# Содержание

Содержание 1

История изменений 2

1 Введение 3

1.1 Цели 3

1.2 Границы применения 3

1.3 Термины, аббревиатуры, сокращения 3

1.4 Ссылки 3

1.5 Краткий обзор 3

2 Общее описание 3

2.1 Описание изделия 3

2.1.1 Интерфейсы системы 3

2.1.2 Интерфейсы пользователя 3

2.1.3 Интерфейсы аппаратных средств ЭВМ 3

2.1.4 Интерфейсы программного обеспечения 3

2.1.5 Интерфейсы коммуникаций 3

2.1.6 Ограничения памяти 4

2.1.7 Действия 4

2.1.8 Требования настройки рабочих мест 4

2.2 Функции изделия 4

2.3 Характеристики пользователей 4

2.4 Ограничения 4

2.5 Предположения и зависимости 4

2.6 Распределение требований 4

3 Детальные требования 4

3.1 Функциональные требования 4

3.1.1 <Functional Requirement One> 5

3.2 Надежность 5

3.2.1 <Reliability Requirement One> 5

3.3 Производительность 5

3.3.1 <Performance Requirement One> 5

3.4 Ремонтопригодность 5

3.4.1 <Maintainability Requirement One> 5

3.5 Ограничения проекта 5

3.5.1 <Design Constraint One> 5

3.6 Требования к пользовательской документации 5

3.7 Используемые приобретаемые компоненты 5

3.8 Интерфейсы 5

3.8.1 Интерфейс пользователя 5

3.8.2 Аппаратные интерфейсы 5

3.8.3 Программные интерфейсы 5

3.8.4 Интерфейсы коммуникаций 5

3.9 Требования лицензирования 5

3.10 Применимые стандарты 5

Индекс 5

# История изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор(ы)** |
| 2023-10-04 | 0.1 | Начальная ревизия | Филатов Максим, Никитин Кирилл, Кочетов Николай |
| 2023-10-18 | 0.2 | Дополнение третьего раздела | Филатов Максим, Никитин Кирилл, Кочетов Николай |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Введение

## Цели

Целью данной системы является проверка студентов в онлайн универе посредством оценки работ по практике одних студентов другими и в определении функциональных требований и характеристик системы онлайн-обучения, а также определении ожидаемого поведения системы и ограничений, которым она должна соответствовать.

Также данный документ предназначен для команды разработчиков, тестировщиков, менеджеров проекта, стейкхолдеров и других участников, связанных с планированием, разработкой, тестированием и внедрением системы онлайн-обучения.

## Границы применения

1. Изделие программного обеспечения, которое будет произведено: cистема онлайн-обучения, которая будет называться "EduPlatform".
2. Изделие программного обеспечения будет предоставлять возможность студентам и преподавателям обмениваться ресурсами для обучения и оценивать работы в рамках курсов.   
   Система позволит загружать работы студентов, автоматически выбирать рецензентов для их оценки, рассчитывать итоговые оценки и передавать результаты студентам и преподавателям.   
   Если оценки сильно различаются, система перенаправит работу на проверку преподавателем.   
   Система не будет предоставлять автоматическую оценку работ студентов без участия других студентов или преподавателей.
3. Применения определяемого программного обеспечения, включая важные преимущества, объекты и цели:   
   Система "EduPlatform" будет использоваться в качестве платформы для организации и осуществления процесса обучения, взаимодействия студентов и преподавателей, а также контроля успеваемости студентов. Преимуществами для студентов является доступность и гибкость онлайн-обучения, а для преподавателей – упрощение процесса проверки работ и обратной связи со студентами.

1. Данный документ соответствует предыдущим утверждениям и высокоуровневым спецификациям требований, определенным для системы онлайн-обучения "EduPlatform".

## Термины, аббревиатуры, сокращения

|  |  |
| --- | --- |
| Система онлайн-обучения | Программное обеспечение, предоставляющее платформу для обучения, взаимодействия и оценки работ студентов и преподавателей в рамках курсов. |
| Студент | Пользователь системы, зарегистрированный на курсе с целью изучения и выполнения заданий. |
| Преподаватель | Пользователь системы, отвечающий за создание и управление курсами, заданиями и оценкой работ студентов. |
| Курс | Набор тем и образовательных материалов, предназначенных для обучения студентов в рамках системы. |
| Рецензент | Студент, выбранный системой для оценки работы другого студента. |
| EduPlatform | Название системы онлайн-обучения |
| СТПО | Спецификация требований программного обеспечения |
|  |  |
|  |  |

## Ссылки

|  |  |
| --- | --- |
| **Обозначение** | **Расшифровка** |
| [IEEE-830] | IEEE Std 830-1998 |

## Краткий обзор

Данный документ структурирован согласно [IEEE-830].

Раздел 2 содержит описание поставляемой системы и схему её использования в Организации. Раздел 3 содержит функциональные и нефункциональные требования, предъявляемые к системе и необходимые для её проектирования.  
  
Перспектива продукта заключается в том, чтобы сделать онлайн-обучение более качественным, гибким, доступным и персонализированным для студентов и преподавателей, интегрировать новые технологии и решения, постоянно развивать и совершенствовать функциональность и возможности системы в соответствии с изменяющимися потребностями пользователя и трендами образовательного процесса.  
  
**Перспективы системы:**  
1) Расширение функциональности платформы для поддержки различных типов контента, включая мультимедийные форматы, виртуальную реальность и дополненную реальность, чтобы обеспечить более интерактивные и гибкие образовательные возможности для студентов и преподавателей.

2) Использование методов искусственного интеллекта и машинного обучения для автоматического анализа и оценки выполненных работ, а также для обнаружения плагиата и фильтрации некачественных материалов.

3) Разработка мобильных приложений для платформы, доступных на разных операционных системах (iOS, Android), что позволит пользователям удобно и на ходу воспользоваться всеми возможностями системы.

4) Разработка и внедрение системы персонализированных рекомендаций для студентов, основанных на их предыдущих интересах, успехах и предпочтениях.

5) Улучшение системы оценки производительности преподавателей и качества обучения на основе рекомендаций и обратной связи от студентов.

6) Расширение взаимодействия с другими образовательными платформами и институтами для создания партнерских программ и обмена опытом.  
  
**Функции продукта:**

1) Регистрация и авторизация пользователей (студентов и преподавателей) на платформе

2) Создание и управление курсами, темами и заданиями преподавателями.

3) Выбор и просмотр курсов и тем студентами.

4) Загрузка выполненных работ студентами.

5) Автоматический выбор рецензентов для оценки работ.

6) Оценивание работ студентами (рецензентами) по заданной шкале (например, пятибалльной).

7) Расчёт итоговой оценки системой с учетом оценок рецензентов.

8) Передача работы преподавателю для оценивания в случае неоднозначных оценок.

9) Отображение результатов оценок студентам и преподавателям.

10) Формирование статистики и аналитики по курсам, студентам и преподавателям.

**Пользовательские характеристики:**

1) Студенты - основные пользователи системы, их функции включают выбор курсов, выполнение заданий и оценку работ других студентов.

2) Преподаватели - управляют курсами, темами и заданиями, проверяют работы в случаях неоднозначных оценок, просматривают статистику и аналитику по своим курсам и студентам.  
  
**Ограничения целостности:**

1) Механизмы проверки на плагиат должны быть встроены в систему для повышения качества предоставляемых работ.

2) Учетная запись преподавателя должна быть связана с одним или несколькими курсами, а учетная запись студента должна быть связана с выбранными студентом курсами.  
  
**Допущения и зависимости:**

1) Студенты и преподаватели имеют стабильный интернет-доступ для использования системы.

2) Все пользователи считаться компетентными в работе с компьютером и интернет-технологиями, имеющими достаточное знание темы для выполнения или оценки заданий.

**Подмножества требований:**

1) Возможность импорта и экспорта данных по курсам, студентам и оценкам

2) Интеграция с ресурсами образовательных материалов, видео и текстовых форматов.

3) Возможность коммуникации между студентами и преподавателями в рамках системы через личные сообщения или обсуждение тем на форуме.

4) Поддержка на платформе системы обратной связи для решения возникающих вопросов и конфликтных ситуаций.

5) Возможность интеграции со сторонними сервисами для авторизации (например, Google, UNN или VK Group).

# Общее описание

**Описание изделия:**

Система онлайн-обучения "EduPlatform" является веб-приложением, предназначенным для организации, проведения и управления курсами, позволяет студентам и преподавателям взаимодействовать, обмениваться знаниями и оценивать выполненные работы.

**Функции изделия**:

1. Регистрация и авторизация пользователей.

2. Создание, редактирование и управление курсами, темами и заданиями преподавателями.

3. Выбор, поиск и просмотр курсов и тем студентами.

4. Загрузка выполненных работ студентами.

5. Автоматический выбор рецензентов для оценки работ.

6. Оценивание работ студентами (рецензентами).

7. Рассчет итоговых оценок системой.

8. Передача работы преподавателю для оценивания в случае расхождения между оценками.

9. Отображение результатов оценок студентам и преподавателям.

10. Формирование статистики и аналитики по курсам, студентам и преподавателям.

**Характеристики пользователей:**

1. Студенты - основные пользователи платформы, взаимодействуют с курсами, темами, выполняют и оценивают работы.

