МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Новосибирский национальный исследовательский государственный университет»

(Новосибирский государственный университет, НГУ)

Структурное подразделение Новосибирского государственного университета – Высший колледж информатики Университета (ВКИ НГУ)

КАФЕДРА ИНФОРМАТИКИ

Отчет по курсовому проекту

ПМ 01. Разработка программных модулей

**Управление личными финансами**

|  |  |
| --- | --- |
| Руководитель | Волховицкий Г. В.  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2023 г. |
| Студент 3 курса  гр. 107г2 | Кузнецов Д. Н.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2023 г. |

Новосибирск

2024

СОДЕРЖАНИЕ

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ 4](#_Toc158762050)

[ВВЕДЕНИЕ 5](#_Toc158762051)

[1 ПОСТАНОВКА ЗАДАЧИ ВКР 6](#_Toc158762052)

[1.1 Бизнес-требования 6](#_Toc158762053)

[1.2 Пользовательские требования 6](#_Toc158762055)

[1.3 Системные требования 6](#_Toc158762056)

[1.4 Требования к графическому пользовательскому интерфейсу 7](#_Toc158762057)

[1.4.1 Требования к внешнему виду пользовательского интерфейса: 7](#_Toc158762058)

[1.4.2 Требования по доступу к функциональности системы при помощи пользовательского интерфейса: 7](#_Toc158762059)

[1.5 План-график выполнения ВКР 8](#_Toc158762060)

[2 АНАЛИЗ ТРЕБОВАНИЙ И ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ 10](#_Toc158762061)

[2.1 Описание предметной области задачи ВКР 10](#_Toc158762062)

[2.1.1 Информационные объекты предметной области и взаимосвязи между ними 10](#_Toc158762063)

[2.1.2 Информационные и функциональные потребности пользователей разрабатываемой ПС (ПМ) 10](#_Toc158762064)

[2.1.3 Обзор существующих программных реализаций решения задачи 10](#_Toc158762065)

[2.1.4 Концептуальное обоснование разработки 12](#_Toc158762066)

[2.2 Классы и характеристики пользователей 12](#_Toc158762067)

[2.3 Функциональные требования 13](#_Toc158762068)

[2.3.1 Определение функциональных возможностей ПС 13](#_Toc158762069)

[2.3.2 Описание прецедентов 14](#_Toc158762070)

[2.4 Нефункциональные требования 15](#_Toc158762071)

[3 ВЫБОР ПРОГРАММНЫХ СРЕД И СРЕДСТВ РАЗРАБОТКИ 16](#_Toc158762072)

[3.1 Сравнительный анализ имеющихся возможностей по выбору средств разработки 16](#_Toc158762073)

[3.2 Характеристика выбранных программных сред и средств 17](#_Toc158762074)

[4 АЛГОРИТМ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ 19](#_Toc158762075)

[4.2 Пользовательский интерфейс ПС 20](#_Toc158762076)

[4.2.1 Взаимодействие пользователей с ПС 20](#_Toc158762077)

[4.2.2 Проектирование пользовательских сценариев 22](#_Toc158762078)

[4.2.2.1 Проектирование пользовательских сценариев для проекта «Афиша концертов» в г. Новосибирск: 22](#_Toc158762079)

[4.2.2.2 Структура диалогов: 22](#_Toc158762080)

[4.2.3 Определение операций пользователей 23](#_Toc158762081)

[4.2.4 Составление функциональных блоков 24](#_Toc158762082)

[4.2.5 Проектирование структуры экранов ПС (ПМ) и схемы навигации 24](#_Toc158762083)

[4.2.6 Низкоуровневое проектирование 25](#_Toc158762084)

[4.3 Входные, выходные и промежуточные данные 28](#_Toc158762085)

[4.4 Разработка базы данных, реализуемой в рамках ПС 30](#_Toc158762086)

[4.6 Алгоритмы использования применяемых программных технологий 32](#_Toc158762087)

[4.7 Архитектура и схема функционирования ПС (ПМ) 33](#_Toc158762088)

