МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ И МОЛОДЕЖИ РЕСПУБЛИКИ КРЫМ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РЕСПУБЛИКИ КРЫМ «ФЕОДОСИЙСКИЙ ПОЛИТЕХНИЧЕСКИЙ ТЕХНИКУМ»

КУРСОВОЙ ПРОЕКТ

по профессиональному модулю ПМ 01 «Разработка программных модулей программного обеспечения для компьютерных систем»

по междисциплинарному курсу МДК 01.01 «Разработка программных модулей»

На тему: Разработка программного модуля «Пассажир поезда дальнего следования»

Студента 3 курса группы ИСПП 22 1/9

Сейдалиев Эрвин Серверович

Руководитель: Ульяницкая Н.Н.\_\_\_\_\_\_\_

Оценка: \_\_\_\_\_\_\_

Феодосия 2024

РЕФЕРАТ

Сейдалиев Эрвин Серверович: Разработка программного модуля «Пассажир поезда дальнего следования”.

Курсовой проект по МДК 01.01 Разработка программных модулей. ГБПОУ РК «ФПТ». Группа ИСПП 22 1/9. 2024 г.

Пояснительная записка 28 листов.

Таблицы 2 таблиц.

Иллюстрации 15 иллюстраций.

Ссылки 0 ссылок.

В курсовом проекте рассмотрены концепции создания просмотра и покупки железо дорожные билетов на ПК, чтобы сделать более простой и автоматизировать работу предприятия

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc188817714)

[1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6](#_Toc188817715)

[1.1 Описание предметной области 6](#_Toc188817716)

[1.2 Обзор аналогичных разработок 6](#_Toc188817717)

[1.3 Обоснование необходимости разработки 6](#_Toc188817718)

[1.4 Постановка задачи 7](#_Toc188817719)

[1.5 Минимальные требования к оборудованию 8](#_Toc188817720)

[1.6 Используемое программное обеспечение 8](#_Toc188817721)

[2 РАЗРАБОТКА ПРИЛОЖЕНИЯ «ПОЕЗД ДАЛЬНЕГО СЛЕДОВАНИЯ» 10](#_Toc188817722)

[2.1 Обоснование выбора среды разработки 10](#_Toc188817723)

[2.2 Обоснование выбора СУБД 11](#_Toc188817724)

[2.3Требования к программе 12](#_Toc188817725)

[2.4 Проектирование программного продукта 13](#_Toc188817726)

[2.5 Описание программного интерфейса 16](#_Toc188817727)

[2.6 Тестирование и отладка 18](#_Toc188817728)

[ЗАКЛЮЧЕНИЕ 26](#_Toc188817729)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 27](#_Toc188817730)

[ПРИЛОЖЕНИЕ А ЛИСТНИГ КОДА 29](#_Toc188817731)

[ПРИЛОЖЕНИЕ В СЛАЙДЫ ПРЕЗИНТАЦИИ 41](#_Toc188817732)

# ВВЕДЕНИЕ

Поезд Дальнего Следования выполняет функции в поисках расписания поездов, чтобы понимать во сколько поезд приезжает и уезжает. Данная программа ищет маршруты и расписания по определенным запросам

Актуальность темы состоит в не сильном спросе Поезд Дальнего Следования так такого из-за существования сайтов для просмотра расписания. Автоматизации данной сферы услуг поможет намного проще и быстрее подбирать желаемые маршруты и не только. И поможет найти более выгодные и быстрые маршруты.

Предметом курсовой работы является разработка программного продукта «Поезд дальнего следования».

В данной курсовой работе будет разработано программное обеспечения для автоматизации процесса нахождения маршрутов и просмотра расписаний.

Для выполнения поставленной цели и задач в работе будут использованы такие программные средства как: MS SQL Server 2022, Visual Studio 2022, MS Word 2016.

Курсовая работа состоит из трех глав, в первой главе описана предметная область, обзор существующих решений задач и выбор языка программирования, вторая глава описывает разработанное программное обеспечение и производится тестирование программного обеспечение.

# 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Описание предметной области

Поезд дальнего следования — организация помощи и удобства в планировании путешествия, нахождения наиболее оптимальных маршрутов и просмотра расписания. Наш же Поезд Дальнего Следования берет более простые запросы на поиск билетов.

Предметом курсовой работы является разработка программного продукта «Поезд Дальнего Следования». В самом продукте указано возможность заносить информацию о маршрутах и делать фильтрацию маршрутов для поиска нужного.

Каждая таблица имеет четкое количество столбцов и строк для отображения нужной информации, которая вносится пользователем для того, чтобы админ мог проводить фильтрацию по БД.

## 1.2 Обзор аналогичных разработок

Аналогами Нашего Поезд Дальнего Следования могут выходит все различные платформы или места в которым можно с кем-то познакомится

1. Tutu.ru — российский онлайн-сервис для организации путешествий. Позволяет покупать билеты на поезд, самолёт и автобус, бронировать отели, туры и экскурсии.
2. Rzd.ru — это официальный сайт ОАО «РЖД», где можно, например, посмотреть расписание рейсов и цены на билеты, оформить покупку билетов онлайн.

## 1.3 Обоснование необходимости разработки

Удобство для пользователей: Программа позволяет пользователям легко и быстро приобретать билеты на поезда дальнего следования, не выходя из дома. Это особенно важно для людей, живущих в отдаленных районах или имеющих ограниченные возможности передвижения. Возможность просмотра маршрутов и расписаний в режиме реального времени помогает пользователям планировать свои поездки более эффективно, избегая очередей и сэкономив время.

Автоматизация и оптимизация процессов: Программа автоматизирует процесс бронирования и покупки билетов, что снижает нагрузку на кассы и операторов. Это позволяет железнодорожным компаниям более эффективно управлять своими ресурсами, минимизировать ошибки и ускорить обслуживание клиентов. Кроме того, автоматизация позволяет интегрировать систему с другими сервисами, такими как оповещения о задержках и изменениях в расписании, что повышает общую удовлетворенность клиентов.

Аналитика и улучшение сервиса: Программа собирает данные о предпочтениях и поведении пользователей, что позволяет железнодорожным компаниям анализировать спрос и оптимизировать свои услуги. Например, можно выявить наиболее популярные маршруты и время поездок, что поможет в планировании дополнительных рейсов или улучшении существующих. Аналитика также позволяет предлагать персонализированные предложения и скидки, что способствует увеличению лояльности клиентов и привлечению новых пассажиров.

