

**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

«Приднестровский государственный университет им. Т.Г. Шевченко»

ЭКОНОМИЧЕСКИЙ ФАКУЛЬТЕТ

**кафедра Бизнес-информатики и информационных технологий**

Курсовая работа

по курсу «Web программирование (back)»

НА ТЕМУ: Проектирование и разработка backend для системы поддержки учебной деятельности преподавателей

Выполнила:

Студентка группы ЭФ23ВР68УР1

Направление «Менеджмент»

Профиль «Управление и разработка информационных систем в экономике»

Бурлачук Александра Валерьевна

Проверил:

к.ф.-м.н., доцент

Надькин Леонид Юрьевич

Тирасполь – 2024

**Глава 1. Введение**

1.1. Актуальность темы  
1.2. Цель и задачи исследования  
1.3. Объект и предмет исследования  
1.4. Методы исследования

**Глава 2. Анализ предметной области и существующих решений**

2.1. Анализ существующих систем управления учебными процессами  
2.2. Проблемы и ограничения текущих решений  
2.3. Требования к системе

**Глава 3. Проектирование системы**

3.1. Архитектура системы  
3.2. Диаграммы UML (Use Case, Sequence, Class)  
3.3. Проектирование базы данных (ER-диаграмма)  
3.4. Описание основных сущностей

**Глава 4. Разработка серверной части системы**

4.1. Выбор технологий и инструментов разработки  
4.2. Реализация функционала управления учебными материалами  
4.3. Реализация функционала для тестирования  
4.4. Работа с учетными записями пользователей  
4.5. Интеграция с клиентской частью

**Глава 5. Тестирование и отладка системы**

5.1. Методология тестирования  
5.2. Результаты тестирования  
5.3. Обнаруженные ошибки и их исправление

**Глава 6. Экономическая и социальная эффективность системы**

6.1. Расчет затрат на разработку  
6.2. Оценка эффективности системы для пользователей

**Глава 7. Заключение**

7.1. Выводы по результатам работы  
7.2. Перспективы дальнейшего развития системы

**Приложения**

* Полный код реализованной системы
* Диаграммы и схемы
* Техническая документация

ВВЕДЕНИЕ

Современные образовательные процессы требуют внедрения цифровых решений для повышения эффективности учебной деятельности. Преподаватели сталкиваются с рядом задач, таких как управление учебными материалами, проведение тестирований, контроль успеваемости студентов, которые сложно реализовать без использования современных информационных систем.

Разработка системы, которая предоставляет инструменты для автоматизации этих задач, становится неотъемлемой частью образовательной среды. Такая система позволяет упростить организацию учебного процесса, минимизировать рутинные операции, повысить качество предоставляемых материалов и обеспечить удобный доступ к данным для преподавателей и студентов.

**Проблемы, которые решает данная система**

|  |  |
| --- | --- |
| **Упрощение работы преподавателей** | Создание, редактирование и удаление учебных материалов. |
| Формирование тестов и автоматизация проверки результатов. |
| Учет успеваемости учащихся. |
| **Эффективная организация учебного процесса:** | Централизованное хранение учебных материалов |
| Упрощение взаимодействия между преподавателями и учащимися через цифровую платформу. |
| **Сокращение временных затрат:** | Быстрая загрузка и управление ресурсами. |
| Автоматическая проверка тестов и анализ результатов. |
| **Обеспечение безопасности данных:** | Защита учебных материалов и учетных записей пользователей. |
| Контроль доступа к ресурсам |

Цель курсовой работы является разработка информационной системы, обеспечивающей поддержку учебной деятельности преподавателей, с акцентом на управление учебными материалами, организацию тестирования и эффективное взаимодействие с учетными записями пользователей.

Задачи, которые необходимо решить для достижения цели:

1. Провести исследование существующих решений для поддержки учебной деятельности, выявить основные потребности преподавателей и определить функциональные требования к системе.

2. Разработать структурированную архитектуру backend-системы, включая базу данных и основные модули, обеспечивающие хранение данных, взаимодействие с клиентом и выполнение бизнес-логики.

3. Реализовать управление учебными материалами (создание, редактирование, удаление и просмотр).

