Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Объектно-ориентированное программирование»

|  |  |
| --- | --- |
|  | «К ЗАЩИТЕ ДОПУСТИТЬ» |
|  | Руководитель курсового проекта  ассистент кафедры информатики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.Д.Владымцев |
|  | \_\_\_.\_\_\_\_.2023 |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовому проекту

на тему:

**«Агрегатор такси»**

БГУИР КП 1-40 04 01 48 ПЗ

|  |  |
| --- | --- |
|  | Выполнил студент группы 153503  Кахновский Евгений Сергеевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |
|  | Курсовой проект представлен  на проверку \_\_\_.\_\_\_\_.2023  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |

Минск 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc135232383)

[1 Анализ используемых источников 4](#_Toc135232384)

[2 Теоретическое обоснование разработки 4](#_Toc135232385)

[3 Паттерны программирования, используемые в разработке приложения 4](#_Toc135232386)

[4 Функциональные возможности программы 4](#_Toc135232387)

[5 Архитектура разрабатываемой программы 4](#_Toc135232388)

[ЗАКЛЮЧЕНИЕ 4](#_Toc135232389)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 4](#_Toc135232390)

[ПРИЛОЖЕНИЕ Б (обязательное) Функциональная схема алгоритма 6](#_Toc135232391)

[ПРИЛОЖЕНИЕ В (обязательное) Блок схема программы 7](#_Toc135232392)

[ПРИЛОЖЕНИЕ Г (обязательное) Графический интерфейс 8](#_Toc135232393)

[ПРИЛОЖЕНИЕ Д (обязательное) Ведомость 9](#_Toc135232394)

****ВВЕДЕНИЕ****

**В современном цифровом мире услуги такси стали неотъемлемой частью нашей повседневной жизни. Они предоставляют удобный и эффективный способ перемещения по городу, позволяя пользователям вызывать и заказывать такси с помощью мобильных устройств. Однако разнообразие приложений и платформ для такси создает вызов для пользователей и разработчиков в поиске подходящего агрегатора такси. В связи с этим, разработка агрегатора такси представляет собой актуальную и перспективную задачу. Агрегатор такси предлагает удобную платформу, объединяющую различные службы такси и предоставляющую пользователям возможность выбора наиболее подходящей и доступной опции такси. Целью данного курсового проекта является разработка агрегатора такси на базе веб-приложения с использованием технологий объектно-ориентированного программирования. Основной задачей проекта является создание функциональной и удобной системы, которая позволит пользователям заказывать такси, выбирать оптимальные маршруты и получать информацию о доступных вариантах такси. Для достижения поставленных целей и задач, в проекте используются следующие технологии и подходы: ASP.NET MVC: выбранная платформа разработки веб-приложения, которая обеспечивает удобную модель разделения ответственности и обработку запросов пользователя. Entity Framework: используется для работы с базой данных и обеспечения удобного взаимодействия с данными, включая операции чтения, записи и изменения. JavaScript (JS), HTML и CSS: используются для создания интерактивного пользовательского интерфейса и обеспечения привлекательного внешнего вида приложения. Razor Pages: применяются для удобной генерации динамического содержимого страниц и интеграции кода на серверной стороне. Bootstrap 5: фреймворк CSS, который обеспечивает адаптивный дизайн и упрощает создание стильного и современного пользовательского интерфейса. N-Layer architecture (DAL, Domain, BLL, WEB): применение многоуровневой архитектуры позволяет разделить компоненты системы на логические слои и обеспечить их независимость, повышая гибкость и переиспользуемость кода. Паттерны Unit of Work и Repository: используются для организации работы с базой данных, обеспечивая единообразный и удобный доступ к данным и их модификацию. В рамках проекта администратор будет иметь возможность управлять пользователями и просматривать информацию о заказах, а клиенты смогут редактировать свой профиль, загружать фотографии, делать заказы и просматривать их историю.**