## 1.4 Постановка задачи

Целью разработки данного приложения является создание компактной и удобной системы управления проектами сотрудников. Для достижения этой цели были поставлены следующие задачи:

1. Разработка интерфейса авторизации:

* Создание окон регистрации и авторизации с возможностью входа и создания аккаунта.
* Обеспечение безопасности с использованием логинов и паролей.

2. Реализация функционала:

* Разработка окна ввода логина и пароля.

3. Создание основного окна приложения:

* Разработка интерфейса для отображения существующих проектов.

4. Добавление проектов в базу данных:

* Реализация функции добавления новых проектов через соответствующий интерфейс

## 1.5 Минимальные требования к оборудованию

Для корректной работы программы необходимо, чтобы компьютер, на котором запускается приложение, соответствовал следующим минимальным требованиям:

1. Операционная система: Windows 10.

2. Процессор: не ниже Intel Core I3

3. Оперативная память: 512 МБ и выше.

4. Свободное место на диске: 50 МБ.

5. Экран: Разрешение 1024x768 и выше.

Указанные требования позволят запускать приложение и осуществлять базовые операции в удобном режиме. Для обеспечения стабильной работы рекомендуется использовать обновленные версии операционных систем.

## 1.6 Используемое программное обеспечение

В процессе разработки приложения для записи проектов и просмотра были использованы следующие технологии и инструменты:

1. C#: Язык программирования, на котором написано приложение. C# был выбран за свою простоту, универсальность и богатые возможности для работы с базами данных

2. SQLServer: Система управления базами данных, используемая для хранения информации о поездах. SQLServer предоставляет легкую и компактную реляционную базу данных, что идеально подходит для небольших приложений.

3. Visual Studio — это интегрированная среда разработки (IDE), разработанная компанией Microsoft, которая предоставляет все необходимые инструменты для создания графического интерфейса приложения. В Visual Studio есть множество функций и инструментов, которые делают процесс разработки графического интерфейса быстрыми и удобными.

Эти технологии были выбраны с учетом простоты разработки, эффективности использования и соответствия требованиям проекта.

# 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ «ПОЕЗД ДАЛЬНЕГО СЛЕДОВАНИЯ»

## 2.1 Обоснование выбора среды разработки

Для создания программного обеспечения для создания тестирования использовалась среда разработки Microsoft Visual Studio 2022 и язык программирования С#.

Microsoft Visual Studio — продукт компании [Майкрософт](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B9%D0%BA%D1%80%D0%BE%D1%81%D0%BE%D1%84%D1%82), который включает [интегрированную среду разработки](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) программного обеспечения и ряд других инструментальных средств.

Преимущества Visual Studio 2022:

* Поколение компиляторов. Компилятор семейства Roslyn. Одно из главных преимуществ этих компиляторов — модульность. Она послужила ключом к организации компилятора в виде компонентной управляемой платформы.
* Отладка. В Visual Studio 2022 расширяются возможности работы с кодом, благодаря функциям IntelliTrace, CodeMap, CodeLens.
* Тестирование. Visual Studio 2022 позволяет использовать все возможности платформы Microsoft, для обеспечения контроля за качеством выпускаемого продукта.
* Нагрузочное тестирование. Для проверки производительности доступны инструменты для записи последовательности действий пользователей в браузере и оформления этих этапов в качестве теста.
* Управление процессами разработки. Благодаря возможностям MSDN подписки становятся доступны инструменты для построения процессов управления проектами.

C# — объектно-ориентированный язык программирования, который относится к семье языков с Cи-подобным синтаксисом. Язык имеет поддерживает перегрузку операторов, полиморфизм, атрибуты, делегаты, обобщённые типы и методы, итераторы, анонимные функции.

Основные достоинства С#:

1. C# создавался параллельно с каркасом Framework .Net и полностью учитывает все его возможности.
2. C# - объектно-ориентированный язык, в котором типы представлены в виде классов.
3. C# является объектно-ориентированным языком, в котором существует возможность универсализации и наследования.
4. Программисты C# получают такие же возможности работы с виртуальной машиной, что и программисты Java, благодаря платформе Framework .Net.
5. Библиотека поддерживает удобное построение разных типов программных продуктов на C#, очень просто сохранять и получать информацию из БД.

Реализация, которая сочетает построение надежного и эффективного кода.

## 2.2 Обоснование выбора СУБД

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название характеристики | СУБД MySQL | СУБД MS SQL Server |
| 1 | Архитектура | Клиент-сервер | Клиент-сервер |
| 2 | Разграничение прав доступа | Присутствует, привилегии, пароли, роли и права доступа для пользователей | Присутствует, можно ограничивать столбцы, роли и права доступа для пользователей |
| 3 | Механизм ролей | Присутствует | Присутствует |
| 4 | Язык программирования | SQL | Transact SQL, XML |
| 5 | Репликация | Присутствует | Присутствует |
| 6 | Целостность данных | Поддерживается с помощью ограничения целостности и механизмов хранения InnoDB | Поддерживается с помощью ограничений стандарта ANSI. |
| 7 | Защита данных | SSL/TLS шифрование при работе сервер-клиент, Шифрование данных на диске | Защита данных осуществляется с помощью шифрования |

Выбор MySQL объясняется следующими причинами:

* возможность развернуть данную СУБД на маломощном сервере.
* полная поддержка большинства дистрибутивов ОС Linux.
* не сложна для использования непрофессиональным пользователем.
* не требует значительных системных ресурсов для развертывания и эксплуатации.

Таким образом, учитывая специфику приложения для приёмной комиссии СПО, выбор в пользу MySQL является обоснованным и целесообразным.

## 2.3Требования к программе

Разработанное программное приложение ориентировано на обеспечение удобного просмотра проектов и добавления новых. Ниже приведены основные требования к функциональности и характеристикам программы:

1. Авторизация:

* Возможность входа в систему с помощью логина и пароля.

2. Основное окно:

* Просмотр существующих проектов.
* Внос своего проекта.

3. Интерфейс пользователя:

* Интуитивно понятный графический интерфейс.
* Использование диалоговых окон для ввода данных и отображения уведомлений.

4. База данных:

* Использование базы данных SQLServer для хранения информации.
* Создание таблиц.

