тип данных | Описание |
| 1 | Id | INT (IDENTITY) | Идентификатор типа пользователя |
| 2 | Name | NVARCHAR(50) | Название |

Таблица 10 – User

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Название поля | Тип данных | Описание |
| 1 | Id | INT | Идентификатор пользователя |
| 2 | Login | NVARCHAR(50) | Логин |
| 3 | Password | NVARCHAR(50) | Пароль |
| 4 | TypeId | INT | Идентификатор типа пользователя |

Таблица 11 – Room

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Название поля | Тип данных | Описание |
| 1 | Id | INT (IDENTITY) | Идентификатор комнаты |
| 2 | Number | DOUBLE | Номер |
| 3 | TechPasport | NVARCHAR(50) | Тех. паспорт |
| 4 | Area | DOUBLE | Площадь |
| 5 | NumberOfPlace | INT | Количество мест |

1.3 Описание языка программирования и СУБД

Основным языком программирования для дипломного проекта, был выбран C# (Си Шарп).

C# – объектно-ориентированный язык программирования. Разработан в 1998 – 2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников — языков C++, Pascal, Модула, Smalltalk и, в особенности, Java – С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ и некоторых других языков, не поддерживает множественное наследование классов (между тем допускается множественное наследование интерфейсов).

C# разрабатывался как язык программирования прикладного уровня для CLR (исполняющая среда для байт-кода CIL, в которой компилируются программы) и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает BCL (стандартная библиотека классов платформы NET. Framework). Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем (однако, эта закономерность была нарушена с выходом C# 3.0, представляющего собой расширения языка, не опирающиеся на расширения платформы .NET). CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, сборка мусора не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др.

Преимущества:

* поддержка майкрософтом. В отличии от Java, которой не пошел на пользу переход в собственность Oracle, C# хорошо развивается благодаря усилиям Microsoft;
* в последнее время много совершенствуется. Так как C# был создан позже, чем Java и другие языки, то требовалось очень много доработать. Также это касается популяризации и бесплатности - было обещано открыть исходный код, а инструменты (Visual Studio, Xamarin) стали бесплатными для частных лиц и небольших компаний;
* много синтаксического сахара;
* средний порог вхождения. Синтаксис похожий на C, С++ или Java облегчает переход для других программистов. Для новичков это также один из самых перспективных языков для изучения;
* Xamarin. Благодаря покупке Xamarin на C# теперь можно писать под Android и iOS. Это, несомненно, большой плюс, так как их собственная мобильная ОС (Windows Phone) не завоевала большой популярности;
* добавлено функциональное программирование (F#);
* большое сообщество программистов;
* много вакансий на должность C# программиста в любом регионе.

Недостатки:

* ориентированность, в основном, только на .NET (на Windows платформу);
* бесплатность только для небольших компании, учащихся и программистов-одиночек. Для больших команд покупка лицензий обойдется недешево. Поэтому если у вас есть своя фирма, то придется раскошелиться;
* сохранили оператор go to.

Реализация проекта осуществлялась в интегрированной среде разработки Microsoft Visual Studio 2019.

Microsoft Visual Studio обладает огромным набором средств и возможностей: позволяет разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения и веб-службы для всех поддерживаемых платформ: Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone, .NET Compact Framework и Silverlight, Android, iOS. Простейшего рефакторинг кода.

Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня.

Язык имеет простой синтаксис, а также удобный, замечательный, простой и надежный Entity Framework, который обеспечивает взаимодействие с базой данных. Также использование этого фреймворка значительно ускоряет процесс разработки приложения.

А использование Material Dising for Wpf обеспечит приложение красивым видом и анимацией за минимум кода. Также при использовании с# мы получаем надежный код и простую отладку, что является очередным плюсом.

Для взаимодействия и хранения данных использовалась система управления базами данных (СУБД) MS SQL Serser.

Microsoft SQL Server — система управления реляционными базами данных (РСУБД). Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями.

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире. Данная СУБД подходит для самых различных проектов: от небольших приложений до больших высоконагруженных проектов.

SQL Server был создан компанией Microsoft. Первая версия вышла в 1987 году. А текущей версией является версия 16, которая вышла в 2016 году, и которая актуальна и сейчас.

SQL Server долгое время был исключительно системой управления базами данных для Windows, однако начиная с версии 16 эта система доступна и на Linux.

SQL Server характеризуется такими особенностями как:

* Производительность. SQL Server работает очень быстро;
* Надежность и безопасность. SQL Server предоставляет шифрование данных;
* Простота. С данной СУБД относительно легко работать и вести администрирование.

Чтобы осуществлять связь между базой данных и приложением на C# необходим посредник. И именно таким посредником является технология Entity Framework, о которой было сказано выше.

Entity Framework представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает Entity Framework, мы уже работает с объектами.

Первая версия Entity Framework – 1.0 вышла еще в 2008 году и представляла очень ограниченную функциональность, базовую поддержку ORM (object-relational mapping – отображения данных на реальные объекты) и один единственный подход к взаимодействию с бд – Database First. С выходом версии 4.0 в 2010 году многое изменилось – с этого времени Entity Framework стал рекомендуемой технологией для доступа к данным, а в сам фреймворк были введены новые возможности взаимодействия с бд – подходы Model First и Code First.

Отличительной чертой Entity Framework является использование запросов LINQ для выборки данных из БД. С помощью LINQ мы можем не только извлекать определенные строки, хранящие объекты, из базы данных, но и получать объекты, связанные различными ассоциативными связями.

Для экспорта данных в Word документы и Excel таблиц мы использовали специализированные библиотеки, загруженные из Microsoft Visual Studio.

2 СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1 Эскизный проект

В данной программе будут использоваться функции:

* Функция блокировки кнопки входа на 1 минуту после трех неудачных попыток авторизации с последующим увеличением времени на 1 минуту после каждой следующей попытки;
* Функция авторизации пользователя;
* Функция добавления студента (имя, фамилия, отчество, телефон, дата рождения, группа, и возможность загрузить фото);
* Функция добавления паспорта студента (серия, номер, кем выдан, дата выдачи, адрес прописки, код подразделения);
* Функция добавления представителей студента (имя, фамилия, отчество, телефон, кем приходится, место работы, место жительства, домашний телефон);
* Функция добавления паспорта представителя (серия, номер, кем выдан, дата выдачи, адрес прописки, код подразделения);
* Функция добавления списка родственников студента (имя, фамилия, отчество, телефон, степень родства, адрес);
* Функция добавления договора;
* Функция вывода списка всех зарегистрированных студентов;
* Функция мягкого поиска студента в списке студентов;
* Функция удаления студента;
* Функция редактирования данных студента
* Функция редактирования паспортных данных студента;
* Функция редактирования данных представителя студента;
* Функция редактирования паспортных данных представителя студента;
* Функция редактирования списка родственников студента;
* Функция редактирования договора студента;
* Функция удаления родственника у студента;
* Функция вывода списка родственников каждого студента;
* Функция мягкого поиска нужного списка родственников;
* Функция вывода списка всех договоров;
* Функция мягкого поиска нужного договора в списке договоров;
* Функция удаления договора студента;
* Функция вывода списка всех комнат;
* Функция удаления комнаты;
* Функция добавления комнаты;
* Функция формирования документа «Договор о найме жилого помещения в специализированном жилом фонде»;
* Функция формирования документа «Дополнительное соглашение к договору о найме жилого помещения в специализированной жилом фонде»;
* Функция формирования документа «Анкета студента»;
* Функция изменения данных текущего профиля;
* Функция вывода списка всех пользователей программы;
* Функция изменения логина и/или пароля для выбранного пользователя;
* Функция добавления нового пользователя;
* Функция сортировки таблицы;

Функция блокировки кнопки входа.

Входные данные: нет.

Выходные данные: заблокированная кнопка входа на определенное время.

Описание функции программы: программа блокирует кнопку входа на 1 минуту после трех неудачных попыток авторизации с последующим увеличением времени на 1 минуту после каждой следующей неудачной попытки.

Функция авторизации пользователя.

Входные данные: введенные в поля логин и пароль.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает введенные логин и пароль с полей, проверяет их на корректность, а также сверяет с хранящимися в базе данных, после чего переходит в другое окно.

Функция добавления студента.

Входные данные: введенные в поля данные студента.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает введенные данные с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция добавления паспорта студента.

Входные данные: введенные в поля данные паспорта студента.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает введенные данные с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция добавления представителя.

Входные данные: введенные в поля данные представителя.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает введенные данные с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция добавления паспорта представителя.

Входные данные: введенные в поля данные паспорта представителя.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает введенные данные с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция добавления списка родственников студента.

Входные данные: список родственников с заполненными данными.

Выходные данные: переход в другое окно.

Описание функции программы: программа проверяет список родственников на наличие ошибок и сохраняет его в базу данных при успешной проверке, после чего переходит в другое окно

Функция добавления договора.

Входные данные: введенные в поля данные договора студента.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает введенные данные с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция вывода списка всех зарегистрированных студентов.

Входные данные: нет.

Выходные данные: список всех зарегистрированных студентов.

Описание функции программы: программа выводит в таблицу всех хранящихся в базе данных студентов.

Функция мягкого поиска студента в списке студентов.

Входные данные: введенная в поле фамилия студента.

Выходные данные: отфильтрованный список студентов.

Описание функции программы: программа выполняет мягкий поиск без учета регистра букв среди всех студентов, сравнивая введенное значение с их фамилиями и выводит в таблицу список с совпадениями.

Функция удаления студента.

Входные данные: выбранный в таблице студент.

Выходные данные: обновленная таблица студентов.

Описание функции программы: программа удаляет из базы данных выбранного студента, со всеми прикрепленными к нему данными (родственники, представители, договора и др.) и затем обновляет список студентов в таблице.

Функция редактирования данных студента.

Входные данные: измененные данные студента с полей.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает измененные данные с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция редактирования паспортных данных студента.

Входные данные: измененные паспортные данные студента с полей.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает измененные паспортные данные с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция редактирования данных представителя студента.

Входные данные: измененные данные представителя студента с полей.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает измененные данные представителя с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция редактирования паспортных данных представителя студента.

Входные данные: измененные паспортные данные представителя студента с полей.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает измененные паспортные данные представителя с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция редактирования списка родственников студента.

Входные данные: измененный список родственников студента.

Выходные данные: переход в другое окно.

Описание функции программы: программа проверяет измененный список родственников на наличие ошибок и сохраняет в базу данных при успешной проверке, после чего переходит в другое окно.

Функция редактирования договора студента.

Входные данные: измененные данные договора студента с полей.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает измененные данные договора с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция удаления родственника у студента.

Входные данные: выбранный в таблице родственник.

Выходные данные: обновленная таблица родственников.

Описание функции программы: программа удаляет из базы данных выбранного родственника и затем обновляет список имеющихся родственников в таблице.

Функция вывода списка родственников каждого студента.

Входные данные: нет.

Выходные данные: список родственников по каждому студенту.

Описание функции программы: программа выводит в таблицу все хранящиеся в базе данных списки родственников по каждому студенту отдельно.

Функция мягкого поиска нужного списка родственников.

Входные данные: введенная в поле фамилия студента.

Выходные данные: отфильтрованная таблица со списками родственников.

Описание функции программы: программа выполняет мягкий поиск без учета регистра букв среди всех имеющихся студентов, сравнивая введенное значение с их фамилиями и выводит в таблицу полученных студентов со списками их родственников.

Функция вывода списка всех договоров.

Входные данные: нет.

Выходные данные: список всех имеющихся договоров.

Описание функции программы: программа выводит в таблицу все хранящиеся в базе данных договора.

Функция мягкого поиска нужного договора в списке договоров.

Входные данные: введенная в поле фамилия студента.

Выходные данные: отфильтрованный список договоров.

