СОДЕРЖАНИЕ

[Введение 3](#_Toc196374612)

[1. Общая часть 5](#_Toc196374613)

[1.1. Цель разработки 5](#_Toc196374614)

[1.2. Средства разработки 5](#_Toc196374615)

[2. Специальная часть 8](#_Toc196374616)

[2.1. Постановка задачи 8](#_Toc196374617)

[2.1.1. Входные данные 11](#_Toc196374618)

[2.1.2. Выходные данные 11](#_Toc196374619)

[2.1.3. Подробные требования к проекту 12](#_Toc196374620)

[2.2. Внешняя спецификация 12](#_Toc196374621)

[2.2.1. Описание задачи 12](#_Toc196374622)

[2.2.2. Входные и выходные данные 13](#_Toc196374623)

[2.2.3. Методы 16](#_Toc196374624)

[2.2.4. Тесты 18](#_Toc196374625)

[2.3.Проектирование 20](#_Toc196374626)

[2.3.1. Схема архитектуры приложения 20](#_Toc196374627)

[2.3.2. Логическая схема данных 21](#_Toc196374628)

[2.3.3. Физическая схема данных 22](#_Toc196374629)

[2.3.4. Диаграмма классов 27](#_Toc196374630)

[2.3.4. Функциональная схема 28](#_Toc196374631)

[2.4. Результат работы программы 28](#_Toc196374632)

[3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 31](#_Toc196374633)

[3.1. Инструментальная среда разработки 31](#_Toc196374634)

[Заключение 33](#_Toc196374635)

[Список используемых материалов 34](#_Toc196374636)

ПРИЛОЖЕНИЕ А. Техническое задание

ПРИЛОЖЕНИЕ Б. Текст программы

ПРИЛОЖЕНИЕ В. Программа испытаний

ПРИЛОЖЕНИЕ Г. Руководство пользователя

ПРИЛОЖЕНИЕ Д. Скрипт базы данных

# Введение

Традиционная система очного образования, безусловно, обладает своими преимуществами, которые делают ее привлекательной для многих студентов и преподавателей. Во-первых, личное взаимодействие между студентами и преподавателями способствует более глубокому пониманию материала и позволяет задавать вопросы в реальном времени. Во-вторых, обучение в классе помогает развивать социальные связи, что способствует формированию сообщества и улучшает атмосферу обучения. Кроме того, очное образование предлагает структурированное обучение с четким графиком, что помогает студентам организовать свое время и сосредоточиться на учебе.

Однако традиционная система сталкивается с рядом ограничений, которые могут затруднить процесс обучения. Необходимость физического присутствия в учебных заведениях может стать преградой для многих обучающихся, особенно для тех, кто совмещает учебу с работой или семейными обязанностями. Фиксированный график занятий не всегда соответствует индивидуальным потребностям студентов, что может негативно сказаться на их мотивации и успеваемости.

Система дистанционного образования предоставляет все больше возможностей для получения знаний и их закрепления. Этот формат обучения позволяет студентам учиться в удобном для них темпе, выбирать актуальные и интересные темы, что способствует более глубокому усвоению материала. Онлайн-образование также открывает новые горизонты для преподавателей, позволяя им поддерживать связь со студентами на расстоянии, предоставлять лекционные материалы и проводить оценку знаний.

Переход от очного к дистанционному обучению решает ряд проблем, связанных с традиционной системой. Во-первых, оценка знаний становится более оперативной, что сокращает время, необходимое для проверки тестов и выполнения заданий. Во-вторых, дистанционный формат позволяет обучаться людям из различных регионов и даже стран, что расширяет доступ к образовательным ресурсам.

Кроме того, применение дистанционной системы обучения выходит за рамки традиционного образования. Она может быть использована для повышения квалификации сотрудников, что особенно актуально в условиях постоянных изменений на рынке труда. Также дистанционное обучение может служить инструментом для приема на работу, позволяя работодателям проводить тестирование и оценку кандидатов удаленно.

# 1. ОБЩАЯ ЧАСТЬ

## 1.1. Цель разработки

Целью разработки веб-приложения является предоставление возможности удобной организации взаимодействия студентов и преподавателей в процессе обучения.

## 1.2. Средства разработки

Для разработки программного обеспечения, а также реализации базы данных и создания сервиса API были использованы технические средства, представленные в Таблице 1.

Таблица 1 – Технические средства

| № | Тип оборудования | Наименование оборудования |
| --- | --- | --- |
| 1 | 2 | 3 |
| Asus Zenbook 14 UM433IQ | | |
| 1 | Размер экрана | 14 |
| 2 | Разрешение экрана | 1920x1080 |
| 3 | Линейка процессора | AMD Ryzen 7 4700U with Radeon Graphics |
| 4 | Количество ядер процессора | 8 |
| 5 | Оперативная память | 16 Гб |
| 6 | Видеокарта | NVIDIA GeForce MX350 |
| 7 | Конфигурация накопителей | SSD |
| 8 | Общий объем всех накопителей | 1000 |

Для разработки программного обеспечения были использованы программные средства, представленные в Таблице 2.

Таблица 2 – Программные средства

| № | Тип средства | Название средства | Назначение |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 1 | Операционная система | Microsoft Windows 11 | Организация взаимодействия программ и пользователя |
| 2 | Инструментальные средства разработки программного обеспечения | Microsoft Visual Studio 2022 17.9 | Разработка API |
| 3 | Инструментальные средства разработки программного обеспечения | Microsoft Visual Studio Code 1.99 | Разработка веб-приложения |
| 4 | СУБД | SQL Server Management Studio 19 | Разработка базы данных |
| 5 | Браузер | Google Chrome | Тестирование API, запуск веб-приложения |

# 2. Специальная часть

## 2.1. Постановка задачи

Разработать веб-приложение, в котором преподаватель сможет управлять учебным процессом, а студент выполнять задания преподавателя.

### 2.1.1. Входные данные

Входными данными является информация о лекциях (название, описание, файл лекции), практических работах (название, описание, файл работы), тестах (название, описание, время выполнения, вопросы и ответы к вопросам, балл за вопрос, критерии оценивания), выполненных работах (оценка, файл выполненной работы), данные пользователя.

### 2.1.2. Выходные данные

Выходными данными является информация о лекциях, практических работах, тестах, выполненных работах.

### 2.1.3. Требования к проекту

Спроектировать и разработать веб-приложение «YumlSchool».

Назначением веб-приложения является упрощение ведения учебного процесса.

Интерфейс приложения должен быть простым и понятным пользователю, а также иметь возможность подстраиваться под разные разрешения экрана устройства.

Пользователю необходимо авторизоваться под своим аккаунтом, используя логин и пароль. Далее, пользователю предоставляется функционал в зависимости от его роли.

Под ролью «Администратор» пользователю доступны функции:

* Редактирование списка групп с возможностью добавления, изменения, удаления группы;
* Редактирование списка предметов с возможностью добавления, изменения, удаления предмета;
* Редактирование списка пользователей с возможностью изменения роли и группы пользователя.

Под ролью «Преподаватель» пользователю доступны функции:

* Манипулирование данными лекций, практических работ, тестов;
* Оценивание выполненных работ студентов;
* Выгрузка файла в формате PDF о текущей успеваемости студентов по практической работе;
* Просмотр оценки и количества баллов студентов за прохождение тестирования.

Под ролью «Студент» пользователю доступны функции:

* Просмотр лекций;
* Добавление выполненной работы и просмотр результата оценивания работы;
* Прохождение тестирования и просмотр оценки и количества баллов за пройденный тест.

Всем пользователям доступен выход из системы обратно к авторизации и регистрации.

## 2.2. Внешняя спецификация

## 2.2.1. Описание задачи

Контекстная диаграмма процесса контроля освоения знаний и практических умений обучающихся представлена на Рисунке 1. На данной диаграмме показаны процессы, выполняемые в организации для обеспечения учебного процесса преподавателями для обучающихся.

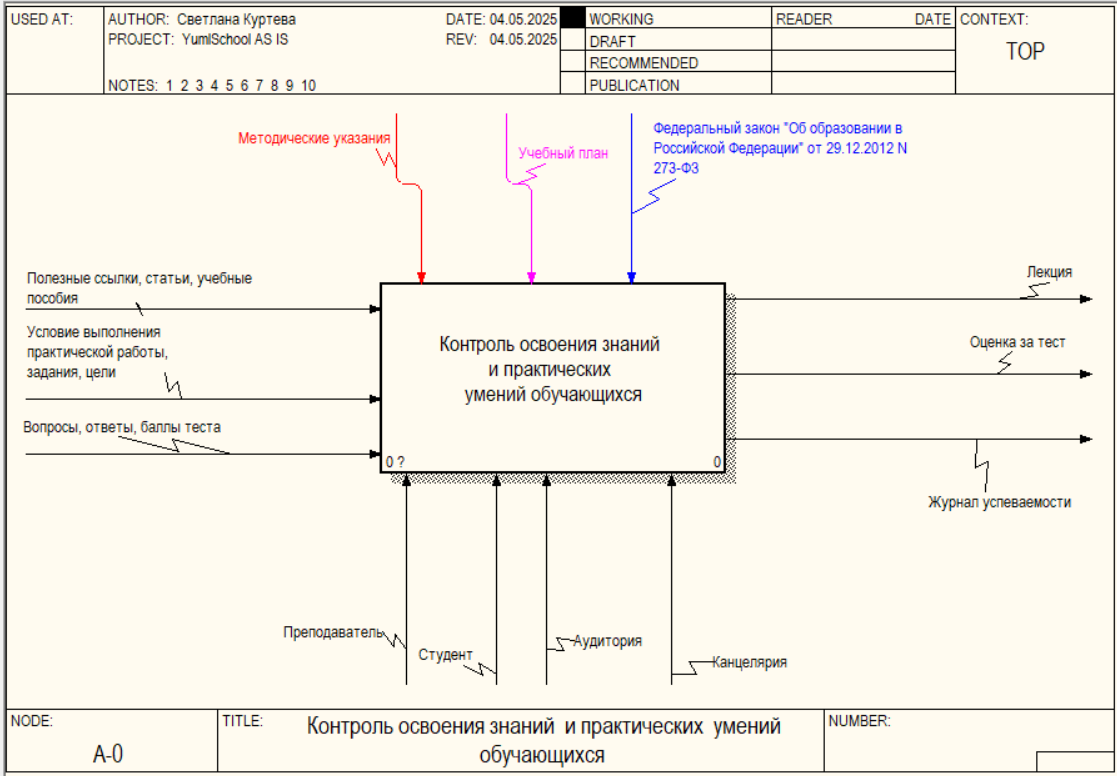


Рисунок 1 – Контекстная диаграмма

Объектами на входе процесса контроля освоения знаний и практических умений обучающихся являются:

* Полезные ссылки, статьи, учебные пособия – материалы, которые отбираются для ведения лекции;
* Условие выполнения практической работы, задания, цели – требования для практической работы;
* Вопросы, ответы, баллы теста – материалы для составления теста.