**Настоящая пояснительная записка охватывает основные аспекты разработки и реализации агрегатора такси на платформе ASP.NET MVC. В дальнейших разделах будут подробно рассмотрены этапы проектирования, реализации и тестирования системы, а также описаны использованные технологии и методологии разработки.**

# Анализ используемых источников

Анализ

# Теоретическое обоснование разработки

Заголовок: Разработка агрегатора такси с использованием технологий .NET, ASP.NET и Bing Maps

Введение:

В современном мире услуги такси стали неотъемлемой частью нашей повседневной жизни. С целью повышения удобства для пассажиров и оптимизации процесса заказа такси был разработан агрегатор такси, который объединяет несколько служб такси на одной платформе. В данной курсовой работе мы рассмотрим разработку агрегатора такси с использованием платформы .NET, ASP.NET, ASP.NET Core MVC, Razor Page, MVC, c#, SQLite, EF, HTML5+CSS3+JS, Bootstrap5, WEB API и Bing Maps.

Описание функциональности агрегатора такси:

Регистрация и аутентификация пользователей: Реализация системы регистрации и входа пользователей для доступа к функциональности агрегатора такси.

Поиск и отображение доступных такси: Разработка механизма поиска и отображения доступных такси на основе информации, предоставленной службами такси.

Заказ такси: Создание механизма оформления заказа такси, включая выбор типа автомобиля, указание места назначения и прочих дополнительных опций.

Отслеживание поездки: Реализация функции отслеживания текущего положения такси в режиме реального времени для обеспечения удобства пассажиров.

Оплата и отзывы: Интеграция с платежными системами для обеспечения безопасной оплаты поездки, а также возможность оставления отзывов и оценок о сервисе.

Архитектура и технологии разработки:

Использование платформы .NET: Для разработки агрегатора такси выбрана платформа .NET, которая предоставляет мощный инструментарий для создания веб-приложений.

ASP.NET и ASP.NET Core MVC: Для реализации веб-интерфейса и бизнес-логики выбраны ASP.NET и ASP.NET Core MVC, которые обеспечивают высокую производительность и гибкость разработки.

Razor Pages и MVC: Для организации представления и контроллеров веб-приложения используются Razor Pages и MVC, обеспечивая разделение задач и повышение модульности.

C# и SQLite: В разработке используется язык программирования C# и база данных SQLite для хранения информации о пользователях, заказах такси и отзывах.

HTML5+CSS3+JS и Bootstrap5: Для создания пользовательского интерфейса применяются современные веб-технологии HTML5, CSS3 и JS, а также фреймворк Bootstrap5 для обеспечения адаптивности и стилизации.

WEB API: Для обмена данными между клиентской и серверной частями приложения используется WEB API, обеспечивая простоту интеграции и расширяемость.

Интеграция с Bing Maps:

Использование Bing Maps API: Для интеграции карты и маршрутизации в агрегаторе такси применяется Bing Maps API, предоставляющий функциональность для отображения местоположения такси и определения оптимального маршрута.

Отображение местоположения такси: С помощью Bing Maps API реализуется отображение текущего местоположения доступных такси на карте, что помогает пользователям принять решение о выборе подходящего такси.

Маршрутизация и навигация: Интеграция с Bing Maps позволяет определить оптимальный маршрут от места отправления до места назначения, обеспечивая эффективную навигацию и сокращение времени поездки.

Заключение:

В данной курсовой работе был представлен процесс разработки агрегатора такси с использованием технологий .NET, ASP.NET, ASP.NET Core MVC, Razor Page, MVC, c#, SQLite, EF, HTML5+CSS3+JS, Bootstrap5, WEB API и Bing Maps. Результатом работы является функциональное веб-приложение, позволяющее пользователям удобно заказывать такси, отслеживать поездку и оставлять отзывы о сервисе. Использование данных технологий обеспечивает высокую производительность, гибкость и удобство использования для пользователей.

**.NET**

Платформа .NET является мощным инструментарием для разработки разнообразных приложений, включая веб-приложения, на различных языках программирования, таких как C#, VB.NET и F#. Она разработана компанией Microsoft и предоставляет программистам и разработчикам набор инструментов, фреймворков и библиотек для эффективного создания высокопроизводительных и масштабируемых приложений.

