

TanyaJawab

Asisten pembelajaran cerdas yang didukung oleh Retrieval-Augmented Generation

Final Project Kelompok 14

Praktikum Sistem Basis Data

Anggota Kelompok

- Ryan Adidaru
(2306266994)
- Teufik Ali Hadzalic
(2306267012)
- Muhamad Rey Kafaka Fadlan
(2306250573)
- Adhikananda Wira Januar
(2306267113)

Overview

- Introduction
- Purpose
- Features
- System Workflow
- Architecture
- Entity Relationship Diagram
- UML Class Diagram
- Database Schema Highlights
- Future Works
- GitHub Repository



Introduction

TanyaJawab adalah asisten pintar berbasis web yang memungkinkan pengguna untuk mengunggah materi kuliah dan mengajukan pertanyaan tentangnya.

Dengan memanfaatkan teknologi AI, platform ini memberikan jawaban yang kontekstual dan berbasis dokumen, serta membantu pengguna mengelola mata kuliah, materi belajar, dan tugas dalam satu tempat.

Purpose

TanyaJawab dirancang untuk mengatasi tantangan umum yang dihadapi oleh siswa saat belajar dengan materi digital.

- Meningkatkan interaktivitas pembelajaran
Memungkinkan siswa mengajukan pertanyaan yang kontekstual berdasarkan materi studi mereka.
- Sentralisasi dokumen akademik
Mengorganisasi catatan kuliah, slide, dan PDF berdasarkan kursus.
- Menghemat waktu saat belajar
Memungkinkan pengambilan informasi relevan dari dokumen yang diunggah secara cepat menggunakan Retrieval-Augmented Generation

Features

Unggah & Kelola Dokumen

- Unggah slide kuliah, PDF, dan catatan, yang diorganisasi per matkul

Tanya Jawab Cerdas

- Pertanyaan akan dijawab menggunakan bagian yang relevan dari dokumen

Mesin Pencari Semantik

- Cari keyword dengan pencocokan yang mempertimbangkan konteks

Manajemen Kelas & Tugas

- Pengguna dapat memantau jadwal kelas dan tugas yang akan datang

Chat UI yang Cepat & Responsif

- Interface chat minimalis dengan pengalaman pengguna yang mulus

Otentikasi Pengguna

- Pengguna dapat log in menggunakan akun GitHub atau email

System Workflow

Pemrosesan Dokumen

- Pengguna mengunggah dokumen PDF (seperti catatan kuliah atau silabus).
- Sistem mengekstrak teks dan gambar menggunakan Gemini Vision API.
- Konten dibagi menjadi potongan-potongan kecil dan diubah menjadi vektor embedding.
- Embedding ini disimpan di Qdrant, database vektor untuk pencarian semantik.

Tanya Jawab Cerdas

- Saat pengguna mengajukan pertanyaan, sistem mencari potongan dokumen paling relevan di Qdrant.
- Potongan-potongan ini digunakan untuk membangun prompt yang kaya konteks.
- Prompt dikirim ke Large Language Model (LLM) untuk menghasilkan jawaban yang tepat dan terarah.
- Jawaban dikembalikan ke pengguna, lengkap dengan konteks dari dokumen asli.

Architecture

Frontend

React.js + Tailwind CSS, interface web yang responsif dan bersih untuk interaksi pengguna.

Backend

FastAPI (Python), server backend cepat dan ringan untuk menangani permintaan pengguna dan logika aplikasi.

Worker terpisah untuk pemrosesan berat seperti ekstraksi dokumen dan embedding.

Databases

PostgreSQL: Menyimpan data pengguna, jadwal kuliah, dokumen, dan tugas.

Qdrant (Vector DB): Menyimpan embedding dokumen untuk pencarian semantik.

Redis: Sebagai cache untuk mempercepat respons sistem.