[5 ТЕСТИРОВАНИЕ И ОПТИМИЗАЦИЯ 36](#_Toc158762089)

[5.1 План тестирования 36](#_Toc158762090)

[5.2 Результаты тестирования 36](#_Toc158762091)

[6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 39](#_Toc158762092)

[ЗАКЛЮЧЕНИЕ 41](#_Toc158762093)

ЛИТЕРАУРА…………………………………………………………………….42

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ

ВВЕДЕНИЕ

В современном мире досуговая деятельность играет важную роль в жизни общества, а управление личными финансами становятся все более популярными и востребованными среди широкой аудитории. Информация о личный финансах таких как: доходы, расходы, крупные и маленькие транзакции.

Стремительное развитие технологий и наличие доступа к интернету у большинства населения позволили создать удобные ресурсы для ознакомления с информацией как управлять своими финансами. Однако, несмотря на наличие различных онлайн-платформ, часто возникают проблемы с поиском удобного сервиса.

Целью настоящего проекта является разработка системы «управления финансами», которая будет предоставлять пользователям актуальную информацию о их личных финансах. Это позволит упростить процесс работы с финансами более удобным и эффективным для каждого интересующегося.

Несмотря на существующие веб-ресурсы с информацией, многие из них могут быть неактуальными, неудобными в использовании или не предоставлять всю необходимую информацию. Поэтому необходимо разработать новое программное решение, которое устранит эти недостатки и обеспечит пользователям полноценный доступ работе с личными финансами.

1 ПОСТАНОВКА ЗАДАЧИ ВКР

## **Бизнес-требования**

## Целью разработки программного средства «управление личными финансами» является создание удобного инструмента для предоставления информации о финансах. Проект предназначен для людей которые хотят следить за своими финансами.

## **Пользовательские требования**

1. Пользователи должны иметь возможность просматривать информацию о своих поступлениях, тратах.
2. Разработка функционала регистрации пользователей для создания персонализированных аккаунтов.

## **Системные требования**

1. Модуль управления финансами: содержит информацию о датах поступления средств,.
2. Модуль регистрации пользователей: обеспечивает возможность создания и управления личными аккаунтами пользователей.

## **1.4** **Требования к графическому пользовательскому интерфейсу**

### ***1.4.1 Требования к внешнему виду пользовательского интерфейса:***

* Привлекательный дизайн: Интерфейс должен быть современным, стильным и привлекательным для пользователей всех возрастов.
* Цветовая гамма: Использование приятных цветов, сочетающихся и не утомляющих глаза.
* Интуитивная навигация: Простая и понятная структура интерфейса, легко доступные и понятные разделы с информацией.

### ***1.4.2 Требования по доступу к функциональности системы при помощи пользовательского интерфейса:***

* Отображение финансов: Информация о датах поступления, расходах должна быть представлена ясно.
* Регистрация пользователя: Легкий и понятный процесс регистрации новых пользователей с возможностью заполнения основной информации и структурированных полей.
* Функционал фильтрации: Пользователи должны иметь возможность легко находить интересующие их правки.
* Интерактивные элементы: Использование интерактивных элементов, таких как кнопки, карусели и т.д., для повышения удобства использования и привлекательности интерфейса.
* Простота и понятность: Дизайн должен быть минималистичным, информативным и легким для восприятия и навигации пользователями разного уровня опыта.

## **1.5 План-график выполнения ВКР**

1. Анализ требований (1 неделя):

* Изучение основных требований к разработке проекта.
* Проведение анализа рынка

1. Проектирование пользовательского интерфейса (2 недели):

* Создание макетов и прототипов пользовательского интерфейса.
* Утверждение дизайн-концепции.

1. Разработка базы данных (1 неделя):

* Проектирование структуры базы данных для хранения информации о финансах и пользователях.
* Разработка схемы базы данных.

1. Разработка пользовательского интерфейса (4 недели):

* Верстка и программирование интерфейса на основе утвержденного дизайна.

1. Разработка функциональности (3 недели):

* Создание возможности добавления новых концертов в систему.
* Реализация функционала регистрации и аутентификации пользователя.