Механизмами процесса контроля освоения знаний и практических умений обучающихся являются:

* Преподаватель – сотрудник места обучения, ответственный за учебный процесс;
* Студент – обучающийся, присутствует на лекциях, практических работах, проходит тестирование;
* Аудитория – место обучения для проведения лекций, практических работ, тестирования;
* Канцелярия – оборудования, использующиеся для проведения лекций, практических работ, мониторинга успеваемости и тестирования.

Управлениями процесса контроля освоения знаний и практических умений обучающихся являются:

* Методические указания – документы, которые содержат рекомендации и требования по организации и проведению учебного процесса;
* Учебный план – нормативный документ образовательной организации, который определяет перечень, трудоемкость, последовательность и распределение по периодам обучения учебных предметов;
* Федеральный закон "Об образовании в Российской Федерации" от 29.12.2012 N 273-ФЗ – нормативно-правовой акт, регулирующий систему образования в Российской Федерации.

Объектами на выходе процесса контроля освоения знаний и практических умений обучающихся являются:

* Лекция – проведенная лекция преподавателем;
* Оценка за тест – оценка студента за проведенный тест;
* Журнал успеваемости – журнал с успеваемостью студентов заполненный преподавателем.

На Рисунке 2 представлена декомпозиция процесса контроля освоения знаний и практических умений обучающихся согласно нотации IDEF0.

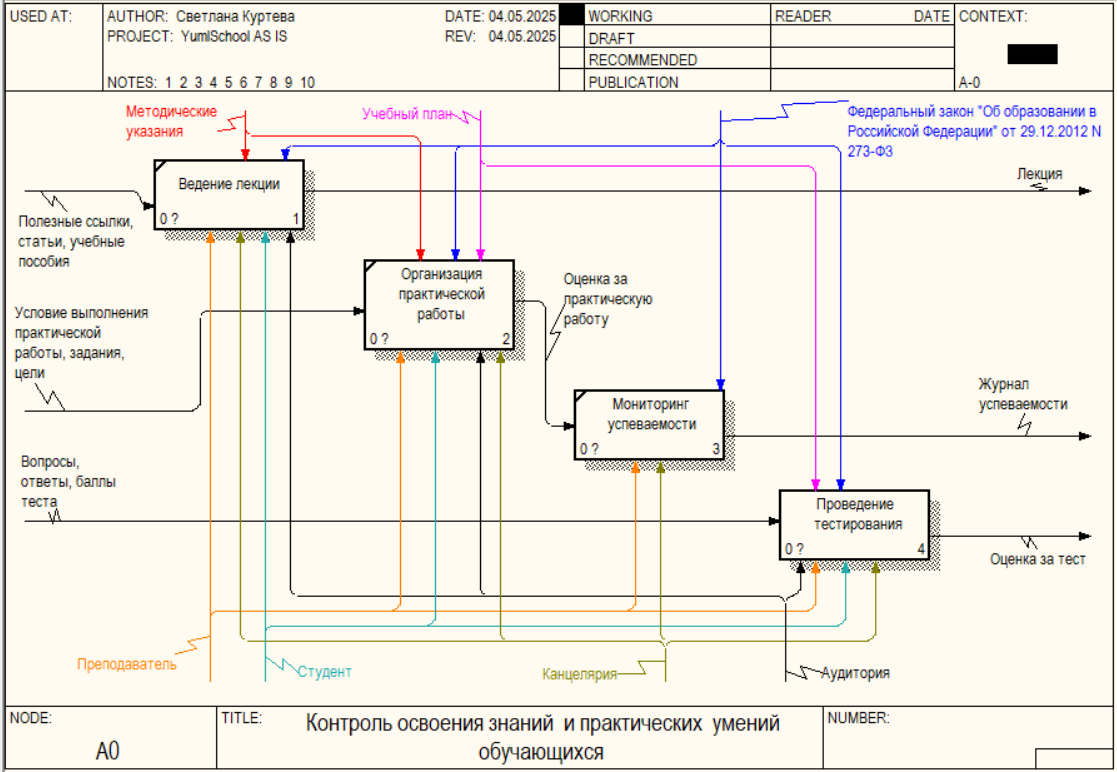


Рисунок 2 – Декомпозиция процесса контроля освоения знаний и практических умений обучающихся

Процесс контроля освоения знаний и практических умений обучающихся можно разделить на следующие подпроцессы:

* Ведение лекции – процесс ведения лекции, состоящего из полезных ссылок, статей, учебные пособия, результатом которого станет проведенная лекция;
* Организация практической работы – процесс проведения практической работы входными данными которого станут требования практической работы, задания и цели, а результатом – оценка за практическую работу студента;
* Мониторинг успеваемости – процесс закрепления результатов практической работы путем записи их в журнал успеваемости;
* Проведение тестирования – процесс проведения тестирования, данными для теста которого будут вопросы, ответы и баллы теста, а результатом – оценка за тест студента.

На Рисунке 3 изображена декомпозиция процесса проведения тестирования.

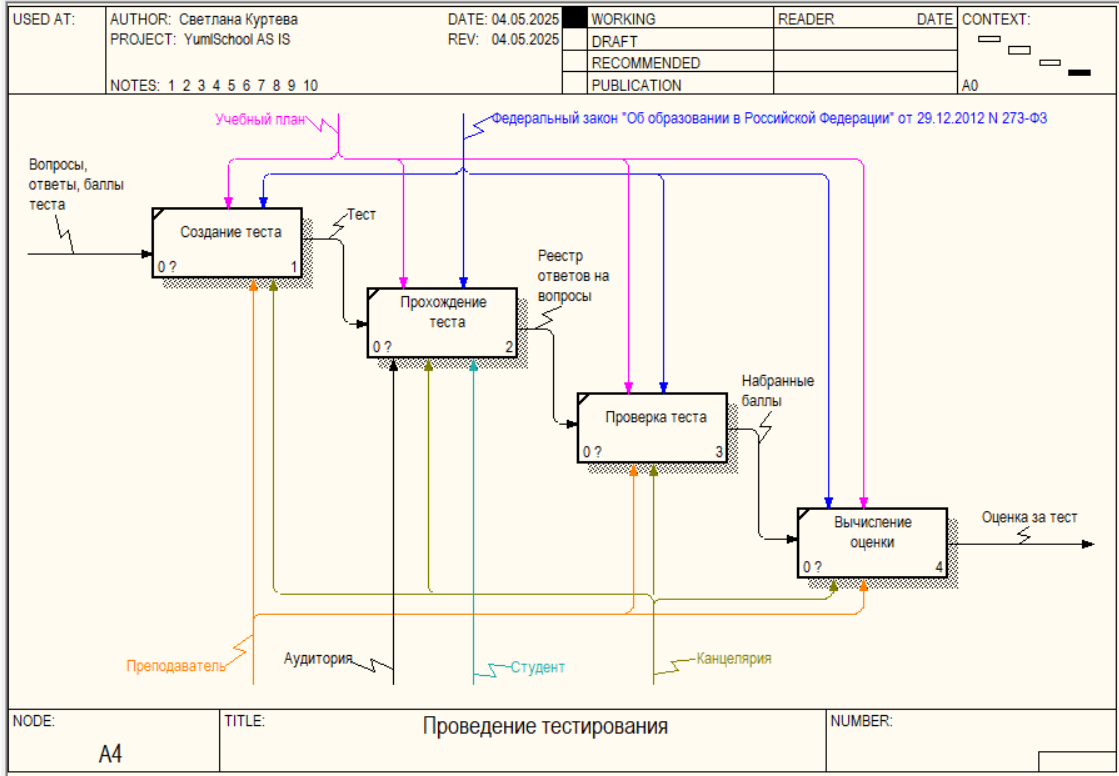


Рисунок 3 – Декомпозиция процесса проведения тестирования

Процесс проведения тестирования можно разделить на следующие подпроцессы:

* Создание теста – процесс создания теста, который создает преподаватель, начальными данными для которого будут вопросы, ответы и баллы теста для самого формирования теста;
* Прохождение теста – процесс прохождения теста который проходит студент, занося ответы на вопросы. Выходными данными будет реестр ответов на вопросы, который необходимо проверить;
* Проверка теста – процесс проверки теста, в котором преподаватель сверяет ответы студентов с заданными правильными ответами, итогом станут набранные баллы каждого теста студента;
* Вычисление оценки – процесс вычисления оценки из информации о набранных баллах студента за тест. Преподаватель согласно критериям теста вычисляет итоговую оценку за тест.

Данный процесс является множеством различных действий, которые выполняет преподаватель для выполнения задачи – ведения учебного процесса. Проблемой данного процесса является высокая трудоемкость и большой объем бумажной работы при мониторинге успеваемости студентов. Преподавателю необходимо вести множество бумажных журналов успеваемости, фиксируя оценки за практические работы, лабораторные, контрольные работы для каждого студента. Это занимает значительное количество времени и усилий, особенно при большом количестве обучающихся в группах. Ведение бумажной документации подвержено ошибкам, потере данных, неудобно в хранении и анализе. Журнал успеваемости не всегда под рукой и таким образом сложно оперативно отслеживать динамику успеваемости студентов и вовремя принимать корректирующие меры.

Другая проблема связана с процессом тестирования. Преподавателю приходится самостоятельно проверять тесты студентов, вручную вычислять набранные баллы и выставлять оценки. При большом количестве тестируемых студентов это становится физически сложной задачей. Ручная проверка тестов подвержена человеческим ошибкам, что может привести к неточным результатам. Сложно обеспечить объективность и единообразие оценивания при ручной проверке тестов. Отсутствует возможность быстрого анализа результатов тестирования, выявления слабых мест в знаниях студентов.

Одним из решений данной проблемы является внедрение системы контроля освоения знаний и практических умений обучающихся. Внедрение данной системы позволит преподавателю быстрее мониторить успеваемость студентов и проверять большое количество тестов.

В процессе рассмотрения основных возможных вариантов приобретения системы, позволяющей автоматизировать данный процесс были выделены следующие:

1) Покупка готового решения. Преимущества данного варианта в том, что готовое решение уже готово к внедрению, не требует затрат времени, не имеет ошибок. При покупке системы она будет скорее всего испытана на множестве предприятий похожей сферы деятельности.

В ходе анализа программ автоматизированных систем по контролю освоения знаний и практических умений обучающихся не было обнаружено решения, которое хотя бы частично подходило под функциональные требования к системе.

В ходе анализа программ автоматизированных систем по контролю освоения знаний и практических умений обучающихся было обнаружено решение, которое частично подходит под функциональные требования к системе – Google Classroom. Система подходит под часть функций требуемой программы, но в ней отсутствует система мониторинга успеваемости.