Описание функции программы: программа выполняет мягкий поиск без учета регистра букв среди всех студентов, сравнивая введенное значение с их фамилиями и выводит в таблицу список договоров с совпадениями.

Функция удаления договора студента.

Входные данные: выбранный в таблице договор.

Выходные данные: обновленная таблица договоров.

Описание функции программы: программа удаляет из базы данных выбранный договор студента и затем обновляет список имеющихся договоров в таблице.

Функция вывода списка всех комнат.

Входные данные: нет.

Выходные данные: список всех имеющихся комнат.

Описание функции программы: программа выводит в таблицу все хранящиеся в базе данных комнаты общежития и количество свободных мест в каждой на данный момент.

Функция удаления комнаты.

Входные данные: выбранный в таблице комната.

Выходные данные: обновленная таблица комнат.

Описание функции программы: программа удаляет из базы данных выбранную комнату и затем обновляет список имеющихся комнат в таблице.

Функция добавления комнаты.

Входные данные: введенные в поля данные комнаты.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает введенные данные с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция формирования документа «Договор о найме жилого помещения в специализированном жилом фонде».

Входные данные: выбранный в таблице студент, заданный в диалоговом окне путь к папке.

Выходные данные: документ в формате Word.

Описание функции программы: программа экспортирует данные выбранного студента в документ в формате Word по заранее подготовленному образцу и сохраняет его в выбранной пользователем папке.

Функция формирования документа «Дополнительное соглашение к договору о найме жилого помещения в специализированной жилом фонде».

Входные данные: выбранный в таблице студент, заданный в диалоговом окне путь к папке.

Выходные данные: документ в формате Word.

Описание функции программы: программа экспортирует данные выбранного студента в документ в формате Word по заранее подготовленному образцу и сохраняет его в выбранной пользователем папке.

Функция формирования документа «Анкета студента».

Входные данные: выбранный в таблице студент, заданный в диалоговом окне путь к папке.

Выходные данные: документ в формате Excel.

Описание функции программы: программа экспортирует данные выбранного студента в документ в формате Excel по заранее подготовленному образцу и сохраняет его в выбранной пользователем папке.

Функция изменения данных текущего профиля.

Входные данные: измененные данные текущего профиля с полей.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает введенные данные с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция вывода списка всех пользователей программы.

Входные данные: нет.

Выходные данные: список всех пользователей программы.

Описание функции программы: программа выводит в таблицу всех хранящихся в базе данных пользователей программы.

Функция изменения логина и/или пароля для выбранного пользователя.

Входные данные: измененный логин и/или пароль с полей.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает введенные логин и/или пароль с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция добавления нового пользователя.

Входные данные: введенные в поля данные пользователя.

Выходные данные: переход в другое окно.

Описание функции программы: программа считывает введенные данные пользователя с полей, проверяет их на корректность и сохраняет в базу данных, после чего переходит в другое окно.

Функция сортировки таблицы.

Входные данные: выбранный столбец таблицы.

Выходные данные: отсортированная таблица.

Описание функции программы: программа сортирует таблицу по выбранному столбцу и возвращает ее пользователю. Эта функция работает для всех имеющихся таблиц.

2.2 Технический проект

Функция блокировки кнопки входа.

Начало

Нет

Да

Конец

Рассчитать время и заблокировать кнопку

Попыток входа>3?

Функция авторизации пользователя.

Начало

Нет

Нет

Да

Да

Конец

Данные админа?

Данные юзера?

Вывод окна юзера

Вывод окна админа

Ввод логин и пароля

Функция добавления студента.

Начало

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Сохранение данных в бд

Ввод данных студента

Функция добавления паспорта студента.

Начало

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Сохранение данных в бд

Ввод данных паспорта студента

Функция добавления представителя.

Начало

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Сохранение данных в бд

Ввод данных представителя

Функция добавления паспорта представителя.

Начало

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Сохранение данных в бд

Ввод данных паспорта представителя

Функция добавления списка родственников студента.

Начало

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Сохранение данных в бд

Ввод списка родственников

Функция добавления договора.

Начало

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Сохранение данных в бд

Ввод данных договора

Функция вывода списка всех зарегистрированных студентов.

Начало

Конец

Вывод в таблицу списка

Ошибок нет?

Запрос списка студентов в бд

Нет

Да

Функция мягкого поиска студента в списке студентов.

Начало

Нет

Да

Ошибок нет?

Вывод в таблицу результата поиска

Поиск в списке совпадений

Конец

Ввод фамилии студента

Функция удаления студента.

Начало

Вывод обновленного списка

Конец

Нет

Да

Ошибок нет?

Удаление студента из бд и связанных с ним данных

Выбор студента в списке

Функция редактирования данных студента.

Начало

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Изменение данных в бд

Ввод измененных данных студента

Функция редактирования паспортных данных студента.

Начало

Ввод

изменённых паспортных данных студента

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Изменение данных в бд

Функция редактирования данных представителя студента.

Начало

Ввод

изменённых данных представителя

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Изменение данных в бд

Функция редактирования паспортных данных представителя студента.

Начало

Ввод измененных паспортных данных

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Изменение данных в бд

Функция редактирования списка родственников студента.

Начало

Ввод измененного списка родственников

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Изменение данных в бд

Функция редактирования данных договора студента.

Начало

Ввод

изменённых данных договора

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Изменение данных в бд

Функция удаления родственника у студента.

Начало

Вывод обновленного списка

Конец

Нет

Да

Ошибок нет?

Удаление родственника из бд

Выбор родственника в списке

Функция вывода списка родственников каждого студента.

Начало

Конец

Вывод в таблицу списков

Ошибок нет?

Запрос списков родственников в бд

Нет

Да

Функция мягкого поиска нужного списка родственников.

Начало

Нет

Да

Ошибок нет?

Вывод в таблицу результата поиска

Поиск в списке совпадений

Конец

Ввод фамилии студента

Функция вывода списка всех договоров.

Начало

Конец

Вывод в таблицу списка

Ошибок нет?

Запрос списка договоров в бд

Нет

Да

Функция мягкого поиска нужного договора в списке договоров.

Начало

Нет

Да

Ошибок нет?

Вывод в таблицу результата поиска

Поиск в списке совпадений

Конец

Ввод фамилии студента

Функция удаления договора студента.

Начало

Вывод обновленного списка

Конец

Нет

Да

Ошибок нет?

Удаление договора из бд

Выбор договора в списке

Функция вывода списка всех комнат.

Начало

Конец

Вывод в таблицу списка

Ошибок нет?

Запрос списка комнат в бд

Нет

Да

Функция удаления комнаты.

Начало

Вывод обновленного списка

Конец

Нет

Да

Ошибок нет?

Удаление комнаты из бд

Выбор комнаты в списке

Функция добавления комнаты.

Начало

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Сохранение данных в бд

Ввод данных комнаты

Функция формирования документа «Договор о найме жилого помещения в специализированном жилом фонде».

Начало

Нет

Да

Ошибок нет?

Формирование Word документа и сохранение в папке

Выбор папки для сохранения

Конец

Выбор студента в списке

Функция формирования документа «Дополнительное соглашение к договору о найме жилого помещения в специализированной жилом фонде».

Начало

Нет

Да

Ошибок нет?

Формирование Word документа и сохранение в папке

Выбор папки для сохранения

Конец

Выбор студента в списке

Функция формирования документа «Анкета студента».

Начало

Нет

Да

Ошибок нет?

Формирование Excel документа и сохранение в папке

Выбор папки для сохранения

Конец

Выбор студента в списке

Функция редактирования данных текущего профиля.

Начало

Ввод

изменённых данных профиля

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Изменение данных в бд

Функция вывода списка всех пользователей.

Начало

Конец

Вывод в таблицу списка

Ошибок нет?

Запрос списка пользователей в бд

Нет

Да

Функция изменения логина и/или пароля для выбранного пользователя.

Начало

Ввод

изменённых логина и/или пароля.

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Изменение данных в бд

Функция добавления нового пользователя.

Начало

Ошибок нет?

Конец

Нет

Да

Переход в другое окно

Сохранение данных в бд

Ввод данных пользователя

Функция сортировки таблицы.

Начало

Нет

Да

Ошибок нет?

Сортировка таблицы по выбранному столбцу

Конец

Вывод отсортированной таблицы

Выбор столбца таблицы

2.3 Руководство пользователя

В данном дипломном проекте реализована информационная система «РЕГИСТРАЦИЯ СТУДЕНТОВ».

Для начала работы с программой необходимо запустить файл Diplom.exe, который лежит в корне каталога приложения.

На рисунке 2 представлено окно авторизации, содержащее два поля для ввода логина и пароля, а также кнопку «Войти», которое открывается при запуске программы. Для входа в систему необходимо пройти процедуру авторизации пользователя: нужно ввести данные своего профиля (логин и пароль) в соответствующие поля и нажать кнопку «Войти». Если у вас нет профиля, то необходимо попросить администратора создать вам его. Сам администратор также должен для этого авторизоваться. При первом запуске программы уже есть заранее созданный админ для этих целей с логин и паролем «admin», «admin». Позже его можно будет удалить изменить или удалить и создать другого.

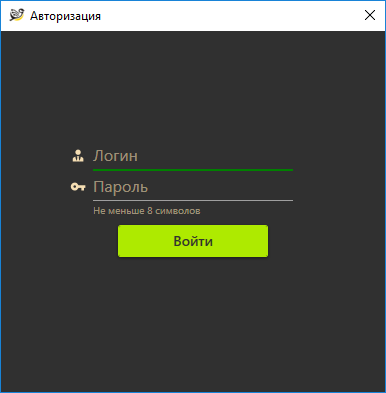


Рисунок 2 – Окно «Авторизация»

При не заполнении одного из полей и попытке авторизоваться появиться ошибка, которая сообщит о том, что не все поля были заполнены (Рисунок 2).

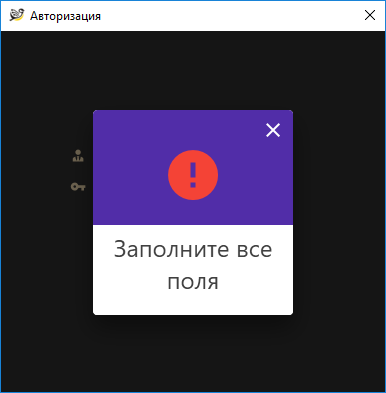


Рисунок 3 – Сообщение о не заполнении всех полей

Если пользователь введет неверный логин или пароль, то приложение выдаст ошибку о неверно заполненных данных (Рисунок 4), а если пользователь введет неверные данные 3 и более раз, то приложение на время заблокирует кнопку входа (Рисунок 5).

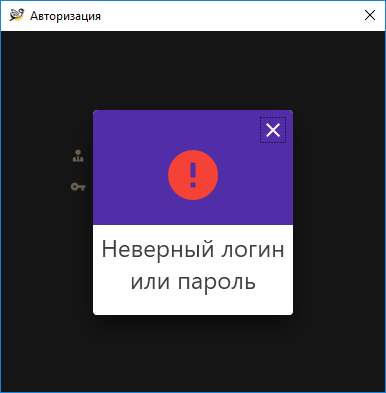


Рисунок 4 – Сообщение о неверных данных

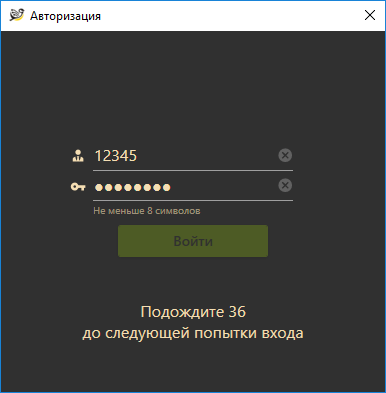


Рисунок 5 – Окно со включенной блокировкой входа

Далее будет рассмотрен функционал обычного пользователя.

