Министерство образования и науки Хабаровского края

Краевое государственное бюджетное

профессиональное образовательное учреждение  
«Комсомольский-на-Амуре колледж технологий и сервиса»

Кафедра математики, информатики и программирования

Специальность 09.02.07 «Информационные системы и программирование»

|  |  |
| --- | --- |
|  | Допущен к защите  Зам. директора по учебной работе  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. В. Гринева  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ДИПЛОМНОМУ ПРОЕКТУ

Информационная система маршрутно-транспортных средств

|  |  |
| --- | --- |
| Н.КОНТР.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.С. Сегиневич | РУКОВОДИТЕЛЬ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.М. Морошкин |
|  | ИСПОЛНИТЕЛЬ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.А. Дьяченко |
| РЕЦЕНЗЕНТ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Н. Анисимов |

2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc137210624)

[1 Расчетно-теоретический раздел 8](#_Toc137210625)

[1.1 Состояние проблемы 8](#_Toc137210626)

[1.2 Основные понятия и определения 9](#_Toc137210627)

[1.3 Методы решения проблемы 11](#_Toc137210628)

[1.4 Обоснование выбранного метода 15](#_Toc137210629)

[2 Раздел программной документации 17](#_Toc137210630)

[2.1 Техническое задание 18](#_Toc137210631)

[2.1.1 Введение 18](#_Toc137210632)

[2.1.2 Основание для разработки 18](#_Toc137210633)

[2.1.3 Назначение разработки 19](#_Toc137210634)

[2.1.4 Требования к программе и программному обеспечению 19](#_Toc137210635)

[2.1.5 Требование к надежности 20](#_Toc137210636)

[2.1.6 Требования к программным документам 20](#_Toc137210637)

[2.1.7 Требования к составу и параметрам технических средств 21](#_Toc137210638)

[2.1.8 Порядок контроля и приемки программного обеспечения 21](#_Toc137210639)

[2.1.9 Требования к маркировке и упаковке 21](#_Toc137210640)

[2.1.10 Требования к транспортированию и хранению 21](#_Toc137210641)

[2.2 Спецификация 23](#_Toc137210642)

[2.3 Описание проблемы 24](#_Toc137210643)

[2.3.1 Общие сведения 24](#_Toc137210644)

[2.3.2 Функциональное назначение 25](#_Toc137210645)

[2.3.3 Описание логической структуры 25](#_Toc137210646)

[2.3.4 Используемые технические средства 27](#_Toc137210647)

[2.3.5 Вызов и загрузка 27](#_Toc137210648)

[2.3.6 Входные данные 27](#_Toc137210649)

[2.3.7 Выходные данные 28](#_Toc137210650)

[2.4.1 Код страницы «VoditeliAddPage.xaml.cs» 28](#_Toc137210651)

[2.4.2 Код страницы «PostupleniaPage.xaml.cs» 29](#_Toc137210652)

[2.4.3 Код страницы «PostupleniaAddPage.xaml.cs» 31](#_Toc137210653)

[2.4.4 Код страницы «MainWindow.xaml.cs» 32](#_Toc137210654)

[2.4.5 Код страницы «RegPage.xaml.cs» 33](#_Toc137210655)

[2.4.6 Код страницы «VoditeliPage.xaml.cs» 34](#_Toc137210656)

[2.4.7 Код страницы «Manager.cs» 35](#_Toc137210657)

[2.4.8 Код страницы «App.xaml» 36](#_Toc137210658)

[2.4.9 Код страницы «MainWindow.xaml» 36](#_Toc137210659)

[2.4.10 Код окна «PostupleniaAddPage.xaml» 36](#_Toc137210660)

[2.4.11 Код файла «PostupleniaPage.xaml» 37](#_Toc137210661)

[2.5 Руководства программиста 41](#_Toc137210662)

[2.5.1 Общие положения 41](#_Toc137210663)

[2.5.2 Назначение и цель создания 41](#_Toc137210664)

[2.5.3 Характеристики программы 41](#_Toc137210665)

[2.5.4 Обращение к программе 42](#_Toc137210666)

[2.5.5 Входные и выходные данные 42](#_Toc137210667)

[2.5.6 Возможности модификации программы 43](#_Toc137210668)

[2.6 Руководства оператора 44](#_Toc137210669)

[2.6.1 Назначение программы 44](#_Toc137210670)

[2.6.2 Условия выполнения программы 45](#_Toc137210671)

[2.6.3 Выполнение программы 45](#_Toc137210672)

[2.7 Тестирование программного модуля 50](#_Toc137210673)

[ЗАКЛЮЧЕНИЕ 51](#_Toc137210674)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 52](#_Toc137210675)

# **ВВЕДЕНИЕ**

Развитие компьютерных технологий не стоит на месте. В данное время происходит большой спрос на качественную и удобную автоматизированную работув каждой отрасли промышленности, так как в наше время бурно развиты компьютерные технологии.

Многие процессы на предприятиях можно облегчить при помощи внедрения информационных технологий. Они могут улучшить документационные процессы на предприятии, позволяя руководству использовать больший объем информации, и устраняя наиболее трудоемкие операции при работе с документацией.

Каждый год, после выпуска девятых и одиннадцатых классов, многие обучающиеся стремятся поступить в новое учебное заведение. Из-за большого потока абитуриентов, информацияпродолжает увеличиваться, возникает необходимость в ее обработке. Понятно, что работать с огромной кучей бумажной информации долго и трудоемко, выходом из такого положения дел стало создание автоматизированного рабочего места (АРМ).

Основные преимущества автоматизации и новой технологии обработки информации сказываются там, где приходится выполнять повторяющиеся задачи, предусматривающие запрограммированные решения, либо задачи с большим объемом информации или чисто механического труда.

Тема дипломного проекта – «Информационная система маршрутно-транспортных средств». Объектом исследование является общество с ограниченной ответственностью «Дальавтотранс» (ООО «Дальавтотранс»).

Целью дипломного проекта является разработка программного обеспечения, необходимого для автоматизации процессов работы бухгалтерий.

Тема дипломного проекта актуальна тем, что в ООО «Дальавтотранс» нет такого программного продукта, которым предприятие могло пользоваться, с помощью которого требуется обрабатывать постоянно увеличивающуюся информацию, из-за чего возникает необходимость в автоматизации процесса.

Этим программным продуктом смогут пользоваться несколько людей. Сотрудник бухгалтери который ответсвенен за составление ежедневного финансого отчёта, и водители маршрутно транспортных средств, которые присылают отчёт о прибыли за прохождение машрута.

Главная цель программного продукта заключается в ведении учета прибыли и запраты водителей маршрутно транспортых средст.

Практика показывает, что применение автомотизации подсчёта прибыли позволяет в значительной мере снизить время, затрачиваемое на учет и анализ полученных данных.

Задачи дипломного проектирования:

* охарактеризовать основные элементы АРМ, которые должны соответствовать поставленной цели дипломного проектирования;
* проектирование и разработка информационной базы;
* спроектировать функциональный и понятный интерфейс для пользователя программой, а именно сотрудника приёмной комиссии;
* разработать программное обеспечение, удовлетворяющее требования заказчика;
* осуществить комплексное тестирование и отладку программного обеспечения;
* сформировать полный комплект программной документации;
* осуществить внедрение и сопровождение программного обеспечения.

# **Расчетно-теоретический раздел**

## **Состояние проблемы**

На сегодняшний день управление всякой деятельностью основывается на анализе огромного объёма информации, и её обработки при помощи автоматизированной системы. Использование вычислительной техники во всевозможных областях деятельности человека прошло большой путь, который определялся не только развитием собственно техники, но и развитием принципов и методов обработки информации как с точки зрения областей применения, так и с точки зрения широты использования.

Популярны системы, обхватывающие практически целый ансамбль учётных задач.

Как правило, такие системы имеют дело с большими объёмами информации, имеющей достаточно сложную структуру. Решающее значение для эффективности систем подобного рода имеет то обстоятельство, что они опираются на автоматизированные информационные базы. Это означает, что в памяти ЭВМ постоянно сохраняется информация, нужная для решения тех задач, на которые рассчитана система. Она и составляет содержимое информационной базы соответствующей системы.

Хранение информации в памяти ЭВМ придаёт этой информации принципиально новое качество динамичности, т.е. способности к быстрой перестройке и непосредственному её использованию в решаемых на ЭВМ задачах.

Целью выполнения дипломного проекта является разработка программного обеспечения, необходимого для системы поступающих средств, где будет реализован подсчёт прибыли и зарплаты водителям. Данное приложение должно обеспечить эффективную работу с данными сотрудников бухгалтерий.

## **Основные понятия и определения**

При программировании используются следующие понятия и определения:

Алгоритм – это набор инструкций или правил, предназначенных для решения определенной проблемы. Проблема может быть простой, как добавление двух чисел или сложное, например, преобразование видеофайла из одного формата в другой.

Интерфейс прикладного программирования (API) – это набор правил, процедур и протоколов для создания программных приложений. API помогают в общении со сторонними программами или службами, которые можно использовать для создания различного программного обеспечения.

Объект – это комбинация связанных переменных, констант и других структур данных, которые могут выбираться и обрабатываться вместе.