2. Преподаватели - создают и управляют курсами, темами, заданиями, оценивают работы и контролируют образовательный процесс.

**Ограничения:**

1. Система должна быть совместима с различными браузерами и операционными системами.

2. Система должна поддерживать аутентификацию и авторизацию пользователей безопасным способом.

3. Система должна иметь защиту от несанкционированного доступа и атак.

**Предположения и зависимости:**

1. Все пользователи имеют стабильный доступ в интернет.

2. Все пользователи имеют достаточные знания и опыт для пользования системой.

3. Возможное изменение законодательных актов или технологических требований, влияющих на систему.

**Поднаборы требований:**

1. Возможность работы с различными типами контента (текст, изображения, видео, презентации).

2. Механизмы для обнаружения плагиата и проверки качества предоставленных работ.

3. Возможность масштабирования системы для удовлетворения возрастающих потребностей пользователей.

4. Интеграция с сервисами и системами сторонних разработчиков API или авторизации

5. Поддержка мобильных устройств, включая создание соответствующих мобильных приложений.

## Описание изделия

Система "EduPlatform" является независимым и полностью автономным продуктом, предназначенным для онлайн-обучения студентов и взаимодействия студентов с преподавателями. Однако, данная система может быть впоследствии расширена и интегрирована с другими продуктами, связанными с онлайн-обучением и образовательными платформами.

**Интерфейсы системы:**

Система будет взаимодействовать с веб-сервером и базой данных для хранения информации о студентах, преподавателях, курсах и работах студентов.

**Интерфейсы пользователя:**

Пользовательский интерфейс будет представлен в виде веб-приложения, доступного через браузер и адаптированного для различных устройств (компьютеры, планшеты, смартфоны).

**Интерфейсы аппаратных средств ЭВМ:**

Система будет работать на любых устройствах, поддерживающих современные веб-браузеры и имеющих доступ в Интернет.

**Интерфейсы программного обеспечения:**

Система "EduPlatform" будет взаимодействовать с серверной частью, реализованной на основе веб-сервера и базы данных. Для разработки клиентской части могут быть использованы современные фреймворки и библиотеки.

**Интерфейсы коммуникаций:**

Система будет использовать протоколы HTTP/HTTPS для передачи данных между клиентом и сервером.

**Ограничения памяти:**

Требования к памяти зависят от количества пользователей, курсов и работ, которое система должна обрабатывать и хранить. Однако, современные технологии облачных хранилищ и оптимизации баз данных помогут снизить нагрузку на память.

**Функционирование (Operations):**

Система должна обеспечивать непрерывное и стабильное функционирование для всех пользователей. Возможны периодические технические работы для обновления и обслуживания системы.

**Требования настройки рабочих мест:**

Система будет работать на любых устройствах с поддержкой современных веб-браузеров и доступом в Интернет, без дополнительных требований к установке специального программного обеспечения.

### Интерфейсы системы

**1. Веб-сервер:** ответственный за обработку запросов от клиентов и обеспечение взаимодействия с базой данных. Веб-сервер будет реализован на основе современных технологий, таких как Node.js, Ruby on Rails, Django или PHP.

**2. База данных:** Ответственная за хранение информации о студентах, преподавателях, курсах и работах студентов. Будет использоваться реляционная или нереляционная база данных в зависимости от предпочтений разработчиков и требований системы (например, MySQL, PostgreSQL, MongoDB).

**3. API (Application Programming Interface):** предоставляет стандартизованный набор методов для обращения клиентской части системы к серверной части. API будет разработано с использованием RESTful принципов и обеспечит интеграцию между клиентом и сервером через JSON формат.

**4. Клиентская часть системы:** веб-приложение, доступное через браузер и отвечающее за взаимодействие студентов и преподавателей с системой. Клиентская часть будет разработана с использованием современных технологий (React, Angular, Vue.js) и будет адаптирована для различных устройств (компьютеры, планшеты, смартфоны).

**5. Сервис аутентификации:** отвечает за проверку идентификационных данных пользователей и обеспечивает безопасность системы. Сервис аутентификации может использовать стандартные протоколы, такие как OAuth 2.0, для интеграции с внешними сервисами (например, Google, UNN, VK Group).

**6. Сервис уведомлений:** отвечает за информирование студентов и преподавателей о новых событиях, связанных с курсами, работами и оценками. Уведомления могут быть отправлены по электронной почте, внутри системы или через мобильное приложение.

Каждый из упомянутых интерфейсов должен быть проектирован и реализован с применением лучших практик разработки программного обеспечения и учитывая требования системы, пользователей и бизнес-логики. Взаимодействие между интерфейсами должно быть оптимизировано для обеспечения высокой производительности, масштабируемости и стабильности системы.

### Интерфейсы пользователя Здесь необходимо определить следующие характеристики пользовательских интерфейсов:

**a) Логические характеристики каждого интерфейса:**

- Адаптивный дизайн: интерфейс должен корректно отображаться на различных устройствах (компьютеры, планшеты, смартфоны) и разрешениях экрана.

- Интуитивная навигация: меню, кнопки, и ссылки должны быть легко доступными и понятными для пользователей без дополнительных объяснений (например, использование иконок с подписями).

- Вывод информации: система должна предоставлять информацию о курсах, работах студентов и оценках в удобном и структурированном виде (например, таблицы, списки, графики).

- Сообщения об ошибках и прогрессе: система должна информировать пользователей о возможных ошибках (например, неправильно заполненные поля или сбои) и о статусе обработки запросов (например, индикация загрузки).

**b) Аспекты оптимизации интерфейса:**

- Минимизация количества шагов для выполнения задач: интерфейс должен быть разработан таким образом, чтобы пользователи могли выполнять свои задачи за минимальное количество шагов и времени.

- Дружественность к пользователю: интерфейс должен быть привлекательным, удобным, и разработанным с учетом особенностей целевой аудитории (например, возраст, опыт использования подобных систем).

- Доступность функций: необходимые функции должны быть легко доступными и понятными для пользователей. При возникновении ошибок система должна предоставлять краткие и полные объяснения об ошибках, чтобы помочь пользователям устранить проблему без затруднений.

- Обучаемость: пользователи должны быстро осваивать систему. Например, студент должен способен загрузить и отредактировать свою работу за 10 минут после 30 минут обучения.

Таким образом, интерфейсы пользователя системы "EduPlatform" должны быть разработаны с учетом требований и ожиданий пользователей, а также с последующей проверкой и анализом пользовательского опыта для оптимизации взаимодействия с системой.

### Интерфейсы аппаратных средств ЭВМ

Для корректной работы системы "EduPlatform" необходимо определить логические характеристики каждого взаимодействия между программным обеспечением и аппаратными компонентами системы. В данном случае, поскольку система представляет собой веб-приложение, основные требования к аппаратным интерфейсам направлены на поддержку работы веб-браузеров и подключение к Интернету.

**1. Поддержка веб-браузеров:** система должна корректно работать в современных веб-браузерах, таких как Google Chrome, Mozilla Firefox, Safari, и Microsoft Edge. Браузеры должны поддерживать последние стандарты и технологии веб-разработки, такие как HTML, CSS, и JavaScript.

**2. Интернет-соединение:** для доступа к системе необходимо наличие стабильного интернет-соединения со скоростью, достаточной для загрузки веб-страниц, обработки запросов, и передачи файлов.

**3. Распределение нагрузки и оптимизация производительности:** серверная часть системы должна быть оптимизирована для работы с различными аппаратными решениями, включая облачные сервера и по возможности адаптироваться к изменяемой нагрузке со стороны пользователей.

**4. Интеграция с аппаратными компонентами:** веб-приложение должно корректно работать с различными устройствами ввода (клавиатура, мышь, сенсорный экран) и воспроизведения мультимедиа (аудио и видео).

**5. Поддержка мобильных устройств:** система должна быть адаптирована для работы на мобильных устройствах, таких как смартфоны и планшеты, с различными разрешениями экрана и операционными системами (iOS, Android).

В целом, интерфейсы аппаратных средств ЭВМ для системы "EduPlatform" должны быть максимально адаптированы к современным технологиям и устройствам пользователей для обеспечения качественного функционирования системы и удобства ее использования.

### Интерфейсы программного обеспечения

Для корректной работы и взаимодействия системы "EduPlatform" с другими программными компонентами, необходимо определить следующие интерфейсы программного обеспечения:

**1. Система управления базами данных** (Database Management System, DBMS):

a) Наименование: MySQL / PostgreSQL / MongoDB (в зависимости от выбранной базы данных)

b) Мнемоническое наименование: DB

c) Номер спецификации: зависит от версии и документации выбранной DBMS

d) Номер версии: ссылаться на последнюю стабильную версию выбранной DBMS

e) Источник: официальный сайт выбранной DBMS (например, mysql.com, postgresql.org, mongodb.com)

**2. Веб-сервер:**

a) Наименование: Node.js / Ruby on Rails / Django / PHP (в зависимости от выбранной технологии серверной части)

b) Мнемоническое наименование: WebServer

c) Номер спецификации: зависит от версии и документации выбранного веб-сервера

d) Номер версии: ссылаться на последнюю стабильную версию выбранного веб-сервера

e) Источник: Официальный сайт выбранного веб-сервера (например, nodejs.org, rubyonrails.org, djangoproject.com, php.net)

**3. Фреймворк клиентской части:**

a) Наименование: React / Angular / Vue.js (в зависимости от выбранной технологии клиентской части)

b) Мнемоническое наименование: FrontendFramework

c) Номер спецификации: зависит от версии и документации выбранного фреймворка

d) Номер версии: ссылаться на последнюю стабильную версию выбранного фреймворка

e) Источник: Официальный сайт выбранного фреймворка (например, reactjs.org, angular.io, vuejs.org)

Для каждого интерфейса:

**a) Назначение взаимодействующих программ:**

- DBMS: Обеспечение хранения, обработки и доступа к данным системы.