5. Обработка ошибок:

Вывод информативных сообщений об ошибках при неправильном вводе данных.

6. Стиль кода:

* Применение объектно-ориентированного программирования для структурирования кода

Эти требования обеспечивают функциональность приложения, которая соответствует ожиданиям пользователей и обеспечивает удобство в использовании.

## 2.4 Проектирование программного продукта

Проектирование программы представляет собой важный этап, нацеленный на определение структуры и организации кода для достижения эффективной и устойчивой работы. В разработанном приложении "Поезд дальнего следования" использован подход, основанный на объектно-ориентированном программировании (ООП) с использованием среды Visual Studio.

Основные аспекты проектирования включают:

1. Классы и объекты:

- Применение классов для представления различных окон приложения

(Form1, Form2, Form3, Form4,Form5).

* Создание объектов для взаимодействия с интерфейсом и базой данных.

2. Интерфейс пользователя:

* Организация элементов интерфейса в виде фреймов и виджетов для достижения структурированного и легко воспринимаемого вида.

3. База данных:

Входят 4 таблицы

Первая таблица состоит из ключа и второстепенных элементов (kodv, pynt, vog, mest, vr) что изображено на рисунке 1

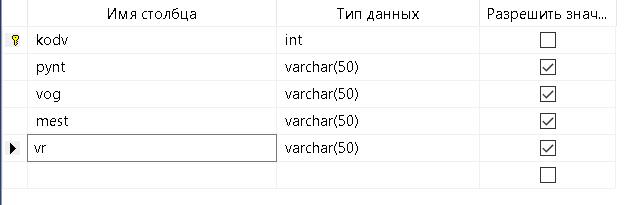


Рисунок 1 – Структура таблицы Obshee

Вторая таблица состоит из ключа и второстепенных элементов (kodt, pynt, vog, mest, price) что изображено на рисунке 2

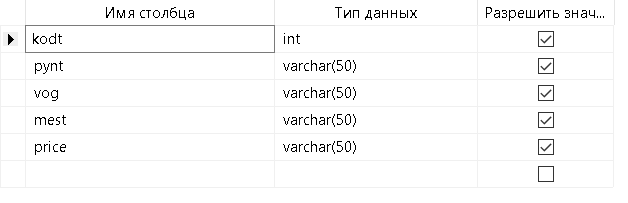


Рисунок 2 – Структура таблицы Homeroki

Третья таблица состоит из ключа и второстепенных элементов (kodr, kodv, price) что изображено на рисунке 3

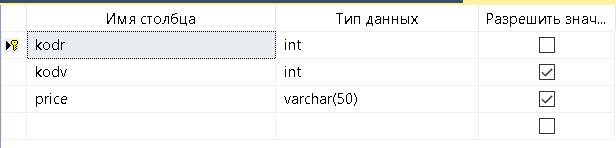


Рисунок 3 – Структура таблицы tic

Четвертая таблица состоит из ключа и второстепенных элементов (kodp, name, pass) что изображено на рисунке 4

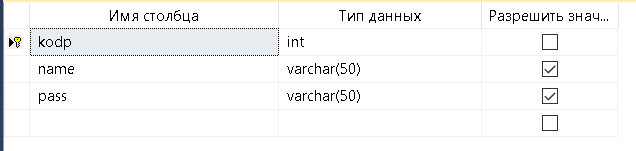


Рисунок 4 – Структура таблицы admin

* Реализация соединения с базой данных SQLServer.
* Создание отдельных таблиц для каждой задачи.

4. Обработка событий:

* Назначение функций-обработчиков для кнопок и событий взаимодействия с интерфейсом.
* Управление переходами между окнами.

5. Обработка ошибок:

* Включение механизмов обработки ошибок при вводе данных пользователем.
* Отображение диалоговых окон с информацией об ошибках.

6. Удобство использования:

* Обеспечение интуитивно понятного пользовательского интерфейса с минимальным количеством шагов для выполнения основных задач.

Проектирование приложения "Поезд дальнего следования" сосредоточено на обеспечении легкости в использовании, четкости интерфейса и эффективности работы с базой данных. Это позволяет пользователям сотрудникам уверенно и эффективно взаимодействовать с программой при добавлении и поиске проектов.

## 2.5 Описание программного интерфейса

Программный интерфейс разработанного приложения "Поезд дальнего следования" представляет собой интуитивно понятный и легко взаимодействующий пользовательский интерфейс. Вот основные элементы интерфейса:

1.Окно авторизации (Form1)

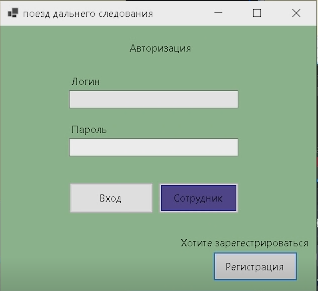


Рисунок 5 – Окно авторизации

* Кнопка для входа, кнопка для перехода на окно регистрации.
* Вход требует ввода логина и пароля.

2. Окно регистрации (Form2)

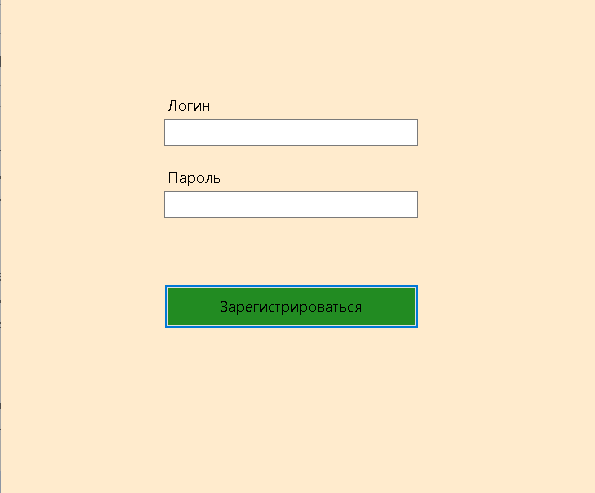


Рисунок 6 – Авторизация сотрудника

Поля для регистрации пользователя путѐм добавления имени, фамилии логина и пароля.

