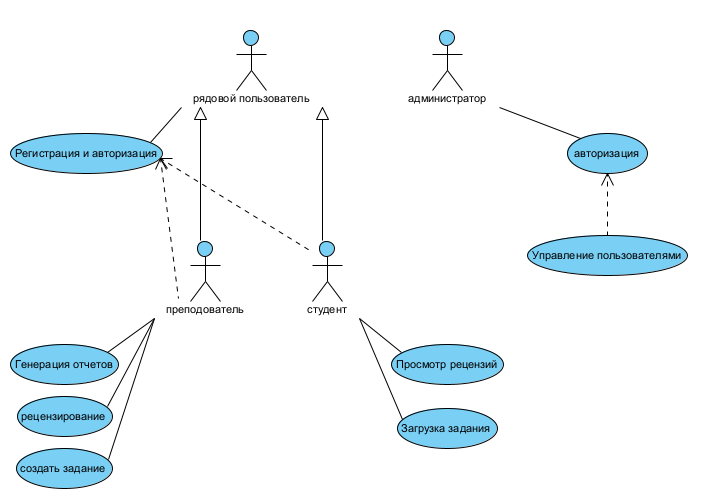
# Диаграмма вариантов использования (Use Case Diagram)



**Основные акторы системы**

1. **Рядовой пользователь** (AC03)
   * Обобщающая роль, от которой наследуются более специализированные роли
   * Выполняет базовую регистрацию и авторизацию в системе
2. **Студент** (AC01) - специализируется от "Рядового пользователя"
   * Загружает задания для рецензирования
   * Просматривает полученные рецензии
3. **Преподаватель** (AC02) - специализируется от "Рядового пользователя"
   * Создает учебные задания
   * Выполняет рецензирование работ студентов
   * Генерирует отчеты по результатам работы
4. **Администратор** (AC04)
   * Управляет пользователями системы
   * Выполняет авторизационные функции

**Основные функции системы (Use Cases)**

**Для студентов:**

* **Загрузка задания** (UC02) - отправка работ на проверку
* **Просмотр рецензий** (UC06) - ознакомление с результатами проверки

**Для преподавателей:**

* **Создать задание** (UC03) - создание учебных заданий
* **Рецензирование** (UC04) - проверка и оценка студенческих работ
* **Генерация отчетов** (UC07) - формирование аналитических отчетов

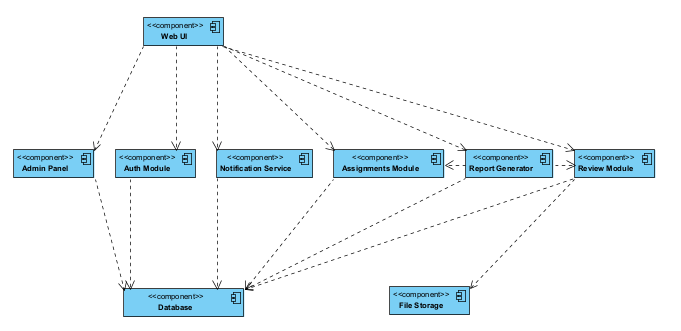
**Для администратора:**

* **Авторизация** (UC08) - специализированная функция для администратора
* **Управление пользователями** (UC05) - администрирование учетных записей

