УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 2-40 01 01

Учебная группа ПО-455

**ДИПЛОМНЫЙ ПРОЕКТ**

**РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ**

**«ПОСТРОЕНИЕ ЖЕЛЕЗНОДОРОЖНЫХ МАРШРУТОВ»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**ДП ПО.00.00.000 ПЗ**

Разработал A.О.Савич

Руководитель проекта А.Д. Базанова

Консультант

экономического раздела Е.А. Шапчиц

Нормоконтролер Е.А. Сенькевич

2024

**Содержание**

Введение 4

1. Постановка задачи 5
   1. Описание предметной области 5
   2. Актуальность решаемой задачи 5
   3. Характеристика решаемой задачи 7
2. Проектирование программного продукта 9
   1. Разработка модели данных 9
   2. Выбор программного обеспечения 14
   3. Определение требований к техническим средствам 14
   4. Защита информации 16
3. Разработка программного обеспечения 17
   1. Определение формы представления входных и выходных

данных 22

* 1. Тестирование программного модуля 22
  2. Разработка справочной системы 23
  3. Описание разрабатываемого программного продукта 24

1. Экономическая часть 28

5 Охрана труда 36

6 Энерго- и ресурсосбережение 46

Заключение 49

Список используемых источников 50

Приложение А (обязательное) Входные и выходные

формы данных программы 51

Приложение Б (обязательное) Текст программы 55

Приложение В (обязательное) Результаты тестирования 119

Приложение Г (обязательное) Описание программы 122

Приложение Д (обязательное) Документация пользователя 125

Приложение Е (обязательное) Расчет затрат на оплату труда и

отчислений на социальные нужды 136

**Введение**

**За последние десятилетия, программное обеспечение претерпело огромные изменения, пройдя путь от программ, которые способны выполнять лишь простейшие логические и арифметические операции, до сложных систем управления организациями. На сегодняшний день управление организацией без компьютера просто немыслимо. Для принятия какого-либо решения в условиях неопределенности и риска нужно регулярно держать под контролем разные аспекты финансово-хозяйственной деятельности. На современном этапе развития информатизации и автоматизации характерно применение распределенной обработки информации. В качестве более эффективной и перспективной сферы применения концепции распределенной обработки информации выступает автоматизация планово-управленческих функций на основе персональных компьютеров, которые установлены непосредственно на рабочем месте того или иного специалиста. Системы, которые выполняют данные функции, наделены широким распространением, которые называются автоматизированные рабочие места (АРМ).**

Распределенная обработка информации и автоматизация планово-управленческих функций на основе персональных компьютеров позволяют создать эффективные автоматизированные рабочие места (АРМ) для специалистов, занимающихся построением железнодорожных маршрутов.

Целью данного дипломного проекта является разработка автоматизированной системы «Построение железнодорожных маршрутов», которая будет управлять базой данных маршрутов. Реализация базы данных может быть осуществлена через:

* использование существующих систем управления базами данных (СУБД)
* создание собственной структурированной базы данных.

Преимущества использования СУБД включают централизованное хранение информации и встроенные алгоритмы обработки данных.

Программа будет способствовать автоматизации процессов планирования и управления маршрутами, повышая эффективность работы железнодорожного транспорта и удовлетворяя современным требованиям к точности и скорости обработки информации.

**1. Постановка задачи.**

**1.1 Описание предметной области.**

Услуги по перевозке пассажиров в пределах определенного региона должны обеспечиваться железнодорожным вокзалом. Для этого организация должна иметь в штате машинистов, кассиров, системных администраторов (при необходимости), бухгалтеров, юристов, механиков, проводников, офис-менеджеров и т.д. Также необходим парк поездов. Основные клиенты вокзала - жители данного региона.

**Депо** - это территория с комплексом зданий, сооружений и устройств для стоянки, технического обслуживания и текущего ремонта подвижного состава железнодорожного транспорта.

Любая железнодорожная компания, заботящаяся о своей репутации на рынке перевозок, содержит большой парк поездов и старается регулярно обновлять модельный ряд локомотивов и вагонов. От качества подвижного состава напрямую зависит качество услуг, предоставляемых клиентам. Поезд, который имеет технические проблемы, может остановиться в самый неожиданный момент, что неизбежно приведет к финансовым убыткам. Парк поездов любой компании обычно состоит из тех локомотивов и вагонов, которые наиболее востребованы на рынке. Управление парком поездов имеет несколько ключевых аспектов:

– отслеживание местоположения подвижного состава

– управление расписанием движения поездов

– управление расходом топлива

– управление работой машинистов.

Компания, инвестирующая в систему управления железнодорожными перевозками, может получить следующие преимущества:

– **снижение затрат** (функции планирования, выполнения перевозок, логистика, отслеживание движения поездов, отчеты и анализ все это может привести к сокращению затрат)

– **улучшенное обслуживание клиентов и подотчетность** (выбор подходящего перевозчика, а также улучшенное управление логистикой и отчетами могут повысить качество обслуживания клиентов и помочь компании быть более ответственной)

– **улучшенный финансовый менеджмент**(улучшенное планирование маршрутов, оптимизация загрузки, лучшие выбранные перевозчики могут обеспечить лучшее управление финансами)

– **гибкость** (системы управления железнодорожными перевозками обеспечивают гибкость при изменении планов доставки)

– **простота установки и использования** (большинство систем управления железнодорожными перевозками имеют простую установку и должны быть удобными для пользователя)

Использование системы управления железнодорожным парком позволяет компании получать обработанную информацию, которая может использоваться для организационных решений по управлению парком поездов и ресурсами компании.

Эффективная работа железнодорожной компании зависит от оперативности и рациональности выполняемых действий. В современном мире для компании очень важно быть конкурентоспособной, особенно в сфере предоставления транспортных услуг. От качества услуг напрямую зависит имидж компании среди клиентов и ее авторитет на рынке. Транспортные услуги являются одной из ключевых сфер обслуживания, поскольку ежедневно в мире перевозятся миллионы тонн грузов и сотни тысяч пассажиров.

Разрабатываемая программа позволит упростить работу сотрудников железнодорожного депо, а также учет информации о продажах билетов, техническом обслуживании поездов, сотрудниках депо, расписании движения поездов.

**1.2 Актуальнность решаемой задачи.**

Актуальность задачи по разработке системы управления железнодорожным парком обусловлена необходимостью повышения эффективности работы железнодорожных вокзалов и депо. Система позволит оптимизировать работу персонала, улучшить качество обслуживания клиентов и повысить надежность транспортных услуг. Внедрение такой системы способствует снижению операционных затрат, улучшению финансового менеджмента и повышению гибкости в управлении расписанием и ресурсами. Это особенно важно для поддержания конкурентоспособности на рынке транспортных услуг, где качество и надежность являются ключевыми факторами успеха.

**1.3 Характеристика решаемой задачи.**

В результате выполнения дипломного проекта требуется разработать программу, позволяющую оперативно учитывать следующую информацию: о продажах билетов, техническом обслуживании поездов, сотрудниках депо, расписании движения поездов.

В программе также присутствует функционал создания и экспорта отчетов. В своей основе программа имеет базу данных Microsoft Sql Server.

Список функций:

* просмотр отчетов;
* экспорт простых и сложных отчетов;
* просмотр и занесение информации о продажах билетов;
* просмотр и занесение информации о техническом обслуживании поездов;
* просмотр и занесение информации о сотрудниках железной дороги;
* просмотр и занесение информации о маршрутах поездов.

Значительным преимуществом автоматизации данных операций является скорость их выполнения, возможность в любой момент осуществить запрос к данным, находящимся в базе и получить по этому запросу исчерпывающую информацию. Программа позволит отказаться от необходимости ведения бумажных архивов.

**2. Проектирование программного продукта**

**2.1 Разработка модели данных**

Все действия и данные, производимые программой, можно показать с помощью диаграмм UML. С помощью диаграммы вариантов использования проектируемая система представляется в виде множества сущностей или актёров, взаимодействующих с системой с помощью вариантов использования. Диаграмма вариантов использования – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

Диаграмма вариантов использования приведена на рисунке 1.

**Изображение выглядит как диаграмма, круг, линия, текст

Автоматически созданное описание**

**Рисунок 1 – Диаграмма вариантов использования**

Основным средством для предоставления статических моделей являются диаграммы классов.

Статические модели обеспечивают представление структуры систем в терминах базовых строительных блоков и отношений между ними. «Статичность» этих моделей состоит в том, что здесь не показывается динамика изменений системы во времени. Вместе с тем, эти модели несут в себе не только структурные описания, но и описания операций, реализующих заданное поведение системы.

Вершины диаграммы классов нагружены классами, а дуги (ребра) – отношениями между ними. На рисунке 2 представлена диаграмма классов

**Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание**

**Рисунок 2 – Диаграмма классов**

Диаграмма схем состояний – одна из пяти диаграмм UML, моделирующих динамику систем. Диаграмма схем состояний отображает конечный автомат, выделяя поток управления, следующий от состояния к состоянию. Конечный автомат – поведение, которое определяет последовательность состояний в ходе существования объекта. Эта последовательность рассматривается как ответ на события и включает реакции на эти события. Диаграмма состояния представлена на рисунке 3.

**Изображение выглядит как текст, снимок экрана, линия, диаграмма

Автоматически созданное описание**

**Рисунок 3 – Диаграмма состояния**

Диаграмма последовательности отображает взаимодействие объектов, упорядоченное во времени. На ней показаны объекты и классы, используемые в сценарии, и последовательность сообщений, которыми обмениваются объекты, для выполнения сценария.

Диаграмма последовательности для формы **информации о сотрудниках** представленна на рисунке 4.

****

**Рисунок 4 – Диаграмма последовательности для формы информации о сотрудниках**

Диаграмма последовательности для формы **продажи билетов** представленна на рисунке 5



Рисунок 5– Диаграмма последовательности для формы продажи билетов

Диаграмма последовательности для формы **информации о состоянии поездов** представленна на рисунке 6



Рисунок 6 – Диаграмма последовательности для формы информации о состоянии поездов

Диаграмма последовательности для формы **информации о промежуточных пунктах маршрутов** представленна на рисунке 7



Рисунок 7 – Диаграмма последовательности для формы информации о промежуточных пунктах маршрутов

Диаграмма последовательности для формы **информации о проданных билетах** на рисунке 8



Рисунок 8 – Диаграмма последовательности для формы информации о проданных билетах

Диаграмма последовательности для формы **информации о маршрутах** представленна на рисунке 9



Рисунок 9 – Диаграмма последовательности для формы информации о маршрутах

Диаграммы компонентов показывают, как выглядит модель на физическом уровне. На них изображены компоненты программного обеспечения и связи между ними. При этом на такой диаграмме выделяют два типа компонентов: исполняемые компоненты и библиотеки кода.

Каждый класс модели (или подсистема) преобразуется в компонент исходного кода. После создания они сразу добавляются к диаграмме компонентов. Между отдельными компонентами изображают зависимости, соответствующие зависимостям на этапе компиляции или выполнения программы.

Диаграмма компонентов представлена на рисунке 10.



Рисунок 10 – Диаграмма компонентов

Диаграмма развертывания предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения (runtime). При этом представляются только компоненты-экземпляры программы, являющиеся исполнимыми файлами или динамическими библиотеками. Те компоненты, которые не используются на этапе исполнения, на диаграмме развертывания не показываются. Так, компоненты с исходными текстами программ могут присутствовать только на диаграмме компонентов. На диаграмме развертывания они не указываются.

Цели, преследуемые при разработке диаграммы развертывания:

определить распределение компонентов системы по ее физическим узлам;

показать физические связи между всеми узлами реализации системы на этапе ее исполнения;

выявить узкие места системы и реконфигурировать ее топологию для достижения требуемой производительности.

Диаграмма развертывания представлена на рисунке 11.



Рисунок 11 – Диаграмма развертывания

**2.2 Выбор программного обеспечения**

Интегрированная среда разработки (IDE) - это многофункциональная программа, которая поддерживает многие аспекты разработки программного обеспечения. Интегрированная среда разработки Visual Studio - это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для улучшения процесса **разработки. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).**

**Система Microsoft Visual Studio является самой сложной интегрированной средой разработки (integrated development enviroment - IDE), доступной для программистов в настоящее время. Она является результатом долгой истории развития языков программирования и интерфейсов и вобрала в себя достижения многих сред разработки программного обеспечения.**

**Microsoft SQL Server - система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. 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. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.**

**2.3 Определение требований к техническим средствам**

Программа Построение железнодорожных маршрутов функционирует на компьютерах с определенным аппаратным и программным обеспечением.

Минимальные требования для корректной работы системы:

* центральный процессор - 2 ГГц ;
* оперативная память - 2 ГБ;
* свободное место на HDD - не менее 5 ГБ;
* мышь;
* клавиатура;
* монитор 14 и выше дюймов;
* операционная система Windows.

**2.4 Защита информации**

**Защита информации – комплекс действий, направленных на обеспечение информационной безопасности (целостности, доступности и конфиденциальности информации и ресурсов, используемых для обработки и передачи данных).**

**Система считается безопасной, если используя соответствующие средства, доступ к информации имеют только авторизованн**ые лица или же действующие от их имени процессы.

Очевидно, что абсолютно безопасных систем нет, и здесь речь идет о надежной системе в смысле «система, которой можно доверять». Система считается надежной, если она с использованием достаточных средств обеспечивает одновременную обработку информации разной степени секретности группой пользователей без нарушения прав доступа.

В программном модуле для автоматизации работы железнодорожного транспорта присутствует система входа для обеспечения различных прав доступа к базе данных. Вид формы приведен ниже на рисунке А.1, приложения А.

Вход в систему работает через логин и пароль от базы данных и позволяет при отсутствии у пользователя прав доступа к определенным таблицам выдать окно ошибки показанное на рисунке А.15, приложения А.

**3. Разработка программного обеспечения**

**3.1 Определение формы представления входных и выходных данных**

При открытии программы отображается главное окно программы и главная страница, содержащая форму с выбором страниц. Первый пункт - это страница с информацией о билетах.

Основное назначение формы «Билеты» – просмотр данных о билетах (например, к какому маршруту они относятся и их стоимость) и занесение информации о билетах.

Внешний вид формы приведен на рисунке А.3, приложения А.

Форма «Продажа билетов»» содержит следующие компоненты:

–  Label – служит для показа названия полей базы данных;

–  Textbox – служит для ввода данных добавляемых в базу данных или обновления существующих;

– Button – отправка новых данных в базу данных или обновление существующих;

– DataGrid – служит для показа информации находящийся в базе данных;

–  DropDown – служит для выбора информации из таблиц, связанных внешним ключом базы данных.

Для открытия формы «Билеты» в главном окне необходимо нажать кнопку «Билеты» в боковом меню главного окна. После чего выбранная страница изменится с страницы «Главная» на страницу «Билеты» и появится загруженная информация по билетам из базы данных на текущий момент времени. После этого, если необходимо отредактировать информацию о существующем билете, нужно дважды нажать левой кнопкой мыши по одному из существующих билетов и перед вами появится диалог для редактирования информации, в котором можно изменить необходимые значения. Затем нажать кнопку «Сохранить» для сохранения информации. После чего страница «Билеты» обновится.

Для удаления данных в компоненте DataGrid имеется кнопка с изображением корзины, при нажатии на которую удаляется неактуальная информация.

Для добавления новой информации в базу данных имеется кнопка «Добавить информацию».

Основное назначение формы «Продажа билетов» – просмотр данных о уже проданных билетах и занесение информации о новых проданных билетах.

Внешний вид формы приведен на рисунке А.4, приложения А.

Форма «Продажа билетов» содержит следующие компоненты:

–  Label – служит для показа названия полей базы данных;

– Textbox – служит для ввода данных добавляемых в базу данных или обновления существующих данных;

– Button – отправка новых данных в базу данных или обновление существующих;

– DataGrid – служит для показа информации находящийся в базе данных;

–  DropDown – служит для выбора информации из таблиц связанных внешним ключом базы данных.

Для открытия формы «Продажа билетов» в главном окне необходимо нажать кнопку «Продажи» в боковом меню главного окна. После чего выбранная страница изменится с страницы «Главная» на страницу «Продажа билетов» и появится загруженная информация по продажам билетов из базы данных на текущий момент времени. После этого, если необходимо отредактировать информацию о существующей продаже билета, нужно дважды нажать левой кнопкой мыши по одному из существующих пунктов и перед вами появится диалог для редактирования информации, в котором можно изменить необходимые значения. Затем нажать кнопку «Сохранить» для сохранения информации. После чего страница «Продажа билетов» обновится.

Для удаления данных в компоненте DataGrid имеется кнопка с изображением корзины, при нажатии на которую удаляется неактуальная информация.

Для добавления новой информации в базу данных имеется кнопка «Добавить информацию».

Основное назначение формы «Информация о сотрудниках» - просмотр информации о сотрудниках и добавление или редактирование данных о сотрудниках. Внешний вид формы приведен на рисунке А.5, приложения А.

Форма «Информация о сотрудниках» содержит следующие компоненты:

– Label -служит для показа названия полей базы данных;

– Textbox – служит для ввода данных добавляемых в базу данных или обновления существующих данных;

– Button – Обновление DataGrid или отправка новых данных в базу данных или обновление существующих;

–  DataGrid – служит для показа информации находящийся в базе данных.

– DropDown – служит для выбора информации из таблиц, связанных внешним ключом базы данных.

Для открытия формы «Информация о сотрудниках» в главном окне необходимо нажать кнопку «Сотрудники» в боковом меню главного окна. После чего выбранная страница изменится с страницы «Главная» на страницу «Информация о сотрудниках» и появится загруженная информация о сотрудниках из базы данных на текущий момент времени. После этого, если необходимо отредактировать информацию о существующем сотруднике, нужно дважды нажать левой кнопкой мыши по одному из существующих пунктов и перед вами появится диалог для редактирования информации, в котором можно изменить необходимые значения. Затем нажать кнопку «Сохранить» для сохранения информации. После чего страница «Информация о сотрудниках» обновится.

Для удаления данных в компоненте DataGrid имеется кнопка с изображением корзины, при нажатии на которую удаляется неактуальная информация.

Для добавления новой информации в базу данных имеется кнопка «Добавить информацию».

Основное назначение формы «Должности» - просмотр информации о наименовании должности, необходима ли стажировка для данной должности, добавление или редактирование информации о должностях. Внешний вид формы приведен на рисунке А.6, приложения А.

Форма «Должности» содержит следующие компоненты:

– Label – служит для показа названия полей базы данных;

– Textbox – служит для ввода данных добавляемых в базу данных или обновления существующих данных;

– Button – Обновление DataGrid или отправка новых данных в базу данных или обновление существующих;

–  DataGrid – служит для показа информации находящийся в базе данных.

– DropDown – служит для выбора информации из таблиц связанных внешним ключом базы данных

Для открытия формы «Должности» в главном окне необходимо нажать кнопку «Должности» в боковом меню главного окна. После чего выбранная страница изменится с страницы «Главная» на страницу «Должности» и появится загруженная информация по должностям из базы данных на текущий момент времени. После этого, если необходимо отредактировать информацию о существующей дожности, нужно дважды нажать левой кнопкой мыши по одному из существующих пунктов и перед вами появится диалог для редактирования информации, в котором можно изменить необходимые значения. Затем нажать кнопку «Сохранить» для сохранения информации. После чего страница «Должности» обновится.

Для удаления данных в компоненте DataGrid имеется кнопка с изображением корзины, при нажатии на которую удаляется неактуальная информация.

Для добавления новой информации в базу данных имеется кнопка «Добавить информацию».

Форма «Информация о маршрутах» cлужит для просмотра информации о существующих маршрутах. Внешний вид формы приведен на рисунке А.7, приложения А.

Форма «Информация о маршрутах» содержит следующие компоненты:

– Label – служит для показа названия полей базы данных;

– Textbox – служит для ввода данных добавляемых в базу данных или обновления существующих данных;

– Button – Обновление DataGrid или отправка новых данных в базу данных или обновление существующих;

–  DataGrid – служит для показа информации находящийся в базе данных.

– DropDown – служит для выбора информации из таблиц связанных внешним ключом базы данных

Для открытия формы «Информация о маршрутах» в главном окне необходимо нажать кнопку «Маршруты» в боковом меню главного окна. После чего выбранная страница изменится с страницы «Главная» на страницу «Информация о маршрутах» и перед вами появится загруженная информация по маршрутам из базы данных на текущий момент времени. После этого если необходимо отредактировать информацию о существующем маршруте, нужно дважды нажать левой кнопкой мыши по одному из существующих пунктов и перед вами появится диалог для редактирования информации, в котором можно изменить необходимые значения. Затем нажать кнопку «Сохранить» для сохранения информации. После чего страница «Информация о маршрутах» обновится.

Для удаления данных в компоненте DataGrid имеется кнопка с изображением корзины, при нажатии на которую удаляется неактуальная информация. Для добавления новой информации в базу данных имеется кнопка «Добавить информацию».

Форма «Промежуточные пункты маршрутов» cлужит для просмотра информации о промежуточных пунктах маршрутов. Внешний вид формы приведен на рисунке А.8, приложения А.

Форма «Промежуточные пункты маршрутов» содержит следующие компоненты:

– Label – служит для показа названия полей базы данных;

– Textbox – служит для ввода данных добавляемых в базу данных или обновления существующих данных;

– Button – Обновление DataGrid или отправка новых данных в базу данных или обновление существующих;

–  DataGrid – служит для показа информации находящийся в базе данных.

– DropDown – служит для выбора информации из таблиц связанных внешним ключом базы данных

Для открытия формы «Промежуточные пункты маршрутов» в главном окне необходимо нажать кнопку «Промежуточные пункты маршрутов» в боковом меню главного окна. После чего выбранная страница изменится с страницы «Главная» на страницу «Промежуточные пункты маршрутов» и перед вами появится загруженная информация по промежуточным пунктам маршрутов из базы данных на текущий момент времени. После этого если необходимо отредактировать информацию о существующем промежуточном пункте, нужно дважды нажать левой кнопкой мыши по одному из существующих пунктов и перед вами появится диалог для редактирования информации, в котором можно изменить необходимые значения. Затем нажать кнопку «Сохранить» для сохранения информации. После чего страница «Промежуточные пункты маршрутов» обновится.

Для удаления данных в компоненте DataGrid имеется кнопка с изображением корзины, при нажатии на которую удаляется неактуальная информация.

Для добавления новой информации в базу данных имеется кнопка «Добавить информацию».

Основное назначение формы «Поезда» - просмотр данных о том какие вагоны поездов используются на маршрутах и занесение или обновление данных о поездах. Вид формы «Поезда» приведен на рисунке А.9, приложения А.

Форма «Поезда» содержит следующие компоненты:

– Label – служит для показа названия полей базы данных;

– Textbox – служит для ввода данных добавляемых в базу данных или обновления существующих данных;

– Button – обновление DataGrid или отправка новых данных в базу данных или обновление существующих;

– DataGrid – служит для показа информации находящийся в базе данных.

Для открытия формы «Поезда» в главном окне необходимо нажать кнопку «Поезда» в боковом меню главного окна. После чего выбранная страница изменится с страницы «Главная» на страницу «Поезда» и перед вами появится загруженная информация по поездам из базы данных на текущий момент времени. После этого если необходимо отредактировать информацию о существующем поезде , нужно дважды нажать левой кнопкой мыши по одному из существующих пунктов и перед вами появится диалог для редактирования информации, в котором можно изменить необходимые значения. Затем нажать кнопку «Сохранить» для сохранения информации. После чего страница «Поезда» обновится.