Вот несколько ключевых преимуществ платформы .NET, которые привели к выбору этой платформы для разработки агрегатора такси:

Многоплатформенность: Платформа .NET поддерживает разработку приложений для различных операционных систем, включая Windows, macOS и Linux. Это позволяет агрегатору такси быть доступным для широкой аудитории пользователей, независимо от используемой платформы.

Мощный язык программирования: С использованием языка программирования C#, который является одним из основных языков платформы .NET, разработчики получают преимущества типизации, объектно-ориентированного программирования и многопоточности. C# обладает лаконичным синтаксисом, обширной стандартной библиотекой и широкой поддержкой сообщества разработчиков.

Большая экосистема инструментов и фреймворков: Платформа .NET предоставляет широкий выбор инструментов и фреймворков для разработки веб-приложений. В случае агрегатора такси, были использованы ASP.NET, ASP.NET Core MVC, Razor Page и WEB API, которые обеспечивают высокую производительность, гибкость разработки, модульность и удобство взаимодействия с клиентами.

Безопасность: Платформа .NET предоставляет набор механизмов и инструментов для обеспечения безопасности приложений, включая автоматическую проверку типов, обработку исключений, аутентификацию и авторизацию. Это важно для агрегатора такси, где обработка персональных данных и обеспечение безопасности платежей являются критическими аспектами.

Поддержка баз данных: Платформа .NET обеспечивает простую интеграцию с различными системами управления базами данных. В случае агрегатора такси была использована SQLite для хранения информации о пользователях, заказах такси и отзывах.

Интеграция с внешними сервисами: Платформа .NET обладает богатым набором инструментов для интеграции с внешними сервисами, такими как Bing Maps API, который был использован для интеграции карт и маршрутизации в агрегаторе такси.

В целом, выбор платформы .NET для разработки агрегатора такси обусловлен её многоплатформенностью, поддержкой современных языков программирования, мощными инструментами разработки, безопасностью и поддержкой интеграции с внешними сервисами, что позволяет эффективно создавать надежные и высокофункциональные веб-приложения.

ASP.NET

ChatGPT

ASP.NET Core MVC является фреймворком для разработки веб-приложений на платформе .NET Core. Он предоставляет мощные инструменты для создания веб-интерфейса и бизнес-логики, обеспечивая высокую производительность и гибкость разработки. Вот несколько преимуществ ASP.NET Core MVC и причины выбора этого фреймворка для реализации агрегатора такси:

Кросс-платформенность: ASP.NET Core MVC основан на платформе .NET Core, что позволяет разрабатывать приложения, которые могут работать на различных операционных системах, включая Windows, macOS и Linux. Это дает возможность обеспечить доступность агрегатора такси для широкой аудитории пользователей.

Высокая производительность: ASP.NET Core MVC предлагает оптимизации производительности, такие как механизмы кэширования, асинхронное выполнение и масштабируемость. Это позволяет обеспечить отзывчивость и быстродействие приложения, что особенно важно для агрегатора такси, где время отклика имеет большое значение.

Модульность и гибкость: ASP.NET Core MVC основан на принципах модульности и разделения ответственности. Он позволяет разделить приложение на логические компоненты (контроллеры, представления, модели), что упрощает поддержку кода и его масштабируемость. Также фреймворк предлагает гибкую конфигурацию маршрутов и позволяет использовать различные стратегии маршрутизации.

Богатая экосистема: ASP.NET Core MVC имеет обширную экосистему инструментов, библиотек и расширений, которые облегчают разработку веб-приложений. Это включает в себя ORM-фреймворк Entity Framework для работы с базой данных, библиотеки для обработки входящих запросов, валидации данных, авторизации и аутентификации пользователей и многое другое.