3. Основное окно приложения (Form3)



Рисунок 7 – Главное окно программы

4.Окно администрации

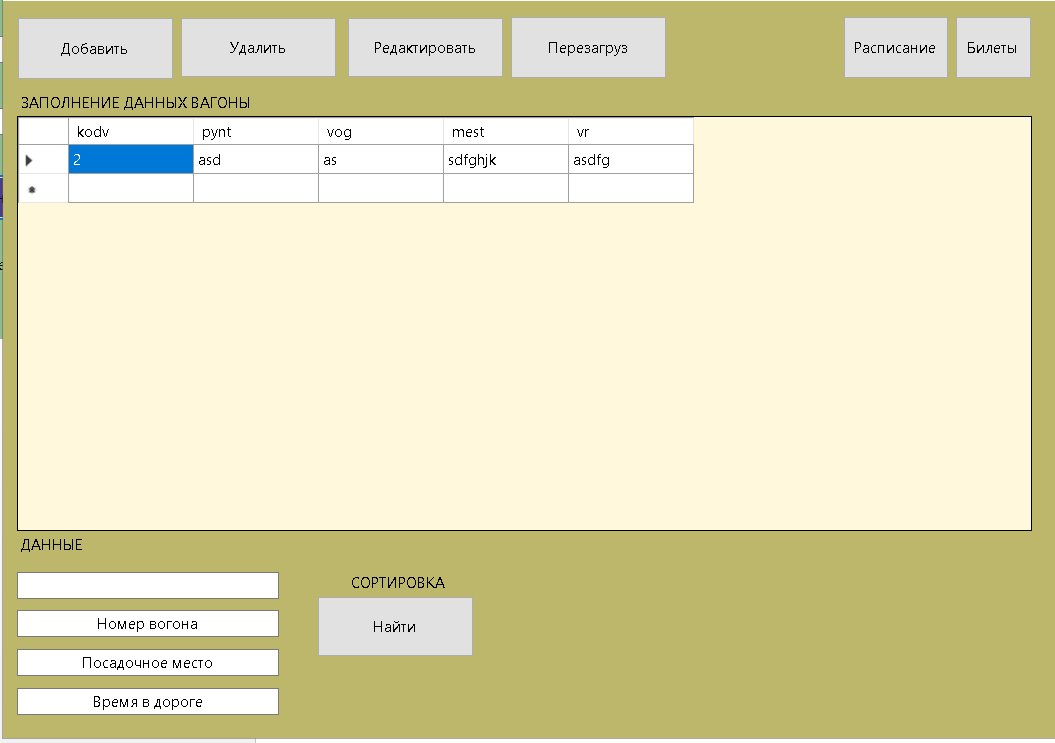


Рисунок 8- Окно администрации

Программный интерфейс спроектирован с учетом удобства использования и предоставляет пользователям возможность легко взаимодействовать с функциональностью приложения. Он сочетает в себе ясность представления данных с простотой ввода информации, что делает приложение доступным для пользователей различного уровня опыта.

## 2.6 Тестирование и отладка

Для обеспечения надежной и стабильной работы приложения "Поезд дальнего следования", были проведены тщательные процессы тестирования и отладки. В данном разделе представлены основные этапы и результаты этих процессов.

1.Модульное тестирование:

Каждый модуль приложения был протестирован отдельно для проверки корректности его функционирования.

Модули, такие как авторизация, регистрация, добавление проекта и просмотр проекта, прошли успешные тесты.

2.Интеграционное тестирование:

Проведено тестирование взаимодействия различных модулей для обнаружения возможных конфликтов и ошибок.

Убеждаемся, что передача данных между модулями осуществляется корректно и без потерь информации.

3.Пользовательское тестирование:

Приложение было предоставлено тестовой группе пользователей для оценки удобства использования и обнаружения потенциальных проблем. Получены обратная связь и рекомендации по улучшению интерфейса и функциональности.

4.Отладка:

Проведена отладка кода для выявления и устранения возможных ошибок и исключений.

Решены выявленные проблемы с вводом данных, обработкой запросов к базе данных и отображением результатов.

5.Тестирование безопасности:

Приняты меры для обеспечения безопасности хранения и обработки данных пользователя.

Тестовый случай — это алгоритмы проверки функциональности программы. Каждый тестовый случай должен обладать следующими свойствами: четкой целью проверки, известными начальными условиями тестирования, строго определенной средой тестирования, тестовыми данными и ожидаемым результатом тестирования.

Результаты тестирования модулей представлены в таблице 3.1.

Таблица 3.1 – Результаты тестирования модулей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название  модуля/экрана | Описание тестового случая | Ожидаемые результаты | Тестовый случай пройден? | Комментарии |
| Кнопка «Вход» | Одинарный клик левой кнопкой мыши | Соединение с базой данных и авторизация пользователя | Да | Работает корректно |
| Кнопка «Забронировать » | Одинарный клик левой кнопкой мыши | Появление сообщения о успешной брони | Да | Работает корректно |
| Пункт меню «Написать отзыв» | Одинарный клик левой кнопкой мыши | Окно выбора тура о котором хотите написать отзыв | Да | Работает корректно |
| Кнопка “Оставить отзыв” | Одинарный клик левой кнопкой мыши | Появление сообщения о том что отзыв оставлен | Да | Работает корректно |
| Пункт меню “Забронированные” | Одинарный клик левой кнопкой мыши | Показывает список забронированных туров | Да | Работает корректно |
| Кнопка «Экспорт в MS Excel» | Одинарный клик левой кнопкой мыши | Экспорт данных в MS Excel | Да | Работает корректно |
| Пункт меню «Зарегистрироваться» | Одинарный клик левой кнопкой мыши | Открытие окна «Регистрация» | Да | Работает корректно |
| Кнопка «Войти» | Одинарный клик левой кнопкой мыши | Открытие окна «Программа(Form3)» | Да | Работает корректно |
| Кнопка «Выход» | Одинарный клик левой кнопкой мыши | Приложение закрыто | Да | Работает корректно |
| Ввод информации в поля | Одинарный клик левой кнопкой мыши | Добавление информации в БД | Да | Работает корректно |

Результаты тестирования и отладки свидетельствуют о высоком уровне стабильности и функциональности приложения. Обнаруженные и устраненные проблемы позволяют утверждать, что разработанное приложение готово к использованию в реальных условиях и обеспечивает надежное взаимодействие с пользователем и базой данных.

UML-ДИАГРАММЫ

Построение модели базы данных базировалось на использовании унифицированного языка моделирования UML (Unified Modeling Language).

Чтобы не вдаваться в подробности разработки базы данных на языке UML в данной статье опишем основные UML-диаграммы.