Для удаления данных в компоненте DataGrid имеется кнопка с изображением корзины, при нажатии на которую удаляется неактуальная информация.

Для добавления новой информации в базу данных имеется кнопка «Добавить информацию».

Основное назначение формы «Техническое обслуживание» - просмотр данных о состоянии технического обслуживания вагонов поезда и занесение или обновление данных о техническом обслуживании поездов. Вид формы «Техническое обслуживание» приведен на рисунке А.10, приложения А.

Форма «Техническое обслуживание» содержит следующие компоненты:

– Label – служит для показа названия полей базы данных;

– Textbox – служит для ввода данных добавляемых в базу данных или обновления существующих данных;

– Button – обновление DataGrid или отправка новых данных в базу данных или обновление существующих;

– DataGrid – служит для показа информации находящийся в базе данных.

Для открытия формы «Техническое обслуживание» в главном окне необходимо нажать кнопку «Обслуживание» в боковом меню главного окна. После чего выбранная страница изменится с страницы «Главная» на страницу «Техническое обслуживание» и перед вами появится загруженная информация по техническому обслуживанию поездов из базы данных на текущий момент времени. После этого если необходимо отредактировать информацию о уже проведенном обслуживании, нужно дважды нажать левой кнопкой мыши по одному из существующих пунктов и перед вами появится диалог для редактирования информации, в котором можно изменить необходимые значения. Затем нажать кнопку «Сохранить» для сохранения информации. После чего страница «Техническое обслуживание» обновится.

Для удаления данных в компоненте DataGrid имеется кнопка с изображением корзины, при нажатии на которую удаляется неактуальная информация.

Для добавления новой информации в базу данных имеется кнопка «Добавить информацию».

Приложение также работает на мобильных телефонах и имеет вид, приведенный на рисунках А.11-А.18, приложения А.

На телефонах элементы управления и меню приложения находится внизу экрана. Для открытия одной из форм необходимо нажать на соответствующий пункт меню внизу экрана. После чего действия аналогичны работе на ПЭВМ и различие только в том, что все элементы управления адаптированы по размеру для работе на экране телефона.

**3.2 Тестирование программного модуля**

**В программу из таблицы** ««**Employees**» базы данных Microsoft SQL server входят данные представленные в таблице 2.

Таблица 2 – Данные таблицы **Employees.**

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| **Num** | Код сотрудника | **bigint** | Первичный |
| **Surname** | **Фамилия сотрудника** | **ntext** |  |
| **Name** | **Имя сотрудника** | **ntext** |  |
| **Patronym** | **Отчество сотрудника** | **ntext** |  |
| **Employed\_Since** | **Работает с** | **ntext** |  |
| **Job\_Num** | **Внешний ключ с информацией о должности** | **bigint** | **Внешний** |

Таблица 3 – Данные таблицы Jobs**.**

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| **Job\_Key** | **Код должности** | **bigint** | Первичный |
| **Job** | **Должность** | **ntext** |  |
| **Internship** | **Стажировка** | **ntext** |  |

Из таблицы «Obsluzhivanie» в программу входят данные представленные в таблице 4.

Таблица 4 – Данные таблицы Obsluzhivanie**.**

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| 1 | 2 | 3 | 4 |
| NomerObsluzhivania | **Код обслуживания** | **bigint** | **Первичный** |
| Model\_TrainKey | **Внешний ключ с информацией о модели поезда** | **bigint** | **Внешний** |
| **Data\_Poslednego\_Obsluzhivania** | **Дата последнего обслуживания** | **ntext** |  |

Продолжение таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| **Ingener\_Obsluzhivania** | **Инженер, проводивший обслуживание** | **ntext** |  |
| **Problemi\_Sostava** | **Найденные проблемы состава вагонов** | **ntext** |  |
| **Data\_Sledueschego\_Obsluzivania** | **Дата следующего обслуживания** | **ntext** |  |
| **Goden\_K\_Doroge** | **Готовность состава вагонов к рейсу** | **ntext** |  |

Из таблицы «**Trains**» в программу входят данные представленные в таблице 5.

Таблица 5 – Данные таблицы Trains.

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| **NumVagona** | Код | **bigint** | Первичный |
| **Model\_Sostava\_Vagonov** | **Модель состава вагонов поезда** | **ntext** |  |
| **Kolichestvo\_Mest** | **Количество мест в составе вагонов** | **bigint** |  |

Из таблицы «**Marshuti**» в программу входят данные представленные в таблице 6.

Таблица 6 – Данные таблицы **Marshuti.**

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| **Nomer\_Marshuta** | Код маршрута | **bigint** | Первичный |
| **Nachalni\_Punkt** | **Начальный пункт** | **ntext** |  |
| **Konechni\_Punkt** | **Конечный пункт** | **ntext** |  |
| **Emp\_Num** | **Код сотрудника, назначенного машинистом на маршрут** | **bigint** |  |
| ModelTrainKey | **Внешний ключ с информацией о модели поезда** | **bigint** | **Внешний** |
| **Vremya\_Proezda** | **Время проезда маршрута** | **ntext** |  |

Из таблицы «Prodazhi» в программу входят данные представленные в таблице 7.

Таблица 7 – Данные таблицы Prodazhi

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| **Num** | Код | **bigint** | Первичный |
| **Sale\_Date** | **Дата продажи** | **ntext** |  |
| **Nomer\_BiletaKey** | **Внешний ключ с информацией о номере билета** |  |  |

Таблица 8 – Данные таблицы Bilety.

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| **Nomer\_Bileta** | Код | **bigint** | Первичный |
| **Nomer\_MarshutaKey** | **Внешний ключ с информацией о маршуте** | **bigint** | **Внешний** |
| **Stoimost** | **Стоимость билета** | **ntext** |  |
| **Konechnaya\_Ostanovka** | **Конечная остановка пассажира купившего билет** | **ntext** |  |

Таблица 9 – Данные таблицы Promezhutochinye\_Punkty.

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| **PunktKey** | Код | **bigint** | Первичный |
| **PunktPoint1** | Название промежуточного пункта | **Nvarchar(500)** |  |
| **PunktPoint2** | Название промежуточного пункта | **Nvarchar(500)** |  |
| **PunktPoint3** | Название промежуточного пункта | **Nvarchar(500)** |  |
| **PunktPoint4** | Название промежуточного пункта | **Nvarchar(500)** |  |
| **PunktPoint5** | Название промежуточного пункта | **Nvarchar(500)** |  |
| **PunktPoint6** | Название промежуточного пункта | **Nvarchar(500)** |  |
| **PunktPoint7** | Название промежуточного пункта | **Nvarchar(500)** |  |

Продолжение таблицы 9

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| **PunktPoint8** | Название промежуточного пункта | **Nvarchar(500)** |  |
| **PunktPoint9** | Название промежуточного пункта | **Nvarchar(500)** |  |
| **PunktPoint10** | Название промежуточного пункта | **Nvarchar(500)** |  |

Данные, которые отправляются в базу данных из программы представлены в таблице 10.

Таблица 10 – Данные, которые отправляются в базу данных из программы

|  |  |  |  |
| --- | --- | --- | --- |
| Выход | Имя Таблицы | Имя Поля Таблицы | Тип Поля |
| 1 | 2 | 3 | 4 |
| Выход в | **Employees** | **Num** | **bigint** |
| Выход в | **Employees** | **Surname** | **ntext** |
| Выход в | **Employees** | **Name** | **ntext** |
| Выход в | **Employees** | **Patronym** | **ntext** |
| Выход в | **Employees** | **Employed\_Since** | **ntext** |
| Выход в | **Employees** | **Job\_Num** | **bigint** |
| Выход в | **Jobs** | **Job\_Key** | **bigint** |
| Выход в | Jobs | **Job** | **ntext** |
| Выход в | **Jobs** | **Internship** | **ntext** |
| Выход в | Obsluzhivanie | **Num** | **bigint** |
| Выход в | Obsluzhivanie | ModelTrainKey | **bigint** |
| Выход в | Obsluzhivanie | **Ingener\_Obsluzhivania** | **ntext** |
| Выход в | Obsluzhivanie | **Problemi\_Sostava** | **ntext** |
| Выход в | Obsluzhivanie | **Data\_Sledueschego\_Obsluzivania** | **ntext** |
| Выход в | Obsluzhivanie | **Goden\_K\_Doroge** | **ntext** |
| Выход в | Trains | **NumVagona** | **bigint** |
| Выход в | Trains | **Model\_Sostava\_Vagonov** | **ntext** |
| Выход в | Trains | **Kolichestvo\_Mest** | **bigint** |
| Выход в | **Marshuti** | **Nomer\_Marshuta** | **bigint** |

Продолжение таблицы 10

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Выход в | **Marshuti** | ModelTrainKey | **bigint** |
| Выход в | **Marshuti** | **Emp\_Num** | **bigint** |
| Выход в | **Marshuti** | **Nachalni\_Punkt** | **ntext** |
| Выход в | **Marshuti** | **Konechni\_Punkt** | **ntext** |
| Выход в | **Marshuti** | **Vremya\_Proezda** | **ntext** |
| Выход в | Prodazhi | **Num** | **bigint** |
| Выход в | Prodazhi | **Sale\_Date** | **ntext** |
| Выход в | Prodazhi | **Nomer\_BiletaKey** | **bigint** |
| Выход в | **Bilety** | **Nomer\_Bileta** | **bigint** |
| Выход в | **Bilety** | **Nomer\_MarshutaKey** | **bigint** |
| Выход в | **Bilety** | **Stoimost** | **ntext** |
| Выход в | **Bilety** | **Konechnaya\_Ostanovka** | **ntext** |
| Выход в | Promezhutochinye\_Punkty | **PunktKey** | **bigint** |
| Выход в | Promezhutochinye\_Punkty | **PunktPoint 1 - 10** | **ntext** |

Тестирование проводилось с целью проверки корректности работы программы и её отладки.

Тестирование программы осуществлялось на персональном компьютере со следующими техническими характеристиками:

* процессор – AMD Ryzen 5 7530U with Radeon Graphics 2.00 GHz
* видеокарта – AMD Radeon Vega 7 Graphics
* оперативная память - 16,00 ГБ LPDDR4
* операционная система – Windows 11

Для обеспечения устойчивости программы и базы данных в каждой форме присутствует проверка на ошибки. Сообщение об ошибке имеет вид приведенный на рисунке В.2, приложения В.

**В результате тестирования установлено, что программа работает корректно, выполняет все заложенные в нее функции.**

**3.3 Разработка справочной системы**

Разработанная справочная система для автоматизированной информационной системы «ЖД маршруты» позволяет пользователю ознакомиться с краткой информацией об авторе программы и для чего она предназначена.

Вид справки «О программе» представлен на рисунке А.19, приложения А.

**3.4 Описание разрабатываемого программного продукта**

Программный продукт для дипломной работы по теме «Разработка автоматизированной системы «Построение железнодорожных маршрутов» представляет собой комплексное программное решение, предназначенное для оптимизации и управления железнодорожными маршрутами. Основные характеристики продукта:

* **модуль построения маршрутов** – форма, позволяющая создавать маршруты на основе данных из СУБД;
* **база данных железнодорожных станций**  - содержит информацию о всех железнодорожных станциях, путях и расписаниях, доступных для построения маршрутов;
* **пользовательский интерфейс** -интуитивно понятный интерфейс, позволяющий пользователям легко выбирать станции, вводить параметры поиска и просматривать возможные маршруты;
* **интеграция с СУБД Microsoft SQL Server** - использование надежной и масштабируемой системы управления базами данных для обработки и хранения больших объемов данных;
* **использование SQL-запросов** - программа использует SQL-запросы для фильтрации и сортировки данных, что обеспечивает гибкость при работе с информацией;
* **масштабируемость** - возможность расширения функционала и увеличения объема обрабатываемых данных без потери производительности.

Этот программный продукт будет способствовать повышению эффективности работы железнодорожной сети, улучшению планирования маршрутов и предоставлению качественного сервиса для пассажиров.

Таблица 1 − Трудоемкость разработки программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| Виды работ | Кол-во  операций | Норма времени, ч | |
| на одну  операцию | на все  операции |
| 1 | 2 | 3 | 4 |
| 1. Подготовка исходных данных | 25 | 0,30 | 7,50 |
| 1. Реализация алгоритмов контрольных задач с использованием ПС ПЭВМ | 15 | 0,30 | 4,50 |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 1. Обработка данных и получение результатов | 16 | 0,28 | 4,48 |
| 1. Анализ ошибок обработки данных и подготовка заключения о результатах проверки | 12 | 0,35 | 4,20 |
| 1. Ознакомление сотрудников службы сопровождения с содержанием задач, структурой входных и выходных данных | 9 | 0,79 | 7,11 |
| 1. Определение параметров настройки | 12 | 0,17 | 2,04 |
| 1. Ознакомление с объектом внедрения | 5 | 0,94 | 4,70 |
| 1. Консультации по подготовке пользователями исходных данных в соответствии с требованиями и ограничениями ОС ПЭВМ | 8 | 1,00 | 8,00 |
| 1. Разработка рекомендаций по реализации алгоритмов и требований пользователя к обработке данных с использованием ППП ПЭВМ по подготовке задач к опытной эксплуатации | 4 | 2,20 | 8,80 |
| 1. Оценка соответствия функциональных и эксплуатационных характеристик ПС требованиям к обработке данных | 9 | 1,10 | 9,90 |
| 1. Проведение консультаций и анализ ошибок комплексирования в период опытной эксплуатации | 7 | 1,70 | 11,90 |
| 1. Разработка рекомендаций по созданию программных средств сопряжения (программ и блоков пользователя, осуществляющих промежуточную обработку данных) | 6 | 1,70 | 10,20 |
| 1. Корректировка программ с целью изменения незначительных функциональных характеристик | 15 | 1,06 | 15,90 |
| 1. Разработка дополнительных модулей и включение их в состав ПС | 9 | 2,40 | 21,60 |
| 1. Анализ требований задач пользователя к обработке данных и характеристик среды их функционирования | 6 | 0,70 | 4,20 |
| 1. Разработка требований к тестированию и подготовка тестовых единиц | 10 | 0,70 | 7,00 |
| 1. Анализ результатов прогона и разработка функциональных спецификаций на корректировку ПС | 5 | 0,60 | 3,00 |
| 1. Внесение изменений в программы и эксплуатационную документацию ПС у пользователя | 4 | 0,29 | 1,16 |
| 1. Демонстрация функционирования на контрольных задачах службы сопровождения | 8 | 0,20 | 1,60 |
| 1. Анализ организационно- экономических и технических характеристик объекта внедрения ПС | 2 | 0,40 | 0,80 |
| 1. Разработка требований к выбору ПС для реализации задач пользователя | 3 | 0,40 | 1,20 |
| 1. Рекомендации по выбору ПС | 3 | 0,23 | 0,69 |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 1. Оценка полноты охвата функциональными возможностями ПС | 4 | 0,40 | 1,60 |
| 1. Требования задач пользователей данного класса | 5 | 0,40 | 2,00 |
| 1. Выработка рекомендаций по расширению функциональных возможностей ПС | 3 | 0,10 | 0,30 |
| 1. Оценка необходимости проведения обучения работе по утвержденной технологии | 3 | 0,50 | 1,50 |
| 1. Анализ характеристик и производственных условий разработки, изготовления и сопровождения ПС у пользователя | 2 | 0,60 | 1,20 |
| 1. Разработка рекомендаций по применению у пользователя поставляемых технологических процессов разработки, производства и сопровождения ПС | 2 | 1,10 | 2,20 |
| 1. Практическая работа пользователей с ПС в вычислительном центре службы сопровождения | 3 | 1,40 | 4,20 |
| 1. Разработка рекомендаций по подготовке исходных данных в соответствии с требованиями и ограничениями ПС, по созданию технологии обработки данных с помощью ПС | 4 | 0,58 | 2,32 |
| 1. Консультации по устранению ошибок подготовки данных и решению задач | 3 | 0,30 | 0,90 |
| 1. Вывод на печать | 112 | 0,0028 | 0,31 |
| Итого трудоемкость  в т.ч ПЭВМ,  принтер |  |  | 157,01  156,70  0,31 |

Реализация программы выполнялась техником-программистом шестого разряда.

Расход бумаги составит 112 листов, носителей информации – 1 диск DVD‑R.

Кроме программного модуля разработана сопровождающая программная документация в соответствии с ГОСТ 19.402-2000 «Описание программы», которая представлена в приложении Г и документация пользователя в соответствии с ГОСТ ИСО 9127-2002 «Документация пользователя», которая представлена в приложении Д.

**4 Экономическая часть**

**4.1 Расчет материальных затрат**

К материальным затратам относятся затраты на расходные материалы и затраты на электроэнергию на технологические цели.

Материальные затраты МЗ, руб., рассчитываются по формуле

МЗ = Ср.м + Сэн, (1)

где Ср.м – стоимость расходных материалов, руб.;

Сэн – стоимость электроэнергии, руб.

Затраты на расходные материалы Ср.м, руб., определяются по формуле

Ср.м = Сб + Ск + Сн, (2)

где Сб – стоимость бумаги, руб.;

Ск – стоимость картриджа для принтера, руб.;

Сн – стоимость носителя информации, руб.

Затраты на бумагу Сб, руб., определяются по формуле

Сб = Цб × Рб, (3)

где Цб – цена за 1 лист бумаги, руб.;

Рб – расход листов бумаги при разработке и печати программного

продукта, шт.

Затраты на носители информации Сн, руб., определяются по формуле

Сн = Цн × Рн, (4)

где Цн – цена носителя информации, руб.;

Рн – расход CD, CD-RW, шт.

Затраты на носители информации составят

Сн = 4,50 × 1= 4,50 руб.

Затраты на картридж Ск, руб., определяются по формуле:

Ск=Цк×Рк (5)

где Цк – цена картриджа, руб,;

Рк – расход картриджей, шт.

Затраты на картридж составят

Ск = 55,00 × 1= 55,00 руб.

Затраты на бумагу составят

Стоимость расходных материалов по вариантам составит

Ср.м = 2,69 + 55,00 + 4,50 = 62,19 руб.

Затраты на электроэнергию определяют исходя из загруженности персонального компьютера программиста и частичной занятости принтера за время разработки (учитывая, что при разработке программного продукта принтер используется меньше, чем персональный компьютер).

Затраты на электроэнергию Сэн, руб., определяют по формуле

Сэн = Цэн × (Тпк × Wпк + Тприн × Wприн), (6)

где Цэн – тариф за 1 кВт-ч электроэнергии, руб.;

Тпк – время работы персонального компьютера, ч;

Тприн – время работы принтера, ч;

Wпк – потребляемая мощность ПК, 0,40 кВт-ч;

Wприн – потребляемая мощность принтера, 0,35 кВт-ч;

Затраты на электроэнергию составят

Сэн = 0,39(156,700,400,310,35) = 24,49 руб.

Материальные затраты составили:

МЗ = 62,19 + 24,49 = 86,68 руб.

**4.2 Расчет затрат на оплату труда и отчислений на социальные нужды**

Расчет затрат на оплату труда и отчислений на социальные нужды осуществляется при помощи прикладной программы на ПЭВМ и данный расчет приведен в приложениях Е.

**4.3 Расчет себестоимости разработки программного продукта**

Себестоимость разработки программного продукта Спол., руб., рассчитывается по формуле

Спол = МЗ + ФОТ + Ос.н + Ао + Зпр, (7)

где Ао – амортизационные отчисления основных средств и

нематериальных активов, руб.;

Зпр – прочие затраты, руб.

Амортизационные отчисления Ао, руб., рассчитываются по формуле

, (8)

, (9)

 , (10)

где ОС – стоимость основных средств и нематериальных активов, руб.;

Тн – нормативный срок службы, лет (принять в размере 10 лет);

 – коэффициент, учитывающий долю занятости ПЭВМ;

Фд – действительный фонд времени работы ПЭВМ, ч;

Фн – утвержденный номинальный годовой фонд времени на момент

разработки КР или ДП, ч;

%П – процент простоя оборудования в ремонте, %. (принять в размере от 2% до 10%);

Тр – трудоемкость ПЭВМ, ч.

Действительный фонд времени работы ПЭВМ составит

Коэффициент, учитывающий долю занятости ПЭВМ равен

Амортизационные отчисления составят

Прочие затраты Зпр, руб., включают оплату услуг связи, ВЦ, банков, сигнализаций, консультаций, аудиторских и рекламных услуг, за охрану; вознаграждения работникам за изобретение и рационализаторские предложения; арендная плата за имущество; плата по процентам за краткосрочный и долгосрочный кредиты под пополнение оборотных средств; земельный налог; налог за пользование природными ресурсами и другие налоги, включаемые в себестоимость, рассчитываются по формуле

(11)

где %Зпр – процент прочих затрат, %;

Зпр – 300-600%.

Рассчитаем прочие затраты

Итого полная себестоимость составит

Результаты расчетов представлены в таблице 2.

Таблица 2 – Смета затрат себестоимости разработки программного продукта

|  |  |
| --- | --- |
| Наименование элементов затрат | Сумма, руб. |
| 1. Материальные затраты (за вычетом стоимости возвратных отходов) | 86,68 |
| 1. Затраты на оплату труда | 586,82 |
| 1. Отчисления на социальные нужды | 199,52 |
| 1. Амортизация основных средств и нематериальных активов, используемых в предпринимательской деятельности | 31,50 |
| 1. Прочие затраты | 1530,84 |
| Итого затрат на производство и реализации продукции | 2435,36 |

* 1. **Расчет отпускной цены разработки программного продукта**

Отпускная цена разработки программного продукта без учета НДС Ц, руб., рассчитывается по формуле

Ц = Спол + Пр(12)

Прибыль Пр, руб., рассчитывается по формуле

Пр =, (13)

где Нр – норматив рентабельности, %;

Нр – до 20%.

Прибыль составит

Отпускная цена продукта без учета НДС составит

Ц = 2435,36 + 487,07 = 2922,43 руб.

Цена отпускная с учетом НДС Цотп, руб., рассчитывается по формуле

Цотп = Ц + НДС, (14)

где НДС – налог на добавленную стоимость, руб.

Налог на добавленную стоимость определяется по формуле

НДС = , (15**)**

где hндс – ставка налога на добавленную стоимость, %.

Налог на добавленную стоимость составит

Отпускная цена с учетом НДС составит

Цотп = + = 3506,92 руб.

Результаты расчетов занесены в таблицу 3.

Таблица 3 – Калькуляция отпускной цены разработки программного продукта

|  |  |
| --- | --- |
| Наименование статей калькуляции | Сумма, руб. |
| 1. Стоимость расходных материалов | 62,19 |
| 1. Стоимость электроэнергии | 24,49 |
| Итого материальные затраты | 86,68 |
| 1. Затраты на оплату труда | 586,82 |
| 1. Отчисления на социальные нужды | 199,52 |
| 1. Амортизация основных средств и нематериальных активов, используемых в предпринимательской деятельности | 31,50 |
| 1. Прочие затраты | 1530,84 |
| Итого полная себестоимость | 2435,36 |
| 1. Прибыль | 487,07 |
| Отпускная цена без учета НДС | 2922,43 |
| 1. Налог на добавленную стоимость | 584,49 |
| Итого цена отпускная с учетом НДС | 3506,92 |

* 1. **Расчет технико-экономических показателей**

Эффективность разработки программного продукта подтверждается технико-экономическими показателями:

* трудоемкость разработки программного продукта;
* полная себестоимость;
* прибыль;
* отпускная цена;
* рентабельность продукта;
* материалоемкость;
* дельный вес ТЭР в себестоимости;
* затраты на 1 рубль реализованной продукции.