После авторизации пользователь попадает в главное окно работника общежития (Рисунок 6), которое содержит меню программы слева, кнопку возврата в верхнем правом углу и область для отображения нужного контента посередине, в которую уже загружена страница под названием «Главная».

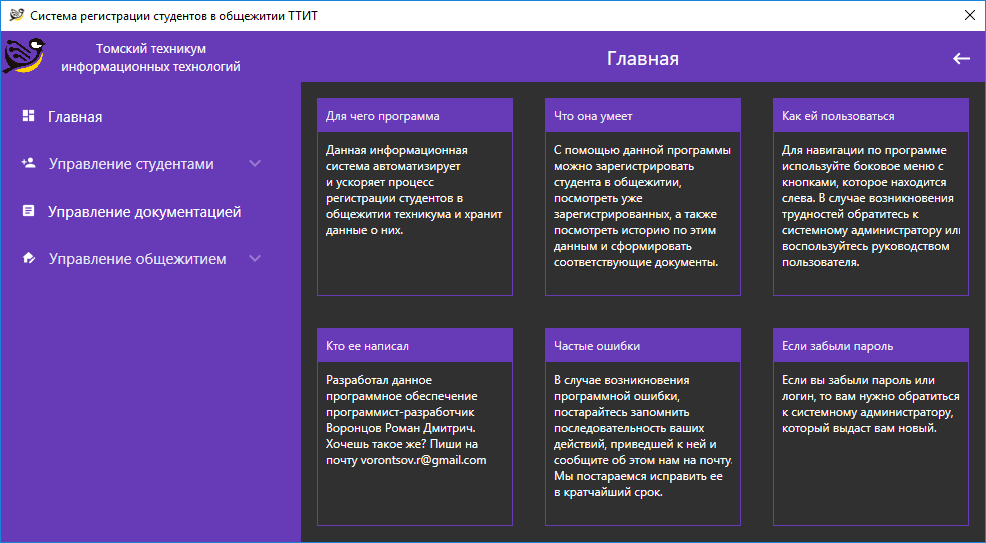


Рисунок 6 – Главное окно программы

Меню программы выполнено в виде заголовков с раскрывающимися списками, в которые помещены подпункты (Рисунок 7). Размещены они в соответствии с частотой использования пользователем. Для перехода требуется кликнуть мышью на требуемый раздел меню.

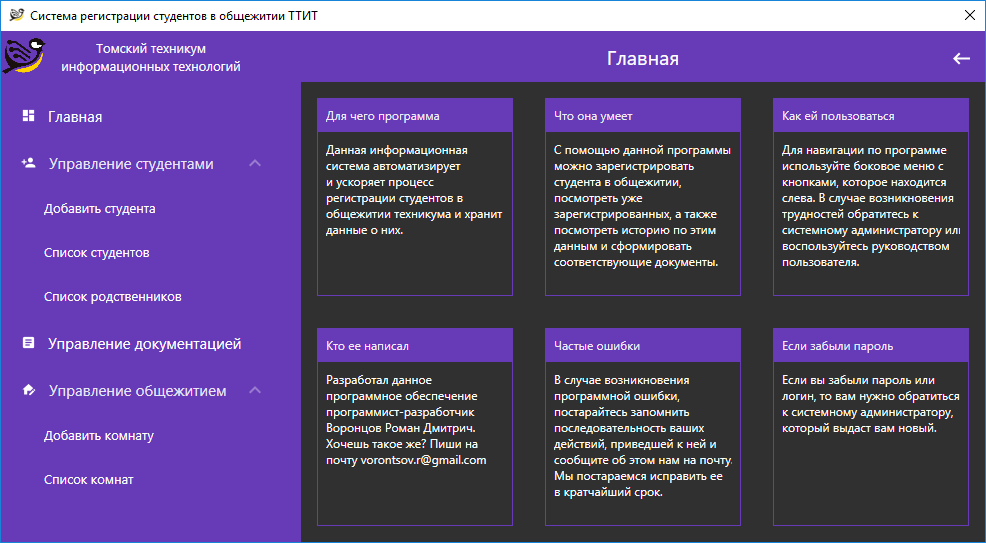


Рисунок 7 – Полностью раскрытое меню программы

При выборе в меню пункта «Добавить студента» в область контента будет загружена форма для заполнения данных студента (Рисунок 8), в которой будут находиться две области с полями: одна для заполнения основных данных о самом студенте, другая – для его паспортных данных, также есть кнопка «Загрузить фото» для загрузки фотографии студента в формате jpg или png и кнопка «Далее» для перехода в следующую форму заполнения данных. Для добавления студента необходимо заполнить все поля без ошибок (Рисунок 9) и нажать кнопку «Далее», загрузка фотографии является не обязательной, и выполнятся по желанию.

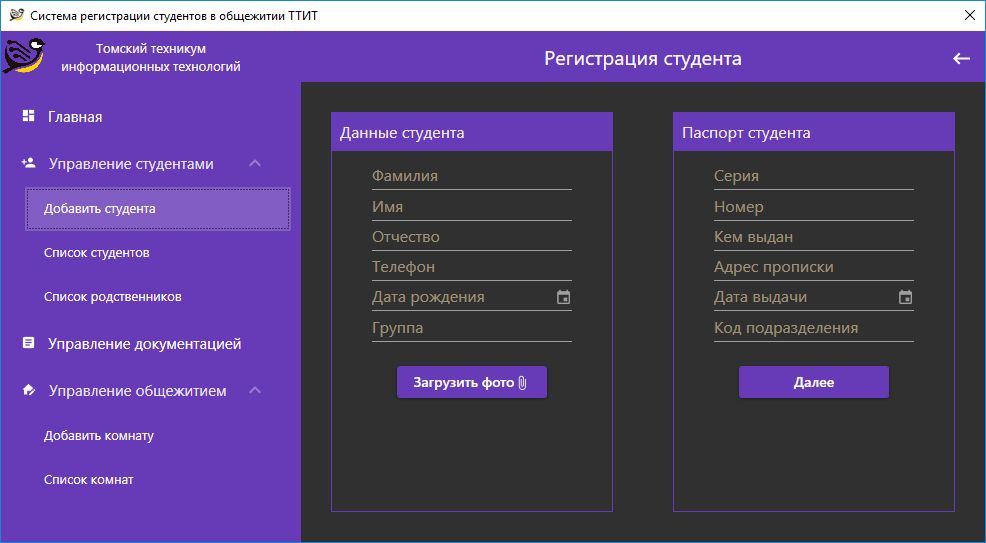


Рисунок 8 – Форма заполнения данных о студенте

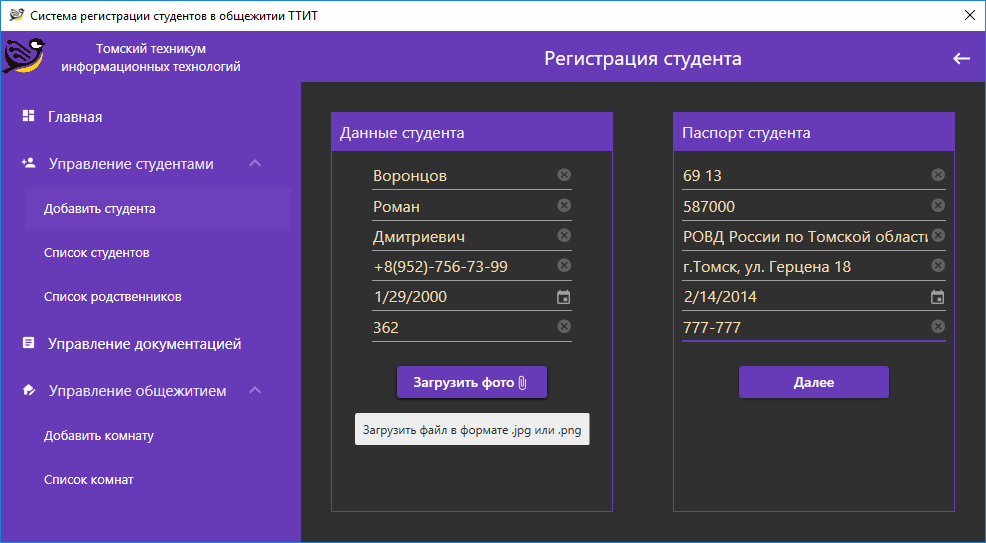


Рисунок 9 – Форма заполнения данных о студенте с заполненными полями

Если пользователь допустил ошибку при заполнении данных, то система автоматически найдет её и сообщит о ней (Рисунки с 10 по 14)