Тестирование и отладка: ASP.NET Core MVC предоставляет средства для тестирования и отладки приложений, что упрощает процесс разработки и обеспечивает стабильность и надежность приложения. Разработчики могут проводить модульное, интеграционное и функциональное тестирование своего кода, а также использовать мощные инструменты отладки для идентификации и устранения ошибок.

Итак, выбор ASP.NET Core MVC для реализации агрегатора такси обусловлен его кросс-платформенностью, высокой производительностью, модульностью, богатой экосистемой инструментов и удобством в тестировании и отладке. Этот фреймворк обеспечивает разработчикам мощный инструментарий для создания надежных и масштабируемых веб-приложений, соответствующих требованиям агрегатора такси.

RAZOR PAGE

ChatGPT

Технология Razor Pages является частью фреймворка ASP.NET Core и предоставляет удобный способ разработки веб-приложений, основанных на модели просмотра и шаблонах Razor. Вот несколько преимуществ и причин выбора технологии Razor Pages для реализации агрегатора такси:

Простота и интуитивность: Razor Pages предлагает простую и интуитивно понятную модель программирования, основанную на разделении страниц на разделы (Razor Pages) и соответствующие им модели просмотра (View Models). Это делает процесс разработки более понятным и легким для новых разработчиков.

Удобство взаимодействия с данными: Razor Pages обеспечивает прямую интеграцию с моделями просмотра и обработку данных. Разработчики могут легко передавать данные между страницами и моделями просмотра, что упрощает реализацию бизнес-логики и взаимодействие с базой данных.

Гибкость и масштабируемость: Razor Pages позволяет создавать маленькие и самодостаточные страницы, что облегчает разделение и модульность кода. Разработчики могут легко масштабировать приложение, добавляя новые Razor Pages или модели просмотра, не затрагивая другие части приложения.

Использование шаблонов Razor: Razor Pages использует синтаксис Razor для создания динамических и гибких шаблонов представлений. Это позволяет разработчикам создавать легко читаемые и поддерживаемые представления, в которых можно использовать код C# для динамической генерации контента.

Поддержка асинхронности: Razor Pages предоставляет встроенную поддержку асинхронных операций, что позволяет выполнять длительные операции без блокирования основного потока выполнения. Это повышает отзывчивость приложения и общую производительность.

Интеграция с другими компонентами ASP.NET Core: Razor Pages хорошо интегрируется с другими компонентами ASP.NET Core, такими как маршрутизация, инъекция зависимостей, аутентификация и авторизация. Это облегчает взаимодействие с другими частями приложения и использование их функциональности.

Итак, выбор технологии Razor Pages для реализации агрегатора такси основан на её простоте и интуитивности, удобстве взаимодействия с данными, гибкости и масштабируемости, использовании шаблонов Razor, поддержке асинхронности и интеграции с другими компонентами ASP.NET Core. Это позволяет разработчикам эффективно создавать и поддерживать веб-приложение, отвечающее требованиям агрегатора такси.

C#

ChatGPT

C# (произносится "си шарп") — это мощный и элегантный объектно-ориентированный язык программирования, разработанный компанией Microsoft. Вот несколько преимуществ и причин выбора языка C# для реализации агрегатора такси:

Язык с широким функциональным спектром: C# обладает обширным функциональным набором, включая поддержку сильной типизации, управления памятью, событий, делегатов, асинхронного программирования, LINQ (Language Integrated Query) и многое другое. Это обеспечивает разработчикам гибкость и возможность реализовывать сложные бизнес-логики и функциональности в приложении.

Интеграция с платформой .NET: C# является основным языком программирования для разработки на платформе .NET, которая предлагает мощный набор инструментов и библиотек для создания веб-приложений. Благодаря этому, C# обладает преимуществами интеграции с другими технологиями и компонентами .NET, такими как ASP.NET, Entity Framework, LINQ и многими другими.

Широкая поддержка сообщества и ресурсов: C# является одним из самых популярных языков программирования, и за ним стоит большое активное сообщество разработчиков. Это обеспечивает доступность множества ресурсов, форумов, библиотек и инструментов поддержки, которые могут быть использованы для разработки агрегатора такси.