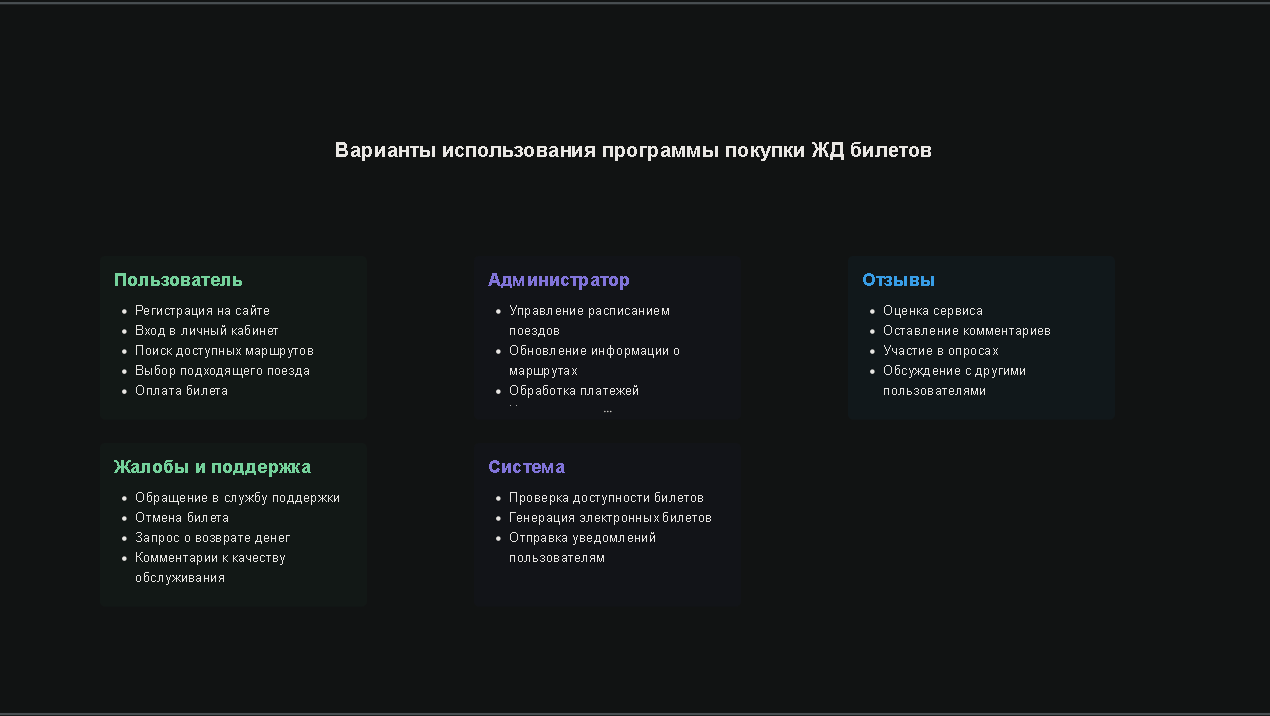


Рисунок 9 – Диаграмма вариантов использования

Диаграмма использования на языке UML (рис. 9) дает ответ о назначении системы во внешнем мире

UML-диаграмма деятельности

В общем случае, на диаграмме деятельности показано разложение некоторой деятельности на ее составные части. Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

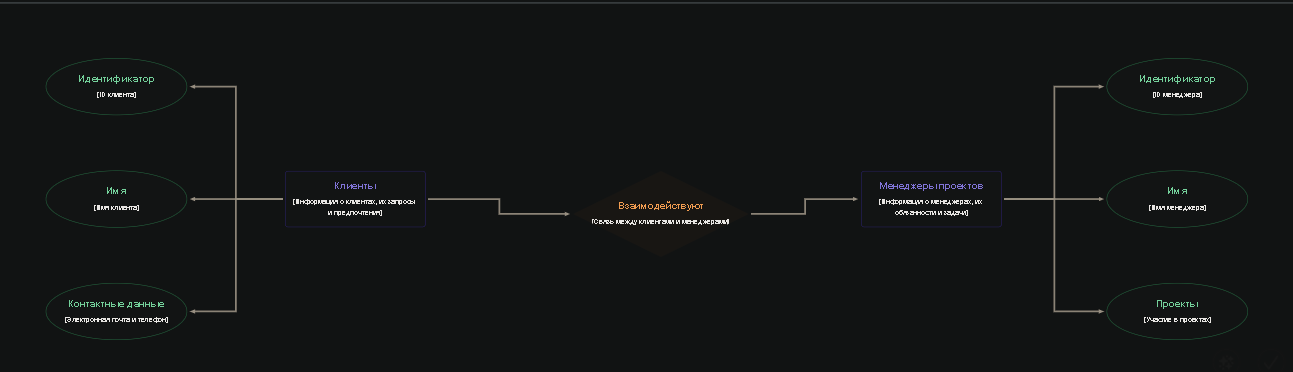


Рисунок 10 – UML-диаграмма пользователей разного уровня доступа»

UML-диаграмма взаимодействия

На диаграмме взаимодействия отображаются только те объекты, которые непосредственно принимают участие во взаимодействии.

Ключевым моментом для диаграмм взаимодействия (или последовательности) есть динамика взаимодействия объектов во времени



Рисунок 11 – UML-диаграмма взаимодействия

UML-диаграмма классов

При построении UML модели различают понятие класса. Классом называется описание множеств объектов с общими атрибутами. Графически класс изображается в виде прямоугольника.

В разработанной программной системе, с точки зрения модели базы данных, используется шесть классов, каждый из которых есть объектом, который отвечает той или другой таблице базы данных.

