

1. Тема проекта

Разработка образовательной платформы с централизованной структурой знаний, системой авторизации, тестированием, интерактивными элементами.

2. Исполнители

Проект разрабатывается командой из трёх человек:

- **Кадыркулов Эртай** – backend-разработка, проектирование базы данных, API.
- **Тухтаназаров Нурсултан** – backend-разработка, клиентская часть.
- **Бекболотова Айгерим** – frontend-разработка, UI/UX-дизайн

3. Краткое описание проекта

Образовательная платформа представляет собой веб-систему с возможностью обучения, тестирования. Основные функциональные возможности включают:

1. **Структура знаний**
 - a. Иерархическая организация контента: Факультеты > Направления > Уроки > Темы.
 - b. Доступ к материалам без авторизации (только для чтения).
2. **Управление учебными материалами**
 - a. Добавление контента студентами (с модерацией).
 - b. Вложения (изображения, PDF, видео и др.).
 - c. Система голосования за качество контента.
 - d. (Встроенный текстовый редактор с поддержкой Markdown).
3. **Тестирование**
 - a. Различные форматы тестов: множественный выбор, открытые вопросы и др.
 - b. Автоматическая проверка тестов.
4. **Система хранения и обмена знаниями**
 - a. Загрузка файлов (PDF, DOCX, PPT и др.).
 - b. Комментирование материалов.

Проект направлен на создание удобной образовательной платформы с интерактивными элементами, улучшая процесс обучения и вовлеченность студентов.

4)Цель написания этой программы.

Кто будет использовать?

- **Студент** – изучает материалы, дополняет темы, предлагает новые, участвует в обсуждениях.
- **Модератор** – проверяет и утверждает контент, следит за качеством материалов.
- **Администратор** – управляет пользователями, настраивает систему, контролирует работу проекта.

Почему это нужно студентам? Какие проблемы оно решает?

1. Разрозненные учебные материалы

Проблема:

Студенты хранят учебные материалы в разных местах: Telegram-группах, личных файлах, облачных дисках, а иногда просто в тетрадях. Это затрудняет поиск нужной информации.

Решение:

МКHub объединяет все учебные материалы в одном месте, где студенты могут легко находить, изучать и дополнять темы.

2. Отсутствие единой базы знаний**Проблема:**

Когда студент хочет изучить тему, ему приходится искать информацию в учебниках, интернете или спрашивать у других студентов, а доступ к записям прошлых лет часто невозможен.

Решение:

МКHub предоставляет единую базу знаний, где по каждой теме хранятся проверенные материалы, добавленные студентами и подтверждённые модераторами.

4. Отсутствие практики и проверки знаний**Проблема:**

Многие студенты читают теорию, но не могут проверить, насколько хорошо они её поняли.

Решение:

МКHub позволяет студентам создавать тесты и задачи по темам, а другие пользователи могут их решать для проверки знаний.

5. Сложность самостоятельного изучения других направлений**Проблема:**

Студентам сложно изучать материалы других факультетов, так как у них нет доступа к учебным программам.

Решение:

МКHub даёт возможность изучать материалы не только своего факультета, но и других направлений, что расширяет кругозор и помогает в перекрёстном обучении.

Вывод

МКHub – это удобная система, которая решает ключевые проблемы студентов, связанные с хранением, поиском и обменом учебными материалами. Она упрощает процесс обучения, делает его доступнее и эффективнее.

5. Что нового / Чем отличается от других?**В чём уникальность МКHub?**

МКHub – это не просто платформа для хранения учебных материалов, а **интерактивная система знаний**, адаптированная под нужды университета. Она объединяет студентов, модераторов и учебные ресурсы в **единое цифровое пространство**, предлагая больше возможностей, чем стандартные платформы для онлайн-обучения.

1. Централизованная база знаний внутри университета

Чем отличается?

Обычно учебные материалы разбросаны по разным источникам: преподаватели загружают файлы на Google Drive, студенты обмениваются конспектами в мессенджерах, а тесты проводятся в разных системах.

МКHub собирает всё в одном месте, создавая единый цифровой ресурс для университета.

2. Открытая система, дополняемая студентами

Чем отличается?