2) Разработка собственного программного решения. Отличительной чертой данного варианта является возможность разработки всех функций, которые необходимы заказчику. Такой способ решения проблемы требует много затрат, но за неимением готовых программных продуктов данный способ является единственным решением проблемы автоматизации процесса.

В ходе анализа предметной области была выявлена следующая проблема: невозможность быстро проверить большое количество тестов студентов и рассчитать оценку исходя из критериев и трудоемкая работа с мониторингом успеваемости, где преподавателю необходимо заполнять необходимые данные о результатах тестирования в отчетные документы, такие как журнал успеваемости и зачетная книжка.

Разрабатываемая система позволила бы не только перевести очный процесс обучения в приложение, но и позволит увеличить общее количество студентов и преподавателей, так как система позволит проводить учебный процесс в дистанционной системе, позволяя студентам и преподавателям из различных регионов и даже стран присоединиться к процессу обучения.

Исходя из вышесказанного верным решением будет разработка собственного программного продукта, который позволит студентам и преподавателям осуществлять учебную деятельность в рамках образовательного процесса. Данная система позволит ускорить процесс проверки тестирования, а также настроит удобный мониторинг успеваемости студентов.

На Рисунке 4 представлена контекстная диаграмма процесса контроля освоения знаний и практических умений обучающихся с использованием проектируемой автоматизированной системой.

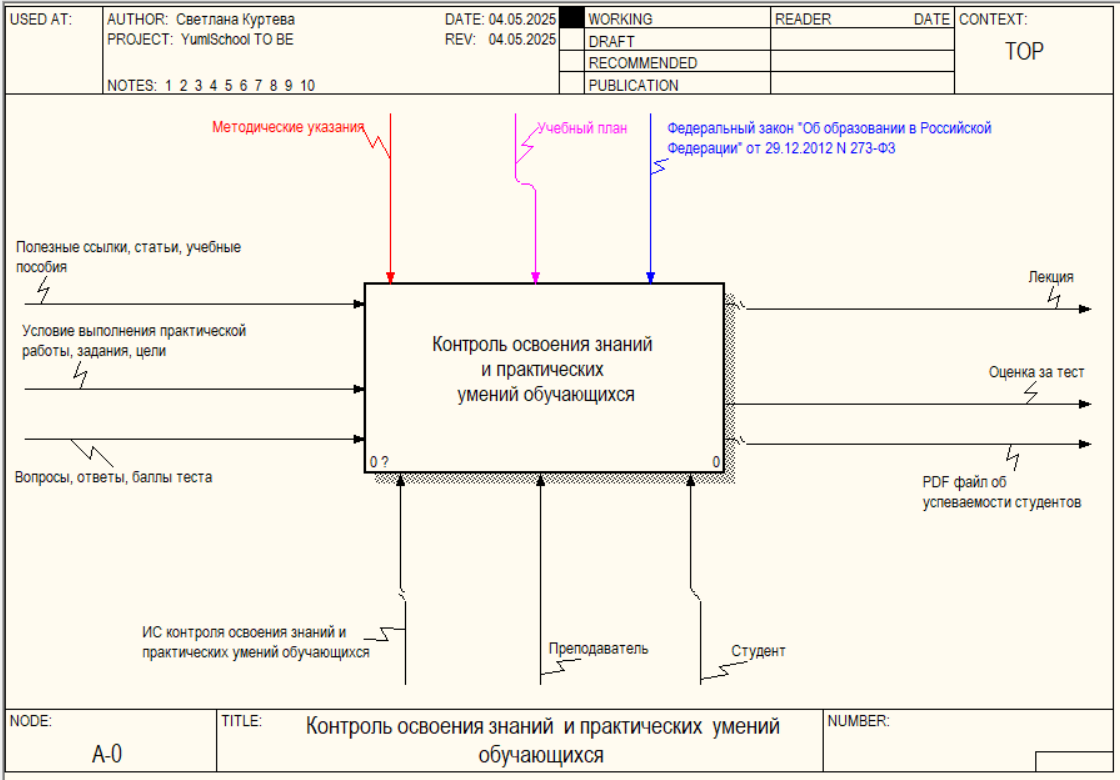


Рисунок 4 – Контекстная диаграмма

Объектами на входе процесса контроля освоения знаний и практических умений обучающихся являются:

* Полезные ссылки, статьи, учебные пособия – материалы, которые отбираются для ведения лекции;
* Условие выполнения практической работы, задания, цели – требования для практической работы;
* Вопросы, ответы, баллы теста – материалы для составления теста.

Механизмами процесса контроля освоения знаний и практических умений обучающихся являются:

* Преподаватель – сотрудник места обучения, ответственный за учебный процесс;
* Студент – обучающийся, присутствует на лекциях, практических работах, проходит тестирование;
* ИС контроля освоения знаний и практических умений обучающихся – система предоставляющая проводить учебный процесс в приложении.

Управлениями процесса контроля освоения знаний и практических умений обучающихся являются:

* Методические указания – документы, которые содержат рекомендации и требования по организации и проведению учебного процесса;
* Учебный план – нормативный документ образовательной организации, который определяет перечень, трудоемкость, последовательность и распределение по периодам обучения учебных предметов;
* Федеральный закон "Об образовании в Российской Федерации" от 29.12.2012 N 273-ФЗ – нормативно-правовой акт, регулирующий систему образования в Российской Федерации.

Объектами на выходе процесса контроля освоения знаний и практических умений обучающихся являются:

* Лекция – проведенная лекция преподавателем;
* Оценка за тест – оценка студента за проведенный тест;
* PDF файл об успеваемости студентов – файл с успеваемостью студентов заполненный автоматически ИС контроля освоения знаний и практических умений обучающихся.

На Рисунке 5 представлена декомпозиция процесса контроля освоения знаний и практических умений обучающихся с использованием проектируемой автоматизированной системой.

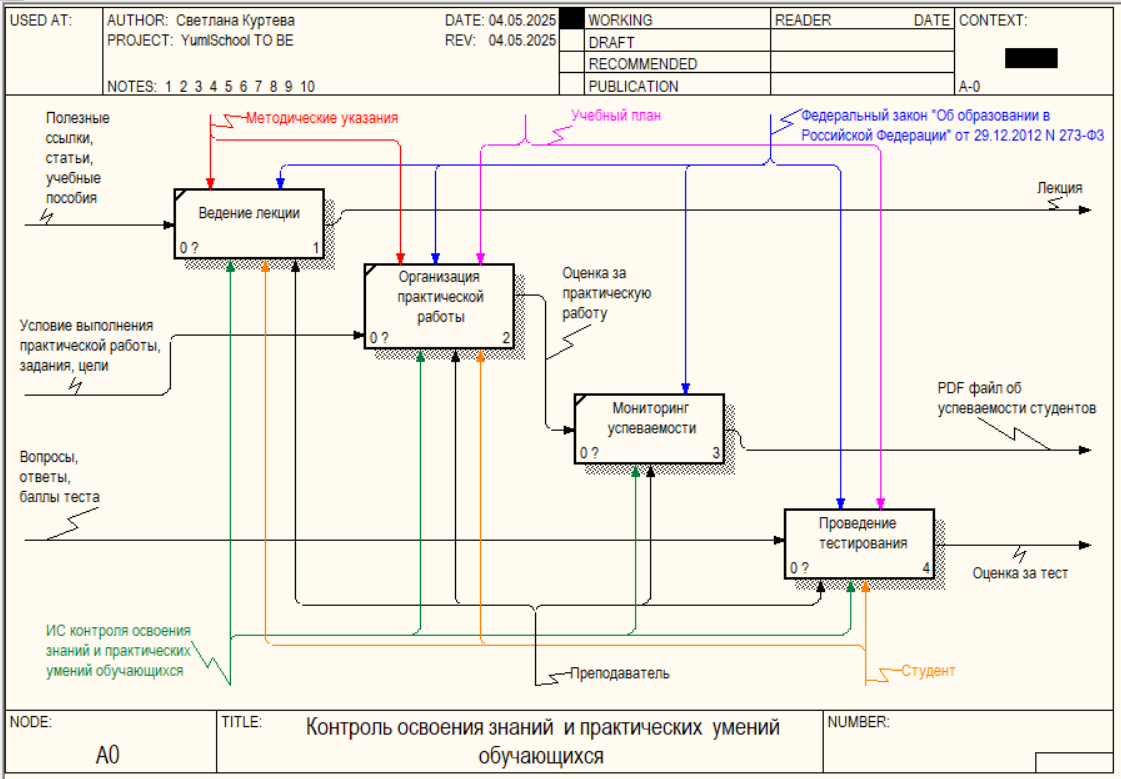


Рисунок 5 – Декомпозиция процесса контроля освоения знаний и практических умений обучающихся

Процесс контроля освоения знаний и практических умений обучающихся можно разделить на следующие подпроцессы:

* Ведение лекции – процесс ведения лекции, состоящего из полезных ссылок, статей, учебные пособия, результатом которого станет проведенная лекция;
* Организация практической работы – процесс проведения практической работы входными данными которого станут требования практической работы, задания и цели, а результатом – оценка за практическую работу студента;
* Мониторинг успеваемости – процесс закрепления результатов практической работы путем автоматического переноса оценок за практическую работу в файл PDF об успеваемости студентов;
* Проведение тестирования – процесс проведения тестирования, данными для теста которого будут вопросы, ответы и баллы теста, а результатом – оценка за тест студента.

На Рисунке 6 изображена декомпозиция процесса проведения тестирования с использованием проектируемой автоматизированной системой.