- WebServer: Обработка запросов клиентской части, обеспечение взаимодействия с базой данных и другими сервисами.

- FrontendFramework: Реализация пользовательского интерфейса и взаимодействие с серверной частью через API.

**b) Определение интерфейса в терминах содержания и формата сообщений:**

- DBMS: SQL-запросы для реляционных баз данных (MySQL, PostgreSQL) или JSON-запросы для нереляционных баз данных (MongoDB).

- WebServer: RESTful API с использованием JSON для обмена данными между клиентской и серверной частями.

- FrontendFramework: HTTP/HTTPS запросы к API серверной части и обработка полученных данных для отображения пользователю.

Таким образом, система "EduPlatform" предполагает интеграцию и взаимодействие с рядом сторонних программных компонентов, обеспечивающих ее корректную работу и удобство использования.

### Интерфейсы коммуникаций

В системе "EduPlatform" используются следующие интерфейсы коммуникаций и протоколы для поддержания взаимодействия между компонентами системы и обеспечения ее корректной работы:

**1. Протокол HTTP/HTTPS:** используется для передачи данных между клиентской частью (веб-браузер пользователей) и серверной частью системы (веб-сервер, API). Протокол HTTPS обеспечивает шифрование данных и безопасность взаимодействия.

**2. WebSocket:** дополнительно может использоваться для обеспечения двунаправленного общения между клиентом и сервером в реальном времени, например, для функций чата или обновлений информации без необходимости перезагрузки страницы.

**3. Протоколы сетевого уровня (TCP/IP):** используются для обеспечения обмена данными между серверами, хостами и другими сетевыми устройствами, а также для поддержания стабильной работы системы.

### Ограничения памяти

Ограничения памяти для системы "EduPlatform" зависят от количества пользователей, курсов, работ и других данных, которые система должна обрабатывать и хранить. В качестве ориентиров для определения ограничений памяти, могут быть использованы следующие параметры:

**1. Оперативная память:** необходимый объем оперативной памяти (RAM) для сервера зависит от нагрузки и масштабируемости системы. Рекомендуется мониторить использование памяти и при необходимости увеличивать ее объем для обеспечения стабильности работы системы.

**2. Постоянная память:** размер требуемого дискового пространства для базы данных и хранения пользовательских данных (файлы, изображения) зависит от количества зарегистрированных пользователей, созданных курсов, и загруженных работ. Рекомендуется использовать технологии сжатия данных, оптимизации баз данных и облачные хранилища для экономии пространства и обеспечения доступности данных.

**3. Память клиентского устройства:** система должна быть оптимизирована для работы на устройствах с различным объемом оперативной памяти и дискового пространства. Клиентская часть системы должна корректно работать на устройствах с низким объемом памяти, минимизировать использование памяти и не создавать наращивания данных за счет кеширования или частичной загрузки информации при навигации.

Таким образом, система "EduPlatform" должна быть разработана с учетом возможных ограничений памяти и адаптирована к различным условиям использования для обеспечения стабильности и производительности.

### Действия

В "EduPlatform" предусмотрены следующие типы действий:

**a) Различные способы действий в организации пользователя:**

- загрузка работ студентами

- выбор студентов-рецензентов системой

- оценка работ рецензентами

- расчет итоговой оценки системой

- перенаправление работы для проверки преподавателю, если оценки сильно различаются

- получение оценок и отзывов студентами

- мониторинг успеваемости и предоставление отчетов преподавателям

**b) Периоды диалоговых действий и периоды оставленных без отклика действий:**

- периоды активного взаимодействия пользователей с системой (при работе с оценками и отзывами)

- периоды ожидания обработки данных системой (при расчете итоговой оценки, подборе рецензентов и отправке работы на проверку преподавателю)

**c) Функции поддержки обработки данных:**

- хранение загруженных работ, оценок и отзывов

- статистический анализ данных для определения средних оценок, распределения оценок и выявления аномалий

- создание резюме результатов для студентов и преподавателей

**d) Действия резервного копирования и восстановления:**

- автоматическое резервное копирование данных о загруженных работах, оценках и отзывах

- возможность восстановления данных в случае потери или повреждения

- создание архивных копий результатов текущего периода обучения для использования в будущем

Таким образом, "EduPlatform" будет выполнены действия, связанные с процессами оценивания работ студентами, их анализом и предоставлением результатов обратной связи пользователям. Кроме того, система обеспечивает безопасность и сохранность данных, что является одной из важных функций для успешной работы образовательной платформы.

### Требования настройки рабочих мест

a) Для корректной работы "EduPlatform" требуется определить следующие требования:

- разрешение экрана: оптимальное разрешение экрана для комфортной работы с системой должно быть не менее 1280x720 пикселей (телефоны, планшеты, компьютеры, ноутбуки)

- используемый браузер: обеспечить совместимость системы с наиболее распространенными браузерами (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge)

- скорость интернет-соединения: минимальная рекомендуемая скорость интернет-соединения должна быть не менее 1 Mbps для стабильной работы системы

b) Характеристики рабочего места или решаемых задач, которые должны быть изменены для настройки программного обеспечения на специальную конфигурацию:

- операционная система: убедиться, что рабочие места пользователей оснащены поддерживаемыми операционными системами (Windows, macOS, Linux)

- антивирусное ПО и файервол: настройка защитных систем таким образом, чтобы разрешить обмен данными между рабочими местами пользователей и системой "EduPlatform"

- программное обеспечение для просмотра и редактирования документов типов, используемых в "EduPlatform" (например, Microsoft Office, Adobe Acrobat Reader или их аналоги)

- настройка прокси-сервера или VPN соединения для безопасного доступа к системе, если это необходимо это для выполнения корпоративных политик безопасности

В целом, настройка рабочих мест пользователей "EduPlatform" должна быть сфокусирована на обеспечении комфортных условий для работы с системой, поддержании совместимости программного обеспечения и обеспечении надежной и безопасной связи с платформой.

## Функции изделия

В этом разделе представлены основные функции, выполняемые программным обеспечением "EduPlatform". Функции организованы в группы для обеспечения понятности структуры.

**1. Управление курсами и пользователями:**

a) Создание и редактирование курсов и учебных модулей.

b) Управление доступом студентов и преподавателей к курсам.

c) Мониторинг успеваемости студентов и предоставление отчетов.

**2. Загрузка и хранение работ студентов:**

a) Предоставление пользовательского интерфейса для загрузки работ студентами.

b) Хранение загруженных работ в облачном хранилище.

c) Поддержка различных типов файлов для загружаемых работ.

**3. Оценивание работ студентами и системой:**

a) Автоматический выбор рецензентов для оценки работ.

b) Предоставление интерфейса для студентов-рецензентов для оценки и предоставления отзывов.

c) Расчет итоговой оценки на основе оценок рецензентов и определение дальнейшего действия (например, пересылка преподавателю в случае сильно различающихся оценок).

**4. Обратная связь и коммуникация:**

a) Получение оценок и отзывов студентами.

b) Функция обсуждения работ и отзывов между студентами и преподавателями.

c) Возможность получения поддержки и решения технических проблем.

Таким образом, "EduPlatform" управляет процессом обучения и взаимодействия студентов и преподавателей, организует проверку и оценку работ студентов, обеспечивает хранение данных и позволяет обмениваться отзывами для достижения успеха в обучении.

## Характеристики пользователей

Рассмотрим общие характеристики пользователей, для которых предназначена "EduPlatform":

**1. Студенты:**

- Образовательный уровень: студенты могут иметь различный образовательный уровень, включая среднее, среднее специальное, высшее и послевузовское образование.

- Опыт: студенты могут не иметь опыта работы в своей области или наоборот, обладать определенным профессиональным опытом.

- Техническая квалификация: студенты должны обладать базовыми знаниями и навыками в работе с компьютером, интернетом и офисными приложениями.

**2. Преподаватели:**

- Образовательный уровень: преподаватели должны иметь высшее образование, а также возможно, научные степени и звания.

- Опыт: преподаватели должны обладать опытом работы в своей области, а также опытом преподавания и наставничества.

- Техническая квалификация: преподаватели должны иметь продвинутые навыки в работе с компьютером, интернетом и офисными приложениями, а также опыт использования образовательных платформ.

Определение характеристик пользователей помогает обеспечить наиболее удобный и доступный интерфейс, а также оптимальный набор функциональных возможностей "EduPlatform". Эти характеристики будут влиять на требования к системе и на формирование функций, рассмотренных в разделе 3 СТПО.

## Ограничения

В этом разделе приведем общее описание факторов, ограничивающих выбор разработчика для "EduPlatform":

**a) Регулирующая политика:** соблюдение правил и норм местного законодательства, а также правил и политик, принятых в образовательных учреждениях, которые будут использовать систему.