Объектно-ориентированное программирование (ООП) – это модель, определяемая программистами, которые вращаются вокруг объектов и данных, а не «действий» и «логики». В ООП определяется не только тип данных структуры данных, но и типы функций, которые могут быть применены к ней. Благодаря этому структура данных становится объектом, состоящим из данных и функций.

Переменная – это место, в котором хранятся временные данные в программе, которые могут быть изменены, сохранены и отображены при необходимости.

Класс – в ООП класс относится к набору связанных объектов с общими свойствами. Классы и возможность создавать новые классы делают ООП мощной и гибкой моделью программирования.

Исключение – особое, неожиданное и аномальное состояние, возникающее во время выполнения программы. Это также можно назвать ошибкой или условием, которое изменяет путь программы или микропроцессора на другой путь.

Фреймворк в программировании – это фундамент с определенным уровнем сложности, который может быть изменен программистом, используя свой код. Платформа может включать в себя различные библиотеки программного обеспечения, API, компиляторы и многое другое.

При работе с базой данных используются следующие понятия и определения:

Структурные элементы базы данных.

База данных – это хранилище для большого количества систематизированных данных, с которыми можно производить определенные действия (добавление, удаление, изменение, копирование, упорядочивание и т. д.). Все данные, находящиеся в базе данных, можно представить в виде записей или объектов. Для успешной работы с базами данных нужны программные средства, которые обеспечивали бы доступ к нужной информации, внесение каких-либо изменений в базу данных и другие действия с данными. Понятие базы данных тесно связано с такими понятиями структурных элементов, как поле, запись, файл (таблица).

Поле – элементарная единица логической организации данных, которая соответст­вует неделимой единице информации – реквизиту. Имя поля часто называют атрибутом.

Запись – совокупность данных, относящихся к одному человеку, месту, событию или другому элементу базы данных. В режиме таблицы для формы, таблицы или запроса запись выводится в виде строки.

Файл (таблица) – совокупность экземпляров записей одной структуры.

В структуре записи файла указываются поля, значения которых являются ключами первичными (ПК), которые идентифицируют экземпляр записи, и вторичными (ВК), которые выполняют роль поисковых или группированных признаков (по значению вторичного ключа можно найти несколько записей).

Индекс – совокупность указателей, содержащих информацию о местоположении записи.

Связь – это функциональная зависимость между сущностями. Если между некоторыми сущностями существует связь, то факты из одной сущности ссылаются или некоторым образом связаны с фактами из другой сущности.

Запрос – требование на отбор данных, хранящихся в таблицах, или требование на выполнение определенных действий с данными. Запрос позволяет создавать общий набор записей из данных, находящихся в разных таблицах, который будет служить источником данных для формы, отчета.

Система управления базами данных (СУБД) – это совокупность языковых и программных средств, обеспечивающих создание, использование и ведение базы данных. СУБД основывается на использовании иерархической, сетевой или реляционной моде­ли, на комбинации этих моделей или на некотором их подмножестве.

Реляционная модель данных – модель характеризуется простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Первичный ключ – это уникальный идентификатор для каждой записи, значение которого однозначно определяет только эту запись.

Первичный ключ должен обладать двумя свойствами:

* однозначной идентификацией записи;
* отсутствием избыточности - никакое поле нельзя удалить из ключа, не нарушая при этом однозначности (первого свойства).
  1. Методы решения проблемы

Для создания структуры базы данных была выбрана реляционная модель данных. Реляционная модель данных (РМД) – логическая модель данных, строгая математическая теория, описывающая структурный аспект, аспект целостности и аспект обработки данных в реляционных базах данных.

Для решения поставленной задачи используется метод нормализации. Целью нормализации является устранение недостатков структуры базы данных, приводящих к вредной избыточности в данных, которая в свою очередь потенциально приводит к различным аномалиям и нарушениям целостности данных.

Нормализация – это процесс, позволяющий гарантировать, эффективность структур данных в реляционной базе данных.

Нормальная форма – формальное свойство отношения, которое характеризует степень избыточности хранимых данных и возможные проблемы.

Реляционная модель ориентирована на организацию данных в виде двумерных таб­лиц. Каждая реляционная таблица представляет собой двумерный массив и обла­дает следующими свойствами:

* каждый элемент таблицы – один элемент данных;
* все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
* каждый столбец имеет уникальное имя;
* одинаковые строки в таблице отсутствуют.

Отношения представлены в виде таблиц, строки которых соответствуют кортежам или записям, а столбцы – атрибутам отношений, доменам, полям.

Чтобы связать две реляционные таблицы, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей); в противном случае нужно ввести в структуру первой таблицы внешний ключ – ключ второй таблицы. Элементы реляционной модели представлены в таблице 1.

Отношение является важнейшим понятием и представляет собой двумерную таблицу, содержащую некоторые данные.

Сущность есть объект любой природы, данные о котором хранятся в базе данных. Данные о сущности хранятся в отношении.

Таблица 1 – Элементы реляционной модели

|  |  |
| --- | --- |
| Элементы реляционной модели | Форма представления |
| Отношение | Таблица |
| Кортеж | Строка таблицы |
| Схема отношения | Строка заголовков таблицы (заголовок таблицы) |
| Сущность | Описание свойств объекта |
| Атрибут | Заголовок столбца таблицы |
| Домен | Множество допустимых значений атрибута |
| Значение атрибута | Значение поля в записи |
| Первичный ключ | Один или несколько атрибутов |
| Тип данных | Тип значений элементов таблицы |

Атрибуты представляют собой свойства, характеризующие сущность. В структуре таблицы каждый атрибут именуется и ему соответствует заголовок некоторого столбца таблицы.

Домен представляет собой множество всех возможных значений определенного атрибута отношения.

Схема отношения (заголовок отношения) представляет собой список имен атрибутов. Множество собственно кортежей отношения часто называют содержимым (телом) отношения.

Ключ отношения или первичный ключ – атрибут отношения, однозначно идентифицирующий каждый из его кортежей.

Для лучшего понимания РМД следует отметить три важных обстоятельства:

* модель является логической, т.е. отношения являются логическими (абстрактными), а не физическими (хранимыми) структурами;
* для реляционных баз данных верен информационный принцип: все информационное наполнение базы данных представлено одним и только одним способом, а именно — явным заданием значений атрибутов в кортежах отношений; в частности, нет никаких указателей (адресов), связывающих одно значение с другим;
* наличие реляционной алгебры позволяет реализовать декларативное программирование и декларативное описаний ограничений целостности, в дополнение к навигационному (процедурному) программированию и процедурной проверке условий.

Построение пользовательского интерфейса и реализация программного обеспечения, а также создания БД осуществляется с помощью Microsoft SQL Server 2019, среды разработки Visual Studio на языке C# на платформе WPF (.NET Framework).

Microsoft SQL Server – система управления реляционными базами данных (РСУБД), которая была разработана фирмой Microsoft. В данной системе используется язык запросов Transact-SQL, созданный совместными усилиями корпорации Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Microsoft SQL Server предназначен для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия.

Microsoft Visual Studio – интегрированная среда разработки (IDE), разработанная корпорацией Microsoft. Visual Studio позволяет разрабатывать множество различных приложений, среди который можно упомянуть: консольные приложения, игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, WPF, UWP а также веб-сайты, веб-приложения, веб-службы. Visual Studio включает в себя редактор исходного кода и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне.

Язык программирования C# – объектно-ориентированный язык программирования. Разработан всеми известной корпорацией Microsoft как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов, делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Платформа Windows Presentation Foundation (WPF) позволяет создавать клиентские приложения для настольных систем Windows с привлекательным пользовательским интерфейсом. В основе WPF лежит независимый от разрешения векторный модуль визуализации, использующий возможности современного графического оборудования. Возможности этого модуля расширяются с помощью комплексного набора функций разработки приложений, которые включают в себя язык XAML, элементы управления, привязку к данным, макет, двумерную и трехмерную графику, анимацию, стили, шаблоны, документы, мультимедиа, текст и типографические функции. WPF является частью .NET.

## **Обоснование выбранного метода**

База данных, спроектированная по принципам реляционной модели, обладает рядом достоинств, таких как:

* простота и доступность понимания конечным пользователем – единственной информационной конструкцией является таблица;
* при проектировании реляционной БД применяются строгие правила, базирующие на математическом аппарате;
* полная независимость данных. При изменении структуры реляционной базы данных изменения, которые требуется произвести в прикладных программах, как правило, минимальны;
* для построения запросов и написания прикладных программ нет необходимости знания конкретной организации БД во внешней памяти.

Для решения задачи рассматривались два варианта, «1C:Бухгатерия» и Microsoft SQL Server.

Перечислим основные достоинства Microsoft SQL Server:

* масштабирование системы. Взаимодействовать с ней можно как на простых ноутбуках, так и на ПК с мощным процессором, который способен обрабатывать большой объем запросов;
* размер страниц – до 8 Кб. Данные извлекаются быстро, а сложную информацию удобнее хранить. Система обрабатывает транзакции в интерактивном режиме, есть динамическая блокировка;
* автоматизация рутинных административных задач. Например, управление блокировками и памятью, редактура размеров файлов. В программе продуманы настройки, можно создавать профили пользователей;
* удобный поиск. Его можно осуществлять по фразам, словам, тексту либо создавать ключевые индексы;
* поддержка работы с другими решениями Майкрософт, в том числе с Excel, Access;
* распространенность, которая обусловлена тем, что SQL Server является продуктом компании Microsoft;
* постоянно обновляется производителем, поддерживает множество языков;
* полностью совместим с операционной системой Windows;
* ориентированность на пользователя с разной профессиональной подготовкой, что выражается в наличии большого количества Мастеров, развитую систему справки и понятный интерфейс.