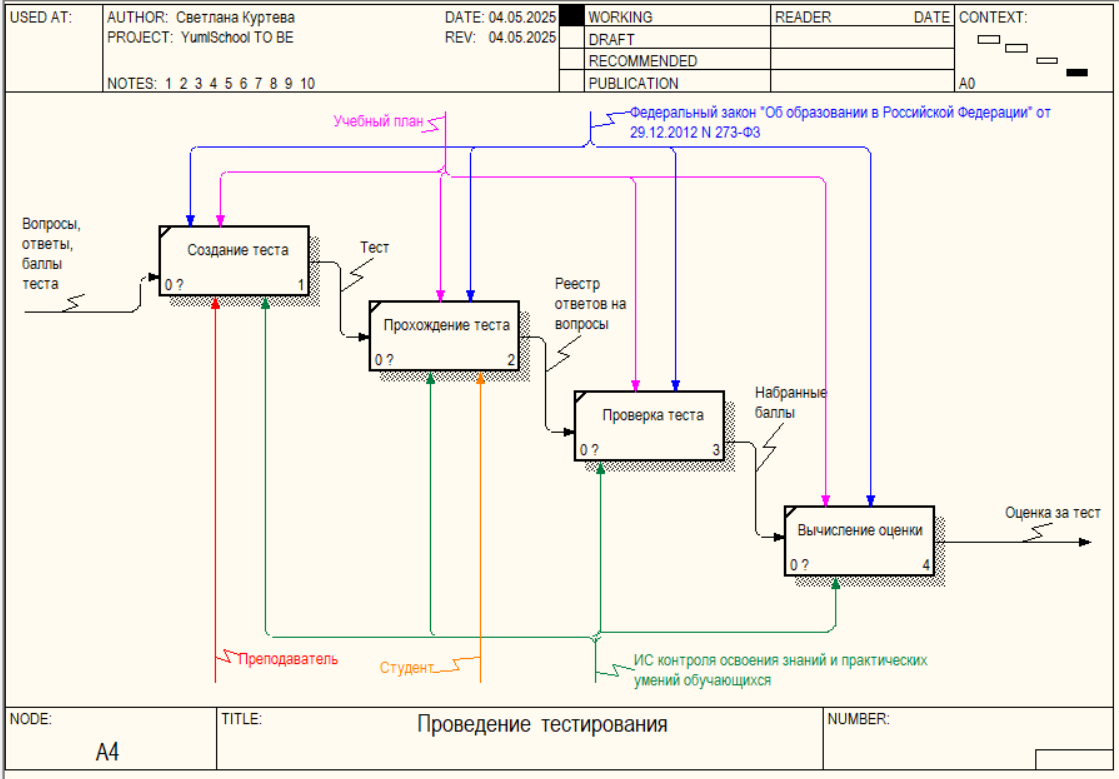


Рисунок 6 – Декомпозиция процесса проведения тестирования

Процесс проведения тестирования можно разделить на следующие подпроцессы:

* Создание теста – процесс создания теста, который создает преподаватель, начальными данными для которого будут вопросы, ответы и баллы теста для самого формирования теста;
* Прохождение теста – процесс прохождения теста который проходит студент, занося ответы на вопросы. Выходными данными будет реестр ответов на вопросы, который необходимо проверить;
* Проверка теста – процесс проверки теста, в котором ИС контроля освоения знаний и практических умений обучающихся сверяет ответы студентов с заданными правильными ответами, итогом станут набранные баллы каждого теста студента;
* Вычисление оценки – процесс вычисления оценки из информации о набранных баллах студента за тест. ИС контроля освоения знаний и практических умений обучающихся согласно критериям теста вычисляет итоговую оценку за тест.

После введения ИС контроля освоения знаний и практических умений обучающихся, проверка тестирования проводится программой как и мониторинг успеваемости студентов. Система позволит преподавателю быстрее осуществлять проведение тестирования и мониторинг успеваемости студентов посредством быстрого переноса оценок за практическую работу в PDF файл.

Для определения взаимодействия между пользователями исследуемого процесса была разработана UML диаграмма вариантов использования (диаграмма прецедентов). Данная диаграмма позволит визуализировать типы ролей и их взаимодействие друг с другом.

На рисунке 7 изображена диаграмма прецедентов процесса контроля освоения знаний и практических умений обучающихся.



Рисунок 7 – Диаграмма прецедентов

В процессе участвуют три актера:

* Администратор;
* Преподаватель;
* Студент;

Каждой роли присущи следующие прецеденты:

Администратор:

* Регистрировать пользователей – распределение ролей между пользователями;
* Назначить группам и преподавателям предметы – создание предметов для групп и преподавателей;
* Управлять безопасностью – разграничение прав доступа (выбора ролей и групп) для пользователей. Выгрузка данных в JSON формате списка групп;

Преподаватель:

* Ведение учебного процесса – в процесс входит ведение практических работ, лекций, тестов;
* Выдача лекций – создание лекционного материала для студентов;
* Выдача практических работ – создание практической работы для студентов;
* Выдача тестов – создание тестов для студентов;
* Проверять выполненную работу – проверка выполненной работы студента по практической работе;
* Выгрузить PDF файл о статистике успеваемости – выгрузка файла о статистике успеваемости по практической работе;
* Поставить оценку --

Сперва диаграмма прецендентов – AS IS декомпозиции описание всего ПРОБЛЕМА ПРОЦЕССА который надо автоматизировать - одно из решений – to be декомпозиции – вывод что изменилось – диаграмма прецендентов пример 2.1.

Основной задачей является создание Разработать Web-приложение для контроля освоений знаний и практических умений обучающихся.

Данное программное решение должно состоять из реляционной базы данных под управлением СУБД Microsoft SQL Server Management Studio.

Приложение должно иметь возможность отображения различной информации, хранящейся в базе данных, с помощью технологии API. Помимо отображения данных должно быть предоставлено пользователю возможности добавления, изменения и удаления различной информации.

В приложении предусмотрены три роли:

* Администратор БД отвечает за администрирование системы.
* Преподавателю доступны управление тестами, лекциями, практическими работами и их проверка.
* Студенту доступно взаимодействие с практическими работами, прохождение тестирования, просмотр лекций.

Описание перечня пользователей с их функциями изображено на рисунке 7.

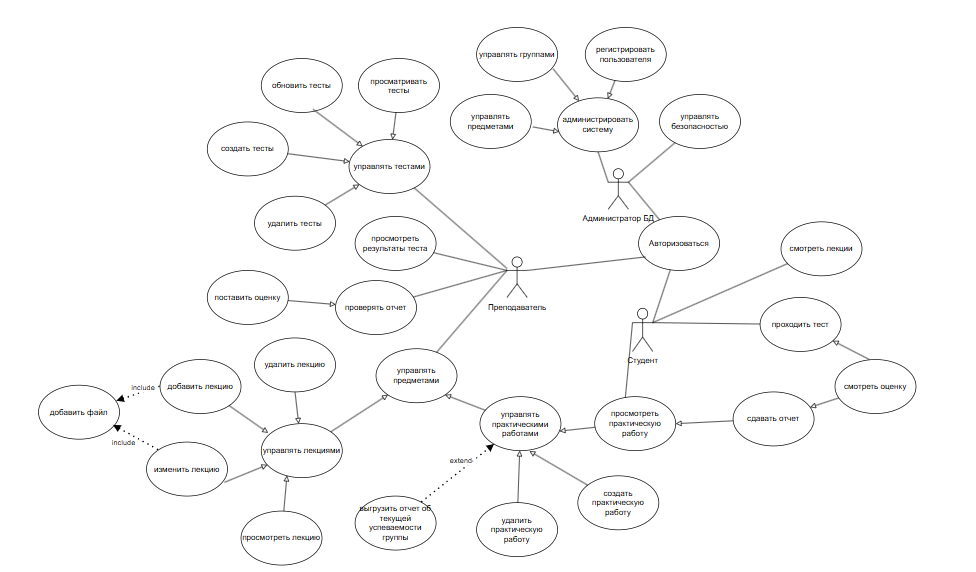


Рисунок 8 - Диаграмма прецендентов

### 2.2.2. Входные и выходные данные

В таблице 4 представлены входные данные приложения.

Таблица 3 - Входные данные

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя | Тип | Ограничение | | | Формат ввода | | Описание |
| 1 | 2 | 3 | | | 4 | | 5 |
| Персональные данные | | | | | | | |
| Login | Строка | | [A-Za-z0-9@]  {5, 100} | Поле ввода | | Логин пользователя | |
| Password | Строка | | [A-Za-z0-9@]  {5, 100} | Поле ввода | | Пароль пользователя | |
| First\_Name | Строка | | [А-Яа-я]{2, 45} | Поле ввода | | Имя пользователя | |
| Second\_Name | Строка | | [А-Яа-я]{2, 45} | Поле ввода | | Фамилия пользователя | |
| Middle\_Name | Строка | | [А-Яа-я]{2, 45} | Поле ввода | | Отчество пользователя | |
| Данные лекции | | | | | | | |
| Name\_Lection | Строка | | [A-Za-z0-9@]  {5, 100} | Поле ввода | | Название лекции | |
| Description\_lection | Строка | | [A-Za-z0-9@]  {5, 100} | Поле ввода | | Описание лекции | |
| URL\_file\_lection | Строка | | Нет  ограничений | Поле ввода | | Ссылка на файл лекции | |
| Данные практической работы | | | | | | | |
| Name\_ PracWork | Строка | | [A-Za-z0-9@]  {5, 100} | Поле ввода | | Название практической работы | |
| Description\_ Name\_ PracWork | Строка | | [A-Za-z0-9@]  {5, 100} | Поле ввода | | Описание практической работы | |
| URL\_file\_ Name\_ PracWork | Строка | | Нет  ограничений | Поле ввода | | Ссылка на файл практической работы | |
| Данные о тесте | | | | | | | |
| Name\_ Test | Строка | | [A-Za-z0-9@]  {5, 100} | Поле ввода | | Название теста | |
| Description\_ Name\_ Test | Строка | | [A-Za-z0-9@]  {5, 100} | Поле ввода | | Описание теста | |
| Question | Строка | | [A-Za-z0-9@]  {5, 100} | Поле ввода | | Вопрос | |
| Answer | Строка | | [A-Za-z0-9@]  {5, 100} | Поле ввода | | Ответ | |
| Данные о предмете | | | | | | | |
| Subject\_Name | Строка | | [А-Яа-я]{2, 45} | Поле ввода | | Название предмета | |

В таблице 5 представлены выходные данные приложения.

Таблица 5. Выходные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Тип | Ограничение | Описание |
| 1 | 2 | 3 | 4 |
| Выходные данные | | | |
| Персональные данные | | | |
| First\_Name | Строка | [А-Яа-я]{2, 45} | Имя пользователя |
| Second\_Name | Строка | [А-Яа-я]{2, 45} | Фамилия  пользоваться |
| Middle\_Name | Строка | [А-Яа-я]{2, 45} | Отчество  пользователя |
| Данные о лекции | | | |
| Name\_Lection | Строка | [A-Za-z0-9@]  {5, 100} | Название лекции |
| Description\_lection | Строка | [A-Za-z0-9@]  {5, 100} | Описание лекции |
| URL\_file\_lection | Строка | Нет  ограничений | Ссылка на файл лекции |
| Данные о практической работе | | | |
| Name\_ PracWork | Строка | [A-Za-z0-9@]  {5, 100} | Название практической работы |
| Description\_ Name\_ PracWork | Строка | [A-Za-z0-9@]  {5, 100} | Описание практической работы |
| URL\_file\_ Name\_ PracWork | Строка | Нет  ограничений | Ссылка на файл практической работы |
| Grade\_ PracWork | Строка | [2-5] | Оценка за практическую работу |
| Данные о тесте | | | |
| Grade\_Test | Строка | [2-5] | Оценка за тест |
| Score\_Test | Строка | [2-5] | Балл за тест |
| Name\_ Test | Строка | [A-Za-z0-9@]  {5, 100} | Название теста |
| Description\_ Name\_ Test | Строка | [A-Za-z0-9@]  {5, 100} | Описание теста |
| Question | Строка | [A-Za-z0-9@]  {5, 100} | Вопрос |
| Answer | Строка | [A-Za-z0-9@]  {5, 100} | Ответ |

### 2.2.3. Методы

В ходе написания программы были использованы следующие методы разработки программного обеспечения:

При разработке API использовалась методология ООП, а именно абстракция, в виде классов моделей с характеристиками объекта, которым обладает он же в БД, инкапсуляция, в виде методов, позволяющих скрыть подробности характеристик объекта, которые возвращают значения и наследование, каждый контроллер наследуется от родительского класса «ControllerBase».