**b) Ограничения аппаратных средств ЭВМ:** система должна быть готова к работе на различных устройствах (ПК, мобильные устройства) с определенными минимальными и рекомендуемыми системными требованиями.

**c) Интерфейсы с другими приложениями:** "EduPlatform" должна обеспечивать интеграцию с внешними системами, такими как системы управления базами данных, системы аутентификации пользователей, а также популярные службы облачного хранения.

**d) Параллельная работа:** система должна поддерживать одновременную работу нескольких пользователей с обеспечением надежного доступа и сохранности данных.

**e) Функции протоколирования:** обеспечить регистрацию действий пользователей и системных событий для контроля и анализа работы системы.

**f) Функции управления:** наличие административных функций для управления и контроля доступа.

**g) Требования к языкам высокого уровня:** система должна быть разработана на современных и распространенных языках программирования, например, Python, Java или JavaScript.

**h) Протоколы интерфейсов синхронизации сигналов:** система должна поддерживать соответствующие протоколы для обеспечения стабильного пересылки данных и коммуникаций между пользователями и сервером.

**i) Требования надежности:** "EduPlatform" должна обеспечивать высокую надежность и доступность для пользователей, а также устойчивость к различным типам сбоев.

**j) Критичность приложения:** система должна поддерживать высокие стандарты, потому что образовательный процесс и карьера студентов зависят от качества работы "EduPlatform".

**k) Соображения безопасности и секретности:** обеспечение защиты и конфиденциальности персональных данных пользователей, а также интеллектуальной собственности, связанной с контентом курсов и проверяемыми работами.

## Предположения и зависимости

В этом разделе определены факторы, которые влияют на требования "EduPlatform", но не являются непосредственными ограничениями. Изменения в этих факторах могут повлиять на требования, представленные в документе СТПО.

**1. Операционная система:** предполагается, что на аппаратных средствах ЭВМ пользователей будет установлена совместимая операционная система (Windows, macOS, Linux). Если доступна другая операционная система, требования к "EduPlatform" могут измениться.

**2. Сетевая инфраструктура:** предполагается наличие стабильного доступа пользователей к интернету для работы с "EduPlatform". Изменение качества или доступности интернет-соединения может повлиять на функциональность системы и удовлетворение требований пользователей.

**3. Технологические инструменты:** предполагается использование современных инструментов разработки программного обеспечения, фреймворков и библиотек, которые обеспечивают быстрое и качественное развертывание "EduPlatform". Обновление или замена используемых технологий может изменить требования к системе.

**4. Стандарты обучения и нормативные документы:** система основана на существующих стандартах обучения и нормативных документах, которые определяют образовательные процессы. Изменения в этих стандартах и документах могут повергнуть требования к "EduPlatform".

**5. Интеграция с внешними сервисами:** "EduPlatform" предполагает возможность интеграции с различными внешними сервисами, например, системами облачного хранения или аутентификации пользователей. Изменения в доступности или спецификациях этих сервисов могут влиять на требования к системе.

Оценка предположений и зависимостей позволяет адаптироваться к возможным изменениям, а также предоставляет возможность для пересмотра и корректировки требований к "EduPlatform" при необходимости.

## Распределение требований

В данном разделе приводятся требования, реализация которых может быть отложена до будущих версий системы "EduPlatform".

**1. Мобильное приложение:** хотя основной функционал "EduPlatform" должен быть расположен в веб-приложении, доступном через интернет-браузер, в последующих версиях можно разработать полноценное мобильное приложение для улучшения доступности и удобства использования системы на мобильных устройствах.

**2. Интеграция с онлайн-сервисами для совместной работы:** в будущих версиях системы можно добавить возможность интеграции с популярными онлайн-сервисами для совместной работы над документами или проектами, например, Google Docs или Microsoft Office 365.

**3. Расширение функционала обратной связи:** добавление возможностей видеоконференции, текстового и голосового чатов, чтобы студенты и преподаватели могли обмениваться мнениями и знаниями в режиме реального времени.

**4. Встроенные учебные ресурсы и курсы:** в дополнение к создаваемым академическими модулям, "EduPlatform" может предоставить доступ к каталогу внешних образовательных ресурсов и курсов, например, интеграция с ресурсами открытого образования.

**5. Система уведомлений:** реализация сложной системы уведомлений для пользователей о новых событиях или действиях, связанных с образовательным процессом и их курсами.

**6. Поддержка нескольких языков:** возможность использования "EduPlatform" пользователями, говорящими на различных языках с добавлением переведенного интерфейса и контента.

Учет возможности распределения требований позволяет разработчикам системы сфокусироваться на реализации ключевых функций в первой версии "EduPlatform", а также планировать долгосрочное развитие и улучшение продукта.

# Детальные требования

**Входы системы**

a) Регистрация студентов и преподавателей: Система должна позволять пользователям регистрироваться в роли студента или преподавателя с использованием адреса электронной почты, имени пользователя и пароля.

b) Загрузка работ студентов: Система должна предоставлять студентам возможность загружать их работы в формате .doc, .docx, png, jpg или .pdf.

c) Выбор студентов для оценки: Система должна автоматически выбирать некоторое количество студентов для оценивания загруженной работы.

**Выходы системы**

a) Итоговая оценка: Система должна рассчитывать итоговую оценку для работы студента на основе оценок, выставленных другими студентами.

b) Отправка результатов на проверку преподавателю: если разброс оценок слишком велик, система должна автоматически отправлять работу на проверку преподавателю.

c) Обратная связь между студентами и преподавателями: Система должна обеспечивать возможность обмена сообщениями между студентами и преподавателями.

**Функции системы**

a) Управление пользователями: Система должна обеспечивать управление учетными записями студентов и преподавателей, включая регистрацию, вход в систему, изменение личной информации и удаление учетной записи.

b) Управление курсами: Система должна обеспечивать создание, обновление и удаление курсов преподавателями, а также регистрацию студентов на выбранные курсы.

c) Управление работами студентов: Система должна обеспечивать загрузку, оценивание и хранение работ студентов, а также передачу результатов оценивания студентам и преподавателям.

d) Обеспечение взаимодействия: Система должна предоставлять инструменты для обмена сообщениями и обратной связи между студентами и преподавателями.

**Идентификация требований**

a) Все детальные требования должны быть идентифицированы с уникальными идентификаторами, начиная с "REQ-001" и далее по порядку.

b) Ссылки на соответствующие разделы других документов должны быть предоставлены для каждого требования, где это применимо.

**Структурирование требований**

Для повышения читаемости требования должны быть структурированы и организованы в соответствии с разделами входов системы, выходов системы и функций системы. Следует использовать списки и таблицы для упрощения восприятия информации.

## Внешние интерфейсы

1. Входные данные (Inputs):

a) Наименование пункта: Регистрационная информация

b) Описание цели: Сбор информации о студентах и преподавателях для их регистрации в системе.

c) Источник входных данных: Пользователь

d) Диапазон допустимых значений: Адрес электронной почты, имя пользователя и пароль должны быть заполнены.

e) Единицы измерения: N/A

f) Временные характеристики: N/A

g) Отношения к другим входам / выходам: N/A

h-j) Форматы / организация экрана и окна: Форма регистрации с полями для ввода электронной почты, имени пользователя и пароля.

k) Формат данных: строка

l) Конечные сообщения: Сообщение об успешной регистрации или ошибке.

2. Входные данные (Inputs):

a) Наименование пункта: Загрузка работы студента

b) Описание цели: позволить студентам отправлять свои работы для проверки и оценивания.

c) Источник входных данных: Студент

d) Диапазон допустимых значений: Файлы форматов .doc, .docx или .pdf размером не более 10 МБ.

e) Единицы измерения: N/A

f) Временные характеристики: N/A

g) Отношения к другим входам / выходам: Связь с оценками других студентов и итоговой оценкой.

h-j) Форматы / организация экрана и окна: Форма загрузки файла с возможностью выбора файла и кнопкой "Отправить".

k) Формат данных: Файл (.doc, .docx, .pdf)

l) Конечные сообщения: Сообщение об успешной загрузке или ошибке.

3. Выходные данные (Outputs):

a) Наименование пункта: Итоговая оценка работы студента

b) Описание цели: информировать студента о полученной оценке за работу.

c) Назначение выходных данных: Студент

d) Диапазон допустимых значений: Оценка от 1 до 5 баллов.

e) Единицы измерения: Баллы

f) Временные характеристики: сразу после объявления результатов.

g) Отношения к другим входам / выходам: Связь с загруженной работой студента и оценками от других студентов.

h-j) Форматы / организация экрана и окна: Вывод текста с полученной оценкой или сообщением о передаче работы на проверку преподавателю.

k) Формат данных: Число (1-5)

l) Конечные сообщения: Сообщение с итоговой оценкой или передачей работы на проверку преподавателю.

