МИНОБРНАУКИ РОССИИ

РГУ нефти и газа (НИУ) имени И.М. Губкина

|  |  |
| --- | --- |
| Факультет: | **Автоматики и вычислительной техники** |
| Кафедра: | **Автоматизированных систем управления** |
| Направление (специальность): | **09.03.01 Информатика и вычислительная техника** |
| Программа (специализация): | **09.03.01.01 «Автоматизированные системы обработки информации и управления»** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Оценка: |  | | | Рейтинг: |  |
| Подпись секретаря ГЭК: | | | | | |
|  | |  | Арбузова А. В. | | |
| (подпись) | |  | (фамилия, имя, отчество) | | |
|  | | | | | |
| (дата) | | | | | |

**ДИПЛОМНАЯ РАБОТА**

|  |  |
| --- | --- |
| на тему | Разработка интегрированного модуля взаимного |
| рецензирования для образовательного портала университета | |
|  | |

|  |  |  |  |
| --- | --- | --- | --- |
| РУКОВОДИТЕЛЬ: |  | ВЫПОЛНИЛ: |  |
|  |  | Студент группы | **АС-21-05** |
|  |  |  | (номер группы) |
| Шеляго Наталья Дмитриевна |  | Самородов Роман Алексеевич | |
| (фамилия, имя, отчество) |  | (фамилия, имя, отчество) | |
|  |  |  | |
| (подпись) |  | (подпись) | |
|  |  |  | |
| (дата) |  | (дата) | |
| КОНСУЛЬТАНТ ПО РАЗДЕЛУ: |  |  | |
|  |  |  | |
| (наименование раздела) |  |  | |
|  |  |  | |
| (должность, степень) |  |  | |
|  |  |  | |
| (фамилия, имя, отчество) |  |  | |
|  |  |  | |
| (подпись) |  |  | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Москва, 20 | 25 |  |

МИНОБРНАУКИ РОССИИ

РГУ нефти и газа (НИУ) имени И.М. Губкина

|  |  |
| --- | --- |
| Факультет | **Автоматики и вычислительной техники** |
| Кафедра | **Автоматизированных систем управления** |

**ЗАДАНИЕ НА ВЫПОЛНЕНИЕ**

**ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

|  |  |  |  |
| --- | --- | --- | --- |
| ДАНО студенту | **Самородову Роману Алексеевичу** | группы | **АС-21-05** |
|  | (фамилия, имя, отчество в дательном падеже) |  | (номер группы) |

|  |  |
| --- | --- |
| Тема ВКР: | Разработка интегрированного модуля взаимного |
| рецензирования для образовательного портала университета | |
|  | |

|  |  |
| --- | --- |
| Руководитель: | Шеляго Н.Д., старший преподаватель |
| (фамилия И.О., должность, ученая степень) | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Тема ВКР, руководитель и консультант(ы) (при наличии) закреплены приказом | | | | | | | | |
| № | 5078-у | от « | 11 | » | ноября | 20 | 24 | года |
| Закрепление изменено приказом (при наличии): | | | | | | | | |
| № |  | от « |  | » |  | 20 |  | года |

|  |  |  |  |
| --- | --- | --- | --- |
| **Раздел ВКР:** | | **Анализ предметной области** | |
|  | (наименование раздела) | | |
| Задание и исходные данные по разделу: | | | |
|  | Анализ подходов к взаимному рецензированию в образовании | |
|  | Анализ существующих инструментов взаимного рецензирования | |
|  |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Раздел ВКР:** | | **Проектирование модуля взаимного рецензирования** | |
|  | (наименование раздела) | | |
| Задание и исходные данные по разделу: | | | |
|  | Выбор инструментов для разработки модуля | |
|  | Проектирование модуля взаимного рецензирования | |
|  |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Раздел ВКР:** | | **Реализация программного модуля** | |
|  | (наименование раздела) | | |
| Задание и исходные данные по разделу: | | | |
|  | Реализация структуры базы данных для модуля | |
|  | Реализация основных функций модуля | |
|  | Реализация пользовательского интерфейса с использованием шаблонов | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Раздел ВКР:** | |  | | | | |
|  | (наименование раздела) | | | | | |
|  |  | | | | | |
| Консультант (при наличии): | | |  |  |  | |
|  | | | (фамилия И.О., должность, ученая степень) |  | подпись | |
| Задание и исходные данные по разделу: | | | | | | |
|  |  | | | | |
|  |  | | | | |
|  |  | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Раздел ВКР:** | |  | | | | |
|  | (наименование раздела) | | | | | |
|  |  | | | | | |
| Консультант (при наличии): | | |  |  |  | |
|  | | | (фамилия И.О., должность, ученая степень) |  | подпись | |
| Задание и исходные данные по разделу: | | | | | | |
|  |  | | | | |
|  |  | | | | |
|  |  | | | | |

**Рекомендуемая литература:**

|  |  |
| --- | --- |
|  | Харченко В.С. Взаимное рецензирование в учебных курсах вуза // Высшее образование в России. — 2017. — № 2. — С. 86–92. — URL: https://vovr.elpub.ru/jour/article/view/969/869. |
|  | Громаков Е.И., Сидорова А.А. Современные технологии. Киберфизические системы – Томск: Изд-во Томского политехнического университета, 2021. – 166 с. |
|  |  |
|  |  |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задание выдал: |  |  |  | Шеляго Н.Д. |
|  |  | (подпись руководителя) |  | (фамилия, имя, отчество) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Задание принял к исполнению: |  |  |  |  | Самородов Р.А. |
|  | |  | (подпись студента) |  | (фамилия, имя, отчество) |

**Аннотация**

Самородов Р.А. Разработка интегрированного модуля взаимного рецензирования для образовательного портала университета, дипломная работа, 2025 – 77 с., 1 таблица, 42 рисунка, 3 листинга кода, 3 приложения. Руководитель Шеляго Н.Д., старший преподаватель. Кафедра автоматизированных систем управления.

Разработан модуль для взаимного рецензирования студентов, интегрированный в образовательный портал университета. Проведено проектирование структуры базы данных. Реализована серверная часть модуля, обеспечивающая обработку данных и интеграцию с образовательным порталом. Разработан алгоритм выявления недобросовестных студентов по результатам рецензирования.

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 6](#_Toc200214946)

[1.1 Современные подходы к взаимному рецензированию в образовании 7](#_Toc200214947)

[1.2 Существующие инструменты взаимного рецензирования в образовательных платформах 9](#_Toc200214948)

[1.3 Психолого-педагогические аспекты взаимного рецензирования 13](#_Toc200214949)

[1.4 Анализ потребностей пользователей 14](#_Toc200214950)

[1.5 Опыт внедрения взаимного рецензирования в учебный процесс 16](#_Toc200214951)

[1.6 Анализ рисков и ограничений внедрения модуля 17](#_Toc200214952)

[1.7 Постановка задачи разработки модуля для Moodle 18](#_Toc200214953)

[1.8 Выводы по главе 1 20](#_Toc200214954)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ МОДУЛЯ ВЗАИМНОГО РЕЦЕНЗИРОВАНИЯ 21](#_Toc200214955)

[2.1 Выбор инструментов для разработки модуля 21](#_Toc200214956)

[2.2 Проектирование модуля взаимного рецензирования 32](#_Toc200214957)

[2.3 Выводы по главе 2 38](#_Toc200214958)

[ГЛАВА 3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО МОДУЛЯ 40](#_Toc200214959)

[3.1 Реализация структуры базы данных с использованием XMLDB 40](#_Toc200214960)

[3.2 Реализация основных функций модуля на PHP 43](#_Toc200214961)

[3.3 Реализация API Moodle для модуля взаимного рецензирования 48](#_Toc200214962)

[3.4 Реализация пользовательского интерфейса с использованием шаблонов Moodle 51](#_Toc200214963)

[3.5 Реализация системы выявления недобросовестных студентов 58](#_Toc200214964)

[3.6 Выводы по главе 3 62](#_Toc200214965)

[ЗАКЛЮЧЕНИЕ 63](#_Toc200214966)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 64](#_Toc200214967)

[ПРИЛОЖЕНИЯ 66](#_Toc200214968)

ВВЕДЕНИЕ

Современная система образования требует внедрения инновационных методов оценки, способствующих развитию критического мышления и самостоятельной работы студентов. Взаимное рецензирование представляет собой ценный педагогический инструмент, позволяющий студентам развивать аналитические навыки через оценку работ своих коллег. Актуальность исследования обусловлена необходимостью создания эффективных механизмов формирования у обучающихся компетенций критического анализа и конструктивной оценки.

Целью дипломной работы является разработка модуля взаимного рецензирования для системы Moodle, направленного на развитие у студентов навыков экспертной оценки через анализ работ сокурсников. Модуль призван создать условия для формирования профессионального подхода к оцениванию и развитию рефлексивных способностей обучающихся.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести исследование предметной области;
2. Определить функциональные требования к модулю и разработать его архитектуру;
3. Изучить технологии для реализации программного обеспечения.
4. Спроектировать структуру базы данных;
5. Реализовать серверную часть модуля;
6. Разработать пользовательские интерфейсы для:
   * студентов (загрузка работ, рецензирование);
   * преподавателей (управление процессом, мониторинг).ГЛАВА 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ
   1. Современные подходы к взаимному рецензированию в образовании

Эффективность взаимного рецензирования как педагогического инструмента во многом определяется качеством разработанных критериев оценивания. В современной образовательной практике сложились различные подходы к формированию таких критериев, каждый из которых имеет свою специфику и область применения.

Структурные критерии оценивания фокусируются на формальных аспектах работы. Они включают проверку соответствия установленным требованиям к объему работы, наличию всех необходимых структурных элементов, таких как введение, основная часть и заключение. Особое внимание уделяется правильности оформления библиографических ссылок и списка литературы, а также соблюдению академического стиля изложения. Эти критерии особенно важны на начальных этапах обучения, когда у студентов формируются базовые навыки академического письма.

Содержательные критерии направлены на оценку качества проработки темы и глубины анализа. Они предполагают проверку логичности и последовательности изложения материала, научной обоснованности выводов, степени самостоятельности выполнения работы. В гуманитарных дисциплинах дополнительно оценивается оригинальность интерпретации материала, а в технических - точность расчетов и корректность применения методик.

На практике наиболее эффективными оказываются комбинированные системы оценивания, которые объединяют структурные и содержательные критерии. Такой подход позволяет всесторонне оценить работу студента, учитывая как ее формальные характеристики, так и содержательную составляющую. Важным аспектом является адаптация критериев к специфике конкретной учебной дисциплины и уровня подготовки студентов.

Разработка четких и понятных критериев оценивания требует от преподавателя не только глубокого знания предмета, но и понимания психологических аспектов процесса оценивания. Хорошо разработанные критерии значительно повышают объективность взаимного рецензирования и способствуют более осознанному участию студентов в этом процессе [1].

Метод взаимного рецензирования занимает важное место в современной образовательной практике, предлагая уникальные возможности для развития профессиональных компетенций студентов.

С педагогической точки зрения, взаимное рецензирование способствует глубокому усвоению материала. Когда студенты анализируют работы своих коллег, они не только знакомятся с разными подходами к решению учебных задач, но и начинают лучше понимать критерии оценки качества работы. Этот процесс развивает критическое мышление и способность к саморефлексии, что является важными компонентами профессионального становления.

Когнитивные преимущества метода проявляются в развитии аналитических навыков. Студенты учатся выявлять как сильные стороны работ, так и типичные ошибки, что способствует более осознанному подходу к собственной учебной деятельности. Кроме того, участие в процессе рецензирования готовит студентов к будущей профессиональной деятельности, где часто требуется оценивать работы коллег и конструктивно воспринимать критику [1].

Однако метод имеет и определенные ограничения. Субъективность оценок, особенно среди студентов младших курсов, может снижать точность результатов. Некоторые учащиеся испытывают психологический дискомфорт при необходимости оценивать работы своих одногруппников, что может отражаться на качестве рецензий. Организационные сложности, связанные с необходимостью тщательного инструктажа и постоянного контроля со стороны преподавателя, также ограничивают широкое применение этого метода.

При использовании технологических решений многие из этих ограничений могут быть минимизированы. Автоматизация процессов распределения работ и проверки качества оценок позволяет снизить нагрузку на преподавателя и повысить объективность результатов. Это делает взаимное рецензирование ценным инструментом современного образовательного процесса, особенно в условиях массового высшего образования.

* 1. Существующие инструменты взаимного рецензирования в образовательных платформах

Современные цифровые образовательные среды предлагают широкий спектр инструментов для организации взаимного рецензирования, каждый из которых обладает уникальными характеристиками и подходит для различных педагогических сценариев. Анализ существующих решений позволяет выделить три основные категории: встроенные инструменты систем управления обучением (LMS), специализированные платформы для взаимного оценивания и универсальные средства совместной работы.

Рассматривая встроенные инструменты LMS, стоит обратить внимание на их функциональность и ограничения. Moodle, лидирующая платформа LMS в высшем образовании, предоставляет несколько механизмов для организации взаимного рецензирования. Наиболее продвинутым из них является модуль “Семинар” (Workshop), специально предназначенный для peer assessment (экспертной оценки). Этот инструмент позволяет преподавателям создавать сложные схемы оценивания, включающие многоэтапный процесс рецензирования (подготовка, отправка работ, оценка, калибровка), гибкую систему критериев оценки с различными типами шкал, возможность анонимного и именного рецензирования, алгоритмы распределения работ (случайное, ручное, групповое) и механизмы нормализации оценок.

Однако практическое использование модуля "Семинар" выявляет ряд существенных ограничений. Интерфейс настройки отличается сложностью и требует от преподавателя значительных временных затрат на освоение. Система не предусматривает автоматического выявления недобросовестных оценок, а процесс калибровки (согласования оценок между рецензентами) часто оказывается слишком громоздким для массовых курсов. Кроме того, модуль не поддерживает сложные аналитические отчеты, что затрудняет мониторинг процесса рецензирования в больших группах.

Использование форумов с элементами оценивания в Moodle предлагает гибкость и простоту, позволяя студентам публиковать работы и комментировать работы друг друга. Однако, этот подход имеет недостатки: отсутствие структурированной системы критериев оценки, невозможность автоматического распределения работ, проблемы с конфиденциальностью, поскольку все работы видны всем участникам, и сложности с подсчетом итоговых оценок.

Рассмотрим специализированные платформы и их расширенный функционал. Среди специализированных решений для взаимного рецензирования особого внимания заслуживают такие платформы как Peergrade, Kritik и Peerceptiv. Эти системы разрабатывались специально для организации peer assessment и предлагают значительно более продвинутый функционал по сравнению со встроенными инструментами LMS [2].

Peergrade выделяется интуитивным интерфейсом и мощными аналитическими возможностями. Система автоматически генерирует отчеты о качестве оценок, выявляя расхождения между рецензентами. Особенностью платформы является встроенная система "калибровки", где студенты сначала оценивают тренировочные работы, а затем получают обратную связь о соответствии своих оценок эталонным.

Kritik делает акцент на геймификации процесса рецензирования. Студенты получают баллы не только за качество своих работ, но и за полезность оставленных рецензий. Платформа использует сложные алгоритмы сопоставления текстовых комментариев с оценками, что позволяет автоматически определять наиболее качественные рецензии.

Peerceptiv (ранее известный как SWoRD) предлагает уникальную систему многоуровневого рецензирования, где каждая работа оценивается несколькими студентами, а затем проходит этап мета-рецензирования. Особенностью системы является адаптивный алгоритм распределения работ, учитывающий предыдущую активность студентов.