Обычные онлайн-курсы или образовательные платформы закрыты – только преподаватели могут загружать контент.

В МКHub **студенты сами могут дополнять материалы**, предлагать новые темы и улучшать существующие знания, проходя модерацию.

3. Интерактивные тесты и задачи от студентов

Чем отличается?

Обычно тесты создаются преподавателями и не обновляются.

В МКHub **любой студент может добавить вопросы и задачи**, чтобы другие могли их решать и проверять знания.

4. Возможность изучать материалы других направлений

Чем отличается?

Обычно студенты получают доступ только к материалам своего факультета.

В МКHub **можно изучать темы и курсы с других факультетов**, расширяя свои знания в разных областях.

5. Расширяемая микросервисная архитектура

Чем отличается?

Большинство университетских систем жёстко ограничены функционалом и сложно модернизируются.

В МКHub **легко добавить новые сервисы** – например, симулятор лаборатории для химии или кодингую платформу для программирования.

Вывод

МКHub – это не просто база знаний, а **интерактивное образовательное пространство**, где студенты могут учиться, обсуждать темы, дополнять материалы и проверять свои знания, объединяя всё в **одну удобную платформу**.

6. Функционал (Functional Diagram)

Централизованное хранилище знаний:

- Доступ ко всем учебным материалам в одном месте (контенты, темы, задания).
- Навигация по факультетам и направлениям.
- Просмотр и поиск информации по темам, урокам и материалам.
- История изменений и предложений по материалам.

Управление контентом:

- Студенты могут добавлять свои материалы, предложения и исправления к существующим материалам.
- Все предложения студентов проходят проверку модераторами (проверка корректности и полноты контента).
- Возможность интеграции различных форматов контента: текст, изображения, ссылки и т.д.

Создание и управление курсами:

- Модераторы сами могут создавать и управлять контентом и уроками.
- Контенты могут быть разбиты на несколько тем, которые охватывают различные аспекты обучения.
- Модераторы могут обновлять материалы, добавлять новые уроки и тематики.

Взаимодействие между студентами:

- Обсуждения для студентов по каждой теме или контенту.
- Взаимопомощь между студентами (возможность добавлять комментарии и решения к материалам).

Тестирование и задания:

- Студенты могут создавать свои задания и тесты для других студентов.
- Задания могут быть добавлены к материалам урока для закрепления знаний.
- Модераторы могут проверять и модифицировать тесты и задания, предложенные студентами.