Рентабельность продукта R, %, – показатель оценки эффективного использования текущих затрат на разработку программного продукта и рассчитывается по формуле

R = × 100(16)

Рентабельность продукта составит

Материалоемкость Ме, руб./руб., – показывает долю материальных затрат в выручке продукции и рассчитывается по формуле

Ме = (17)

Материалоемкость составит

Удельный вес топливно-энергетических ресурсов в себестоимости , %, – показывает долю топливно-энергетических затрат в себестоимости продукции и рассчитывается по формуле

= 100 (18)

Удельный вес топливно-энергетических ресурсов в себестоимости составит

Затраты на 1 рубль реализованной продукции Зреал, руб./руб., – это один из показателей эффективности производства и определяется по формуле

Зреал = (19)

Затраты на 1 рубль реализованной продукции составят

Технико-экономические показатели приведены в таблице 4.

Таблица 4 – Технико-экономические показатели

|  |  |  |
| --- | --- | --- |
| Наименование показателей | Единица  измерения | Величина показателя |
|
| 1. Трудоемкость разработки программного продукта | ч | 157,01 |
| 1. Полная себестоимость | руб. | 2435,36 |
| 1. Прибыль | руб. | 487,07 |
| 1. Рентабельность продукта | % | 20,00 |
| 1. Отпускная цена | руб. | 3506,92 |
| 1. Материалоемкость | руб./руб. | 0,02 |
| 1. Удельный вес топливно-энергетических ресурсов в себестоимости | % | 1,01 |
| 1. Затраты на 1 рубль реализованной продукции | руб./руб. | 0,69 |

Отпускная цена разработки программного продукта составит 3506,92 руб.

Удельный вес топливно-энергетических ресурсов в себестоимости показывает долю топливно-энергетических затрат в себестоимости продукции и составляет 1,01 %.

Себестоимость данного программного продукта составила 2435,36 руб.

Проанализировав полученные данные, можно сделать вывод, что внедрение приложения является экономически выгодным.

**5 Охрана труда**

Правовые, нормативные, социально-экономические и организационные вопросы охраны труда.

Государственная политика в области охраны труда. Цель государственной политики в области охраны труда – сохранение жизни и здоровья граждан в процессе трудовой деятельности.

Основными принципами государственной политики в области охраны труда являются:

– приоритет жизни и здоровья работников по отношению к результатам производственной деятельности;

– обеспечение гарантий права работников на охрану труда.

Основные направления государственной политики в области охраны труда:

– государственное управление деятельностью в области охраны труда, включая государственный надзор и контроль за соблюдением законодательства по этим вопросам;

– принятие законов и иных нормативных правовых актов, направленных на совершенствование правоотношений в этой сфере, установление единых нормативных требований в области безопасности и гигиены труда;

– комплексное решение задач обеспечения здоровых и безопасных условий труда с учетом других направлений социальной и экономической политики, достижений в области науки, техники, технологий и охраны окружающей среды;

– обеспечение социально-экономической защиты прав работников в области охраны труда, в том числе потерпевших на производстве, а также членов их семей на основе обязательного социального страхования от несчастных случаев на производстве, профессиональных заболеваний, установление компенсаций за тяжелую работу и работу с вредными и (или) опасными условиями труда;

– подготовка специалистов, обучение и повышение квалификации работников по вопросам охраны труда и др.

Государственное управление охраной труда реализуется на следующих уровнях:

– республиканском;

– отраслевом;

– региональном.

На республиканском уровне государственное управление охраной труда осуществляет Правительство Республики Беларусь непосредственно или уполномоченные им министерства, другие республиканские органы государственного управления.

На отраслевом уровне государственное управление охраной труда осуществляют министерства, другие республиканские органы государственного управления, объединения (учреждения), подчиненные Правительству Республики Беларусь.

На региональном уровне государственное управление охраной труда осуществляют местные исполнительные и распорядительные органы.

В соответствии с законодательством о труде и охране труда наниматель обязан обеспечивать здоровые и безопасные условия труда, соблюдать установленные нормативными правовыми актами (документами) требования охраны труда, принимать необходимые меры по профилактике производственного травматизма, профессиональных и других заболеваний.

Деятельность службы охраны труда. Охрана труда – система обеспечения безопасности жизни и здоровья работников в процессе трудовой деятельности, включающая правовые, социально-экономические, организационные, технические, психофизиологические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия и средства.

Типовое положение о службе охраны труда организации разработано в соответствии со статьей 227 Трудового кодекса Республики Беларусь и направлено на выполнение обязанностей по охране труда государственной политики в этой области.

Управление охраной труда в организации осуществляет ее руководитель.

Для организации работы и осуществления контроля по охране труда руководитель организации создает службу охраны труда (управление, отдел, другое структурное подразделение, специалист по охране труда и т.п.).

По своему статусу, в том числе по оплате труда, служба включается в состав основных производственно-технических подразделений организации, и ее решения, принятые в пределах предоставленных прав и полномочий, являются обязательными для выполнения руководителями и другими работниками всех подразделений организации.

На службу охраны труда возлагаются следующие функции по организации работы по охране труда:

– анализ состояния условий и охраны труда, причин нарушений законодательства о труде и охране труда, производственного травматизма, профессиональной и производственно обусловленной заболеваемости;

– разработка и осуществление мероприятий по функционированию и совершенствованию системы управления охраной труда;

– проведение, в том числе совместно с представителями соответствующих подразделений и с участием представителей по охране труда профсоюза или иного представительного органа работников, проверок (обследований) состояния условий и охраны труда, санитарно-бытового обеспечения работников, соблюдения требований охраны труда при проведении технологических процессов, эксплуатации зданий, сооружений; станков, машин, механизмов, другого оборудования, транспортных средств, приспособлений, инструмента, средств коллективной и индивидуальной защиты;

– организация совместно с подразделениями проведения обучения и проверки знаний работников по вопросам охраны труда;

– подготовка с участием подразделений перечней действующих и подлежащих разработке инструкций по охране труда. Разработка программы вводного инструктажа по охране труда и его проведение;

– оказание организационно-методической помощи подразделениям.

Проведение инструктажа по охране труда. По характеру и времени проведения инструктаж по охране труда подразделяют на:

– вводный;

– первичный на рабочем месте;

– повторный;

– внеплановый;

– целевой.

Вводный инструктаж проводится с лицами при:

– приеме их на постоянную или временную работу в организацию;

– участии в производственном процессе, привлечении к работам в организации или на ее территории, выполнении работ по заданию организации (по заключенному с организацией договору).

Вводный инструктаж проводится также с работниками других организаций, в том числе командированными, при участии их в производственном процессе или выполнении работ на территории организации.

Вводный инструктаж проводится по утвержденной руководителем организации программе (инструкции), которая разрабатывается с учетом специфики деятельности организации на основании примерного перечня вопросов программы вводного инструктажа.

Вводный инструктаж проводит инженер по охране труда или специалист организации, на которого возложены эти обязанности.

Регистрация вводного инструктажа осуществляется в журнале регистрации вводного инструктажа по охране труда.

Первичный инструктаж на рабочем месте до начала работы проводят с лицами:

– принятыми на работу;

– переведенными из одного подразделения в другое или с одного объекта на другой;

– участвующими в производственном процессе, привлеченными к работам в организации или выполняющими работы по заданию организации (по заключенному с организацией договору).

Первичный инструктаж на рабочем месте проводится также с работниками других организаций, в том числе командированными, при участии их в производственном процессе или выполнении работ на территории организации.

С работниками других организаций, выполняющими работы на территории организации, данный инструктаж проводит руководитель работ при участии руководителя или специалиста организации, на территории которой проводятся работы.

Первичный инструктаж на рабочем месте проводится индивидуально с практическим показом безопасных приемов и методов труда. Первичный инструктаж допускается проводить с группой лиц, обслуживающих однотипное оборудование и в пределах общего рабочего места.

Первичный инструктаж на рабочем месте проводится по утвержденной руководителем организации программе, составленной с учетом особенностей производства (выполняемых работ) и требований нормативных правовых актов по охране труда, или по инструкциям по охране труда для профессий и видов работ.

Повторный инструктаж проводится не реже одного раза в шесть месяцев по программе первичного инструктажа на рабочем месте или по инструкциям по охране труда для профессий и видов работ.

Первичный инструктаж на рабочем месте и повторный инструктаж могут не проводиться с лицами, которые не заняты на работах по монтажу, эксплуатации, наладке, обслуживанию и ремонту оборудования, использованию инструмента, хранению и применению сырья и материалов (за исключением работ с повышенной опасностью).

Перечень профессий и должностей работников, освобождаемых от первичного и повторного инструктажей на рабочем месте, составляется службой охраны труда с участием профсоюза и утверждается руководителем организации.

Внеплановый инструктаж проводится при:

* принятии новых нормативных правовых актов, технических нормативных правовых актов, локальных нормативных правовых актов по охране труда или внесении изменений и дополнений к ним;
* изменении технологического процесса, замене или модернизации оборудования, приборов и инструмента, сырья, материалов и других факторов, влияющих на безопасность труда;
* нарушении нормативных правовых актов, технических нормативных правовых актов, локальных нормативных правовых актов по охране труда, которое привело или могло привести к аварии, несчастному случаю на производстве и другим тяжелым последствиям;
* перерывах в работе по профессии (в должности) более шести месяцев;
* поступлении информации об авариях и несчастных случаях, происшедших в однопрофильных организациях.

Внеплановый инструктаж проводится также по требованию представителей государственного органа надзора и контроля, вышестоящих государственных органов или государственных организаций, должностного лица организации, на которого возложены обязанности по обеспечению охраны труда, при нарушении нормативных правовых актов, технических нормативных правовых актов, локальных нормативных правовых актов по охране труда.

Внеплановый инструктаж проводится индивидуально или с группой лиц, работающих по одной профессии (должности). Объем и содержание инструктажа определяются в зависимости от причин и обстоятельств, вызвавших необходимость его проведения.

Целевой инструктаж проводят при:

* выполнении разовых работ, не связанных с прямыми обязанностями по специальности (погрузка, разгрузка, уборка территории и другие);
* ликвидации последствий аварий, стихийных бедствий и катастроф;
* производстве работ, на которые оформляется наряд-допуск;
* проведении экскурсий в организации;
* организации массовых мероприятий с учащимися (походы, спортивные соревнования и другие).

Периодический контроль за состоянием охраны труда на предприятии. Периодический контроль осуществляется представителями нанимателя с участием общественных инспекторов профсоюзов по охране труда. При отсутствии в организации профсоюза с согласия нанимателя для участия в проведении периодического контроля могут привлекаться уполномоченные лица по охране труда работников.

Периодический контроль, осуществляемый представителями нанимателя с участием общественных инспекторов профсоюзов по охране труда (уполномоченных лиц по охране труда работников), в зависимости от деятельности и структуры организации может проводиться:

– ежедневно – на участке, в смене, бригаде, лаборатории и иных аналогичных структурных подразделениях организации;

– ежемесячно – в цехе, отделе, иных аналогичных структурных подразделениях организации;

– ежеквартально – в организации в целом. Ежедневный контроль за состоянием охраны труда осуществляется руководителем структурного подразделения (мастером, начальником смены, заведующим лабораторией, механиком, другими руководителями участка) с участием общественного инспектора профсоюза по охране труда (уполномоченного лица по охране труда работников).

Ежедневно проверяются:

* состояние рабочих мест, проходов, переходов, проездов;
* безопасность технологического оборудования, оснастки и инструмента, грузоподъемных и транспортных средств;
* исправность вентиляционных систем и установок;
* наличие инструкций по охране труда и соблюдение их работниками;
* наличие и правильное использование средств индивидуальной защиты;
* выполнение мероприятий по устранению нарушений, выявленных предыдущими проверками;
* соблюдение работниками требований безопасности при выполнении работ, в том числе при - работе в электроустановках.

Ежемесячный контроль проводится начальником цеха с участием общественного инспектора профсоюза по охране труда (уполномоченного лица по охране труда работников), руководителей служб цеха и представителей службы охраны труда (инженера по охране труда).

В ходе ежемесячного контроля проверяются:

* организация и результаты ежедневного контроля;
* выполнение мероприятий, намеченных в результате проведения всех видов контроля, а также мероприятий, предусмотренных коллективным договором, соглашением, планом мероприятий по охране труда;
* исправность и соответствие производственного оборудования, транспортных средств и технологических процессов требованиям охраны труда;
* соблюдение работниками правил, норм и инструкций по охране труда;
* состояние рабочих мест, проходов, проездов, переходов и прилегающей к цеху территории;
* наличие и состояние защитных, сигнальных и противопожарных средств и устройств, контрольно-измерительных приборов;
* своевременность и качество проведения инструктажа работников по охране труда;
* наличие средств индивидуальной защиты и правильность их использования работниками;
* предоставление работникам компенсаций по условиям труда;
* состояние санитарно-бытовых помещений и устройств, обеспечение работников смывающими и обезвреживающими средствами;
* соблюдение установленного режима труда и отдыха, трудовой дисциплины;
* состояние кабинетов и уголков по охране труда, наличие и состояние учебных пособий и средств наглядной агитации.

Ежеквартальный контроль осуществляется руководителем организации (его заместителями) с участием руководителей служб, отделов, общественного инспектора по охране труда, (уполномоченного лица по охране труда работников).

В ходе ежеквартального контроля проверяются:

* организация и результаты проведения ежедневного и ежемесячного контроля;
* выполнение мероприятий по охране труда, предусмотренных коллективными договорами, соглашениями и другими локальными нормативными актами;
* выполнение мероприятий по документам расследования несчастных случаев на производстве и профессиональных заболеваний;
* своевременность проведения аттестации рабочих мест по условиям труда и выполнения мероприятий по ее результатам;

техническое состояние и содержание зданий, сооружений, помещений и прилегающей к ним территории в соответствии с требованиями охраны труда, состояние дорог, тротуаров, проходов и проездов.

Режим труда и отдыха. Работу за экраном монитора следует периодически прерывать на регламентированные перерывы, которые устанавливаются для обеспечения работоспособности и сохранения здоровья, или заменять другой работой с целью сокращения рабочей нагрузки у экрана.

Время регламентированных перерывов в течение рабочего дня (смены) устанавливается в зависимости от его (ее) продолжительности, вида и категории трудовой деятельности согласно приложению 1 к настоящей Инструкции.

При 8-часовой рабочей смене и работе с ПК регламентированные перерывы устанавливаются:

* для I категории работ через 2 часа от начала рабочей смены и через 2 часа после обеденного перерыва продолжительностью 15 минут каждый;
* для II категории работ через 2 часа от начала рабочей смены и   
  через 1,5-2 часа после обеденного перерыва продолжительностью 15 минут каждый или продолжительностью 10 минут через каждый час работы;
* для III категории работ через 1,5-2 часа от начала рабочей смены и через 1,5-2 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

При 12-часовой рабочей смене и работе с ПК регламентированные перерывы устанавливаются в первые 8 часов работы аналогично перерывам при 8-часовой рабочей смене, а в течение последних 4 часов работы, независимо от категории и вида работ, каждый час продолжительностью 15 минут. При работе с ПК в ночную смену (с 22.00 до 6.00) независимо от категории и вида трудовой деятельности суммарная продолжительность регламентированных перерывов увеличивается на 60 минут. Продолжительность непрерывной работы с ПК без регламентированного перерыва не должна превышать 2 часов. Во время регламентированных перерывов для снижения нервно-эмоционального напряжения, утомления зрительного анализатора, улучшения функционального состояния нервной, сердечно-сосудистой, дыхательной систем, а также мышц плечевого пояса, рук, спины, шеи и ног целесообразно выполнять комплексы упражнений.

Требования к организации и оборудованию рабочих мест с ВДТ, ЭВМ и ПЭВМ.

К работе с ПК допускаются работники, не имеющие медицинских противопоказаний, прошедшие инструктаж по вопросам охраны труда, с группой по электробезопасности не ниже I.

Женщины со времени установления беременности и в период кормления грудью к выполнению всех видов работ, связанных с использованием ПК, не допускаются.

Организация рабочего места с ПК должна учитывать требования безопасности, удобство положения, движений и действий работника. При работе с ПК работники обязаны:

соблюдать режим труда и отдыха, установленный законодательством, правилами внутреннего трудового распорядка организации, трудовую дисциплину, выполнять требования охраны труда, правил личной гигиены;

выполнять требования пожарной безопасности, знать порядок действий при пожаре, уметь применять первичные средства пожаротушения;

курить только в специально предназначенных для курения местах;

знать приемы оказания первой помощи при несчастных случаях на производстве;

о неисправностях оборудования и других замечаниях по работе с ПК сообщать непосредственному руководителю или лицам, осуществляющим техническое обслуживание оборудования.

Не допускается:

* выполнять работу, находясь в состоянии алкогольного опьянения либо в состоянии, вызванном употреблением наркотических средств, психотропных или токсических веществ, а также распивать спиртные напитки, употреблять наркотические средства, психотропные или токсические вещества на рабочем месте или в рабочее время;
* устанавливать системный блок в закрытых объемах мебели, непосредственно на полу;
* использовать для подключения ПК розетки, удлинители, не оснащенные заземляющим контактом (шиной).

Перед началом работы с ПК работник обязан:

* проветрить рабочее помещение, устойчивость положения оборудования на рабочем столе, отсутствие видимых повреждений оборудования, дискет в дисководе системного блока, исправность и целостность питающих и соединительных кабелей, разъемов и штепсельных соединений, защитного заземления (зануления), исправность мебели;
* отрегулировать положение стола, стула (кресла), подставки для ног, клавиатуры, экрана монитора;
* отрегулировать освещенность на рабочем месте. При необходимости включить местное освещение;
* протереть поверхность экрана монитора, защитного фильтра (при его наличии) сухой мягкой тканевой салфеткой;
* убедиться в отсутствии отражений на экране монитора, встречного светового потока;
* включить оборудование ПК в электрическую сеть, соблюдая следующую последовательность –­ стабилизатор напряжения (если он используется), блок бесперебойного питания, периферийные устройства (принтер, монитор, сканер и другие устройства), системный блок.

Запрещается приступать к работе при:

* выраженном дрожании изображения на мониторе;
* обнаружении неисправности оборудования;
* наличии поврежденных кабелей или проводов, разъемов, штепсельных соединений;
* отсутствии или неисправности защитного заземления (зануления) оборудования.

При работе с ПК не разрешается:

* при включенном питании прикасаться к панелям с разъемами оборудования, разъемам питающих и соединительных кабелей, экрану монитора;
* загромождать верхние панели оборудования, рабочее место бумагами, посторонними предметами;
* производить переключения, отключение питания во время выполнения активной задачи;
* допускать попадание влаги на поверхность оборудования;
* включать сильно охлажденное (принесенное с улицы в зимнее время) оборудование;
* производить самостоятельно вскрытие и ремонт оборудования;
* вытирать пыль на включенном оборудовании;
* допускать нахождение вблизи оборудования посторонних лиц.

Время регламентированных перерывов в течение рабочего дня устанавливается в зависимости от его продолжительности, вида и категории трудовой деятельности. Продолжительность непрерывной работы с ПК без регламентированного перерыва не должна превышать двух часов. В помещении должно быть произведено заземление компьютеров, а также заземление защитного фильтра на вентиляционную решетку системного блока. В помещении должны соблюдаться все условия для организации правильной работы на компьютере:

* расстояние от края стола до клавиатуры – 10–30 см;
* подъемно–поворотные кресла, регулируемые по высоте и углам наклона сидения и спинки;
* высота стола – 725 мм;
* длина стола – 800 – 1200 мм;
* ширина стола – 800 – 1000 мм;
* ширина кресла 400 мм;
* столы с выдвижной доской для клавиатуры.
* Минимальные размеры рабочего места одного оператора: площадь – 6 м2,
* объем – 20 м3.

Соблюдение требований производственной санитарии

Требования к микроклимату помещений при эксплуатации   
ВДТ, ЭВМ и ПЭВМ. В производственных помещениях, в которых работа   
на ВДТ, ЭВМ и ПЭВМ является основной, должны обеспечиваться оптимальные параметры микроклимата (таблица 5).

Таблица 5 – Оптимальные параметры микроклимата для помещений с ВДТ, ЭВМ и ЭВМ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Период года | Категория работ | Температура воздуха, оС  не более | Относительная влажность воздуха, % | Скорость движения  воздуха, м/с |
| Холодный | легкая-1а | 22-24 | 40-60 | 0,1 |
| легкая-1б | 21-23 | 40-60 | 0,1 |
| Теплый | легкая-1а | 23-25 | 40-60 | 0,1 |
| легкая-1б | 22-24 | 40-60 | 0,2 |

Требования к освещению помещений и рабочих мест с ВДТ, ЭВМ и ПЭВМ. Производственное освещение, правильно спроектированное и выполненное, улучшает условия зрительной работы, снижает утомление, способствует повышению производительности труда и качеству выпускаемой продукции; благоприятно влияет на производственную среду, оказывает положительное психологическое воздействие на работающих, повышает безопасность труда и снижает травматизм на производстве. В зависимости от источника света производственное освещение может быть искусственным, естественным и комбинированным. Искусственное освещение в помещениях эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. Допускается применение комбинированного освещения. Освещенность на рабочем месте в зоне расположения документов должно составлять 300-500 лк, нормы освещенности экрана 100–250 лк.

Требования к параметрам физических факторов. При работе с ПК на работников могут оказывать неблагоприятное воздействие следующие опасные и вредные производственные факторы:

* повышенный уровень электромагнитных излучений;
* повышенный уровень ионизирующих излучений;
* повышенный уровень шума и вибрации;
* повышенный уровень статического электричества;
* повышенная напряженность электростатического поля;
* повышенная или пониженная ионизация воздуха;
* повышенная яркость света;
* прямая и отраженная блесткость;
* повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека;
* статические перегрузки костно-мышечного аппарата и динамические локальные перегрузки мышц кистей рук;
* перенапряжение зрительного анализатора;
* умственное перенапряжение;
* эмоциональные перегрузки;
* монотонность труда.

В зависимости от условий труда, в которых применяются ПК, и характера работы на работников могут воздействовать также другие опасные и вредные производственные факторы. Допустимый уровень шума при работе без печатающего устройства 50 дБ, при включенном печатающем устройстве – 75 дБ. Печатающее устройство установлено на звукопоглощающую поверхность автономно от рабочего места оператора.

Общая вибрация с частотой 0,7 Гц не приводит к вибрационной болезни, а наиболее опасными являются от 2 до 30 Гц, т.к. они совпадают с собственной частотой внутренних органов и может привести к механическим повреждениям или разрыву.

Вентиляционные отверстия монитора не должны быть загорожены. Должны отсутствовать сгибы и защемления питающего кабеля. Расстояние между боковыми стенками компьютера – около 1,5 м, а между столами – 2 м.

Техника безопасности.