Несмотря на очевидные преимущества, специализированные платформы имеют ряд существенных недостатков. Основной проблемой является их оторванность от основной образовательной среды. Необходимость использования внешних сервисов создает дополнительные сложности как для преподавателей (дублирование списков студентов, ручной перенос оценок), так и для студентов (изучение нового интерфейса, дополнительные учетные записи). Кроме того, большинство таких решений являются коммерческими и требуют существенных финансовых затрат на внедрение.

В образовательной практике широко используются универсальные инструменты для совместной работы, такие как Google Docs, Microsoft 365 и Overleaf, для организации взаимного рецензирования. Эти платформы предоставляют инструменты для комментирования и совместного редактирования документов. Например, Google Docs позволяет оставлять комментарии к конкретным фрагментам текста, отслеживать историю изменений с возможностью сравнения версий, обеспечивает одновременную работу нескольких пользователей и позволяет настраивать права доступа для разных участников.

Однако использование таких инструментов для систематического peer assessment сталкивается с рядом проблем. Отсутствие структурированной системы критериев оценки приводит к тому, что рецензии носят произвольный характер. Невозможность автоматического распределения работ и сбора оценок существенно увеличивает нагрузку на преподавателя. Кроме того, такие платформы не интегрированы с системами управления обучением, что затрудняет учет результатов рецензирования в общей успеваемости студентов.

Для наглядного представления различий между рассмотренными инструментами целесообразно привести сравнительную таблицу 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Характеристика** | **Moodle Workshop** | **Moodle Forum** | **Peergrade** | **Google Docs** |
| Автоматическое распределение | Да | Нет | Да | Нет |
| Структурированные критерии | Да | Нет | Да | Нет |
| Анонимность | Да | Нет | Да | Нет |
| Анализ качества оценок | Ограниченный | Нет | Да | Нет |
| Интеграция с LMS | Полная | Полная | Нет | Нет |
| Стоимость использования | Бесплатно | Бесплатно | Платная | Бесплатно |

Таблица 1. Сравнительный анализ возможностей инструментов взаимного рецензирования

Проведенный анализ показывает, что существующие инструменты либо слишком ограничены в функционале, как встроенные средства LMS, либо требуют значительных дополнительных затрат, как специализированные платформы. Это подтверждает необходимость разработки специализированного модуля для Moodle, способного объединить преимущества разных решений. Такой модуль должен обеспечивать глубокую интеграцию с образовательной средой, предлагать расширенные возможности автоматизации и мощные аналитические инструменты. При этом важно сохранить гибкость в настройке критериев и обеспечить простой, интуитивно понятный интерфейс для пользователей.

Особое значение имеет реализация механизмов контроля качества оценок, которые отсутствуют в стандартных инструментах Moodle. Разрабатываемый модуль должен автоматически выявлять недобросовестных рецензентов и минимизировать субъективный фактор в итоговых оценках, что особенно важно для обеспечения объективности процесса взаимного рецензирования.

* 1. Психолого-педагогические аспекты взаимного рецензирования

Взаимное рецензирование представляет собой сложный педагогический инструмент, который оказывает значительное влияние на формирование профессиональных компетенций студентов. С психологической точки зрения, данный метод способствует развитию критического мышления, однако его эффективность во многом зависит от правильной организации процесса. Исследования в области образовательных технологий показывают, что студенты, активно участвующие в оценке работ своих коллег, демонстрируют более глубокое понимание предмета по сравнению с теми, кто работает исключительно под контролем преподавателя. Это связано с тем, что анализ чужих ошибок и достоинств требует осознанного применения полученных знаний, что способствует их лучшему усвоению.

Одним из ключевых факторов успешного внедрения взаимного рецензирования является мотивация студентов. С одной стороны, возможность выступать в роли эксперта повышает их вовлеченность в учебный процесс и чувство ответственности. С другой стороны, отсутствие четких инструкций и недостаточная подготовка могут привести к обратному эффекту — страху допустить ошибку или нежеланию критиковать одногруппников. Особенно это актуально в культурах, где открытая критика не поощряется. Для минимизации подобных рисков необходимо создавать комфортную среду, в которой обратная связь воспринимается как инструмент развития, а не как личная оценка. Важную роль здесь играет анонимность. Во многих случаях она позволяет снизить психологический дискомфорт и повысить объективность оценок.

Педагогическая ценность взаимного рецензирования также зависит от роли преподавателя. В отличие от традиционных форм контроля, где преподаватель выступает единственным арбитром, в данном случае его функция смещается в сторону модерации и поддержки. Он должен не только разрабатывать четкие критерии оценивания, но и обучать студентов принципам конструктивной обратной связи. Практика показывает, что наиболее эффективным является постепенное внедрение метода: от простых заданий, таких как проверка структуры работы или оформления, к более сложным — анализу содержания, аргументации и оригинальности идей. Такой подход позволяет студентам поэтапно развивать навыки критического анализа, что в конечном итоге приводит к повышению качества их собственных работ [2].

* 1. Анализ потребностей пользователей

Разработка эффективного модуля взаимного рецензирования требует глубокого понимания потребностей всех участников образовательного процесса. Современные исследования в области педагогического дизайна показывают, что успешная реализация peer assessment возможна только при условии тщательного учета особенностей работы как преподавателей, так и студентов на всех этапах взаимодействия с системой.

Для преподавательского состава критически важным оказывается баланс между гибкостью настройки и простотой использования инструмента. В условиях современного образовательного ландшафта, где сосуществуют принципиально разные подходы к оцениванию в различных дисциплинах, система должна позволять адаптировать процесс рецензирования под конкретные учебные задачи. В технических направлениях, таких как программирование или инженерное проектирование, особое значение приобретают специализированные инструменты для проверки кода на соответствие стандартам и оценки эффективности алгоритмов, тогда как в гуманитарных науках на первый план выходят возможности анализа аргументации и оригинальности концепции.

Не менее важной для преподавателей является комплексная система мониторинга, которая должна предоставлять не просто данные о процессе рецензирования, но и содержать инструменты для их осмысленного анализа. Речь идет о визуализации динамики оценивания, автоматическом выявлении статистических аномалий и механизмах оперативного реагирования на проблемные ситуации. Особую сложность представляет проектирование системы распределения работ, которая должна учитывать множество факторов — от уровня подготовки студентов до специфики учебного задания, сохраняя при этом простоту настройки для преподавателя.

Со стороны студентов ключевым требованием становится прозрачность и понятность всего процесса взаимного оценивания. Многочисленные исследования подтверждают [2], что успешность внедрения peer assessment напрямую зависит от того, насколько четко студенты понимают как критерии оценки, так и механизмы формирования итоговых результатов. Система должна не просто предоставлять возможность оставлять рецензии, но и помогать студентам развивать навыки конструктивной обратной связи через примеры качественных оценок, подсказки по структуре отзыва и проверку полноты рецензии перед отправкой.

Особого внимания заслуживает психологический аспект взаимодействия студентов с системой. Интерфейс должен минимизировать стресс и неопределенность, которые неизбежно возникают при необходимости оценивать работы своих коллег и получать критические замечания в свой адрес. Это достигается за счет продуманной визуализации процесса, четкого разделения этапов работы и возможности постепенного освоения навыков рецензирования через тренировочные задания.

Важнейшей характеристикой успешной системы является ее способность поддерживать баланс между стандартизацией и гибкостью. С одной стороны, необходимо обеспечить единообразие процесса оценивания для всех участников, с другой — сохранить возможность адаптации под конкретные учебные ситуации. Это особенно актуально для междисциплинарных курсов, где могут одновременно применяться разные подходы к peer assessment.

Интеграционные возможности системы также играют ключевую роль в ее практическом применении. Глубокая связь с образовательной экосистемой Moodle, поддержка современных стандартов обмена данными и возможность подключения дополнительных инструментов аналитики существенно расширяют сферу применения модуля, делая его полезным как для небольших специализированных курсов, так и для массовых образовательных программ.

Особенностью современных систем взаимного рецензирования становится их способность не просто автоматизировать процесс оценивания, но и способствовать развитию когнитивных навыков студентов. Через анализ работ своих коллег и рефлексию полученных замечаний учащиеся развивают критическое мышление, учатся аргументировать свою точку зрения и более осознанно подходить к собственной учебной деятельности. Таким образом, хорошо спроектированная система peer assessment выходит за рамки инструмента оценки и становится важным элементом образовательного процесса.

* 1. Опыт внедрения взаимного рецензирования в учебный процесс

Практика применения взаимного рецензирования в высшем образовании демонстрирует как его потенциал, так и ограничения, которые необходимо учитывать при разработке модуля для Moodle. В технических дисциплинах, таких как программирование или инженерное проектирование, этот метод успешно используется для оценки корректности решений и эффективности предложенных подходов. Однако в творческих заданиях, где нет однозначных критериев оценки, субъективность может стать серьезной проблемой.

В гуманитарных науках взаимное рецензирование чаще всего применяется для развития навыков критического анализа текстов. В таких дисциплинах, как филология или история, студенты учатся оценивать аргументацию, работу с источниками и логику изложения. Однако здесь возникает другая сложность: без должной подготовки рецензии могут сводиться к поверхностным комментариям, не несущим реальной пользы.

Особый интерес представляет опыт массовых открытых онлайн-курсов (МООК), где взаимное рецензирование зачастую является единственным способом оценки работ тысяч студентов. Платформы вроде Coursera или edX используют сложные алгоритмы для распределения заданий и проверки качества оценок, что делает их ценным источником идей для разработки аналогичных решений в Moodle. Ключевым выводом из этого опыта является необходимость обеспечения масштабируемости модуля. Он должен оставаться стабильным и удобным как в небольших группах, так и в условиях массового обучения.

* 1. Анализ рисков и ограничений внедрения модуля

Внедрение модуля взаимного рецензирования в образовательный процесс представляет собой комплексную задачу, требующую учета многочисленных факторов технического, педагогического и организационного характера. При этом каждый из аспектов внедрения сопряжен с определенными рисками, которые могут существенно повлиять на эффективность всей системы.

С технической стороны помимо проблем с нагрузкой на сервер и совместимостью, важно учитывать вопросы безопасности данных и защиты конфиденциальности. В условиях ужесточения требований к обработке персональных данных система должна обеспечивать надежное хранение студенческих работ и рецензий, а также контролируемый доступ к этой информации. Особую сложность представляет реализация различных режимов анонимности - от полной конфиденциальности до частичного раскрытия информации о рецензентах, что требует тщательной проработки архитектуры системы.

Педагогические риски не ограничиваются проблемой качества рецензий. Серьезным вызовом является обеспечение равномерного распределения учебной нагрузки среди студентов. На практике часто возникает ситуация, когда более ответственные учащиеся получают непропорционально большое количество работ для оценки, в то время как другие практически не участвуют в процессе. Это не только создает несправедливую нагрузку, но и может привести к снижению мотивации наиболее активных студентов. Решение данной проблемы требует разработки интеллектуальных алгоритмов распределения, учитывающих предыдущую активность участников.

Важным аспектом является также интеграция системы взаимного рецензирования в существующую образовательную экосистему университета. Модуль должен не просто технически работать в Moodle, но и органично вписываться в утвержденные учебные процессы, регламенты оценивания и систему контроля качества образования.

Опыт передовых образовательных учреждений в России показывает, что успешное внедрение подобных систем требует формирования целостной организационной культуры взаимного оценивания [3]. Это включает в себя не только технические решения, но и регулярное обучение всех участников процесса, создание системы мотивации для качественного рецензирования, а также постоянный мониторинг и совершенствование практик использования инструмента. Только комплексный подход, учитывающий все перечисленные аспекты, может обеспечить устойчивое и эффективное применение технологии взаимного рецензирования в образовательном процессе.

* 1. Постановка задачи разработки модуля для Moodle

Проведенный анализ современных подходов к организации взаимного рецензирования в образовательном процессе и существующих технологических решений позволяет сформулировать комплекс задач, решение которых необходимо для разработки эффективного модуля взаимного оценивания в системе Moodle.

Современные образовательные реалии демонстрируют возрастающую потребность в инструментах, способных не только автоматизировать процесс взаимного оценивания, но и обеспечить его качество, объективность и педагогическую эффективность. Разрабатываемый модуль должен стать органичной частью образовательной экосистемы Moodle, предоставляя преподавателям и студентам удобный и функциональный инструмент для организации peer review (взаимное рецензирование).

Ключевой задачей проектирования является создание архитектуры модуля, которая бы органично вписывалась в существующую инфраструктуру Moodle. Это предполагает глубокую интеграцию с системой управления пользователями, курсами и оценками, что обеспечит единообразие работы и простоту администрирования. Особое внимание необходимо уделить проектированию механизма хранения данных, который должен эффективно работать с информацией о заданиях, отправленных работах, рецензиях и оценках, обеспечивая при этом необходимую производительность даже при работе с большими группами студентов.

Важнейшим аспектом разработки является создание полноценного функционала для преподавателей. В отличие от существующих решений, новый модуль должен предоставлять расширенные возможности по настройке параметров заданий - от определения типов принимаемых работ (текстовые ответы или файлы) до установки индивидуальных критериев оценивания для разных типов заданий. Система мониторинга должна давать преподавателю полную картину процесса рецензирования - от общего прогресса до детальной статистики по каждому студенту, позволяя своевременно вмешиваться в процесс при необходимости.

Для студентов модуль должен предложить интуитивно понятный и удобный интерфейс, максимально упрощающий процесс как загрузки собственных работ, так и их рецензирования. Особое значение имеет продуманная система отображения полученных рецензий и оценок, которая не просто показывала бы конечный результат, но и помогала студентам анализировать замечания и совершенствовать свои работы.

Особенностью разрабатываемого решения станет реализация сложных алгоритмов анализа качества оценок. В отличие от простых систем усреднения, предлагаемый модуль будет использовать комплекс математических методов, включая расчет стандартных отклонений, выявление статистических выбросов и сравнение с медианными значениями. Это позволит автоматически выявлять случаи недобросовестного оценивания и минимизировать субъективный фактор в итоговых оценках.

Масштабируемость и гибкость системы - еще одна важная задача разработки. Модуль должен одинаково эффективно работать как в небольших группах, так и в массовых курсах, поддерживая различные типы учебных заданий - от небольших эссе до комплексных проектных работ. При этом система должна предлагать широкие возможности настройки под конкретные образовательные задачи и дисциплинарные особенности.

Отдельное внимание в процессе разработки будет уделено вопросам производительности и стабильности работы модуля. Учитывая, что процессы взаимного рецензирования часто связаны с пиковыми нагрузками (например, при сроках сдачи заданий), архитектура системы должна обеспечивать бесперебойную работу даже в условиях высокой одновременной активности пользователей.

Перспективы развития модуля включают возможность интеграции с системами антиплагиата, поддержку мобильных устройств, а также развитие инструментов аналитики для более глубокого анализа результатов взаимного оценивания. Все это делает разработку не просто техническим заданием, а значимым вкладом в развитие цифровых образовательных технологий.

* 1. Выводы по главе 1

Таким образом, поставленная задача разработки модуля взаимного рецензирования для Moodle представляет собой комплексную проблему, требующую решения как технических, так и педагогических вопросов. Успешная реализация проекта позволит создать инструмент, который не только автоматизирует процесс peer review, но и значительно повысит его эффективность как образовательной технологии.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ МОДУЛЯ ВЗАИМНОГО РЕЦЕНЗИРОВАНИЯ

2.1 Выбор инструментов для разработки модуля

**Классификация модулей Moodle и выбор архитектуры**

Система управления обучением Moodle обладает гибкой модульной архитектурой, позволяющей расширять её функциональность за счёт разработки дополнительных компонентов. В контексте создания модуля взаимного рецензирования необходимо рассмотреть существующие типы модулей Moodle и обосновать выбор наиболее подходящего варианта. Moodle поддерживает несколько видов модулей, каждый из которых предназначен для решения определённого круга задач.