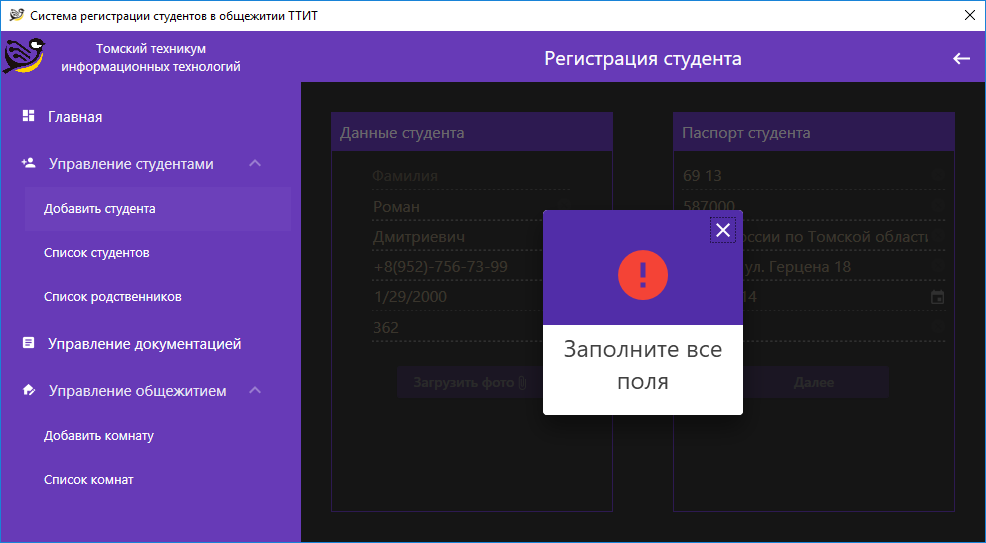


Рисунок 10 – Сообщение о не заполнении всех полей

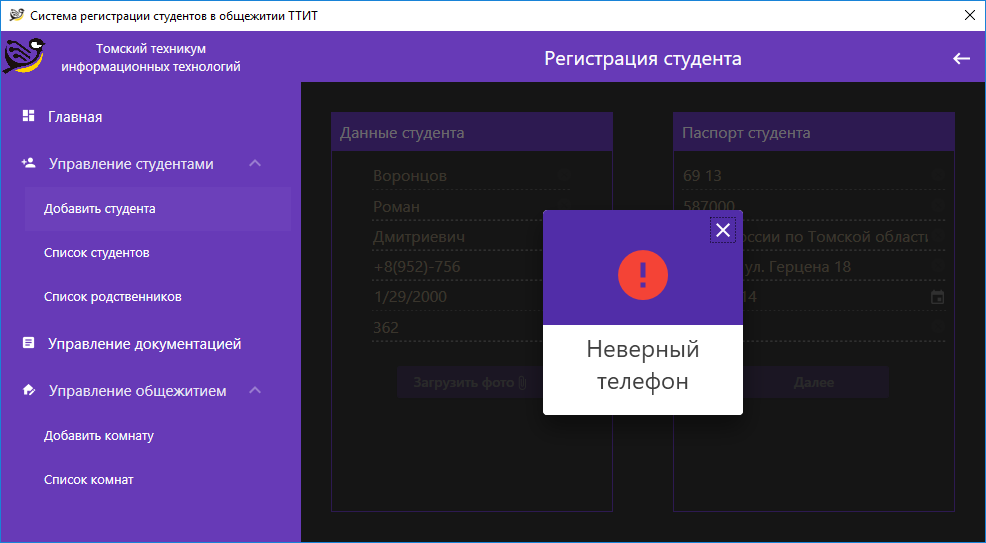


Рисунок 11 – Сообщение о неверном номере телефона

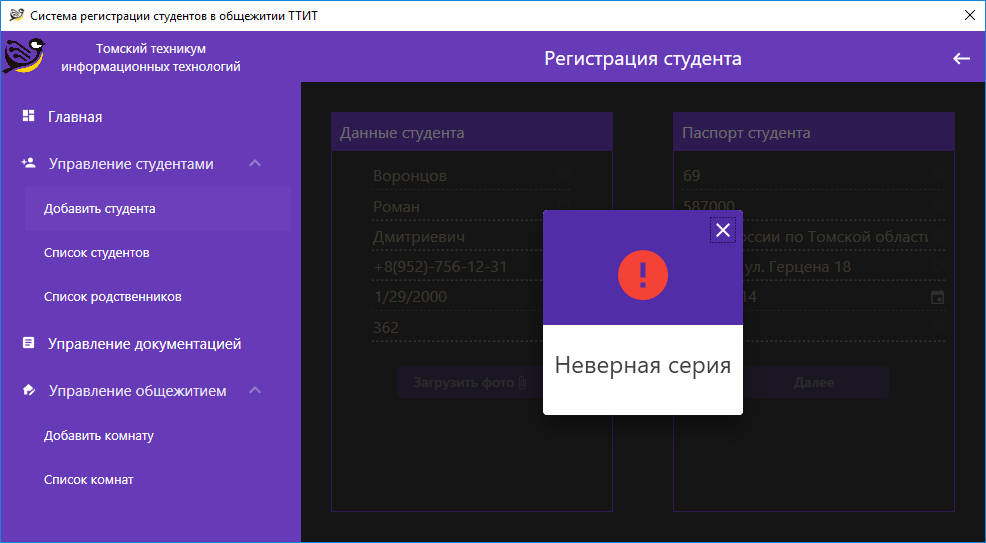


Рисунок 12 – Сообщение о неверной серии паспорта

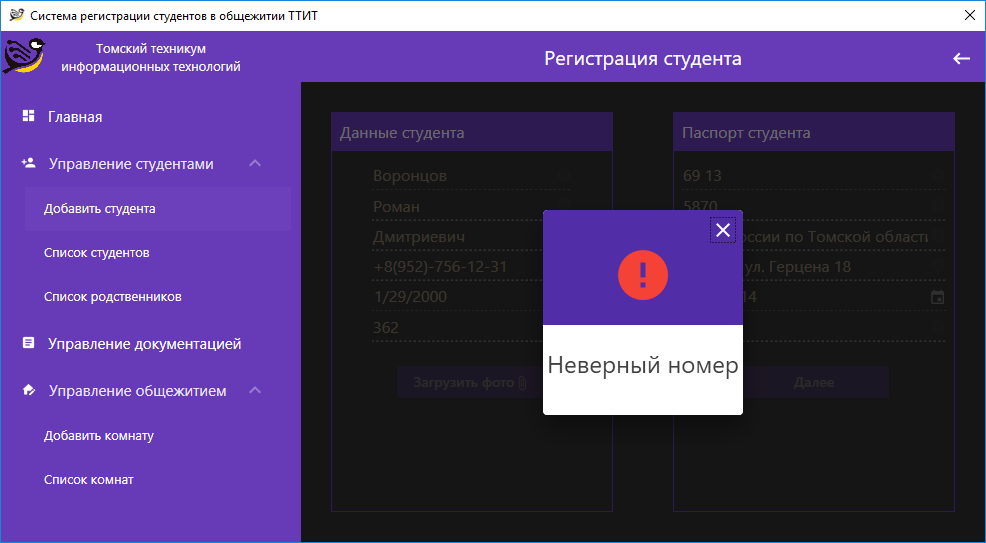


Рисунок 13 – Сообщение о неверном номере паспорта

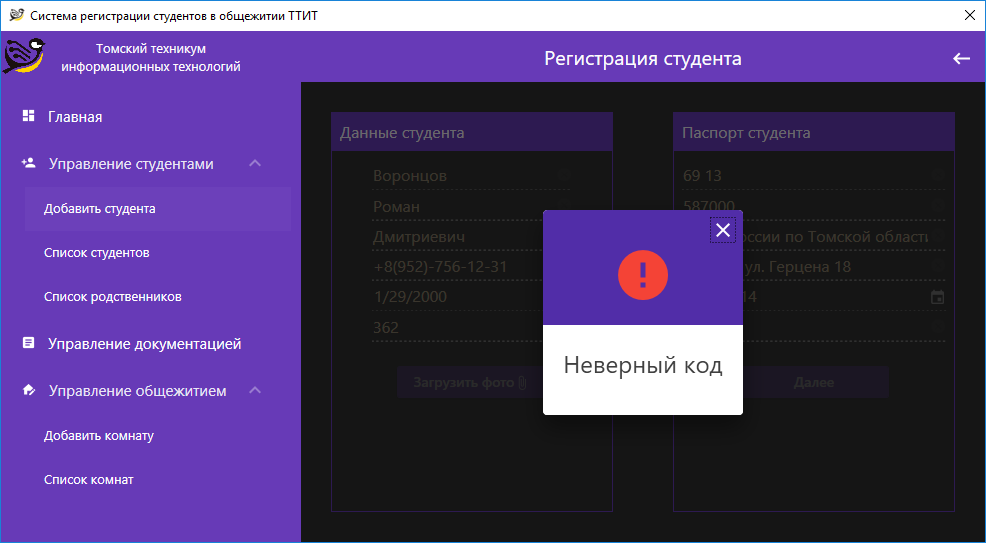


Рисунок 14 – Сообщение о неверном коде подразделения

После нажатия кнопки «Далее», если никаких ошибок нет, то приложение сообщит о сохранении данных и загрузит следующую форму для заполнения данных о представителе студента (Рисунок 15), в которой будут находиться две области с полями: одна для заполнения основных данных о самом представителе, другая – для его паспортных данных, также есть кнопка «Далее» для перехода в следующую форму заполнения данных. Для добавления представителя необходимо заполнить все поля без ошибок и нажать кнопку «Далее» (Рисунок 16).

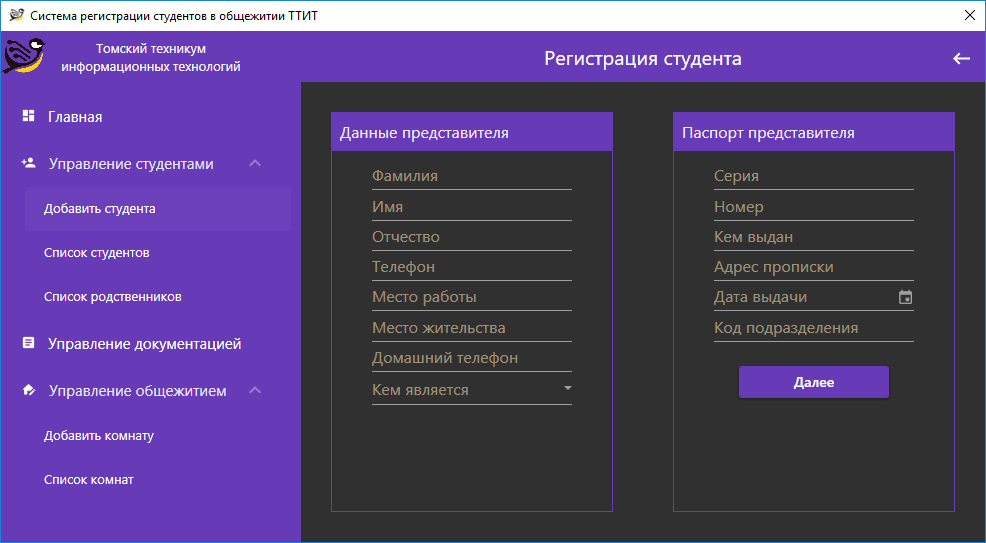


Рисунок 15 – Форма заполнения данных о представителе студента

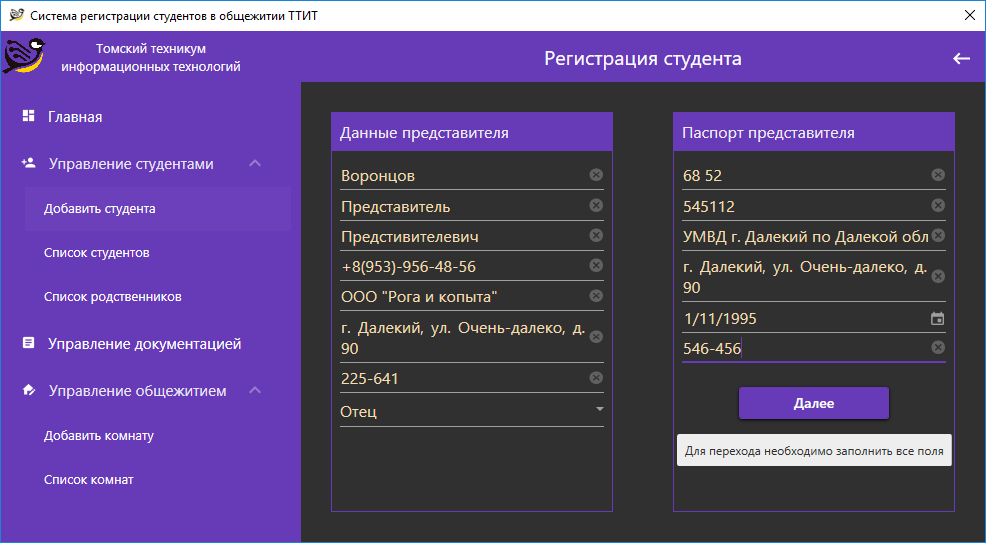


Рисунок 16 – Форма представителя с заполненными полями

Система также, как и в случае заполнением данных студента, проверит введенные данные на наличие ошибок и сообщит о их наличии, если они присутствуют. Если же пользователь не допустил ошибок, то приложение сообщит об успешности сохранения данных и загрузит форму для добавления списка родственников, которая содержит таблицу для отображения добавленных родственников (Рисунок 17) и две кнопки: «+» – для вызова окна добавления (Рисунок 18), и «→» – для перехода в форму заполнения договора. Для того чтобы добавить родственника в список необходимо вызвать окно добавления кнопкой «+», заполнить все данные и нажать «Добавить», либо «Отмена» для закрытия диалогового окна. Далее система проверит введенные данные на наличие ошибок, и если их нет, то внесет родственника в список и закроет диалоговое окно. Если ли же ошибки были найдены, то об этом будет сообщено пользователю, как в случае заполнения данных о студенте. Созданный список можно будет позже отредактировать.