4. Выходные данные (Outputs):

a) Наименование пункта: Обратная связь между студентами и преподавателями

b) Описание цели: обеспечить возможность обмениваться сообщениями между студентами и преподавателями.

c) Назначение выходных данных: Студенты и преподаватели

d) Диапазон допустимых значений: Текст сообщения не должен превышать 1000 символов.

e) Единицы измерения: N/A

f) Временные характеристики: непосредственно после отправки сообщения.

g) Отношения к другим входам / выходам: Связь с отправленными сообщениями другими участниками.

h-j) Форматы / организация экрана и окна: Интерфейс обмена сообщениями с отображением истории сообщений, поля для ввода текста и кнопкой "Отправить".

k) Формат данных: Текст

l) Конечные сообщения: Сообщения между студентами и преподавателями.

Дополнительно, система может иметь следующие внешние интерфейсы:

5. Входные данные (Inputs):

a) Наименование пункта: авторизационные данные

b) Описание цели: Идентификация пользователя при входе в систему.

c) Источник входных данных: Пользователь

d) Диапазон допустимых значений: Имя пользователя и пароль должны быть заполнены и соответствовать зарегистрированным данным.

e) Единицы измерения: N/A

f) Временные характеристики: N/A

g) Отношения к другим входам / выходам: N/A

h-j) Форматы / организация экрана и окна: Форма авторизации с полями для ввода имени пользователя и пароля, а также кнопкой "Войти".

k) Формат данных: строка

l) Конечные сообщения: Сообщение об успешном входе или ошибке.

6. Выходные данные (Outputs):

a) Наименование пункта: Список курсов

b) Описание цели: Отображение доступных курсов для студентов и преподавателей.

c) Назначение выходных данных: Студенты и преподаватели

d) Диапазон допустимых значений: Все доступные в системе курсы.

e) Единицы измерения: N/A

f) Временные характеристики: сразу после запроса пользователя.

g) Отношения к другим входам / выходам: Связь с информацией о зарегистрированных на курсах студентах и преподавателях.

h-j) Форматы / организация экрана и окна: Таблица или список с информацией о названии курса, преподавателе и количестве зарегистрированных студентов.

k) Формат данных: Текст, изображение.  
l) Конечные сообщения: N/A.

## Функции

1. **Проверка допустимости данных на входах**

a) Регистрация: Система должна проверять корректность вводимых данных пользователем при регистрации, таких как наличие адреса электронной почты, имени пользователя и пароля.

b) Авторизация: Система должна проверять наличие существующего имени пользователя и соответствующего пароля при попытке входа в систему.

c) Загрузка работ: Система должна проверять корректный формат файла (doc, docx, png, jpg, pdf) и размер файла (не более 10 МБ) при загрузке работы студента.

**2 Точная последовательность действий**

a) Регистрация студента или преподавателя в системе

b) Вход в систему с использованием зарегистрированных учетных данных

c) Выбор курсов для участия (для студентов) или создание и управление курсами (для преподавателей)

d) Загрузка работ студента для оценки

e) Оценивание работ студентами и выставление итоговых оценок

f) Обратная связь между студентами и преподавателями

**3. Действия при возникновении исключительных ситуаций**

a) Переполнение: Система должна обрабатывать ситуации переполнения данных (например, слишком большой файл при загрузке работы), отображая соответствующее сообщение об ошибке для пользователя.

b) Средства связи: Система должна обеспечивать стабильное и надежное соединение между сервером и пользовательским интерфейсом, обрабатывая проблемы со связью и информируя пользователя о возможных ошибках.

c) Обработка ошибок и восстановление: Система должна предоставлять механизмы для обработки ошибок и восстановления данных, обеспечивая сохранность данных пользователей и надежность сервиса.

**4. Влияние параметров**

Система должна определять количество выбираемых студентов для оценки работ, определение порогов разброса оценок для передачи работы на проверку преподавателю.

**5. Отношения входных данных к выходным данным**

a) Последовательность входных данных / выходных данных: Система должна обеспечивать непрерывность взаимодействия входных и выходных данных, таких как загрузка работы, оценка и выдача итоговой оценки.

b) Формулы для преобразования входных данных в выходные: Система должна рассчитывать итоговые оценки на основе оценок, выставленных студентами-рецензентами, а также определение разброса и передачу работы на проверку преподавателю.

## Требования исполнения

a) Число терминалов, которое должны быть поддержаны: Система должна иметь возможность поддерживать одновременное подключение и использование до 500 терминалов.

b) Количество одновременно работающих пользователей, которое должно быть поддержано: Система должна обеспечивать стабильную работу при одновременном использовании до 300 студентов и 50 преподавателей.

c) Количество и тип информации, которая должна обрабатываться: Система должна иметь возможность обрабатывать различные типы информации, такие как текстовые данные (имена пользователей, пароли, наименования курсов), мультимедиа (загружаемые работы студентов) и пользовательские сообщения.

**Требования для статических численных характеристик:**

1. Нагрузка системы: Система должна иметь возможность поддерживать стабильную работу при нагрузке до 1000 активных курсов и до 10 000 зарегистрированных пользователей (студентов и преподавателей).

**Требования для динамических численных характеристик:**

1. Процент успешных сделок: Система должна обеспечивать успешную обработку минимум 95% всех пользовательских операций, таких как загрузка работ, отправка сообщений и изменение настроек профиля.

2. Время ответа системы: 95 % всех запросов к системе должны быть обработаны в течение 1 секунды.

3. Пиковая рабочая нагрузка: Система должна обеспечивать стабильную работу и надежность при пиковой рабочей нагрузке (например, в периоды сдачи работ и экзаменов). В пределах 10-минутных интервалов система должна быть способна обработать до N/A транзакций и до N/A МБ загружаемых данных.

Все эти требования должны быть заявлены в установленных единицах измерения. Например, "95 % сделок должны быть обработаны меньше чем 1 с" предпочтительнее чем "оператор не должен ждать, чтобы завершить сделку".

## Требования логики базы данных

a) Типы информации, используемые различными функциями: База данных должна хранить данные о пользователях (студентах и преподавателях), курсах, загружаемых работах студентов, оценках и обратной связи.

b) Частота использования: База данных должна обеспечивать быстрый и надежный доступ к информации для поддержания непрерывной работы системы. Операции чтения могут возникать часто в случае просмотра курсов и получения обратной связи, в то время как операции записи могут возникать реже, например при создании курсов или загрузке работ.

c) Способы доступа: Система должна предоставлять различные способы доступа к данным в базе данных, такие как добавление, удаление, обновление, чтение данных и поиск.

d) Сущности данных и их отношения: База данных должна включать следующие сущности: пользователи (студенты, преподаватели), курсы, работы, оценки и сообщения обратной связи. Отношения между этими сущностями должны быть определены для логической структуры базы данных.

e) Ограничения целостности: Ограничения целостности должны обеспечивать правильность и надежность хранения данных. Например, оценка за работу должна быть числом от 1 до 5, а каждый пользователь должен иметь уникальный идентификатор.

f) Требования хранения данных: База данных должна обеспечивать безопасное и надежное хранение информации, а также возможность восстановления данных в случае сбоя системы или потери информации. Требования хранения могут включать резервное копирование данных, уровень избыточности и стратегии оптимизации производительности.

## Ограничения проекта

В разработке системы "EduPlatform" будут присутствовать определенные ограничения, которые следует учесть в процессе проектирования и реализации. Они включают:

**1. Соблюдение стандартов:** Разработка программного обеспечения должна соответствовать принятым стандартам в области образования и технологии в обучении (например, SCORM, xAPI). Это позволит обеспечить совместимость с другими системами и ресурсами.

**2. Ограничения аппаратных устройств:** Разработка должна учитывать различные характеристики и возможности аппаратного обеспечения пользователей, таких как ноутбуки, планшеты и смартфоны. Программное обеспечение должно быть адаптировано для работы на различных устройствах и платформах с учетом ограничений производительности, ресурсов и программного обеспечения.

**3. Оптимизация и масштабируемость:** Система должна быть оптимизирована для обеспечения быстрой работы и возможности масштабирования по мере роста количества пользователей и курсов, предоставляемых в рамках платформы.

**4. Безопасность и конфиденциальность:** Проект должен учитывать аспекты безопасности и защиты данных пользователей, работы и результатов оценивания. Система должна предусматривать меры по защите от несанкционированного доступа и манипуляций с данными сторонами.

**5. Веб-доступность и кросс-браузерность:** Обеспечить доступность разработанного программного обеспечения для пользователей с различными настройками и возможностями веб-браузеров, а также применять стандарты веб-доступности для удовлетворения потребностей людей с ограниченными возможностями.

**6. Интеграция с существующими системами:** Учитывать возможность интеграции с другими образовательными ресурсами и платформами, такими как системы управления курсами (LMS), авторизации и аутентификации пользователей, системы аналитики и отчетности.

С учетом этих ограничений, команда разработчиков будет стремиться к созданию инновационной, удобной в использовании и надежной системы для обучения и взаимодействия студентов и преподавателей в более эффективной и контролируемой среде.

### Соглашение о стандартах

Для системы "EduPlatform" требуется обеспечить соблюдение существующих стандартов и инструкций, которые могут включать:

**a) Формат сообщений:** разработка программного обеспечения должна соответствовать стандартам обмена данными и сообщениями, таким как JSON и XML. Это позволит обеспечить совместимость и корректное взаимодействие с другими системами и ресурсами, а также облегчит интеграцию с ними.

**b) Именование данных:** система должна использовать единые принципы именования переменных, функций, классов и других элементов программного кода. Это позволит обеспечить консистентность и упростит процедуру поддержки и доработок программного обеспечения.