Активностные модули (activity modules) представляют собой один из основных типов расширений в Moodle. Они интегрируются непосредственно в учебные курсы и позволяют организовывать различные виды учебной деятельности. Характерной особенностью таких модулей является их тесная интеграция с системой оценивания и журналом успеваемости. В отличие от них, блоки (block modules) предназначены в первую очередь для отображения информации и не предполагают сложного взаимодействия с пользователем. Отчётные модули (report modules) ориентированы на анализ данных и формирование статистики, а локальные плагины (local plugins) расширяют общесистемную функциональность [4].

Для реализации функционала взаимного рецензирования наиболее подходящим вариантом является создание именно активностного модуля. Такой выбор обусловлен несколькими ключевыми факторами. Во-первых, процесс взаимного рецензирования по своей сути представляет собой учебную активность, которая должна быть встроена в структуру курса наравне с другими заданиями. Во-вторых, активностные модули имеют доступ ко всем необходимым API Moodle для работы с пользовательскими данными, системами оценивания и файловыми хранилищами. В-третьих, такой подход обеспечивает единообразие пользовательского интерфейса и привычный для преподавателей и студентов способ взаимодействия с элементом курса.

Архитектура разрабатываемого модуля строится по трёхуровневой модели, включающей уровень данных, уровень логики и уровень представления. На уровне данных используется встроенная в Moodle система XMLDB, которая предоставляет удобные средства для описания структуры базы данных и её миграции между различными версиями системы. Этот уровень отвечает за хранение всех необходимых для работы модуля данных: параметров заданий, отправленных студенческих работ, рецензий и оценок. XMLDB был выбран как стандартное решение для Moodle, обеспечивающее совместимость с различными СУБД и упрощающее процесс развёртывания модуля.

Уровень логики реализуется на языке PHP с активным использованием API Moodle. Именно здесь сосредоточена основная функциональность модуля: алгоритмы распределения работ между студентами, механизмы проверки качества рецензирования, расчёт итоговых оценок. Особое внимание уделено реализации алгоритма выявления недобросовестных оценок, который основывается на анализе стандартного отклонения и методах обнаружения статистических выбросов. Использование стандартных API Moodle обеспечивает интеграцию с системой ролей, журналом оценок и другими компонентами платформы.

Уровень представления отвечает за взаимодействие с пользователями. Для преподавателей разрабатывается интерфейс создания и настройки заданий взаимного рецензирования, а также инструменты мониторинга процесса оценивания. Студенты получают интуитивно понятный интерфейс для загрузки своих работ, просмотра назначенных для рецензирования работ других студентов и выставления оценок. При разработке пользовательского интерфейса активно используются стандартные шаблоны Moodle, что обеспечивает визуальную и функциональную согласованность с остальными элементами системы.

Выбор именно такой архитектуры обусловлен требованиями к масштабируемости, производительности и удобству сопровождения модуля. Трёхуровневая структура позволяет разделить ответственность между различными компонентами системы: уровень данных отвечает за целостность и доступность информации, уровень логики реализует основные алгоритмы работы, а уровень представления обеспечивает удобное взаимодействие с пользователем. Такой подход также упрощает процесс дальнейшего развития модуля, так как изменения на одном уровне минимально затрагивают другие уровни системы.

Интеграция с существующей инфраструктурой Moodle достигается за счёт использования стандартных механизмов расширения системы. Модуль регистрируется в Moodle как активность, подключается к системе событий для обработки различных сценариев (например, истечения сроков сдачи работ) и интегрируется с подсистемой оценивания. Это обеспечивает его беспроблемное сосуществование с другими компонентами платформы и возможность использования в различных учебных курсах без необходимости дополнительной настройки.

Таким образом, выбор активностного модуля в сочетании с трёхуровневой архитектурой представляет собой оптимальное решение для реализации функционала взаимного рецензирования в Moodlе. Такой подход обеспечивает необходимую функциональность, хорошую производительность и простоту интеграции с существующей образовательной платформой, что подтверждается анализом требований и существующих аналогов.

**Обзор Moodle API для разработки модулей**

Разработка функционального модуля для Moodle требует глубокого понимания API платформы, которое предоставляет разработчикам систематизированный набор инструментов для интеграции с ядром системы. В контексте создания модуля взаимного рецензирования особое значение приобретают следующие группы API:

API модулей активности образует фундамент для разработки нового типа задания. Этот программный интерфейс определяет обязательные методы, которые должны быть реализованы в модуле: инициализация задания, обработка отправки работ, управление оценками и формирование пользовательского интерфейса. Особенностью работы с этим API является необходимость строгого следования жизненному циклу активности в Moodle, включая обработку событий создания, обновления и удаления заданий. Для модуля взаимного рецензирования критически важна интеграция с подсистемой оценивания, что обеспечивается реализацией интерфейсов grading API.

File API играет ключевую роль в реализации функционала загрузки студенческих работ. Этот компонент предоставляет унифицированные методы для работы с файловым хранилищем Moodle, включая механизмы контроля доступа, квот и версионности. В модуле рецензирования File API используется не только для хранения отправленных работ, но и для управления прикрепленными файлами в рецензиях. Важной особенностью является поддержка drag-and-drop загрузки через интеграцию с API платформы [5].

Events API позволяет модулю реагировать на системные события и инициировать собственные. Для функционала взаимного рецензирования существенны события, связанные с истечением сроков сдачи работ, изменением статусов рецензий и системными уведомлениями. Реализация подписки на события обеспечивает декларативную обработку сложных сценариев, таких как автоматическое напоминание студентам о необходимости завершить рецензирование.

Особого внимания заслуживает Groups API, который обеспечивает корректную работу модуля в курсах с групповым режимом. Этот интерфейс позволяет реализовать различные стратегии распределения работ: внутри групп, между группами или общекурсовое рецензирование. Интеграция с Groups API гарантирует соблюдение правил видимости и доступности работ в соответствии с настройками курса.

Для реализации алгоритмов распределения работ используется User API, предоставляющий доступ к спискам участников курса с учетом их ролей и статусов. Этот же API применяется при проверке прав доступа к различным функциям модуля. Важной особенностью является поддержка pagination при работе с большими списками пользователей, что обеспечивает стабильную производительность в масштабных курсах [6].

Gradebook API обеспечивает интеграцию с журналом оценок Moodle, позволяя автоматически создавать элементы оценивания, передавать итоговые оценки, формировать сложные расчетные формулы (включая учет самооценки) и визуализировать результаты в стандартных отчетах.

Для построения пользовательского интерфейса задействован Output API, который включает систему шаблонов Mustache, механизмы локализации, стандартные UI-компоненты (формы, таблицы, пагинацию) и обеспечивает seamless-интеграцию с темой оформления.

Безопасность модуля реализована через Security API, предоставляющий инструменты для санации входных данных, CSRF-защиты форм, валидации прав доступа и безопасной работы с сессиями.

Асинхронная обработка ресурсоемких задач (анализ рецензий, массовые уведомления) вынесена на Task API, что особенно критично для курсов с высокой нагрузкой.

Интеграция перечисленных API в единую систему требует глубокой проработки архитектурных решений, включая оптимизацию последовательности вызовов, обработку ошибок, обеспечение производительности и поддержку кросс-версионной совместимости с различными релизами Moodle.

Такой комплексный подход к использованию API Moodle позволяет создать модуль, который не просто добавляет новую функциональность, но и органично вписывается в экосистему платформы, обеспечивая согласованную работу всех компонентов системы [7].

**Локальная среда разработки на базе XAMPP для модуля взаимного рецензирования**

Разработка модуля для системы дистанционного обучения требует тщательной подготовки рабочего окружения, которое должно обеспечивать стабильную работу всех компонентов платформы Moodle и при этом предоставлять разработчику удобные инструменты для тестирования и отладки. В качестве основы для локальной среды разработки был выбран пакет XAMPP – комплексное решение, включающее веб-сервер Apache, систему управления базами данных MySQL и интерпретатор PHP в едином устанавливаемом пакете. Этот выбор обусловлен несколькими ключевыми факторами, главным из которых является полная совместимость компонентов XAMPP с системными требованиями современной версии Moodle.

Настройка локального окружения начинается с установки стабильной версии XAMPP, соответствующей требованиям целевой версии Moodle. Для разработки модуля взаимного рецензирования использовалась версия XAMPP с PHP 8.2, MySQL 10.4 и Apache 2.4, что полностью покрывает минимальные требования платформы. После базовой установки выполняется тонкая настройка конфигурационных файлов всех компонентов. В файле php.ini увеличиваются лимиты выполнения скриптов и размер загружаемых файлов, что особенно важно для образовательного портала, где возможна работа с объемными материалами. Параметры MySQL оптимизируются для обеспечения стабильной работы с потенциально большими объемами данных – увеличивается размер буферного пула, настраивается кэширование запросов, что критически важно при тестировании функционала взаимного рецензирования в условиях, приближенных к реальной эксплуатации.

Установка самой платформы Moodle в подготовленное окружение представляет собой многоэтапный процесс, включающий создание базы данных, настройку прав доступа и пошаговую установку через веб-интерфейс. Особое внимание уделяется конфигурационным параметрам, влияющим на процесс разработки – активируется режим отладки с максимальным уровнем детализации, включается вывод всех ошибок PHP, настраивается система логирования SQL-запросов. Эти меры позволяют оперативно выявлять и устранять потенциальные проблемы в работе разрабатываемого модуля.

Для эффективной работы над проектом в локальной среде создается комплекс тестовых данных, имитирующих реальные условия эксплуатации модуля. Формируются тестовые курсы с различными настройками и структурой, создаются учетные записи пользователей с разными ролями – от администраторов и преподавателей до обычных студентов. Такой подход позволяет сразу тестировать функционал модуля в условиях, максимально приближенных к реальным, выявляя потенциальные проблемы с правами доступа или специфические сценарии использования.

Интеграция системы контроля версий Git в процесс разработки является неотъемлемой частью настройки рабочего окружения. Локальная среда конфигурируется для удобной работы с репозиторием – настраивается .gitignore для исключения из контроля версий временных и конфиденциальных файлов, устанавливаются клиенты для визуализации истории изменений. Это позволяет поддерживать порядок в кодовой базе и упрощает командную разработку, если над проектом работает несколько человек.

Производительность локальной среды тщательно оптимизируется – настраивается OPcache для ускорения работы PHP-скриптов, конфигурируется кэширование статических ресурсов на уровне веб-сервера, оптимизируются параметры работы СУБД. Эти меры особенно важны при тестировании модуля с большими объемами данных, когда в курсе может быть несколько сотен студентов, каждый из которых загружает работы и пишет рецензии.

Для всестороннего тестирования пользовательского интерфейса модуля локальная среда настраивается для доступа с различных устройств в локальной сети. Это требует дополнительной конфигурации файервола, настройки виртуальных хостов и создания SSL-сертификатов для обеспечения безопасного соединения по HTTPS. Такой подход позволяет тестировать интерфейс модуля на разных устройствах и в различных браузерах, что критически важно для обеспечения кроссплатформенной совместимости.

Инструменты отладки и профилирования составляют важную часть настроенной среды разработки. XDebug конфигурируется для интеграции с выбранной IDE, что позволяет проводить пошаговую отладку кода. Настраивается система логирования операций модуля, что помогает отслеживать сложные сценарии взаимодействия пользователей с системой. Браузерные инструменты разработчика используются для анализа и оптимизации клиентской части модуля.

Полноценно настроенная локальная среда на базе XAMPP предоставляет разработчику все необходимые инструменты для эффективной работы над модулем взаимного рецензирования. Она позволяет не только разрабатывать новый функционал, но и комплексно тестировать его в условиях, приближенных к реальной эксплуатации, оперативно выявлять и устранять потенциальные проблемы, а также экспериментировать с различными вариантами реализации тех или иных функций. Возможность работы в офлайн-режиме и полный контроль над всеми компонентами системы значительно ускоряют процесс разработки по сравнению с использованием удаленных тестовых стендов.

**Интегрированная среда разработки Visual Studio**

Для разработки модуля взаимного рецензирования была выбрана интегрированная среда разработки Microsoft Visual Studio, которая предоставляет комплексный набор инструментов для эффективной работы с PHP-проектами. Данная среда обладает значительными преимуществами при создании сложных модулей для систем дистанционного обучения, таких как Moodle.

Visual Studio обеспечивает глубокую интеграцию со всеми компонентами разрабатываемого модуля благодаря своей гибкой архитектуре. Среда предлагает продвинутые возможности анализа кода, учитывающие специфику работы с Moodle API. Интеллектуальная система подсказок и автодополнения значительно ускоряет процесс написания кода, особенно при работе с обширными библиотеками платформы.

Особое внимание уделено отладочным возможностям среды. Интеграция с XDebug позволяет проводить детальный анализ работы модуля в реальных условиях. Разработчик может отслеживать выполнение кода шаг за шагом, анализировать состояние переменных в любой момент времени и исследовать стек вызовов при сложных сценариях взаимодействия компонентов системы.

Среда разработки обеспечивает удобную работу с различными аспектами проекта - от PHP-кода логики до XML-описаний структуры базы данных. Встроенные инструменты навигации по проекту позволяют быстро находить нужные фрагменты кода даже в больших модулях. Поддержка шаблонов Moodle и специализированных форматов данных упрощает работу с пользовательским интерфейсом.

Интеграция с системой контроля версий Git реализована на глубоком уровне, что позволяет комфортно работать с историей изменений и взаимодействовать с другими разработчиками при необходимости. Визуализация изменений и инструменты сравнения версий кода помогают поддерживать порядок в проекте.

Производительность среды разработки оптимизирована для работы с большими проектами. Эффективное управление ресурсами позволяет комфортно работать даже при одновременной обработке множества файлов проекта. Возможность настройки рабочих пространств под конкретные задачи разработки повышает эффективность работы.

Таким образом, использование Visual Studio в качестве основной среды разработки создает оптимальные условия для реализации сложных образовательных модулей, сочетая в себе мощные инструменты анализа кода, удобные средства отладки и гибкие возможности настройки под конкретные задачи проекта [8].

**PHP: особенности разработки модулей Moodle**

Разработка модулей для Moodle на языке PHP имеет ряд существенных особенностей, обусловленных архитектурными решениями самой платформы и спецификой образовательной среды. Система Moodle предъявляет строгие требования к структуре и организации кода, что необходимо учитывать при создании модуля взаимного рецензирования.

Основой разработки является строгое соблюдение coding style Moodle, который включает правила именования переменных и функций, оформления кода, комментирования. Этот стандарт обеспечивает единообразие кодовой басы и упрощает совместную разработку. Особое внимание уделяется безопасности - все внешние данные обязательно проходят валидацию через встроенные функции Moodle, такие как optional\_param() и required\_param(), что защищает от SQL-инъекций и XSS-атак.

Архитектура модуля строится вокруг концепции плагинов Moodle. Каждый модуль представляет собой самостоятельный компонент с четкой структурой директорий, где db содержит файлы для работы с базой данных, lib - основную логику, а templates - шаблоны отображения. Такая организация обеспечивает модульность и упрощает сопровождение кода.

Работа с базой данных осуществляется исключительно через Moodle Database API, который предоставляет абстракцию над различными СУБД. Это означает, что вместо прямых SQL-запросов используются методы класса moodle\_database, такие как get\_records(), insert\_record() и execute(). Такой подход гарантирует совместимость с разными версиями Moodle и различными системами управления базами данных.

Система событий (Events API) играет ключевую роль в архитектуре модуля. Вместо прямого вызова функций используются триггеры событий и их обработчики, что делает модуль более гибким и расширяемым. Например, отправка уведомлений при завершении рецензирования реализуется через подписку на соответствующие события.

Особенностью разработки является необходимость интеграции с подсистемой ролей и прав доступа Moodle. Модуль должен корректно работать в различных контекстах (курс, модуль, система) и учитывать права пользователей. Для этого используются capabilities - специальные метки прав доступа, которые объявляются в файле access.php.