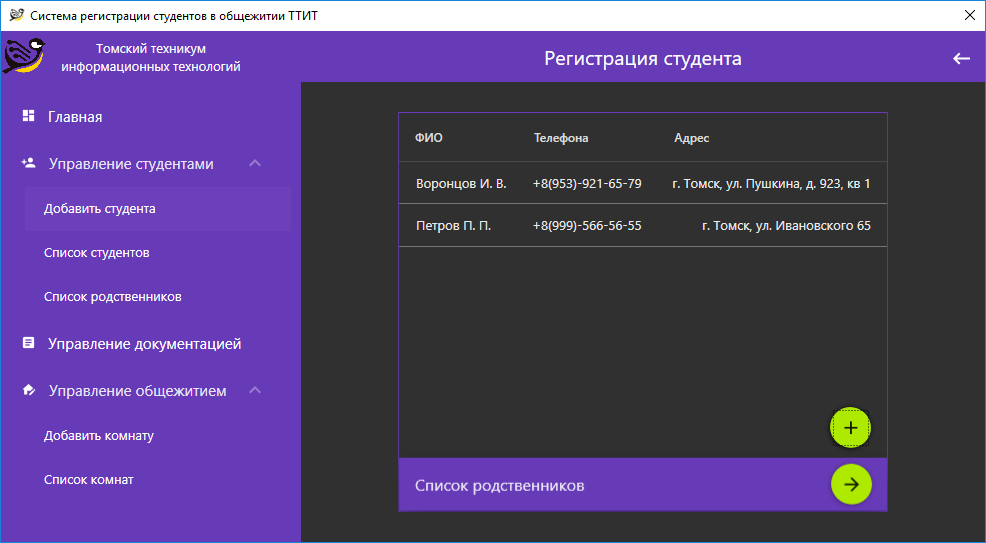


Рисунок 17 – Форма для добавления списка родственников

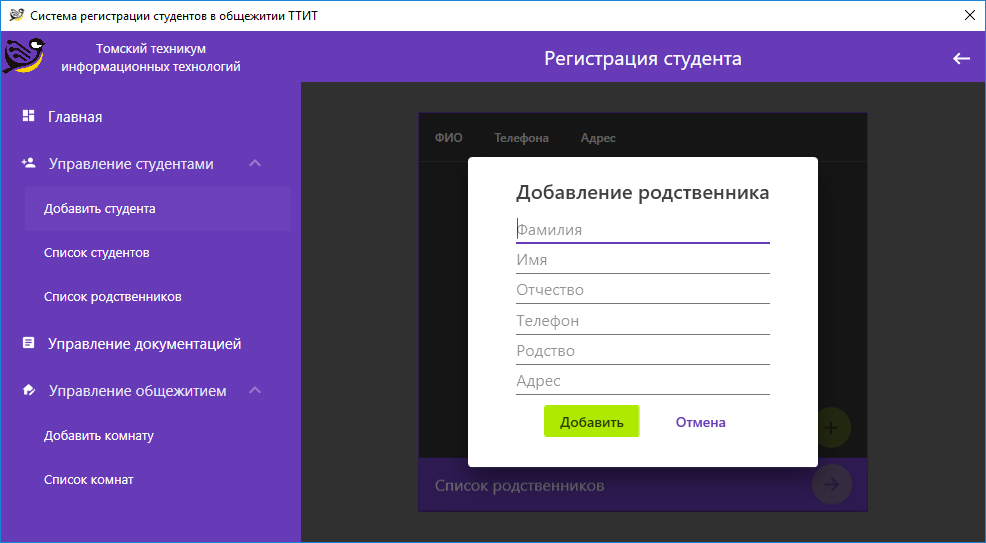


Рисунок 18 – Форма для добавления родственника в список

Как только добавление родственников будет окончено, можно переходить к форме заполнения данных договора (Рисунок 19), на которой находятся два поля для выбора срока, на какой будет заключен договор, и выпадающий список, в котором отображаются комнаты, имеющие на сегодняшний день свободные места для заселения, и кнопка «Заселить», которая сохраняет введенные данные и завершает регистрацию студента.

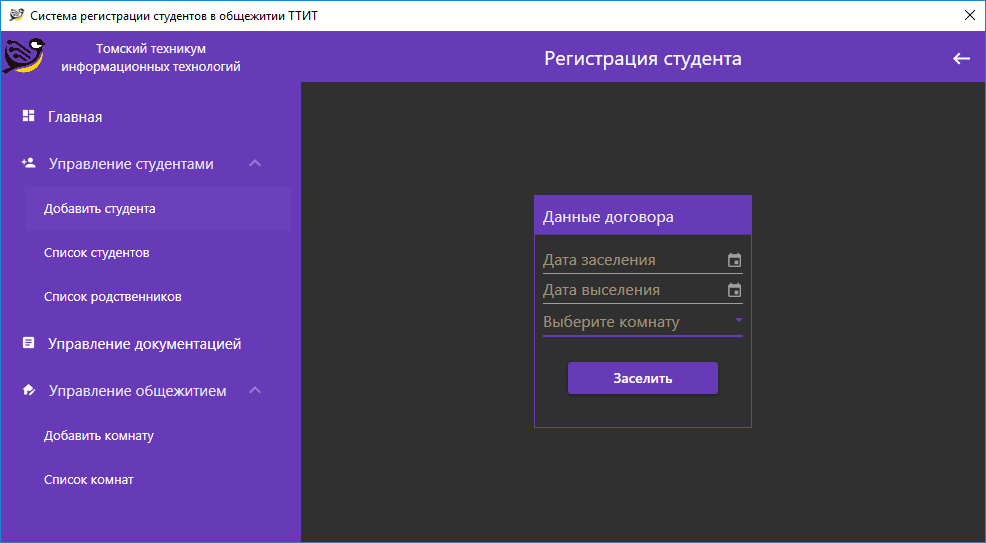


Рисунок 18 – Форма заполнения данных договора

Для добавления договора в систему необходимо заполнить все поля и выбрать из списка доступную комнату, после чего нажать кнопку «Заселить». Далее система проверит данные на наличие ошибок, и при их отсутствии сообщит об успешном сохранении, и затем сразу перейдет на начальный экран заполнения данных о студенте, готовая добавить нового студента. На этом процесс регистрации студента будет завершен. Если же ошибки были найдены, то система сообщит о их наличии пользователю, как в случае заполнения данных о студенте.

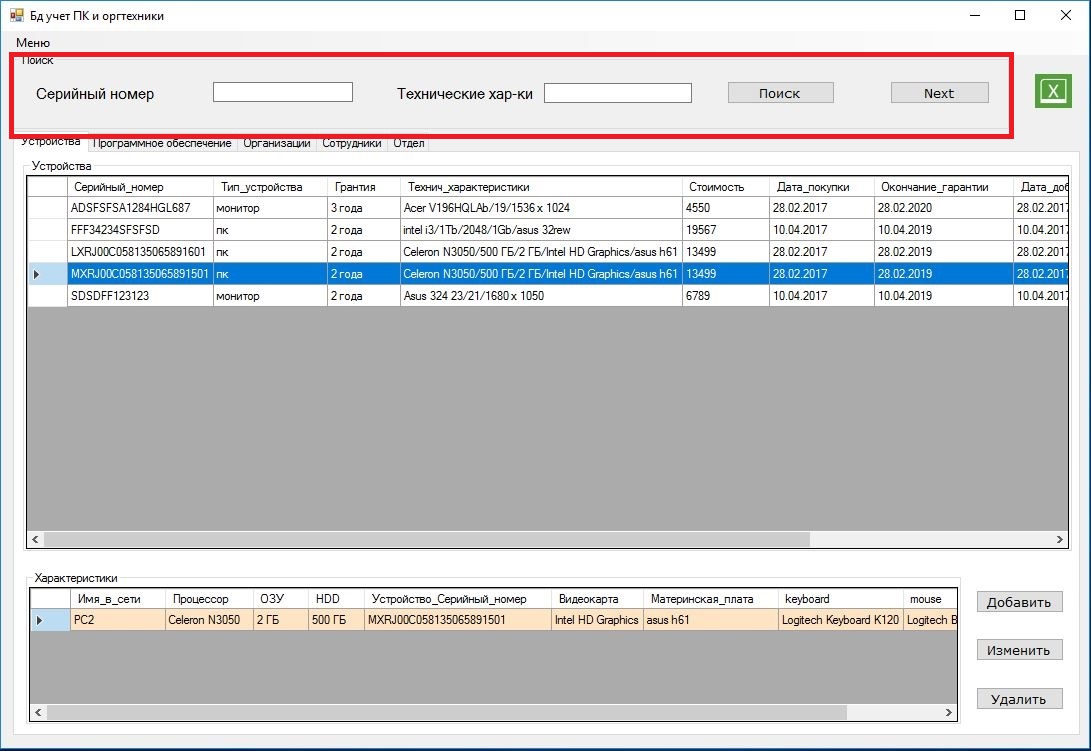


Рисунок 7 – Поиск по главной таблице

Если вы хотите сменить пароль для «Пользователя» или «Администратора», нужно выбрать в программе выпадающий список «Меню» (рисунок 9) и выбрать «сменить пароль» или «выход» для выхода из программы.

Внимание! Смена пароля осуществляется только под Администратором.

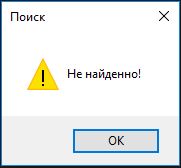


Рисунок 8 – Диалоговое окно сообщающее,  
 что при поиске ничего не найдено

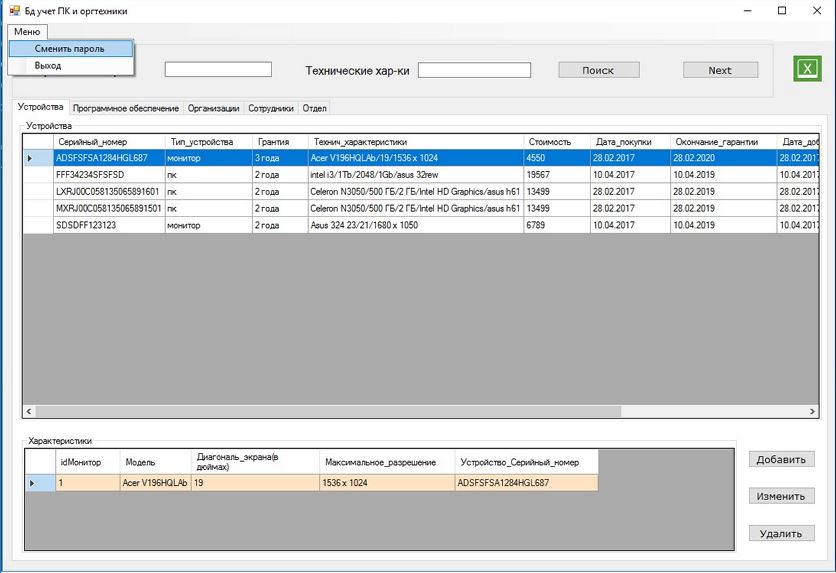


Рисунок 9 – Выпадающее меню

После нажатия «меню» и «сменить пароль» вы падает диалоговое окно «Смена пароля» (рисунок 10) где нужно выбрать пользователя для которого будет производиться смена пароля, и указать в текстовых полях старый пароль текущего выбранного пользователя, и указать новый пароль для пользователя после чего нажмите «OK» для смены пароля или «Отмена» для выхода и закрытия окна «Смена пароля». Если ввести не правильный старый пароль или не указать пользователя появится предупреждение «Введен неверный пароль или не указан пользователь» (рисунок 11).

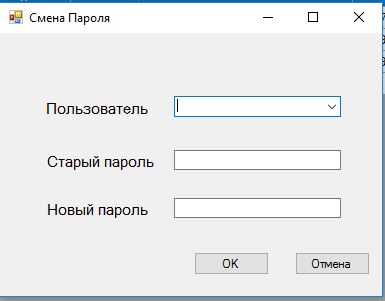


Рисунок 10 – Смена пароля

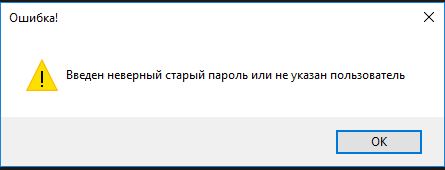


Рисунок 11 – Ошибка при смене пароля

Для добавления, удаления, изменения данных, выберете нужную вкладку в главном окне программы и выберете нужную кнопку (рисунок 12). Внимание! Изменять, добавлять, удалять и менять пароль, выполняется только под Администратором, под Пользователем можно только в режиме просмотра данных.