Удобство и читаемость кода: C# имеет чистый и понятный синтаксис, что делает код на языке C# легко читаемым и понятным для разработчиков. Это способствует разработке и поддержке приложения, особенно в командной работе, где читаемый и понятный код играет важную роль.

Платформонезависимость: C# можно использовать для разработки приложений, которые могут работать на различных платформах, таких как Windows, macOS и Linux. Это обеспечивает гибкость и доступность приложения для широкой аудитории пользователей.

Итак, выбор языка C# для реализации агрегатора такси обусловлен его широким функциональным спектром, интеграцией с платформой .NET, поддержкой сообщества и ресурсов, удобством и читаемостью кода, а также платформонезависимостью. Это обеспечивает эффективную разработку приложения с использованием C#.

SQLITE

ChatGPT

SQLite - это компактная, серверная база данных, которая работает без необходимости отдельного сервера. Вот несколько преимуществ и причин выбора технологии SQLite для реализации агрегатора такси:

Простота использования: SQLite предоставляет простой и интуитивно понятный способ хранения и управления данными. Она не требует настройки сложной инфраструктуры сервера баз данных и может быть легко интегрирована в приложение. Это упрощает разработку и развертывание приложения, особенно для малых и средних проектов.

Компактность и эффективность: SQLite представляет собой самодостаточную библиотеку, которая хранит базу данных в одном файле. Это делает ее компактной и эффективной в использовании ресурсов, что особенно важно для приложений с ограниченными ресурсами, таких как мобильные приложения.

Поддержка SQL: SQLite полностью поддерживает стандартный язык структурированных запросов SQL. Это позволяет разработчикам выполнять сложные запросы и манипулировать данными с помощью знакомого и мощного языка. SQLite также поддерживает транзакции, что обеспечивает целостность данных и безопасность операций.

Переносимость: SQLite может работать на различных операционных системах, включая Windows, macOS, Linux и другие. Это обеспечивает переносимость приложения, позволяя запускать его на различных платформах без необходимости внесения значительных изменений в код.

Надежность: SQLite обеспечивает надежное хранение данных и обеспечивает целостность информации при возникновении сбоев или аварийных ситуаций. Она также предоставляет механизмы резервного копирования и восстановления данных для обеспечения безопасности информации.

Выбор технологии SQLite для реализации агрегатора такси обусловлен ее простотой использования, компактностью, поддержкой SQL, переносимостью и надежностью. SQLite предоставляет эффективное и надежное решение для хранения и управления данными в приложении.

HTML+CSS+JSs

ChatGPT

Технологии HTML5, CSS3 и JavaScript (JS) являются основными инструментами для разработки современных веб-интерфейсов. Вот несколько преимуществ и причин выбора этого набора технологий для реализации агрегатора такси:

HTML5: HTML5 представляет собой последнюю версию стандарта HTML, который используется для создания структуры и разметки веб-страниц. Он предлагает множество новых элементов и функций, таких как семантические теги, мультимедиа-элементы, хранилище данных (localStorage) и другие. Это позволяет разработчикам создавать более современные и интерактивные пользовательские интерфейсы.

CSS3: CSS3 является последней версией стандарта CSS и предоставляет богатые возможности для стилизации и визуализации веб-страниц. Он включает в себя новые свойства и модули, такие как гибкая сетка (flexbox), анимации, переходы, тени, градиенты и многое другое. Это позволяет создавать эффектные дизайны и адаптивные пользовательские интерфейсы.

JavaScript: JavaScript является клиентским языком программирования, который позволяет добавлять динамическую функциональность и взаимодействие на веб-страницах. Он поддерживает множество возможностей, таких как манипуляция DOM-деревом, обработка событий, асинхронные запросы к серверу (AJAX), анимации и многое другое. JavaScript является неотъемлемой частью создания интерактивных и динамических пользовательских интерфейсов.