Также использовалась методология REST API – это архитектурный стиль взаимодействия между клиентов и сервером по протоколу HTTP. REST API использует принципы REST:

• Клиент-серверная архитектура – клиент делает запрос на сервер, сервер обрабатывает запросы и отправляет клиенту ответ

• Отсутствие состояния – сервер не хранит состояние клиента между запросами;

• Единообразие интерфейса – ресурсы представлены в виде URL, для всех ресурсов используется одни и те же HTTP-методы (GET, POST, PUT, DELETE).

Для разработки веб-приложения использовался паттерн MVC (Model-View-Controller), но в рамках фреймворка «Django» он именуется как MTV (Model-Template-View). В данном паттерне подразумевается, что модели (Model) используются для работы с базой данных, описывает структуру данных и логику их обработки; шаблон (Template) отвечает за отображение данных пользователю – это файлы HTML, в которых можно динамически выводить данные, переданные из представлений; представления (View) прослойка между моделями и шаблонами, которая позволяет управлять логикой обработки запросов, т.е. представления запрашивают данные из моделей, после чего передают их в шаблоны для вывода на страницу. Схема паттерна MTV представлена на Рисунке 8.



Рисунок 9 - Схема паттерна MTV

Для разработки веб-приложения, использовалась технология программирования – Django. Представляющую свою реализацию в виде паттерна MTV (подробнее в п. 2.2.3).

Для разработки веб-API, использовалась технология программирования – ASP.NET Core WEB-API, с использованием EntityFramework.

Для разработки базы данных, использовалась технология программирования – Microsoft SQL Server Management.

### 2.2.4. Тесты

1. По формальности тестирования.

Тестирование по тестам – тестирование по предварительно написанным тест-кейсам.

2. По исполнению кода.

Динамическое тестирование - во время тестирования код исполняется.

3. По уровню тестирования.

Системное – проверка работы всей системы на соответствие заявленным требованиям к программному продукту.

4. По целям.

Функциональное тестирование направлено на проверку того, какие функции ПО реализованы, и того, насколько верно они реализованы.

5. По степени автоматизации.

Ручное – без использования дополнительных программных средств.

6. По позитивности сценария.

Позитивным – проверка ПО на соответствие ожидаемому поведению. Негативным – проверяет, будет ли ПО работать в случае, когда поведение пользователя отличается от ожидаемого.

7. По знанию системы.

Тестирование «белого ящика» – тестирование программного продукта с доступом к коду.

Тестирование «черного ящика» – тестирование без доступа к коду продукта.

8. По разработке тестовых испытаний.

На основе требований – требование было определено до начала тестирования.

Контроль целостности, описывающих ситуации и реакции приложения на выполнения функций представлен в таблице

Таблица 4 - Тест возможности регистрации

| № | Действие  (входные данные) | Ожидаемый  результат | Фактический  результат | Статус теста (пройден/не пройден) |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Логин: vasin3485  Пароль: vasim4ek | Успешная  авторизация | Успешная  авторизация | Тест  пройден |
| 2 | Логин: vasin3485  Пароль: 123 | Авторизация не прошла. Над полями ввода отображается надпись «Неверный пароль» | Авторизация не прошла. Над полями ввода отображается надпись «Неверный пароль» | Тест  пройден |
| 3 | Логин: vasin34851  Пароль: vasim4ek | Авторизация не прошла. Над полями ввода отображается надпись «Такого логина не существует» | Авторизация не прошла. Над полями ввода отображается надпись «Такого логина не существует» | Тест  пройден |
| 4 | Логин: пустота  Пароль: пустота | Авторизация не прошла. Над полем ввода логина и пароля отображается сообщение «Заполните это поле.» | Авторизация не прошла. Над полем ввода логина и пароля отображается сообщение «Заполните это поле.» | Тест  пройден |
| 5 | Логин: vasin3485  Пароль: пустота | Авторизация не прошла. Над полем ввода пароля отображается сообщение «Заполните это поле» | Авторизация не прошла. Над полем ввода пароля отображается сообщение «Заполните это поле» | Тест  пройден |
| 6 | Логин: пустота  Пароль: vasim4ek | Авторизация не прошла. Над полем ввода логина отображается сообщение «Заполните это поле.» | Авторизация не прошла. Над полем ввода логина отображается сообщение «Заполните это поле.» | Тест  Пройден |

## 2.3.Проектирование

### 2.3.1. Схема архитектуры приложения

На Рисунке 9 представлена схема «Клиент-серверной» архитектуры веб-приложения.

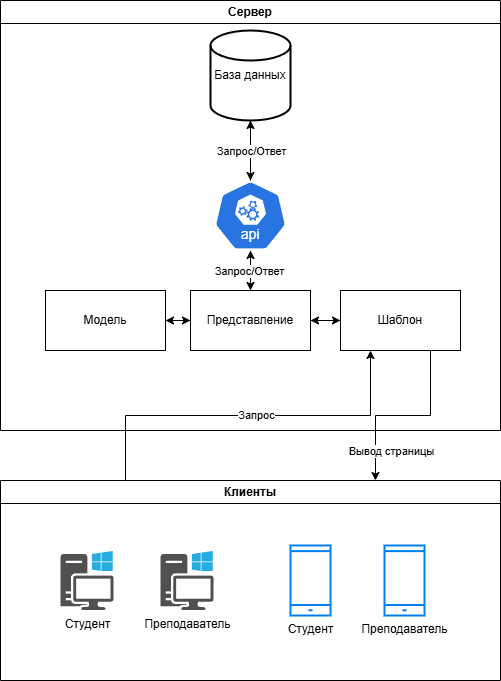


Рисунок 10 - Схема архитектуры веб-приложения

### 2.3.2. Логическая схема данных

Разработанная логическая модель базы данных, представленная на Рисунке 10, послужила основой для реализации логики манипуляции данными в проектируемой базе данных.

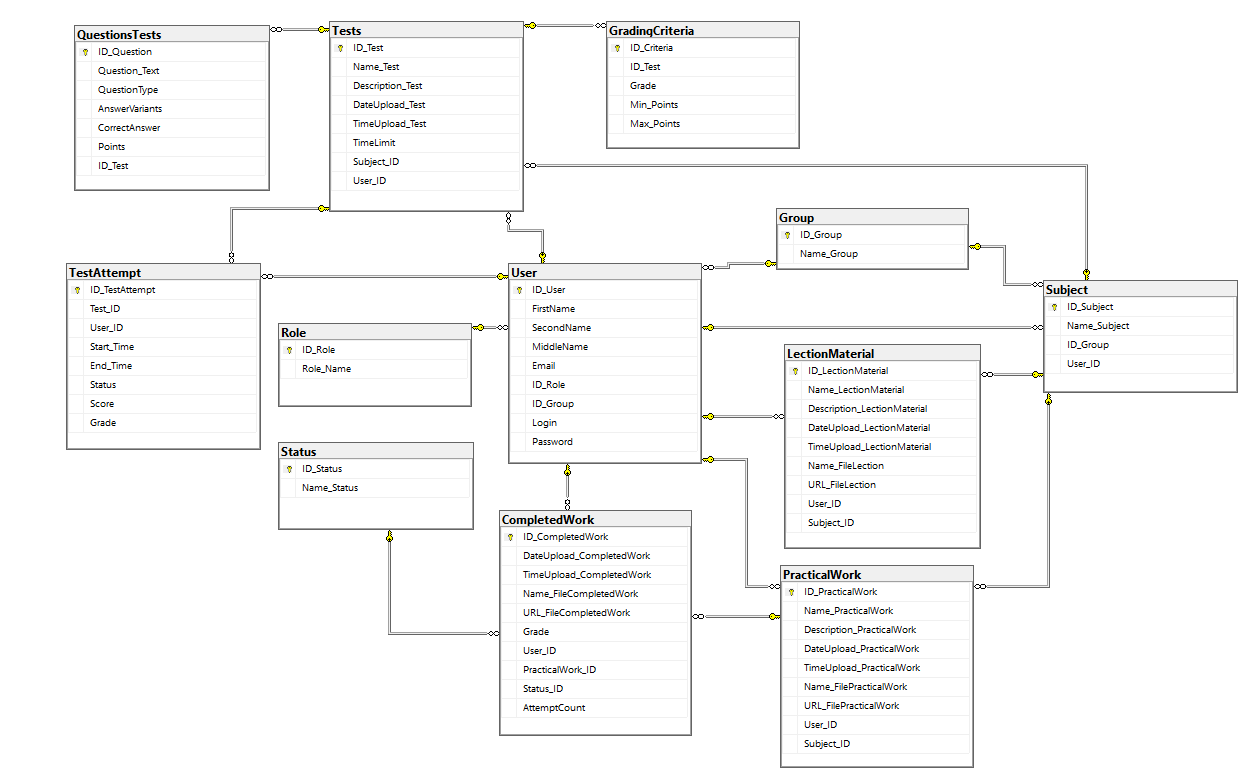


Рисунок 11 - Логическая схема данных

### 2.3.3. Физическая схема данных

На Рисунке 11 представлена физическая схема данных.