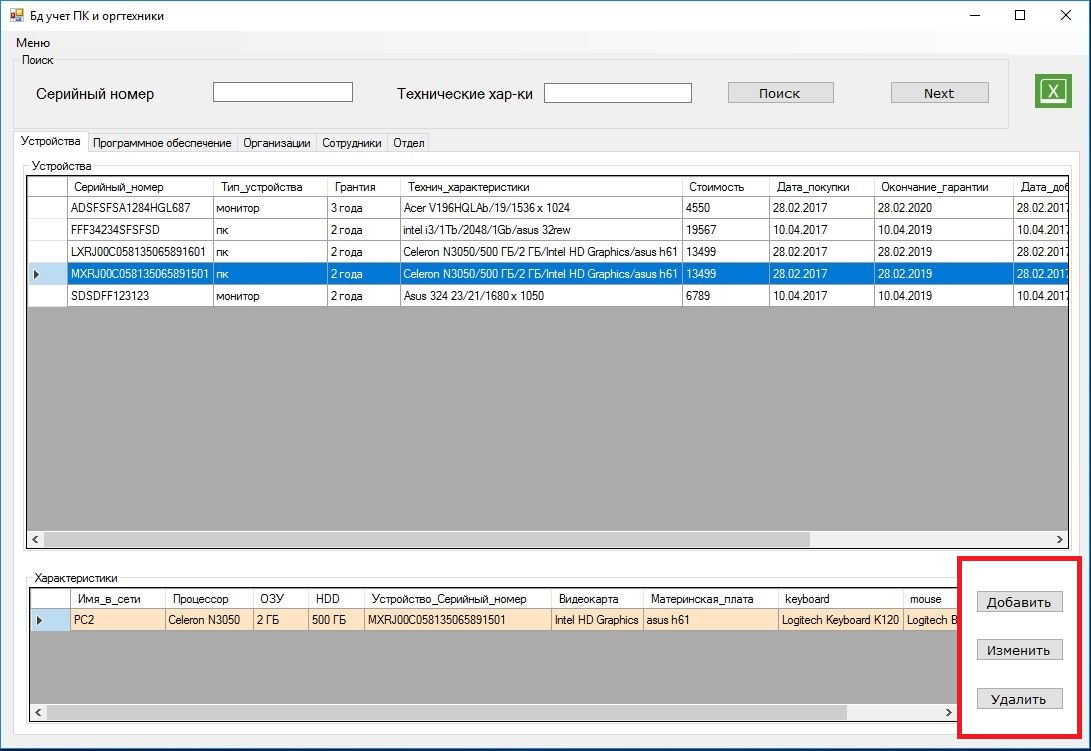


Рисунок 12 – Кнопки для добавления, изменения и удаления данных

Нажмите кнопку «Добавить» что бы добавить запись в таблицу. После нажатия появится форма для заполнения данных (рисунок 13). Каждой вкладке соответствует своя форма для заполнения данных. Если вы заполнили все поля, нажмите кнопку «Добавить» (рисунок 13). Каждое текстовое поле в форме для заполнения данных должно быть заполнено, если текстовое поле не заполнено возникает диалоговое окно с ошибкой (рисунок 14). Если текстовые поля заполнены, после успешного добавления данных в БД возникает диалоговое окно с сообщением «Данные добавлены» (рисунок 15). При этом главная таблица автоматически обновляется.

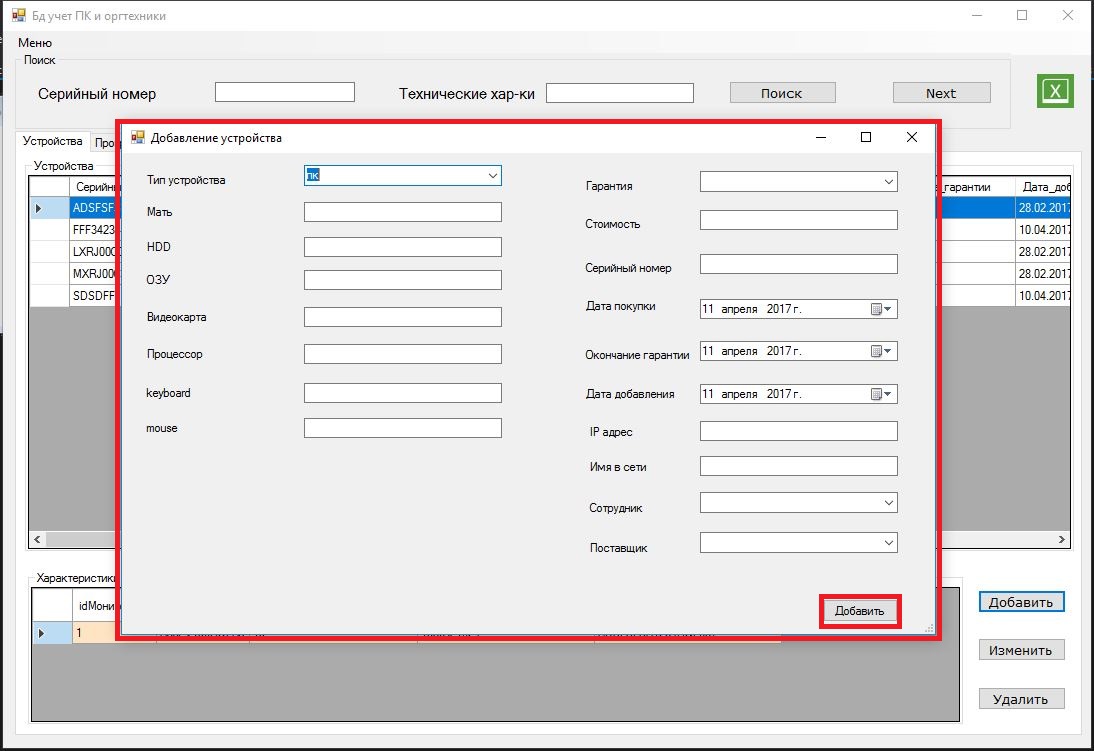


Рисунок 13 - Добавление данных на примере вкладки «Устройства»

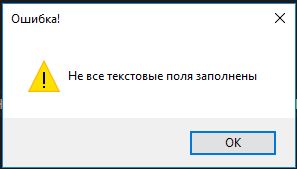


Рисунок 14 – Диалоговое окно при незаполненных полях

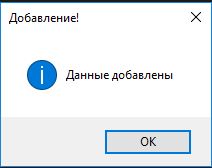


Рисунок 15 – Диалоговое окно при успешном добавлении данных в БД

Что бы изменить запись нужно выбрать нужную вкладку и выделить в главной таблице строку, которую хотите изменить и нажать кнопку «Изменить». После нажатия появится форма для изменения данных, как показано на рисунке 16. Форма автоматически заполняется данными. Текстовые поля в форме не должны быть пустыми иначе появляется диалоговое окно «Ошибка!», как показано на рисунке 14. При успешном изменение выпадает сообщение «данные изменены» (рисунок 17).

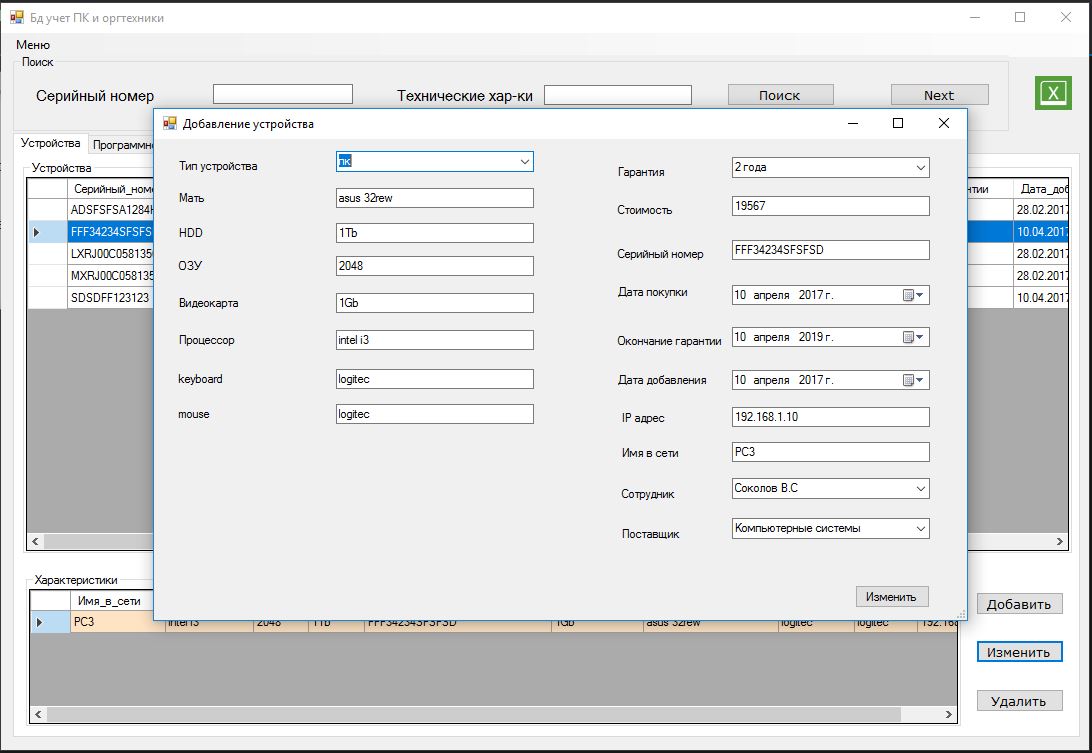


Рисунок 15 – Изменение данных

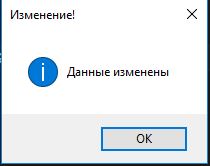


Рисунок 17 – Диалоговое окно при успешном изменение данных в БД

Чтобы удалить запись нужно выбрать нужную вкладку и выделить в главной таблице строку, которую хотите изменить и нажать кнопку «Удалить». После нажатия появится диалоговое окно с вопросом для подтверждения на удаление записи в главной таблице (рисунок 18). При нажатии на кнопку «OK» для подтверждения или «Отмена» для отмены удаления. На рисунке 19 показано диалоговое окно с информацией «Данные удалены».

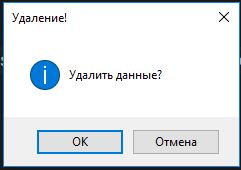


Рисунок 18 – Удаление данных

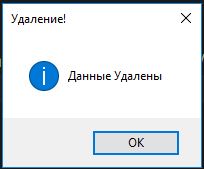


Рисунок 19 – Успешное удаление данных

Для экспорта данных таблицы в MS Excel. Нужно нажать кнопку Excel (рисунок 20). После нажатия появится окно выбора пути для сохранения Excel файла и указания формата(xls, xlsx) и названия файла (рисунок 21). После экспорта можно открыть файл Excel. Экспортные данные показаны на рисунок 22.