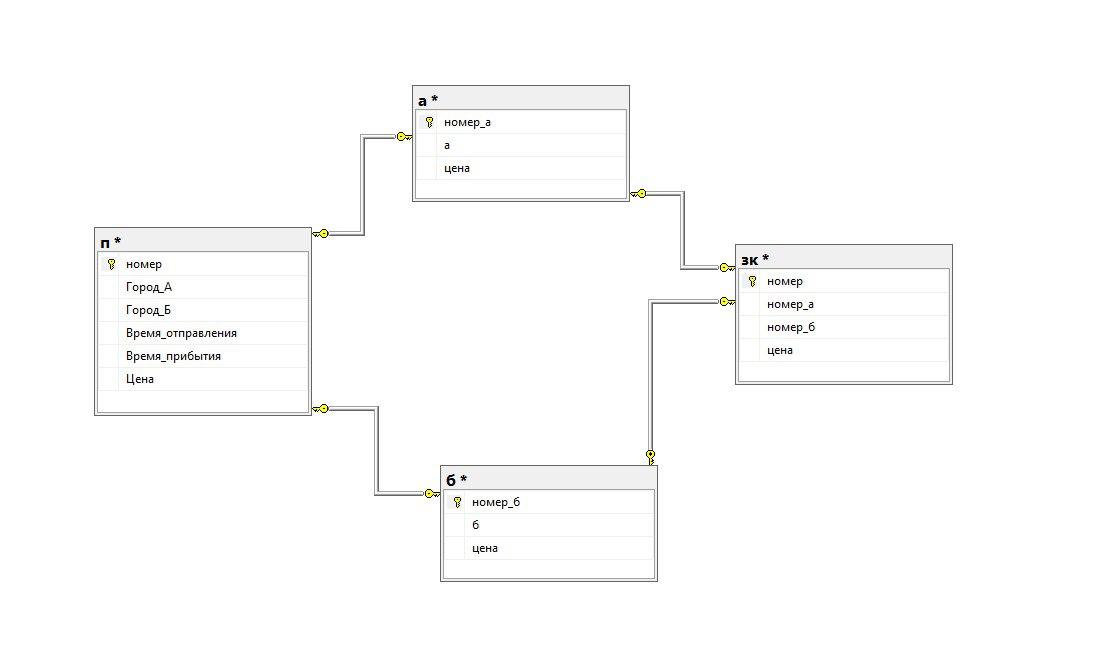


Рисунок 12 – Отображение зависимостей между классами программы

# ЗАКЛЮЧЕНИЕ

В ходе курсовой работы было разработано приложение "Поезд дальнего следования", цель которого – улучшение пользовательского опыта пассажиров на дальних маршрутах. В процессе работы были решены следующие задачи:

1. Анализ потребностей пользователей и определение ключевых функций приложения.
2. Разработка интерфейса, ориентированного на удобство и простоту использования.
3. Реализация основных функций, таких как покупка билетов, отслеживание маршрута, информация о станциях и услугах на борту.
4. Проведение тестирования и устранение выявленных недостатков.

В результате проделанной работы было создано приложение, которое не только упрощает процесс планирования и покупки билетов для пользователей, но и предоставляет актуальную информацию о путешествии в режиме реального времени. Это способствует повышению комфорта и удовлетворенности пассажиров.

Дальнейшее развитие приложения может включать внедрение дополнительных сервисов, таких как виртуальные экскурсии по маршруту, рекомендации по интересным местам в городах назначения и интеграция с другими транспортными средствами. Таким образом, приложение "Поезд дальнего следования" имеет большой потенциал для дальнейшего совершенствования и масштабирования.

Полученные результаты подтверждают готовность приложения к использованию конечными пользователями, обеспечивая им удовлетворение от использования функционала и способствуя развитию культурной сферы в сфере исследований.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Стивен Райт. "Программирование для Microsoft SQL Server 2019". — М.: ДМК Пресс, 2020. — 576 с.
2. Джозеф Альбахари, Бен Альбахари. "C# 7.0 в действии". — М.: Питер, 2018. — 896 с.
3. Адам Фримен. "ASP.NET Core MVC с примерами на C# для профессионалов". — СПб.: БХВ-Петербург, 2020. — 880 с.
4. Джозеф Альбахари, Бен Альбахари. "C# 8.0 в действии". — М.: Питер, 2019. — 844 с.
5. Брэд Уилсон, Фил Хэк. "ASP.NET. Тренировка по созданию веб-приложений на языке C# и VB.NET для Visual Studio 2019". — СПб.: БХВ-Петербург, 2021. — 736 с.
6. Крейг Шаллер. "Язык программирования C# 8.0 и платформа .NET Core 3.0". — М.: ДМК Пресс, 2020. — 800 с.
7. Бен Форт, Джованни Доменичини, Билл Вагнер. "C# 8.0 Pocket Reference. Instant Help for C# 8.0 Programmers". — O'Reilly Media, 2019. — 244 p.
8. Лесли Лампорт. "SQL: язык структурированных запросов. Теория и практика программирования". — М.: Издательский дом "Вильямс", 2019. — 992 с.
9. Джон Скит. "ASP.NET Core в действии". — Киев: ДМК Пресс, 2019. — 576 с.
10. Джозеф Альбахари, Бен Альбахари. "C# 9.0 и платформа .NET 5.0. Полное руководство". — М.: Питер, 2021. — 992 с
11. Рамур, В. NET Сетевое программирование/ В. Рамур, Э. Кравчек, Н. Логорни. – М.: Дори, 2015. – 500 с.: ил.
12. Троелсен Э. C# и платформа .NET. Библиотека программиста/ Э. Троелсен – СПб.: Русская редакция, 2014. – 796 с.
13. Сейт, К. Дж. SQL и реляционная теория/ К. Дж. Сейт. – М.: Издательский дом «Джорджа», 2010. – 474 с.
14. Петзольд Ч. Программирование для Microsoft Windows на С#. В 2 томах/ Ч. Петзольд. – СПб.: Русская редакция, 2013. – 536 с.
15. Вирт, Никлаус. Алгоритмы и структуры данных/ Н. Вирт. – СПб.: БХВ-Петербург, 2015. – 274 с
16. Шумаков П.В. ADO.NET и создание приложений баз данных в среде Microsoft Visual Studio.NET/ П.В. Шумаков. –: М.: ДИАЛОГ-МИФИ, 2013. – 258 с.
17. Рихтер Д. Программирование на платформе Microsoft .NET Framework/ Д. Рихтер. – СПб.: Русская редакция, 2013. – 512 с.
18. Крёнке, Д. Теория и практика построения баз данных. 8-е изд./ Д. Крёнке. — СПб.: Питер, 2013. — 800 с.: ил.
19. Visual Studio 2017 [Электронный ресурс]. – 2003-2017. - Электрон дан. -– Режим доступа: <https://www.visualstudio.com/ru-ru/visual-studio-homepage-vs.aspx> (дата обращения: 10.04.2018)
20. Кватрани Т. RationalRose 2000 и UML. Визуальное моделирование/ Т. Кватрани. – М.: Издательский дом «Вильямс», 2013. – 583 с