Электробезопасность. Виды помещений по степени опасности поражения током:

* помещения без повышенной опасности характеризуются нормальной температурой и влажностью, отсутствием пыли, наличием не токопроводящих полов;
* помещения с повышенной опасностью характеризуются наличием одного из следующих признаков - токопроводящих полов, повышенной температуры воздуха, повышенной относительной влажности воздуха, токопроводящей пыли на оборудовании и поводке, возможности прикосновения работающих одновременно и электрооборудованию и металлоконструкциям зданий или инженерному оборудованию имеющим связь с землей;
* помещения особо опасные характеризуются: сочетанием двух и более признаков помещений с повышенной опасностью, наличием особой сырости, наличием химически активной или органической пыли, разрушающе действующей на изоляцию.

Электрический ток представляет большую опасность для человека. Действие электрического тока на организм человека. Поражение человека электрическим током может произойти при:

* прикосновении к открытым токоведущим частям оборудования и проводам;
* прикосновении к корпусам электроустановок, случайно оказавшихся под напряжением (повреждение изоляции);
* шаговом напряжении;
* освобождении человека, находящегося под напряжением; действии электрической дуги.

Помещения, в которых находятся компьютеры, относятся к помещениям с условиями без повышенной опасности поражения людей электрическим током,   
т.е. отсутствие условий, создающих повышенную или особую опасность.

В соответствии с ГОСТом 12.1.019 в качестве средств и методов защиты от поражения электрическим током применяют:

* изоляцию токоведущих частей, проводов путем нанесения на них диэлектрического материала (пластмасс, резины, лаков, красок, эмалей и т.п.);
* двойную изоляцию, когда кроме рабочей изоляции на случай ее повреждения предусматривают дополнительную изоляцию (корпуса или ручки электроинструментов из диэлектрического материала, покрытие изолированных проводов общей нетокопроводной оболочкой и т.п.;
* изоляцию рабочего места (пола, настила);
* заземление или зануление корпусов электроустановок, которые могут оказаться под напряжением при повреждении изоляций.

В соответствии с ГОСТом 12.1.030-81 защитному заземле­нию или занулению подлежат:

* все электроустановки переменного тока при номинальном напряжении 380 В и выше и постоянного – 440 В и выше;
* электроустановки с номинальным напряжением переменного тока выше 42 В, использующиеся в условиях с повышенной опасностью и особо опасных;
* все электроустановки во взрывоопасных зонах.

Пожарная безопасность.

Пожарная безопасность – состояние объекта, при котором исключается возможность пожара, а в случае его возникновения предотвращается воздействие на людей опасных факторов пожара и обеспечивается защита материальных ценностей.

Пожарная безопасность объекта обеспечивается системами предотвращения пожара и противопожарной защиты, в том числе проведением организационно-технических мероприятий.

Для тушения пожара, возникшего в помещении, где находятся компьютеры, применяются переносные углекислотные огнетушители.

В них в качестве огнетушащего вещества применяют сжиженный диоксид углерода (углекислоту), а также аэрозольные и углекислотно-бромэтиловые, в качестве заряда в которых применяют галоидированные углеводороды, при подаче которых в зону горения тушение наступает при относительно высокой концентрации кислорода (14-18 %).

Углекислотно-бромэтиловые огнетушители ОУБ-3А и ОУБ-7А представляют собой стальные тонкостенные баллоны (толщина стенки 1,5-2 мм) сварной конструкции. В горловину баллона ввернута запорная головка рычажного типа с распыляющей насадкой и сифонной трубкой. Емкость баллонов соответственно 3,2 и 7,4 л.

Огнетушители аэрозольные (хладоновые) используют в тех же случаях, что и углекислотно-бромэтиловые. Огнетушащий состав хладон (фреон), 114В2, 13В1 в процессе пожаротушения не оказывает воздействия на защищаемые материалы и оборудование, что позволяет использовать данные огнетушители при тушении пожаров электронного оборудования, картин и музейных экспонатов. Наша промышленность выпускает огнетушители марок ОАХ, ОХ-3 и др.

При обнаружении пожара основными способами подавления горения (тушения пожара) являются:

* снижение концентрации кислорода в воздухе до пределов, при которых не может происходить горение и тление путем введения в зону горения разбавителей – инертных газов, паров, аэрозолей и других негорючих веществ;
* охлаждение зоны горения и горючих веществ ниже критической температуры (ниже 100 °С);
* изолирование горючих веществ от зоны горения;
* химическое торможение реакции горения;
* механический срыв пламени сильной струей воды или газа.

Для организации борьбы с огнем необходимо знать основные приемы тушения пожаров. При загорании изоляции проводов или обмоток электродвигателя, прежде всего надо обесточить электрическую сеть. Тушить пожары в электроустановках, находящихся под напряжением, при помощи воды или пенного огнетушителя недопустимо, для этого можно использовать углекислотные или порошковые огнетушители, также можно сбить пламя покрывалом. Приводить огнетушитель в действие следует по возможности ближе к месту очага пожара.

**6 Энерго- и ресурсосбережение**

Энергосбережение – это система мер, направленных на уменьшение потребления энергии путем внедрения новых энергосберегающих технологий и рационального использования энергоресурсов.

Экономия материальных ресурсов при разработке программного продукта может быть достигнута за счет следующих мероприятий:

– использования ждущего режима;

– использования спящего режима;

– использования LCD мониторов;

– приобщения к менее ресурсоемким передовым технологиям.

В настоящее время энергетическая проблема является одной из наиболее актуальных для Республики Беларусь. В комплексе мероприятий, направленных на повышение эффективности использования энергии, важнейшая роль отводится формированию информационно-образовательной системы по вопросам энергосбережения.

Режимы труда и отдыха при работе с ЭВМ, ПЭВМ и ВДТ должны определяться видом и категорией трудовой деятельности.

Виды трудовой деятельности разделяются на 3 группы:

– группа А – работа по считыванию информации с экрана ВДТ, ПЭВМ или ЭВМ с предварительным запросом;

– группа Б – работа по вводу информации;

– группа В – творческая работа в режиме диалога с ЭВМ, при выполнении в течение рабочей смены работ, относящихся к разным видам трудовой деятельности с ЭВМ, ПЭВМ и ВДТ следует принимать такую, которая занимает не менее 50% времени в течение рабочей смены или рабочего дня.

Для видов трудовой деятельности устанавливается 3 категории тяжести и напряженности работы с ВДТ, ПЭВМ и ЭВМ (таблица 5), которые определяются:

– для группы А – по суммарному числу считываемых знаков за рабочую смену, но не более 60000 знаков за смену;

– для группы Б – по суммарному числу считываемых или вводимых знаков за рабочую смену, но не более 40000 знаков за смену;

– для группы В – по суммарному времени непосредственной работы с ВДТ, ПЭВМ и ЭВМ за рабочую смену, но не более 6 часов за смену.

Стандарт управления энергопотреблением компьютеров, описывает три различных режима работы компьютера. Эти режимы отличаются потребляемой мощностью электроэнергии.

Ждущий режим позволяет экономить энергию, за счет отключения всех периферийных устройств, кроме ЦПУ, ОЗУ и материнской платы.

Время регламентированных перерывов в течение рабочей смены следует устанавливать в зависимости от ее продолжительности, вида и категории трудовой деятельности (таблица 5).

Таблица 5 – Время регламентированных перерывов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Категории работы с ВДТ, ЭВМ и ПЭВМ | Уровень нагрузки за рабочую смену при видах работ с ВДТ | | | Суммарное время  регламентированных  перерывов, мин | |
| группа А, количество знаков | группа Б, количество знаков | группа В, час | при 8-ми  часовой  смене | при 12-ти  часовой  смене |
| I | до 20000 | до 15000 | до 2,0 | 30 | 70 |
| II | до 40000 | до 30000 | до 4,0 | 50 | 90 |
| III | до 60000 | до 40000 | до 6,0 | 70 | 120 |

Основной режим – при этом режиме экономии нет. Потребляемая мощность около 400Ватт.

Для нахождения суммарной продолжительности регламентированных перерывов в течение всего времени разработки программного модуля используем формулу

Трп=n×tрп,(20)

где Трп – суммарная продолжительность регламентированных

перерывов в течение всего времени разработки

программного модуля, ч.

Для нахождения количества рабочих дней n, в течение которых разрабатывался программный продукт, используем формулу

n = Тпк /(8-tрп), (21)

где Тпк – время работы компьютера, ч;

tрп – суммарное время регламентированных перерывов, в течение

рабочего дня, ч.

n = 156,70/(8-1,5) = 24,11 д,

Трп = 24,11×1,5 = 36,17 ч

Стоимость сэкономленной электроэнергии Сэн рассчитывается по формуле

Сэн=Трп×(Wпк-Wсп)×Сэ, (22)

где Wпк – потребляемая мощность ПК, кВт;

Wпк = 0,4 кВт;

Wсп – потребляемая мощность компьютера в «спящем» режиме, кВт;

Wсп = 0,02 кВт;

Сэ – стоимость 1 кВт электроэнергии, руб.

Сэн = 36,17×(0,4 – 0,02)× 0,39 = 5,36 руб.

Использование мер по энерго- и ресурсосбережению, при разработке программного модуля, позволило сэкономить денежные средства в размере 5,36 руб.

### **Заключение**

В результате выполнения дипломного проекта была разработана программа для автоматизации работы железнодорожных маршрутов. Кроме этого были расширены знания из данной предметной области и укреплены знания языка программирования C#. При реализации проекта были пройдены этапы описания и постановки задачи автоматизации работы железнодорожных маршрутов, кодирования программы на алгоритмический язык и тестирования полученного приложения.

Программа позволяет:

– просмотр отчетов;

– экспорт простых и сложных отчетов;

– просмотр и занесение информации о продажах билетов;

– просмотр и занесение информации о техническом обслуживании железнодорожных маршрутов;

– просмотр и занесение информации о сотрудниках железнодорожных вокзалов;

– просмотр и занесение информации о маршрутах;

– программа обладает интуитивно-понятным интерфейсом, проста и удобна в использовании.

### **Список используемых источников**

ГОСТ ИСО/ МЭК 2382-99. Информационные технологии. Словарь. Ч. 1. Основные термины.

ГОСТ19.701-90 (ИСО 5807-85). Описание символов. Правила применения символов и выполнения схем.

СТУ 01-32-2019. Стандарт учреждения. Общие требования к оформлению текстовых документов».

1. Абрамян, М.А. Visual C# на примерах / М.А. Абрамян. - Москва: БХВ-Петербург, 2016.
2. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон; Пер. с англ. Мухин Н. – 2-е изд. – М.: ДМК Пресс, 2007. – 496 с.
3. Вагнер, Б. С# Эффективное программирование / Б. Вагнер. - Москва: ЛОРИ, 2017.
4. Дейтел, П. Как программировать на Visual C# 2012 / П. Дейтел. - Москва: Питер, 2016.
5. Зиборов, В.В. Visual C# 2012 на примерах / В.В. Зиборов. - Москва: БХВ-Петербург, 2015.
6. Ишкова, Э. А. Самоучитель С#. Начала программирования / Э.А. Ишкова.- Москва: Наука и техника, 2017.
7. Магда Ю. С. NI Measurement Studio. Практика разработки систем измерения и управления на C# / Ю.С. Магда. - Москва: ДМК Пресс, 2017.
8. Прайс, Д.Visual C# 2.0. Полное руководство / Д.Прайс. - Москва: Век +, Корона-Век, Энтроп, 2015.
9. Рихтер,Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C# / Д. Рихтер. - Москва: Питер, 2016.
10. Троелсен,Э.Язык программирования C#7 и платформы .NET и .NET Core/ Эндрю Троелсен,Филипп Джепикс – СПб.: ООО «Диалектика»,2018
11. Фленов, М.Е. Библия C# (+ CD-ROM) / М.Е. Фленов. - Москва: БХВ-Петербург, 2015.

# **Приложение А**

(обязательное)

**Входные и выходные формы**

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Автоматически созданное описание

Рисунок А.1 – Форма входа в систему

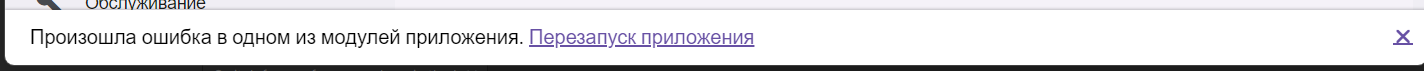


Рисунок А.2 – Вид сообщения об ошибке

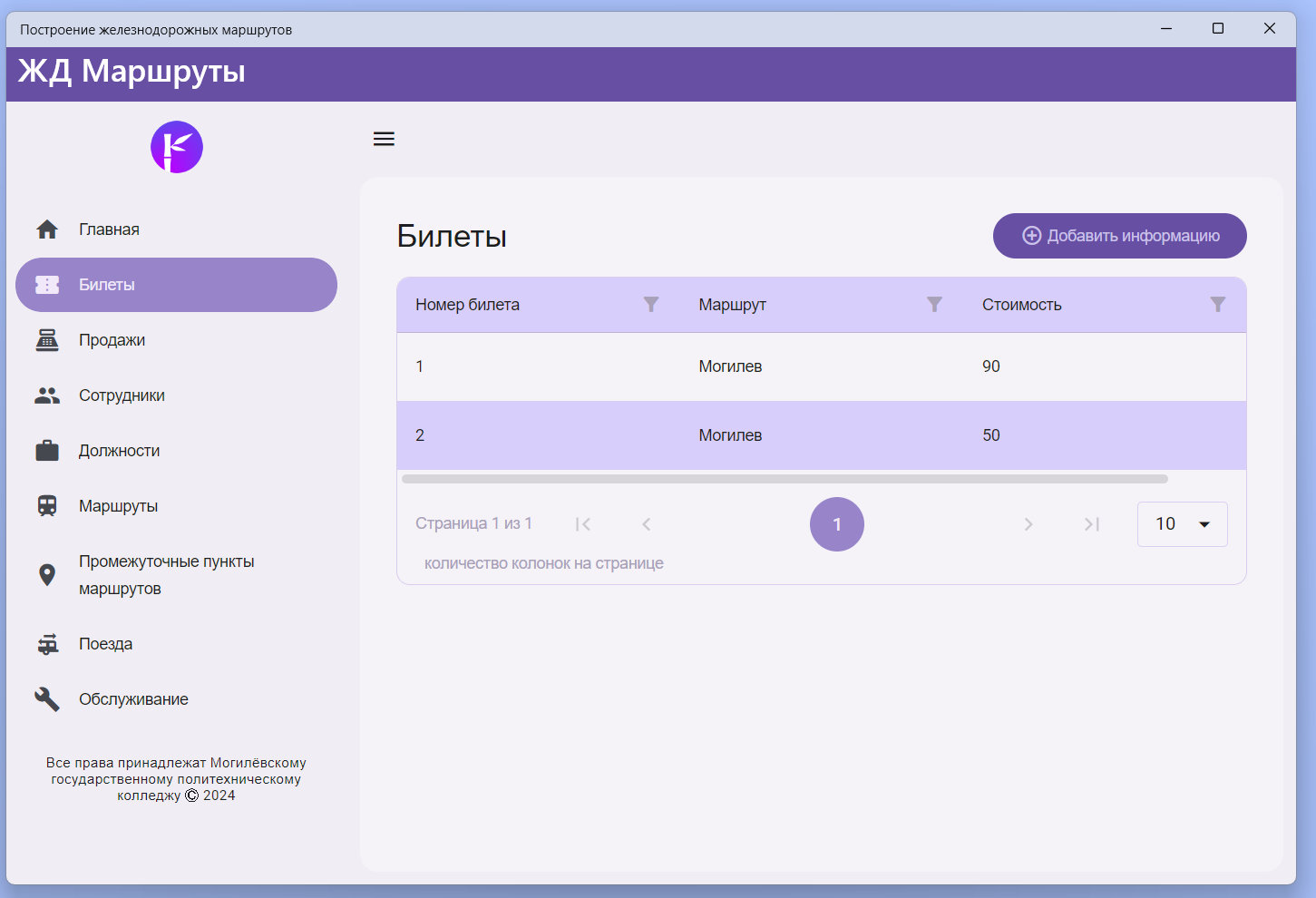


Рисунок А.3 – Форма «Билеты».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок А.4 – Форма «Продажа билетов».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок А.5 – Форма «Информация о сотрудниках».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок А.6 – Форма «Должности».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок А.7 – Форма «Информация о маршрутах».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок А.8 – Форма «Промежуточные пункты маршрутов».

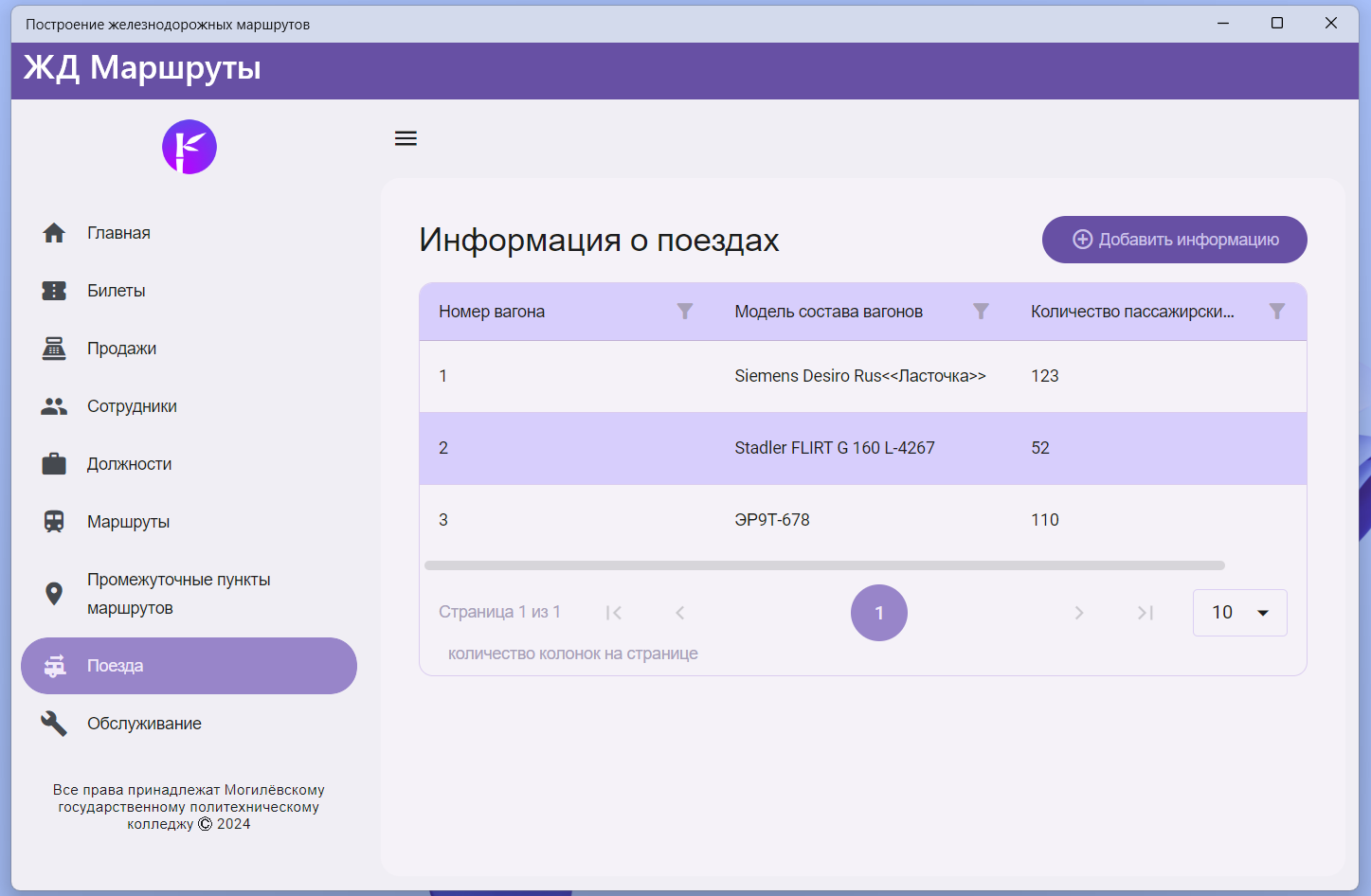


Рисунок А.9 – Форма «Поезда».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок А.10 – Форма «Техническое обслуживание»

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок А.11 – Форма «Информация о сотрудниках» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок А.12 – Форма «Промежуточные пункты маршрутов» на мобильном телефоне.