1. Интеграция и тестирование (1 неделя):

* Интеграция всех компонентов проекта (база данных, функциональность, интерфейс).
* Проведение тестирования работоспособности и безопасности системы.

1. Доработка и оптимизация (3 недели):

* Исправление выявленных ошибок и недочетов.
* Оптимизация производительности и интерфейса.

1. Документирование и подготовка к защите (1 неделя):

* Написание отчётов и документации по проекту.

Контрольные точки проекта:

* Подтверждение дизайн-концепции и прототипа интерфейса.
* Успешное завершение тестирования функциональности и безопасности.
* Окончательная подготовка к защите и сдача выпускной квалификационной работы.

Календарный план

|  |  |  |  |
| --- | --- | --- | --- |
| Вид работы | Дата начала | Примерная дата окончания | Срок завершения |
| Анализ требований | 16.09.2023 | 22.09.2023 | 1 неделя |
| Проектирование пользовательского интерфейса | 23.09.2023 | 05.10.2023 | 2 недели |
| Разработка базы данных | 06.11.2023 | 12.11.2023 | 1 неделя |
| Разработка пользовательского интерфейса | 13.11.2023 | 10.12.2023 | 4 недели |
| Разработка функциональности | 11.12.2023 | 30.01.2024 | 3 недели |
| Интеграция и тестирование | 04.01.2024 | 10.01.2024 | 1 неделя |
| Доработка и оптимизация | 11.01.2024 | 31.01.2024 | 3 недели |
| Документирование и подготовка к защите | 02.02.2024 | 09.02.2024 | 1 неделя |

# **2 АНАЛИЗ ТРЕБОВАНИЙ И ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ**

## **2.1 Описание предметной области задачи ВКР**

### ***2.1.1 Информационные объекты предметной области и взаимосвязи между ними***

Информационные объекты предметной области «управление личными финансами» включают:

* Финансы такие как транзакции(поступления и расходы) .
* Пользователи: информация о зарегистрированных пользователях.
* //Регистрация: процесс регистрации пользователей в системе.

### ***2.1.2 Информационные и функциональные потребности пользователей разрабатываемой ПС (ПМ)***

Информационные потребности:

* Просмотр своих трат и поступлений.
* Получение подробной информации .
* Возможность регистрации в системе.

Функциональные потребности:

* Регистрация пользователей.
* Поиск и фильтрация по различным критериям.

### ***2.1.3 Обзор существующих программных реализаций решения задачи***

Для проекта «работа с личными финансами» с ограничением можно рассмотреть следующие аналоги:

### ***2.1.4 Концептуальное обоснование разработки***

Разработка системы «работа с личными финансами» обосновывается удовлетворением потребностей пользователей в получении удобного инструмента для работы со своими финансами. Уникальный аспект проекта может быть в предоставлении персонализированных рекомендаций на основе предпочтений пользователя.

## **2.2 Классы и характеристики пользователей**

1. Гость:

* Описание: Пользователь, который посещает сайт без авторизации, но также может зарегистрироваться.
* Характеристики: Может просматривать информацию о финансах, дате, времени, описании.

1. Администратор:

* Описание: Пользователь с расширенными правами для управления платформой. Может добавлять информацию о своих мероприятиях на платформу «работа с личными финансами».
* Характеристики: Может моделировать контент, управлять пользователями, решать спорные ситуации, проводить аналитику и развитие проекта. Также создает и редактирует события, добавляет описания, загружает фотографии и видео.

Каждая из этих ролей играет важную функцию в обеспечении удобства и полезности для пользователей платформы «работа с личными финансами».

## **Функциональные требования**

Функциональные требования для работа с личными финансами определяют, какие функции должны быть доступны веб-приложению или сервису, предоставляющем информацию о финансах.

### ***2.3.1 Определение функциональных возможностей ПС***

Некоторые примеры функциональных требований для такого проекта:

1. Регистрация и аутентификация:

* Пользователи должны иметь возможность создавать учетные записи с использованием электронной почты.
* Предусмотреть процесс аутентификации пользователей при входе в систему.