**Общие функции:**

* **Регистрация и авторизация** (UC01) - вход в систему для всех пользователей

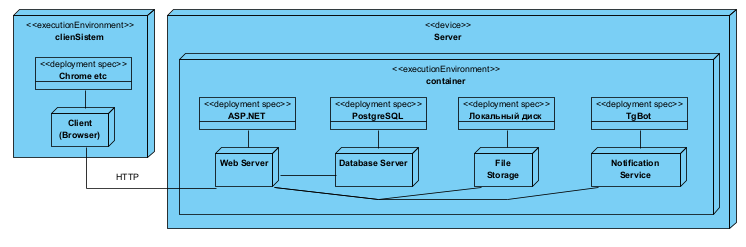
# Диаграмма компонентов



**Компоненты системы («строительные блоки»)**

1. **Web UI (Веб-интерфейс)**
   * **Роль:** Главный точк входа для пользователей. Предоставляет пользовательский интерфейс.
   * **Зависимости:** Зависит от всех основных сервисных модулей, чтобы отображать данные и обрабатывать действия пользователя.
2. **Auth Module (Модуль аутентификации)**
   * **Роль:** Отвечает за аутентификацию и авторизацию пользователей.
   * **Зависимости:** Для хранения и проверки учетных данных зависит от **Database**.
3. **Assignments Module (Модуль назначений)**
   * **Роль:** Управляет назначениями (например, задачами для рецензирования).
   * **Зависимости:** Зависит от **Database** для хранения данных о назначениях.
4. **Review Module (Модуль рецензирования)**
   * **Роль:** Ядро системы. Отвечает непосредственно за процесс рецензирования, создание и управление рецензиями.
   * **Зависимости:** Для хранения рецензий зависит от **Database**, а для хранения файлов зависит от **File Storage**.
5. **Notification Service (Сервис уведомлений)**
   * **Роль:** Отправляет уведомления пользователям.
   * **Зависимости:** Может хранить шаблоны уведомлений или логи в **Database**.
6. **Report Generator (Генератор отчетов)**
   * **Роль:** Формирует отчеты на основе данных о рецензиях и назначениях.
   * **Зависимости:** Для получения данных зависит от **Review Module**, **Assignments Module** и **Database**.
7. **Admin Panel (Панель администратора)**
   * **Роль:** Специализированный интерфейс для администраторов системы для управления пользователями, настройками и т.д.
   * **Зависимости:** Для выполнения административных функций зависит от **Database**.
8. **Database(База данных)**
   * **Роль:** Основное хранилище данных системы. Хранит информацию о пользователях, назначениях, рецензиях и т.д.
   * **Зависимости:** Это конечный пункт для хранения данных, от него зависят почти все сервисные модули.
9. **File Storage (Файловое хранилище)**
   * **Роль:** Внешнее хранилище для файлов (например, документы для рецензирования, итоговые отчеты в формате PDF).
   * **Зависимости:** Используется **Review Module** для загрузки и выгрузки файлов.

# Диаграмма развертывания



**Ключевые Узлы (Nodes) и их иерархия**

Диаграмма имеет четкую иерархическую структуру:

**1. Уровень Устройств (Device)**

* Server**(Сервер)**
  + **Стереотип:** «device»
  + **Роль:** Основной физический или виртуальный сервер, на котором размещена вся бэкенд-часть системы.
* clienSistem**(Клиентская система)**
  + **Стереотип:** «executionEnvironment»
  + **Роль:** Представляет среду выполнения на стороне пользователя (персональный компьютер, ноутбук).

**2. Уровень Среды выполнения (Execution Environment) - внутри**Server

* container**(Контейнер)**
  + **Стереотип:** «executionEnvironment»
  + **Роль:** Абстракция для среды выполнения, в которой размещены основные программные компоненты. Скорее всего, символизирует **Docker-контейнер** или аналогичную технологию, инкапсулирующую среду выполнения приложения.

**3. Уровень Программных компонентов (внутри**container**)**

* Web Server**(Веб-сервер)**
  + **Роль:** Узел, который обрабатывает входящие HTTP-запросы от клиентов и служит точкой входа для веб-приложения.
* Database Server**(Сервер базы данных)**
  + **Роль:** Узел, отвечающий за хранение и управление данными приложения.
* File Storage**(Файловое хранилище)**
  + **Роль:** Узел, предоставляющий услуги хранения файлов (документы, отчеты, загружаемые файлы).
* Notification Service**(Сервис уведомлений)**
  + **Роль:** Независимый сервис, отвечающий за отправку уведомлений (например, через Telegram).

**4. Уровень Клиента (внутри**clienSistem**)**

* Client (Browser)**(Клиент/Браузер)**
  + **Роль:** Веб-браузер пользователя, где выполняется клиентская часть приложения (HTML, CSS, JavaScript).

**Артефакты (Deployment Specifications)**

Это конкретные программные продукты и технологии, развернутые на узлах:

* **На**Web Server**:**
  + ASP.NET**:** Основная технология, на которой построено веб-приложение (бэкенд).
* **На**Database Server**:**
  + PostgreSQL**:** Используемая система управления базами данных (СУБД).
* **На**File Storage**:**
  + Локальный диск**:** Указывает, что файловое хранилище реализовано на локальных дисках сервера.
* **На**Notification Service**:**
  + TgBot**:** Конкретная реализация сервиса уведомлений через Telegram Bot API.
* **На**Client (Browser)**:**
  + Chrome etc**:** Указывает, что клиентское приложение - это веб-сайт, работающий в браузерах, подобных Chrome.

**Ключевые Связи (Associations) и протоколы**

Связи показывают, как узлы взаимодействуют в сети:

1. Web Server**<-->**Client (Browser)
   * **Протокол:** HTTP
   * **Описание:** Основное взаимодействие между фронтендом и бэкендом через веб-протокол.
2. Web Server**-->**Database Server
   * **Описание:** Веб-приложение обращается к базе данных для сохранения и извлечения информации.
3. Web Server**-->**File Storage
   * **Описание:** Веб-приложение загружает файлы от пользователей и сохраняет их в файловое хранилище, а также выгружает их по запросу.
4. Web Server**-->**Notification Service
   * **Описание:** Веб-приложение отправляет запросы сервису уведомлений, когда необходимо уведомить пользователя (например, о новом задании на рецензию).