После была рассмотрена систему «1С: Бухгатерия», но из-за следующих недостатков было принято решение отказаться от данной идеи. Недостатки «1С: Бухгатерия»:

* необходимость изначального внесения большого количество информации, что требует лишнее время и чьё-то внимание;
* большая стоимость продукта;
* перегруженность лишними модулями и неиспользуемыми функциями;
* недружелюбный интерфейс.

Таким образом, СУБД Microsoft SQL Server оптимально подходит для разработки данного программного продукта, так как с ее помощьювозможно решать широкий спектр задач, при этом ее инструментарий прост в использовании.

Среди множества языков программирования, C# является наиболее удобным для меня языком программирования, так как именно ему я обучался на протяжении всего учебного процесса в колледже. При этом его удобно использовать для создания сложных приложений, управления базами данных, приложений масштаба предприятия.

Основные конструкции языка четко выделяются в программе, что способствует хорошему восприятию написанного кода. Графический интерфейс языка отлично продуман, среда разработки предоставляет доступ только к тем участкам кода, с которыми необходимо работать, скрывая основную часть программы, которая создаётся автоматически самой разработкой.

Инструментальные средства языка позволяют:

* создавать интерфейс, используя стандартные компоненты;
* передавать управление процессам, в зависимости от состояния системы;
* создавать оболочки для баз данных, как и сами базы данных;
* разрабатывать более надежные программы путём обработки исключительных ситуаций, возникающих при некорректной работе программы.

# **2 Раздел программной документации**

В разделе программной документации содержатся следующие программные документы:

1. техническое задание, в котором представлены технические характеристики, программное обеспечение, общие требования, которые необходимы для функционирования данной программы;
2. спецификация;
3. описание программы, в котором описываются обозначение и наименование программы, программное обеспечение, необходимое для функционирования программы. указывается назначение языка программирования, на котором написана программа и ее функциональное назначение;
4. текст программы, в котором представлен текст программы;
5. руководство программиста, где описываются действия программиста для установки программы и работы с ней, а также возможная модернизация текста программы;
6. руководство оператора, в котором описываются все необходимые действия, которые должен сделать оператор для работы с приложением.

## **2.1 Техническое задание**

### **2.1.1 Введение**

Наименование разработки дипломного проекта: «Информационная система маршрутно-транспортных средств».

Данная программа предназначена для бухгалтеров и водителей маршрутно транспортных средств.

Разработчик: студент группы 541 М.А.Дьяченко.

Заказчик – Общество с ограниченной ответственностью «Дальавтотранс»

Сроки работы с 20 апреля 2023г по 17 мая 2023г.

### **2.1.2 Основание для разработки**

Использование созданного программного обеспечения позволит упростить работу сотрудникам бухгалтерии, а также позволит формировать отчеты по дате, что существенно увеличит скорость и качество выполняемой работы.

Использование созданного программного обеспечения позволит упростить работу водителей маршрутно транспортных средств.

Настоящее техническое задание распространяется на разработку информационной системы маршрутно-транспортных средств. Предполагается, что использовать данную систему будет сотрудники бухгалтери.

### **Назначение разработки**

Данная программа является средством учета прибыли поступающих с маршрутов автобусов.

Приложение предназначено для применения в Общество с ограниченной ответственностью «Дальавтотранс», и разработана специально для работников бухгалтерий. Основная задача программы: обеспечить учет и быстрый поиск сведений о поступающих средствах с маршрутов.

Графический удобный и понятный интерфейс позволяет быстро разобраться в работе с программой и не предполагает наличия специальных навыков для работы с базой.

Предложенная система значительно упрощает труд сотрудников. Эффективность программы позволяет сократить время на заполнение данных, избавляет от рутинной работы, существенно уменьшает количество возможных ошибок.

* + 1. Требования к программе и программному обеспечению

1. Системные требования:

* операционная система: Microsoft Windows7/10/Server 2016 (32 или 64 bit);
* процессор: двухъядерный Intel Core i3 M350 2.27 ~ 2.3 Ghz;
* оперативная память (ОЗУ): 8ГБ и выше;
* HDD or SSD: от 50 ГБ.

1. Прикладные программы:

* Microsoft SQL Server 2019;
* SQL Server Management Studio;
* Visual Studio 2019.

### **2.1.5 Требование к надежности**

Для предупреждения потери данных на компьютере следует периодически делать резервную копию базы данных, а также хранить копии еженедельных и ежемесячных резервных записей на диске в отдельном от компьютера безопасном месте.

Для успешной работы программы следует исключить перепады напряжения в электросети, к которой непосредственно подсоединен компьютер. В случае сбоя в работе следует загрузить программу заново. Для восстановления поврежденных данных необходимо выполнить операцию «Сжать и восстановить» или восстановить базу данных из резервной копии.

Для исключения несанкционированного доступа к данным, хранимым в базе данных, реализованна защита, состоящая из созданных групп пользователей с разными уровнями привилегий доступа к объектам БД.

### **2.1.6 Требования к программным документам**

Программный продукт будет обеспечен следующими документами:

* спецификация;
* описание программы;
* текст программы;
* руководство программиста;
* руководство оператора.

### **2.1.7 Требования к составу и параметрам технических средств**

Требования, предъявляемые к техническим средствам, необходимым для реализации данного программного продукта:

* процессор с тактовой частотой от 2200 МГц и выше;
* оперативную память от 8 ГБ и выше;
* накопители на жестком диске с достаточным свободным объемом от 30 ГБ и выше.

### **2.1.8 Порядок контроля и приемки программного обеспечения**

Приемка, тестирование и оценка программного обеспечения осуществляется заказчиком в соответствие с требованиями технического задания. Если в процессе тестирования выявлены недостатки, разработчик должен их исправить. В случае успешного тестирования осуществляется приемка программного обеспечения.

### **2.1.9 Требования к маркировке и упаковке**

Программный продукт упаковывается на диск, который хранится в боксе или антистатическом пакете. Маркировка на диске CD-R, CD-RW. И также на переносимом USB-накопителе.

### **2.1.10 Требования к транспортированию и хранению**

Хранение программы осуществляется на диске CD-R, CD-RW, который находится в боксе или в антистатическом пакете, а также на переносимом USB-накопителе.

Если диск какое-то время был на морозе (чего лучше не допускать), перед использованием дать ему несколько минут полежать, что бы его температура сравнялась с комнатной и испарилась сконденсировавшаяся влага.

Также необходимо помнить, что любой пользователь, даже достаточно опытный, может случайно удалить необходимые файлы. Чтобы избежать потери данных, необходимо делать резервные копии файлов, содержащих необходимые данные. Частота, с которой должны выполнятся резервные копии, зависит от объема необходимых данных, содержащихся на диске.

При транспортировке диска, на котором находится программный продукт, необходимо поместить его в специально предназначенный для дисков бокс для того, чтобы избежать попадания частиц пыли на магнитную поверхность диска или царапин, которые могут нанести нежелательный урон, и информация может быть недоступной.

## **2.2 Спецификация**

Таблица 2 – Спецификация

|  |  |  |
| --- | --- | --- |
| Обоснование | Наименование | Примечания |
|  |  |  |
|  | Документация |  |
|  |  |  |
| 046. 06610.54122 –04 12 01 | Текст программы |  |
| 046. 06610.54122 – 04 13 01 | Описание программы |  |
| 046. 06610.54122 – 04 33 01 | Руководство программиста |  |
| 046. 06610.54122 – 04 34 01 | Руководство оператора |  |
|  |  |  |
|  | Компоненты |  |
| 046. 06610.54122 – 04 12 01 | DaliaftransApp.sln |  |
| VoditeliAddPage.xaml |  |
| PostupleniaPage.xaml |  |
| PostupleniaAddPage.xaml |  |
| App.xaml |  |
| MainWindow.xaml |  |
| MainWindow.xaml.cs |  |
|  | RegPage.xaml |  |
|  | RegPage.xaml.cs |  |
|  | VoditeliPage.xaml. |  |
|  | PostupleniaPage.xaml.cs |  |
|  | PostupleniaAddPage.xaml.cs |  |
|  | VoditeliPage.xaml.cs |  |
|  | Manager.cs |  |
|  | VoditeliAdd.xaml.cs |  |

## **2.3 Описание проблемы**

### **2.3.1 Общие сведения**

##### **2.3.1.1 Обозначение и наименование программы**

Наименование разработки: «Информационная система маршрутно-транспортных средств».

Данная программа предназначена для бухгалтеров.

Приложение предназначено для применения в Общество с ограниченной ответственностью «Дальавтотранс», и разработана специально для работников бухгалтерий. Основная задача программы: обеспечить учет и быстрый поиск сведений о поступающих средствах с маршрутов.