1. Просмотр и поиск концертов:

* Пользователи должны получать доступ к работе с личными финансами
* Реализовать фильтры для удобного поиска по дате и сумме.

1. Администрирование платформы:

* Администратор должен иметь возможность управлять пользователями, контентом, модерировать комментарии и разрешать спорные ситуации.
* Предусмотреть интерфейс для добавления информации.
* Обеспечить возможность управления данными о событиях: редактирование, удаление, добавление фотографий и видео.

Диаграмма вариантов использования ПС «Афиша концертов»

|  |  |
| --- | --- |
| Посетитель | Администратор |
| Работа с финансами | Управление |
| Фильтрация поиска | Добавление |
| Регистрация | Редактирование |
| Вход с систему | Управление контентом |
|  | Модерация |

Эти функциональные требования помогут обеспечить удобство использования и эффективное взаимодействие пользователей с платформой «работа с личными финансами».

### ***2.3.2 Описание прецедентов***

1. работа с личными финансами:

* Описание: Пользователь может работать с личными финансами, узнавая поступления, расходы и транзакции
* Основной успешный сценарий:
  1. Пользователь открывает приложение «работа с личными финансами».
  2. Переходит на страницу просмотра.
  3. Просматривает список поступлений и расходов.

1. Регистрация пользователя:

* Описание: Пользователь может создать учетную запись для доступа ко всем функциональным возможностям платформы.
* Основной успешный сценарий:
  1. Пользователь открывает приложение «работа с личными финансами».
  2. Нажимает на кнопку «Зарегистрироваться».
  3. Заполняет форму регистрации с указанием электронной почты, номера телефона и других необходимых данных.
  4. Успешно завершает процесс регистрации и может авторизоваться на платформе.

## **2.4** **Нефункциональные требования**

1. Производительность и масштабируемость:

* Система обеспечивает быстрый доступ к информации о концертах даже при увеличении числа пользователей.

1. Надежность, доступность, ремонтопригодность:

* Система стабильная и надежная, минимизирует возможность сбоев и обеспечивает доступность для пользователей большую часть времени.

1. Безопасность**:**

* Данные пользователей зашифрованы и защищены, а процессы аутентификации и авторизации должны быть безопасными.

1. Локализация**:**

* Система предоставляет информацию на языке, принятом в г. Новосибирск.

1. Удобство использования:

* Интерфейс приложения интуитивно понятный, легко навигируемый и приятный для пользователей всех возрастных категорий и уровня навыков владения технологиями.

# **3 ВЫБОР ПРОГРАММНЫХ СРЕД И СРЕДСТВ РАЗРАБОТКИ**

## **3.1 Сравнительный анализ имеющихся возможностей по выбору средств разработки**

В ходе анализа были рассмотрены различные варианты сред разработки, учитывая требования проекта " Работа с личными финансами ". Остановимся на двух популярных и распространенных средах разработки, которые были рассмотрены:

1. Visual Studio с использованием C# и ASP.NET Core MVC:
   * Достоинства:
     + Широкий выбор инструментов разработки ASP.NET Core приложений.
     + Интеграция с различными сервисами и платформами для разработки и деплоя проекта.
     + Обширное сообщество и поддержка, что упрощает поиск решений и помощь.
   * Недостатки:
     + Могут возникнуть сложности с производительностью и оптимизацией при работе с большими объемами данных.
2. SQL Server Management Studio для базы данных:
   * Достоинства:
     + Мощный инструмент для управления базами данных SQL Server.
     + Интуитивно понятный интерфейс для работы с запросами и схемами баз данных.
     + Поддержка различных возможностей и функций SQL.
   * Недостатки**:**
     + Ограничены возможности расширенной работы с другими типами баз данных, что может быть недостатком при необходимости масштабирования проекта.