Кросс-платформенность: HTML5, CSS3 и JavaScript являются кросс-платформенными технологиями, что означает, что веб-приложение, созданное с использованием этих технологий, может работать на различных платформах и устройствах, включая компьютеры, мобильные устройства и планшеты. Это обеспечивает гибкость и доступность приложения для широкой аудитории пользователей.

Обширное сообщество и ресурсы: HTML5, CSS3 и JavaScript имеют огромное и активное сообщество разработчиков, а также обилие ресурсов, библиотек и инструментов для поддержки разработки. Это обеспечивает доступность множества ресурсов, помощи и решений, которые могут быть использованы в процессе разработки.

Выбор HTML5, CSS3 и JavaScript для реализации агрегатора такси обусловлен их мощными возможностями в создании современных пользовательских интерфейсов, кросс-платформенностью, поддержкой сообщества и обширными ресурсами. Это позволяет разработчикам создавать эстетически привлекательные и интерактивные интерфейсы для удобства пользователей.

BOOTSTRAP

ChatGPT

2 / 2

Bootstrap 5 - это популярный фреймворк для разработки веб-интерфейсов, основанный на HTML, CSS и JavaScript. Вот несколько преимуществ и причин выбора Bootstrap 5 для реализации агрегатора такси:

Готовые компоненты и макеты: Bootstrap 5 предоставляет широкий набор готовых компонентов и макетов, таких как кнопки, формы, навигационные панели, модальные окна, карусели и многое другое. Это позволяет разработчикам быстро и легко создавать стильные и современные интерфейсы без необходимости разработки с нуля.

Адаптивный дизайн: Bootstrap 5 обеспечивает адаптивный дизайн, который автоматически адаптируется к различным размерам экранов и устройствам, включая мобильные устройства. Это позволяет создавать веб-интерфейсы, которые выглядят хорошо и функционируют эффективно на любом устройстве.

Гибкость и настраиваемость: Bootstrap 5 предоставляет множество настраиваемых опций и переменных, которые позволяют разработчикам легко настроить внешний вид и поведение компонентов в соответствии с требованиями проекта. Это дает возможность создавать уникальные дизайны и адаптировать фреймворк под конкретные потребности приложения.

Кросс-браузерная совместимость: Bootstrap 5 обеспечивает совместимость с различными веб-браузерами, включая последние версии Chrome, Firefox, Safari, Edge и других популярных браузеров. Это гарантирует, что ваше веб-приложение будет работать правильно и выглядеть одинаково на разных платформах.

Поддержка и активное сообщество: Bootstrap имеет огромную базу пользователей и активное сообщество разработчиков. Это означает, что есть множество ресурсов, документации, тем оформления, плагинов и поддержки, которые могут помочь в разработке и решении возникающих проблем.

Выбор Bootstrap 5 для реализации агрегатора такси обусловлен его готовыми компонентами, адаптивным дизайном, гибкостью, кросс-браузерной совместимостью и поддержкой сообщества

EF

ChatGPT

Entity Framework (EF) - это фреймворк объектно-реляционного отображения (ORM), разработанный для работы с базами данных. Он предоставляет удобный способ взаимодействия с базой данных через объектно-ориентированный подход. Вот несколько преимуществ и причин выбора EF вместе с SQLite для реализации агрегатора такси:

Упрощенное взаимодействие с базой данных: EF предоставляет простой и интуитивно понятный API для работы с базой данных. Он позволяет разработчикам работать с данными в виде объектов и коллекций, а не напрямую с SQL-запросами и таблицами. Это делает процесс взаимодействия с базой данных более удобным и эффективным.

Автоматическое отображение объектов на таблицы: EF обеспечивает автоматическое отображение объектов приложения на соответствующие таблицы в базе данных. Разработчикам не нужно писать SQL-запросы или создавать схему базы данных вручную. Это упрощает и ускоряет процесс разработки и поддержки приложения.

Работа с различными базами данных: EF поддерживает несколько провайдеров баз данных, включая SQLite, SQL Server, MySQL, PostgreSQL и другие. Это позволяет разработчикам использовать EF с разными базами данных, в зависимости от требований проекта. В случае выбора SQLite в качестве базы данных для агрегатора такси, использование EF позволяет удобно и эффективно работать с этой легкой и компактной базой данных.