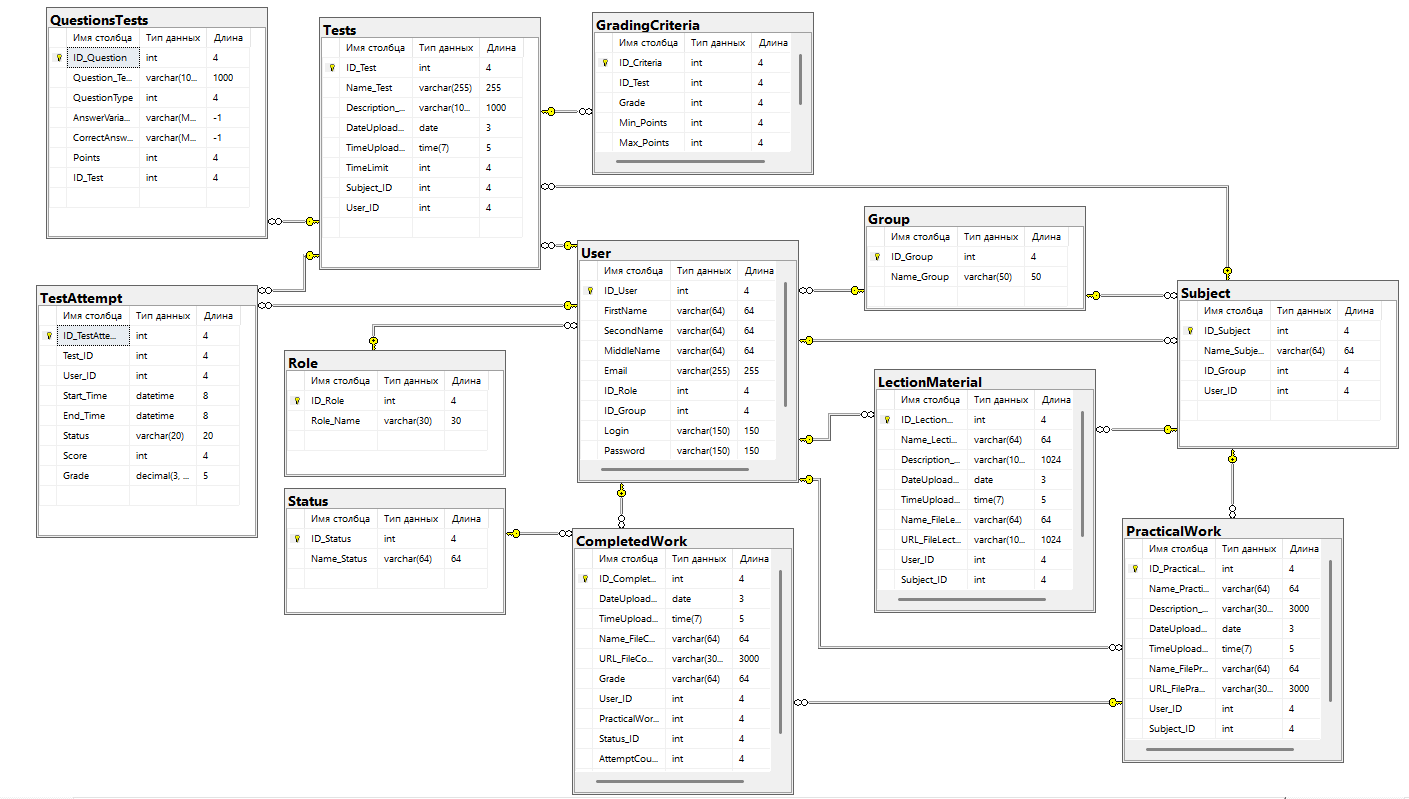


Рисунок 12 - Физическая схема данных

В проектируемой базе данных было выделено 12 сущностей (таблиц), каждая из которых содержит набор атрибутов (полей) для записи информации. В таблице 6 представлен словарь данных реализуемой базы данных для данного мобильного приложения.

Таблица 6. Словарь

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Поле | Тип данных | Обязательность заполнения | Описание |
| 1 | 2 | 3 | 4 | 5 |
| PK | ID\_TestAttempt | Int | Not Null | Идентификатор попытки теста |
| FK | Test\_ID | int | Not Null | Код теста |
| FK | User\_ID | int | Not Null | Код пользователя |
|  | Start\_Time | datetime | Not Null | Начало времени |
|  | End\_Time | datetime | Не обязательно | Конец времени |
|  | Status | varchar(20) | Not Null | Статус |
|  | Score | int | Not Null | Выполненная работа |
|  | Grade | decimal(3, 1) | Not Null | Оценка за тест |
| PK | ID\_Criteria | int | Not Null | Идентификатор критерия теста |
| FK | ID\_Test | int | Not Null | Код теста |
|  | Grade | int | Not Null | Оценка критерия |
|  | Min\_Points | int | Not Null | Минимальное количество баллов |
|  | Max\_Points | int | Not Null | Максимальное количество баллов |
| PK | ID\_Test | int | Not Null | Идентификатор теста |
|  | Name\_Test | varchar(255) | Not Null | Название |
|  | Description\_Test | varchar(1000) | Не обязательно | Описание |
|  | DateUpload\_Test | date | Не обязательно | Дата выдачи |
|  | TimeUpload\_Test | time(7) | Не обязательно | Время выдачи |
|  | TimeLimit | int | Не обязательно | Лимит теста |
| FK | Subject\_ID | int | Не обязательно | Код предмета |
| FK | User\_ID | int | Не обязательно | Код пользователя |
| PK | ID\_LectionMaterial | int | Not Null | Идентификатор лекции |
|  | Name\_LectionMaterial | varchar(64) | Not Null | Название лекции |
|  | Description\_LectionMaterial | varchar(1024) | Not Null | Описание |
|  | DateUpload\_LectionMaterial | date | Не обязательно | Дата выдачи |
|  | TimeUpload\_LectionMaterial | time(7) | Не обязательно | Время выдачи |
|  | Name\_FileLection | varchar(64) | Not Null | Название для файла лекции |
|  | URL\_FileLection | varchar(1024) | Not Null | URL лекции |
| FK | User\_ID | int | Не обязательно | Код пользователя |
| FK | Subject\_ID | int | Не обязательно | Код предмета |
| PK | ID\_Subject | int | Not Null | Идентификатор предмета |
|  | Name\_Subject | varchar(64) | Not Null | Название предмета |
| FK | ID\_Group | int | Not Null | Код группы |
| FK | User\_ID | int | Not Null | Код пользователя |
| PK | ID\_Group | int | Not Null | Идентификатор группы |
|  | Name\_Group | varchar(50) | Not Null | Название группы |
| PK | ID\_Question | int | Not Null | Идентификатор вопроса в тесте |
|  | Question\_Text | varchar(1000) | Not Null | Текст вопроса |
|  | QuestionType | int | Not Null | Тип вопроса |
|  | AnswerVariants | varchar(MAX) | Not Null | Вариант ответа |
|  | CorrectAnswer | varchar(MAX) | Not Null | Правильный ответ |
|  | Points | int | Не обязательно | Баллы за тест |
| FK | ID\_Test | int | Не обязательно | Код теста |
| PK | ID\_User | int | Not Null | Идентификатор пользователя |
|  | FirstName | varchar(64) | Not Null | Имя |
|  | SecondName | varchar(64) | Not Null | Фамилия |
|  | MiddleName | varchar(64) | Не обязательно | Отчество |
|  | Email | varchar(255) | Not Null | Почта |
| FK | ID\_Role | int | Not Null | Код роли |
| FK | ID\_Group | int | Not Null | Код группы |
|  | Login | varchar(150) | Not Null | Логин |
|  | Password | varchar(150) | Not Null | Пароль |
| PK | ID\_Group | int | Not Null | Идентификатор группы |
|  | Name\_Group | varchar(50) | Not Null | Название группы |
| PK | ID\_Role | int | Not Null | Идентификатор роли |
|  | Role\_Name | varchar(30) | Not Null | Название роли |
| PK | ID\_Status | int | Not Null | Идентификатор статуса |
|  | Name\_Status | varchar(64) | Not Null | Название статуса |
| PK | ID\_CompletedWork | int | Not Null | Идентификатор выполненной работы |
|  | DateUpload\_CompletedWork | date | Не обязательно | Дата сдачи |
|  | TimeUpload\_CompletedWork | time(7) | Не обязательно | Время сдачи |
|  | Name\_FileCompletedWork | varchar(64) | Not Null | Название файл |
|  | URL\_FileCompletedWork | varchar(3000) | Not Null | URL файла |
|  | Grade | varchar(64) | Not Null | Оценка |
| FK | User\_ID | int | Не обязательно | Код пользователя |
| FK | PracticalWork\_ID | int | Не обязательно | Код практической работы |
| FK | Status\_ID | int | Не обязательно | Статус работы |
|  | AttemptCount | int | Не обязательно | Количество попыток |
| PK | ID\_PracticalWork | int | Not Null | Идентификатор практической работы |
|  | Name\_PracticalWork | varchar(64) | Not Null | Название практической |
|  | Description\_PracticalWork | varchar(3000) | Not Null | Описание |
|  | DateUpload\_PracticalWork | date | Не обязательно | Дата выдачи |
|  | TimeUpload\_PracticalWork | time(7) | Не обязательно | Время выдачи |
|  | Name\_FilePracticalWork | varchar(64) | Not Null | Название файла |
|  | URL\_FilePracticalWork | varchar(3000) | Not Null | URL файла |
| FK | User\_ID | int | Не обязательно | Код пользователя |
| FK | Subject\_ID | int | Не обязательно | Код предмета |

### 2.3.4. Диаграмма классов

На рисунке 12 представлена диаграмма классов Web-приложения. На ней графически изображены классы, которые состоят из полей, методов и свойств.