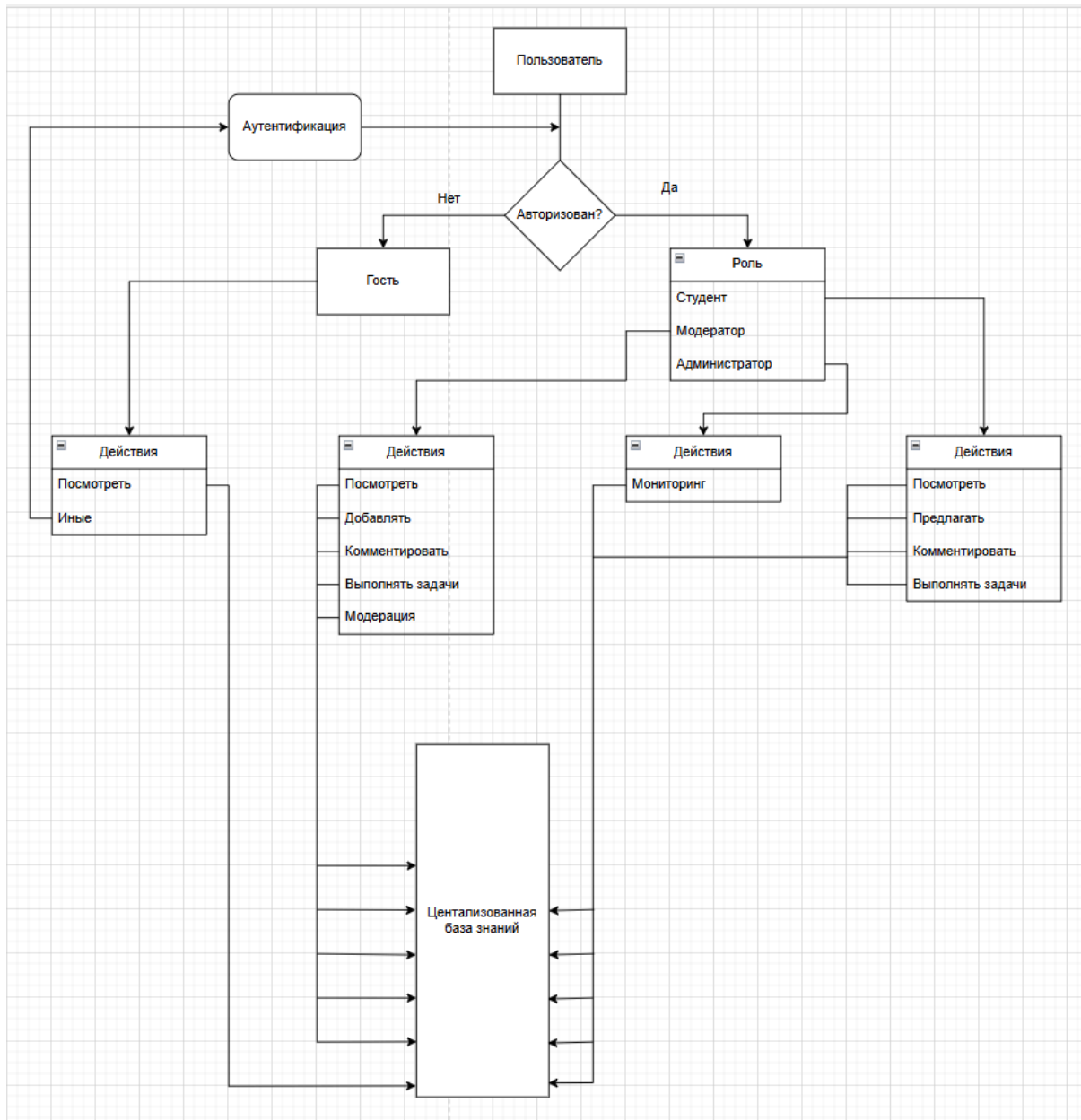
Поиск (Advanced)

Отзывы и рейтинг материалов:

- Студенты могут оценивать материалы, курсы и уроки.
- Возможность оставлять отзывы на уроки, тесты и другие материалы.

Управление пользователями:

- Регистрация студентов и модераторов.
- Роли и права пользователей (например, модераторы могут редактировать контенты, студенты могут только просматривать или добавлять предложения).
- Аутентификация и авторизация для обеспечения безопасности.



7. Роли и права пользователей (Use Case Diagram)

Студент
Просмотр материалов (курсы, темы, тесты, конспекты).
Дополнение материалов (предложение новых тем, редактирование существующих).
Взаимодействие с другими студентами (комментарии).
Выполнение тестов и заданий для закрепления знаний.
Загрузка и скачивание конспектов.
Поиск по библиотеке знаний.
Оценка материалов (лайки/дизлайки, отзывы, рейтинги).

Модератор
Проверка и модерация контента (редактирование, объединение, удаление).
Оценка студентов (механизм подтверждения участия или активности).
Контроль за обсуждениями (модерация комментариев, защита от спама).
Управление предложенными изменениями (редактирование и публикация).
Управление глобальными настройками платформы (добавление новых факультетов, направлений)

Администратор
Управление ролями и правами пользователей (назначение ролей, блокировка пользователей).
Контроль за системой (мониторинг, логирование).
Управление API Gateway (маршрутизация запросов).

Гость – может просматривать ограниченное количество материалов без регистрации.

8. Инструменты, которые можно использовать (Used Technologies Stack)

1. Язык программирования

- **Java** – основной язык для серверной части.

2. Фреймворки

- **Spring Framework** (Spring Boot)
 - **Spring Web MVC** – для обработки HTTP-запросов.
 - **Spring Data JPA** – для работы с базой данных через ORM.
 - **Spring Security** – для аутентификации и авторизации.
 - **Spring Validation** – для проверки входных данных.