.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок А.13 – Форма «Техническое обслуживание» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок А.14 – Форма «Билеты» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Автоматически созданное описание

Рисунок А.15 – Форма «Продажа билетов» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок А.16 – Форма «Маршруты» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок А.17 – Форма «Должности» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, веб-страница, Шрифт

Автоматически созданное описание

Рисунок А.18 – Форма «Поезда» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок А.19 – Вид справки «О программе»

Приложение Б

(обязательное)

**Текст программы**

using CommunityToolkit.Maui;  
using CustomShellMaui;  
using DevExpress.Maui;  
using EntityFrameworkCore.UseRowNumberForPaging;  
using Material.Components.Maui.Extensions;  
using Microsoft.EntityFrameworkCore;  
using Microsoft.Extensions.Configuration;  
using Microsoft.AspNetCore.Components.Authorization;  
using Microsoft.AspNetCore.Identity;  
using Microsoft.Extensions.Logging;  
  
using Radzen;  
using TrainsMauiHybrid.Data;  
using TrainsMauiHybrid.Helpers;  
using TrainsMauiHybrid.Models;  
using TrainsMauiHybrid.Services;  
  
  
namespace TrainsMauiHybrid  
{  
 public static class MauiProgram  
 {  
 public static MauiApp CreateMauiApp()  
 {  
 var builder = MauiApp.CreateBuilder();  
 builder  
 .UseMauiApp<App>()  
 .UseMauiCommunityToolkit()  
 .UseCustomShellMaui()  
 .UseDevExpress()  
 .UseMaterialComponents()  
   
 .ConfigureFonts(fonts =>  
 {  
 fonts.AddFont("OpenSans-Regular.ttf", "OpenSansRegular");  
 });  
 builder.Services.AddHttpClient("Generator").ConfigurePrimaryHttpMessageHandler(() => new HttpClientHandler { UseCookies = false });  
   
 builder.Services.AddAuthentication();  
 builder.Services.AddAuthorizationCore();  
 builder.Services.AddScoped<TrainsMauiHybrid.Services.SecurityService>();  
 builder.Services.AddSingleton<TrainsMauiHybrid.Services.ThemeService>();  
 builder.Services.AddScoped<TrainsMauiHybrid.Services.DiplomnyProektService>();  
 builder.Services.AddIdentity<ApplicationUser, ApplicationRole>().AddEntityFrameworkStores<ApplicationIdentityDbContext>().AddDefaultTokenProviders();  
   
 builder.Services.AddScoped<AuthenticationStateProvider, TrainsMauiHybrid.Services.ApplicationAuthenticationStateProvider>();  
 builder.Services.AddDbContext<TrainsMauiHybrid.Data.DiplomnyProektContext>(options =>  
 {  
 options.UseSqlServer("Server=sql.bsite.net\\MSSQL2016;Persist Security Info=False;TrustServerCertificate=True; Encrypt=false;User ID=rsncra\_DiplomnyProekt;Password=ctrt55xx;Initial Catalog=rsncra\_DiplomnyProekt",  
 sqlServerOptionsAction: sqlOptions =>  
 {  
 sqlOptions.EnableRetryOnFailure();; // retry 6 times on fail  
 sqlOptions.UseRowNumberForPaging();  
 });  
 });  
 builder.Services.AddMauiBlazorWebView();  
 builder.Services.AddMasaBlazor();  
 builder.Services.AddRadzenComponents();  
   
 builder.Services.AddSingleton<EventHandleHelper>();  
  
#if DEBUG  
 builder.Services.AddBlazorWebViewDeveloperTools();  
 builder.Logging.AddDebug();  
#endif  
   
  
 return builder.Build();  
 }  
 }  
}

<Shell  
 x:Class="TrainsMauiHybrid.AppShell"  
 xmlns="http://schemas.microsoft.com/dotnet/2021/maui"  
 xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"  
 xmlns:local="clr-namespace:TrainsMauiHybrid"  
 xmlns:pages="clr-namespace:TrainsMauiHybrid.Components.Pages"  
 xmlns:views="clr-namespace:TrainsMauiHybrid.Views"  
 xmlns:platformSpecific="clr-namespace:TrainsMauiHybrid.Views.PlatformSpecific"  
 Title="Построение железнодорожных маршрутов"  
  
 Shell.FlyoutBehavior="Disabled"  
>  
 <Shell.TitleView>  
 <HorizontalStackLayout HorizontalOptions="Center">  
 <Label Text="ЖД Маршруты" FontSize="15" TextColor="{AppThemeBinding Light={StaticResource White},  
 Dark={StaticResource Primary}}" VerticalOptions="Center" HorizontalOptions="Center" />  
 </HorizontalStackLayout>  
 </Shell.TitleView>  
   
 <TabBar >  
 <Tab x:Name="HomeandroidTab" Icon="home.png" Title="Главная">  
 <ShellContent  
 Title="ЖД Маршруты"  
 IsVisible="{Binding isVisible\_Android}"  
 ContentTemplate="{DataTemplate platformSpecific:MainPagePhoneView}"  
 Route="MainPagePhoneView" />  
 <ShellContent  
 Title="ЖД Маршруты"  
 IsVisible="{Binding isVisible\_iOS}"  
 ContentTemplate="{DataTemplate platformSpecific:MainPagePhoneView}"  
 Route="MainPagePhoneView" />  
 <ShellContent  
 Title="ЖД Маршруты"  
 IsVisible="{Binding isVisible\_Desktop}"  
 ContentTemplate="{DataTemplate platformSpecific:MainPageDesktopView}"  
 Route="MainPageDesktopView" />  
   
 </Tab>  
   
 <!--<Tab x:Name="settingsandroidTab" Icon="appsetting.png" Title="Настройки">  
 <ShellContent  
 ContentTemplate="{DataTemplate views:SettingsPage}"  
 IsVisible="{Binding isVisible\_iOS}"  
 Route="SettingsPage"  
 Title="Настройки" />  
 <ShellContent  
 ContentTemplate="{DataTemplate views:SettingsPage}"  
 IsVisible="{Binding isVisible\_Android}"  
 Route="SettingsPage"  
 Title="Настройки" />  
 </Tab>-->  
 </TabBar>  
   
</Shell>

using CustomShellMaui.Enum;  
using TrainsMauiHybrid.Views;  
using TrainsMauiHybrid.Views.PlatformSpecific;  
  
namespace TrainsMauiHybrid;  
  
public partial class AppShell : Shell  
{  
 public bool isVisible\_Android { get; set; }  
  
 public bool isVisible\_iOS { get; set; }  
 public bool isVisible\_Desktop { get; set; }  
 public bool isVisible\_Mobileordesktop { get; set; }  
 public AppShell()  
 {  
 InitializeComponent();  
  
 Shell.Current.DisableSwipeBackiOS(true);  
  
 Routing.RegisterRoute(nameof(EditBioPage), typeof(EditBioPage));  
 Routing.RegisterRoute(nameof(SettingsPage), typeof(SettingsPage));  
 Routing.RegisterRoute(nameof(MainPagePhoneView), typeof(MainPagePhoneView));  
   
 Routing.RegisterRoute(nameof(MainPageDesktopView), typeof(MainPageDesktopView));  
  
 Routing.RegisterRoute("SettingsPage/EditBioPage", typeof(EditBioPage));  
 if (Microsoft.Maui.Devices.DeviceInfo.Platform == DevicePlatform.iOS )  
 {  
 isVisible\_iOS = true;  
 }  
 else if (DeviceInfo.Platform == DevicePlatform.Android)  
 {  
 isVisible\_Android = true;  
 }  
 else  
 {  
 isVisible\_Desktop = true;  
 }  
 BindingContext = this;  
  
 Shell.Current.CustomShellMaui(new CustomShellMaui.Models.Transitions  
 {  
 Root = new CustomShellMaui.Models.TransitionRoot  
 {  
 CurrentPage = TransitionType.FadeOut //is Above by default  
 },  
 Push = new CustomShellMaui.Models.Transition  
 {  
 CurrentPage = TransitionType.LeftOut,  
 NextPage = TransitionType.RightIn //is Above  
 },  
 Pop = new CustomShellMaui.Models.Transition  
 {  
 CurrentPage = TransitionType.RightOut, //is Above  
 NextPage = TransitionType.LeftIn  
 },  
 });  
 }  
}

<?xml version="1.0" encoding="utf-8" ?>  
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"  
 xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"  
 xmlns:local="clr-namespace:TrainsMauiHybrid.Views.PlatformSpecific"  
 xmlns:pages="clr-namespace:TrainsMauiHybrid.Components.Pages"  
 xmlns:icon="clr-namespace:IconPacks.IconKind;assembly=IconPacks.Material"  
 xmlns:md="clr-namespace:Material.Components.Maui;assembly=Material.Components.Maui"  
 xmlns:mds="clr-namespace:Material.Components.Maui.Styles;assembly=Material.Components.Maui"  
 xmlns:mdc="clr-namespace:Material.Components.Maui;assembly=Material.Components.Maui"  
 xmlns:toolkit="http://schemas.microsoft.com/dotnet/2022/maui/toolkit"  
 xmlns:trainsMauiHybrid="clr-namespace:TrainsMauiHybrid"  
 x:Class="TrainsMauiHybrid.Views.PlatformSpecific.MainPageDesktopView"  
 >  
 <ContentPage.Behaviors>  
 <toolkit:StatusBarBehavior StatusBarStyle="{ AppThemeBinding Light=LightContent, Dark=LightContent }"  
 StatusBarColor=  
 "{AppThemeBinding Light={StaticResource Primary},  
 Dark={StaticResource MaterialDark}}" />  
 </ContentPage.Behaviors>  
 <ContentPage.Content>  
 <BlazorWebView x:Name="blazorWebView" HostPage="wwwroot/index.html">  
 <BlazorWebView.RootComponents>  
 <RootComponent Selector="#app" ComponentType="{x:Type trainsMauiHybrid:Routes}" />  
 </BlazorWebView.RootComponents>  
 </BlazorWebView>  
 </ContentPage.Content>  
  
  
  
  
  
  
</ContentPage>

@using TrainsMauiHybrid.Helpers  
@using TrainsMauiHybrid.Services  
@using AlignItems = Radzen.AlignItems  
@using ButtonType = Radzen.ButtonType  
@using DialogService = Radzen.DialogService  
@using Orientation = Radzen.Orientation  
@using Variant = Radzen.Variant  
@inherits LayoutComponentBase  
  
  
  
<RadzenComponents />  
  
<RadzenLayout style="grid-template-areas: 'rz-sidebar rz-header' 'rz-sidebar rz-body';">  
 <RadzenHeader>  
 <RadzenRow JustifyContent="JustifyContent.Start" AlignItems="AlignItems.Center" Gap="0">  
 <RadzenColumn Size="5">  
 <RadzenSidebarToggle Click="@SidebarToggleClick"></RadzenSidebarToggle>  
 </RadzenColumn>  
 <RadzenColumn Size="7">  
 <RadzenStack AlignItems="AlignItems.Center" Orientation="Orientation.Horizontal" JustifyContent="JustifyContent.End"></RadzenStack>  
 </RadzenColumn>  
 </RadzenRow>  
 </RadzenHeader>  
 <RadzenBody Expanded="@sidebarExpanded">  
 <RadzenRow class="rz-mx-auto rz-px-4 rz-pt-2 rz-pt-md-4 rz-pt-lg-6 rz-pt-xl-12 rz-pb-2 rz-pb-lg-12" Style="max-width: 1440px;">  
 <RadzenColumn Size="12">  
 @Body  
 </RadzenColumn>  
 </RadzenRow>  
 </RadzenBody>  
 <RadzenSidebar Expanded="@sidebarExpanded" style="z-index: 2">  
 <RadzenStack AlignItems="Radzen.AlignItems.Center" class="rz-py-4 rz-py-lg-6" Style="padding: var(--rz-panel-menu-item-padding); border-bottom: var(--rz-panel-menu-item-border);">  
 <RadzenImage Path="images/logo.png" style="width: 48px; height: 48px;" AlternateText="Application logo"></RadzenImage>  
   
 </RadzenStack>  
 <RadzenPanelMenu DisplayStyle="Radzen.MenuItemDisplayStyle.IconAndText">  
 <RadzenPanelMenuItem Text="Главная" Path="" Icon="home" />  
 <RadzenPanelMenuItem Text="Билеты" Path="bileties" Icon="confirmation\_number" />  
 <RadzenPanelMenuItem Text="Продажи" Path="prodazhis" Icon="point\_of\_sale" />  
 <RadzenPanelMenuItem Text="Сотрудники" Path="employees" Icon="people" />  
 <RadzenPanelMenuItem Text="Должности" Path="jobs" Icon="work" />  
 <RadzenPanelMenuItem Text="Маршруты" Path="marshutis" Icon="train" />  
 <RadzenPanelMenuItem Text="Промежуточные пункты маршрутов" Path="promezhutochinye-punkties" Icon="location\_on" />  
 <RadzenPanelMenuItem Text="Поезда" Path="trains" Icon="rv\_hookup" />  
 <RadzenPanelMenuItem Text="Обслуживание" Path="obsluzhivanies" Icon="build" />  
 <RadzenPanelMenuItem Text="О программе" Path="aboutsoftware" Icon="info" />  
 </RadzenPanelMenu>  
 <RadzenStack AlignItems="Radzen.AlignItems.Center" Gap="0" class="rz-py-4 rz-py-lg-6" Style="padding: var(--rz-panel-menu-item-padding);">  
   
 <RadzenText Text="Все права принадлежат Могилёвскому государственному политехническому колледжу Ⓒ 2024" TextStyle="Radzen.Blazor.TextStyle.Caption" class="rz-mb-0" TagName="Radzen.Blazor.TagName.P" TextAlign="Radzen.TextAlign.Center" />  
 </RadzenStack>  
 </RadzenSidebar>  
</RadzenLayout>

using System.Net.NetworkInformation;  
using Microsoft.AspNetCore.Components;  
using Microsoft.JSInterop;  
using Radzen;  
using Radzen.Blazor;  
using TrainsMauiHybrid.Services;  
using TrainsMauiHybrid.Texts;  
using DialogService = Radzen.DialogService;  
  
namespace TrainsMauiHybrid.Components.Layout  
{  
 public partial class MainLayout  
 {  
 [Inject]  
 protected IJSRuntime JSRuntime { get; set; }  
  
 [Inject]  
 protected NavigationManager NavigationManager { get; set; }  
  
 [Inject]  
 protected DialogService DialogService { get; set; }  
  
 [Inject]  
 protected TooltipService TooltipService { get; set; }  
  
 [Inject]  
 protected ContextMenuService ContextMenuService { get; set; }  
  
 [Inject]  
 protected NotificationService NotificationService { get; set; }  
  
 private bool sidebarExpanded = true;  
  
 [Inject]  
 protected SecurityService Security { get; set; }  
  
 void SidebarToggleClick()  
 {  
 sidebarExpanded = !sidebarExpanded;  
 }  
  
 protected void ProfileMenuClick(RadzenProfileMenuItem args)  
 {  
 if (args.Value == "Logout")  
 {  
 Security.Logout();  
 }  
 }  
 }  
}

@using TrainsMauiHybrid.Helpers  
@using TrainsMauiHybrid.Services  
@using AlignItems = Radzen.AlignItems  
@using ButtonType = Radzen.ButtonType  
@using DialogService = Radzen.DialogService  
@using Orientation = Radzen.Orientation  
@using Variant = Radzen.Variant  
@inherits LayoutComponentBase  
@inject MasaBlazor MasaBlazor  
  
  
<RadzenComponents />  
  
<MApp >  
 <MErrorHandler ShowDetail="true">  
   
  
  
 @\*<MNavigationDrawer App @bind-Value="\_drawer">  
 <MList Nav Routable>  
 <MListItem Href="/" ActiveClass="primary--text">  
 <MListItemIcon>  
 <MIcon>mdi-home</MIcon>  
 </MListItemIcon>  
 <MListItemContent>  
 <MListItemTitle>Home</MListItemTitle>  
 </MListItemContent>  
 </MListItem>  
 <MListItem Href="/bileties" ActiveClass="primary--text">  
 <MListItemIcon>  
 <MIcon>mdi-plus</MIcon>  
 </MListItemIcon>  
 <MListItemContent>  
 <MListItemTitle>Обслуживание</MListItemTitle>  
 </MListItemContent>  
 </MListItem>  
 <MListItem Href="/bileties" ActiveClass="primary--text">  
 <MListItemIcon>  
 <MIcon>mdi-plus</MIcon>  
 </MListItemIcon>  
 <MListItemContent>  
 <MListItemTitle>Билеты</MListItemTitle>  
 </MListItemContent>  
 </MListItem>  
 <MListItem Href="/trains" ActiveClass="primary--text">  
 <MListItemIcon>  
 <MIcon>mdi-list-box</MIcon>  
 </MListItemIcon>  
 <MListItemContent>  
 <MListItemTitle>Поезда</MListItemTitle>  
 </MListItemContent>  
 </MListItem>  
 </MList>  
 </MNavigationDrawer>  
 \*@  
  
 @\*<MNavigationDrawer App @bind-Value="drawer">  
 <NavMenu />  
 </MNavigationDrawer>\*@  
  
  
  
 <MMain>  
 <!-- Provides the application the proper gutter -->  
 <MContainer Fluid>  
 @Body  
 </MContainer>  
 </MMain>  
  
 <MBottomNavigation Grow Color="purple" >  
  
  
 <MButton Href="employees">  
 <span>Cотрудники</span>  
 <MIcon>mdi-account</MIcon>  
 </MButton>  
 <MButton Href="jobs">  
 <span>Должности</span>  
 <MIcon>mdi-card-text</MIcon>  
 </MButton>  
 <MButton Href="marshutis">  
 <span>Маршруты</span>  
 <MIcon>mdi-train</MIcon>  
 </MButton>  
   
 <MMenu  
 Bottom  
 Left Transition="scale-transition" >  
 <ActivatorContent>  
 <MButton  
   
 Icon  
 @attributes="@context.Attrs">  
 <MIcon>mdi-dots-vertical</MIcon>  
 </MButton>  
 </ActivatorContent>  
   
 <ChildContent>  
 <MList>  
 <MListItem Class="m-list-item--default" Href="prodazhis">  
 <MListItemTitle>Продажи</MListItemTitle>  
 </MListItem>  
 <MListItem Class="m-list-item--default" Href="bileties">  
 <MListItemTitle>Билеты</MListItemTitle>  
 </MListItem>  
 <MListItem Class="m-list-item--default" Href="trains">  
 <MListItemTitle>Поезда</MListItemTitle>  
 </MListItem>  
 <MListItem Class="m-list-item--default" Href="obsluzhivanies">  
 <MListItemTitle>Обслуживание</MListItemTitle>  
 </MListItem>  
 <MListItem Class="m-list-item--default" Href="promezhutochinye-punkties">  
 <MListItemTitle>Промежуточные пункты маршрутов</MListItemTitle>  
 </MListItem>  
 <MListItem Class="m-list-item--default" Href="aboutsoftware">  
 <MListItemTitle>О программе</MListItemTitle>  
 </MListItem>  
   
 <MSwitch Value="\_isDark" TValue="bool" ValueChanged="IsDarkChanged" Label="@(\_isDark ? "Темная" : "Светлая")" />  
   
   
 </MList>  
 </ChildContent>  
 </MMenu>  
  
 @\*<MButton Href="prodazhis">  
 <span>Продажи</span>  
 <MIcon>mdi-ticket-confirmation</MIcon>  
 </MButton>  
 <MButton Href="bileties">  
 <span>Билеты</span>  
 <MIcon>mdi-passport</MIcon>  
 </MButton>  
 <MButton Href="trains">  
 <span>Поезда</span>  
 <MIcon>mdi-train-car-container</MIcon>  
 </MButton>  
 <MButton Href="obsluzhivanies">  
 <span>Обслуживание</span>  
 <MIcon>mdi-table-filter</MIcon>  
 </MButton>  
 <MButton Href="promezhutochinye-punkties">  
 <span>Промежуточные пункты маршрутов</span>  
 <MIcon>mdi-table-filter</MIcon>  
 </MButton>\*@  
  
 </MBottomNavigation>  
  
 </MErrorHandler>  
</MApp>

using System.Net.NetworkInformation;  
using Microsoft.AspNetCore.Components;  
using Microsoft.JSInterop;  
using Radzen;  
using TrainsMauiHybrid.Services;  
using TrainsMauiHybrid.Texts;  
using DialogService = Radzen.DialogService;  
  
namespace TrainsMauiHybrid.Components.Layout  
{  
 public partial class MainLayoutMobile  
 {  
 string[] items =  
 {  
 "Click Me",  
 "Click Me",  
 "Click Me",  
 "Click Me 2"  
 };  
 static AppTheme currentTheme = Application.Current.RequestedTheme;  
 string getwiptheme = currentTheme.ToString();  
 private bool \_drawer;  
 [Inject]  
 // Inject the theme service  
 private ThemeService ThemeService { get; set; }  
 [Inject]  
 protected IJSRuntime JSRuntime { get; set; }  
  
 [Inject]  
 protected NavigationManager NavigationManager { get; set; }  
  
 [Inject]  
 protected DialogService DialogService { get; set; }  
  
 [Inject]  
 protected TooltipService TooltipService { get; set; }  
  
 [Inject]  
 protected ContextMenuService ContextMenuService { get; set; }  
  
 [Inject]  
 protected NotificationService NotificationService { get; set; }  
  
 private bool sidebarExpanded = true;  
  
 void SidebarToggleClick()  
 {  
 sidebarExpanded = !sidebarExpanded;  
 }  
 void hidedrawer()  
 {  
 \_drawer = !\_drawer;  
 }  
 private bool \_isDark { get; set; }  
  
 private void IsDarkChanged(bool isDark)  
 {  
 var appTheme = ThemeService.GetAppTheme();  
 // Whether the theme of the system is DARK, assigning a value for the ISDARK attribute  
 \_isDark = appTheme == AppTheme.Dark;  
   
 Console.WriteLine(\_isDark.ToString());  
   
 MasaBlazor.ToggleTheme();  
   
 }  
   
   
   
  
 /// <summary>  
 /// Processing system theme switch  
 /// </summary>  
 /// <param name="sender"></param>  
 /// <param name="e"></param>  
 private void HandlerAppThemeChanged(object sender, AppThemeChangedEventArgs e)  
 {  
 \_isDark = e.RequestedTheme == AppTheme.Dark;  
 InvokeAsync(StateHasChanged);  
 }  
 protected override async Task OnAfterRenderAsync(bool firstRender)  
 {  
 if (firstRender)  
 {  
 // Get the system theme  
 var appTheme = ThemeService.GetAppTheme();  
 // Whether the theme of the system is DARK, assigning a value for the ISDARK attribute  
 \_isDark = appTheme == AppTheme.Dark;  
 Console.WriteLine(\_isDark.ToString());  
 ThemeService.ThemeChanged(HandlerAppThemeChanged);  
 StateHasChanged();  
 }  
 await base.OnAfterRenderAsync(firstRender);  
 }  
 }  
}

@page "/aboutsoftware"  
@inject IJSRuntime JSRuntime  
@inject NavigationManager NavigationManager  
@inject DialogService DialogService  
@inject ContextMenuService ContextMenuService  
@inject TooltipService TooltipService  
@inject NotificationService NotificationService  
  
<PageTitle>О программе</PageTitle>  
<RadzenRow>  
 <RadzenColumn Size="12">  
 <RadzenText Text="О программе" TextStyle="TextStyle.H3" TagName="TagName.H1" />  
 </RadzenColumn>  
</RadzenRow>  
<RadzenCard>  
   
 <RadzenText Text="Автоматизированная система «Железнодорожные маршруты»" Style="text-align: center; text-transform: none; font-size: 25px; line-height: 35px; height: 199px; position: relative" />  
 <RadzenAlert Text="Программа создана для дипломной работы по теме РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ «ПОСТРОЕНИЕ ЖЕЛЕЗНОДОРОЖНЫХ МАРШРУТОВ» в 2024 году" AlertStyle="Radzen.AlertStyle.Success" ShowIcon="true" AllowClose="false" Shade="Radzen.Shade.Dark"></RadzenAlert>  
 <RadzenAlert Text="Создано учащимся учебной группы ПО-455 Савичем Андреем Олеговичем" AllowClose="false" AlertStyle="Radzen.AlertStyle.Success"></RadzenAlert>  
 <RadzenAlert AllowClose="false" AlertStyle="Radzen.AlertStyle.Primary" Text="Все права принадлежат Могилевскому государственному политехническому колледжу" Size="Radzen.AlertSize.Large"></RadzenAlert>  
   
</RadzenCard>  
@code {  
  
}

@page "/bileties"  
@using TrainsMauiHybrid.Services  
@inject IJSRuntime JSRuntime  
@inject NavigationManager NavigationManager  
@inject DialogService DialogService  
@inject ContextMenuService ContextMenuService  
@inject TooltipService TooltipService  
@inject NotificationService NotificationService  
  
@inject DiplomnyProektService rsncra\_DiplomnyProektService  
  
<PageTitle>Билеты</PageTitle>  
<RadzenStack>  
 <RadzenRow AlignItems="AlignItems.Center">  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenText Text="Билеты" TextStyle="TextStyle.H3" TagName="TagName.H1" style="margin: 0" />  
 </RadzenColumn>  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenStack Orientation="Orientation.Horizontal" AlignItems="AlignItems.Center" JustifyContent="JustifyContent.End" Gap="0.5rem">  
 <RadzenButton Icon="add\_circle\_outline" Text="Добавить информацию" Click="@AddButtonClick" Variant="Variant.Flat" />  
 </RadzenStack>  
 </RadzenColumn>  
 </RadzenRow>  
 <RadzenRow>  
 <RadzenColumn SizeMD=12>  
 <RadzenDataGrid @ref="grid0" ColumnWidth="350px" AllowFiltering="true" FilterMode="FilterMode.Advanced" AllowPaging="true" AllowSorting="true"  
 PageTitleFormat="Страница {0}" PagingSummaryFormat="Страница {0} из {1} " PageSizeText="количество колонок на странице" Responsive="true"  
 ShowPagingSummary="true" PageSizeOptions=@(new int[]{5, 10, 20, 30} )  
 Data="@bileties" TItem="TrainsMauiHybrid.Models.Bilety" RowSelect="@EditRow">  
 <Columns>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Bilety" Property="Nomer\_Bileta" Title="Номер билеты">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Bilety" Property="Marshuti.Nachalni\_Punkt" Title="Маршруты">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Bilety" Property="Stoimost" Title="Стоимость">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Bilety" Property="Konechnaya\_Ostanovka" Title="Конечная остановка пассажира">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Bilety" Filterable="false" Sortable="false" Width="70px" TextAlign="TextAlign.Center">  
 <Template Context="bilety">  
  