##### **2.3.1.2 Программное обеспечение**

Автоматизированная ситема представляет собой базуданных, разработанную с помощью Microsoft SQL Server и написанногоприложенияв среде разработки Visual Studio на платформе WPF (.NET Framework) на языке программирования С#. Для функционирования программы «Автоматизация процессов приемной комиссии колледж» рекомендуется использовать операционную систему Microsoft Windows 7/8/10 и пакет программ Office 2002/2010, а также Microsoft SQL Server.

##### **2.3.1.3 Язык программирования**

Программный продукт разработан с помощью среды разработки Visual Studio на платформе WPF (.NET Framework) на языке программирования С#.

Microsoft Visual Studio – интегрированная среда разработки, разработанная корпорацией Microsoft. Visual Studio позволяет разрабатывать множество различных приложений, среди который можно упомянуть: консольные приложения, игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, WPF, UWP а также веб-сайты, веб-приложения, веб-службы.

Язык программирования C# – объектно-ориентированный язык программирования. Разработан корпорацией Microsoft как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core.

Платформа Windows Presentation Foundation позволяет создавать клиентские приложения для настольных систем Windows с привлекательным пользовательским интерфейсом.

### **2.3.2 Функциональное назначение**

В качестве предметной области рассматривается работа сотрудника бухгатерии.

Основная задача программы: обеспечить учет и быстрый поиск сведений о поступивших начислениях с работы мрашрутов.

### **2.3.3 Описание логической структуры**

При построении баз данных важнейшее значение имеет их проектирование.

Даталогическая модель базы данных может иметь следующий вид, представленный в таблицах 3-5:

Таблица 3 – Структура таблицы «User»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Описание | Размер поля | Индексированное поле |
| IdUser | Числовой | Код пользователя | Длинное целое | Да |
| F | Текстовый | Фамилия | 50 | Нет |
| I | Текстовый | Имя | 50 | Нет |
| O | Текстовый | Отчество | 50 | Нет |
| Login | Текстовый | Логин для входа | 50 | Нет |
| Password | Текстовый | Пароль | 50 | Нет |

Таблица 4 – Структура таблицы «Sredstva»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Описание | Размер поля | Индексированное поле |
| Idzapisi | Числовой | Код транзанкции | Длинное целое | Да |
| IdVoditeli | Числовой | Какой водитель | Длинное целое | Нет |
| IdUser | Числовой | Кто вносил запись | Длинное целое | Нет |
| Money | Денежный | Сумма полученая за маршрут | Нет | Нет |
| Data | Дата | Время вноса информаций | Нет | Нет |

Таблица 5 – Структура таблицы «Voditeli»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Описание | Размер поля | Индексированное поле |
| IdVoditeli | Числовой | Код водителя | Нет | Да |
| F | Текстовый | Фамилия | 50 | Нет |
| I | Текстовый | Имя | 50 | Нет |
| O | Текстовый | Отчество | 50 | Нет |
| AutoPrava | Текстовый | Номер прав | 10 | Нет |

Схема данных, создаваемая базой данных, представлена на рисунке 1:



Рисунок 1 – Схема данных

### **2.3.4 Используемые технические средства**

Требования, предъявляемые к техническим средствам, необходимым для реализации данного программного продукта:

* процессор с тактовой частотой от 2200 МГц;
* оперативную память от 8 ГБ и выше;
* накопители на жестком диске с достаточным свободным объемом от 50 ГБ и выше;
* CD–ROM, CD–RW привод – для установки программы;
* принтер.

### **2.3.5 Вызов и загрузка**

Для инсталляции программы на жесткий диск компьютера необходимо выполнить следующие действия:

* вставить прилагаемый диск в устройство чтения CD-ROM или вставить USB-накопитель в порт компьютера;
* открыть диск устройство чтения с соответствующим именем;
* найти в корневом каталоге диска файл: «start.exe»;
* скопировать его на жесткий диск и cоздать ярлык на рабочем столе.

Запуск программы осуществляется двойным нажатием по ярлыку программы или по файлу «start.exe».

### **2.3.6 Входные данные**

Входными данными являются:

1. данные об водителях, их документах, специальных условиях при необходимости;
2. данные о поступлениях средств;

### **2.3.7 Выходные данные**

Выходными данными являются:

1. Отчёт о поступлениях;

**2.4 Описание программы**

### **2.4.1 Код страницы «VoditeliAddPage.xaml.cs»**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace DaliAftotrans

{

public partial class VoditeliAddPage : Page

{

private Voditeli \_currentVoditel = new Voditeli();

public VoditeliAddPage(Voditeli selectVoditel)

{

InitializeComponent();

if (selectVoditel != null)

\_currentVoditel = selectVoditel;

DataContext = \_currentVoditel;

}

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

{

if (MessageBox.Show("Вы точно хотите сохранить?", "Внимание!!", MessageBoxButton.YesNo) == MessageBoxResult.Yes)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentVoditel.F))

errors.AppendLine("Укажите Фамилию");

if (string.IsNullOrWhiteSpace(\_currentVoditel.I))

errors.AppendLine("Укажите Имя");

if (string.IsNullOrWhiteSpace(\_currentVoditel.AutoPrava))

errors.AppendLine("Укажите Номер водительский удостовирение");

foreach(var item in Autoprava.Text)

if (!char.IsDigit(item))

errors.AppendLine("Водительского удостоверение не может содержать символы");

if(Autoprava.Text.Length!=10)

errors.AppendLine("Длина водительского удостоверения должа быть 10 чисел");

if (\_currentVoditel.IdVoditeli == 0)

DaliAftotransEntities.GetContext().Voditelis.Add(\_currentVoditel);

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString());

return;

}

try

{

DaliAftotransEntities.GetContext().SaveChanges();

MessageBox.Show("Информация сохранена!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

}

}

### **2.4.2 Код страницы «PostupleniaPage.xaml.cs»**

using Excel = Microsoft.Office.Interop.Excel;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Microsoft.Win32;

namespace DaliAftotrans

{

public partial class PostupleniaPage : Page

{

private User \_currentUser = null;

public PostupleniaPage(User user)

{

InitializeComponent();

UpdatePostuplenia();

TextData.DisplayDateEnd = DateTime.Now;

\_currentUser = user;

}

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

{

Manager.MainFrame.Navigate(new PostupleniaAddPage(\_currentUser, null));

}

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

{

var PostupleniaDel = DBGridPostuplenia.SelectedItems.Cast<Sredstva>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {PostupleniaDel.Count()} Элементов ? ", "Внимание",

MessageBoxButton.YesNo) == MessageBoxResult.Yes)

{

try

{

DaliAftotransEntities.GetContext().Sredstvas.RemoveRange(PostupleniaDel);

DaliAftotransEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

UpdatePostuplenia();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

DaliAftotransEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

}

}

private void TBoxSearching\_TextChanged(object sender, TextChangedEventArgs e)

{

UpdatePostuplenia();

}

public void UpdatePostuplenia()

{

DaliAftotransEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

var currentPostuplenia = DaliAftotransEntities.GetContext().Sredstvas.ToList();

if (TBoxSearching.Text != "")

currentPostuplenia = currentPostuplenia.Where(p => p.Voditeli.FIO.ToLower().Contains(TBoxSearching.Text.ToLower())).ToList();

if (TextData.Text != "")

currentPostuplenia = currentPostuplenia.Where(p => p.Data.Equals(TextData.SelectedDate)).ToList();

DBGridPostuplenia.ItemsSource = currentPostuplenia;

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

UpdatePostuplenia();

}

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

{

Manager.MainFrame.Navigate(new VoditeliPage());

}

private void TextData\_DataContextChanged(object sender, SelectionChangedEventArgs e)

{

UpdatePostuplenia();

}

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

{

Manager.MainFrame.Navigate(new PostupleniaAddPage(\_currentUser, (sender as Button).DataContext as Sredstva));

}

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

{

Excel.Application excel = new Excel.Application();

excel.Visible = true;

Excel.Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Excel.Worksheet sheet1 = (Excel.Worksheet)workbook.Sheets[1];

for (int j = 0; j < DBGridPostuplenia.Columns.Count; j++)

{

Excel.Range myRange = (Excel.Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = DBGridPostuplenia.Columns[j].Header;

}

for (int i = 0; i < DBGridPostuplenia.Columns.Count-1; i++)

{

for (int j = 0; j < DBGridPostuplenia.Items.Count; j++)

{

TextBlock b = DBGridPostuplenia.Columns[i].GetCellContent(DBGridPostuplenia.Items[j]) as TextBlock;

Excel.Range myRange = (Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

}

}

### **2.4.3 Код страницы «PostupleniaAddPage.xaml.cs»**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace DaliAftotrans

{

public partial class PostupleniaAddPage : Page

{

private Sredstva \_currentPostuplenia = new Sredstva();

public PostupleniaAddPage(User user, Sredstva selectedPostuplenia)

{

InitializeComponent();

if (selectedPostuplenia != null)

{

\_currentPostuplenia = selectedPostuplenia;

}

DataContext = \_currentPostuplenia;

VoditeliCB.ItemsSource = DaliAftotransEntities.GetContext().Voditelis.ToList();

\_currentPostuplenia.IdUser = user.IdUser;

}

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

{

if (MessageBox.Show("Вы точно хотите сохранить?", "Внимание!!", MessageBoxButton.YesNo) == MessageBoxResult.Yes)