## **3.2 Характеристика выбранных программных сред и средств**

В ходе анализа были рассмотрены различные варианты сред разработки, учитывая требования проекта " Работа с личными финансами ". Остановимся на двух популярных и распространенных средах разработки, которые были рассмотрены:

1. Visual Studio с использованием C# и ASP.NET Core MVC:
   * Достоинства**:**
     + Широкий выбор инструментов разработки ASP.NET Core приложений.
     + Интеграция с различными сервисами и платформами для разработки и деплоя проекта.
     + Обширное сообщество и поддержка, что упрощает поиск решений и помощь.
   * Недостатки**:**
     + Могут возникнуть сложности с производительностью и оптимизацией при работе с большими объемами данных.
2. SQL Server Management Studio для базы данных:
   * Достоинства:
     + Мощный инструмент для управления базами данных SQL Server.
     + Интуитивно понятный интерфейс для работы с запросами и схемами баз данных.
     + Поддержка различных возможностей и функций SQL.
   * Недостатки:
     + Ограничены возможности расширенной работы с другими типами баз данных, что может быть недостатком при необходимости масштабирования проекта.

Подводя итог, выбор C# и ASP.NET Core MVC в сочетании с SQL Server Management Studio для данного проекта оправдан, учитывая требования к функциональности и удобству разработки. Эти инструменты предоставляют необходимый набор функций для создания веб-приложения "Работа с личными финансами" с базой данных на уровне, отвечающем поставленным задачам и целям проекта.

# **4 АЛГОРИТМ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ**

Этапы реализации проекта «Афиша концертов» с функционалом информации о ближайших концертах в г. Новосибирск и регистрацией пользователя можно представить в виде последовательности шагов:

1. Планирование и анализ требований:

* Определение основных целей проекта, составление требований к функциональности и нефункциональным характеристикам.

1. Дизайн и разработка базы данных:

* Создание структуры базы данных для хранения информации о концертах и пользователях, определение связей между сущностями.

1. Разработка интерфейса:

* Проектирование пользовательского интерфейса (UI) с учетом удобства использования и навигации для пользователей.

1. Реализация функциональности:

* Написание кода для работы с базой данных, отображения информации о концертах, возможности регистрации и аутентификации пользователей.

1. Тестирование и отладка:

* Проведение тестирования приложения на соответствие требованиям, исправление ошибок и устранение недочетов.

1. Внедрение и запуск:

* Развертывание приложения на сервере, настройка рабочих процессов.

1. Поддержка и сопровождение:

* Оказание технической поддержки пользователям, постоянное обновление информации о концертах и услуг платформы, улучшение функциональности.

## **4.2 Пользовательский интерфейс ПС**

### ***4.2.1 Взаимодействие пользователей с ПС***

Алгоритмы взаимодействия пользователей с ПС «Работа с личными финансами»:

1. Регистрация нового пользователя:

* Пользователь открывает приложение.
* Пользователь выбирает опцию «Регистрация».
* Система предоставляет форму для ввода личных данных.
* Пользователь заполняет форму и нажимает кнопку «Зарегистрироваться».
* Система проверяет данные, создает новую учетную запись и перенаправляет пользователя на домашнюю страницу.

1. Просмотр личных финансов:

* Пользователь заходит в раздел «Работа с личными финансами».
* Система отображает список.

Рисунок 1 - Диаграмма последовательности

Рисунок 2 - Диаграмма активности

Рисунок 3 - Диаграмма состояний

### ***4.2.2 Проектирование пользовательских сценариев***

### ***4.2.2.1 Проектирование пользовательских сценариев для проекта «Работа с личными финансами»:***

1. Сценарий регистрации нового пользователя:

* Вход на главный экран приложения.
* Нажатие кнопки «Регистрация».
* Открытие формы для ввода личных данных.
* Заполнение формы (имя, электронная почта, номер телефона).
* Нажатие кнопки «Зарегистрироваться».
* Проверка данных и успешная регистрация.
* Перенаправление на главный экран с доступом к афише концертов.

1. Сценарий просмотра ближайших концертов:

* Переход в раздел «Работа с личными финансами».
* Отображение списка.

### ***4.2.2.2 Структура диалогов:***

1. Главный экран:

* Меню навигации (Главная, финансы, Регистрация).
* Список.
* Кнопка для перехода к форме регистрации.