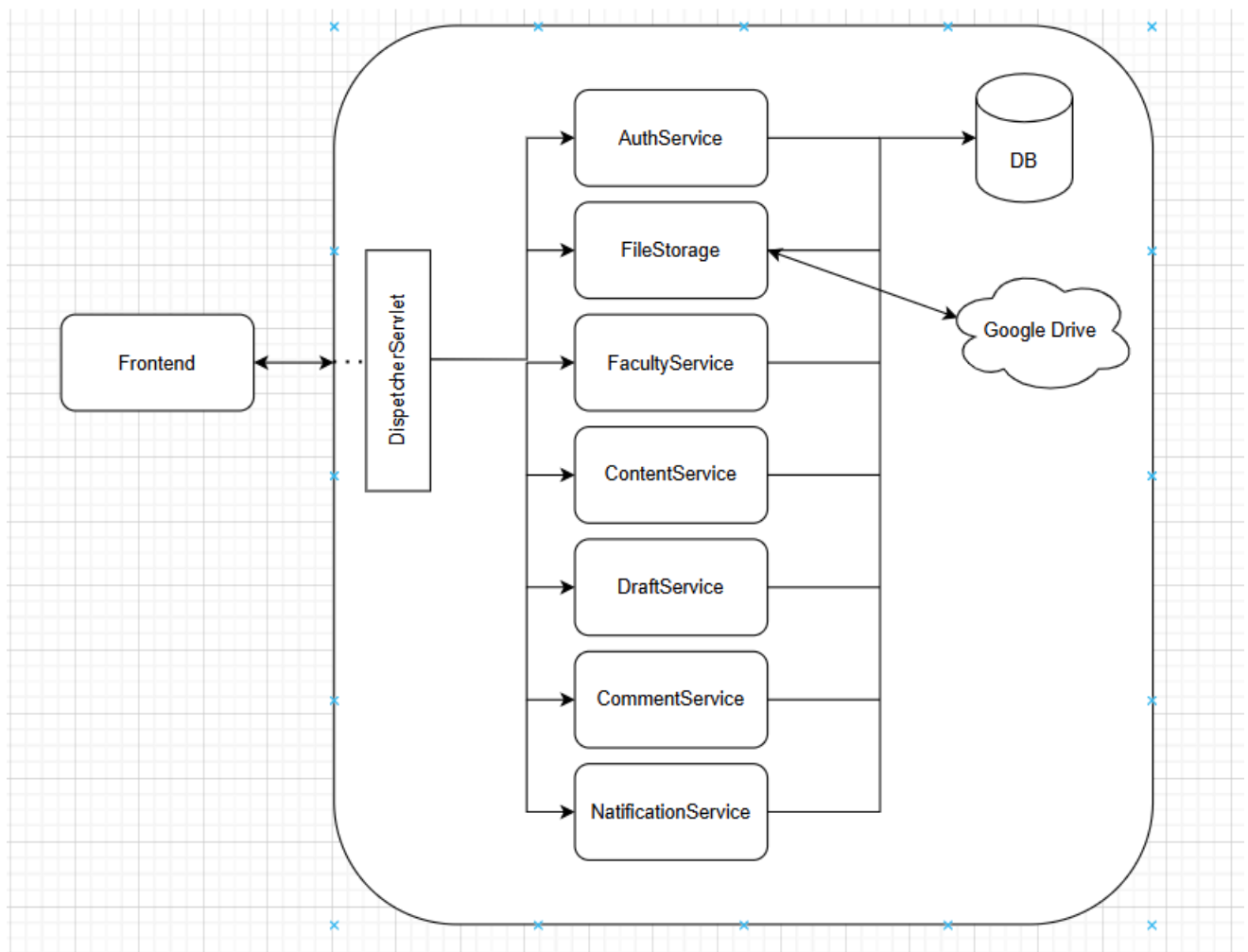
3. База данных

- **PostgreSQL** – основная реляционная база данных.