4. Организовать функционал для автоматизированного тестирования (создание тестов, проверка ответов, хранение результатов).

5. Обеспечить работу с учетными записями пользователей, включая регистрацию, аутентификацию и авторизацию.

6. Применить ASP.NET Core для разработки backend-части системы, Entity Framework для работы с базой данных и AutoMapper для упрощения преобразования данных между уровнями.

7. Проверить работоспособность системы на разных этапах разработки, исправить возможные ошибки и оптимизировать производительность.

6. Подготовить техническую документацию, включая описание архитектуры, структуры базы данных, интерфейсов API и примеров использования системы.

Объектом исследования является процесс организации и проведения учебной деятельности преподавателей, включающий управление учебными материалами, проведение тестирования и работу с учетными записями пользователей.

Предметом исследования выступают методы и технологии проектирования и разработки серверной части информационной системы для поддержки учебной деятельности преподавателей.

Методы исследования, используемые в данной работе, включают анализ литературы и существующих решений в области автоматизации учебных процессов, проектирование системы с применением современных методологий, таких как диаграммы UML и ER-диаграммы, разработку программного обеспечения с использованием объектно-ориентированного подхода, а также проведение тестирования созданного программного продукта для обеспечения его функциональности и надежности.

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИХ РЕШЕНИЙ

**1.1. Анализ существующих систем управления учебными процессами для индивидуальных преподавателей**

Современные системы управления учебными процессами (Learning Management Systems, LMS) в основном создаются для образовательных учреждений, где работает множество преподавателей и обучаются большие группы студентов. Однако для индивидуальных преподавателей, ведущих несколько предметов с учениками разных возрастов, такие решения зачастую избыточны и неудобны. Это создает потребность в разработке специализированных инструментов.

Особенности и возможности существующих систем:

Основные недостатки существующих решений заключаются в их избыточности, сложности настройки и высокой стоимости. Большинство популярных LMS не предлагают возможности персонализации под конкретного преподавателя, что делает их неудобными для работы с небольшими группами учеников. Вдобавок техническая поддержка и настройка таких систем требуют значительных ресурсов, которыми индивидуальный учитель может не обладать. Это создает потребность в специализированном инструменте, который бы учитывал особенности работы одного преподавателя, вел занятия по нескольким дисциплинам и предоставлял удобные инструменты для управления процессом обучения.

Предлагаемая разработка сайта нацелена на решение этих проблем. Этот инструмент позволит преподавателю централизованно управлять всеми аспектами своей работы: хранением материалов, подготовкой тестов и проверкой результатов. Основное внимание уделяется простоте использования, что исключает необходимость в технической поддержке. Разработка будет учитывать индивидуальные потребности преподавателя, включая работу с разными возрастными группами учеников. Такая система не только упрощает организацию учебного процесса, но и повышает его эффективность за счет удобного доступа к материалам и возможностям тестирования.

Важно отметить, что разработка сайта для индивидуального преподавателя позволяет создать экономически эффективное решение, адаптированное под конкретные задачи. В отличие от крупных платформ, этот инструмент будет учитывать специфику преподавания одного учителя, обеспечивая высокую степень персонализации. Такой подход способствует улучшению качества обучения, облегчает взаимодействие с учениками и помогает преподавателю сосредоточиться на образовательной деятельности, а не на технических аспектах управления процессом.

**1.2. Проблемы и ограничения текущих решений**

Существующие решения для управления учебными процессами, такие как коммерческие Learning Management Systems (LMS), изначально разрабатывались для использования в крупных образовательных учреждениях, где предполагается участие большого числа преподавателей и студентов. Однако такие системы часто оказываются избыточными и неудобными для индивидуальных преподавателей, работающих с ограниченным количеством учеников и ведущих занятия по нескольким дисциплинам. Эти решения обладают рядом проблем и ограничений, которые делают их малоэффективными в контексте индивидуального преподавания.

Во-первых, значительная часть LMS характеризуется избыточностью функционала. Они предлагают широкий спектр возможностей, таких как управление кампусами, распределение курсов между несколькими преподавателями, статистика по группам студентов и другие функции, которые не востребованы индивидуальными преподавателями. Это создает ненужную сложность в использовании таких систем и может отвлекать от основных задач.