1. Форма регистрации:

* Поля для ввода личных данных.
* Кнопка «Зарегистрироваться».

1. Страница «финансы»:

* Список транзакций.

Рисунок 4 - Диаграмма состояний интерфейса

### ***4.2.3 Определение операций пользователей***

Прецеденты - это сценарии использования или типичные действия, которые пользователи могут выполнять в системе. Определение операций пользователей поможет в разработке функциональности системы и обеспечит более удобный пользовательский опыт при использовании приложения «Афиша концертов».

Операции пользователей в проекте «Афиша концертов»:

1. Просмотр ближайших концертов:

* Пользователь может просматривать список личных финансов.
* Пользователь может увидеть информацию о дате, времени и описании каждого концерта.

1. Просмотр подробной информации о конкретном концерте:

* Пользователь может выбрать конкретный концерт из списка и просмотреть подробную информацию о нем (описание, время начала и др.).

1. Регистрация пользователя:

* Пользователь может зарегистрироваться в приложении, указав свое имя, электронную почту и номер телефона.

### ***4.2.4 Составление функциональных блоков***

Составление функциональных блоков проекта «Афиша концертов»:

1. Блок «Управление пользователями»:

* Регистрация новых пользователей.
* Авторизация пользователей.
* Управление профилями пользователей.

1. Блок «Концерты»:

* Отображение списка ближайших концертов.
* Поиск концертов по различным критериям (дата, место, жанр и т.д.).
* Просмотр подробной информации о конкретном концерте.
* Добавление концертов в избранное или планировщик.

1. Блок «Административная панель»:

* Управление контентом транзакций (добавление, изменение, удаление).
* Управление пользователями.

### ***4.2.5 Проектирование структуры экранов ПС (ПМ) и схемы навигации***

Структура экранов:

1. Главный экран:

* Название приложения «ФИНАНСЫ».
* Краткое описание сервиса.
* Кнопка «Главная».
* Кнопка «Войти/Зарегистрироваться» для пользователей.

1. Экран Регистрации:

* Форма для ввода данных пользователя: имя, электронная почта, номер телефона.
* Кнопка «Зарегистрироваться» для завершения процесса.

1. Экран Транзакций:

* Список.
* Фильтрация

Рисунок 5 - Схема навигации

Такая структура экранов и схема навигации позволяет пользователям легко ориентироваться в приложении «Афиша концертов» в г. Новосибирск, быстро находить информацию о концертах и регистрироваться.

### ***4.2.6 Низкоуровневое проектирование***

1. Экран «Главная страница»:

* Верхняя панель с логотипом и меню навигации.
* Краткое описание сервиса.
* Кнопка для перехода к страницам «Транзакции» и «Войти/Зарегистрироваться».

Экран «Главная страница» изображен на рисунке 6.

Рисунок 6 – Скриншот экрана «Главная страница»

1. Экран «Транзакции»:

* Список Концертов.
* Кнопка для открытия подробной информации.
* Фильтрация по жанру и дате.

Экран «транзакции» изображен на рисунках 7 и 8.

Рисунок 7 – Скриншот 1 экрана «»

Рисунок 8 – Скриншот 2 экрана «»

1. Экран «Войти/Зарегистрироваться»:

* Форма для заполнения данных: имя, электронная почта, номер телефона.
* Кнопка «Зарегистрироваться» для отправки данных.

Экран «Войти/Зарегистрироваться» изображен на рисунке 10.

Рисунок 10 – Скриншот экрана «Войти/Зарегистрироваться»

Эти разработанные экранные формы и прототипы окон позволили создать удобный пользовательский интерфейс для проекта "Афиша концертов", обеспечивая удобство и понятность взаимодействия пользователя с приложением.

## **4.3 Входные, выходные и промежуточные данные**

Входные данные:

1. Информация о концертах:

* Файл в формате Controller.cs с данными . На рисунке 11 изображен пример структуры файла:

Рисунок 11 –

1. Данные пользователя для регистрации:

* Форма на веб-странице с полями для ввода имени, электронной почты и номера телефона.