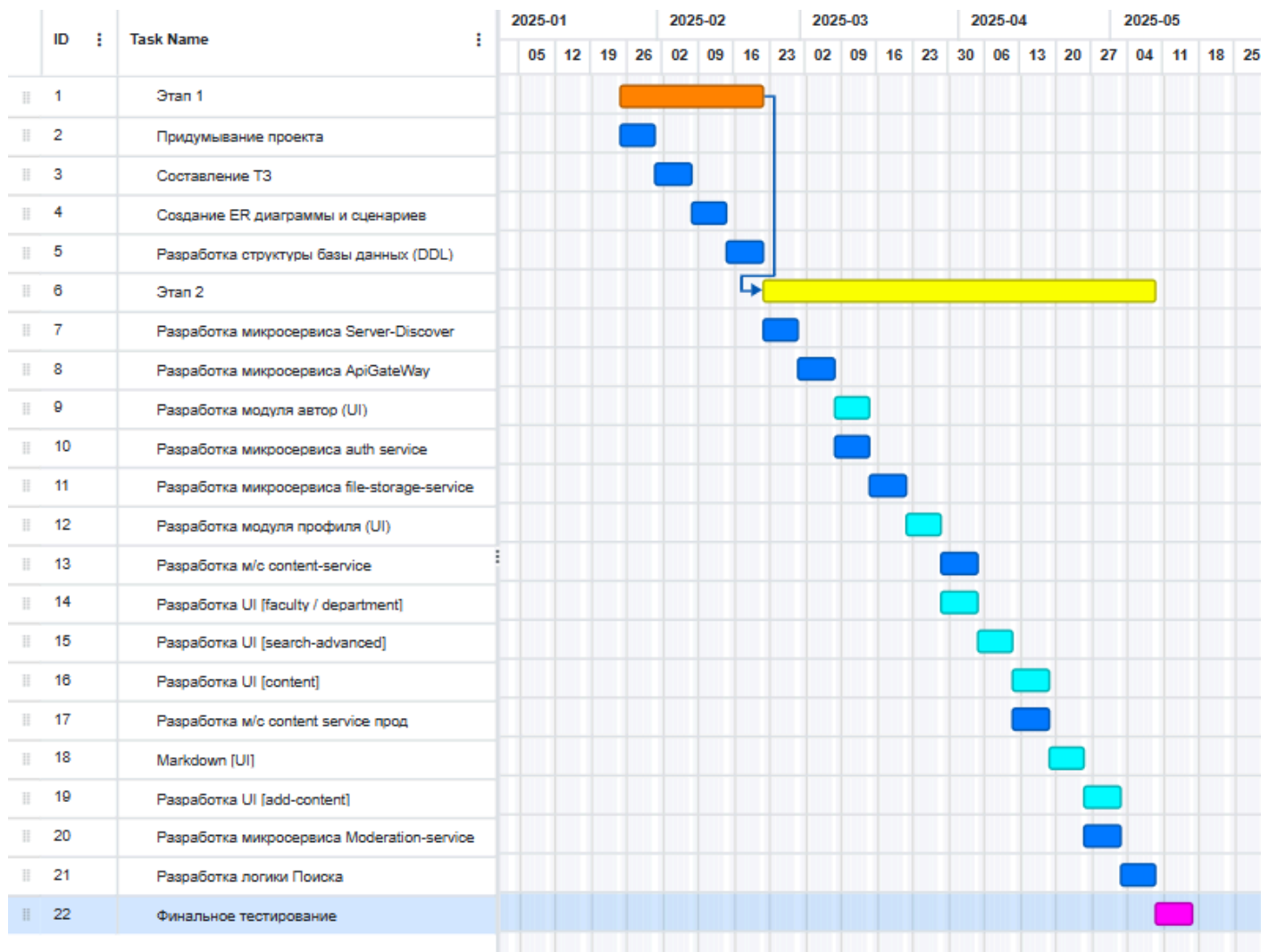
6. Интерфейс пользователя (Frontend)

- **Vue.js** – основной фреймворк для фронтенда.
- **Tailwind CSS** – стилизация.
- **Axios** – для работы с API.