Локализация реализуется через стандартную систему строк Moodle. Все текстовые сообщения выносятся в отдельные языковые файлы в директории lang, что позволяет легко адаптировать модуль для разных языков без изменения основного кода.

Производительность кода оптимизируется за счет использования кэширования через Cache API Moodle. Особенно это важно для операций, требующих сложных вычислений или частых запросов к базе данных, таких как расчет статистики по оценкам или распределение работ между студентами.

Отдельного внимания заслуживает работа с файлами через File API Moodle. Все загружаемые студентами работы и прикрепляемые к рецензиям файлы хранятся в специальном файловом хранилище системы с соблюдением правил доступа и квот.

Разработка пользовательского интерфейса ведется с использованием Output API, который обеспечивает совместимость с различными темами оформления Moodle. Вместо прямого вывода HTML применяются шаблоны и рендереры, что гарантирует единообразие интерфейса.

Таким образом, разработка модуля Moodle на PHP требует не только знания самого языка, но и глубокого понимания архитектуры платформы, ее API и философии построения расширений. Это обеспечивает создание надежного, безопасного и совместимого с различными версиями Moodle решения для организации взаимного рецензирования в образовательном процессе.

2.2 Проектирование модуля взаимного рецензирования

**Организация данных в модуле Moodle (MySQL/XMLDB)**

При разработке модуля взаимного рецензирования для Moodle особое внимание необходимо уделить вопросам организации хранения и обработки данных. Moodle как система управления обучением предлагает разработчикам несколько принципиально разных подходов к работе с данными, каждый из которых имеет свои особенности, преимущества и ограничения. Рассмотрение этих подходов и обоснование выбора конкретной технологии представляет собой важный этап проектирования модуля.

Основным механизмом работы с данными в Moodle является использование встроенной подсистемы XMLDB. Данная технология была специально разработана для обеспечения переносимости модулей между различными установками Moodle и разными версиями системы. XMLDB представляет собой абстрактный слой, который позволяет описывать структуру базы данных в XML-формате, после чего система автоматически генерирует необходимые SQL-запросы для создания соответствующих таблиц в конкретной СУБД. Такой подход обеспечивает независимость модуля от типа используемой системы управления базами данных (MySQL, PostgreSQL, MariaDB и др.), что значительно упрощает процесс развертывания модуля в различных образовательных учреждениях.

Принцип работы с XMLDB в Moodle предполагает создание специального файла install.xml, в котором декларативно описывается структура всех таблиц, необходимых для работы модуля. Этот файл включает определения полей, их типов, индексов и связей между таблицами. Во время установки модуля система автоматически анализирует этот файл и создает соответствующие таблицы в базе данных. Важным преимуществом такого подхода является возможность автоматического обновления структуры базы данных при обновлении модуля - разработчик может внести изменения в install.xml, и система корректно выполнит миграцию данных при обновлении.

Альтернативным подходом является прямое использование возможностей СУБД (в большинстве случаев MySQL) через стандартный API Moodle. Такой метод предоставляет разработчику большую гибкость в создании сложных запросов и оптимизации производительности, однако требует более глубоких знаний конкретной СУБД и усложняет процесс переноса модуля между разными системами. Прямая работа с MySQL может быть оправдана в случаях, когда требуется реализация сложных запросов, которые трудно или невозможно выразить средствами XMLDB, либо когда критически важна максимальная производительность при работе с большими объемами данных.

В контексте разработки модуля взаимного рецензирования выбор был сделан в пользу XMLDB по нескольким причинам. Во-первых, данный подход полностью соответствует философии разработки расширений для Moodle, предполагающей максимальную переносимость и простоту установки. Во-вторых, функциональности XMLDB вполне достаточно для реализации всех требуемых операций с данными: хранения параметров заданий, студенческих работ, рецензий и оценок. В-третьих, использование XMLDB значительно упрощает процесс сопровождения и обновления модуля в будущем, так как все изменения структуры базы данных могут быть выполнены централизованно через модификацию install.xml.

Структура базы данных для модуля взаимного рецензирования включает несколько взаимосвязанных таблиц. Основная таблица содержит параметры заданий: название, описание, сроки сдачи работ и рецензий, количество требуемых рецензий для каждой работы, используемую шкалу оценивания. Отдельная таблица предназначена для хранения самих работ, отправленных студентами, с указанием автора, времени отправки и ссылки на файл (или текстового содержимого). Таблица рецензий хранит оценки и комментарии, выставленные студентами, с указанием рецензента и рецензируемой работы. Дополнительные таблицы могут содержать критерии оценивания, историю изменений и другую служебную информацию.

Важным аспектом проектирования структуры базы данных является обеспечение целостности данных и согласованности при выполнении операций. Для этого активно используются механизмы внешних ключей, которые поддерживаются XMLDB. Например, при удалении задания автоматически должны удаляться все связанные с ним работы и рецензии, что обеспечивается соответствующими ограничениями целостности. Также предусмотрены индексы для ускорения наиболее частых операций, таких как поиск всех работ конкретного студента или всех рецензий для определенной работы.

Отдельное внимание уделено вопросам безопасности при работе с данными. Все операции с базой данных выполняются через стандартный API Moodle, который обеспечивает защиту от SQL-инъекций за счет использования параметризованных запросов. Доступ к данным контролируется системой ролей и прав Moodle, что гарантирует, что студенты смогут видеть только те данные, которые им предназначены, а преподаватели получат полный доступ ко всем данным в рамках своих курсов.

Производительность системы при работе с данными обеспечивается несколькими способами. Во-первых, тщательно проектируются индексы для таблиц, чтобы ускорить выполнение наиболее частых запросов. Во-вторых, для сложных операций аналитики (например, расчета статистики по оценкам) могут использоваться предварительно агрегированные данные. В-третьих, реализовано кэширование часто запрашиваемых данных с использованием механизмов кэширования Moodle. Эти меры особенно важны для курсов с большим количеством участников, где одновременно могут обрабатываться сотни работ и рецензий [4].

Таким образом, выбор XMLDB в качестве основы для работы с данными в модуле взаимного рецензирования представляется оптимальным решением, которое сочетает в себе достаточную функциональность, простоту сопровождения и соответствие стандартам разработки для платформы Moodle. Такой подход обеспечивает надежное хранение данных, удобство администрирования и возможность дальнейшего расширения функциональности модуля без необходимости кардинальных изменений в структуре базы данных.

**Анализ требований программного модуля**

Разработка модуля взаимного рецензирования для образовательного портала университета требует тщательного анализа функциональных и нефункциональных требований, которые определяют архитектуру и ключевые характеристики системы. Требования формировались на основе изучения потребностей учебного процесса и анализа существующих аналогов в других системах дистанционного обучения.

Функциональные требования включают базовый набор возможностей, необходимых для организации процесса взаимного оценивания студенческих работ. Модуль должен предоставлять преподавателю интерфейс для создания заданий с возможностью указания типа представляемой работы (текст или файл), количества рецензий на каждую работу и набора критериев оценивания. Система обязана автоматически распределять работы между студентами согласно заданным правилам, обеспечивая при этом анонимность оценивания при необходимости. Для студентов должен быть реализован удобный интерфейс подачи работ и их рецензирования с учетом установленных сроков.

Особое внимание уделено требованиям к системе оценивания. Модуль должен поддерживать различные шкалы оценок, предусмотренные в Moodle, и предоставлять инструменты для анализа полученных оценок. Алгоритмы выявления необъективных оценок на основе статистических методов (стандартное отклонение, анализ выбросов) являются обязательным компонентом системы. Формирование сводных отчетов для преподавателя с визуализацией результатов оценивания также входит в основные требования.

Нефункциональные требования определяют качественные характеристики модуля. Система должна обеспечивать стабильную работу большого количества студентов в одном курсе. Время отклика интерфейса не должно превышать 2 секунд даже при максимальной нагрузке. Особые требования предъявляются к безопасности - все операции должны проводиться с проверкой прав доступа, а передача данных обязана использовать стандартные механизмы защиты Moodle.

Требования к удобству использования предполагают интуитивно понятный интерфейс как для преподавателей, так и для студентов, не требующий дополнительного обучения. Модуль должен органично вписываться в стандартный интерфейс Moodle, соответствуя его стилю и принципам взаимодействия. Доступность для пользователей с ограниченными возможностями обеспечивается соблюдением стандартов WCAG.

**UI/UX-проектирование**

При разработке пользовательского интерфейса модуля взаимного рецензирования для Moodle основной задачей было обеспечение удобства и интуитивной понятности для всех пользователей - как преподавателей, так и студентов. Поскольку модуль разрабатывается в рамках экосистемы Moodle, ключевым требованием было строгое соблюдение стандартов оформления и поведения элементов интерфейса, принятых в этой платформе.

Для проектирования интерфейса были использованы встроенные средства Moodle, что гарантирует единообразие визуального стиля и поведения компонентов. Все формы модуля, включая формы отправки работ, рецензирования и административные интерфейсы, созданы с использованием стандартного API Moodle для работы с формами. Это обеспечивает автоматическую интеграцию с системой тем оформления Moodle и корректное отображение на различных устройствах.

Особое внимание было уделено простоте и понятности интерфейса для студентов. Форма отправки работы (реализованная в submit.php) содержит минимально необходимые элементы - поле для ввода текста или загрузки файла, с четкими инструкциями. Интерфейс рецензирования (review.php) организован таким образом, чтобы студент мог последовательно оценивать работу по заданным критериям, без отвлекающих элементов.

Для преподавательского интерфейса был разработан компактный и информативный dashboard (view.php), который отображает все необходимые данные - список работ, статус их проверки, полученные оценки. Использование стандартных табличных представлений Moodle позволяет преподавателю быстро ориентироваться в информации.

Все текстовые элементы интерфейса вынесены в языковые файлы (lang/en/mypeerreview.php), что обеспечивает простоту локализации и соответствие стандартам Moodle. Визуальные компоненты, такие как кнопки, формы ввода и таблицы, используют стандартные классы стилей Moodle, что гарантирует их корректное отображение независимо от выбранной темы оформления.

Важным аспектом проектирования стала доступность интерфейса. Все формы модуля разработаны с учетом требований доступности, включая правильную разметку, семантические HTML-элементы. Это обеспечивает возможность использования модуля студентами с ограниченными возможностями.

**Проектирование общей архитектуры модуля**

Архитектура модуля (Рисунок 1) реализована по трехуровневой модели. Пользовательские интерфейсы взаимодействуют с ядром логики через специализированные скрипты: преподавательские функции (start\_review.php, teacher\_review.php) и студенческие (submit.php, review.php). Логика, сосредоточенная в lib.php, обрабатывает запросы и взаимодействует с тремя основными таблицами БД. Файловые операции выполняются через Moodle File API с проверкой прав доступа. Критические операции защищены транзакционными механизмами.

Взаимодействие в системе выглядит следующим образом: Преподаватель создает задание, что инициирует процесс через файл mod\_form.php, который затем вызывает функцию mypeerreview\_add\_instance(). Эта функция отвечает за добавление записи в таблицу mypeerreview, где хранятся данные о созданном задании. После создания задания преподаватель запускает процесс рецензирования через start\_review.php, в результате чего происходит распределение работ для рецензирования, информация о котором сохраняется в таблице mypeerreview\_submissions. Студенты, в свою очередь, могут отправить свои работы через submit.php, что приводит к записи в таблицу mypeerreview\_submissions. Далее, студенты рецензируют работы других студентов через review.php, при этом информация о рецензиях записывается в таблицу mypeerreview\_reviews. Системные процессы включают в себя автоматическое распределение работ для рецензирования, выполняемое функцией mypeerreview\_distribute\_submissions(), и статистический анализ рецензирования, осуществляемый функцией mypeerreview\_get\_reviewer\_stats().

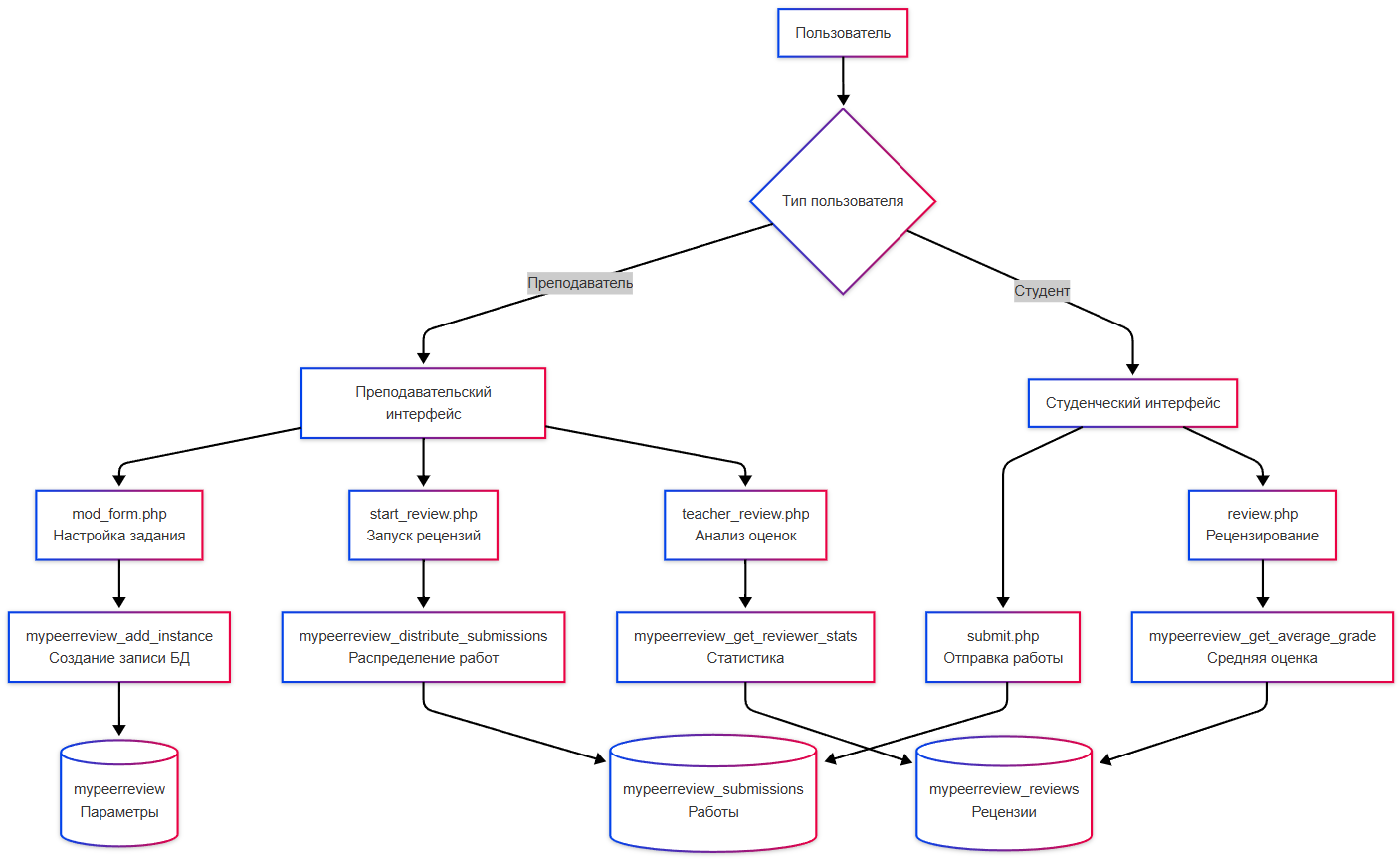


Рисунок 1 — Архитектура модуля

2.3 Выводы по главе 2

Таким образом, при проектировании модуля был сделан осознанный выбор в пользу использования стандартных средств Moodle, что позволило создать интуитивно понятный, удобный и доступный интерфейс, полностью интегрированный в экосистему платформы. Такой подход минимизирует затраты на обучение пользователей и обеспечивает стабильную работу модуля в различных окружениях.