{

StringBuilder errors = new StringBuilder();

if (MoneyText.Text.Any(Char.IsLetter))

errors.AppendLine("В зароботке не должно быть букв");

if (DateText.Text.Any(Char.IsLetter))

errors.AppendLine("В дате не должно быть букв");

if (\_currentPostuplenia.Voditeli == null)

errors.AppendLine("Укажите водителя");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString());

return;

}

if (\_currentPostuplenia.IdZapisi == 0)

{

DaliAftotransEntities.GetContext().Sredstvas.Add(\_currentPostuplenia);

}

try

{

DaliAftotransEntities.GetContext().SaveChanges();

MessageBox.Show("Данные сохранены!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

}

}

### **2.4.4 Код страницы «MainWindow.xaml.cs»**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace DaliAftotrans

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MainFrame.Navigate(new RegPage());

Manager.MainFrame = MainFrame;

}

private void Frame\_ContentRendered(object sender, EventArgs e)

{

if (MainFrame.CanGoBack)

{

BtnBack.Visibility = Visibility.Visible;

}

else

{

BtnBack.Visibility = Visibility.Hidden;

}

}

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

{

Manager.MainFrame.GoBack();

}

}

}

### **2.4.5 Код страницы «RegPage.xaml.cs»**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace DaliAftotrans

{

public partial class RegPage : Page

{

public RegPage()

{

InitializeComponent();

}

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

{

foreach(var user in DaliAftotransEntities.GetContext().Users.ToList())

{

if(UserLogin.Text==user.Login && UserPassword.Text == user.Password)

{

Manager.MainFrame.Navigate(new PostupleniaPage(user));

break;

}

else

{

MessageBox.Show("Неверный логин или пароль");

UserLogin.Text = null;

UserPassword.Text = null;

break;

}

}

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

UserLogin.Text = null;

UserPassword.Text = null;

}

}

}

### **2.4.6 Код страницы «VoditeliPage.xaml.cs»**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace DaliAftotrans

{

public partial class VoditeliPage : Page

{

public VoditeliPage()

{

InitializeComponent();

DBGridVoditel.ItemsSource = DaliAftotransEntities.GetContext().Voditelis.ToList();

}

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

{

Manager.MainFrame.Navigate(new VoditeliAddPage((sender as Button).DataContext as Voditeli));

}

private void Btn\_delete\_Click(object sender, RoutedEventArgs e)

{

var VoditeliDel = DBGridVoditel.SelectedItems.Cast<Voditeli>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {VoditeliDel.Count()} Элементов ? ", "Внимание",

MessageBoxButton.YesNo) == MessageBoxResult.Yes)

{

try

{

DaliAftotransEntities.GetContext().Voditelis.RemoveRange(VoditeliDel);

DaliAftotransEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

DaliAftotransEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DBGridVoditel.ItemsSource = DaliAftotransEntities.GetContext().Voditelis.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void Btn\_add\_Click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.Navigate(new VoditeliAddPage(null));

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

DaliAftotransEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DBGridVoditel.ItemsSource = DaliAftotransEntities.GetContext().Voditelis.ToList();

}

}

}

}

### **2.4.7 Код страницы «Manager.cs»**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace DaliAftotrans

{

class Manager

{

public static Frame MainFrame { get; set; }

}

}

### **2.4.8 Код страницы «App.xaml»**

<Application x:Class="DaliAftotrans.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:DaliAftotrans"

StartupUri="MainWindow.xaml">

<Application.Resources>

<Style TargetType="Button">

<Setter Property="Background" Value="#FFFF99"/>

<Setter Property="FontSize" Value="14"/>

</Style>

</Application.Resources>

</Application>

### **2.4.9 Код страницы «MainWindow.xaml»**

<Window x:Class="DaliAftotrans.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:DaliAftotrans"

mc:Ignorable="d"

Title="Дальавтотранс" Height="600" Width="1000" MinHeight="600" MinWidth="1000">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="26\*"></RowDefinition>

<RowDefinition Height="253\*"></RowDefinition>

</Grid.RowDefinitions>

<Frame Background="#D1FFFF"></Frame>

<Image Source="Resurses/Logo.png" Stretch="Uniform" HorizontalAlignment="Left" Width="57" Height="54" VerticalAlignment="Center"/>

<Button Content="Назад" Name="BtnBack" Click="Button\_Click" Margin="0,0,10,0" Height="24" VerticalAlignment="Center" HorizontalAlignment="Right" Width="119"/>

<Frame Background="AliceBlue" ContentRendered="Frame\_ContentRendered" Name="MainFrame" NavigationUIVisibility="Hidden" Grid.Row="2"/>

</Grid>

</Window>

### **2.4.10 Код окна «PostupleniaAddPage.xaml»**

<Page x:Class="DaliAftotrans.PostupleniaAddPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:DaliAftotrans"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="PostupleniaAddPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<ComboBox Name="VoditeliCB" Grid.Column="1" Margin="5,0,0,0" DisplayMemberPath="FIO" SelectedItem="{Binding Voditeli}" Height="32" VerticalAlignment="Center" HorizontalAlignment="Left" Width="304" />

<TextBox Name="MoneyText" Grid.Column="1" Margin="5,0,0,0" Grid.Row="1" Text="{Binding Money,StringFormat ={}{0:F2}}" HorizontalAlignment="Left" Width="223" Height="24" VerticalAlignment="Center"/>

<DatePicker Name="DateText" Grid.Column="1" Grid.Row="2" Height="26" SelectedDate="{Binding Data}" VerticalAlignment="Center" HorizontalAlignment="Left" Width="223" Margin="5,0,0,0">

</DatePicker>

<TextBlock Text="Водитель" Height="20" VerticalAlignment="Center" HorizontalAlignment="Right" Width="89" />

<TextBlock Grid.Row="1" Text="Заработок" Height="22" VerticalAlignment="Center" HorizontalAlignment="Right" Width="89"/>

<TextBlock Grid.Row="2" Text="Дата вноса" Height="22" VerticalAlignment="Center" HorizontalAlignment="Right" Width="89"/>

<Button Content="Сохранить" Grid.Row="3" HorizontalAlignment="Center" Height="40" Width="244" Click="Button\_Click" Margin="0,0,0,10" Grid.ColumnSpan="2" VerticalAlignment="Bottom"/>

</Grid>

</Page>

### **2.4.11 Код файла «PostupleniaPage.xaml»**

<Page x:Class="DaliAftotrans.PostupleniaPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:DaliAftotrans"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="PostupleniaPage" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="37\*"/>

<RowDefinition Height="430\*"/>

<RowDefinition Height="37\*"/>

</Grid.RowDefinitions>

<Button Content="Водители" FontSize="14" Height="23" VerticalAlignment="Top" HorizontalAlignment="Left" Width="96" Margin="5,4,0,0" Click="Button\_Click"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Center" Height="35" Width="710">

<StackPanel Orientation="Horizontal">

<TextBlock Text="Поиск Водителей: " Width="175" Height="19" TextAlignment="Right"/>

<TextBox Width="175" x:Name="TBoxSearching" TextChanged="TBoxSearching\_TextChanged" Height="25"/>

</StackPanel>

<StackPanel Orientation="Vertical">

<StackPanel Orientation="Horizontal" Margin="5">

<TextBlock Text="Поиск по дате: " Width="175" Height="19" TextAlignment="Right"/>

<DatePicker Width="175" Height="25" x:Name="TextData" SelectedDateChanged="TextData\_DataContextChanged"/>

</StackPanel>

</StackPanel>

</StackPanel>

<DataGrid x:Name="DBGridPostuplenia" AutoGenerateColumns="False" IsReadOnly="False" Grid.Row="1">

<DataGrid.Columns>

<DataGridTextColumn Header="ФИО-Водителя" Width="\*" Binding="{Binding Voditeli.FIO}" IsReadOnly="True"/>

<DataGridTextColumn Header="Заработок" Width="\*" Binding="{Binding Money,StringFormat ={}{0:F2}}" IsReadOnly="True"/>

<DataGridTextColumn Header="Дата" Width="\*" Binding="{Binding Data,StringFormat={}{0:dd.MM.yyyy}}" IsReadOnly="True"/>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Name="Redact" Click="Redact\_Click" Content="Редактировать" />

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Grid.Row="3" Name="BtnAddPostuplenia" Content="Добавить поступления" Height="23" Width="162" Click="ButtonAddPostuplenia\_Click" Margin="186,7,452,7"/>

<Button Grid.Row="3" Name="BtnDeletePostuplenia" Content="Удалить поступления" Height="23" Width="162" Click="BtnDeletePostuplenia\_Click" Margin="467,7,171,7" />

<Button Grid.Row="3" Name="BtnOtch" Content="Отчёт" Click="BtnOtch\_Click" Height="23" Margin="25,7,0,7" HorizontalAlignment="Left" Width="96" />

</Grid>

</Page>

**2.4.12 Код файла «RegPage.xaml»**

<Page x:Class="DaliAftotrans.RegPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:DaliAftotrans"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="RegPage" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid VerticalAlignment="Center" HorizontalAlignment="Center" Height="185" Width="244">