**c) Отчетные процедуры:** в рамках проведения финансовых операций в системе (например, оплата курсов или подписки) необходимо соблюдать принятые стандарты и процедуры для обработки и учета транзакций, включая требования безопасности, защиты и анонимности данных пользователей.

**d) Протоколирование:** система должна обеспечивать протоколирование всех важных операций, связанных с процессами обучения, оценивания, взаимодействия студентов и преподавателей, а также финансовых и административных операций. Протоколирование должно позволять восстановить последовательность действий, хранить и анализировать данные вторичного характера для улучшения качества и контроля процессов, а также для обеспечения безопасности и возможности аудита.

Таким образом, обеспечивая соблюдение вышеуказанных стандартов и руководящих принципов, система "EduPlatform" будет соответствовать требованиям качества, совместимости и безопасности, что сделает ее доступной и эффективной для студентов и преподавателей.

## Характеристики программного обеспечения системы

Программное обеспечение системы "EduPlatform" должно соответствовать определенным характеристикам, чтобы обеспечить его эффективность, надежность и удобство использования. Вот некоторые из них:

**1. Надежность:** система должна быть стабильной и устойчивой к ошибкам, обеспечивая бесперебойную работу и сохранение данных пользователей. В рамках этой характеристики важны такие показатели, как время наработки на отказ и предотвращение потери информации, а также логирование.

**2. Масштабируемость:** программное обеспечение должно обеспечивать возможность быстрого и гибкого масштабирования в связи с ростом количества пользователей, курсов и другого образовательного контента, а также в случае изменения требований к ресурсам и инфраструктуре.

**3. Производительность:** система должна обеспечивать быструю и эффективную работу, минимизируя задержки и обеспечивая оптимальное использование ресурсов. Важными показателями производительности являются время отклика и пропускная способность.

**4. Безопасность:** система "EduPlatform" должна обеспечивать защиту данных пользователей, преподавателей и студентов, а также гарантировать конфиденциальность и защиту всех проводимых операций. Это включает в себя аутентификацию пользователей, авторизацию доступа, шифрование данных и превентивные мероприятия против взлома и других угроз.

**5. Удобство использования:** "EduPlatform" должна обеспечивать интуитивно понятный и привлекательный пользовательский интерфейс, в котором каждый пользователь сможет найти необходимые функции и разделы без затруднений. Для этого важны доступность, обратная связь с пользователями и оптимизация интерфейса.

**6. Гибкость и адаптивность:** система должна поддерживать возможность модификации и расширения функционала в соответствии с изменяющимися требованиями и условиями использования. Это включает поддержку различных форматов контента, интеграцию с другими системами и возможность внедрения новых функций и опций.

**7. Совместимость и интеграция:** программное обеспечение должно обеспечивать совместимость с популярными операционными системами, веб-браузерами и устройствами, а также возможность интеграции с существующими образовательными ресурсами, системами управления курсами, авторизации и аналитики.

**8. Соответствие стандартам:** проект должен соответствовать принятым стандартам, регулятивным и техническим требованиям, обеспечивая качество продукта, совместимость с другими решениями и надежность сервиса.

### Надежность

Надежность программного обеспечения системы "EduPlatform" является одним из ключевых факторов, определяющих ее успех среди пользователей. В данном разделе определены факторы, которые способствуют достижению требуемого уровня надежности системы во время поставки:

**1. Отказоустойчивость:** система должна иметь механизмы, обеспечивающие быстрое восстановление после возникновения ошибок, минимизируя время простоя и потери данных. Это включает автоматическое резервное копирование, журналирование ошибок и контроль целостности данных.

**2. Тестирование:** процесс разработки должен содержать всестороннее тестирование программного обеспечения на всех этапах разработки, включая юнит-тестирование, интеграционное тестирование и приемочное тестирование с использованием сценариев, отражающих разнообразные случаи использования системы.

**3. Мониторинг и поддержка:** необходимо наладить процесс контроля работы системы, а также предусмотреть средства и инструменты для быстрого обнаружения и решения возникающих проблем или инцидентов.

**4. Качественное программное обеспечение:** разработка должна включать применение лучших практик программирования и интеграции, а также выполнение стандартов кодирования и документирования, что повысит надежность системы.

**5. Регулярное обновление и улучшение системы:** обеспечивать постоянное развитие и совершенствование продукта с учетом обратной связи, выявленных ошибок и изменений требований, стандартов или рекомендаций.

**6. Обучение и документация:** разрабатывать и предоставлять информацию и обучающие материалы для пользователей и разработчиков о правильном использовании системы и ее возможностях, а также о решении возникающих проблем.

**7. Проактивный подход к определению и устранению ошибок:** целенаправленное изучение и анализ ошибок, выявленных в процессе разработки и эксплуатации системы, позволит своевременно предотвращать возникновение новых ошибок и обеспечит высокий уровень надежности.

Соблюдая вышеуказанные факторы, система "EduPlatform" сможет обеспечить требуемую надежность программного обеспечения во время поставки, что важно для успешного и эффективного использования онлайн-платформы обучения пользователями.

### Эксплуатационная готовность

Для того чтобы гарантировать высокий уровень эксплуатационной готовности системы "EduPlatform", необходимо учесть и реализовать следующие факторы:

**1. Точки останова:** система должна иметь возможность автоматически сохранять состояние работы в определенные моменты времени или при определенных условиях, чтобы обеспечить возможность восстановления работы с сохраненной точки в случае сбоя или ошибки. Это минимизирует потерю данных и обеспечивает непрерывность процесса обучения и оценивания.

**2. Восстановление и перезапуск:** система должна обеспечивать быстрое и эффективное восстановление работы после сбоев или перезапусков. Восстановление может включать в себя процедуры для возвращения к последней сохраненной точке, исправления ошибок и возобновления работы с момента останова.

**3. Резервное копирование и восстановление данных:** регулярное резервное копирование и сохранение данных системы на отдельных устройствах или облачных хранилищах позволяет обеспечить безопасность информации и возможность восстановления в случае катастрофических сбоев или потерь данных.

**4. Мониторинг и оповещение:** внедрение системы мониторинга и оповещения позволяет контролировать работу системы в режиме реального времени, своевременно определять и реагировать на возможные проблемы, минимизируя время простоя и обеспечивая эксплуатационную готовность системы.

**5. Функции автоматического масштабирования и балансировки нагрузки:** реализация таких функций позволяет системе адаптироваться к изменениям в нагрузке и инфраструктуре, обеспечивая стабильную работу и высокую доступность системы.

**6. Регулярное обновление и исправление ошибок:** проведение регулярных обновлений программного обеспечения и исправление ошибок обеспечивает устранение уязвимостей и улучшение стабильности системы, обеспечивая повышение эксплуатационной готовности.

**7. Многоуровневая архитектура:** применение многоуровневой архитектуры позволяет разделить различные функции системы на разные слои и компоненты, что упрощает процесс обслуживания и улучшает надежность и эксплуатационную готовность системы.

С числом разработки этих факторов система "EduPlatform" сможет обеспечить высокий уровень эксплуатационной готовности и стабильности, что позволит пользователям полагаться на систему в процессе обучения и оценивания.

### Безопасность

Для обеспечения безопасности программного обеспечения системы "EduPlatform" следует учесть и реализовать следующие факторы, которые защитят систему от случайного или злонамеренного доступа, использования, модификации, разрушения или раскрытия данных:

**a) Использовать некоторые методы шифрования:** все критические данные, такие как пароли, личная информация и финансовые транзакции, должны быть зашифрованы в соответствии с современными стандартами, такими как AES, TLS или SSL, чтобы обеспечить их конфиденциальность и защиту при передаче и хранении.

**b) Содержать детальные журналы или исторические наборы данных:** система должна вести журналы всех значимых событий и операций, связанных с доступом, использованием и модификацией данных. Это позволит отслеживать подозрительную активность, проводить аудит, а также восстанавливать данные в случае их потери или повреждения.

**c) Назначать некоторые функции в различные модули:** программное обеспечение должно быть разделено на модули с определенными функциями и обязанностями, что облегчит контроль доступа, предотвратит несанкционированные изменения и упростит процесс обслуживания и обновления системы.

**d) Ограничить связи между некоторыми областями программы:** разработка должна предусмотреть ограничение связей между различными компонентами системы и сегментирование сетевых зон, чтобы контролировать доступ к конфиденциальным данным и предотвратить их раскрытие, а также уменьшить вероятность успешных атак на систему.

**e) Проверять целостность данных для критических переменных:** система должна реализовывать процедуры проверки и контроля целостности данных, особенно для критически важных переменных. Это может быть достигнуто через реализацию контрольных сумм, хэш-функций или других методов контроля целостности для гарантии надежности данных.

Реализация этих факторов гарантирует обеспечение высокого уровня безопасности для программного обеспечения системы "EduPlatform", что является необходимым условием для создания доверительной и надежной платформы для студентов и преподавателей в процессе онлайн-обучения и оценки работ.

### Ремонтопригодность

Для обеспечения легкости обслуживания программного обеспечения системы "EduPlatform" необходимо определить и реализовать следующие параметры:

**1. Модульность:** программное обеспечение должно быть организовано в отдельные модули с четким разделением функций и ответственности. Это позволит упростить процесс обновления, исправления ошибок и расширения функционала системы, также облегчит понимание кода разработчиками и поддержку программного обеспечения.