Во-вторых, настройка подобных платформ требует значительного времени и ресурсов. Для полного развертывания системы может потребоваться участие технических специалистов и глубокое понимание процессов администрирования. Для индивидуальных преподавателей, не обладающих необходимой технической подготовкой или ресурсами для привлечения сторонней помощи, такие системы становятся практически недоступными.

Еще одной важной проблемой является высокая стоимость использования LMS. Большинство популярных систем имеют лицензионные ограничения или подписки, которые требуют значительных финансовых затрат, включая техническую поддержку и обновления. Для преподавателей, работающих на индивидуальной основе, это становится серьезным барьером.

Ограниченная возможность персонализации также является значительным недостатком текущих решений. Такие системы, как правило, предоставляют стандартные функции, которые сложно адаптировать под уникальные потребности одного преподавателя. Это усложняет их использование для управления учебным процессом в небольших группах или для работы с учениками различных возрастных категорий.

Дополнительным ограничением является недостаточная гибкость в работе с разными возрастными группами. Структура многих LMS не предусматривает удобных механизмов для разделения учеников по возрасту, уровню знаний или другим критериям, что снижает эффективность преподавания и затрудняет индивидуальный подход.

Все эти ограничения подчеркивают необходимость разработки специализированной системы управления учебными процессами, ориентированной на индивидуальных преподавателей. Такая система должна быть простой и интуитивно понятной, обеспечивать доступ к функционалу без необходимости глубоких технических знаний, быть экономически доступной и учитывать специфические потребности преподавателей, работающих с небольшими группами учеников. Разработка такого инструмента позволит не только упростить организацию учебного процесса, но и повысить его эффективность за счет удобного доступа к материалам и возможности адаптации под конкретные задачи преподавателя.

**1.3. Требование к системе**

На основе выявленных проблем существующих систем управления учебными процессами можно выделить основные требования к разрабатываемой системе, которая будет предназначена для индивидуальных преподавателей, работающих с учениками разных возрастов по нескольким дисциплинам. Эти требования направлены на создание эффективного, удобного и доступного инструмента для организации учебной деятельности.

Прежде всего, система должна быть простой и удобной в использовании. Её интерфейс должен быть интуитивно понятным, а функционал — доступным для освоения без необходимости обладания специальными техническими знаниями. Это позволит преподавателю сосредоточиться на учебном процессе, а не на изучении сложных инструкций или необходимости привлекать специалистов для настройки системы.

Централизация управления является следующим ключевым требованием. Система должна объединять все аспекты работы преподавателя в одном месте, включая управление учениками, предметами, учебными материалами и тестированием. Это упростит организацию учебного процесса и позволит преподавателю эффективно планировать и проводить занятия.

Персонализация системы также играет важную роль. Преподаватель должен иметь возможность настроить интерфейс и функционал системы в соответствии со своими задачами и потребностями. Это включает изменение структуры учебных курсов, редактирование доступных функций и настройку внешнего вида.

Система должна поддерживать работу с несколькими дисциплинами, предоставляя инструменты для хранения и управления учебными материалами по каждому предмету. Это особенно важно для индивидуальных преподавателей, которые ведут занятия по различным направлениям.

Для удобства работы с учениками разных возрастов система должна предусматривать возможность разделения учеников на группы с учётом их возраста, уровня знаний и других параметров. Это обеспечит индивидуальный подход к каждому ученику и позволит преподавателю эффективно организовывать учебный процесс.

Функционал тестирования является неотъемлемой частью системы. Преподаватель должен иметь возможность создавать тесты, проводить их и оценивать результаты. Это позволит автоматизировать проверку знаний учеников и отслеживать их прогресс.

Экономическая доступность системы — ещё одно важное требование. Система должна быть разработана таким образом, чтобы её использование не требовало значительных финансовых затрат. Это включает как начальную стоимость, так и расходы на обслуживание.

Наконец, система должна обеспечивать высокий уровень безопасности данных. Это касается как конфиденциальности личной информации учеников и преподавателя, так и сохранности учебных материалов. Неавторизованный доступ к системе должен быть полностью исключён.