<Grid.RowDefinitions>

<RowDefinition Height="59\*"/>

<RowDefinition Height="56\*"/>

<RowDefinition Height="39\*"/>

</Grid.RowDefinitions>

<StackPanel VerticalAlignment="Center" HorizontalAlignment="Center" Height="64" Width="246">

<TextBlock Text="Логин" FontSize="15"/>

<TextBox Name="UserLogin" Width="198" Height="32" FontSize="20"/>

</StackPanel>

<StackPanel Grid.Row="1" VerticalAlignment="Center" HorizontalAlignment="Center" Height="64" Width="246">

<TextBlock Text="Пароль" FontSize="15"/>

<TextBox Name="UserPassword" Width="198" Height="32" FontSize="20" />

</StackPanel>

<Button Content="Войти" x:Name="BtnLogIn" Click="BtnLogIn\_Click" Grid.Row="2" Margin="18,9,18,9" />

</Grid>

</Page>

**2.4.13 Код файла «VoditeliAddPage.xaml»**

<Page x:Class="DaliAftotrans.VoditeliAddPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:DaliAftotrans"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="VoditeliPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<TextBox Name="Autoprava" Grid.Column="1" Margin="10,0,0,0" Grid.Row="1" Text="{Binding AutoPrava}" Height="30" VerticalAlignment="Center" HorizontalAlignment="Left" Width="324" RenderTransformOrigin="0.5,0.5"/>

<TextBlock Height="18" VerticalAlignment="Center" RenderTransformOrigin="0.5,0.5" HorizontalAlignment="Right" Width="162" >

<Run Language="ru-ru" Text="ФИО"/></TextBlock>

<TextBlock Grid.Row="1" Height="18" VerticalAlignment="Center" HorizontalAlignment="Right" Width="162"><Run Language="ru-ru" Text="Номер прав"/></TextBlock>

<Button Content="Сохранить" VerticalAlignment="Bottom" Height="40" Grid.Row="2" Margin="0,97,0,10" HorizontalAlignment="Center" Width="244" Grid.ColumnSpan="2" Click="Button\_Click" />

<TextBox x:Name="FText" Margin="10,0,0,0" Text="{Binding F}" Grid.Column="1" Height="30" VerticalAlignment="Center" HorizontalAlignment="Left" Width="112"/>

<TextBox x:Name="IText" Margin="144,0,0,0" Text="{Binding I}" Grid.Column="1" Height="30" VerticalAlignment="Center" HorizontalAlignment="Left" Width="112"/>

<TextBox x:Name="OText" Margin="278,0,0,0" Text="{Binding O}" Grid.Column="1" Height="30" VerticalAlignment="Center" HorizontalAlignment="Left" Width="112"/>

</Grid>

</Page>

**2.4.11 Код файла «VoditeliPage.xaml»**

<Page x:Class="DaliAftotrans.VoditeliPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:DaliAftotrans"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="VoditeliPage" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="418\*"/>

<RowDefinition Height="37\*"/>

</Grid.RowDefinitions>

<DataGrid x:Name="DBGridVoditel" AutoGenerateColumns="False" IsReadOnly="False" Grid.Row="0" Margin="0,0,0,1">

<DataGrid.Columns>

<DataGridTemplateColumn Header="ФИО-Водителя" Width="\*">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Text="{Binding FIO}"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Номер прав водителя" Width="\*">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Text="{Binding AutoPrava}"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Name="Redact" Click="Redact\_Click" Content="Редактировать" />

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Height="23" Width="162" Name="Btn\_add" Content="Добавить" Margin="186,7,452,7" Click="Btn\_add\_Click" Grid.Row="1"/>

<Button Height="23" Width="162" Name="Btn\_delete" Click="Btn\_delete\_Click" Content="Удалить" Margin="467,7,171,7" Grid.Row="1"/>

</Grid>

</Page>

## **2.5 Руководства программиста**

### **2.5.1 Общие положения**

Данная программа является средством учета поступающих средств.

Приложение предназначено для применения в Общество с ограниченной ответственностью «Дальавтотранс», и разработана специально для работников бухгалтерий. Основная задача программы: обеспечить учет и быстрый поиск сведений о поступающих средствах с маршрутов.

Основная задача программы: обеспечить учет и быстрый поиск сведений об средствах, а также генерацию отчётов по рейтингу абитуриентов.

Эти возможности реализованы при помощи стэка Microsoft SQL Server 2019, Visual Studio 2019, на платформе WPF (.NET Framework) при помощи языка программирования C#.

### **2.5.2 Назначение и цель создания**

Целью дипломного проекта является разработка программного обеспечения, необходимого для автоматизации процессов бухгалтерий.

Данное приложение должно обеспечить эффективную работу с данными сотрудников бухгалтерии.

### **2.5.3 Характеристики программы**

Автоматизированная ситема представляет собой базу данных, разработанную с помощью Microsoft SQL Server и написанного приложения в среде разработки Visual Studio на платформе WPF (.NET Framework) на языке программирования С#. Для функционирования программы «Информационная система маршрутно-транспортных средств» рекомендуется использовать операционную систему Microsoft Windows 7/8/10 и пакет программ Office 2002/2010, а также Microsoft SQL Server.

Рекомендуемые требования к техническому оборудованию:

* операционная система: Microsoft Windows 7/10/Server 2016 (32 или 64 bit);
* процессор: двухъядерный Intel Core i3 M350 2.27 ~ 2.3 Ghz;
* оперативная память (ОЗУ): 8 ГБ и выше;
* HDD or SSD: от 50 ГБ.

### **2.5.4 Обращение к программе**

Прежде, чем установить и запустить в работу программный продукт «start.exe» на компьютер следует установить требуемое прикладное программное обеспечение. При отсутствии необходимого программного приложения и операционной системы работа с программой будет невозможна.

Запуск программы осуществляется двойным нажатием по файлу «start.exe».

Следует исключить перепады напряжения в электросети, к которой непосредственно подсоединен компьютер. В случае сбоя в работе следует загрузить программу заново.

Для исключения несанкционированного доступа к данным, хранимым в базе данных, реализованна защита, состоящая из созданных групп пользователей с разными уровнями привилегий доступа к объектам БД.

### **2.5.5 Входные и выходные данные**

Входными данными являются:

1. данные об водителях, их документах, специальных условиях при необходимости;
2. данные о поступлениях средств;

Выходными данными являются:

1. Отчёт о поступлениях;

### **2.5.6 Возможности модификации программы**

Для того чтобы изменять и дополнять программный код необходима СУБД Microsoft SQL Server 2019, среда программирования Visual Studio 2019 и файлы проекта «Daliaftotrans».

Таблица 17 – Файлы проекта

|  |  |
| --- | --- |
| Наименование | Описание |
| DaliAftotransApp.sln  App.config | Объектные файлы проекта |
| App.xaml  MainWindow.xaml  RegPage.xaml  PostupleniaPage.xaml  PostupleniaAddPage.xaml  VoditeliPage.xaml  VoditeliAddPage.xaml | Файлы описания форм и компонентов на формах |
| MainWindow.xaml.cs  App.xaml.cs  Manager.cs  RegPage.xaml.cs  PostupleniaPage.xaml.cs  PostupleniaAddPage.xaml.cs  VoditeliPage.xaml.cs  VoditeliAddPage.xaml.cs | Программные модули |
| Daliaftotrans.edmx | Базы данных |

## **2.6 Руководства оператора**

### **2.6.1 Назначение программы**

Данная программа является средством поступающих учета средств.

Приложение предназначено для применения в Общество с ограниченной ответственностью «Дальавтотранс», и разработана специально для работников бухгалтерий.

Основная задача программы: обеспечить учет и быстрый поиск сведений о поступающих средствах с маршрутов.

Стандартный графический интерфейс, удобное пользовательское меню позволяет быстро разобраться в работе с программой и не предполагает наличия специальных навыков для работы с базой.

Пользовательское меню позволяет быстро достичь результата в поиске и редактировании информации. В случае если вопросы все-таки возникнут, в комплексе с проектом поставляется руководство пользователя, в котором можно найти ответы на интересующие вопросы.

Предложенная система значительно упрощает труд сотрудников бухгалтерии. Эффективность программы позволяет сократить время на заполнение данных, избавляет от рутинной работы, существенно уменьшает количество возможных ошибок.

### **2.6.2 Условия выполнения программы**

Автоматизированная ситема представляет собой базу данных, разработанную с помощью Microsoft SQL Server и написанного приложения в среде разработки Visual Studio на платформе WPF (.NET Framework) на языке программирования С#. Для функционирования программы «Информационная система маршрутно-транспортных средств» рекомендуется использовать операционную систему Microsoft Windows 7/8/10 и пакет программ Office 2002/2010, а также Microsoft SQL Server.

Рекомендуемые требования к техническому оборудованию:

* операционная система: Microsoft Windows 7/10/Server 2016 (32 или 64 bit);
* процессор: двухъядерный Intel Core i3 M350 2.27 ~ 2.3 Ghz;
* оперативная память (ОЗУ): 8 ГБ и выше;
* HDD or SSD: от 50 ГБ.

### **2.6.3 Выполнение программы**

Запуск программы осуществляется двойным щелчком по файлу «start.exe».