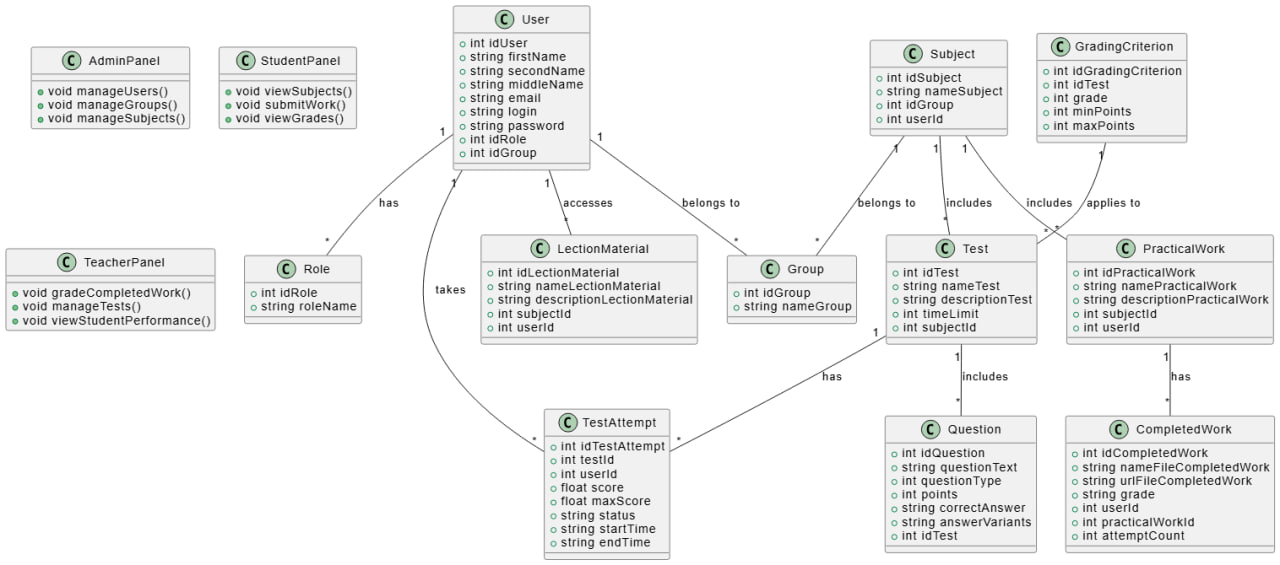


Рисунок 13 - Диаграмма классов

### 2.3.4. Функциональная схема

На рисунке 13 изображена функциональная схема

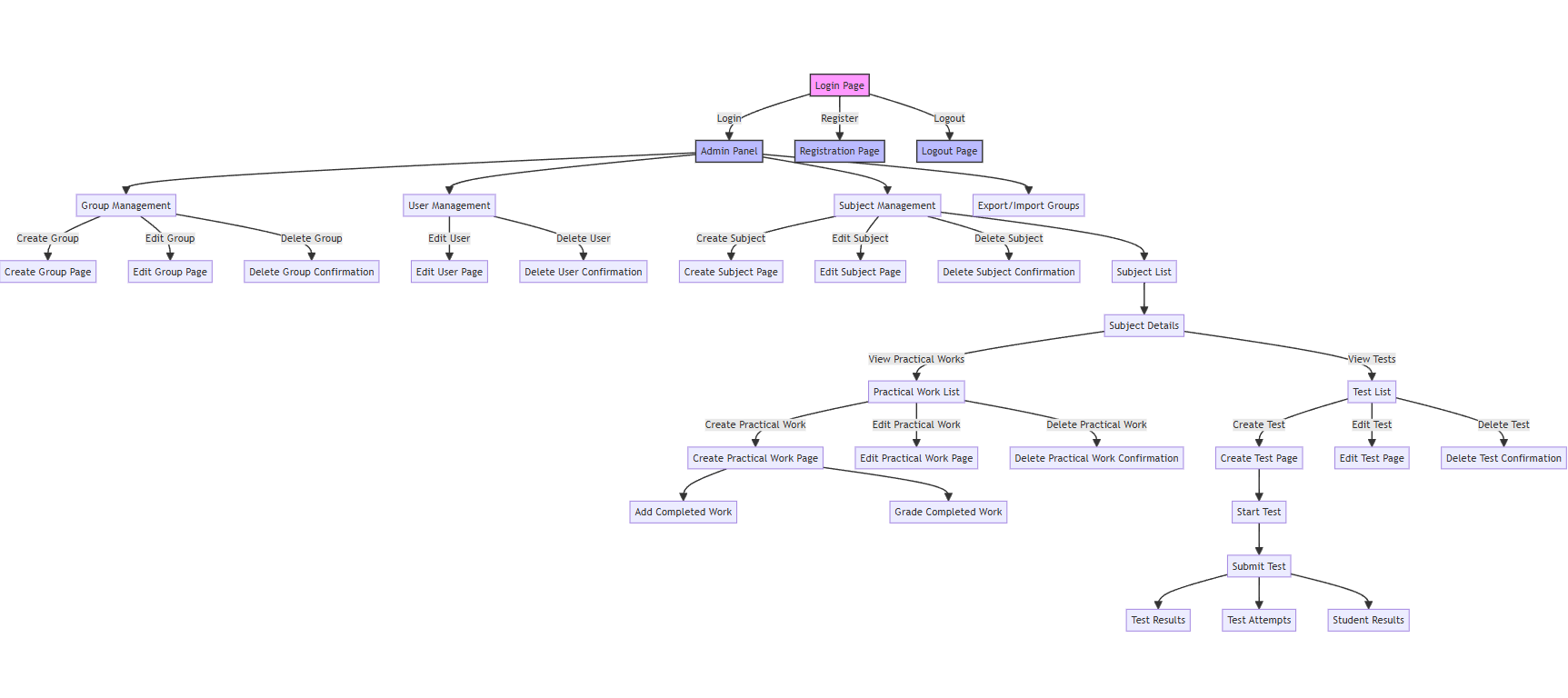


Рисунок 14 - Функциональная схема

## 2.4. Результат работы программы

На рисунке 14 видны элементы основного окна для студента, здесь видны:

- Данные о сессии пользователя, его фамилия и имя, кнопка «Выход» - для выхода из системы и возврата на страницу авторизации.

- Данные о предметах пользователя в которых он состоит. Здесь видны название предмета, группа, преподаватель предмета. По нажатию на предмет происходит переход на страницу предмета.