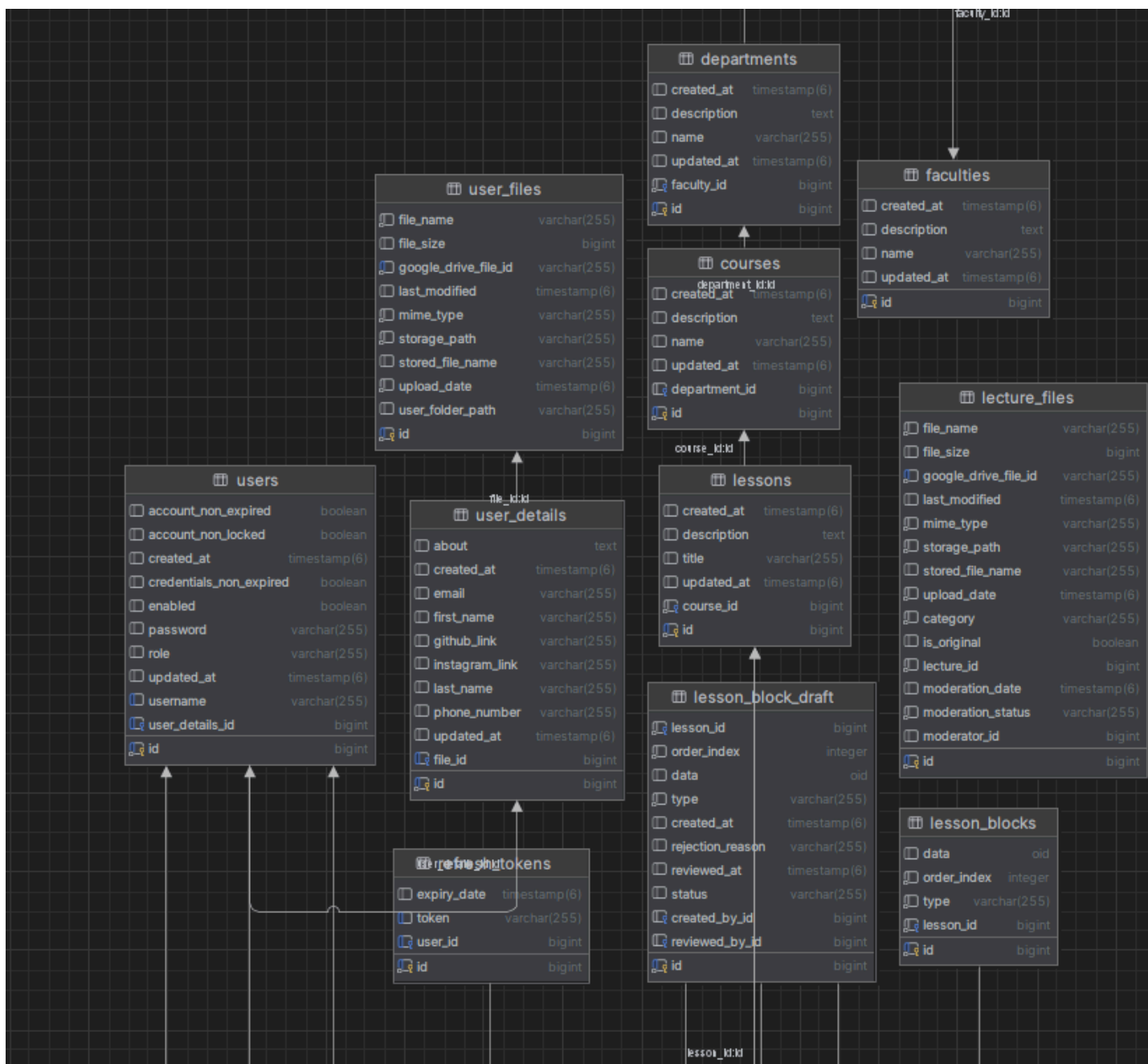
9. Общая архитектура системы



10. Gantt-diagram



11. ER-model



12) Цель написания этой программы – создать удобную платформу MKHub для студентов.

Кто будет использовать?

Студент – изучает, дополняет материалы, участвует в обсуждениях.

Почему это нужно?

1. Разрозненные материалы – всё собрано в одном месте.

2. Отсутствие базы знаний – доступ к проверенной информации.
3. Проверка знаний – тесты и задачи.
4. Самостоятельное изучение – доступ к материалам разных направлений.