Таким образом, разрабатываемая система будет отвечать всем необходимым требованиям индивидуальных преподавателей, обеспечивая удобство, персонализацию и эффективность управления учебными процессами. Она станет универсальным инструментом, который облегчит организацию занятий, улучшит взаимодействие с учениками и повысит качество образовательного процесса.

**Глава 2. Архитектура проекта**

**2.1. Общее описание архитектуры**

Проект реализован с использованием многослойной архитектуры, которая разделяет ответственность на четыре основные части

Многослойная архитектура системы обеспечивает четкое разделение обязанностей, минимизируя зависимость между слоями и упрощая тестирование и поддержку. Каждый слой выполняет свою роль, взаимодействуя только с соседними слоями. Это позволяет повысить гибкость системы и облегчает процесс внесения изменений.

Core Layer представляет собой фундамент всей системы. Он содержит основные модели данных, такие как Student, Material, Subject, TestResult, которые описывают сущности системы, а также общую базовую модель BaseModel, предоставляющую стандартные атрибуты, такие как идентификатор (Id) и временные метки (CreatedAt, UpdatedAt). Кроме того, здесь находятся перечисления (Enums), определяющие типы данных, например, MaterialType и LevelType. Core Layer предоставляет универсальные стандарты и модели для других слоев, используется в Infrastructure для работы с базой данных, в Service для обработки бизнес-логики и в API для передачи данных клиенту. Этот слой является независимым и не зависит от других, что делает его универсальным.

Infrastructure Layer отвечает за взаимодействие с базой данных. Он включает в себя провайдеры, реализующие CRUD-операции для моделей данных, и контекст базы данных ApplicationContext, реализованный с использованием Entity Framework Core, который управляет подключением к базе данных. Infrastructure Layer использует модели из Core для определения схемы базы данных, предоставляет интерфейсы и реализацию для Service Layer, изолируя его от прямой работы с базой данных, и обеспечивает надежный и унифицированный доступ к данным. Например, Service Layer отправляет запрос на получение всех студентов в Infrastructure Layer, который выполняет запрос к базе данных и возвращает данные.

