Design Document – Vexere AI Chatbot (POC)

1. Kiến trúc tổng quan

Hệ thống chatbot được thiết kế để hỗ trợ ba cấp độ năng lực:

L-1 (FAQ): Trả lời các câu hỏi thường gặp bằng kỹ thuật RAG (Retrieval-Augmented Generation) sử dụng Gemini và FAISS.

L-2 (After-Service): Xử lý yêu cầu đổi giờ vé, hỗ trợ hội thoại nhiều lượt (multi-turn).

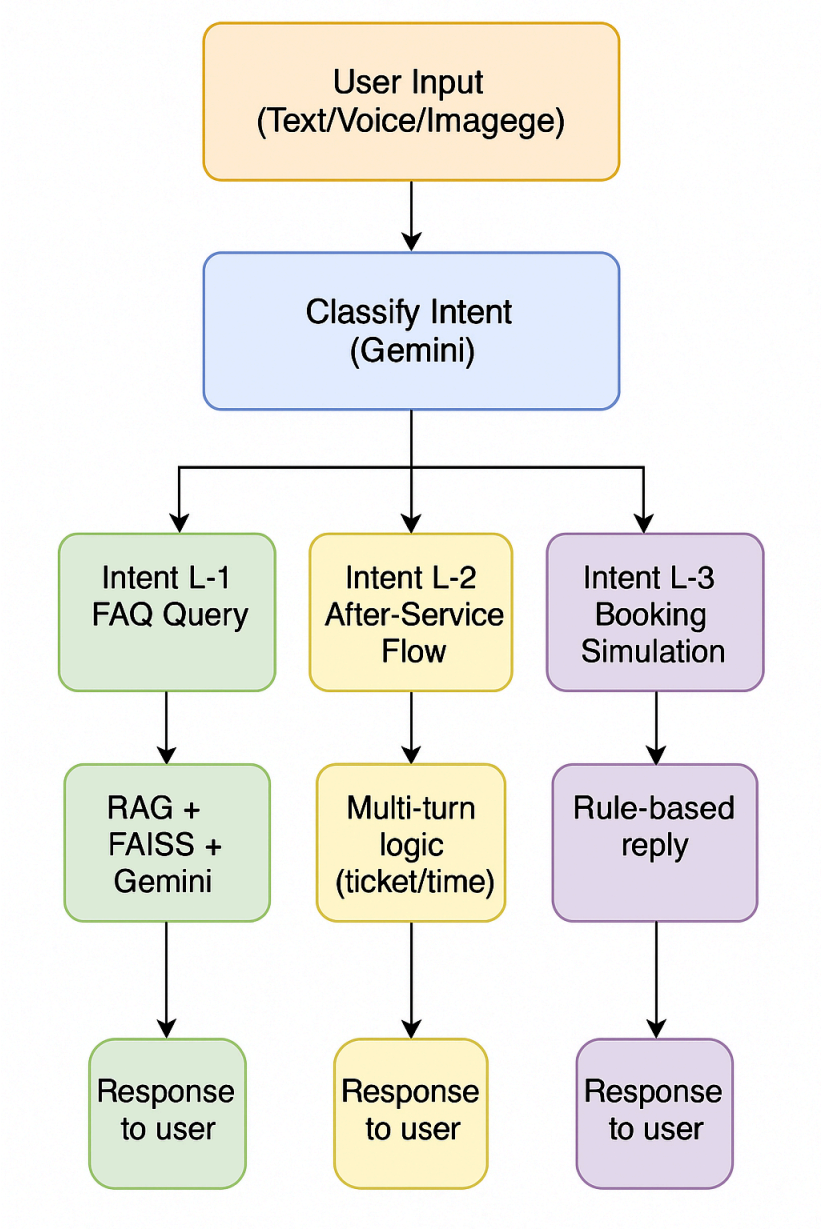
L-3 (Booking): Mô phỏng đặt vé, bao gồm chọn tuyến, giờ đi, thanh toán (chưa kết nối dữ liệu thực).

Hệ thống hỗ trợ các loại input: văn bản (text), giọng nói (voice) và hình ảnh (image). Giao diện người dùng sử dụng Gradio để dễ dàng trình diễn toàn bộ luồng xử lý.

2. Lựa chọn công nghệ và lý do

Thành phần	Công nghệ sử dụng	Lý do lựa chọn
Mô hình ngôn ngữ (LLM)	Gemini 2.0	Hỗ trợ API miễn phí, ổn định, có sẵn embedding
Vector DB	FAISS	Nhanh, nhẹ, phù hợp với demo RAG
Phân loại intent	Prompt-based Gemini	Không cần huấn luyện, dễ tinh chỉnh
API Backend	FastAPI	Nhẹ, async, dễ triển khai REST API
Giao diện người dùng	Gradio	Nhanh gọn, hỗ trợ chat, upload audio/image
Xử lý giọng nói (mocked)	PhoWhisper (định hướng)	Phù hợp tiếng Việt
Xử lý ảnh (mocked)	BLIP / PaddleOCR (định hướng)	Trích thông tin từ ảnh vé

3. Kiến trúc hệ thống



4. Kiểm thử và cải tiến liên tục

4.1. Kiểm thử chức năng

Việc kiểm thử được thực hiện trực tiếp qua giao diện Gradio, bao gồm:

- Luồng L-1 (FAQ): kiểm tra khả năng trả lời từ RAG bằng cách nhập các câu hỏi phổ biến như "Chính sách hoàn vé như thế nào?", "Tôi có được mang hành lý không?"

- Luồng L-2 (After-Service - đổi giờ): kiểm tra luồng hội thoại nhiều bước, ví dụ:
    1. "Tôi muốn đổi vé"
    2. "Mã vé ABC123"
    3. "15h30"
- Luồng L-3 (Booking): nhập câu như "Tôi muốn đặt vé đi Hà Nội lúc 13h" để kiểm tra phản hồi mô phỏng.

4.2. Kiểm thử logic (unit test - đề xuất): Viết test đơn vị với pytest cho các hàm hỗ trợ

4.3. Continuous Improvement (CI/CD - đề xuất)

- Code format: sử dụng black để tự động định dạng code
- Linting: sử dụng ruff để bắt lỗi nhanh và hiệu quả
- CI pipeline: có thể tích hợp GitHub Actions hoặc pre-commit để:
    - Tự động kiểm tra syntax, style khi push
    - Chạy pytest để đảm bảo logic không bị lỗi

5. Các hạng mục liên quan khác

- **Session:** Sử dụng biến session_state để ghi nhớ context tạm thời khi demo luồng đổi giờ (L-2)
- **Multi-turn:** Xử lý L-2 theo luồng hội thoại nhiều bước (mã vé → giờ → xác nhận)
- **Voice / Image:** Có sẵn route /upload-voice và /upload-image để mở rộng xử lý ảnh chụp vé, audio người dùng
- **Scalability:** Có thể mở rộng sang nhiều kênh (Zalo, Facebook, v.v.) thông qua webhook
- **Security & Logging:** Hiện chưa tích hợp nhưng có thể thêm FastAPI middleware để log, auth
- **Multi-user:** Mới hỗ trợ 1 session để demo, có thể mở rộng theo session_id nếu cần