13. Сценарии

User-Service: Сценарии работы с базой данных

1. Регистрация пользователя

Действие: Пользователь регистрируется на платформе.

- Вставка в **User**: создается новая запись с **id**, **name**, **email**, **password**, **role**, **created_at**, **updated_at**.

2. Обновление профиля пользователя

Действие: Пользователь обновляет свои данные (имя, фамилия, email).

- Обновление **User**: **name**, **email**, **updated_at** обновляются по **id**.

3. Изменение пароля

Действие: Пользователь меняет пароль.

- Обновление **User**: **password**, **updated_at** обновляются по **id**.

4. Назначение роли пользователю (администратором)

Действие: Администратор меняет роль пользователя.

- Обновление **User**: **role**, **updated_at** обновляются по **id**.

5. Удаление пользователя

Действие: Пользователь удаляет свой аккаунт.

- Удаление из **User** по **id**.

6. Получение информации о пользователе

Действие: Получение данных о пользователе по **id**.

- Выборка из **User**: **id**, **name**, **email**, **role**, **created_at**, **updated_at**.

Content-Service: Сценарии работы с базой данных

1. Создание факультета

Действие: Администратор добавляет новый факультет.

- Вставка в **Faculty**: создается новая запись с **id**, **name**, **description**, **created_at**, **updated_at**.

2. Обновление информации о факультете

Действие: Администратор изменяет данные факультета.

- Обновление **Faculty**: **name**, **description**, **updated_at** обновляются по **id**.

3. Удаление факультета

Действие: Администратор удаляет факультет.

- Удаление из **Faculty** по **id**.

4. Получение информации о факультете

Действие: Получение данных о факультете по **id**.

- Выборка из **Faculty**: **id**, **name**, **description**, **created_at**, **updated_at**.

5. Получение списка всех факультетов

Действие: Получение списка всех факультетов.

- Выборка из **Faculty**: все записи с **id**, **name**, **description**, **created_at**, **updated_at**.
-

1. Создание департамента

Действие: Администратор добавляет новый департамент.

- Вставка в **Department**: создается новая запись с **id**, **name**, **description**, **faculty_id** (связь с факультетом), **created_at**, **updated_at**.

2. Обновление информации о департаменте

Действие: Администратор изменяет данные департамента.

- Обновление в **Department**: `name`, `description`, `updated_at` обновляются по `id`.

3. Удаление департамента

Действие: Администратор удаляет департамент.

- Удаление из **Department**: удаляется запись по `id` департамента. Все курсы, относящиеся к этому департаменту, также удаляются (каскадное удаление).

4. Получение информации о департаменте

Действие: Получение данных о департаменте по `id`.

- Выборка из **Department**: `id`, `name`, `description`, `faculty_id`, `created_at`, `updated_at`.

5. Получение списка всех департаментов

Действие: Получение списка всех департаментов.

- Выборка из **Department**: все записи с `id`, `name`, `description`, `faculty_id`, `created_at`, `updated_at`.
-

1. Создание курса

Действие: Администратор добавляет новый курс в департамент.

- Вставка в **Course**: создается новая запись с `id`, `name`, `description`, `department_id` (связь с департаментом), `created_at`, `updated_at` также `moderator_id` (назначение модератора для курса).

2. Обновление информации о курсе

Действие: Администратор изменяет данные курса.

- Обновление в **Course**: `name`, `description`, `updated_at` обновляются по `id`.
- Изменение Модератора: обновим поле `moderator_id` на `id` иного модератора

3. Удаление курса

Действие: Администратор удаляет курс.

- Удаление из **Course**: удаляется запись по **id** курса. Все уроки, относящиеся к этому курсу, также удаляются (каскадное удаление).

4. Получение информации о курсе

Действие: Получение данных о курсе по **id**.

- Выборка из **Course**: **id, name, description, moderator_id, department_id, created_at, updated_at**.

5. Получение списка всех курсов

Действие: Получение списка всех курсов.

- Выборка из **Course**: все записи с **id, name, description, moderator_id, department_id, created_at, updated_at**.
-

1. Создание урока

Действие: Модератор добавляет новый урок для курса.