Промежуточные данные:

1. База данных транзакций:

* Таблица в базе данных, содержащая информацию о транзакциях.

CREATE TABLE [dbo].[Transacions] (

[id] INT NOT NULL,

[Name] VARCHAR (255) NULL,

[Description] VARCHAR (255) NULL,

[sum] INT NULL,

[CategoryID] VARCHAR (255) NULL,

);

Выходные данные:

1. Список ближайших концертов, отображаемый на веб-странице:

<h2>все финансы</h2>

<h3>@Model.trnsCategory</h3>

@{

foreach (var transaction in Model.getAllTransactions)

{

<div>

<h2>Название: @transaction.name</h2>

<p>Cумма: @transaction.sum.ToString("c")</p>

</div>

}

}

## **4.4 Разработка базы данных, реализуемой в рамках ПС**

Для проекта " Работа с личными финансами ", который содержит информацию о ближайших транзакциях и регистрацию пользователя, мы можем разработать следующую структуру базы данных:

Таблица Transactions:

* Поля:
* public int id { get; set; }
* public string name { get; set; }
* public string Desc { get; set; }
* public ushort sum { get; set; }
* public int categoryID { get; set; }
* public virtual Category Category { get; set; }
* Таблица Users:
* Поля:
* public int Id { get; set; }
* public string Login { get; set; }
* public string? Email { get; set; }
* public string Password { get; set; }
* // Role
* public int RoleId { get; set; }
* public Role Role { get; set; }
* // UserProfile
* public UserProfile Profile { get; set; }

На рисунке 12 изображена ER-диаграмма базы данных.

Рисунок 12 – ER-диаграмма

На диаграмме показаны три таблицы – Concerts, Users и Location, их поля и связь между ними через поле Concerts\_id и Location\_id.

Таким образом, в проекте мы имеем базу данных для хранения информации о концертах и пользователей, а также связь между концертами и подписанными на них пользователями и связь между концертами и локациями.

## **4.6 Алгоритмы использования применяемых программных** технологий

Алгоритмы использования применяемых программных технологий для проекта "Работа с личными финансами" на C# ASP.NET Core MVC с базой данных SQL Server:

1. Архитектура проекта:

* Использование паттерна MVC (Model-View-Controller) для разделения бизнес-логики, пользовательского интерфейса и управления запросами.
* Определение моделей данных для концертов и пользователей.

1. Интерфейс и веб-страницы:

* Создание представлений (Views) для отображения информации о концертах и форм регистрации пользователя.
* Использование HTML, CSS, и Razor для динамической генерации контента на страницах.

1. Контроллеры и маршрутизация:

* Создание контроллеров для обработки HTTP-запросов, возвращающих данные о концертах и обрабатывающих запросы на регистрацию пользователей.

1. Работа с базой данных:

* Использование Entity Framework Core для взаимодействия с SQL Server базой данных.

1. Авторизация и аутентификация:

* Реализация системы аутентификации пользователей с использованием ASP.NET Identity.

1. Обработка данных о концертах:

* Фильтрация и сортировка концертов по дате и жарнам.

Этот алгоритм позволит разработать полноценное веб-приложение «Работа с личными финансами» на C# ASP.NET Core MVC с интеграцией базы данных SQL Server, обеспечивающее удобное отображение информации о концертах и регистрацию пользователей.

## **4.7 Архитектура и схема функционирования ПС (ПМ)**

Для проекта "Афиша концертов", написанного на C# ASP.NET Core MVC с базой данных, разработанной в SQL Management Studio, были использованы следующие элементы структуры программной системы и их взаимосвязи:

1. Блоки функций:

* Контроллеры (Controllers):
* AccountController
* TransactionController
* Модели (Models):
  + - Category.
    - User.
    - AppDBContent
    - Role
    - Transactions
    - UserProfile
    - ErrorVievModels
* Представления (Views):
  + - Transactions/Iist.cshtml: представление для отображения списка транзакций.
    - Home/Index.cshtml: представление для отображения главной страницы.
    - Account/Index.cshtml: представление для отображения зарегистрированных пользователей.
    - Account/Register.cshtml: представление для регистрации нового пользователя.
    - Account/ptofile.cshtml