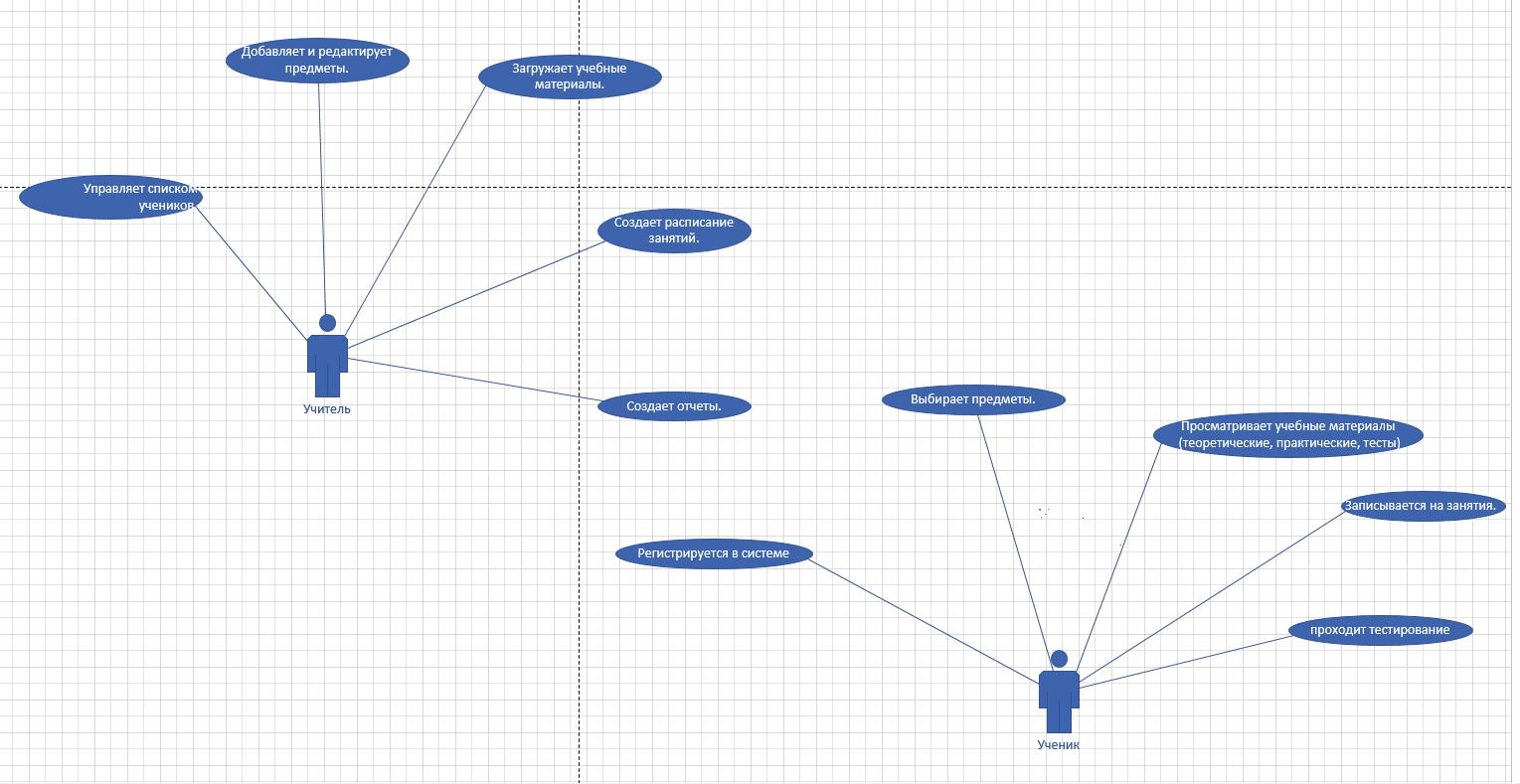
Service Layer реализует бизнес-логику системы. Этот слой отвечает за обработку данных, полученных из Infrastructure, перед их отправкой в API, взаимодействует с провайдерами из Infrastructure для выполнения операций с базой данных и инкапсулирует бизнес-логику для поддержки ключевых сценариев, таких как управление учебными материалами, тестирование и работа с учетными записями. Service Layer использует интерфейсы, предоставленные Infrastructure, для выполнения операций с данными, передает подготовленные данные в API для предоставления их клиентам и поддерживает формат данных, совместимый с Core. Например, когда API Layer запрашивает список материалов, Service Layer обрабатывает запрос, взаимодействует с провайдером Infrastructure Layer, фильтрует или преобразует данные и отправляет их в API.

API Layer отвечает за взаимодействие с клиентом. Этот слой обрабатывает HTTP-запросы, отправленные клиентами, вызывает соответствующие методы из Service Layer и формирует HTTP-ответы в формате JSON для клиента. API Layer получает запросы от клиента, например, на добавление нового учебного материала, передает их в соответствующий контроллер, который вызывает методы Service Layer для обработки материала. Service Layer, в свою очередь, взаимодействует с Infrastructure Layer для сохранения данных в базе и возвращает результат обратно по цепочке в API, который отправляет ответ клиенту.

Общее взаимодействие между слоями выглядит следующим образом: клиент отправляет HTTP-запрос в API Layer, который передает запрос в Service Layer для выполнения бизнес-логики. Service Layer взаимодействует с Infrastructure Layer для получения или изменения данных в базе. Infrastructure Layer использует модели Core для выполнения операций с базой данных. Результат передается обратно по цепочке: из Infrastructure в Service, затем в API, а оттуда клиенту. Такой подход обеспечивает четкое разделение ответственности между слоями, легкость тестирования и изменения каждого слоя независимо от других, а также масштабируемость системы за счет минимизации связей между слоями.

**2.2. Диаграмма вариантов использования (Use Case Diagram)**

Диаграмма вариантов использования (Use Case Diagram) отображает основные действия, которые выполняются в системе, а также взаимодействие между пользователями (акторами) и системой.



На диаграмме вариантов использования показаны основные взаимодействия пользователей системы: Учителя и Ученика.

Учитель обладает широким спектром возможностей, таких как добавление и редактирование предметов, загрузка учебных материалов, создание расписания занятий, управление списком учеников и создание отчетов. Эти функции позволяют ему эффективно организовывать учебный процесс, предоставляя ученикам доступ к актуальной информации и заданиям.