ГЛАВА 3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО МОДУЛЯ

3.1 Реализация структуры базы данных с использованием XMLDB

Проектирование базы данных модуля взаимного рецензирования осуществлялось с учетом всех функциональных требований и особенностей интеграции с платформой Moodle. В основе реализации лежит трехуровневая структура данных, зафиксированная в файле install.xml (Рисунки 2-4) через систему XMLDB.



Рисунок 2 — Листинг кода базы данных, таблица mypeerreview

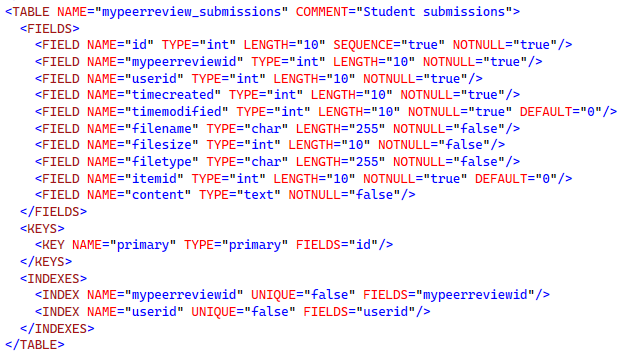


Рисунок 3 — Листинг кода базы данных, таблица mypeerreview\_submissions



Рисунок 4 — Листинг кода базы данных, таблица mypeerreview\_reviews

Центральным элементом структуры стала таблица mypeerreview, хранящая основные параметры задания. Она включает как стандартные для Moodle поля (идентификатор курса, название, описание), так и специализированные настройки: тип принимаемых работ (текст или файл), количество обязательных рецензий для каждой работы, максимально возможная оценка, флаг анонимности оценивания. Особое внимание уделено временным меткам - фиксации сроков выполнения задания и начала этапа рецензирования, что критично для контроля учебного процесса.

Как видно из представленной схемы (Рисунок 5), отношения между таблицами реализуют ключевые процессы модуля: создание заданий, отправку работ и процесс рецензирования.

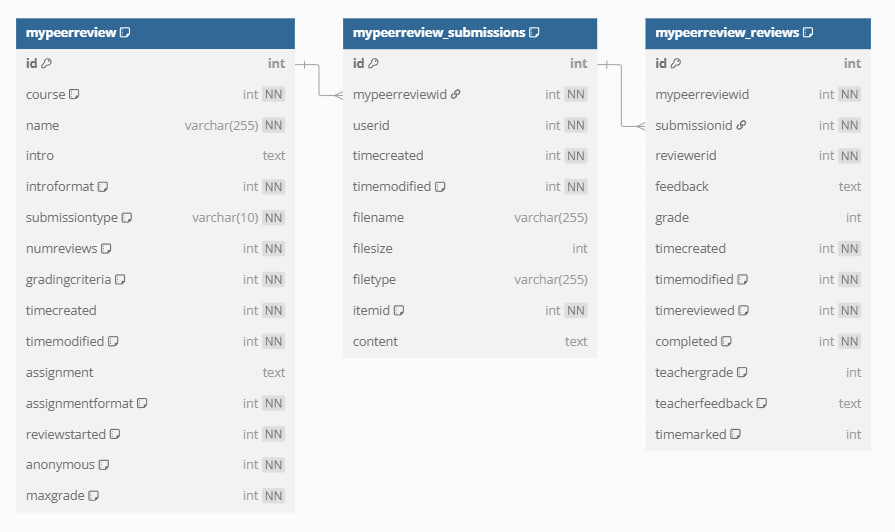


Рисунок 5 — Схема базы данных модуля взаимного рецензирования

Для хранения студенческих работ предназначена таблица mypeerreview\_submissions. Ее структура предусматривает два варианта хранения: текстовое содержание непосредственно в базе данных и файловые работы через систему хранения Moodle (с сохранением имени, размера и типа файла). Каждая работа жестко привязана к конкретному заданию и студенту, что обеспечивает целостность данных. Временные метки создания и изменения позволяют отслеживать соблюдение сроков сдачи.

Наиболее сложной в проектировании оказалась таблица рецензий (mypeerreview\_reviews). Помимо стандартных связей с заданием, проверяемой работой и рецензентом, она содержит несколько принципиально важных полей: текстовый отзыв, оценка, статус завершенности рецензии. Особенностью реализации стала двойная система оценивания - кроме оценки работы студентом-рецензентом, преподаватель может оценить и саму рецензию (поле teachergrade), что реализует механизм контроля качества взаимного оценивания.

Связи между таблицами организованы через систему внешних ключей, что гарантирует целостность данных при удалении или изменении записей. Для ускорения выборок созданы индексы по часто используемым полям (идентификаторы заданий, пользователей, работ).

Разработанная структура полностью поддерживает все заявленные функции модуля, включая гибкую настройку параметров задания преподавателем, поддержку различных форматов сдачи работ (текст или файл), а также контроль сроков выполнения этапов. Кроме того, система реализует механизм двойного оценивания, при котором работы проверяются как студентами, так и преподавателями, анализирует качество рецензий и выявляет недобросовестные оценки.

Использование XMLDB для описания структуры обеспечило простоту развертывания модуля на разных версиях Moodle и возможность автоматического обновления структуры при модификациях. Все таблицы и поля снабжены подробными комментариями, что упрощает дальнейшую поддержку и развитие модуля [9].

Кроме того, файл access.php определяет в системе управления доступом модуля Moodle. А именно, какие действия могут выполнять различные роли пользователей внутри модуля “MyPeerReview”. В частности, данный файл описывает набор возможностей (capabilities), таких как добавление экземпляра модуля, управление модулем, просмотр модуля, отправка работы на оценку и выполнение рецензирования. Каждая возможность связана с определенным уровнем контекста и привязана к ролям пользователей. Система разрешений, определенная в этом файле, напрямую влияет на проектирование базы данных, поскольку определяет, какие данные будут доступны для чтения и записи различным пользователям. Таким образом, проектирование базы данных должно учитывать структуру разрешений, описанную в access.php, для обеспечения целостности и безопасности данных.

Такая организация данных позволяет эффективно работать с модулем даже при большом количестве пользователей и заданий, обеспечивая при этом все необходимые функции для полноценного процесса взаимного рецензирования в учебном процессе [10].

3.2 Реализация основных функций модуля на PHP

Основу модуля взаимного рецензирования составляют PHP-скрипты, реализующие полный цикл работы с учебными заданиями — от их создания преподавателем до финального оценивания работ. Архитектура модуля построена вокруг трех ключевых компонентов: управления заданиями, обработки студенческих работ и системы рецензирования. Каждый компонент включает набор специализированных функций, обеспечивающих гибкость и надежность работы системы [11].

Как показано на Рисунке 6, модуль предоставляет три ролевых интерфейса (для студентов, преподавателей и администраторов), каждый из которых реализует специфический набор функций, таких как отправка работ, настройка критериев оценивания и мониторинг прогресса, что потребовало разработки соответствующих PHP-классов с четким разделением ответственности.

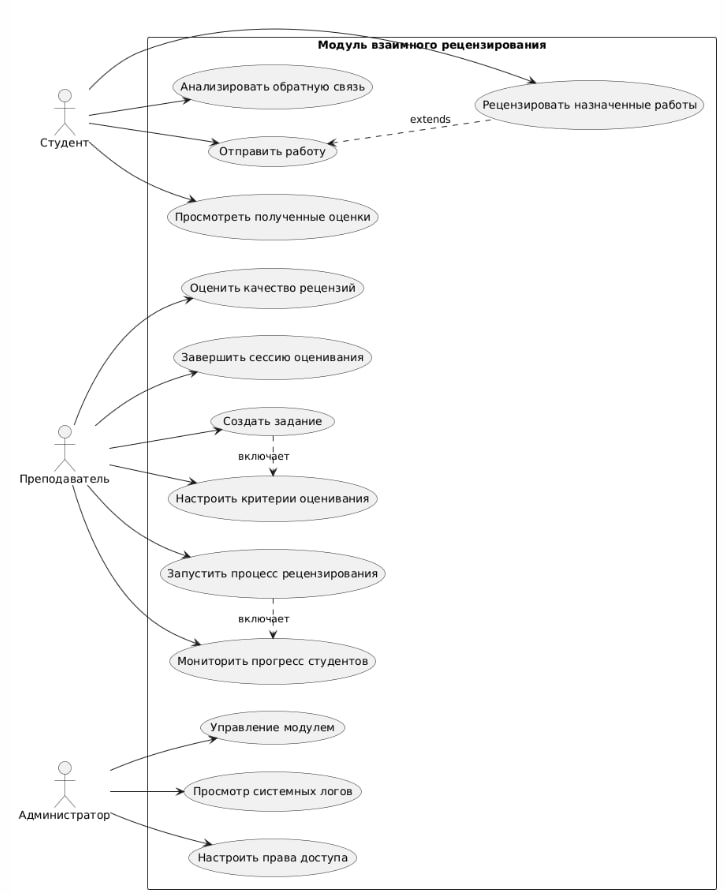


Рисунок 6 — Диаграмма вариантов использования модуля взаимного рецензирования

Основная логика модуля взаимного рецензирования реализована в файле lib.php, который содержит ключевые функции для управления жизненным циклом заданий. Функция mypeerreview\_add\_instance отвечает за создание новых заданий, обрабатывая данные из формы настроек, включая текстовое описание, тип принимаемых работ, количество обязательных рецензий и критерии оценивания. Особое внимание уделено работе с файлами - при загрузке преподавателем файлов критериев оценивания они сохраняются в системе хранения Moodle с привязкой к контексту модуля, что гарантирует их доступность только авторизованным пользователям с соответствующими правами доступа.

Процесс обновления параметров задания реализован в функции mypeerreview\_update\_instance, где предусмотрена комплексная проверка изменений. При обновлении файла критериев оценивания система автоматически удаляет старый файл, предотвращая накопление неиспользуемых данных в хранилище. Для текстовых полей применяется строгая очистка через параметр PARAM\_RAW с последующим форматированием в соответствии со стандартами безопасности Moodle, что исключает возможность внедрения вредоносного кода.

Функция mypeerreview\_delete\_instance обеспечивает полное и безопасное удаление заданий, реализуя каскадное удаление всех связанных данных. Это включает в себя не только удаление основной записи о задании, но и всех связанных студенческих работ из таблицы mypeerreview\_submissions, рецензий из mypeerreview\_reviews, а также соответствующих файлов из системы хранения Moodle. Для работы с файлами используется Moodle File API, что гарантирует корректное выполнение операций вне зависимости от объема хранимых данных и обеспечивает целостность файловой системы.

Модуль поддерживает два основных типа студенческих работ, что реализовано в логике файла submit.php. Текстовые работы сохраняются непосредственно в базе данных с применением специального форматирования через функцию format\_text(), которая позволяет сохранять необходимую разметку текста и одновременно обеспечивает защиту от XSS-атаки. Максимальный объем текстовой работы ограничен на уровне структуры базы данных использованием поля типа TEXT в MySQL.

Для файловых работ реализована система обработки через встроенное хранилище Moodle. Каждый загружаемый файл получает уникальный идентификатор, что исключает возможные конфликты имен. В процессе загрузки проводится комплексная проверка файла, включающая анализ его типа по расширению, контроль размера (сравнение с системным параметром $CFG->maxbytes) и проверку целостности загружаемых данных. В случае обнаружения ошибки на любом этапе обработки файла система выполняет откат транзакции, что предотвращает появление частично загруженных или поврежденных файлов.

Алгоритм распределения работ между студентами, реализованный в start\_review.php, представляет собой механизм, обеспечивающий справедливое и сбалансированное назначение рецензий. Система формирует пул работ, исключая собственные отправки студента, чтобы избежать конфликта интересов. Распределение происходит случайным образом с использованием функции shuffle(), но с дополнительным контролем баланса - каждый студент получает строго определенное количество работ для рецензирования, указанное в параметре numreviews. Особое внимание уделено поддержке анонимности - при включенном параметре anonymous система автоматически скрывает информацию об авторе работы, обеспечивая объективность оценивания.

Функционал рецензирования, реализованный в review.php, включает комплексную систему оценивания с многоуровневой валидацией данных. Форма рецензирования требует обязательного заполнения поля с текстовым отзывом и выбора оценки из строго определенного диапазона (от 1 до maxgrade). Все данные сохраняются с точной меткой времени, что позволяет преподавателю отслеживать соблюдение сроков сдачи рецензий и оперативно реагировать на возможные задержки.

Для обеспечения качества оценивания модуль включает развитую систему статистического анализа. Функция mypeerreview\_get\_reviewer\_stats выполняет комплексный анализ деятельности каждого рецензента, вычисляя средний балл, который он выставляет, стандартное отклонение для выявления аномальных оценок, а также определяет статистические выбросы - оценки, отличающиеся более чем на два стандартных отклонения от среднего значения. Эти данные представляются преподавателю в удобном графическом виде через интерфейс teacher\_review.php, где потенциально необъективные оценки выделяются цветовой индикацией (красным цветом), что позволяет быстро идентифицировать проблемные моменты в процессе взаимного оценивания.

Модуль взаимного рецензирования глубоко интегрирован в экосистему Moodle, используя все основные встроенные механизмы платформы. Система управления доступом реализована через стандартные capabilities (mod/mypeerreview:submit, mod/mypeerreview:review), что обеспечивает гибкую настройку прав для различных ролей пользователей - от студентов до преподавателей и администраторов курсов. Работа с файлами полностью осуществляется через стандартный API Moodle (get\_file\_storage()) с обязательной проверкой контекста, что гарантирует безопасность хранения и передачи файлов.

Важной частью интеграции является система обработки событий. Модуль генерирует специализированные события (review\_submitted, review\_assessed), которые позволяют интегрировать его функционал с журналом событий Moodle и системой уведомлений. Это дает возможность преподавателям оперативно получать информацию о ключевых действиях студентов, таких как отправка работ или сдача рецензий, а также позволяет автоматизировать часть процессов управления курсом.

При разработке модуля особое внимание уделено вопросам надежности и производительности. Все критические операции, такие как распределение работ между студентами, выполняются в рамках транзакций через $DB->start\_delegated\_transaction(), что гарантирует целостность данных даже в случае возникновения непредвиденных ошибок или сбоев системы. Для обеспечения высокой производительности при работе с большими курсами оптимизированы все основные запросы к базе данных - созданы специальные индексы по полям mypeerreviewid и userid, а массовые операции (например, подсчет количества рецензий) выполняются единым сложным SQL-запросом, что значительно снижает нагрузку на систему.

Вопросы безопасности занимают центральное место в архитектуре модуля. Все входные параметры проходят строгую валидацию через фильтры PARAM\_INT и PARAM\_TEXT, что исключает возможность SQL-инъекций или других видов атак. Доступ к файлам и критически важным функциям защищен многоуровневой системой проверки прав через require\_capability(). Дополнительно реализована защита от CSRF-атак через стандартный механизм Moodle require\_sesskey().

Представленная реализация модуля взаимного рецензирования демонстрирует высокую степень надежности и гибкости, позволяя эффективно использовать его в самых разных образовательных сценариях - от небольших семинарских занятий до массовых открытых онлайн-курсов. При этом модуль полностью соответствует стандартам разработки для платформы Moodle и легко интегрируется в существующую образовательную среду.