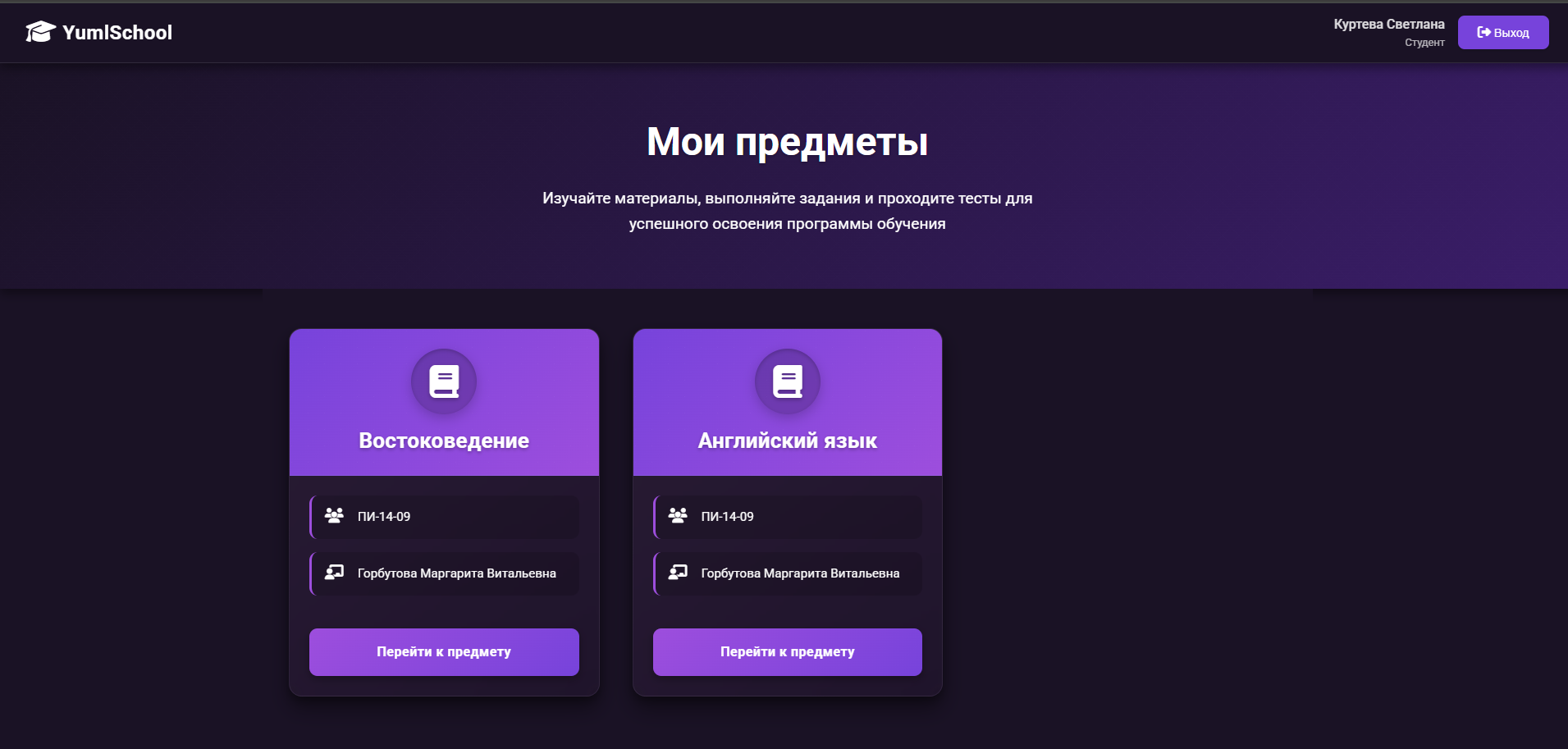


Рисунок 15 - Элементы основного окна у студента

На рисунке 15 изображена авторизация пользователей

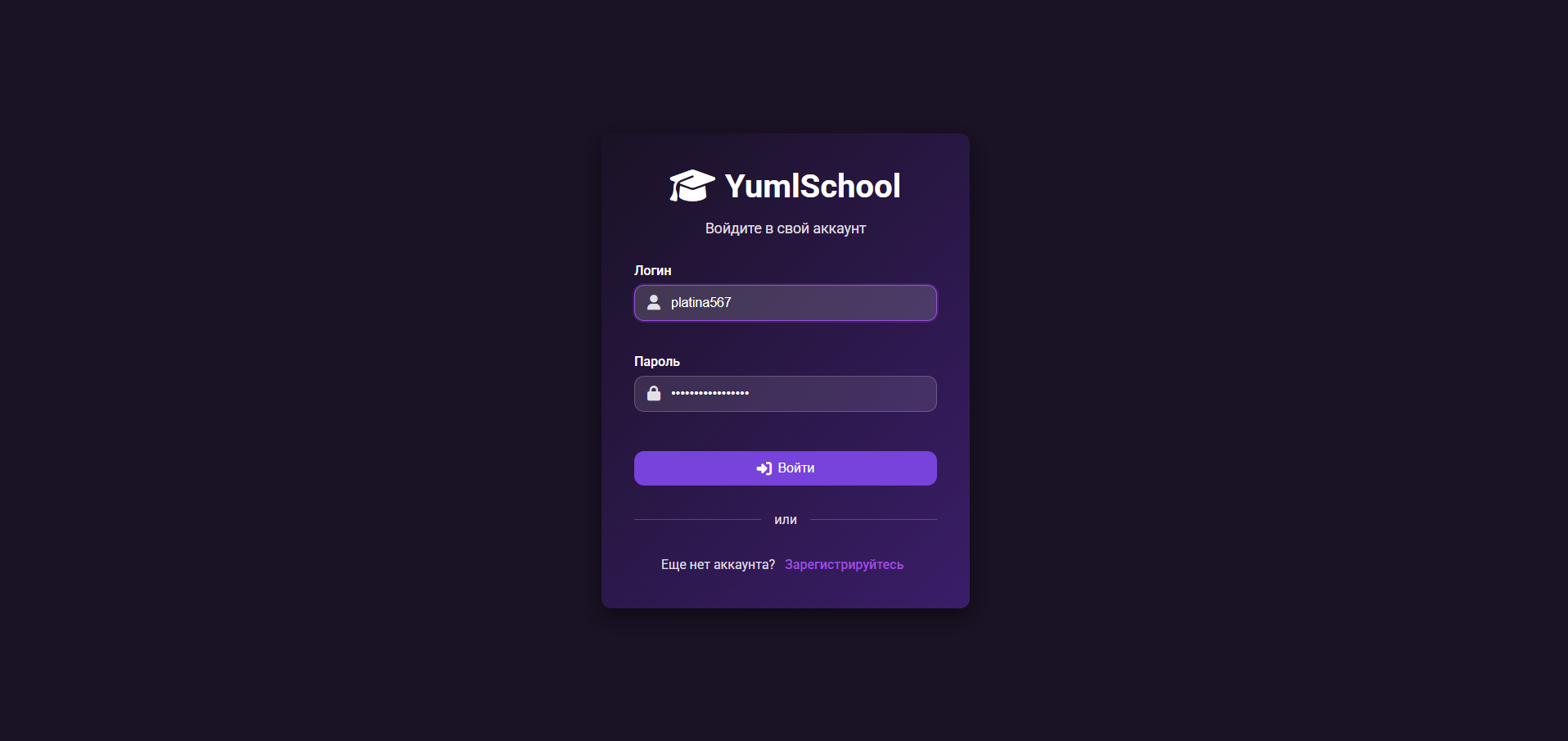


Рисунок 16 – Авторизация

На рисунке 16 изображена страница результатов тестирования

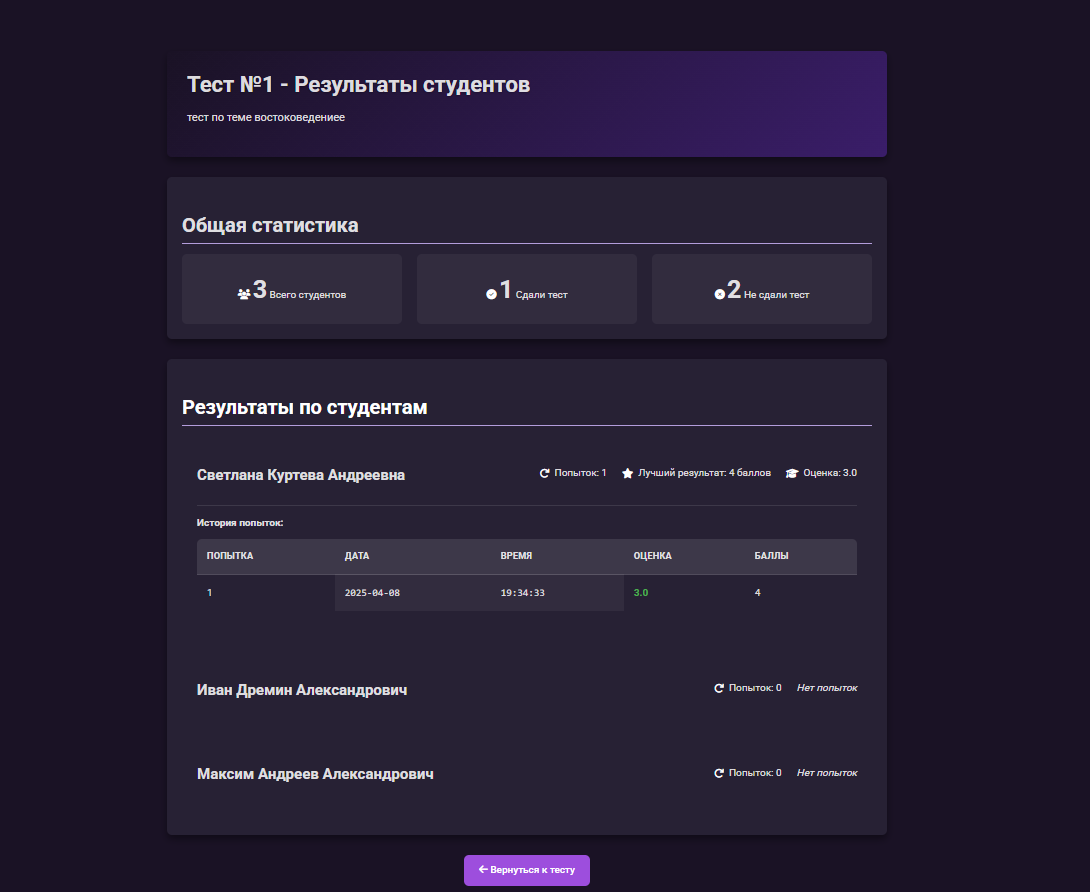


Рисунок 17 - Результаты тестирования

# 3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

## 3.1. Инструментальная среда разработки

Для написания базы данных была выбрана СУБД – SSMS. Выбор обусловлен тем, что разработка серверной части подразумевается на языке программирования «C#», с использованием технологии «ASP.NET Core».

Для написания Веб-API был выбран язык программирования «C#» и интеграционная среда разработки «Visual Studio 2022» и технология программирования – Веб-API ASP.Net Core. Выбор был сделан в пользу данного инструмента потому, что Visual Studio имеет непосредственно напрямую подключаться к базе данных СУБД SSMS. Также не маловажным фактом было наличие обширного количества сторонних библиотек, добавление проект которых, происходит по средству использования пакетов NuGet. Фреймворк, использованный при разработке API, был EntityFramework, позволяющий автоматически, создать модели, контроллеры и контекст базы данных, на основе подключения к самой базе данных, с последующей возможность вручную изменять, добавлять или же удалять модели или контроллеры, добавляя необходимые эндпоинты. Также EntityFramework имеет множество функций, позволяющих выполнять LINQ-запросы с List в асинхронном потоке, что значительно ускоряет выполнения и отправки ответа назад клиенту.

В качестве редактора программного кода, для верстки страниц был выбран «Visual Studio Code 1.78.2». Главным фактором в выборе данного редактора было наличие маркетплейса с внушительным количеством расширений, которые позволяют упростить и сделать процесс разработки комфортней и продуктивней. Второстепенным фактором является возможность работы с файлами любого формата.

Для написания веб-приложения «Информационная система контроля освоения знаний и практических умений обучающихся (на примере ФГБОУ ВО РЭУ им. Г.В. Плеханова)» был выбран язык программирования «Python», текстовый редактор «Microsoft Visual Studio Code» и технология программирования – «Django», данный выбор обусловлен тем, что «Django» предоставляет всё необходимое для разработки веб-приложения изначально, что позволяет использовать уже готовые и протестированные функции вместо реализации их с нуля. Структура проекта «Django» позволяет реализовывать разбиение логики на приложения – это означает, что проект можно без усилий масштабировать, добавляя в него новый функционал. «Django» автоматически предоставляет защиту для веб-приложения от таких критических уязвимостей как: XSS и CSRF-атак, и SQL-инъекций.

# Заключение

Главной задачей данной курсовой работы было: оптимизировать учебный процесс и автоматизировать деятельность преподавателя по выдаче тестов и их проверки для оценивания знаний студентов, по средству реализации веб-приложения «Информационная система контроля освоения знаний и практических умений обучающихся (на примере ФГБОУ ВО РЭУ им. Г.В. Плеханова)» с возможностью управления курсами: создавать их, изменять, приглашать студентов, и материалами: загружать материалы по темам в курсы со студентами; проведением тестирования: создание тестирований, добавление в них вопросов, с возможностью указания вариантов ответа и выбором правильного для того, чтобы автоматизировать их проверку и подведения оценки знаний.

В результате выполнения курсовой работы, были получены навыки по работе с передачей файлов между клиентом и сервером, с возможностью реализации функции скачивания это файла с веб-приложения на устройство клиента. Были получены навыки по реализации WEB-API с использованием фреймворка «ASP.NET Core».

В результате выполнения курсовой работы была разработана система онлайн-образования, которая закрывала проблематику, описанную в задаче курсовой работы, была достигнута поставленная цель по повышению навыков реализации веб-приложения с использованием фреймворка «Django», реализации API с использованием фреймворка «ASP.NET Core», реализации валидации навыков на стороне сервера базы данных.

# Список используемых материалов

1. Django введение - Изучение веб-разработки - MDN Web Docs, URL - <https://developer.mozilla.org/ru/docs/Learn_web_development/Extensions/Server-side/Django/Introduction> (Дата обращения: 23.09.2024)
2. Руководство по созданию веб-API с помощью ASP.NET Core, URL - <https://learn.microsoft.com/ru-ru/aspnet/core/tutorials/first-web-api?view=aspnetcore-9.0> (Дата обращения: 24.09.2024)
3. ASP.NET Core и C# | Полное руководство – Metanit, URL - <https://metanit.com/sharp/aspnet6/> (Дата обращения: 24.09.2024)
4. Руководство. Создание минимального API с помощью ASP.NET Core, URL - <https://learn.microsoft.com/ru-ru/aspnet/core/tutorials/min-web-api?view=aspnetcore-9.0> (Дата обращения: 24.09.2024)
5. Введение в Django: создание веб-приложений на Python – Skypro, URL - <https://sky.pro/wiki/python/vvedenie-v-django-sozdanie-veb-prilozhenij-na-python/> (Дата обращения: 23.09.2024)
6. Общие сведения об Entity Framework, URL - <https://learn.microsoft.com/ru-ru/dotnet/framework/data/adonet/ef/overview> (Дата обращения: 16.10.2024)
7. Официальная документация Django, URL - <https://docs.djangoproject.com/ru/stable/> (Дата обращения: 24.10.2024)
8. Документация по ASP.NET Core, URL - <https://learn.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-9.0> (Дата обращения: 24.10.2024)
9. Документация по Entity Framework, URL - <https://learn.microsoft.com/ru-ru/ef/> (Дата обращения: 16.10.2024)
10. HTML: HyperText Markup Language MDN, URL - <https://developer.mozilla.org/ru/docs/Web/HTML> (Дата обращения: 12.10.2024)
11. CSS: Каскадные таблицы стилей MDN, URL - <https://developer.mozilla.org/ru/docs/Web/CSS> (Дата обращения: 12.10.2024)
12. Интеграция Django с внешними API, URL - <https://sky.pro/wiki/python/integraciya-django-s-vneshnimi-api/> (Дата обращения: 20.11.2024)
13. Понимание архитектуры Django: паттерн MTV, URL - <https://dev.to/vincenttommi/2-understanding-djangos-architecture-the-mtv-pattern-1gl> (Дата обращения: 27.10.2024)
14. Архитектура в Django проектах, URL - <https://habr.com/ru/companies/vivid_money/articles/544856/> (Дата обращения: 27.10.2024)
15. Основы Django: Архитектура Model-View-Template (MVT), URL - <https://angelogentileiii.medium.com/basics-of-django-model-view-template-mvt-architecture-8585aecffbf6> (Дата обращения: 27.10.2024)
16. Архитектура REST, URL - <https://habr.com/ru/articles/38730/> (Дата обращения: 24.09.2024)
17. REST API: принципы, применение, URL - <https://gb.ru/blog/rest-api/> (Дата обращения: 25.09.2024)
18. Что такое RESTful API, URL - <https://aws.amazon.com/ru/what-is/restful-api/> (Дата обращения: 25.09.2024)
19. Что такое RESTful на самом деле, URL - <https://habr.com/ru/companies/hexlet/articles/274675/> (Дата обращения: 26.09.2024)
20. Профессиональная разработка технической документации – ГОСТ 34, URL - [https://www.swrit.ru/gost-34.html](https://www.swrit.ru/gost-34.html%20) (Дата обращения 25.11.2024)