 <RadzenButton ButtonStyle="ButtonStyle.Danger" Icon="delete" Size="ButtonSize.Medium"  
 Shade="Shade.Lighter" Variant="Variant.Flat"  
 Click=@(args => GridDeleteButtonClick(args, bilety)) @onclick:stopPropagation="true" />  
 </Template>  
 </RadzenDataGridColumn>  
 </Columns>  
  
 </RadzenDataGrid>  
  
 </RadzenColumn>  
 </RadzenRow>  
</RadzenStack>  
@code {  
  
 protected IEnumerable<TrainsMauiHybrid.Models.Bilety> bileties;  
  
 protected RadzenDataGrid<TrainsMauiHybrid.Models.Bilety> grid0;  
 protected override async Task OnInitializedAsync()  
 {  
 bileties = await rsncra\_DiplomnyProektService.GetBileties(new Query { Expand = "Marshuti" });  
 }  
  
 protected async Task AddButtonClick(MouseEventArgs args)  
 {  
 await DialogService.OpenAsync<AddBilety>("Добавить информацию о билетах", null);  
 await grid0.Reload();  
 }  
  
 protected async Task EditRow(TrainsMauiHybrid.Models.Bilety args)  
 {  
 await DialogService.OpenAsync<EditBilety>("Редактировать информацию о билетах", new Dictionary<string, object> { {"Nomer\_Bileta", args.Nomer\_Bileta} });  
 }  
  
 protected async Task GridDeleteButtonClick(MouseEventArgs args, TrainsMauiHybrid.Models.Bilety bilety)  
 {  
 try  
 {  
 if (await DialogService.Confirm("Are you sure you want to delete this record?") == true)  
 {  
 var deleteResult = await rsncra\_DiplomnyProektService.DeleteBilety(bilety.Nomer\_Bileta);  
  
 if (deleteResult != null)  
 {  
 await grid0.Reload();  
 }  
 }  
 }  
 catch (Exception ex)  
 {  
 NotificationService.Notify(new NotificationMessage  
 {  
 Severity = NotificationSeverity.Error,  
 Summary = $"Error",  
 Detail = $"Unable to delete Bilety"  
 });  
 }  
 }  
}

@page "/employees"  
@using TrainsMauiHybrid.Services  
@inject IJSRuntime JSRuntime  
@inject NavigationManager NavigationManager  
@inject DialogService DialogService  
@inject ContextMenuService ContextMenuService  
@inject TooltipService TooltipService  
@inject NotificationService NotificationService  
  
@inject DiplomnyProektService DiplomnyProektService  
@attribute [Authorize]  
  
  
<PageTitle>Сотрудники</PageTitle>  
  
<ErrorBoundary>  
<RadzenStack>  
 <RadzenRow AlignItems="AlignItems.Center">  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenText Text="Информация о сотрудниках" TextStyle="TextStyle.H3" TagName="TagName.H1" style="margin: 0" />  
 </RadzenColumn>  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenStack Orientation="Orientation.Horizontal" AlignItems="AlignItems.Center" JustifyContent="JustifyContent.End" Gap="0.5rem">  
 <RadzenButton Icon="add\_circle\_outline" Text="Добавить информацию" Click="@AddButtonClick" Variant="Variant.Flat" />  
 </RadzenStack>  
 </RadzenColumn>  
 </RadzenRow>  
 <RadzenRow>  
 <RadzenColumn SizeMD=12>  
 <RadzenDataGrid @ref="grid0" ColumnWidth="250px" AllowFiltering="true" FilterMode="FilterMode.Advanced" AllowPaging="true" AllowSorting="true"  
 PageTitleFormat="Страница {0}" PagingSummaryFormat="Страница {0} из {1} " PageSizeText="количество колонок на странице" Responsive="true"  
 ShowPagingSummary="true" PageSizeOptions=@(new int[]{5, 10, 20, 30} )  
 Data="@employees" TItem="TrainsMauiHybrid.Models.Employee" RowSelect="@EditRow">  
 <Columns>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Employee" Property="Num" Title="Номер сотрудника">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Employee" Property="Surname" Title="Фамилия">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Employee" Property="Name" Title="Имя">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Employee" Property="Patronym" Title="Отчество">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Employee" Property="Employed\_Since" Title="Работает с (числа)">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Employee" Property="Job.Job1" Title="Должность">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Employee" Filterable="false" Sortable="false" Width="70px" TextAlign="TextAlign.Center">  
 <Template Context="employee">  
  
 <RadzenButton ButtonStyle="ButtonStyle.Danger" Icon="delete" Size="ButtonSize.Medium"  
 Shade="Shade.Lighter" Variant="Variant.Flat"  
 Click=@(args => GridDeleteButtonClick(args, employee)) @onclick:stopPropagation="true" />  
 </Template>  
 </RadzenDataGridColumn>  
 </Columns>  
  
 </RadzenDataGrid>  
  
 </RadzenColumn>  
 </RadzenRow>  
</RadzenStack>  
</ErrorBoundary>  
  
@code {  
 [Inject]  
 protected SecurityService Security { get; set; }  
  
 protected IEnumerable<TrainsMauiHybrid.Models.Employee> employees;  
  
 protected RadzenDataGrid<TrainsMauiHybrid.Models.Employee> grid0;  
   
 protected override async Task OnInitializedAsync()  
 {  
 employees = await DiplomnyProektService.GetEmployees(new Query { Expand = "Job" });  
 }  
   
   
  
 protected async Task AddButtonClick(MouseEventArgs args)  
 {  
 await DialogService.OpenAsync<AddEmployee>("Добавить информацию о сотруднике", null);  
 await grid0.Reload();  
 }  
  
 protected async Task EditRow(TrainsMauiHybrid.Models.Employee args)  
 {  
 await DialogService.OpenAsync<EditEmployee>("Редактировать информацию о сотруднике", new Dictionary<string, object> { {"Num", args.Num} });  
 }  
  
 protected async Task GridDeleteButtonClick(MouseEventArgs args, TrainsMauiHybrid.Models.Employee employee)  
 {  
 try  
 {  
 if (await DialogService.Confirm("Вы уверены что хотите удалить данную строку?") == true)  
 {  
 var deleteResult = await DiplomnyProektService.DeleteEmployee(employee.Num);  
  
 if (deleteResult != null)  
 {  
 await grid0.Reload();  
 }  
 }  
 }  
 catch (Exception ex)  
 {  
 NotificationService.Notify(new NotificationMessage  
 {  
 Severity = NotificationSeverity.Error,  
 Summary = $"Ошибка",  
 Detail = $"Не удалось удалить Employee"  
 });  
 }  
 }  
}

@page "/jobs"  
@using TrainsMauiHybrid.Services  
@inject IJSRuntime JSRuntime  
@inject NavigationManager NavigationManager  
@inject DialogService DialogService  
@inject ContextMenuService ContextMenuService  
@inject TooltipService TooltipService  
@inject NotificationService NotificationService  
  
@inject DiplomnyProektService DiplomnyProektService  
@attribute [Authorize]  
<PageTitle>Должности</PageTitle>  
<ErrorBoundary>  
<RadzenStack>  
 <RadzenRow AlignItems="AlignItems.Center">  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenText Text="Должности" TextStyle="TextStyle.H3" TagName="TagName.H1" style="margin: 0" />  
 </RadzenColumn>  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenStack Orientation="Orientation.Horizontal" AlignItems="AlignItems.Center" JustifyContent="JustifyContent.End" Gap="0.5rem">  
 <RadzenButton Icon="add\_circle\_outline" Text="Добавить информацию" Click="@AddButtonClick" Variant="Variant.Flat" />  
 </RadzenStack>  
 </RadzenColumn>  
 </RadzenRow>  
 <RadzenRow>  
 <RadzenColumn SizeMD=12>  
 <RadzenDataGrid @ref="grid0" ColumnWidth="250px" AllowFiltering="true" FilterMode="FilterMode.Advanced" AllowPaging="true" AllowSorting="true"  
 PageTitleFormat="Страница {0}" PagingSummaryFormat="Страница {0} из {1} " PageSizeText="количество колонок на странице" Responsive="true"  
 ShowPagingSummary="true" PageSizeOptions=@(new int[]{5, 10, 20, 30} )  
 Data="@jobs" TItem="TrainsMauiHybrid.Models.Job" RowSelect="@EditRow">  
 <Columns>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Job" Property="Job\_Num" Title="Номер должности">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Job" Property="Job1" Title="Название должности">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Job" Property="Internship" Title="Необходима ли стажировка">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Job" Filterable="false" Sortable="false" Width="70px" TextAlign="TextAlign.Center">  
 <Template Context="job">  
  
 <RadzenButton ButtonStyle="ButtonStyle.Danger" Icon="delete" Size="ButtonSize.Medium"  
 Shade="Shade.Lighter" Variant="Variant.Flat"  
 Click=@(args => GridDeleteButtonClick(args, job)) @onclick:stopPropagation="true" />  
 </Template>  
 </RadzenDataGridColumn>  
 </Columns>  
  
 </RadzenDataGrid>  
  
 </RadzenColumn>  
 </RadzenRow>  
</RadzenStack>  
</ErrorBoundary>  
@code {  
 [Inject]  
 protected SecurityService Security { get; set; }  
  
 protected IEnumerable<TrainsMauiHybrid.Models.Job> jobs;  
  
 protected RadzenDataGrid<TrainsMauiHybrid.Models.Job> grid0;  
 protected override async Task OnInitializedAsync()  
 {  
 jobs = await DiplomnyProektService.GetJobs();  
 }  
  
 protected async Task AddButtonClick(MouseEventArgs args)  
 {  
 await DialogService.OpenAsync<AddJob>("Добавить должность", null);  
 await grid0.Reload();  
 }  
  
 protected async Task EditRow(TrainsMauiHybrid.Models.Job args)  
 {  
 await DialogService.OpenAsync<EditJob>("Изменить информацию о должности", new Dictionary<string, object> { {"Job\_Num", args.Job\_Num} });  
 }  
  
 protected async Task GridDeleteButtonClick(MouseEventArgs args, TrainsMauiHybrid.Models.Job job)  
 {  
 try  
 {  
 if (await DialogService.Confirm("Вы уверены что хотите удалить данную строку?") == true)  
 {  
 var deleteResult = await DiplomnyProektService.DeleteJob(job.Job\_Num);  
  
 if (deleteResult != null)  
 {  
 await grid0.Reload();  
 }  
 }  
 }  
 catch (Exception ex)  
 {  
 NotificationService.Notify(new NotificationMessage  
 {  
 Severity = NotificationSeverity.Error,  
 Summary = $"Ошибка",  
 Detail = $"Не удалось удалить Job"  
 });  
 }  
 }  
}

@page "/marshutis"  
@using TrainsMauiHybrid.Services  
@inject IJSRuntime JSRuntime  
@inject NavigationManager NavigationManager  
@inject DialogService DialogService  
@inject ContextMenuService ContextMenuService  
@inject TooltipService TooltipService  
@inject NotificationService NotificationService  
  
@inject DiplomnyProektService DiplomnyProektService  
@attribute [Authorize]  
<PageTitle>Маршруты</PageTitle>  
<ErrorBoundary>  
<RadzenStack>  
 <RadzenRow AlignItems="AlignItems.Center">  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenText Text="Маршруты" TextStyle="TextStyle.H3" TagName="TagName.H1" style="margin: 0" />  
 </RadzenColumn>  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenStack Orientation="Orientation.Horizontal" AlignItems="AlignItems.Center" JustifyContent="JustifyContent.End" Gap="0.5rem">  
 <RadzenButton Icon="add\_circle\_outline" Text="Добавить информацию" Click="@AddButtonClick" Variant="Variant.Flat" />  
 </RadzenStack>  
 </RadzenColumn>  
 </RadzenRow>  
 <RadzenTextBox Placeholder="Поиск ..." style="display: block; width: 100%" @oninput="@Search" aria-label="Поиск по информации" />  
 <RadzenRow>  
 <RadzenColumn SizeMD=12>  
 <RadzenDataGrid @ref="grid0" ColumnWidth="250px" AllowFiltering="true" FilterMode="FilterMode.Advanced" AllowPaging="true" AllowSorting="true"  
 PageTitleFormat="Страница {0}" PagingSummaryFormat="Страница {0} из {1} " PageSizeText="количество колонок на странице" Responsive="true"  
 ShowPagingSummary="true" PageSizeOptions=@(new int[]{5, 10, 20, 30} )  
 Data="@marshutis" TItem="TrainsMauiHybrid.Models.Marshuti" RowSelect="@EditRow">  
 <Columns>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Marshuti" Property="Nomer\_Marshuta" Title="Номер маршрута">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Marshuti" Property="Nachalni\_Punkt" Title="Начальный пункт">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Marshuti" Property="PromezhutochinyePunkty.PunktPoint2" Title="Промежуточные пункты">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Marshuti" Property="Konechni\_Punkt" Title="Конечный пункт">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Marshuti" Property="Employee.Surname" Title="Сотрудник">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Marshuti" Property="Train.Model\_Sostava\_Vagonov" Title="Модель состава вагонов">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Marshuti" Property="Vremya\_Proezda" Title="Время проезда">  
 </RadzenDataGridColumn>  
   
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Marshuti" Filterable="false" Sortable="false" Width="70px" TextAlign="TextAlign.Center">  
 <Template Context="marshuti">  
  
 <RadzenButton ButtonStyle="ButtonStyle.Danger" Icon="delete" Size="ButtonSize.Medium"  
 Shade="Shade.Lighter" Variant="Variant.Flat"  
 Click=@(args => GridDeleteButtonClick(args, marshuti)) @onclick:stopPropagation="true" />  
 </Template>  
 </RadzenDataGridColumn>  
 </Columns>  
  
 </RadzenDataGrid>  
  
 </RadzenColumn>  
 </RadzenRow>  
</RadzenStack>  
</ErrorBoundary>  
@code {  
 [Inject]  
 protected SecurityService Security { get; set; }  
  
 protected IEnumerable<TrainsMauiHybrid.Models.Marshuti> marshutis;  
  
 protected RadzenDataGrid<TrainsMauiHybrid.Models.Marshuti> grid0;  
   
 protected string search = "";  
  
 protected async Task Search(ChangeEventArgs args)  
 {  
 search = $"{args.Value}";  
  
 await grid0.GoToPage(0);  
  
 marshutis = await DiplomnyProektService.GetMarshutis(new Query { Filter = $@"i => i.Nachalni\_Punkt.Contains(@0) || i.Konechni\_Punkt.Contains(@0) || i.Vremya\_Proezda.Contains(@0)", FilterParameters = new object[] { search }, Expand = "Employee,Train,PromezhutochinyePunkty" });  
 }  
 protected override async Task OnInitializedAsync()  
 {  
 marshutis = await DiplomnyProektService.GetMarshutis(new Query { Filter = $@"i => i.Nachalni\_Punkt.Contains(@0) || i.Konechni\_Punkt.Contains(@0) || i.Vremya\_Proezda.Contains(@0)", FilterParameters = new object[] { search }, Expand = "Employee,Train,PromezhutochinyePunkty" });  
 }  
  
 protected async Task AddButtonClick(MouseEventArgs args)  
 {  
 await DialogService.OpenAsync<AddMarshuti>("Добавить информацию о маршрутах", null);  
 await grid0.Reload();  
 }  
  
 protected async Task EditRow(TrainsMauiHybrid.Models.Marshuti args)  
 {  
 await DialogService.OpenAsync<EditMarshuti>("Редактировать информацию о маршрутах", new Dictionary<string, object> { {"Nomer\_Marshuta", args.Nomer\_Marshuta} });  
 }  
  
 protected async Task GridDeleteButtonClick(MouseEventArgs args, TrainsMauiHybrid.Models.Marshuti marshuti)  
 {  
 try  
 {  
 if (await DialogService.Confirm("Вы уверены что хотите удалить данную строку?") == true)  
 {  
 var deleteResult = await DiplomnyProektService.DeleteMarshuti(marshuti.Nomer\_Marshuta);  
  
 if (deleteResult != null)  
 {  
 await grid0.Reload();  
 }  
 }  
 }  
 catch (Exception ex)  
 {  
 NotificationService.Notify(new NotificationMessage  
 {  
 Severity = NotificationSeverity.Error,  
 Summary = $"Ошибка",  
 Detail = $"Ошибка удаления информации"  
 });  
 }  
 }  
}

@page "/obsluzhivanies"  
@using TrainsMauiHybrid.Services  
@inject IJSRuntime JSRuntime  
@inject NavigationManager NavigationManager  
@inject DialogService DialogService  
@inject ContextMenuService ContextMenuService  
@inject TooltipService TooltipService  
@inject NotificationService NotificationService  
  
@inject DiplomnyProektService DiplomnyProektService  
@attribute [Authorize]  
<PageTitle>Обслуживание автобусов</PageTitle>  
<ErrorBoundary>  
   
 <RadzenStack>  
 <RadzenRow AlignItems="AlignItems.Center">  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenText Text="Обслуживание поездов" TextStyle="TextStyle.H3" TagName="TagName.H1" style="margin: 0"/>  
 </RadzenColumn>  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenStack Orientation="Orientation.Horizontal" AlignItems="AlignItems.Center" JustifyContent="JustifyContent.End" Gap="0.5rem">  
 <RadzenButton Icon="add\_circle\_outline" Text="Добавить информацию" Click="@AddButtonClick" Variant="Variant.Flat"/>  
 </RadzenStack>  
 </RadzenColumn>  
 </RadzenRow>  
 <RadzenTextBox Placeholder="Поиск по информации ..." style="display: block; width: 100%" @oninput="@Search" aria-label="Search by all string columns"/>  
 <RadzenRow>  
 <RadzenColumn SizeMD=12>  
 <RadzenDataGrid @ref="grid0" ColumnWidth="250px" AllowFiltering="true" FilterMode="FilterMode.Advanced" AllowPaging="true" AllowSorting="true"  
 PageTitleFormat="Страница {0}" PagingSummaryFormat="Страница {0} из {1} " PageSizeText="количество колонок на странице" Responsive="true"  
 ShowPagingSummary="true" PageSizeOptions=@(new int[]{5, 10, 20, 30} )  
 Data="@obsluzhivanies" TItem="TrainsMauiHybrid.Models.Obsluzhivanie" RowSelect="@EditRow">  
 <Columns>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Obsluzhivanie" Property="NomerObsluzhivania" Title="Номер обслуживания">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Obsluzhivanie" Property="Train.Model\_Sostava\_Vagonov" Title="Модель состава вагонов">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Obsluzhivanie" Property="Data\_Poslednego\_Obsluzhivania" Title="Дата последнего обслуживания">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Obsluzhivanie" Property="Ingener\_Obsluzhivania" Title="Инженер обслуживания">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Obsluzhivanie" Property="Problemi\_Sostava" Title="Проблемы состава вагонов">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Obsluzhivanie" Property="Data\_Sledueschego\_Obsluzivania" Title="Дата следующего обслуживания">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Obsluzhivanie" Property="Goden\_K\_Doroge" Title="Годен к дороге">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Obsluzhivanie" Filterable="false" Sortable="false" Width="70px" TextAlign="TextAlign.Center">  
 <Template Context="obsluzhivanie">  
  
 <RadzenButton ButtonStyle="ButtonStyle.Danger" Icon="delete" Size="ButtonSize.Medium"  
 Shade="Shade.Lighter" Variant="Variant.Flat"  
 Click=@(args => GridDeleteButtonClick(args, obsluzhivanie)) @onclick:stopPropagation="true" />  
 </Template>  
 </RadzenDataGridColumn>  
 </Columns>  
  
 </RadzenDataGrid>  
  
 </RadzenColumn>  
 </RadzenRow>  
 </RadzenStack>  
  
</ErrorBoundary>  
@code {  
  
 [Inject]  
 protected SecurityService Security { get; set; }  
 protected IEnumerable<TrainsMauiHybrid.Models.Obsluzhivanie> obsluzhivanies;  
  
 protected RadzenDataGrid<TrainsMauiHybrid.Models.Obsluzhivanie> grid0;  
 protected string search = "";  
  
 protected async Task Search(ChangeEventArgs args)  
 {  
 search = $"{args.Value}";  
  
 await grid0.GoToPage(0);  
  
 obsluzhivanies = await DiplomnyProektService.GetObsluzhivanies(new Query { Filter = $@"i => i.Data\_Poslednego\_Obsluzhivania.Contains(@0) || i.Ingener\_Obsluzhivania.Contains(@0) || i.Problemi\_Sostava.Contains(@0) || i.Data\_Sledueschego\_Obsluzivania.Contains(@0) || i.Goden\_K\_Doroge.Contains(@0)", FilterParameters = new object[] { search }, Expand = "Train" });  
 }  
 protected override async Task OnInitializedAsync()  
 {  
 obsluzhivanies = await DiplomnyProektService.GetObsluzhivanies(new Query { Expand = "Train" });  
 }  
  
 protected async Task AddButtonClick(MouseEventArgs args)  
 {  
 await DialogService.OpenAsync<AddObsluzhivanie>("Добавить информацию об обслуживании", null);  
 await grid0.Reload();  
 }  
  
 protected async Task EditRow(TrainsMauiHybrid.Models.Obsluzhivanie args)  
 {  
 await DialogService.OpenAsync<EditObsluzhivanie>("Редактировать информацию об обслуживании", new Dictionary<string, object> { { "NomerObsluzhivania", args.NomerObsluzhivania } });  
 }  
  
 protected async Task GridDeleteButtonClick(MouseEventArgs args, TrainsMauiHybrid.Models.Obsluzhivanie obsluzhivanie)  
 {  
 try  
 {  
 if (await DialogService.Confirm("Вы уверены что хотите удалить данную строку?") == true)  
 {  
 var deleteResult = await DiplomnyProektService.DeleteObsluzhivanie(obsluzhivanie.NomerObsluzhivania);  
  
 if (deleteResult != null)  
 {  
 await grid0.Reload();  
 }  
 }  
 }  
 catch (Exception ex)  
 {  
 NotificationService.Notify(new NotificationMessage  
 {  
 Severity = NotificationSeverity.Error,  
 Summary = $"Ошибка",  
 Detail = $"Ошибка удаления информации"  
 });  
 }  
 }  
}

@page "/prodazhis"  
@using TrainsMauiHybrid.Services  
@inject IJSRuntime JSRuntime  
@inject NavigationManager NavigationManager  
@inject DialogService DialogService  
@inject ContextMenuService ContextMenuService  
@inject TooltipService TooltipService  
@inject NotificationService NotificationService  
  