3.3 Реализация API Moodle для модуля взаимного рецензирования

Система Moodle предоставляет комплексный программный интерфейс (API), который был использован при разработке модуля взаимного рецензирования. Этот интерфейс включает несколько ключевых компонентов, обеспечивающих глубокую интеграцию с платформой. Основными используемыми компонентами стали: Plugin API для базовой интеграции модулей, Database API для работы с базой данных, File API для управления файлами, Event API для обработки событий и Web Service API для доступа к функционалу через веб-сервисы. Комплексное использование этих компонентов позволило создать модуль, полностью соответствующий стандартам разработки для платформы Moodle.

Базовый функционал модуля реализован через стандартные механизмы Plugin API. В первую очередь были разработаны функции жизненного цикла модуля, включая обработку создания, обновления и удаления экземпляров. Функция mypeerreview\_supports определяет поддерживаемые модулем возможности, такие как введение модуля (FEATURE\_MOD\_INTRO) и поддержку резервного копирования (FEATURE\_BACKUP\_MOODLE2). Для управления правами доступа реализована система кастомных разрешений (capabilities), включающая права на отправку работ (mod/mypeerreview:submit), создание рецензий (mod/mypeerreview:review) и управление процессом (mod/mypeerreview:manage). Проверка прав осуществляется через стандартную функцию require\_capability().

Для организации хранения данных модуль использует Database API, который предоставляет унифицированный интерфейс работы с СУБД. Структура таблиц описана в формате XMLDB (файл install.xml), что позволяет автоматически создавать и обновлять схему базы данных при установке модуля. Для выполнения запросов используются функции DML (Data Manipulation Language), обеспечивающие безопасное выполнение операций выборки, вставки и обновления данных. Особое внимание уделено поддержке транзакций при выполнении сложных операций, а также кэшированию часто выполняемых запросов для повышения производительности системы.

Модуль реализует расширенную функциональность работы с файлами через File API. Система использует два типа хранилищ: временное (draft-файлы) для промежуточного хранения и постоянное, привязанное к контексту модуля. Для каждого файла осуществляется контроль доступа в соответствии с правами пользователя. Особенностью реализации является поддержка различных типов файловых вложений к работам студентов и рецензиям, что обеспечивает гибкость в организации учебного процесса.

Важной частью архитектуры модуля является система обработки событий, построенная на базе Event API. Модуль генерирует кастомные события, такие как отправка рецензии (review\_submitted) или оценка рецензии преподавателем (review\_assessed). Каждое событие содержит полный контекст выполнения, включая идентификаторы связанных объектов и пользователей. Механизм обработки событий позволяет другим компонентам системы реагировать на действия пользователей в модуле, обеспечивая интеграцию с системой логирования, уведомлений и другими функциями Moodle.

Для обеспечения внешнего доступа к функционалу модуля реализована поддержка Web Service API. Разработаны кастомные веб-сервисы, позволяющие получать список работ (mod\_mypeerreview\_get\_submissions) и отправлять рецензии (mod\_mypeerreview\_submit\_review). Конфигурация сервисов осуществляется через стандартный механизм Moodle в файле db/services.php, где для каждого сервиса определяются доступные функции, уровень ограничения доступа и другие параметры. Это позволяет легко интегрировать модуль с внешними системами и мобильными приложениями.

При реализации API особое внимание уделено вопросам безопасности. Все входные параметры проходят строгую валидацию, для каждого вызова выполняется проверка прав доступа, реализована защита от CSRF-атак через стандартный механизм Moodle require\_sesskey(). Для обеспечения высокой производительности используются оптимизированные SQL-запросы, кэширование часто используемых данных и пакетная обработка при массовых операциях. Архитектура модуля разработана с учетом принципов расширяемости: соблюдено четкое разделение интерфейсов, предусмотрены хуки для модификации поведения и возможность переопределения ключевых методов.

Модуль взаимного рецензирования тесно интегрирован с основными подсистемами Moodle. Реализована полная поддержка журнала оценок (Gradebook) через систему callback-функций, включая работу со стандартными шкалами оценивания. Для администраторов и преподавателей предусмотрены подробные отчеты, генерируемые в стандартных форматах Moodle. Модуль поддерживает интеграцию с системами аутентификации, включая механизм единого входа (SSO) и системы электронных подписей. Все значимые действия пользователей фиксируются в системном журнале событий.

Использование API Moodle при разработке модуля взаимного рецензирования позволило достичь нескольких важных преимуществ. Во-первых, обеспечилась стандартизированная интеграция с ядром системы, гарантирующая совместимость с различными версиями платформы. Во-вторых, были реализованы современные механизмы безопасности работы с данными. В-третьих, архитектура модуля предусматривает возможности для дальнейшего расширения функционала. Важным результатом стало значительное сокращение объема кода за счет использования встроенных механизмов платформы, что упрощает поддержку и развитие модуля. Все это делает разработанное решение полноценным компонентом экосистемы Moodle, соответствующим современным стандартам разработки образовательных технологий.

3.4 Реализация пользовательского интерфейса с использованием шаблонов Moodle

Разработка пользовательского интерфейса для модуля взаимного рецензирования осуществлялась с глубокой интеграцией в стандартную систему шаблонов Moodle, что обеспечило не только единообразие визуального оформления, но и полное соответствие принципам юзабилити, принятым в экосистеме Moodle. В основе интерфейсных решений лежал компонентный подход, позволяющий создавать гибкие и переиспользуемые элементы, адаптирующиеся под различные сценарии взаимодействия пользователей с системой.

Архитектура пользовательского интерфейса строилась вокруг ключевых workflow модуля: подачи студенческих работ, распределения рецензий, процесса оценивания и анализа результатов. Для каждого из этих процессов были разработаны специализированные экраны, объединенные общей логикой навигации и единым визуальным языком. Особое внимание уделялось созданию интуитивно понятных путей взаимодействия, минимизирующих когнитивную нагрузку на пользователей при выполнении типовых операций. Как видно из Рисунка 7, пользовательские интерфейсы модуля (для студентов, преподавателей и администраторов) разрабатывались с учетом их тесного взаимодействия с серверной частью через API Moodle, что обеспечило единообразие визуального оформления и согласованность данных при работе с базой.

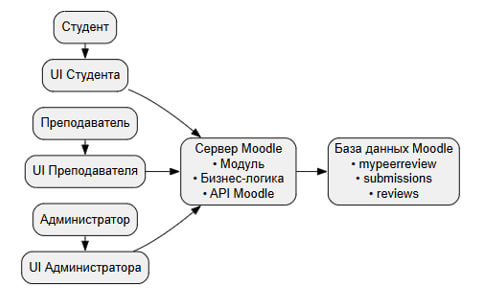


Рисунок 7 — Архитектура взаимодействия пользовательских интерфейсов модуля с серверной частью

При реализации интерфейсных компонентов активно использовались возможности шаблонизатора Mustache, интегрированного в Moodle. Этот подход позволил четко разделить логику представления данных и логику приложения, что значительно упростило поддержку и развитие интерфейсной части модуля. Все шаблоны разрабатывались с учетом требований к адаптивности, обеспечивая корректное отображение на устройствах с различными разрешениями экранов - от десктопных мониторов до мобильных устройств.

Динамические элементы интерфейса, такие как системы комментариев, интерактивные формы оценивания и инструменты предпросмотра работ (Рисунки 8-12), реализовывались с использованием современного фреймворка Moodle, что гарантировало их бесшовную интеграцию с основными компонентами платформы. Для обработки пользовательских действий применялась модель событийно-ориентированного программирования, обеспечивающая быстрый отклик интерфейса и плавность взаимодействия.

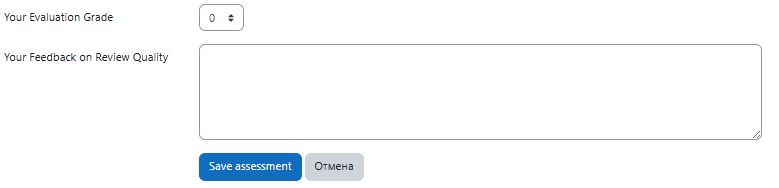


Рисунок 8 — Интерфейс преподавателя при заполнении комментария для работы студента

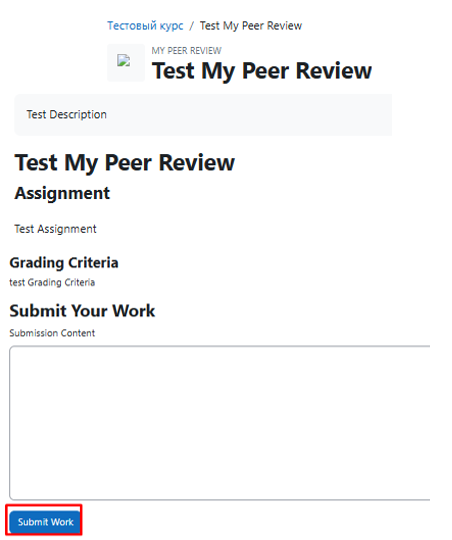


Рисунок 9 — Интерфейс студента до загрузки работы

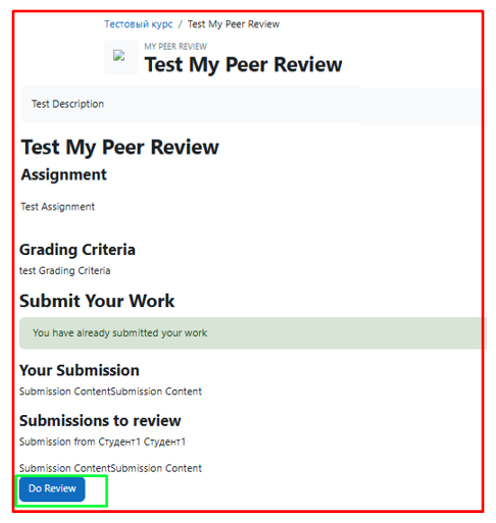


Рисунок 10 — Интерфейс студента после загрузке работы

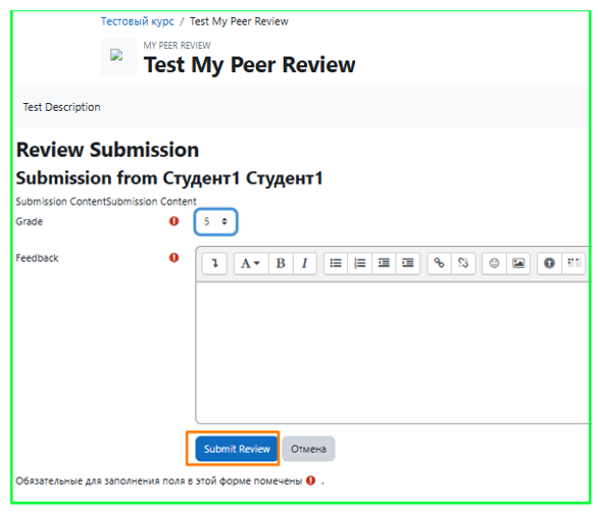


Рисунок 11 — Интерфейс студента до оценивания другого студента

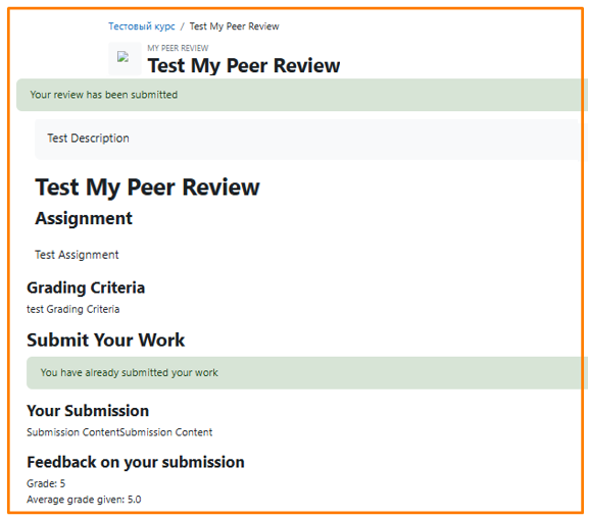


Рисунок 12 — Интерфейс студента после оценивании другого студента

Доступность интерфейса обеспечивалась за счет строгого следования принципам WCAG 2.1, включая семантическую верстку, правильное использование ARIA-атрибутов [11], поддержку клавиатурной навигации и адаптивных цветовых схем, как на Рисунке 13. Все графические элементы сопровождались альтернативными текстовыми описаниями.

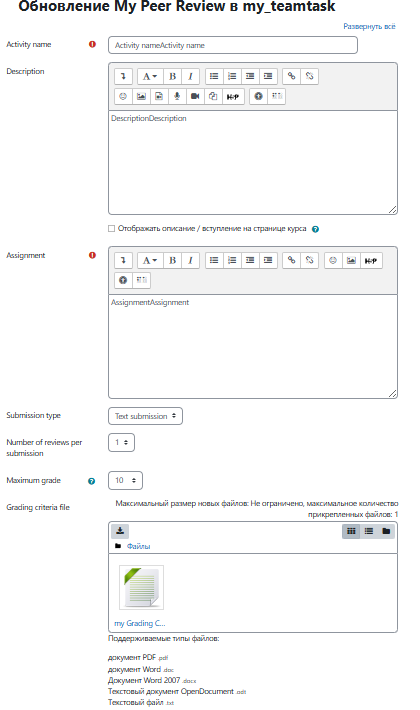


Рисунок 13 — Интерфейс добавления модуля на курс

Система визуализации данных модуля включала специализированные компоненты для отображения статистики по рецензированию, сравнительного анализа оценок и мониторинга прогресса выполнения заданий, как на Рисунках 14-16. Эти элементы разрабатывались с учетом психологических аспектов восприятия информации, используя проверенные паттерны визуализации образовательных данных [12].

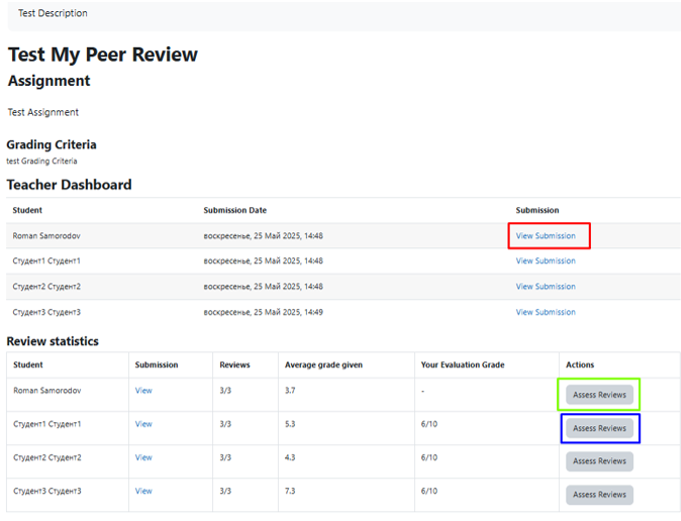


Рисунок 14 — Интерфейс преподавателя

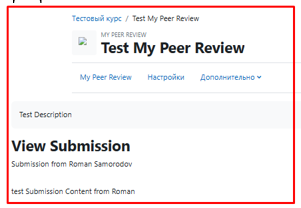


Рисунок 15 — Интерфейс преподавателя при обзоре работ студентов

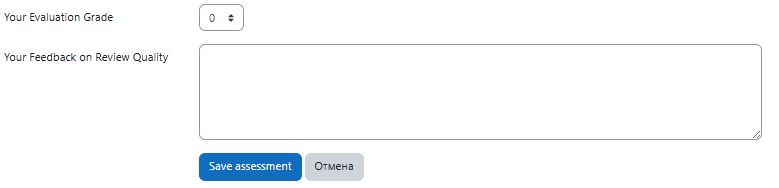


Рисунок 16 — Интерфейс преподавателя при заполнении комментария для работы студента

Локализация интерфейса реализована через стандартную систему языковых пакетов Moodle, с полным выносом всех текстовых элементов в языковые файлы. Это не только упростило процесс перевода на другие языки, но и позволило легко адаптировать терминологию под специфику конкретных образовательных учреждений.