# ПРИЛОЖЕНИЕ А ЛИСТИНГ КОДА

Модуль 1 (Подключение к базе данных):

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Программа\_для\_тестирования

{

class MyDBconnection

{

public MySqlConnection cn;

public void Connect()

{

cn = new MySqlConnection("server=localhost;port=3306;username=root;password=root;database=projects");

}

}

}

Модуль 2 (Базовая структура для работы с базой данных):

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Программа\_для\_тестирования

{

internal class База\_данных

{

MySqlConnection connection = new MySqlConnection("server=localhost;port=3306;username=root;password=root;database=projects");

public void openConnection()

{

if (connection.State == System.Data.ConnectionState.Closed)

connection.Open();

}

public void closeConnection()

{

if (connection.State == System.Data.ConnectionState.Open)

connection.Close();

}public MySqlConnection getConnection()

{return connection;

}} Модуль 3 (Регистрация пользователя):

}using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Программа\_для\_тестирования

{

public partial class Регистрация : Form

{

public Регистрация()

{

InitializeComponent();

usernameF.Text = "Введите имя";

usernameF.ForeColor = Color.Gray;

userSurnameF.Text = "введите Фамилию";

userSurnameF.ForeColor = Color.Gray;

loginF.Text = "Введите логин";

loginF.ForeColor = Color.Gray;

paswordF.Text = "Введите пароль";

paswordF.ForeColor = Color.Gray;

this.paswordF.AutoSize = false;

this.paswordF.Size = new Size(this.paswordF.Size.Width, 57);

}

private void Exit\_Click(object sender, EventArgs e)

{

Application.Exit();

}

Point lastPoint;

private void panel2\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - lastPoint.X;

this.Top += e.Y - lastPoint.Y;

}

}

private void panel2\_MouseDown(object sender, MouseEventArgs e)

{

astPoint = new Point(e.X, e.Y);

}

private void panel1\_MouseDown(object sender, MouseEventArgs e)

{

lastPoint = new Point(e.X, e.Y);

}

private void panel1\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - lastPoint.X;

this.Top += e.Y - lastPoint.Y;

}

}

private void Exit\_MouseMove(object sender, MouseEventArgs e)

{

Exit.ForeColor = Color.Red;

}

private void Exit\_MouseDown(object sender, MouseEventArgs e)

{

Exit.ForeColor = Color.Black;

}

private void usernameF\_Enter(object sender, EventArgs e)

{

if (usernameF.Text == "Введите имя")

usernameF.ForeColor = Color.Black;

usernameF.Text = "";

}

private void usernameF\_Leave(object sender, EventArgs e)

{

if (usernameF.Text == "")

{

usernameF.Text = "Введите имя";

usernameF.ForeColor = Color.Gray;

}

}

private void userSurnameF\_Enter(object sender, EventArgs e)

{

if (userSurnameF.Text == "Введите фамилию")

userSurnameF.ForeColor = Color.Black;

userSurnameF.Text = "";

}

private void userSurnameF\_Leave(object sender, EventArgs e)

{

if (userSurnameF.Text == "")

{

userSurnameF.Text = "Введите фамилию";

userSurnameF.ForeColor = Color.Gray;

}

}

private void loginF\_Enter(object sender, EventArgs e)

{

if (loginF.Text == "Введите логин")

loginF.ForeColor = Color.Black;

loginF.Text = "";

}

private void loginF\_Leave(object sender, EventArgs e)

{

if (loginF.Text == "")

{

loginF.Text = "Введите логин";

loginF.ForeColor = Color.Gray;

}

}

private void paswordF\_Enter(object sender, EventArgs e)

{

if (paswordF.Text == "Введите пароль")

paswordF.ForeColor = Color.Black;

paswordF.Text = "";

}

private void paswordF\_Leave(object sender, EventArgs e)

{

if (paswordF.Text == "")

{

loginF.Text = "пароль";

paswordF.ForeColor = Color.Gray;

}

}

private void buttonRegister\_Click(object sender, EventArgs e)

{

if (usernameF.Text == "Введите имя")

{

MessageBox.Show("Введите имя");

return;

}

if (userSurnameF.Text == "Введите фамилию")

{

MessageBox.Show("Введите фамилию");

return;

}

if (loginF.Text == "Введите логин")

{

MessageBox.Show("Введите логин");

return;

}

if(paswordF.Text == "Введите пароль")

{

MessageBox.Show("Введите пароль");

return;

}

if (isuserExists())

return;

String loginUser = loginF.Text;

String PaswordUser = paswordF.Text;

String UserName = usernameF.Text;

String UserSurname = userSurnameF.Text;

База\_данных db = new База\_данных();

MySqlCommand command = new MySqlCommand("INSERT INTO `??????????????????????` (`login`, `pass`, `name`, `surname`) VALUES (@login, @pass, @name, @surname)", db.getConnection());

command.Parameters.Add("@login", MySqlDbType.VarChar).Value = loginUser;

command.Parameters.Add("@passw", MySqlDbType.VarChar).Value = PaswordUser;

command.Parameters.Add("@name", MySqlDbType.VarChar).Value = UserName;

command.Parameters.Add("@surname", MySqlDbType.VarChar).Value = UserSurname;

db.openConnection();

if (command.ExecuteNonQuery() == 1)

MessageBox.Show("Аккаунт был создан");

else

MessageBox.Show("Аккаунт не был создан");

db.closeConnection();

}

public Boolean isuserExists()

{

База\_данных db = new База\_данных();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM `??????????????????????` WHERE `login` = @uL", db.getConnection());

command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = loginF.Text;

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0)

{

MessageBox.Show("Такой логин уже есть, введите другой");

return true;

}

else

return false;

}

private void label2\_Click(object sender, EventArgs e)

{

this.Hide();

Авторизация authorization = new Авторизация();

authorization.Show();

}

}

}

Модуль 4 (Авторизация пользователя):

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Программа\_для\_тестирования

{

public partial class Авторизация : Form

{

public Авторизация()

{

InitializeComponent();

this.paswordF.AutoSize = false;

this.paswordF.Size = new Size(this.paswordF.Size.Width, 57);

}

private void Exit\_Click\_1(object sender, EventArgs e)

{

Application.Exit();

}

private void Exit\_MouseEnter(object sender, EventArgs e)