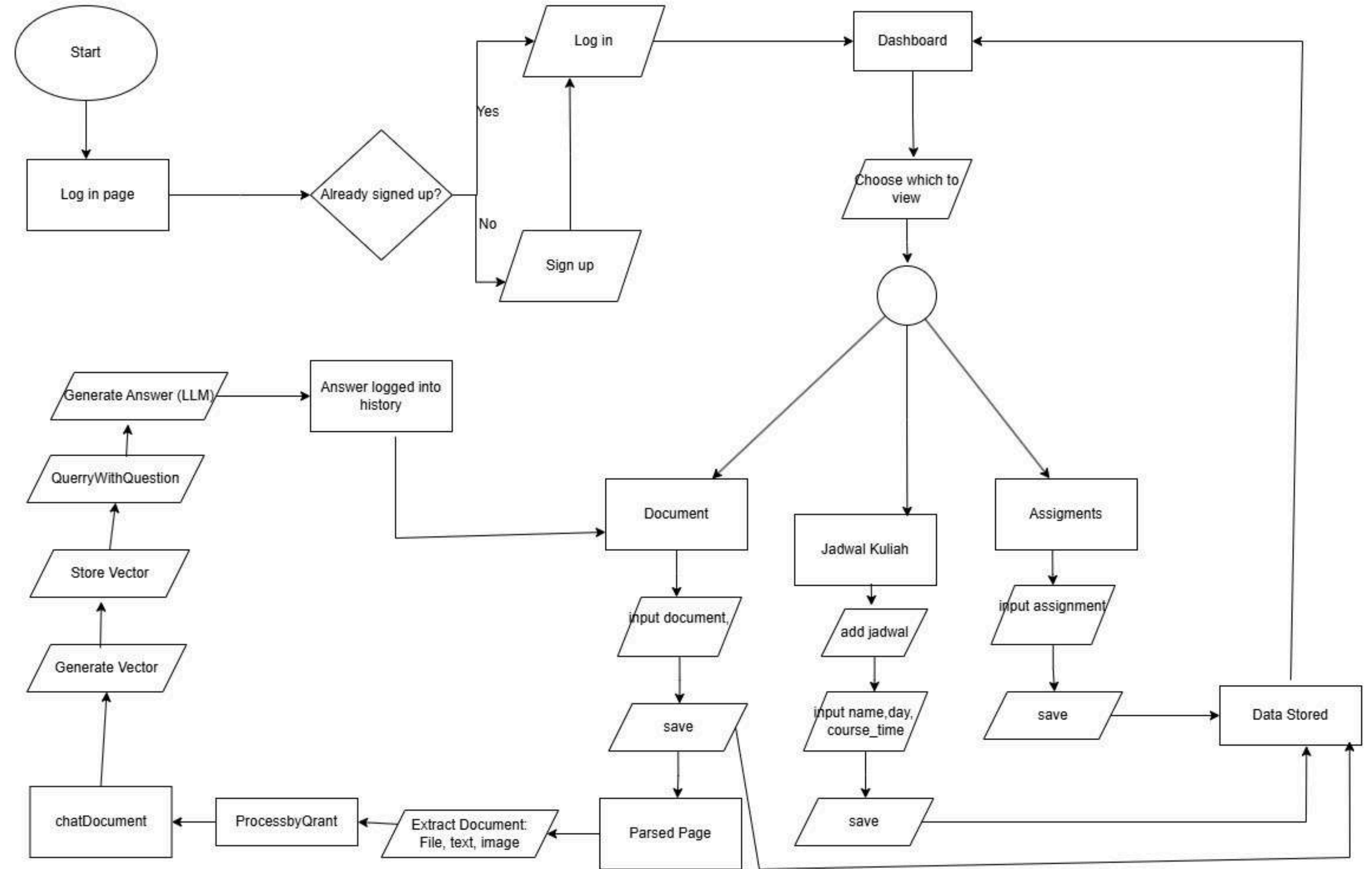
AI Integration

Gemini Vision API: Mengekstrak teks dan gambar dari dokumen PDF.

Model Embedding: Mengubah potongan dokumen menjadi vektor.

LLM API: Menghasilkan jawaban berdasarkan konteks dokumen dan pertanyaan pengguna.

FLOWCHART



FLOWCHART

Pengguna memulai di halaman login, dan jika belum mendaftar, diarahkan ke halaman pendaftaran (Sign Up) terlebih dahulu. Setelah berhasil login, pengguna masuk ke dashboard dan memilih untuk melihat atau mengelola Document, Jadwal Kuliah, atau Assignments.