Особенностью реализации стала глубокая интеграция с системой тем оформления Moodle. Интерфейс модуля автоматически адаптируется к выбранной цветовой схеме и параметрам отображения, сохраняя визуальную целостность с основной платформой. Для мобильных устройств были разработаны оптимизированные версии ключевых экранов, учитывающие особенности touch-взаимодействия.

3.5 Реализация системы выявления недобросовестных студентов

Для эффективного выявления недобросовестных студентов необходимо понимать основные принципы статистического анализа данных, полученных в процессе взаимного рецензирования. Ключевым аспектом является разработка метрик и алгоритмов, способных выявить аномалии в оценках. В рамках данной работы используются следующие статистические методы:

Среднее арифметическое - базовая метрикой, позволяющей определить средний уровень оценки, выставленной конкретным рецензентом. Рассчитывается путем суммирования всех оценок и деления на количество оценок. В контексте данной работы, среднее арифметическое используется для получения общего представления об уровне оценок каждого студента и для сравнения с оценками других рецензентов.

Стандартное отклонение - этот показатель характеризует степень разброса оценок относительно среднего значения. Высокое стандартное отклонение указывает на значительные колебания в оценках рецензента, что может свидетельствовать о нестабильности в его подходе к оцениванию или о возможном предвзятом отношении к некоторым работам. В свою очередь, низкое стандартное отклонение говорит о более последовательном и однородном подходе к оценке.

Межквартильный размах (IQR) и выбросы - этот метод позволяет идентифицировать оценки, которые значительно отличаются от основной массы оценок. Межквартильный размах представляет собой разницу между третьим и первым квартилями (75-й и 25-й перцентили соответственно). Оценки, выходящие за пределы определенного интервала, заданного на основе IQR, считаются выбросами. В данной системе для определения выбросов используется коэффициент 1.5, что позволяет выявлять оценки, которые значительно отклоняются от среднего значения и могут указывать на недобросовестное оценивание.

Для каждой рецензии и для каждого рецензента вычисляется средняя оценка.

php

$avg = array\_sum($grades) / count($grades);

Листинг 1 — Реализация расчета среднего арифметического

Определяется мера разброса оценок для каждого рецензента.

php

$variance = 0.0;

foreach ($grades as $grade) {

$variance += pow($grade - $avg, 2);

}

$deviation = (count($grades) > 1) ?

sqrt($variance / count($grades)):0;

Листинг 2 — Реализация расчета стандартного отклонения.

Выявляются оценки, которые значительно отклоняются от общей массы оценок. Используется межквартильный размах (IQR) и заданный порог (threshold) для определения выбросов.

php

function mypeerreview\_is\_grade\_outlier($grade, $all\_grades, $threshold = 1.5) {

if (count($all\_grades) < 3) return false;

sort($all\_grades);

$q1 = $all\_grades[floor(count($all\_grades) \* 0.25)]; *// Первый квартиль (25-й перцентиль)*

$q3 = $all\_grades[floor(count($all\_grades) \* 0.75)]; *// Третий квартиль (75-й перцентиль)*

$iqr = $q3 - $q1; *// Межквартильный размах*

return ($grade < ($q1 - $threshold \* $iqr)) || ($grade > ($q3 + $threshold \* $iqr));

}

Листинг 3 — Реализация алгоритма поиска выбросов (outliers).

Алгоритм mypeerreview\_is\_grade\_outlier определяет, является ли оценка выбросом, учитывая все оценки, выставленные рецензентом. Сначала проверяется, достаточно ли оценок (минимум 3) для надежного анализа. Затем оценки сортируются, вычисляются первый и третий квартили, а также межквартильный размах. Оценка считается выбросом, если она находится за пределами интервала, определяемого как Q1 - 1.5*IQR и Q3 + 1.5*IQR.

Для облегчения анализа результатов и предоставления преподавателям наглядной информации о потенциально недобросовестных студентах разработан интерфейс визуализации данных (Рисунки 17-18). Он предоставляет статистику рецензента, включая среднюю оценку и стандартное отклонение, что позволяет быстро оценить общий уровень оценок и степень их разброса.

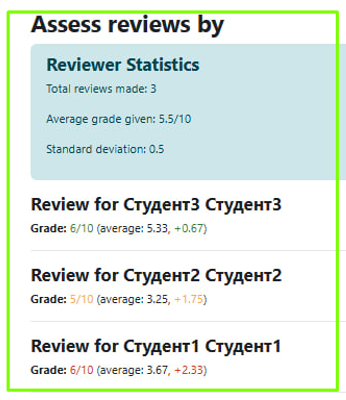


Рисунок 17 — Интерфейс преподавателя при оценивании рецензий честного студента



Рисунок 18 — Интерфейс преподавателя при оценивании рецензий недобросовестного студента

Для наглядности оценки закодированы цветом: красным выделяются оценки, значительно ниже среднего (>2σ), желтым – немного отличающиеся от среднего (1-2σ), а зеленым – близкие к среднему (<1σ). Кроме того, интерфейс включает список выбросов, идентифицированных с использованием алгоритма на основе межквартильного размаха. Разработанная система позволяет преподавателям более эффективно контролировать процесс взаимного рецензирования и выявлять случаи недобросовестного оценивания. Благодаря визуализации данных и алгоритмам статистического анализа, преподаватели могут быстро идентифицировать студентов, чьи оценки вызывают подозрения.

Например, если студент систематически завышает или занижает оценки, это будет отражено в высоком стандартном отклонении и большом количестве выбросов.

Важно отметить, что система не предоставляет окончательный вердикт о недобросовестности студента. Она лишь предоставляет информацию, которая может быть использована преподавателем для дальнейшего расследования. Преподаватель должен учитывать контекст ситуации, личные качества студентов и другие факторы, прежде чем делать какие-либо выводы.

3.6 Выводы по главе 3

В результате реализации модуля на базе стандартных шаблонов Moodle удалось создать удобную, доступную и производительную систему взаимодействия, полностью соответствующую ожиданиям различных категорий пользователей - от студентов, впервые сталкивающихся с системой взаимного рецензирования, до преподавателей, управляющих сложными процессами оценивания в крупных учебных курсах.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы был разработан программный модуль взаимного рецензирования для системы управления обучением Moodle, с использованием языка программирования PHP и базы данных MySQL.

Для достижения поставленной цели были выполнены следующие задачи:

1. Проведено исследование предметной области взаимного оценивания в образовательном процессе;
2. Определены функциональные и нефункциональные требования к программному модулю взаимного рецензирования для Moodle;
3. Изучены API Moodle, PHP и MySQL для разработки и интеграции модуля;
4. Разработана структура базы данных модуля;
5. Реализована серверная часть модуля на PHP, обеспечивающая управление командами, заданиями и оценками;
6. Разработан пользовательский интерфейс модуля Moodle;

В дальнейшем планируются следующие пути развития разработанного программного модуля:

* Улучшить систему критериев оценки;
* Добавить локализацию на другие языки;
* Добавить систему уведомлений.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Moodle. Open-source learning platform. [Электронный ресурс] URL: https://moodle.org (дата обращения: 11.01.2025).
2. Харченко В.С. Взаимное рецензирование в учебных курсах вуза // Высшее образование в России. — 2017. — № 2. — С. 4–6. — URL: https://vovr.elpub.ru/jour/article/view/969/869.
3. Ruth C. Clark, Richard E. Mayer e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning. — Wiley, 2016. — 528 с.
4. Steve Foreman The LMS Guidebook: Learning Management Systems Demystified. — Association for Talent Development, 2017. — 248 с.
5. Громаков Е.И., Сидорова А.А. Современные технологии. Киберфизические системы – Томск: Изд-во Томского политехнического университета, 2021. – 166 с.
6. Зайцев К.А. МЕТОДОЛОГИЯ РЕАЛИЗАЦИИ МОДУЛЕЙ В СДО MOODLE // E-Scio. 2020. №7 (46). URL: https://cyberleninka.ru/article/n/metodologiya-realizatsii-moduley-v-sdo-moodle (дата обращения: 21.01.2025).
7. Системы дистанционного обучения. [Электронный ресурс] URL: https://evergreens.com.ua/ru/articles/best-lms-2020.html (дата обращения: 11.01.2025).
8. Жигалов Е.В. Архитектура Moodle. // Открытое и дистанционное образование. 2012. – № 4. – С. 24-28.
9. Activity modules. [Электронный ресурс] URL: https://docs.moodle.org/dev/Activity\_modules (дата обращения: 11.01.2025).
10. Jason Cole, Helen Foster Using Moodle: Teaching with the Popular Open Source Course Management System. — O'Reilly Media, 2007. — 284 с.
11. PHPUnit. [Электронный ресурс] URL: https://phpunit.de (дата обращения: 11.01.2025).
12. ГОСТ Р 52653-2006 Информационно-коммуникационные технологии в образовании. Термины и определения [Электронный ресурс]. — Введ. 2007-01-01. — М.: Стандартинформ, 2006. — 12 с. — URL: https://files.stroyinf.ru/Index2/1/4293836/4293836364.htm (дата обращения: 25.01.2025).