@inject DiplomnyProektService DiplomnyProektService  
@attribute [Authorize]  
<PageTitle>Продажи билетов</PageTitle>  
<ErrorBoundary>  
<RadzenStack>  
 <RadzenRow AlignItems="AlignItems.Center">  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenText Text="Продажи билетов" TextStyle="TextStyle.H3" TagName="TagName.H1" style="margin: 0" />  
 </RadzenColumn>  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenStack Orientation="Orientation.Horizontal" AlignItems="AlignItems.Center" JustifyContent="JustifyContent.End" Gap="0.5rem">  
 <RadzenButton Icon="add\_circle\_outline" Text="Добавить информацию" Click="@AddButtonClick" Variant="Variant.Flat" />  
 </RadzenStack>  
 </RadzenColumn>  
 </RadzenRow>  
 <RadzenRow>  
 <RadzenColumn SizeMD=12>  
 <RadzenDataGrid @ref="grid0" ColumnWidth="250px" AllowFiltering="true" FilterMode="FilterMode.Advanced" AllowPaging="true" AllowSorting="true"  
 PageTitleFormat="Страница {0}" PagingSummaryFormat="Страница {0} из {1} " PageSizeText="количество колонок на странице" Responsive="true"  
 ShowPagingSummary="true" PageSizeOptions=@(new int[]{5, 10, 20, 30} )  
 Data="@prodazhis" TItem="TrainsMauiHybrid.Models.Prodazhi" RowSelect="@EditRow">  
 <Columns>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Prodazhi" Property="Num" Title="Номер">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Prodazhi" Property="Sale\_Date" Title="Дата продажи">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Prodazhi" Property="Bilety.Stoimost" Title="Стоимость">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Prodazhi" Filterable="false" Sortable="false" Width="70px" TextAlign="TextAlign.Center">  
 <Template Context="prodazhi">  
  
 <RadzenButton ButtonStyle="ButtonStyle.Danger" Icon="delete" Size="ButtonSize.Medium"  
 Shade="Shade.Lighter" Variant="Variant.Flat"  
 Click=@(args => GridDeleteButtonClick(args, prodazhi)) @onclick:stopPropagation="true" />  
 </Template>  
 </RadzenDataGridColumn>  
 </Columns>  
  
 </RadzenDataGrid>  
  
 </RadzenColumn>  
 </RadzenRow>  
</RadzenStack>  
</ErrorBoundary>  
@code {  
  
 [Inject]  
 protected SecurityService Security { get; set; }  
 protected IEnumerable<TrainsMauiHybrid.Models.Prodazhi> prodazhis;  
  
 protected RadzenDataGrid<TrainsMauiHybrid.Models.Prodazhi> grid0;  
 protected override async Task OnInitializedAsync()  
 {  
 prodazhis = await DiplomnyProektService.GetProdazhis(new Query { Expand = "Bilety" });  
 }  
  
 protected async Task AddButtonClick(MouseEventArgs args)  
 {  
 await DialogService.OpenAsync<AddProdazhi>("Добавить информацию об продажах", null);  
 await grid0.Reload();  
 }  
  
 protected async Task EditRow(TrainsMauiHybrid.Models.Prodazhi args)  
 {  
 await DialogService.OpenAsync<EditProdazhi>("Редактировать информацию об продажах", new Dictionary<string, object> { { "Num", args.Num } });  
 }  
  
 protected async Task GridDeleteButtonClick(MouseEventArgs args, TrainsMauiHybrid.Models.Prodazhi prodazhi)  
 {  
 try  
 {  
 if (await DialogService.Confirm("Вы уверены что хотите удалить данную строку?") == true)  
 {  
 var deleteResult = await DiplomnyProektService.DeleteProdazhi(prodazhi.Num);  
  
 if (deleteResult != null)  
 {  
 await grid0.Reload();  
 }  
 }  
 }  
 catch (Exception ex)  
 {  
 NotificationService.Notify(new NotificationMessage  
 {  
 Severity = NotificationSeverity.Error,  
 Summary = $"Ошибка",  
 Detail = $"Unable to delete Prodazhi"  
 });  
 }  
 }  
}

@page "/promezhutochinye-punkties"  
  
@using TrainsMauiHybrid.Services  
@inject IJSRuntime JSRuntime  
@inject NavigationManager NavigationManager  
@inject DialogService DialogService  
@inject ContextMenuService ContextMenuService  
@inject TooltipService TooltipService  
@inject NotificationService NotificationService  
  
@inject DiplomnyProektService DiplomnyProektService  
@attribute [Authorize]  
<PageTitle>Промежуточные пункты</PageTitle>  
<RadzenStack>  
 <RadzenRow AlignItems="AlignItems.Center">  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenText Text="Промежуточные пункты маршрутов" TextStyle="TextStyle.H3" TagName="TagName.H1" style="margin: 0" />  
 </RadzenColumn>  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenStack Orientation="Orientation.Horizontal" AlignItems="AlignItems.Center" JustifyContent="JustifyContent.End" Gap="0.5rem">  
 <RadzenButton Icon="add\_circle\_outline" Text="Добавить информацию" Click="@AddButtonClick" Variant="Variant.Flat" />  
 </RadzenStack>  
 </RadzenColumn>  
 </RadzenRow>  
 <RadzenTextBox Placeholder="Поиск ..." style="display: block; width: 100%" @oninput="@Search" aria-label="Search by all string columns" />  
 <RadzenRow>  
 <RadzenColumn SizeMD=12>  
 <RadzenDataGrid @ref="grid0" ColumnWidth="250px" AllowFiltering="true" FilterMode="FilterMode.Advanced" AllowPaging="true" AllowSorting="true" ShowPagingSummary="true" PageSizeOptions=@(new int[]{5, 10, 20, 30})  
 Data="@promezhutochinyePunkties" TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" RowSelect="@EditRow" PageTitleFormat="Страница {0}" PagingSummaryFormat="Страница {0} из {1} " PageSizeText="количество колонок на странице" Responsive="true">  
 <Columns>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" Property="PunktKey" Title="Номер пункта">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" Property="PunktPoint1" Title="Промежуточный пункт 1">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" Property="PunktPoint2" Title="Промежуточный пункт 2">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" Property="PunktPoint3" Title="Промежуточный пункт 3">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" Property="PunktPoint4" Title="Промежуточный пункт 4">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" Property="PunktPoint5" Title="Промежуточный пункт 5">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" Property="PunktPoint6" Title="Промежуточный пункт 6">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" Property="PunktPoint7" Title="Промежуточный пункт 7">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" Property="PunktPoint8" Title="Промежуточный пункт 8">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" Property="PunktPoint9" Title="Промежуточный пункт 9">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" Property="PunktPoint10" Title="Промежуточный пункт 10">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.PromezhutochinyePunkty" Filterable="false" Sortable="false" Width="70px" TextAlign="TextAlign.Center">  
 <Template Context="promezhutochinyePunkty">  
  
 <RadzenButton ButtonStyle="ButtonStyle.Danger" Icon="delete" Size="ButtonSize.Medium"  
 Shade="Shade.Lighter" Variant="Variant.Flat"  
 Click=@(args => GridDeleteButtonClick(args, promezhutochinyePunkty)) @onclick:stopPropagation="true" />  
 </Template>  
 </RadzenDataGridColumn>  
 </Columns>  
  
 </RadzenDataGrid>  
  
 </RadzenColumn>  
 </RadzenRow>  
</RadzenStack>  
  
@code {  
  
 [Inject]  
 protected SecurityService Security { get; set; }  
 protected IEnumerable<TrainsMauiHybrid.Models.PromezhutochinyePunkty> promezhutochinyePunkties;  
  
 protected RadzenDataGrid<TrainsMauiHybrid.Models.PromezhutochinyePunkty> grid0;  
  
 protected string search = "";  
  
 protected async Task Search(ChangeEventArgs args)  
 {  
 search = $"{args.Value}";  
  
 await grid0.GoToPage(0);  
  
 promezhutochinyePunkties = await DiplomnyProektService.GetPromezhutochinyePunkties(new Query { Filter = $@"i => i.PunktPoint1.Contains(@0) || i.PunktPoint2.Contains(@0) || i.PunktPoint3.Contains(@0) || i.PunktPoint4.Contains(@0) || i.PunktPoint5.Contains(@0) || i.PunktPoint6.Contains(@0) || i.PunktPoint7.Contains(@0) || i.PunktPoint8.Contains(@0) || i.PunktPoint9.Contains(@0) || i.PunktPoint10.Contains(@0)", FilterParameters = new object[] { search } });  
 }  
 protected override async Task OnInitializedAsync()  
 {  
 promezhutochinyePunkties = await DiplomnyProektService.GetPromezhutochinyePunkties(new Query { Filter = $@"i => i.PunktPoint1.Contains(@0) || i.PunktPoint2.Contains(@0) || i.PunktPoint3.Contains(@0) || i.PunktPoint4.Contains(@0) || i.PunktPoint5.Contains(@0) || i.PunktPoint6.Contains(@0) || i.PunktPoint7.Contains(@0) || i.PunktPoint8.Contains(@0) || i.PunktPoint9.Contains(@0) || i.PunktPoint10.Contains(@0)", FilterParameters = new object[] { search } });  
 }  
  
 protected async Task AddButtonClick(MouseEventArgs args)  
 {  
 await DialogService.OpenAsync<AddPromezhutochinyePunkty>("Добавить информацию о промежуточном пункте маршрута", null);  
 await grid0.Reload();  
 }  
  
 protected async Task EditRow(TrainsMauiHybrid.Models.PromezhutochinyePunkty args)  
 {  
 await DialogService.OpenAsync<EditPromezhutochinyePunkty>("Редактировать информацию о промежуточном пункте маршрута", new Dictionary<string, object> { {"PunktKey", args.PunktKey} });  
 }  
  
 protected async Task GridDeleteButtonClick(MouseEventArgs args, TrainsMauiHybrid.Models.PromezhutochinyePunkty promezhutochinyePunkty)  
 {  
 try  
 {  
 if (await DialogService.Confirm("Вы уверены что хотите удалить данную строку?") == true)  
 {  
 var deleteResult = await DiplomnyProektService.DeletePromezhutochinyePunkty(promezhutochinyePunkty.PunktKey);  
  
 if (deleteResult != null)  
 {  
 await grid0.Reload();  
 }  
 }  
 }  
 catch (Exception ex)  
 {  
 NotificationService.Notify(new NotificationMessage  
 {  
 Severity = NotificationSeverity.Error,  
 Summary = $"Error",  
 Detail = $"Unable to delete PromezhutochinyePunkty"  
 });  
 }  
 }  
  
}

@page "/trains"  
@using TrainsMauiHybrid.Services  
@inject IJSRuntime JSRuntime  
@inject NavigationManager NavigationManager  
@inject DialogService DialogService  
@inject ContextMenuService ContextMenuService  
@inject TooltipService TooltipService  
@inject NotificationService NotificationService  
  
@inject DiplomnyProektService DiplomnyProektService  
@attribute [Authorize]  
<PageTitle>Информация о поездах</PageTitle>  
<ErrorBoundary>  
  
<RadzenStack>  
 <RadzenRow AlignItems="AlignItems.Center">  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenText Text="Информация о поездах" TextStyle="TextStyle.H3" TagName="TagName.H1" style="margin: 0" />  
 </RadzenColumn>  
 <RadzenColumn Size="12" SizeMD="6">  
 <RadzenStack Orientation="Orientation.Horizontal" AlignItems="AlignItems.Center" JustifyContent="JustifyContent.End" Gap="0.5rem">  
 <RadzenButton Icon="add\_circle\_outline" Text="Добавить информацию" Click="@AddButtonClick" Variant="Variant.Flat" />  
 </RadzenStack>  
 </RadzenColumn>  
 </RadzenRow>  
 <RadzenRow>  
 <RadzenColumn SizeMD=12>  
 <RadzenDataGrid @ref="grid0" ColumnWidth="250px" AllowFiltering="true" FilterMode="FilterMode.Advanced" AllowPaging="true" AllowSorting="true"  
 PageTitleFormat="Страница {0}" PagingSummaryFormat="Страница {0} из {1} " PageSizeText="количество колонок на странице" Responsive="true"  
 ShowPagingSummary="true" PageSizeOptions=@(new int[]{5, 10, 20, 30} )  
 Data="@trains" TItem="TrainsMauiHybrid.Models.Train" RowSelect="@EditRow">  
 <Columns>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Train" Property="NumVagona" Title="Номер вагона">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Train" Property="Model\_Sostava\_Vagonov" Title="Модель состава вагонов">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Train" Property="Kolichestvo\_Mest" Title="Количество пассажирских мест">  
 </RadzenDataGridColumn>  
 <RadzenDataGridColumn TItem="TrainsMauiHybrid.Models.Train" Filterable="false" Sortable="false" Width="70px" TextAlign="TextAlign.Center">  
 <Template Context="train">  
  
 <RadzenButton ButtonStyle="ButtonStyle.Danger" Icon="delete" Size="ButtonSize.Medium"  
 Shade="Shade.Lighter" Variant="Variant.Flat"  
 Click=@(args => GridDeleteButtonClick(args, train)) @onclick:stopPropagation="true" />  
 </Template>  
 </RadzenDataGridColumn>  
 </Columns>  
  
 </RadzenDataGrid>  
  
 </RadzenColumn>  
 </RadzenRow>  
</RadzenStack>  
</ErrorBoundary>  
@code {  
 [Inject]  
 protected SecurityService Security { get; set; }  
  
 protected IEnumerable<TrainsMauiHybrid.Models.Train> trains;  
  
 protected RadzenDataGrid<TrainsMauiHybrid.Models.Train> grid0;  
 protected override async Task OnInitializedAsync()  
 {  
 trains = await DiplomnyProektService.GetTrains();  
 }  
  
 protected async Task AddButtonClick(MouseEventArgs args)  
 {  
 await DialogService.OpenAsync<AddTrain>("Добавить информацию о моделях состава вагонов", null);  
 await grid0.Reload();  
 }  
  
 protected async Task EditRow(TrainsMauiHybrid.Models.Train args)  
 {  
 await DialogService.OpenAsync<EditTrain>("Редактировать информацию о моделях состава вагонов", new Dictionary<string, object> { { "NumVagona", args.NumVagona } });  
 }  
  
 protected async Task GridDeleteButtonClick(MouseEventArgs args, TrainsMauiHybrid.Models.Train train)  
 {  
 try  
 {  
 if (await DialogService.Confirm("Вы уверены что хотите удалить данную строку?") == true)  
 {  
 var deleteResult = await DiplomnyProektService.DeleteTrain(train.NumVagona);  
  
 if (deleteResult != null)  
 {  
 await grid0.Reload();  
 }  
 }  
 }  
 catch (Exception ex)  
 {  
 NotificationService.Notify(new NotificationMessage  
 {  
 Severity = NotificationSeverity.Error,  
 Summary = $"Ошибка",  
 Detail = $"Unable to delete Train"  
 });  
 }  
 }  
}

Приложение В

(обязательное)

**Результаты тестирования**

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Автоматически созданное описание

Рисунок В.1 – Форма входа в систему

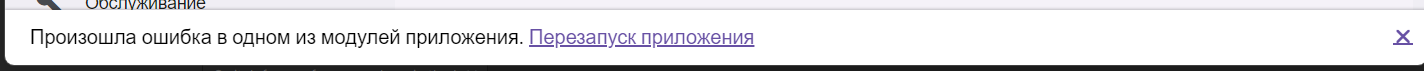


Рисунок В.2 – Вид сообщения об ошибке

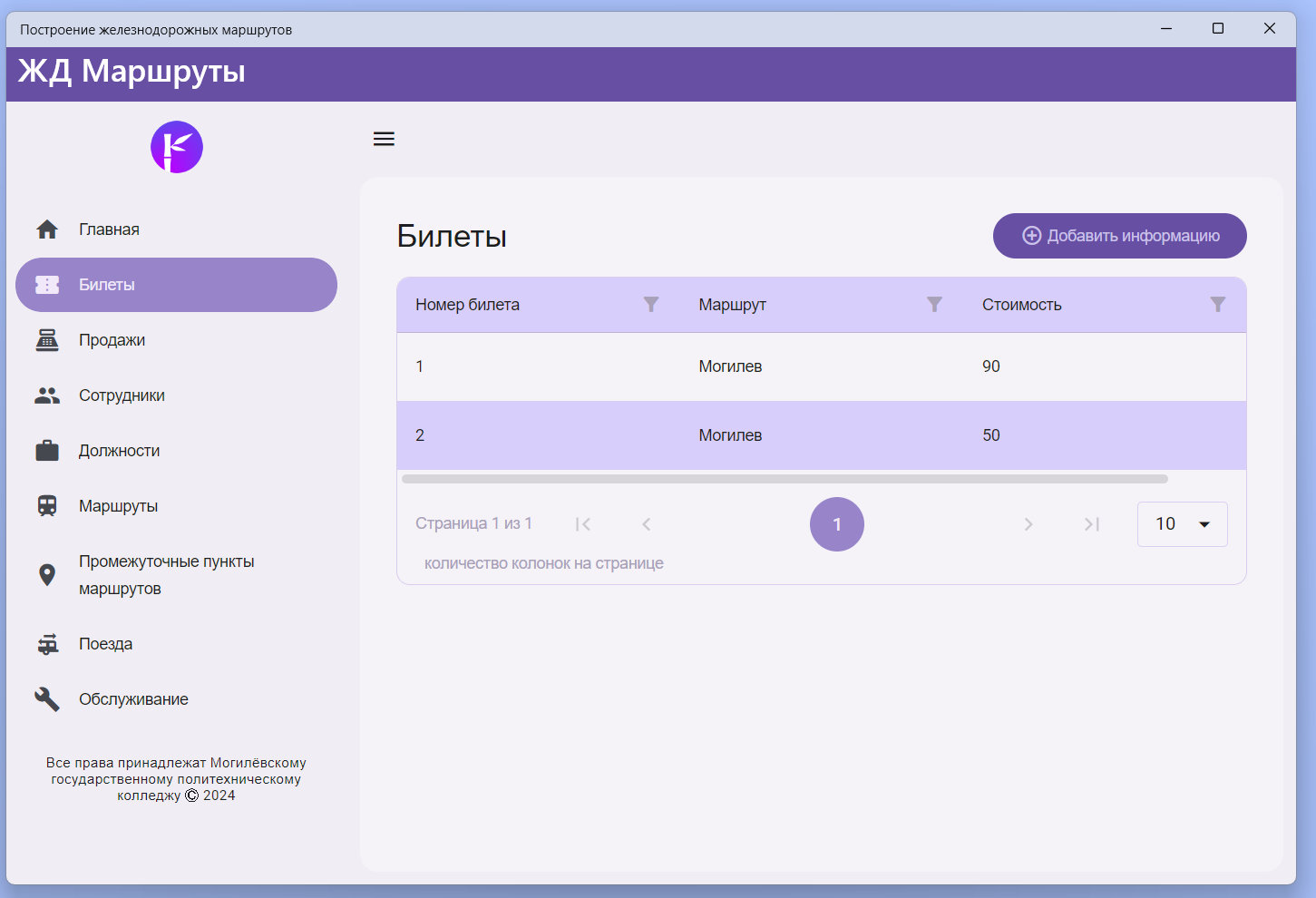


Рисунок В.3 – Форма «Билеты».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок В.4 – Форма «Продажа билетов».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок В.5 – Форма «Информация о сотрудниках».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок В.6 – Форма «Должности».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок В.7 – Форма «Информация о маршрутах».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок В.8 – Форма «Промежуточные пункты маршрутов».

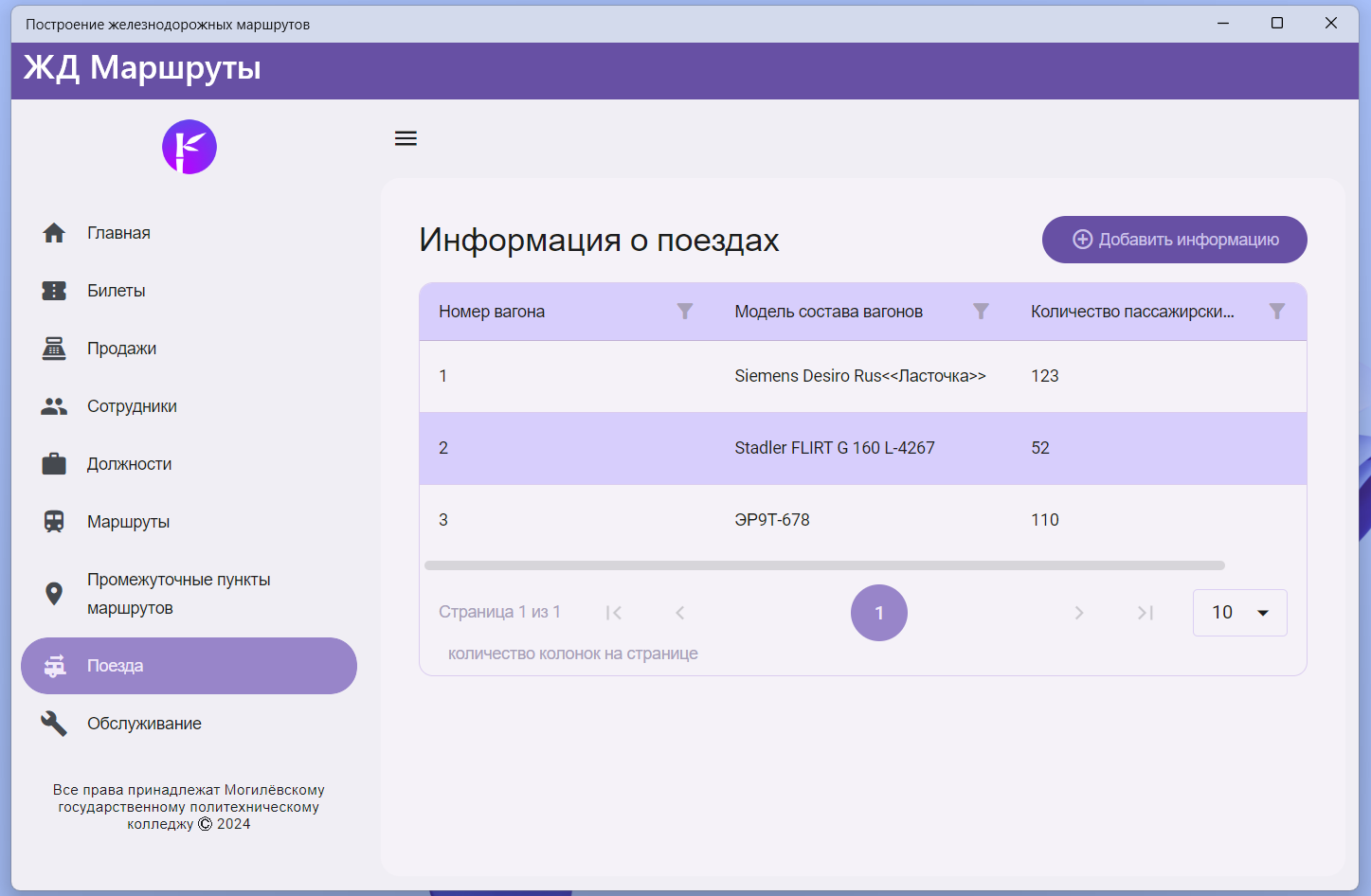


Рисунок В.9 – Форма «Поезда».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок В.10 – Форма «Техническое обслуживание»

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок В.11 – Форма «Информация о сотрудниках» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок В.12 – Форма «Промежуточные пункты маршрутов» на мобильном телефоне.