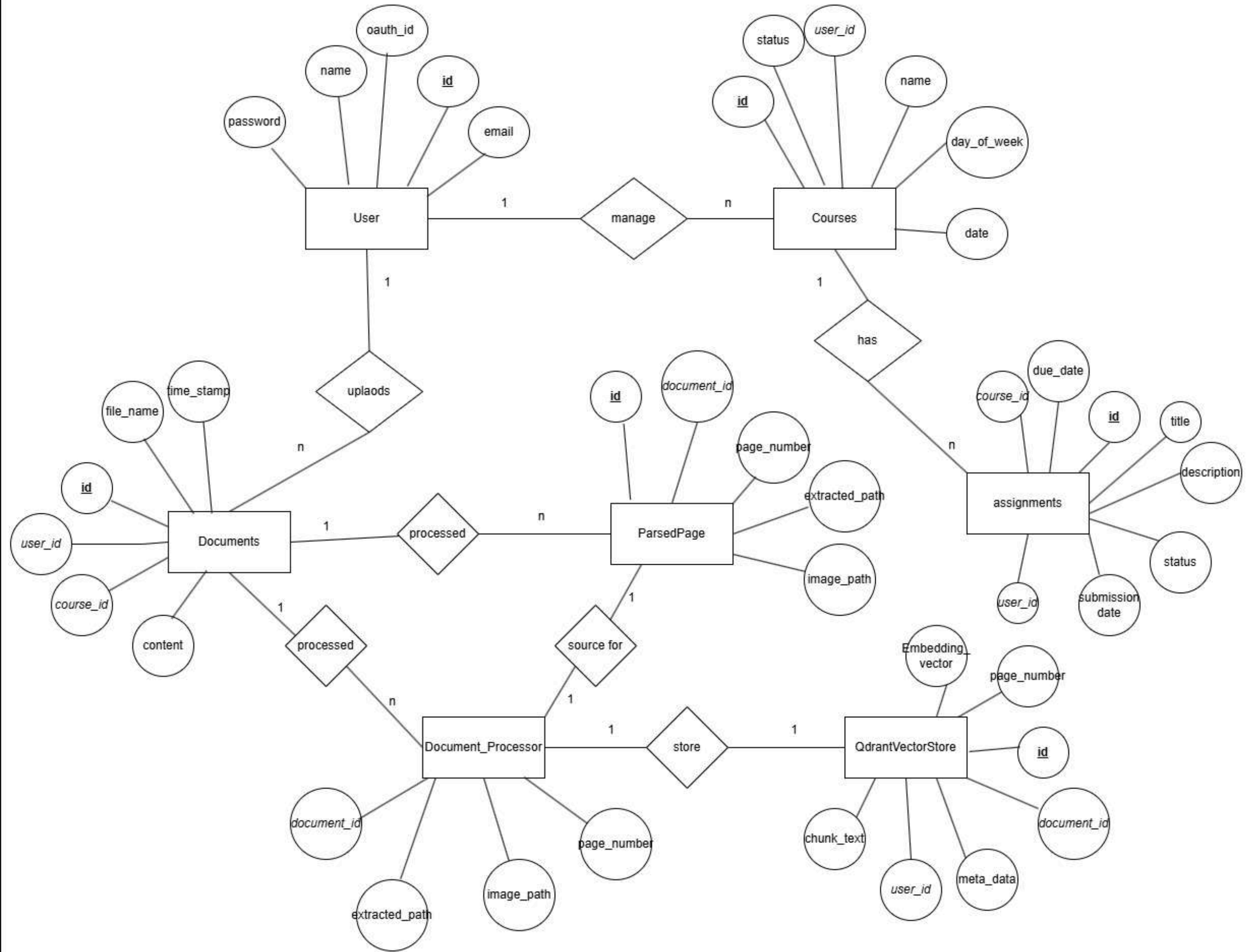
Untuk Dokumen, pengguna mengunggah dokumen yang kemudian diproses, diubah menjadi vektor, dan disimpan. Pengguna dapat mengajukan pertanyaan terkait dokumen, dan sistem akan menjawab menggunakan teknologi LLM serta mencatat jawaban tersebut.