ПРИЛОЖЕНИЯ

**Приложение А - Структурные схемы**

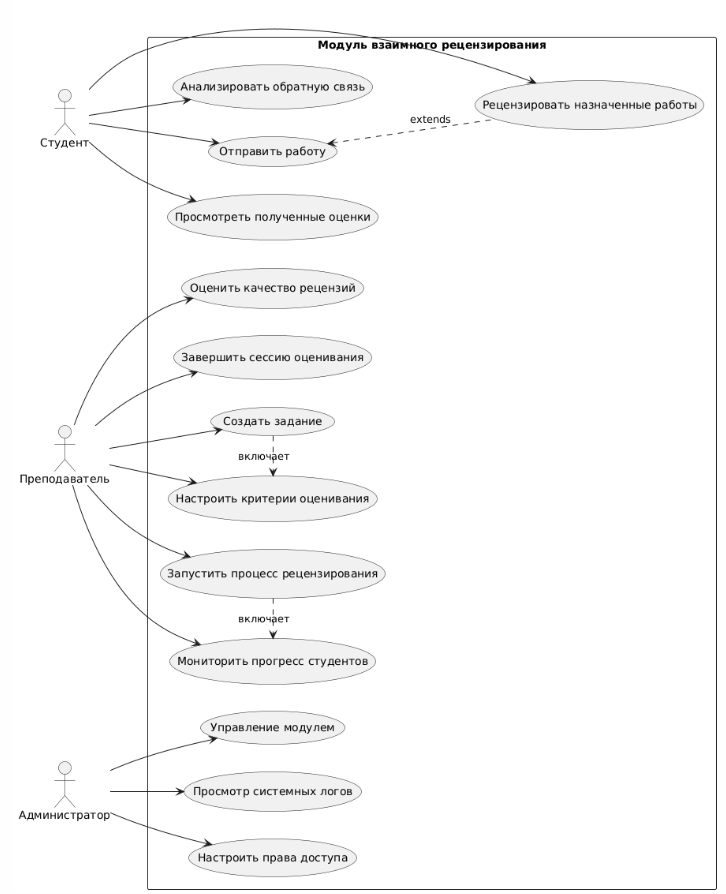


Рисунок 19 — Диаграмма вариантов использования

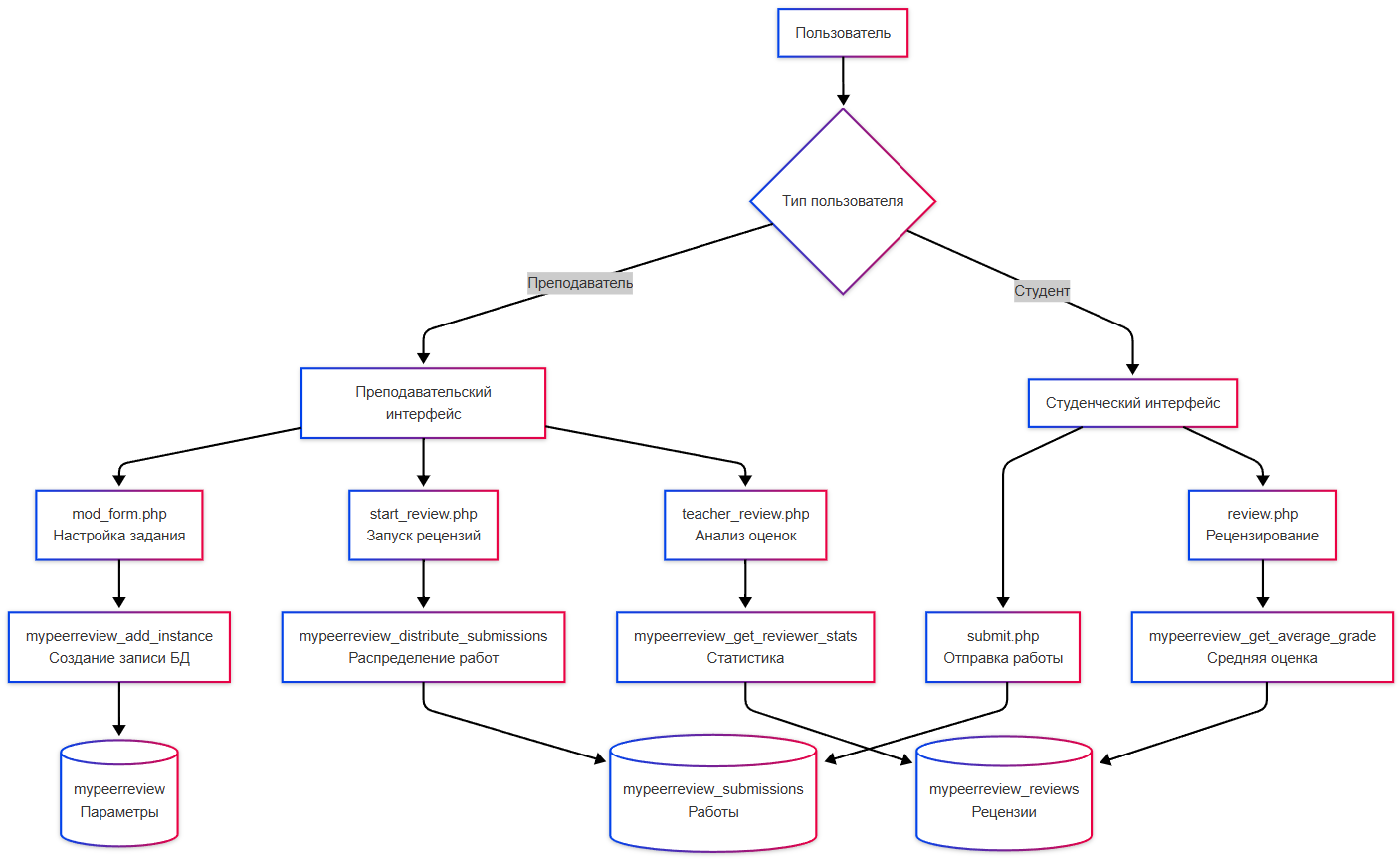


Рисунок 20 — Архитектура системы работы модуля

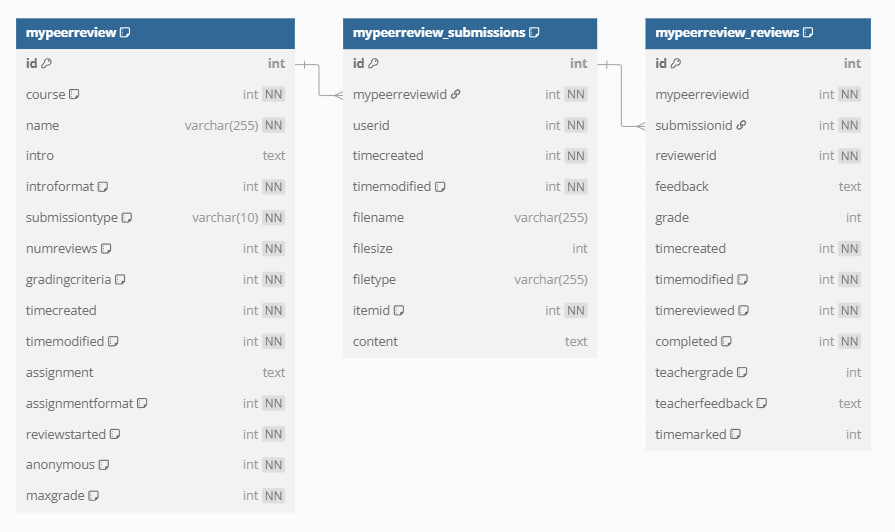


Рисунок 21 — База данных

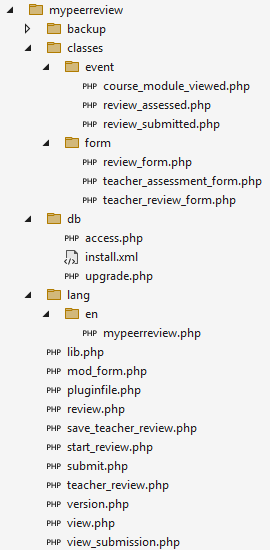


Рисунок 22 — Архитектура системы работы модуля

**Приложение Б - Листинг кода**



Рисунок 23 — Листинг кода базы данных, таблица mypeerreview

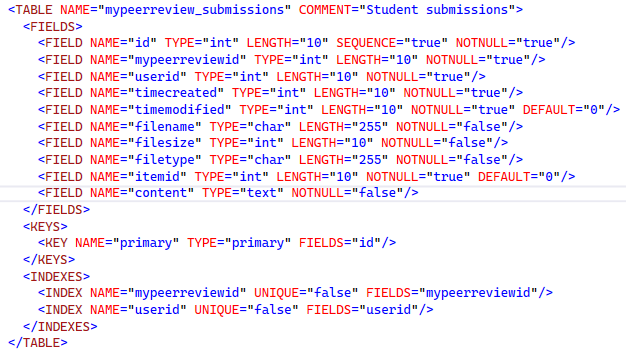


Рисунок 24 — Листинг кода базы данных, таблица mypeerreview\_submissions



Рисунок 25 — Листинг кода базы данных, таблица mypeerreview\_reviews



Рисунок 26 — Листинг кода файла прав доступа



Рисунок 27 — Листинг кода файла локализации



Рисунок 28 — Листинг кода review\_form.php формы для ревью

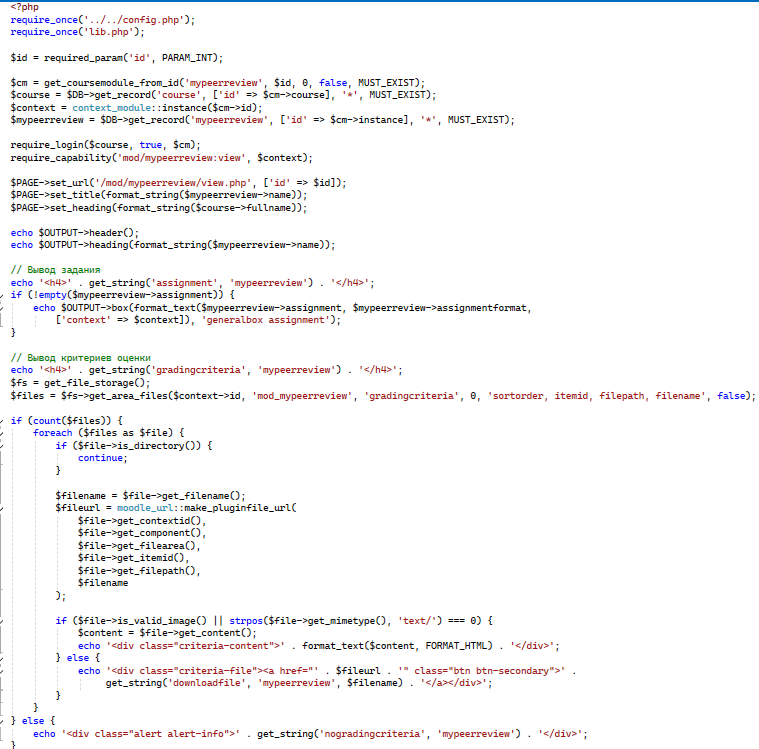


Рисунок 29 — Листинг части кода view.php

**Приложение В - Пользовательский интерфейс**

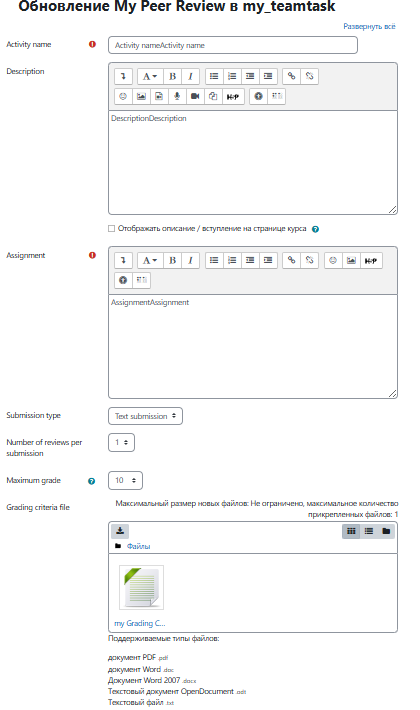


Рисунок 30 — Интерфейс добавления модуля на курс

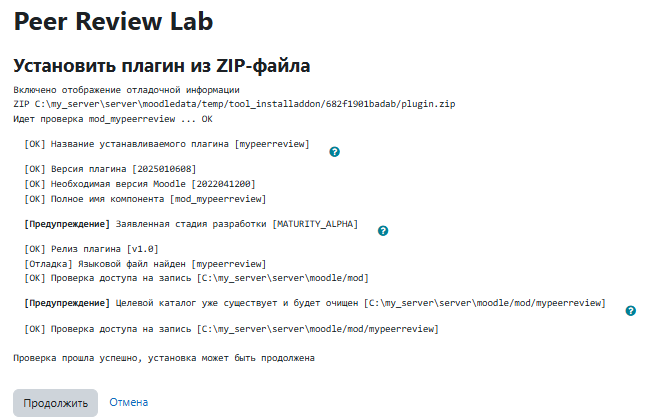


Рисунок 31 — Интерфейс добавления модуля на сервер

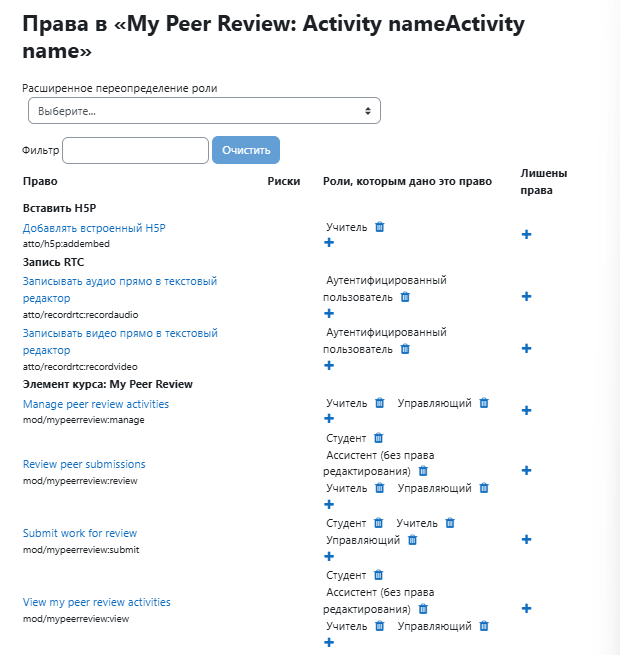


Рисунок 32 — Интерфейс администратора модуля

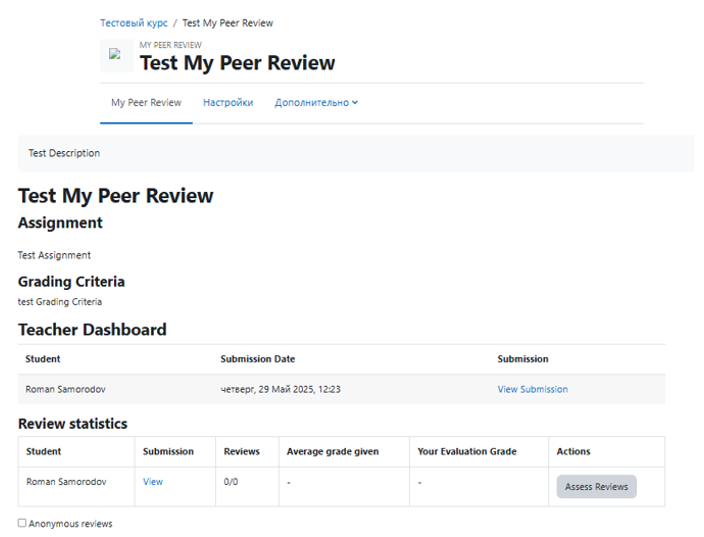


Рисунок 33 — Интерфейс преподавателя до начала рецензирования

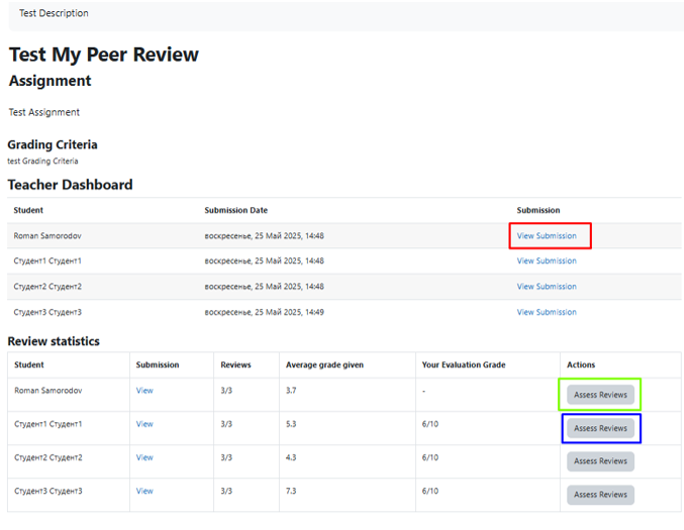


Рисунок 34 — Интерфейс преподавателя после начала рецензирования

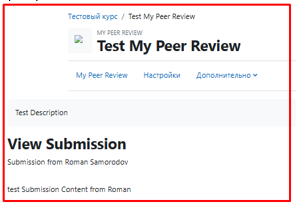


Рисунок 35 — Интерфейс преподавателя при обзоре работ студентов

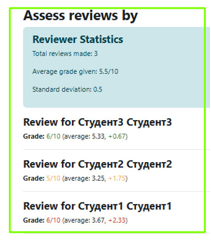


Рисунок 36 — Интерфейс преподавателя при оценивании рецензий честного студента



Рисунок 37 — Интерфейс преподавателя при оценивании рецензий недобросовестного студента

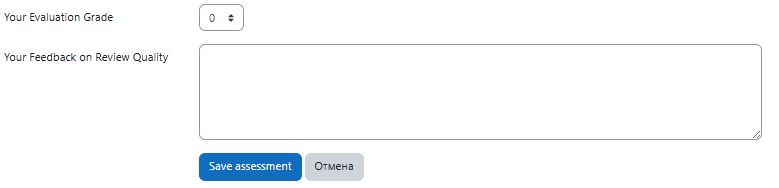


Рисунок 38 — Интерфейс преподавателя при заполнении комментария для работы студента

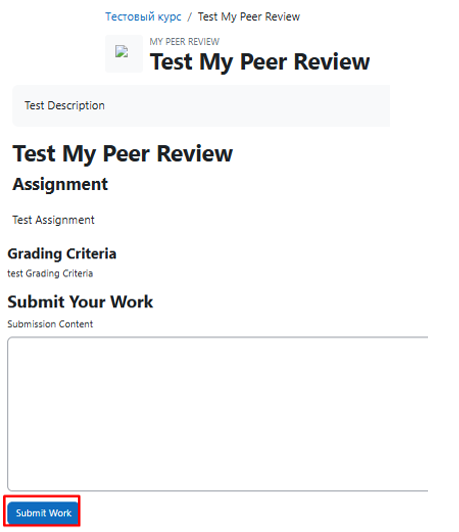


Рисунок 39 — Интерфейс студента до загрузки работы

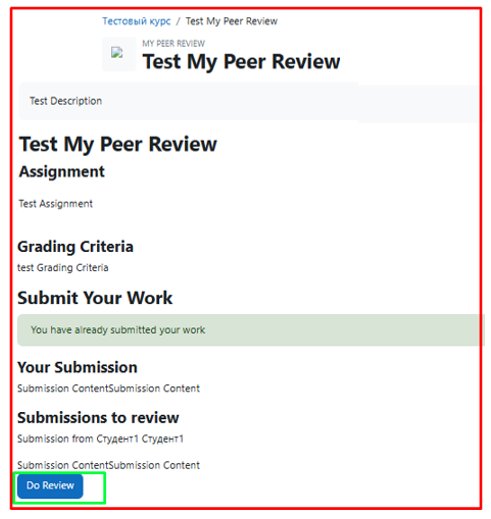


Рисунок 40 — Интерфейс студента после загрузке работы

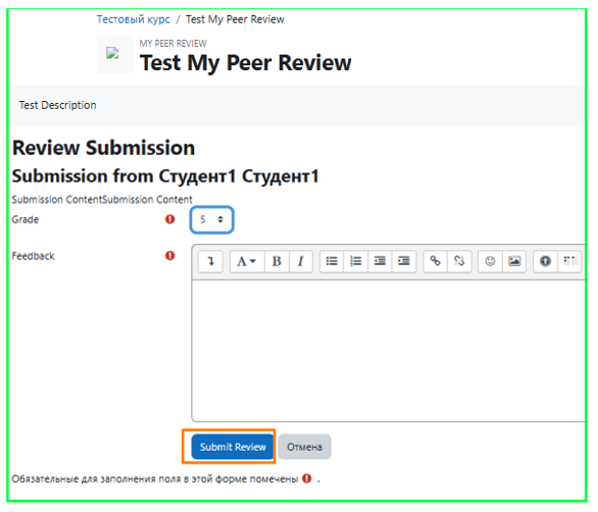


Рисунок 41 — Интерфейс студента до оценивания другого студента

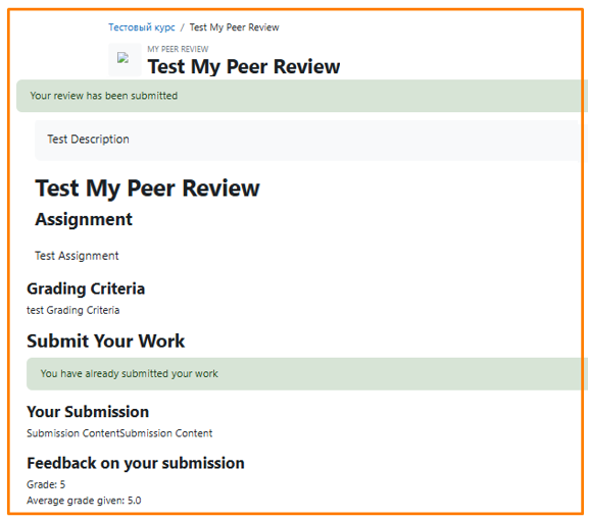


Рисунок 42 — Интерфейс студента после оценивании другого студента