{

Exit.ForeColor = Color.Red;

}

private void Exit\_MouseLeave(object sender, EventArgs e)

{

Exit.ForeColor = Color.Black;

}

Point lastPoint;

private void panel2\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - lastPoint.X;

this.Top += e.Y - lastPoint.Y;

}

}

private void panel2\_MouseDown(object sender, MouseEventArgs e)

{

lastPoint = new Point(e.X, e.Y);

}

private void panel1\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - lastPoint.X;

this.Top += e.Y - lastPoint.Y;

}

}

private void panel1\_MouseDown\_1(object sender, MouseEventArgs e)

{

lastPoint = new Point(e.X, e.Y);

}

private void buttonLogin\_Click(object sender, EventArgs e)

{

String loginUser = loginF.Text;

String passUser = paswordF.Text;

База\_данных db = new База\_данных();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM `??????????????????????` WHERE `login` =@uL AND `password` =@uP", db.getConnection());

command.Parameters.Add("@uL", MySqlDbType.VarChar).Value= loginUser;

command.Parameters.Add("@uP", MySqlDbType.VarChar).Value = passUser;

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0)

MessageBox.Show("Yes");

else

MessageBox.Show("No");

}

private void registerlabel\_Click\_1(object sender, EventArgs e)

{

this.Hide();

Регистрация register = new Регистрация();

register.Show();

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

this.Hide();

Form1 mainform = new Form1();

mainform.Show();

}

}

}

Модуль 5 (Главная форма):

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Программа\_для\_тестирования

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Exit\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void Exit\_MouseEnter(object sender, EventArgs e)

{

Exit.ForeColor = Color.Red;

}

private void Exit\_MouseLeave(object sender, EventArgs e)

{

Exit.ForeColor = Color.Black;

}

Point lastPoint;

private void panel2\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - lastPoint.X;

this.Top += e.Y - lastPoint.Y;

}

}

private void panel2\_MouseDown(object sender, MouseEventArgs e)

{

lastPoint = new Point(e.X, e.Y);

}

private void panel1\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - lastPoint.X;

this.Top += e.Y - lastPoint.Y;

}

}

private void panel1\_MouseDown\_1(object sender, MouseEventArgs e)

{

lastPoint = new Point(e.X, e.Y);

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button4\_Click(object sender, EventArgs e)

{

this.Hide();

проекты project = new проекты();

project.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Hide();

Регистрацияпроекта register = new Регистрацияпроекта();

register.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

Авторизация avtor = new Авторизация();

avtor.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Hide();

Регистрация register = new Регистрация();

register.Show();

} }}}

Модуль 6 (Регистрация проекта):

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.StartPanel;

namespace Программа\_для\_тестирования

{

public partial class Регистрацияпроекта : Form

{

public Регистрацияпроекта()

{

InitializeComponent();

}

Point lastPoint;

private void panel1\_MouseDown(object sender, MouseEventArgs e)

{

lastPoint = new Point(e.X, e.Y);

}

private void panel1\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - lastPoint.X;

this.Top += e.Y - lastPoint.Y;

}

}

private void Exit\_MouseMove(object sender, MouseEventArgs e)

{

Exit.ForeColor = Color.Red;

}

private void Exit\_MouseDown(object sender, MouseEventArgs e)

{

Exit.ForeColor = Color.Black;

}

private void Exit\_Click\_1(object sender, EventArgs e)

{

Application.Exit();

}

private void panel2\_MouseMove\_1(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - lastPoint.X;

this.Top += e.Y - lastPoint.Y;

}

}

private void panel2\_MouseDown\_1(object sender, MouseEventArgs e)

{

lastPoint = new Point(e.X, e.Y);

}

private void label5\_Click(object sender, EventArgs e)

{

this.Hide();

Form1 mainform = new Form1();

mainform.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

String ProjectName = nameprojF.Text;

String UserName = usernameF.Text;

String UserSurname = userSurnameF.Text;

База\_данных db = new База\_данных();

MySqlCommand command = new MySqlCommand("INSERT INTO `projects` (`name`, `surname`, `name of project`) VALUES (@name, @surname, @projectn)", db.getConnection());

command.Parameters.Add("@projectn", MySqlDbType.VarChar).Value = ProjectName;

command.Parameters.Add("@name", MySqlDbType.VarChar).Value = UserName;

command.Parameters.Add("@surname", MySqlDbType.VarChar).Value = UserSurname;

db.openConnection();

if (command.ExecuteNonQuery() == 1)

MessageBox.Show("Проект был создан");

else

MessageBox.Show("Проект не был создан");

db.closeConnection();

}

}

}

Модуль 7 (проекты):

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Программа\_для\_тестирования

{

public partial class проекты : Form

{

public проекты()

{

InitializeComponent();

con.Connect();

}

MyDBconnection con = new MyDBconnection();

MySqlCommand command;

MySqlDataAdapter da;

DataTable dt;

string ConnectionString = ("Server=localhost;Port=3306;Database=projects;User Id=root;Password=root;");

private void Exit\_Click(object sender, EventArgs e)

{

Application.Exit();

}

Point lastPoint;

private void panel2\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - lastPoint.X;

this.Top += e.Y - lastPoint.Y;

}

}

private void panel2\_MouseDown(object sender, MouseEventArgs e)

{

lastPoint = new Point(e.X, e.Y);

}

private void panel1\_MouseDown(object sender, MouseEventArgs e)

{

lastPoint = new Point(e.X, e.Y);

}

private void panel1\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - lastPoint.X;

this.Top += e.Y - lastPoint.Y;

}

}

private void Exit\_MouseMove(object sender, MouseEventArgs e)

{

Exit.ForeColor = Color.Red;

}

private void Exit\_MouseDown(object sender, MouseEventArgs e)

{

Exit.ForeColor = Color.Black;

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

Form1 mainform = new Form1();

mainform.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

con.cn.Open();

command = new MySqlCommand("Select \* from projects", con.cn);

command.ExecuteNonQuery();

dt = new DataTable();

da = new MySqlDataAdapter(command);

da.Fill(dt);

dataGridView1.DataSource = dt.DefaultView;

con.cn.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Exit\_Click\_1(object sender, EventArgs e)

{

Application.Exit();

}

}

}

# ПРИЛОЖЕНИЕ В СЛАЙДЫ ПРЕЗЕНТАЦИИ