С другой стороны, Ученик взаимодействует с системой через следующие функции: регистрация в системе, выбор интересующих предметов, просмотр учебных материалов, запись на занятия и прохождение тестирования. Таким образом, система обеспечивает ученикам доступ к необходимым ресурсам для обучения, а также позволяет отслеживать их успехи и прогресс.

Диаграмма иллюстрирует взаимосвязь между Учителем и Учеником через систему. Учитель выступает источником данных, материалов и задач, в то время как Ученик потребляет эти данные и выполняет учебные задания. Такое разделение обязанностей способствует упрощению взаимодействия с системой и повышает эффективность учебного процесса.

**2.3. Проектирование базы данных (ER-диаграмма)**

Проектирование базы данных является важным этапом разработки системы, так как оно определяет структуру хранения данных, их связи и способы обработки. В данной системе база данных должна обеспечивать удобное управление учебными материалами, тестированием и взаимодействием учеников с преподавателем. Исходя из анализа предметной области и поставленных требований, была спроектирована сущностно-связная (ER) модель базы данных.

Для хранения учебных материалов и тестовых результатов в системе предусмотрены отдельные таблицы, которые позволяют организовать удобную работу с данными. Так как система ориентирована на индивидуального преподавателя, ведущего несколько дисциплин, в базе данных выделены основные сущности: **Student (ученик), Subject (предмет), Material (учебные материалы), TestResult (результаты тестирования), Level (уровень подготовки)** и **StudentSubject (связь между учениками и предметами)**.

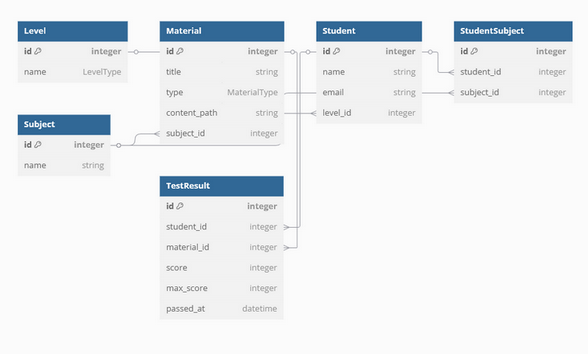
Основу структуры данных составляет таблица **Student**, в которой хранятся сведения об учениках, включая их имя, электронную почту и уровень подготовки. Связь между учениками и предметами осуществляется через промежуточную таблицу **StudentSubject**, что позволяет одному ученику быть записанным на несколько предметов и учитывать связь "многие ко многим".

Таблица **Material** хранит учебные материалы, относящиеся к конкретным предметам, и включает такие атрибуты, как название, тип материала (теоретический, практический, тест) и путь к файлу. Эта таблица связана с таблицей **Subject**, что позволяет преподавателю загружать материалы для различных дисциплин.

Результаты тестирования фиксируются в таблице **TestResult**, которая хранит информацию о прохождении тестов учениками. В данной таблице указываются идентификаторы ученика и материала, оценка за тест, максимальный балл и дата прохождения. Эта структура позволяет отслеживать прогресс каждого ученика.

Для разделения учеников по уровням подготовки введена таблица **Level**, связанная с таблицей **Student**. В ней используется перечисление (**Enum**) для обозначения уровня подготовки (например, 5-6 класс, 7-9 класс и 10-11 класс), что позволяет учитывать различия в учебном процессе.

На рисунке ниже представлена ER-диаграмма базы данных, демонстрирующая основные сущности системы и связи между ними.

****

Спроектированная структура обеспечивает гибкость и удобство в управлении учебным процессом. Использование связи "многие ко многим" между учениками и предметами позволяет эффективно распределять материалы, а таблица тестирования предоставляет возможность хранения истории прохождения тестов. В будущем структура базы данных может быть расширена за счет добавления новых таблиц, например, для хранения информации о домашних заданиях или дополнительной статистики по успеваемости учеников.

Таким образом, разработанная база данных соответствует требованиям системы и позволяет эффективно управлять учебным процессом, обеспечивая целостность данных и их надежное хранение.