**2. Четкие и документированные интерфейсы:** все интерфейсы между модулями и компонентами системы должны быть хорошо определены, документированы и снабжены комментариями. Это поможет новым разработчикам быстрее разобраться в проекте и упростит интеграцию системы с другими сервисами или модулями.

**3. Сложность и структура кода:** код программного обеспечения должен быть чистым, структурированным и легко читаемым. Использование соглашений об именовании и хороших практик программирования облегчает проведение обслуживания и внесение изменений в код системы.

**4. Тестирование и контроль версий:** использование автоматизированных тестов и систем контроля версий программного обеспечения, таких как Git или SVN, обеспечивает возможность отмены изменений, проверки стабильности программного обеспечения и минимизации рисков, связанных с дальнейшим обслуживанием и разработкой системы.

**5. Документация и обучающие материалы:** для успешного обслуживания и улучшения программного обеспечения крайне важно наличие подробной документации по программным компонентам и их взаимодействию, а также руководств и обучающих материалов для разработчиков и пользователей.

**6. Процесс обратной связи:** наличие процесса сбора и учета обратной связи от пользователей и разработчиков позволяет улучшить качество программного обеспечения и своевременно выявлять и устранять проблемы в процессе обслуживания и эксплуатации системы.

Учитывая все вышеуказанные параметры, можно обеспечить удобную и эффективную ремонтопригодность программного обеспечения системы "EduPlatform". Это позволит поддерживать систему в актуальном состоянии, адаптируя ее к меняющимся условиям и потребностям пользователей, а также улучшать ее качество и функционал.

### Переносимость

Система "EduPlatform" разрабатывается с учетом максимальной переносимости на различные платформы и операционные системы.

**a) Процент компонентов с машинозависимым кодом:** в нашем программном обеспечении стараемся минимизировать использование компонентов с зависимостью от машины. Ориентировочно, доля таких компонентов составляет менее N/A%.

**b) Процент кода, который является машинозависимым:** большая часть кода написана с использованием стандартных библиотек и языков программирования, что позволяет обеспечить поддержку не менее N/A% программного кода на различных платформах.

**c) Использование проверенного переносимого языка:** для повышения переносимости мы используем проверенный язык программирования, такой как Python, который является кросс-платформенным и поддерживается на большинстве современных операционных систем.

**d) Использование конкретного компилятора или поднабора языков:** мы используем стандартные компиляторы и наборы языков для обеспечения максимальной совместимости и переносимости.

**e) Использование конкретной операционной системы:** наша система не привязана к определенной операционной системе и может быть запущена на любой современной ОС, поддерживающей используемый язык программирования и необходимые библиотеки.

## Структурирование детальных требований

Для системы "EduPlatform" мы структурируем детальные требования следующим образом:

**1. Функциональные требования**

1.1 Авторизация и аутентификация пользователей

1.2 Управление профилями студентов и преподавателей

1.3 Создание и управление курсами

1.4 Загрузка и хранение материалов курса

1.5 Организация процесса обучения (лекции, семинары, практики)

1.6 Организация оценивания работ студентов

1.7 Предоставление обратной связи студентам и преподавателям

**2. Нефункциональные требования**

2.1 Производительность

2.2 Надежность и безопасность данных

2.3 Масштабируемость

2.4 Удобство использования и единообразие интерфейса

2.5 Возможность интеграции с другими сервисами

2.6 Документация и поддержка

**3. Требования к архитектуре**

3.1 Модульность и слабая связанность компонентов

3.2 Разделение ответственности между архитектурными слоями

3.3 Определение возможных требований к интерфейсам между модулями

**4. Требования к технологиям и инструментам**

4.1 Языки программирования

4.2 Библиотеки и фреймворки

4.3 Базы данных

4.4 Средства разработки и инструменты автоматизации

4.5 Поддержка стандартов взаимодействия и операционных систем

**5. Ограничения и предположения**

5.1 Ограничения по времени разработки и ресурсам

5.2 Уровень компетенции пользователей

Эта структура отражает основные аспекты системы и позволяет упорядочить требования для оптимального понимания и последующей разработки программного обеспечения. Она также может быть дополнена и скорректирована в ходе процесса разработки, чтобы обеспечить лучшее понимание системы и учет всех возможных кейсов использования.

### Режим системы

Система "EduPlatform" предусматривает несколько режимов работы, чтобы обеспечить различные функциональные возможности в зависимости от ситуации и потребностей пользователей. Основные режимы системы включают:

**1. Обучение:** в этом режиме, студенты и преподаватели ознакамливаются с функционалом системы и могут выполнить операции без возможности нанести вред основным данным и процессам. В режиме обучения пользователи имеют доступ к демонстрационным курсам, материалам и заданиям, чтобы практиковаться и понять, как использовать систему.

**2. Нормальная работа:** в режиме нормальной работы система предоставляет полный набор функций и возможностей, необходимых для обучения, управления курсами, взаимодействия между студентами и преподавателями, оценивания работ и предоставления обратной связи. В этом режиме пользователи выполняют реальные задачи и проходят курсы в рамках учебного процесса.

**3. Аварийная ситуация:** в случае возникновения аварийных ситуаций, таких как сбои в системе, обнаружение ошибок или угроз безопасности данных, система переходит в режим аварийной ситуации. В этом режиме возможности пользователей могут быть ограничены, доступ к определенным функциям может быть временно приостановлен, а некоторые операции могут быть отменены в целях обеспечения стабильности и защиты данных. В режиме аварийной ситуации администрация системы и специалисты по обслуживанию занимаются устранением проблем, а пользователи могут получать уведомления и информацию о возникших сбоях и ожидаемых сроках восстановления работы системы.

Для каждого режима работы системы определены соответствующие интерфейсы, операции и ограничения. Эти режимы могут быть активированы администрацией системы или автоматически, в зависимости от условий и ситуаций. Разделение системы на режимы позволяет предоставить пользователям оптимальные условия работы, обеспечить их комфорт и безопасность, а также успешно справляться с различными вызовами и ситуациями, возникающими в процессе эксплуатации программного обеспечения.

### Классы пользователей

Система "EduPlatform" предусматривает разные наборы функций для различных классов пользователей. Это делается для обеспечения соответствующих возможностей и ограничений, адекватных ролям пользователей в процессе обучения и управления. Основные классы пользователей в системе включают:

**1. Студенты:** пользователи, зарегистрированные как студенты, имеют доступ к курсам и обучающим материалам, могут выполнять задания, оценивать работы других студентов (при условии использования пиринговой оценки), общаться с преподавателями и другими студентами, получать обратную связь и оценки за свою работу. Функционал этого класса пользователей ориентирован на изучение и обеспечение взаимодействия в учебном процессе.

**2. Преподаватели:** пользователи, зарегистрированные как преподаватели, могут создавать и управлять курсами, загружать обучающие материалы, задавать и оценивать задания, осуществлять мониторинг успеваемости студентов, общаться со студентами и предоставлять им обратную связь. Преподаватели также могут участвовать в решении спорных ситуаций при оценивании работ и контролировать процесс обучения.

**3. Администрация:** администраторы системы не только имеют доступ к функциям студентов и преподавателей, но и обладают рядом дополнительных возможностей, таких как управление учетными записями пользователей, установление прав и ограничений для различных классов пользователей, активация и настройка различных режимов работы системы, мониторинг безопасности, надежности и производительности платформы.

При организации доступа к функциям системы в рамках разных классов пользователей используется управление доступом на основе ролей (Role-Based Access Control, RBAC). Это позволяет определить соответствующие возможности и ограничения для каждого класса пользователей и обеспечить гибкое управление правами доступа в динамически изменяющейся среде обучения и сотрудничества.

### Объекты

В системе "EduPlatform" определены следующие основные объекты, являющиеся реальными сущностями с дубликатами внутри системы:

**1. Курсы** - образовательные программы, состоящие из различных материалов и заданий.

Атрибуты: название, описание, преподаватель, сроки, темы, статус и т.д.

Функции: создание, редактирование, удаление, назначение студентов, управление материалами.

**2. Студенты** - пользователи системы, зарегистрированные для изучения курсов и выполнения заданий.

Атрибуты: имя, электронная почта, группа, курсы, полученные оценки, обратная связь.

Функции: регистрация, прохождение курсов, выполнение заданий, оценивание работ, коммуникация с преподавателями и другими студентами.

**3. Преподаватели (Instructors)** - пользователи, создающие и управляющие курсами, оценивающие работы студентов и предоставляющие им обратную связь.

Атрибуты: имя, электронная почта, курсы, рейтинг, опыт и т.д.

Функции: создание и управление курсами, оценка работ, предоставление обратной связи, коммуникация со студентами.

**4. Материалы курса** - обучающие материалы, используемые в рамках курса (лекции, видео, слайды, тексты и т.д.).

Атрибуты: название, тип, размер, автор, дата создания, курс, порядок и т.д.

Функции: загрузка, редактирование, удаление, сортировка и группировка.

**5. Задания (Assignments)** - исследовательские, контрольные или практические работы, выполняемые студентами в рамках курса.