- Вставка в **Lesson**: создается новая запись с **id, title, description, course_id** (связь с курсом), **created_at, updated_at**.

2. Обновление информации об уроке

Действие: Модератор изменяет данные урока.

- Обновление в **Lesson**: **title, description, updated_at** обновляются по **id**.

3. Удаление урока

Действие: Модератор удаляет урок.

- Удаление из **Lesson**: удаляется запись по **id** урока. Все темы, относящиеся к этому уроку, также удаляются (каскадное удаление).

4. Получение информации об уроке

Действие: Получение данных об уроке по **id**.

- Выборка из **Lesson**: **id**, **title**, **description**, **course_id**, **created_at**, **updated_at**.

5. Получение списка всех уроков

Действие: Получение списка всех уроков.

- Выборка из **Lesson**: все записи с **id**, **title**, **description**, **course_id**, **created_at**, **updated_at**.
-

1. Создание темы

Действие: Модератор добавляет новую тему для урока.

- Вставка в **Topic**: создается новая запись с **id**, **title**, **description**, **lesson_id** (связь с уроком), **created_at**, **updated_at**.

2. Обновление информации о теме

Действие: Преподаватель изменяет данные темы.

- Обновление в **Topic**: **title**, **description**, **updated_at** обновляются по **id**.

3. Удаление темы

Действие: Преподаватель удаляет тему.

- Удаление из **Topic**: удаляется запись по **id** темы.

4. Получение информации о теме

Действие: Получение данных о теме по **id**.

- Выборка из **Topic**: **id**, **title**, **description**, **lesson_id**, **created_at**, **updated_at**.

5. Получение списка всех тем

Действие: Получение списка всех тем.

- Выборка из **Topic**: все записи с **id**, **title**, **description**, **lesson_id**, **created_at**, **updated_at**.
-

1. Создание тега

Действие: Администратор добавляет новый тег.

Вставка в Tag: создается новая запись с id, name, lesson_id (связь с уроком), created_at, updated_at.

2. Обновление информации о теге

Действие: Администратор изменяет данные тега.

Обновление в Tag: name, updated_at обновляются по id.

3. Удаление тега

Действие: Администратор удаляет тег.

Удаление из Tag: удаляется запись по id тега.

4. Получение информации о теге

Действие: Получение данных о теге по id.

Выборка из Tag: id, name, lesson_id , created_at, updated_at.

5. Получение списка всех тегов

Действие: Получение списка всех тегов.

Выборка из Tag: все записи с id, name, lesson_id , created_at, updated_at.

Content-Service: Сценарии работы с базой данных

1. Создание контента

Действие: Администратор добавляет новый контент.

- Вставка в Content: создается новая запись с id, topic_id (связь с темой), content_type_id (связь с типом контента), content, order, created_at, updated_at.

2. Обновление информации о контенте

Действие: Администратор изменяет данные контента.

- Обновление в Content: topic_id, content_type_id, content, order, updated_at обновляются по id.

3. Удаление контента

Действие: Администратор удаляет контент.

- Удаление из Content: удаляется запись по id контента.

4. Получение информации о контенте

Действие: Получение данных о контенте по id.

- Выборка из Content: id, topic_id, content_type_id, content, order, created_at, updated_at.

5. Получение списка всех контентов

Действие: Получение списка всех контентов.

- Выборка из Content: все записи с id, topic_id, content_type_id, content, order, created_at, updated_at.
-

1. Создание типа контента

Действие: Администратор добавляет новый тип контента.

- Вставка в ContentType: создается новая запись с id, title, created_at, updated_at.

2. Обновление информации о типе контента

Действие: Администратор изменяет данные типа контента.

- Обновление в ContentType: title, updated_at обновляются по id.

3. Удаление типа контента

Действие: Администратор удаляет тип контента.

- Удаление из ContentType: удаляется запись по id типа контента.

4. Получение информации о типе контента

Действие: Получение данных о типе контента по id.

- Выборка из ContentType: id, title, created_at, updated_at.

5. Получение списка всех типов контента

Действие: Получение списка всех типов контента.

- Выборка из ContentType: все записи с id, title, created_at, updated_at.