1. **Диаграмма компонентов UML:**
   * Данная диаграмма показывает различные компоненты программной системы (контроллеры, модели, представления) и их зависимости друг от друга.
2. **Диаграмма развертывания UML:**
   * Диаграмма показывает, как программная система будет развернута на аппаратном оборудовании, включая сервер базы данных SQL Management Studio и сервер приложений ASP.NET Core MVC.

Таким образом, структура программной системы "работа с личными финансами" на C# ASP.NET Core MVC с базой данных SQL Management Studio может быть наглядно представлена с использованием упомянутых диаграмм UML для обеспечения лучшего понимания разработки и взаимосвязей между различными элементами проекта.

# **5 ТЕСТИРОВАНИЕ И ОПТИМИЗАЦИЯ**

## **5.1 План тестирования**

1. Функциональное тестирование:

* Регистрация пользователя**:**
  + Проверка возможности создания нового аккаунта.
  + Проверка правильности валидации введенных данных.
* Отображение ближайших концертов**:**
  + Проверка корректного отображения информации.
  + Проверка фильтрации.

2. Интеграционное тестирование:

* Проверка взаимодействия между компонентами системы (регистрация пользователя, отображение финансов).

Обоснование выбора:

Функциональное тестирование и интеграционное тестирование будут наиболее предпочтительными для данного проекта, так как позволят проверить основной функционал системы - регистрацию пользователей и отображение информации финансах. Эти тесты позволят убедиться в правильной работоспособности системы как целого.

Обоснование отсутствия некоторых видов тестирования:

* Нагрузочное тестирование: Данная система не предполагает больших нагрузок на сервер, поэтому этот вид тестирования не является критически важным.

# **6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

1. Введение:

* Программное средство " Работа с личными финансами " предназначено для отображения информации о личных финансах. Пользователи также могут зарегистрироваться.

1. Установка и настройка:

* Для начала работы с программным средством " Работа с личными финансами " необходимо перейти на сайт и создать аккаунт.
* После успешной регистрации вы будете автоматически перенаправлены на главную страницу, где отображается список ваших финансов.

1. Основные действия пользователя:
   1. Просмотр информации о финансах:

* На странице «финансы» можно увидеть список.
  1. Регистрация пользователя:
* Для регистрации пользователя нажмите на кнопку "Войти/Зарегистрироваться" в шапке профиля.
* Заполните все необходимые поля: имя, адрес электронной почты, номер телефона.
* После успешной регистрации страница обновится и откроется главное окно.

1. Образы экранов:

* Главная страница, где находится важная информация. Изображена на рисунке 6.
* Страница всех концертов. Изображена на рисунках 7 и 8.
* Страница регистрации. Изображена на рисунке 13.
* Подробная информация о концерте. Изображена на рисунке 9.

# **ЗАКЛЮЧЕНИЕ**

Проект "Работа с личными финансами" представляет собой информационную платформу, которая позволяет пользователям получать информацию о транзакциях. В рамках проекта также предусмотрена возможность регистрации пользователей для получения дополнительных функций.

Итак, подводя итоги проделанной работы, можно отметить следующее:

* Разработано и реализовано программное средство "Афиша концертов", которое успешно отображает информацию о транзакциях и обеспечивает возможность регистрации пользователей.
* Проект предлагает новый подход к удобному отслеживанию личных доходов и расходов.

Объем созданного программного средства составляет: 2 контроллеров, 5 моделей и 5 представлений.

Дальнейшее продвижение проекта и его улучшение могут привести к расширению функций и повышению удовлетворенности пользователей.

Таким образом, проект "Работа с личными финансами" имеет практическую значимость для любителей отслеживать свои личные транзакции.

**ЛИТЕРАТУРА**

Документация по ASP NET [Документация по ASP.NET | Microsoft Learn](https://learn.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-8.0)