Атрибуты: название, описание, курс, сроки, оценка, статус, выдавший преподаватель и т.д.

Функции: создание, редактирование, удаление, выдача студентам, оценка, предоставление обратной связи.

Сгруппированные объекты, относящиеся к одному классу, могут иметь общие атрибуты и услуги, что обеспечивает стандартизацию и универсализацию функций в рамках системы. Это подход позволяет упростить процесс разработки и обслуживания системы, а также обеспечить гибкость и адаптивность программного обеспечения при реализации изменений и улучшений.

### Особенности

Особенности системы "EduPlatform" представляют собой набор внешних сервисов, доступных пользователям и обеспечивающих желательный результат при выполнении определенных действий. Ниже приведены примеры основных особенностей и их последовательность взаимодействия "воздействие-реакция":

**1. Взаимодействие со студентами и преподавателями**

Воздействие: студент или преподаватель отправляет сообщение / запрос

Реакция: система обеспечивает доставку сообщения / запроса и отображает его в интерфейсе получателя.

**2. Регистрация на курс**

Воздействие: студент выбирает курс и подает заявку на регистрацию

Реакция: система регистрирует студента на курсе и предоставляет доступ к материалам.

**3. Создание и редактирование курса**

Воздействие: преподаватель создает новый курс или редактирует существующий

Реакция: система сохраняет информацию о курсе и обновляет его контент для доступа студентов и преподавателей.

**4. Выполнение задания**

Воздействие: студент загружает выполненное задание в систему

Реакция: система сохраняет информацию о выполненном задании и отправляет уведомление преподавателю для оценки.

**5. Оценивание работ**

Воздействие: преподаватель или студент (в случае пиринговой оценки) оценивает работу

Реакция: система сохраняет оценку и предоставляет обратную связь студенту

Для организации работы с особенностями системы используется схема, представленная в разделе 5 приложения A. Она позволяет упорядочить доступ к особенностям, обеспечить структурированное представление возможностей системы и предоставить пользователям удобный способ работы с ними.

### Воздействие

В контексте системы "EduPlatform" следующие воздействия могут оказывать влияние на работу системы и ее функциональность. Организация системы с учетом таких воздействий позволяет лучше подготовиться к возможным ситуациям и обеспечить более высокий уровень надежности и стабильности.

**1. Нагрузка на систему**

Описание: Система может столкнуться с повышенной нагрузкой из-за увеличения числа пользователей или загруженных курсов и материалов.

Реакция: Система оптимизирует ресурсы, масштабирует серверную инфраструктуру и распределяет нагрузку, обеспечивая надежную работу платформы.

**2. Проблемы с безопасностью данных**

Описание: Возможные угрозы безопасности, такие как несанкционированный доступ, вредоносное ПО или утечка данных.

Реакция: Система активирует защитные механизмы, такие как шифрование, аутентификация и мониторинг активности пользователей, чтобы минимизировать риск и устранить угрозу.

**3. Технические сбои и ошибки**

Описание: Возникновение технических ошибок в работе системы, скриптов, серверов или сетевого подключения.

Реакция: Система определяет проблемные компоненты, предпринимает автоматические действия для устранения сбоев, включая перезапуск служб, а также уведомляет администраторов для последующей проверки и решения возникших проблем.

**4. Непредусмотренное пользовательское взаимодействие**

Описание: Сценарии, при которых пользователи взаимодействуют с системой непредусмотренными разработчиками способами, которые могут вызвать ошибки или создать беспорядок в данных.

Реакция: Система обеспечивает корректную обработку исключительных ситуаций, предоставляет информацию о возникших проблемах и предлагает возможные пути решения.

Описанные воздействия помогают предусмотреть потенциальные проблемы и определить стратегии реагирования на эти ситуации. Такая организация упрощает анализ возможных рисков и позволяет разработать более устойчивую и надежную систему, которая может реагировать на изменения условий работы и предоставлять оптимальное качество услуг пользователям.

### Реакция

Система "EduPlatform" может быть лучше организована, если ее функции описаны в терминах генерации реакции. В данном контексте реакция — это действия, которые система выполняет в ответ на действия пользователей. Ниже приведены основные реакции системы.

#### Создание платформы проверки работ

В ответ на загрузку работы студентом, система активирует механизм выбора проверяющих и распределения работы для оценивания.

#### Расчет итоговых оценок и обратная связь

Система рассчитывает итоговые оценки на основе оценок, выставленных студентами-рецензентами. Если разброс оценок небольшой, система определяет средний балл и выставляет его студенту. Если разброс оценок велик, система отправляет работу для проверки преподавателю. В обоих случаях студенту предоставляется обратная связь с указанием итоговой оценки.

#### Перенаправление работы преподавателю

В случае существенных различий в оценках студентов-рецензентов, система перенаправляет работу преподавателю для проверки и окончательного определения оценки. Это повышает качество оценивания и делает процесс более справедливым.

#### Обеспечение взаимодействия между студентами и преподавателями

Система предоставляет возможность общения между участниками образовательного процесса, что позволяет студентам задавать вопросы, делиться знаниями и опытом, а преподавателям – участвовать в обсуждении, определении сложных вопросов, уточнении заданий и оказанию консультативной поддержки студентам.

#### Мониторинг успеваемости студентов

Система позволяет преподавателям в режиме реального времени отслеживать успехи студентов, выявлять проблемы и своевременно предпринимать корректирующие действия для улучшения результатов обучения.

Таким образом, "EduPlatform" организована таким образом, что ее функции и реакция на действия пользователей обеспечивают эффективное взаимодействие студентов и преподавателей, достижение целей обучения и повышение качества оценки студенческих работ.

### Функциональные иерархии

Для системы "EduPlatform" функциональные возможности организуются в иерархию функций с использованием принципа общих входов и общих выходов. Ниже приведена иерархическая структура функций системы.

**1. Управление учебными процессами**

1.1 Создание и редактирование учебных курсов

1.2 Назначение преподавателей

1.3 Регистрация и зачисление студентов на курсы

1.4 Управление календарем учебных событий

**2. Работа с обучающими материалами**

2.1 Загрузка и хранение учебных материалов

2.2 Предоставление доступа к материалам для студентов и преподавателей

**3. Оценивание студенческих работ**

3.1 Загрузка работ студентами

3.2 Выбор проверяющих и распределение работ для оценки

3.3 Расчет итоговых оценок и обратная связь

3.3.1 Расчет среднего балла

3.3.2 Перенаправление работы преподавателю при большом разбросе оценок

3.4 Оценивание работ преподавателем

**4. Взаимодействие студентов и преподавателей**

4.1 Обсуждение вопросов и материалов курса

4.2 Получение помощи и консультаций от преподавателей

4.3 Обмен опытом и знаниями между студентами

**5. Мониторинг успеваемости студентов**

5.1 Отслеживание прогресса студентов

5.2 Выявление слабых и сильных сторон студентов

5.3 Принятие корректирующих мер для улучшения обучения

**6. Управление доступом и безопасностью**

6.1 Регистрация и вход пользователей

6.2 Управление правами доступа студентов и преподавателей

6.3 Защита данных и информации пользователей

Используя эту иерархию функциональных возможностей, разработчики могут систематизировать и организовать процесс создания системы "EduPlatform", а пользователи могут лучше понимать функционал и возможности системы.

## Дополнительные комментарии

В процессе определения требований для системы "EduPlatform" были использованы различные способы структурирования, чтобы адаптироваться к особенностям системы и упростить последующую разработку. Были использованы такие подходы, как функциональные иерархии, реакция на воздействия, взаимодействие объектов и организация по ролям пользователей. В этом разделе представлены дополнительные комментарии и предложения по документации и структурированию требований.

**1. Автоматизированные инструменты поддержки:**

В процессе документирования требований может быть полезно использовать автоматизированные инструменты, такие как системы управления требованиями (Requirements Management Systems) или специализированные редакторы, которые будут облегчать структурирование, изменение и связывание требований.

**2. Разделение требований по уровням:**

Возможность разделения функциональных требований на различные уровни, такие как системные, подсистемы и компоненты, может упростить понимание структуры и организации системы.

**3. Использование диаграмм и моделей:**

Применение визуальных инструментов, таких как диаграммы потоков данных, диаграммы состояний, диаграммы классов, может сильно облегчить понимание требований и логики взаимодействия между ними.

**4. Учет языка и нотации:**

Важно, чтобы формулировка требований была ясной, понятной и однозначной. Использование стандартизированных терминов, псевдокода, языков описания систем или специализированных нотаций может помочь сократить ошибки и неоднозначности в процессе разработки.

**5. Четыре подраздела функциональных требований:**

Для конкретизации функциональных требований можно использовать подразделы "Введение", "Входы", "Обработка (Processing)" и "Выходы" для каждой функции. Такой подход может повысить прозрачность логики работы системы и упростить ее разработку и модификацию.

**6. Внимание к не техническим требованиям:**

Необходимо учитывать также нетехнические аспекты, такие как удобство использования, доступность, эргономика, совместимость с существующими системами и стандартами, а также ожидания пользователей от системы.

В целом, качество документации требований является критически важным фактором в разработке программных систем и может значительно повлиять на ее успешность и эффективность. Хорошо структурированные и подробные требования позволят каждому участнику проекта работать продуктивно и избегать недопонимания и ошибок.