##### **2.6.3.1 Окно «Авторизация»**

После запуска программы, открывается окно с авторизацией пользователя, изображённое на рисунке 2. В базе данных предусмотрено две учетные записи: для сотрудника бухаглтерии и ответственного секретаря. Чтобы войти в аккаунт, необходимо в одноимённые поля ввести логин и пароль, после чего нажать на кнопку «Войти». Если пользователь ввёл правильные данные, то будет произведён вход в аккаунт и откроется главное окно программы.

В окне авторизации предусмотрены проверки на ошибки. Если пользователь какое-либо из полей оставит пустым, то система его предупредит об этом событии и не позволит пройти дальше, пока проблема не будет решена. Также, пользователю будет выдана ошибка в том случае, если он введёт неверные данные.

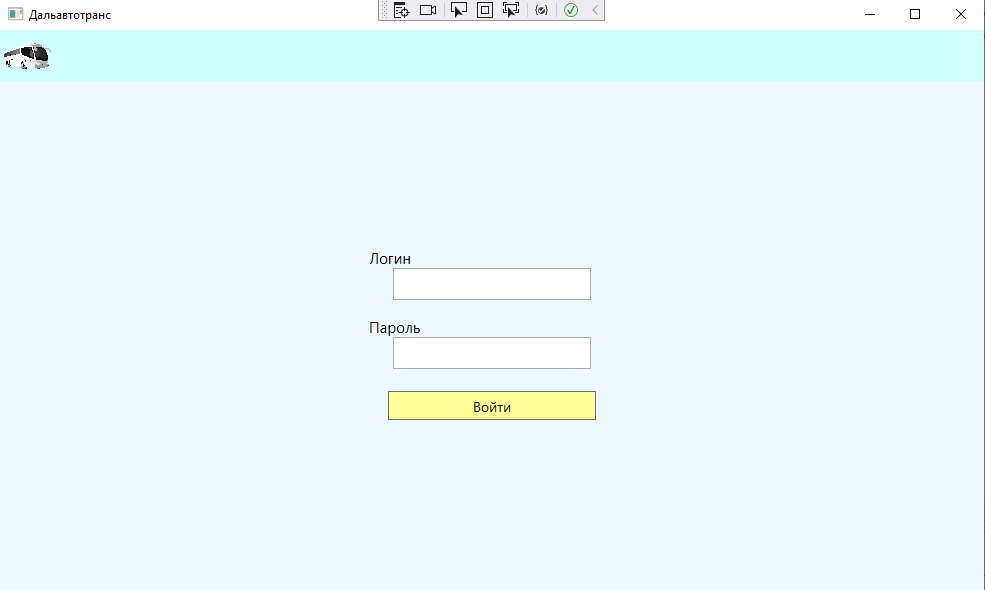


Рисунок 2 – Окно «Авторизация»

##### **2.6.3.2 Главное окно**

После успешного входа в аккаунт, программа открывает пользователю главное окно, изображённое на рисунке 3, с помощью которого можно воздействовать с основным функционалом системы. Сверху программы можно увидеть кнопку «Назад» что позволит вернуться на прошлую страницу. Снизу же расположены кнопки «Добавить поступления» «Удалить поступления» с помощью которых можно выполнять основные задачи программы и кнопка «Отчёт» для автоматического составления отчёта из найденых с помощью полей поиска: «Поиск Водителей» и «Поиск по дате». Так же при нажатии возле записи кнопки «Редактировать» откроется форма для редактирования записи. Для удаления записи требуется выделить курсором мышы нужные записи, после чего нажать на «Удаление поступления» и нажать на да в окне о подвержения действия.

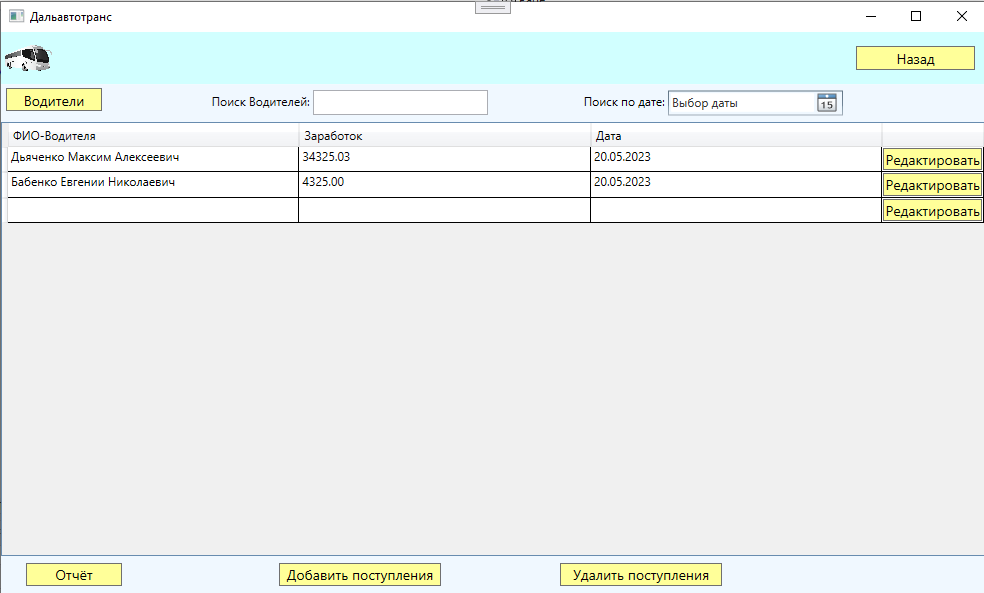
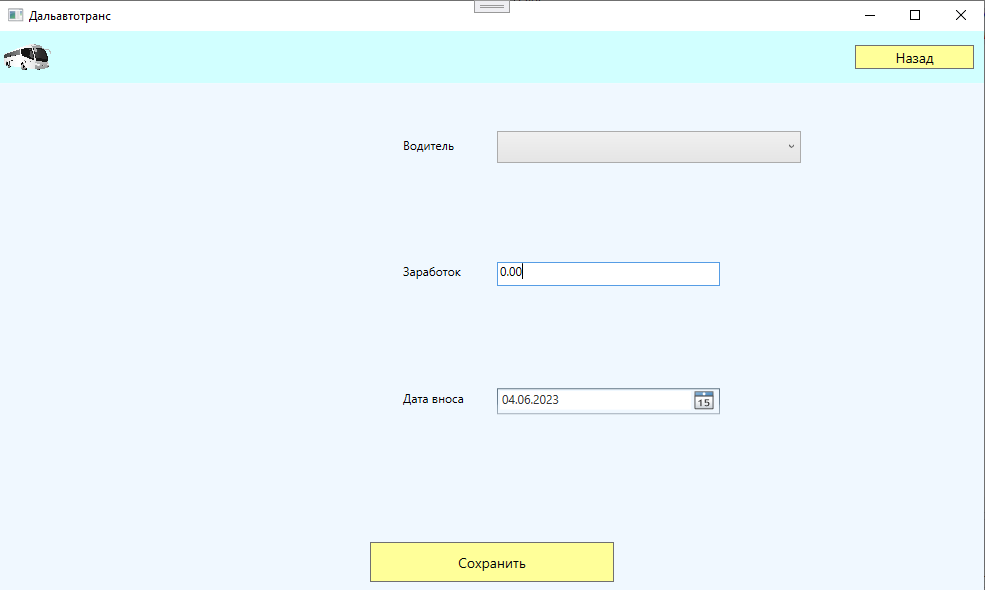


Рисунок 3 – Главное окно

##### **2.6.3.3 Страница «Добавления, редактирования Поступления»**

Кливкнув на кнопку «Добавить поступление» или с помощью редактирование, открывается специальная форма, которая изображена на рисунке 4, в которой есть специальные поля для заполнения информации. После заполнения информации надо нажать «Сохранение» выпадет сообщение об сохранение.

 Рисунок 4 – Окно для работы с записями «Поступления»

##### **2.6.3.4 Страница «Водители»**

Нажав по кнопке, «Водители» на главном окне приложения, откроется окно, которое представлено на рисунке 5, с хранящиемся в базе данных водителей автобуса. Функционал схож с окном «Поступления», где есть кнопки «Добавить», «Удалить», «Редактировать». При нажатии на кнопки «Добавить» или «Редактировать» откроется окно, которое представлено на рисунке 6, для работы с записями. После завершения изменений требуестя нажать на кнопку «Сохранить» и подтвердить изменения.