.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок В.13 – Форма «Техническое обслуживание» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок В.14 – Форма «Билеты» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Автоматически созданное описание

Рисунок В.15 – Форма «Продажа билетов» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок В.16 – Форма «Маршруты» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок В.17 – Форма «Должности» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, веб-страница, Шрифт

Автоматически созданное описание

Рисунок В.18 – Форма «Поезда» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок В.19 – Вид справки «О программе»

Приложение Г

(обязательное)

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

ГЛАВНОЕ УПРАВЛЕНИЕ ПО ОБРАЗОВАНИЮ

МОГИЛЕВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

**ОПИСАНИЕ ПРОГРАММЫ**

2024

**Содержание**

1 Общие сведения 77

2 Функциональное назначение 77

3 Описание логической структуры 77

4 Используемые технические средства 77

5 Вызов и загрузка 78

**1 Общие сведения**

Наименование программы: Автоматизированная Система «Построение Железнодорожных Маршрутов»

Автор программы: Савич Андрей Олегович.

Дата создания: 02.04.2024 г.

Версия 1.0.0.0.

Средой для функционирования программы являются Windows на компьютерах , и Android для телефонов.

Программа написана в среде разработки Jetbrains Rider, на языке программирования C#, с использованием .NET Multi-platform App UI ( .NET MAUI) а также веб платформы Blazor и СУБД Microsoft SQL Server.

**2 Функциональное назначение**

Функциональное назначение автоматизированной системы по теме «Построение Железнодорожных Маршрутов» заключается в предоставлении пользователю информации которая позволяет оперативно учитывать следующие вещи в работе железнодорожного вокзала: продажи билетов, техническое обслуживание поездов, сотрудники депо, расписание движения поездов.. Программа позволяет:

– просмотр отчетов;

– экспорт простых и сложных отчетов;

– просмотр и занесение информации о продажах билетов;

– просмотр и занесение информации о техническом обслуживании железнодорожных маршрутов;

– просмотр и занесение информации о сотрудниках железнодорожных вокзалов;

– просмотр и занесение информации о маршрутах;

– программа обладает интуитивно-понятным интерфейсом, проста и удобна в использовании.

**3 Описание логической структуры**

Алгоритм работы программы “Автоматизированная Система «Построение Железнодорожных Маршрутов»” может быть представлен следующим образом:

Инициализация системы:

Запуск программы на устройстве пользователя.

Подключение к базе данных Microsoft SQL Server.

Загрузка начального интерфейса.

Аутентификация пользователя:

Ввод логина и пароля.

Проверка учетных данных.

Предоставление доступа к функционалу системы в соответствии с ролью пользователя.

Работа с отчетами:

Просмотр отчетов: Выбор типа отчета и просмотр данных.

Экспорт отчетов: Выбор формата экспорта (PDF, Excel и т.д.) и сохранение отчета на устройстве.

Управление продажами билетов:

Просмотр информации о продажах: Отображение текущих данных о продажах билетов.

Занесение информации о продажах: Ввод данных о новых продажах в систему.

Управление техническим обслуживанием:

Просмотр информации о техническом обслуживании: Отображение расписания и истории обслуживания.

Занесение информации о техническом обслуживании: Ввод новых данных о проведенном обслуживании.

Управление персоналом:

Просмотр информации о сотрудниках: Отображение списка сотрудников и их должностей.

Занесение информации о сотрудниках: Ввод данных о новых сотрудниках или изменение информации о текущих.

Управление расписанием движения поездов:

Просмотр расписания: Отображение текущего расписания движения поездов.

Занесение информации о маршрутах: Ввод данных о новых маршрутах или изменение существующих.

Интерфейс пользователя:

Обеспечение интуитивно понятного взаимодействия с системой.

Предоставление возможности простого и удобного доступа к различным функциям программы.

Завершение работы:

Выход из учетной записи пользователя.

Закрытие программы.

**4 Используемые технические средства**

Автоматизированная Система «Построение Железнодорожных Маршрутов» может функционировать на большинстве компьютеров.

Минимальные требования для корректной работы системы:

* центральный процессор - 2 ГГц ;
* оперативная память - 2 ГБ;
* свободное место на HDD - не менее 5 ГБ;
* мышь;
* клавиатура;
* монитор 14 и выше дюймов;
* операционная система Windows.
* доступ к интернету

Для корректного завершения работы программы и операционной системы рекомендуется использование источника бесперебойного питания.

**5 Вызов и загрузка**

Для запуска и корректной работы программы необходимо иметь установочный пакет для развертывания программы и подключение к интернету.

Приложение Д

(обязательное)

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

ГЛАВНОЕ УПРАВЛЕНИЕ ПО ОБРАЗОВАНИЮ

МОГИЛЕВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

**ДОКУМЕНТАЦИЯ ПОЛЬЗОВАТЕЛЯ**

2024

**Содержание**

1 Назначение программного модуля 81

2 Обозначение пакета 81

3 Компоненты пакета. Поставляемые элементы 81

4 Функциональное описание программного средства 81

4.1 Назначение и область применения 81

4.2 Условия эксплуатации 82

4.3 Характеристики программы 82

4.4 Защита информации 82

5 Ввод в действие программного средства 82

6 Техническая информация о программном средстве 82

6.1 Язык 82

6.2 Структура программного средства 83

7 Руководство пользователя 83

8 Тестирование 87

**1 Назначение программного модуля**

Автоматизированная Система «Построение Железнодорожных Маршрутов» предназначена для эффективного и удобного планирования железнодорожного транспорта. Основные цели включают:

Оптимизация маршрутов: Предоставление возможности просмотра информации о маршрутах с учетом времени в пути, стоимости и пересадок.

Улучшение сервиса: Повышение качества обслуживания и комфорта пассажиров.

Функциональное назначение: Система предоставляет информацию для оперативного учета важных аспектов работы железнодорожного вокзала, включая:

Просмотр и экспорт отчетов.

Управление информацией о продажах билетов.

Управление техническим обслуживанием маршрутов.

Управление данными о сотрудниках вокзалов.

Управление информацией о маршрутах.

Программа обладает интуитивно-понятным интерфейсом, проста в использовании.

Эта система является мощным инструментом для планирования и управления железнодорожными маршрутами, обеспечивая удобство и безопасность пассажирских перевозок.

**2 Обозначение пакета**

Наименование программы: Автоматизированная Система «Построение Железнодорожных Маршрутов»

Автор программы: Савич Андрей Олегович.

Дата создания: 02.04.2024 г.

Версия 1.0.0.0.

Средой для функционирования программы являются Windows на компьютерах , и Android для телефонов.

Программа написана в среде разработки Jetbrains Rider, на языке программирования C#, с использованием .NET Multi-platform App UI ( .NET MAUI) а также веб платформы Blazor и СУБД Microsoft SQL Server.

**3 Компоненты пакета. Поставляемые элементы**

В состав программного пакета входит диск с исходным кодом программы и установочным пакетом в формате .msixbundle .

**4 Функциональное описание программного средства**

**4.1 Назначение и область применения**

Алгоритм работы системы:

Инициализация системы: Запуск программы и подключение к базе данных.

Аутентификация пользователя: Вход в систему с проверкой учетных данных.

Работа с отчетами: Просмотр и экспорт отчетов в различных форматах.

Управление продажами билетов: Отображение и ввод данных о продажах.

Техническое обслуживание: Управление информацией о техобслуживании поездов.

Управление персоналом: Просмотр и редактирование данных о сотрудниках.

Расписание движения поездов: Управление информацией о маршрутах и расписании.

Интерфейс пользователя: Интуитивно понятное взаимодействие с программой.

Завершение работы: Выход из системы и закрытие программы.

**4.2 Условия эксплуатации**

Автоматизированная Система «Построение Железнодорожных Маршрутов» может функционировать на большинстве компьютеров.

Минимальные требования для корректной работы системы:

* центральный процессор - 2 ГГц ;
* оперативная память - 2 ГБ;
* свободное место на HDD - не менее 5 ГБ;
* мышь;
* клавиатура;
* монитор 14 и выше дюймов;
* операционная система Windows.
* доступ к интернету

Для корректного завершения работы программы и операционной системы рекомендуется использование источника бесперебойного питания.

**4.3 Характеристики программы**

Размер программы составляет около 80 МБайт.

Данная программа может использоваться без ограничений.

Время выполнения одной операции на разработанном программном модуле составляет около секунды.

**4.4 Защита информации**

Защита информации реализованна в виде мониторига и регистрации событий

**5 Ввод в действие программного средства**

Для запуска и корректной работы программы необходимо иметь компьютер с операционной системой Windows 10 или Windows 11 и доступ в интернет или мобильный телефон с операционной системой Android и доступом в интернет

**6 Техническая информация о программном средстве**

**6.1 Язык**

Программа написана в среде разработки Jetbrains Rider, на языке программирования C#, с использованием .NET Multi-platform App UI ( .NET MAUI) а также веб платформы Blazor и СУБД Microsoft SQL Server.

Интерфейс сайта и оформляемая документация реализованы на русском языке.

**6.2 Структура программного средства**

Работа с программой осуществляется путем перехода между необходимыми страницами, просмотра и редактирования информации на них.

**7 Руководство пользователя**

С первых минут запуска программы пользователь сталкивается со страницей «Главная», которая предостовляет возможность навигации на другие страницы.

Страница «Главная» представлена на рисунке Д.1, приложения Д.

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание**

Рисунок Д.1 – Страница «Главная»

После успешного открытия стрыницы «Главная», пользователь может перейти на любые другие страницы, представленные в боковом меню программы.

Пример страниц представлен на рисунке Д.2, приложения Д.

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

Рисунок Д.2 – Ссылки

Форма «Билеты»

Назначение: Форма «Билеты» используется для просмотра и управления данными о билетах. Внешний вид формы приведен на рисунке Д.3, приложения Д.

Компоненты:

Label: Отображает названия полей базы данных.

Textbox: Позволяет вводить и редактировать данные билетов.

Button: Используется для отправки данных в базу или их обновления.

DataGrid: Показывает текущую информацию о билетах из базы данных.

DropDown: Предоставляет выбор данных из связанных таблиц.

Использование:

Для доступа к форме нажмите кнопку «Билеты» в боковом меню.

Для редактирования билета дважды кликните по нему и введите новые данные в появившемся диалоге.

Нажмите «Сохранить» для обновления информации.

Для удаления билета используйте кнопку с изображением корзины.

Для добавления билета нажмите «Добавить информацию».

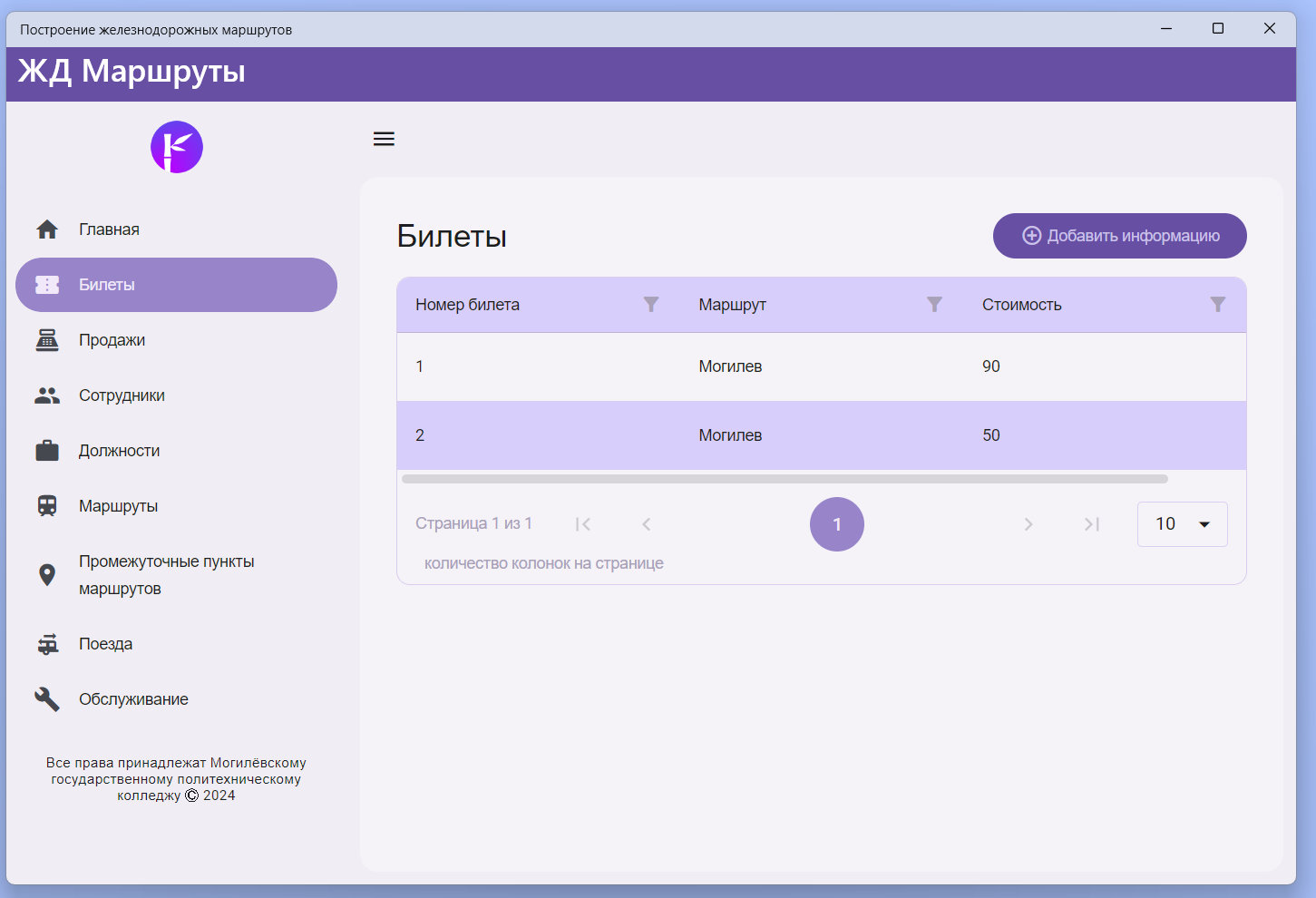


Рисунок Д.3 – Форма «Билеты».

Форма «Продажа билетов»

Назначение: Форма «Продажа билетов» предназначена для отслеживания проданных билетов и ввода информации о новых продажах. Внешний вид формы приведен на рисунке Д.4, приложения Д.

Компоненты и использование: Компоненты и действия аналогичны форме «Билеты», но применяются для управления данными о продажах билетов.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок Д.4 – Форма «Продажа билетов».

Форма «Информация о сотрудниках»

Назначение: Форма предназначена для просмотра и управления данными о сотрудниках. Внешний вид формы приведен на рисунке Д.5, приложения Д.

Компоненты и использование: Компоненты аналогичны предыдущим формам, но используются для работы с информацией о сотрудниках.

Для удаления данных в компоненте DataGrid имеется кнопка с изображением корзины, при нажатии на которую удаляется неактуальная информация.

Для добавления новой информации в базу данных имеется кнопка «Добавить информацию».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок Д.5 – Форма «Информация о сотрудниках».

Форма «Должности»

**Назначение**: Просмотр и управление информацией о должностях, включая названия и требования к стажировке. Внешний вид формы приведен на рисунке Д.6, приложения Д.

**Компоненты формы**:

**Label**: Отображение названий полей базы данных.

**Textbox**: Ввод и обновление данных в базе данных.

**Button**: Обновление данных в DataGrid, отправка и обновление данных в базе данных.

**DataGrid**: Отображение текущей информации из базы данных.

**DropDown**: Выбор данных из связанных таблиц базы данных.

**Использование**:

В главном окне нажмите кнопку «Должности» в боковом меню.

На экране отобразится информация о должностях из базы данных.

**Редактирование информации**:

Дважды кликните левой кнопкой мыши по выбранному пункту в DataGrid.

Введите необходимые изменения в появившемся диалоговом окне.

Нажмите «Сохранить» для обновления информации.

**Удаление информации**:

Нажмите на кнопку с изображением корзины в DataGrid для удаления выбранной записи.

**Добавление новой информации**:

Нажмите кнопку «Добавить информацию» для ввода новых данных.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок Д.6 – Форма «Должности».

Форма «Информация о маршрутах»

**Назначение**: Просмотр и управление информацией о маршрутах. Внешний вид формы приведен на рисунке Д.7, приложения Д.

**Компоненты форм**:

Аналогичны компонентам предыдущих форм

**Использование**:

В главном окне нажмите кнопку «Маршруты» в боковом меню.

На экране отобразится информация о маршрутах из базы данных.

**Редактирование информации**:

Дважды кликните левой кнопкой мыши по выбранному пункту в DataGrid.

Введите необходимые изменения в появившемся диалоговом окне.

Нажмите «Сохранить» для обновления информации.

**Удаление информации**:

Нажмите на кнопку с изображением корзины в DataGrid для удаления выбранной записи.

**Добавление новой информации**:

Нажмите кнопку «Добавить информацию» для ввода новых данных.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок Д.7 – Форма «Информация о маршрутах».

Форма «Промежуточные пункты маршрутов»

**Назначение**: Просмотр и управление информацией о промежуточных пунктах маршрутов. Внешний вид формы приведен на рисунке Д.8, приложения Д.

**Компоненты форм**:

Аналогичны компонентам предыдущих форм

**Использование**:

В главном окне нажмите кнопку «Промежуточные пункты маршрутов» в боковом меню.

На экране отобразится информация о промежуточных пунктах маршрутов из базы данных.

**Редактирование информации**:

Дважды кликните левой кнопкой мыши по выбранному пункту в DataGrid.

Введите необходимые изменения в появившемся диалоговом окне.

Нажмите «Сохранить» для обновления информации.

**Удаление информации**:

Нажмите на кнопку с изображением корзины в DataGrid для удаления выбранной записи.

**Добавление новой информации**:

Нажмите кнопку «Добавить информацию» для ввода новых данных.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок Д.8 – Форма «Промежуточные пункты маршрутов».

Форма «Поезда»

**Назначение**: Просмотр и управление данными о вагонах и поездах, используемых на маршрутах. Внешний вид формы приведен на рисунке Д.9, приложения Д.

**Компоненты форм**:

Аналогичны компонентам предыдущих форм

**Использование**:

В главном окне нажмите кнопку «Поезда» в боковом меню.

На экране отобразится информация о поездах из базы данных.

**Редактирование информации**:

Дважды кликните левой кнопкой мыши по выбранному пункту в DataGrid.

Введите необходимые изменения в появившемся диалоговом окне.

Нажмите «Сохранить» для обновления информации.

**Удаление информации**:

Нажмите на кнопку с изображением корзины в DataGrid для удаления выбранной записи.

**Добавление новой информации**:

Нажмите кнопку «Добавить информацию» для ввода новых данных.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок Д.9 – Форма «Поезда».

Форма «Техническое обслуживание»

**Назначение**: Просмотр и управление данными о техническом обслуживании вагонов. Внешний вид формы приведен на рисунке Д.10, приложения Д.

**Компоненты форм**:

Аналогичны компонентам предыдущих форм

**Использование**:

В главном окне нажмите кнопку «Обслуживание» в боковом меню.

На экране отобразится информация об обслуживании и техническом состоянии поездов из базы данных.

**Редактирование информации**:

Дважды кликните левой кнопкой мыши по выбранному пункту в DataGrid.

Введите необходимые изменения в появившемся диалоговом окне.

Нажмите «Сохранить» для обновления информации.

**Удаление информации**:

Нажмите на кнопку с изображением корзины в DataGrid для удаления выбранной записи.

**Добавление новой информации**:

Нажмите кнопку «Добавить информацию» для ввода новых данных.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок Д.10 – Форма «Техническое обслуживание»

Приложение также работает на мобильных телефонах и имеет вид, приведенный на рисунках Д.11-Д.18, приложения А.

На телефонах все управление и меню приложения находится внизу экрана и для открытия одной из форм внизу экрана необходимо нажать на соответствующий пункт меню после чего действия аналогичны работе на ПЭВМ и различие только в том что все элементы управления адаптированны по размеру для работе на экране телефона

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок Д.11 – Форма «Информация о сотрудниках» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок Д.12 – Форма «Промежуточные пункты маршрутов» на мобильном телефоне.

.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок Д.13 – Форма «Техническое обслуживание» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок Д.14 – Форма «Билеты» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Автоматически созданное описание

Рисунок Д.15 – Форма «Продажа билетов» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок Д.16 – Форма «Маршруты» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок Д.17 – Форма «Должности» на мобильном телефоне.

Изображение выглядит как текст, снимок экрана, веб-страница, Шрифт

Автоматически созданное описание

Рисунок Д.18 – Форма «Поезда» на мобильном телефоне.

**8 Тестирование**

Процесс **тестирования программных модулей** состоит в проверке корректности обработки модулями поступающей информации и получающихся на выходе данных в соответствии с функциями, представленными в спецификациях требований. Должна быть проверена корректность структуры модулей и примененных конструктивных элементов.

Тестирование программы осуществлялось на персональном компьютере со следующими техническими характеристиками:

* процессор – AMD Ryzen 5 7530U with Radeon Graphics 2.00 GHz
* видеокарта – AMD Radeon Vega 7 Graphics
* оперативная память - 16,00 ГБ LPDDR4
* операционная система – Windows 11

Результаты тестирования представлены в приложении В.

**Приложение Е**

(обязательное)

**Расчет затрат на оплату труда и отчислений на социальные нужды**

Выполнил учащийся Савич А.О

Учебная группа ПО-455

Затраты на оплату труда программиста зависят от времени разработки программного продукта, т.е. ожидаемой трудоемкости и квалификации специалиста. Затраты на оплату труда программиста ФОТ, руб., состоят из двух частей – основная и дополнительная заработная плата и рассчитываются по формуле

ФОТ = ЗПо + ЗПд , (Е.1)

где ЗПо – основная заработная плата производственных рабочих, руб.;

ЗПо – основная заработная плата программиста, руб.;

ЗПд – дополнительная заработная плата программиста, руб.

Статья «Основная заработная плата» включает заработную плату программиста, рассчитанную с учетом трудоемкости работ

Основная заработная плата определяется по формуле

ЗПо = Тст1  Те  Кпр , (Е.2)

где Тст1 – часовая тарифная ставка по разряду выполняемых работ с учетом повышающего коэффициента, руб.;

Те – трудоемкость разработки программного продукта, ч.;

Кпр – коэффициент премирования;

Кпр = 1,3.

Тст.6 = 2,5 руб.

Те = 86,62 ч.

Основная заработная плата составит

Статья «Дополнительная заработная плата» отражает выплаты, предусмотренные законодательством за непроработанное в производстве время (оплата отпускных, компенсаций, выполнение гособязанностей, оплата льготных часов подросткам, кормящим матерям). Размер выплат предусмотрен обычно в пределах до 15% от основной зарплаты.

Дополнительная заработная плата определяется по формуле

ЗПд = ЗПо  Кд. , (Е.3)

где Кд. – коэффициент дополнительной зарплаты.

Кд. = 0,15.

Дополнительная заработная плата составит

Затраты на оплату труда программиста составят

ФОТ = 323,74 руб.

Отчисления на социальные нужды, Ос.н., руб., рассчитываются по формуле

Ос.н. = hс.н.  ФОТ / 100 , (Е.4**)**

где hс.н. – норматив отчислений на социальные нужды;

hс.н. – 34 %.