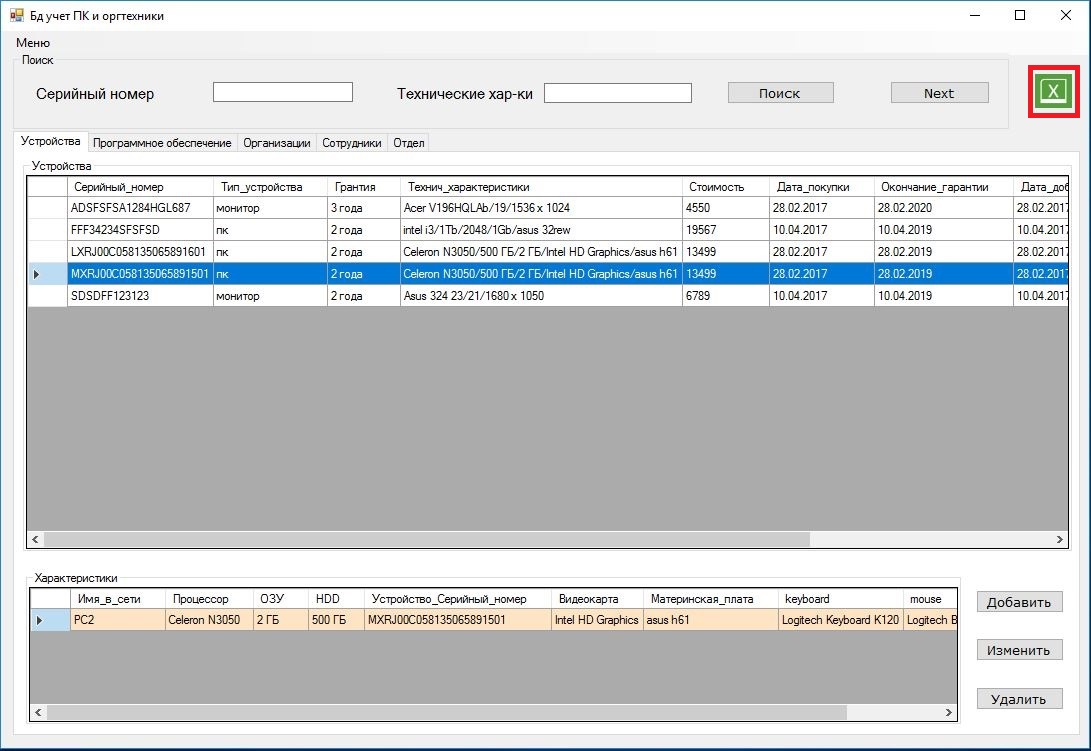


Рисунок 20 – Кнопка экспорта данных таблицы в MS Excel

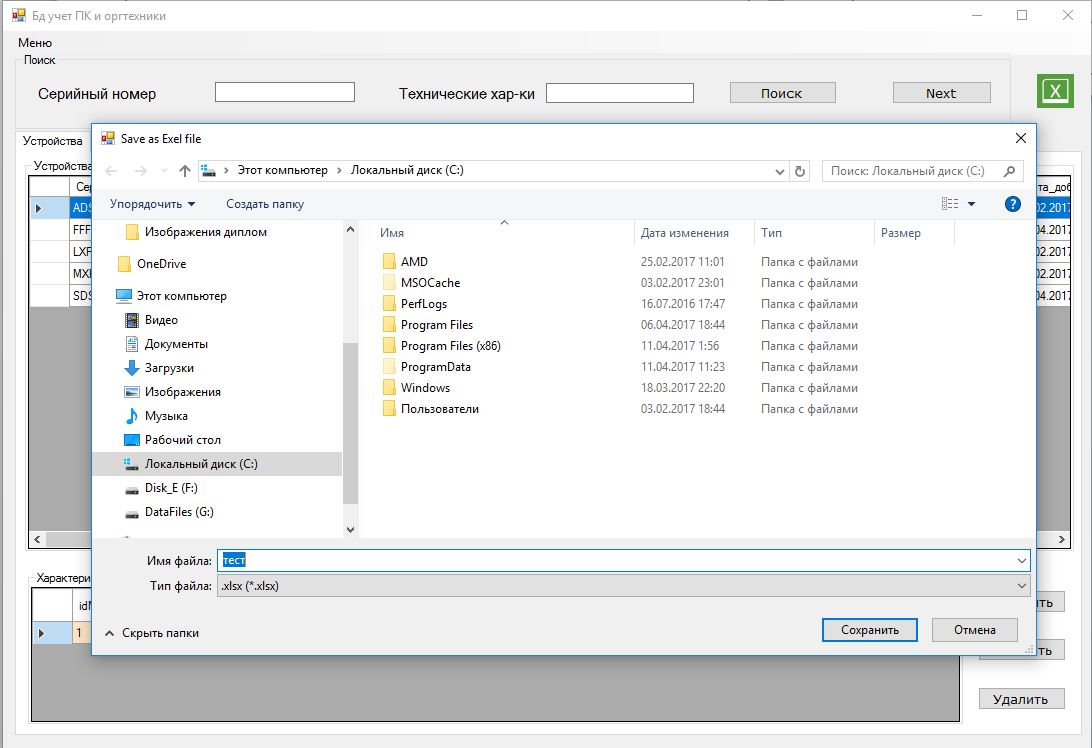


Рисунок 21 – Выбор пути для экспорта данных таблицы в MS Excel

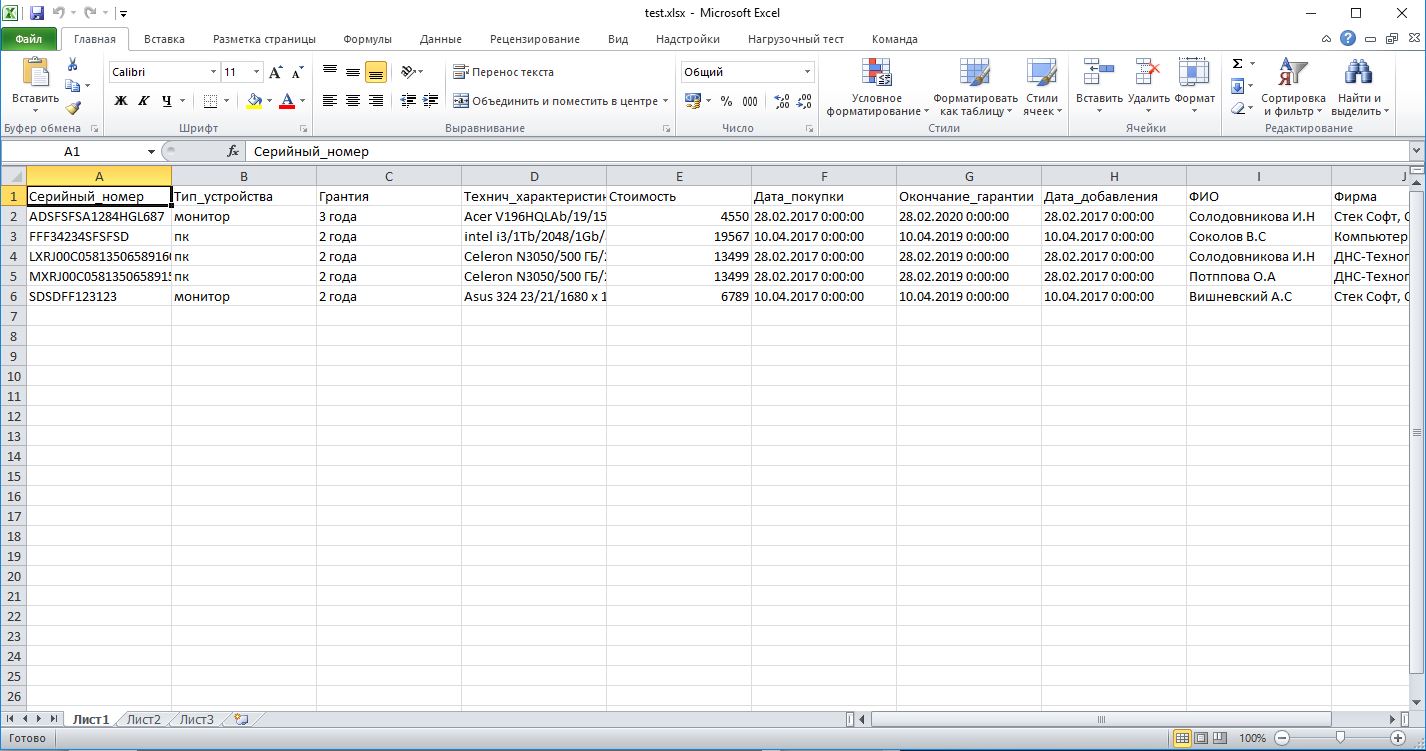


Рисунок 22 – Файл MS Excel экспортированный из таблицы

Для сортировки нужно нажать по любому полю в главной таблице. Форма таблицы показана на рисунке 23.

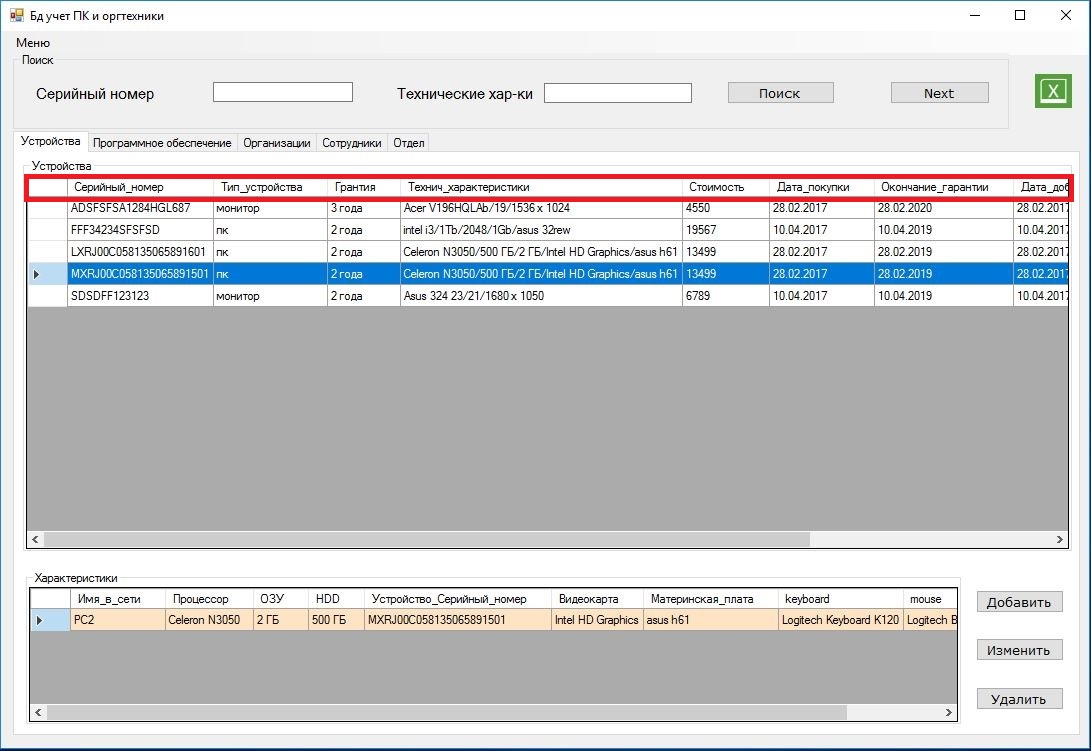


Рисунок 23– Сортировка по полю таблицы

3 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

3.1 Расчет затрат на разработку программы и решение задачи на ЭВМ

Основными компонентами затрат на разработку программ и решение задачи на ЭВМ являются затраты, связанные с оплатой труда специалистов на разработку программы, обслуживание и эксплуатацию ЭВМ в период отладки программы и решения задачи, то есть рассчитываются прямые и косвенные затраты.

При определении полной себестоимости программы учтены все материальные расходы, расходы по заработной плате, отчисления в социальные статьи и составлена калькуляция затрат в следующей последовательности:

- основная заработная плата персонала;

- дополнительная заработная плата персонала;

- отчисления в социальные статьи;

- стоимость работ на ВЦ;

- расчет стоимости материалов;

- расчет косвенных затрат на разработку программы.

При расчете всех экономических показателей была составлена таблица 1, в которой указаны все этапы работы по разработке программы и решению задачи, исполнитель каждого этапа, трудоемкость и стоимость исполнения.

Стоимость каждого этапа определена, исходя из оклада исполнителей и времени выполнения этапа.

Количество рабочих часов в месяце равно 168 часов, то есть 21 рабочий день в месяце по 8 часов.

Стоимость часа работы руководителя и программиста определяется по следующим формулам 1 и 2.

Ст.ч.р = Оклад.р / К.р.ч, (1)

где Оклад.р – оклад руководителя, руб.;

Ст.ч.р. – стоимость часа работы руководителя;

К.р.ч. – количество рабочих часов в месяце, час.