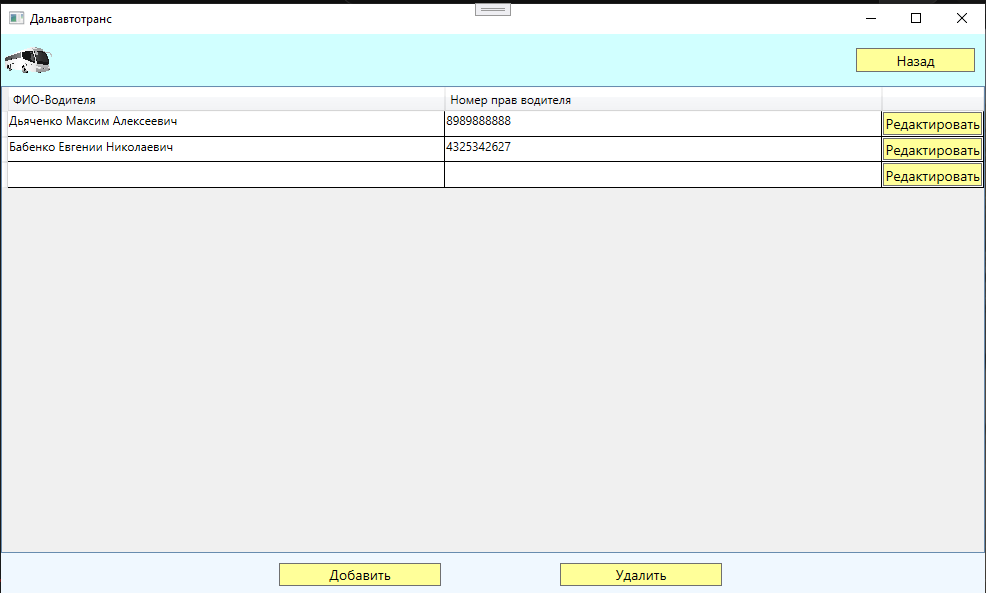


Рисунок 5 – Окно «Водители»

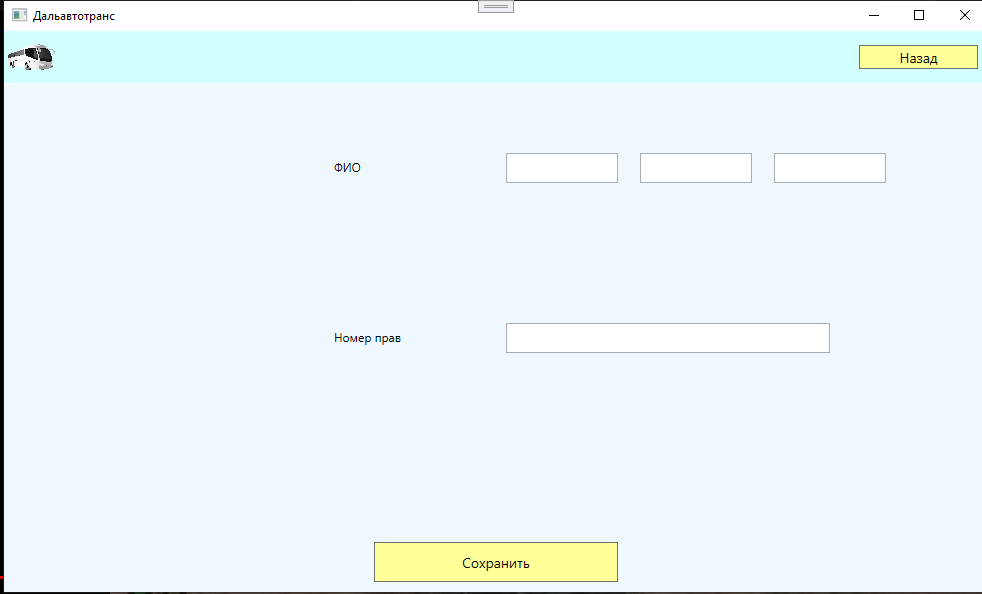


Рисунок 4 – Окно для работы с записями «Водители»

**2.7 Тестирование программного модуля**

Объект испытаний.

Объектом испытаний является программный продукт «Информационная система маршрутно-транспортных средств», предназначенный для обеспечения лучшей работы бухгалтерии предприятия ООО «Дальавтотранс».

Цель испытаний.

Проверка надёжности функционирования программного модуля.

Требования к программе.

Работа модуля не должна приводить к сбою системы. Программа должна выполнять эффективную обработку данных и их хранение.

Средства и порядок испытаний.

Технические средства, которые были использованы во время испытаний:

* процессор Intel Pentium 2,4 ГГц и выше;
* оперативная память 8192 Мб и выше;
* жесткий диск 50 ГБ и выше;
* монитор;
* CD–ROM, CD–RW привод – для установки программы.

Для проведения испытаний используется Release версия программного модуля.

# 

# **ЗАКЛЮЧЕНИЕ**

В настоящее время развитие общественых транспортных средств являестя важной деятельностью любого города, поскольку увиличение качества обслуживание пасажиров, а также увиличение производительности предприятия имеют долго играющие последствия в виде уменьшиение опасности и нагружености дорог, за счёт уменьшающейся потребности в личном транспорте и оптимизации работы города в виде уменьшения задержок прохождения маршрутов, что позволит людам быстрее добираться до мест назначения.

Целью выполнения дипломного проекта является разработка программного обеспечения, необходимого для автоматизации процессов бухгалтерии предприятия.

В настоящее время управление любой деятельностью невозможно без анализа и обработки информации. Данное приложение обеспечит простой и быстрый доступ к любым данным, хранящимся в базе.

Данная задача была реализована с использованием следующих вещей: база данных спроектирована с помощью Microsoft SQL Server, доступ к данным реализован через технологию ADO.NET и LINQ, использовалась среда разработки Visaul Studio, в качестве конструктора графического интерфейса выступала платформе WPF (.NET Framework), язык программирования, используем для реализации всей логики программы - язык C#.

Таким образом, цель дипломного проекта была достигнута, программный продукт оценен заказчиком по его требованиям и запущен в работу.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ГОСТ 19.101–77. Виды программ и программных документов [текст] – Введ. 01.01.80. – М.: Изд–во стандартов, 1987. – 4 с.
2. ГОСТ 19.104–77. Основные надписи [текст]. – Введ. 01.01.80. – М.: Изд–во стандартов, 1987. – 7 с.
3. ГОСТ 19.105–77. Общие требования к программным документам [текст] – Введ. 01.01.80. – М.: Изд–во стандартов, 1987. – 2 с.
4. ГОСТ 19.106–77. Требования к программным документам, выполненным печатным способом [текст] – Введ. 01.01.80. – М.: Изд–во стандартов, 1987. – 13 с.
5. ГОСТ 19.201–78. Техническое задание. Требования к содержанию и оформлению [текст] – Введ. 01.01.80. – М.: Изд–во стандартов, 1987. – 3 с.
6. ГОСТ 19.202–78. Спецификация [текст] – Введ. 01.01.80. – М.: Изд–во стандартов, 1987. – 3 с.
7. ГОСТ 19.301–79. Программа и методика испытаний [текст] – Введ. 01.01.80. – М.: Изд–во стандартов, 1989. – 2 с.
8. ГОСТ 19.401–78. Текст программы [текст] – Введ. 01.01.80. – М.: Изд–во стандартов, 1987. – 1 с.
9. ГОСТ 19.402–78. Описание программы [текст] – Введ. 01.01.80. – М.: Изд–во стандартов, 1987. – 2 с.
10. ГОСТ 19.404–79. Пояснительная записка. Требования к содержанию и оформлению [текст] – Введ. 01.01.80. – М.: Изд–во стандартов, 1987. – 2 с.
11. ГОСТ 19.502–78. Описание применения. Требования к содержанию и оформлению [текст] – Введ. 01.01.80. – М.: Изд–во стандартов, 1987. – 2 с.
12. ГОСТ 19.504–79. Руководство программиста. Требования к содержанию и оформлению [текст] – Введ. 01.01.80. – М.: Изд–во стандартов, 1987. – 2 с.
13. ГОСТ 19.505–79. Руководство оператора. Требования к содержанию и оформлению [текст] – Введ. 01.01.80. – М.: Изд–во стандартов, 1987. – 2 с.
14. Басс, Л. Архитектура программного обеспечения на практике [текст] / Л. Басс, П. Клементс, Р. Кацман. – 2-е изд. – СПб.: Питер, 2006. – 575 с.
15. Брауде, Эрик Дж. Технология разработки программного обеспечения [текст]/ Э. Д. Брауде. – СПб.: Питер, 2004. – 655 с.
16. Гагарина, Л.Г. Технология разработки программного обеспечения: учебное пособие [текст] / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул. – М: Форум Инфра-М, 2013. – 400 с.
17. Гниденко, И. Г. Технология разработки программного обеспечения: учеб. пособие для СПО [текст]/ И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М. : Издательство Юрайт, 2017. — 235 с.
18. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2: учебник для вузов[текст] / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2019. — 501 с.
19. Круз, Р.Л. Структуры данных и проектирование программ: пер. с англ. [текст]/ Р. Л. Круз – Москва: БИНОМ. Лаборатория знаний, 2008. – 765 с.
20. Орлов, С. А. Технологии разработки программного обеспечения. Современный курс по программной инженерии: учебник [текст] / С. А. Орлов, Б. Я. Цилькер. – 4-е изд. – СПб.: Питер, 2012. – 608 с.
21. Рудаков, А.В. Технология разработки программных продуктов: Учеб. пособие для студентов средних профессиональных учебных заведений [текст] А.В. Рудаков – М.: «Академия», 2017.– 192 с.
22. Советов, Б. Я. Базы данных: учебник для прикладного бакалавриата [текст]/ Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 3-е изд., перераб. и доп. — М.: Издательство Юрайт, 2019. — 420 с.
23. Федорова, Г.Н., Участие в интеграции программных модулей: Учеб. пособие для студентов средних профессиональных учебных заведений [текст]/Г.Н. Федорова – М.: «Академия», 2016. – 302 с.