Управление связями между объектами: EF позволяет управлять связями между объектами в приложении, предоставляя механизмы для определения отношений "один-ко-многим", "многие-ко-многим" и других типов связей. Это облегчает работу с данными, требующими связи между различными таблицами.

Повышение производительности и безопасности: EF предлагает возможности кэширования данных, оптимизации запросов и управления транзакциями, что способствует повышению производительности приложения. Он также предоставляет механизмы защиты от атак типа SQL-инъекций и других уязвимостей, обеспечивая безопасность взаимодействия с базой данных.

Выбор EF вместе с SQLite для агрегатора такси обусловлен удобством и простотой работы с данными, автоматическим отображением объектов на таблицы, поддержкой различных баз данных и возможностью управления связями между объектами. Кроме того, EF предоставляет механизмы для повышения производительности и обеспечения безопасности взаимодействия с базой данных.

# Паттерны программирования, используемые в разработке приложения

Технология MVC (Model-View-Controller) является архитектурным шаблоном разработки программного обеспечения, который был применен в фреймворках ASP.NET и ASP.NET Core для создания веб-приложений. Он предлагает разделение бизнес-логики, пользовательского интерфейса и управления состоянием приложения на три компонента: модель (Model), представление (View) и контроллер (Controller). Вот несколько преимуществ и причин выбора технологии MVC для реализации агрегатора такси:

Разделение ответственности: MVC разделяет компоненты приложения на отдельные слои, каждый из которых отвечает за свои функциональности. Модель отвечает за бизнес-логику и обработку данных, представление отображает информацию пользователю, а контроллер обрабатывает входящие запросы и координирует взаимодействие между моделью и представлением. Это делает код более понятным, поддерживаемым и масштабируемым.

Гибкость и расширяемость: MVC позволяет легко добавлять новые функциональности и изменять существующую логику без влияния на другие компоненты приложения. Каждый компонент имеет свою собственную задачу и может быть заменен или модифицирован независимо от других компонентов. Это упрощает разработку и поддержку приложения, особенно при необходимости внесения изменений или добавления новых функций.

Улучшенная тестируемость: Модель, представление и контроллер могут быть легко тестируемыми независимо друг от друга. Таким образом, разработчики могут проводить модульное тестирование каждого компонента для обеспечения его правильной работы. Это способствует созданию надежного и стабильного приложения.

Поддержка множества клиентских платформ: MVC позволяет создавать веб-приложения, которые могут быть использованы на различных клиентских платформах, включая компьютеры, мобильные устройства и планшеты. Это обеспечивает гибкость и доступность приложения для широкой аудитории пользователей.

Четкое разделение ролей: MVC позволяет разделить роли разработчиков на различные задачи. Например, дизайнеры могут работать над представлениями, бизнес-логика может быть реализована разработчиками модели, а контроллеры могут быть созданы разработчиками, ответственными за обработку запросов и управление потоком данных. Это улучшает эффективность работы команды разработки.

Таким образом, выбор технологии MVC для реализации агрегатора такси обусловлен ее способностью к разделению ответственности, гибкостью и расширяемостью, улучшенной тестируемостью, поддержкой множества клиентских платформ и четким разделением ролей в команде разработки. Это обеспечивает эффективность и надежность разработки приложения.

# Функциональные возможности программы

вохможности

# Архитектура разрабатываемой программы

архитектура

ЗАКЛЮЧЕНИЕ

заключение

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

литературПРИЛОЖЕНИЕ А  
(обязательное)  
Исходный код программы

ПРИЛОЖЕНИЕ Б  
(обязательное)  
Функциональная схема алгоритма

ПРИЛОЖЕНИЕ В  
(обязательное)  
Блок схема программы

ПРИЛОЖЕНИЕ Г  
(обязательное)  
Графический интерфейс

ПРИЛОЖЕНИЕ Д  
(обязательное)  
Ведомость