*Стоимость часа работы руководителя*

Ст.ч.р = 25600/ 168 = 152,3 руб./час.

Ст.ч.п=Оклад.п / К.р.ч, (2)

где Ст.ч.п – стоимость часа работы программиста, руб.;

Оклад.п – оклад программиста;

К.р.ч – количество рабочих часов в месяце, час.

*Стоимость часа работы программиста*

Ст.ч.п = 15000 / 168 = 89,2 руб./час.

Таблица 1 – Этапы разработки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование этапов работ | Исполнитель | Трудоемкость (час) | Оклад (руб.) | Плата за час (руб.) | Стоимость исполнения |
| Постановка задачи | Руководитель | 8 | 25600 | 152,3 | 1218,4 |
| Программист | 8 | 15000 | 89,2 | 713,6 |
| Изучение литературы | Программист | 46 | 15000 | 89,2 | 4103,2 |
| Технический проект | Программист | 16 | 15000 | 89,2 | 1427,2 |
| Руководитель | 8 | 25600 | 152,3 | 1218,4 |
| Эскизный проект | Программист | 23 | 15000 | 89,2 | 2051,6 |
| Руководитель | 15 | 25600 | 152,3 | 2284,5 |
| Кодирование | Программист | 79 | 15000 | 89,2 | 7046,8 |
| ЭВМ | 86 | 10 | 10 | 860 |
| Отладка программы  Тестирование | Программист | 35 | 15000 | 89,2 | 3122 |
| ЭВМ | 40 | 10 | 10 | 400 |
| Руководитель | 23 | 25600 | 152,3 | 3502,9 |
| Оптимизация программы | Программист | 26 | 15000 | 89,2 | 2319,2 |
| Руководитель | 8 | 25600 | 152,3 | 1218,4 |
| ЭВМ | 20 | 10 | 10 | 200 |
| Оформление сопроводитель  ной документации | Программист | 37 | 15000 | 89,2 | 3304,4 |
| Руководитель | 32 | 25600 | 152,3 | 4873,6 |
| ЭВМ | 32 | 10 | 10 | 320 |
| Итого: | Программист | 270 |  |  | 24106 |
| Руководитель  ЭВМ | 94  178 |  |  | 14316,2  1780 |

Основная заработная плата персонала рассчитывается по формуле

ЗПо = ЗПч × Тч × Кр, (3)

где ЗПч – заработная плата за час;

Тч – количество отработанных при разработке программы часов;

ЗПо  - основная заработная плата персонала;

Кр – районный коэффициент.

*Основная заработная плата руководителя*

ЗПо = 152,3× 94 × 1,3 = 18611 руб.

*Основная заработная плата программиста*

ЗПо = 89,2 × 270 × 1,3 = 31309,2 руб.

Дополнительная заработная плата персонала рассчитывается по формуле

ЗПдоп = ЗПо × 0,10, (4)

где ЗПдоп  – дополнительная заработная плата персонала;

ЗПо – основная заработная плата персонала, руб.

*Дополнительная заработная плата руководителя*

ЗПдоп = 18611× 0,1 = 1861.1 руб.

*Дополнительная заработная плата программиста*

ЗПдоп = 31309,2 × 0,1 = 3130.9 руб.

Отчисления в социальные статьи рассчитываются по формуле

Осс=(ЗПо+ЗПдоп)×0,30, (5)

где ЗПдоп  - дополнительная заработная плата;

ЗПо – основная заработная плата;

Осс  - отчисления в социальные статьи.

Отчисления в социальные статьи составляют 30%, из них:

- в пенсионный фонд(Опф) отчисляется 22%;

- в фонд социального страхования(Осстр) – 2,9%;

- в фонд медицинского страхования(Омс) – 5,1%.

Отчисления в социальные статьи от заработной платы руководителя рассчитывается по формуле 5

где Опф – отчисления в пенсионный фонд;

Осстр – отчисления в фонд социального страхования;

Омс – отчисления в фонд медицинского страхования;

Осс – отчисления в социальные статьи – это сумма отчислений в пенсионный фонд, отчислений в фонд социального страхования и отчислений в фонд медицинского страхования.

Опф= (18611 + 1861.1) × 0,22 = 4503.8 руб.;

Осстр= (18611 + 1861.1) × 0,029 = 593.6 руб.;

Омс= (18611 + 1861.1) × 0,051 = 1044 руб.;

Осс= 4503.8 + 593.6 + 1044 = 6141.4 руб.

Отчисления в социальные статьи от заработной платы программиста рассчитывается по формуле 5:

Опф= (31309,2 + 3130,9) × 0,22 = 7576,8 руб.;

Осстр= (31309,2 + 3130,9) × 0,029 = 998,7 руб.;

Омс= (31309,2 + 3130,9) × 0,051 = 1756,4 руб.;

Осс= 7576,8 + 998,7 + 1756,4 = 10331,9 руб.

Стоимость работ на ВЦ рассчитывается по формуле 6.

Срм=Смч×Тм, (6)

где Смч – стоимость машинного часа в рублях (10 руб./час);

Срм – стоимость работ на ВЦ;

Тм – общее время работы ЭВМ (час).

*Стоимость работ на ВЦ*

Срм = 10 × 178 = 1780 руб.

Расчет стоимости материалов приведен в таблице 2.

Таблица 2 – Материальные затраты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | Единица измерения | Количество | Цена за 1 ед., руб | Стоимость, руб. |
| Бумага (А4, 500 л, белая) | уп. | 1 | 300 | 300 |
| Тетрадь | шт. | 1 | 5 | 5 |
| Карандаш | шт. | 1 | 10 | 10 |
| Ручка | шт. | 2 | 15 | 30 |
| Степлер | шт. | 1 | 400 | 400 |
| Мультифора | шт. | 10 | 3 | 30 |
| Итого: |  |  |  | 875 |

Расчет косвенных затрат на разработку программы рассчитывается по формуле 7

Рк = ЗПо × 0,08 руб., (7)

где Рк – расчет косвенных затрат на разработку программы;

ЗПо – основная заработная плата персонала.

Рк = (18611 + 31309,2) × 0,08 = 3993.6 руб.

Полная себестоимость программы приведена в таблице 3.

Таблица 3 – Смета затрат на разработку

|  |  |
| --- | --- |
| Наименование статей расходов | Стоимость работ (руб.) |
| Основная заработная плата  - программиста  - руководителя | 31309,2  18611 |
| Дополнительная заработная плата  - программиста  - руководителя | 3130,9  1861.1 |
| Отчисления в социальные статьи  - программиста  - руководителя | 10331,9  6141.4 |
| Стоимость работ на ВЦ | 1780 |
| Стоимость материалов | 875 |
| Косвенные расходы | 3993.6 |
| Полная себестоимость: | 77989,1 |

1. Расчет годовых затрат на эксплуатацию программы

Стоимость одного непосредственного решения на ЭВМ определяется по формуле

Ср.м=Смч×Тр+ЗПо.п.×Q×Кр×Ккр, (8)

где Ср.м – стоимость одного непосредственного решения на ЭВМ;

Смч – стоимость работы на ЭВМ за час (10 руб./час);

Тр – время решения задачи на ЭВМ (0,005 час);

Q – трудоемкость менеджера (1 час);

Кр – районный коэффициент (1,3);

Ккр – коэффициент косвенных расходов (1,08);

ЗПо.п – заработная плата за час работника (89,2).

*Стоимость одного непосредственного решения на ЭВМ*

Ср.м = 10 × 0,005 + 89,2 × 1 × 1,3 × 1,08 = 125.2 руб.

Расчет годовых затрат на разработку программы необходимо провести для последующего анализа эффективности данного программного продукта.

Готовые затраты на эксплуатацию программы рассчитываются по формуле

Ср.м.год=N×Ср.м+Ен×С, (9)

где N – плотность потока заявок (3500);

Ср.м.год – годовые затраты на эксплуатацию программы;

Ср.м – стоимость одного непосредственного решения на ЭВМ;

Ен – нормальный коэффициент сложности (0,2);

С – себестоимость разработки программы (77989,1).

*Расчет годовых затрат на разработку программы*

Ср.м.год = 3500 × 125.2 + 0,2 × 77989,1= 453797.82 руб.

Расчет годовых затрат на выполнение работ без использования программы по учету материалов для полиграфической компании «4 Цвета».

Расходы на выполнение работ без использования программы по учету материалов для полиграфической компании «4 Цвета» рассчитываются по формуле

Ср.сп=ЗПсп×Тсп×Ккр×Кр, (10)

где ЗПсп – заработная плата работника (100 руб.);

Тсп – затраты времени специалиста на оформление денежного перевода (2 ч);

Кр – районный коэффициент (1,3);

Ккр – коэффициент косвенных расходов (1,08).

*Расходы на выполнение работ без использования электронного учебника*

Ср.сп= 100 × 2 × 1,08 × 1,3 = 280,8 руб.

Зная стоимость всех работ по выполнению одного расчета, определим годовые расходы без использования электронного учебника

Ср.сп.год=N×Ср.сп, (11)

где N – плотность потока заявок (3500);

Ср.сп.год – годовые расходы ранее употреблявшимся способом.

3.2 Расчет годовых затрат на выполнение работ без использования программы по учету материалов

Ср.сп.год = 3500 × 280,8 = 982800 руб.

Эгод = Ср.сп.год – Ср.м.год

где Ср.сп.год – годовые затраты на выполнение работ ранее употреблявшимся способом;

Ср.м.год – годовые затраты на эксплуатацию программы.

Определение коэффициента экономической эффективности программы.

Эгод = 982800 - 453797.82 = 529002.1

Коэффициента экономической эффективности показывает сколько на 1 руб. вложенных затрат в разработку и эксплуатацию, получаем экономии. Чем больше данное значение, тем эффективнее проект.

Ер = Эгод/(С+ Ср.м.год)

Ер = 529002.1 / (77989,1 + 453797.82) = 0.994

Таким образом, на один вложенный рубль в разработку и эксплуатацию, получаем 0.994 экономии.

Определение срока окупаемости

Срок окупаемости показывает через какой период времени возместятся затраты на разработку и эксплуатацию.

Ток = (С+ Ср.м.год)/ Эгод

Ток = (77989,1 + 453797.82) /529002.1= 1.005

Ток = 1/ Ер

Ток = 1/0.994 = 1.005

Срок окупаемости программы за счет экономии средств составил порядка 1(одного) года.

Заключение

Требовалось разработать информационную систему, представляющую собой программу для учета компьютеров, оргтехники и установленного ПО (Программное обеспечение).

В ходе разработки дипломного проекта были реализованы все поставленные задачи, а именно была реализована база данных с последующим приведением ее к третьей нормальной форме и реализован графический интерфейс информационной системы. Реализованы функции хранения, добавления, удаления и изменения данных. Так же реализован экспорт данных в таблицу Excel, поиск и сортировка.

За счет понятного графического интерфейса, программой удобно пользоваться как системному администратору, так и обычному сотруднику.

Программа реализована с учетом тех компаний, где имеется большой парк компьютеров и оргтехники.

Решена проблема которые присущи электронным таблицам ([Microsoft Excel](https://ru.wikipedia.org/wiki/Microsoft_Excel), [OpenOffice](https://ru.wikipedia.org/wiki/OpenOffice.org_Calc)) а именно размер данных, простота обновления, точность, безопасность и проблемы с избыточностью.

Список использованных источников

1. Виктор Гольцман. MySQL 5.0. Библиотека программиста. 1-е издание. - СПб.: Питер, 2016. – 546 с.
2. Джеффри Рихтер. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е издание. - СПб.: Питер, 2013. – 1245 с.

3) Эндрю Троелсен. Язык программирования C# 6.0 и платформа .NET 4.6. 7-e издание. – Москва.: Вильямс, 2016. – 1440 с.