Untuk Jadwal Kuliah, pengguna menambahkan jadwal baru dengan memasukkan detail mata kuliah dan waktu, lalu menyimpannya.

Untuk Tugas, pengguna memasukkan dan menyimpan tugas yang diperlukan.

Semua data jadwal dan tugas disimpan dan dapat diakses kembali. Setelah itu, pengguna kembali ke dashboard untuk aktivitas selanjutnya.

ERD



ERD

Main Entity:

- User: Mengelola akun pengguna, mengupload dokumen, dan mengatur Courses.
- Documents: Menyimpan dokumen yang diupload oleh pengguna terkait Courses.
- ParsedPage: Menyimpan halaman dokumen hasil parsing.
- Document_Processor: Memproses dokumen menjadi bagian yang dapat diparse perhalaman.
- QdrantVectorStore: Menyimpan representasi vektor dokumen untuk pencarian dan pencocokan teks.
- Courses: Menyimpan dan mengelola informasi kursus yang dimiliki pengguna.
- Assignments: Mengelola tugas yang diberikan dalam kursus, termasuk status dan pengumpulan.

ERD

Key relations:

- User – Documents:

Relasi 1 to many (1:N): Satu User bisa mengupload banyak dokumen.

- User – Courses:

Relasi 1 to many (1:N): Satu User bisa mengelola banyak Courses.

- Courses – Assignments:

Relasi 1 to many (1:N): Satu Courses bisa punya banyak tugas (Assignments).

- Documents – Document_Processor:

Relasi 1 to many (1:N): Satu dokumen diproses menjadi banyak bagian.

- Document_Processor – ParsedPage:

Relasi 1 to many (1:N): Hasil pemrosesan Document_Processor menghasilkan banyak halaman yang di-parse.

- ParsedPage – QdrantVectorStore:

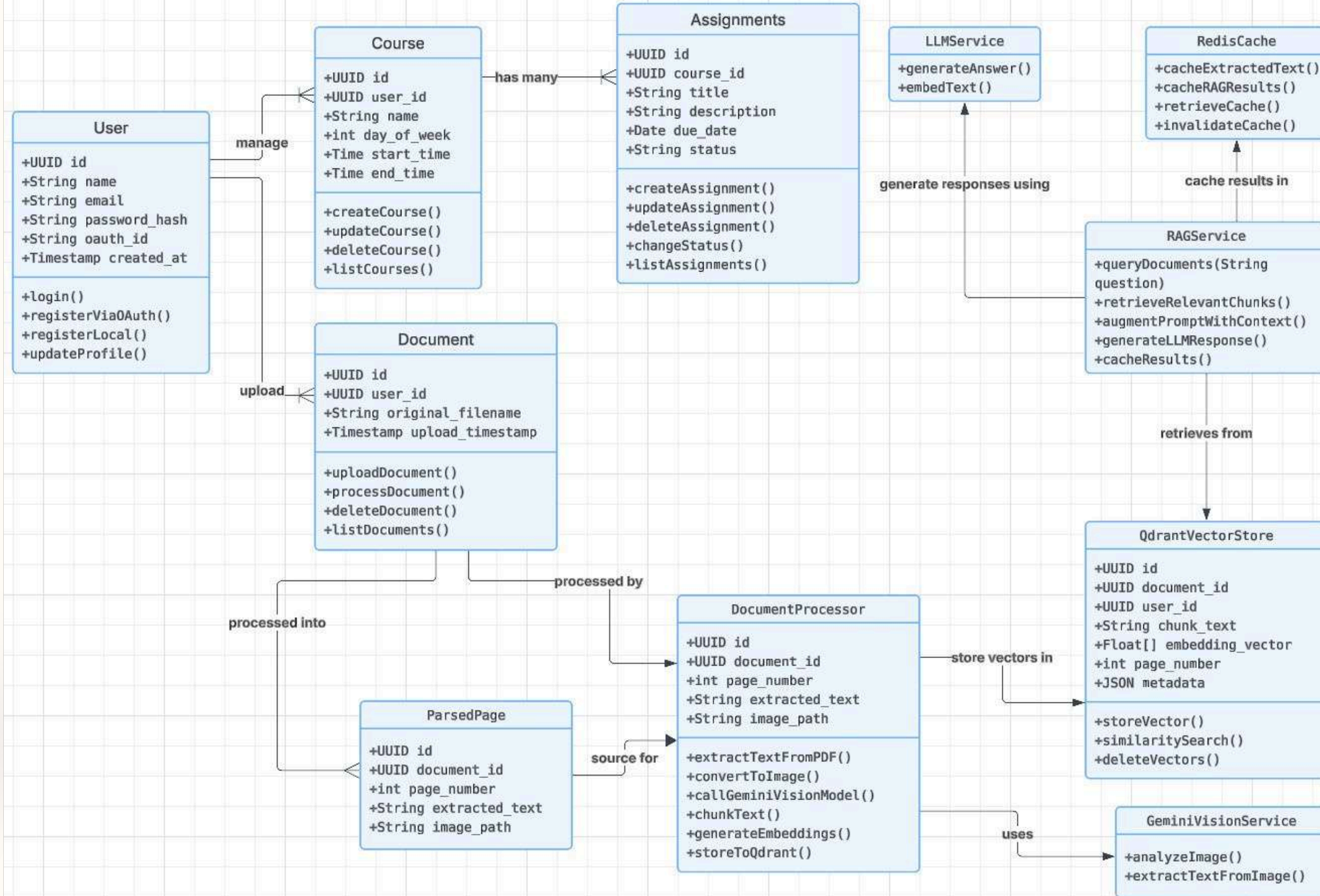
Relasi 1 to 1 (1:1): Setiap halaman yang diparse memiliki satu vektor embedding.

- Document_Processor – QdrantVectorStore

Relasi 1 to many (1:N): Satu Document_Processor dapat menghasilkan banyak vektor embedding yang disimpan di QdrantVectorStore.

UML

TanyaJawab UML Diagram



U M L

Kelas Penting:

- DocumentProcessor

Bertanggung jawab untuk parsing PDF, OCR (jika berupa gambar), pemecahan konten, dan pembuatan embedding.

- RAGService

Menangani pencarian semantik dan pengiriman pertanyaan ke LLM untuk mendapatkan jawaban.

- QdrantVectorStore

Mengatur komunikasi dengan Qdrant untuk menambahkan dan mencari embedding vektor.

Prinsip Desain:

- Pemisahan tanggung jawab (separation of concerns)
- Layanan yang dapat digunakan kembali (reusability)
- Arsitektur pipeline yang fleksibel dan mudah diperluas

Database Schema Highlights

Tabel Utama:

- users

Menyimpan informasi akun pengguna (GitHub OAuth atau lokal)

- courses

Jadwal mata kuliah pengguna, digunakan untuk pelacakan tugas dan manajemen dokumen

- assignments

Menyimpan daftar tugas dan tenggat waktu yang terkait dengan mata kuliah

- documents

Metadata dokumen yang diunggah (nama file, pemilik, waktu unggah)

- parsed_pages

Halaman hasil ekstraksi teks dari dokumen PDF, digunakan untuk pencarian semantik

Database Schema Highlights

Integrasi dengan Qdrant:

- Konten dari `parsed_pages` diubah menjadi embedding vektor dan disimpan dalam koleksi `doc_chunks` di Qdrant.

Integrasi Redis:

- Digunakan untuk menyimpan cache dari hasil ekstraksi dan jawaban RAG agar sistem lebih responsif.



Future Works

- Versi aplikasi mobile
- Dukungan untuk lebih banyak jenis dokumen (DOCX, PPTX)
- Sesi belajar kolaboratif
- Kartu flash spaced repetition yang dibuat dari dokumen
- Analitik lanjutan untuk kebiasaan belajar

GitHub Repository



<https://github.com/coolcmyk/TanyaJawab>

Thank You!

Final Project Kelompok 14

Praktikum Sistem Basis Data