

Báo cáo Kỹ thuật Dự án AI Mentor Feedback: Giai đoạn Phát triển Nhanh (Sprint) 10 ngày

AIHUB (Mô phỏng theo yêu cầu người dùng)

Ngày 15 tháng 11 năm 2025

Tóm tắt nội dung

Dự án này trình bày quá trình phát triển lặp (iterative development) của một hệ thống chatbot cố vấn (AI Mentor Feedback) trong 10 ngày. Mục tiêu là xây dựng một tác nhân AI có khả năng tương tác, thu thập phản hồi, và cung cấp gợi ý học tập cho người dùng qua nền tảng Telegram. Báo cáo này ghi lại quá trình phát triển từ Giai đoạn 1 (v1.0), sử dụng Google Sheets làm cơ sở dữ liệu (CSDL) chính, đến Giai đoạn 2 (v2.0), với việc tái cấu trúc hệ thống và di chuyển sang một module thu thập dữ liệu (collector module) chuyên biệt sử dụng SQLite. Các mốc quan trọng bao gồm việc triển khai logic AI dựa trên quy tắc (rule-based), phát triển bộ nhớ ngữ cảnh (stateful context), tích hợp vòng lặp phản hồi của người dùng (feedback loop), và xây dựng một hệ thống gợi ý (suggestion engine) có khả năng học hỏi.

Mục lục

1 Giới thiệu	2
2 Phương pháp luận và Triển khai Chi tiết	2
2.1 Day 01-02: Lên ý tưởng và Thiết kế Hệ thống (v1.0)	2
2.2 Day 03: Thiết lập Đường ống Dữ liệu (Pipeline)	2
2.3 Day 04: AI Mentor v1.0 (Logic dựa trên Quy tắc)	2
2.4 Day 05: Cơ chế Báo cáo v1.0	3
2.5 Day 06: AI v1.1 (Phân tích Hành vi)	3
2.6 Day 07: Nâng cấp v1.2 (Vòng lặp Phản hồi)	3
2.7 Day 08: Nâng cấp v1.3 (Xử lý Trường hợp biên)	3
2.8 Day 09: AI v1.4 (Learning & Suggestion Engine)	3
2.9 Day 10: AI Feedback Collector v2.0 (Tái cấu trúc)	4
3 Kết luận và Hướng phát triển Tương lai	4

1 Giới thiệu

Trong bối cảnh giáo dục trực tuyến và tự học phát triển mạnh mẽ, nhu cầu về các công cụ hỗ trợ học tập cá nhân hóa và tức thời ngày càng tăng. Dự án "AI Mentor Feedback" được khởi xướng nhằm giải quyết vấn đề này bằng cách cung cấp một chatbot có vấn AI trên Telegram. Chatbot này không chỉ trả lời các truy vấn mà còn chủ động thu thập dữ liệu về quá trình học tập, phân tích hành vi, và đưa ra các gợi ý phù hợp.

Tài liệu này ghi lại chi tiết quá trình phát triển nhanh (sprint) trong 10 ngày đầu tiên, phác thảo các quyết định kiến trúc, các thách thức kỹ thuật, và các giải pháp được triển khai.

2 Phương pháp luận và Triển khai Chi tiết

Quá trình phát triển được chia thành các giai đoạn logic, tập trung vào việc xây dựng và cải tiến lặp.

2.1 Day 01-02: Lên ý tưởng và Thiết kế Hệ thống (v1.0)

Giai đoạn khởi đầu tập trung vào việc xác định yêu cầu:

- **Nền tảng:** Telegram (do tính phổ biến và API mạnh mẽ).
- **Lưu trữ (v1.0):** Google Sheets (dễ dàng thiết lập, không cần máy chủ CSDL, và có khả năng trực quan hóa tức thì cho Day 05).
- **Ngăn xếp Công nghệ:** Python 3, thư viện python-telegram-bot, và gspread (để tương tác với Google Sheets API).

Thiết kế ban đầu là một đường ống dữ liệu (data pipeline) đơn giản: *Telegram ↔ PythonScript ↔ GoogleSheets*.

2.2 Day 03: Thiết lập Đường ống Dữ liệu (Pipeline)

Nhiệm vụ trọng tâm là kết nối 3 thành phần. Thách thức kỹ thuật chính là xác thực với Google API. Giải pháp ban đầu (Service Account) gặp phải chính sách bảo mật của tổ chức, buộc phải chuyển sang giải pháp **OAuth 2.0 Client ID** (cho "Desktop app").

- **Kết quả:** Script Python có khả năng nhận tin nhắn từ Telegram, xử lý, và ghi log (timestamp, user_id, message_text) vào file Google Sheet Log_v1 thành công.

2.3 Day 04: AI Mentor v1.0 (Logic dựa trên Quy tắc)

"Bộ não" AI đầu tiên được triển khai. Đây là một hệ thống dựa trên quy tắc (Rule-Based System) đơn giản.

- **Triển khai:** Một hàm Python `get_ai_feedback_v1_0()` sử dụng logic `if-elif-else` để phát hiện từ khóa (keywords) trong tin nhắn của người dùng (ví dụ: "xin chào", "lỗi", "python").
- **Kết quả:** Bot có thể đưa ra các phản hồi ngữ cảnh cơ bản thay vì chỉ ghi log.

2.4 Day 05: Cơ chế Báo cáo v1.0

Để đánh giá hiệu suất của bot và thu thập thông tin chi tiết (insights), một dashboard đơn giản đã được xây dựng trực tiếp trên Google Sheets.

- **Triển khai:** Sử dụng các hàm tích hợp của Google Sheets: COUNTA (đếm tổng tin nhắn), COUNTUNIQUE (đếm người dùng), và **Pivot Tables** (Bảng tổng hợp).
- **Kết quả:** Một dashboard trực quan thống kê tần suất phản hồi của AI và hoạt động của người dùng theo ngày.

2.5 Day 06: AI v1.1 (Phân tích Hành vi)

Nâng cấp AI v1.0 (stateless) lên v1.1 (stateful) bằng cách thêm "trí nhớ".

- **Triển khai:** Sử dụng tính năng context.user_data của thư viện python-telegram-bot. Đây là một dictionary lưu trữ dữ liệu tạm thời cho mỗi người dùng.
- **Logic mới:** AI v1.1 (get_ai_feedback_v1_1()) nhận vào lịch sử trò chuyện (danh sách tin nhắn trước đó) và có thể hiểu được bối cảnh (ví dụ: nếu AI vừa hỏi về "lỗi", tin nhắn tiếp theo của người dùng được giả định là mô tả về lỗi đó).

2.6 Day 07: Nâng cấp v1.2 (Vòng lặp Phản hồi)

Để thu thập dữ liệu huấn luyện (training data) và đánh giá chất lượng phản hồi, một vòng lặp phản hồi đã được tích hợp.

- **Triển khai:** Sử dụng InlineKeyboardMarkup và CallbackQueryHandler.
- **Luồng (Flow):** Sau mỗi phản hồi của AI, bot gửi kèm 2 nút: " Hữu ích" và " Không hữu ích".
- **Lưu trữ:** Kết quả (timestamp, user_id, ai_text, rating) được ghi vào một sheet mới: Feedback_Log.

2.7 Day 08: Nâng cấp v1.3 (Xử lý Trường hợp biên)

Hệ thống được làm "cứng cáp" (robust) hơn bằng cách xử lý các đầu vào không mong muốn và các lỗi API tiềm ẩn.

- **Xử lý Edge Case:** Thêm các MessageHandler riêng để xử lý tin nhắn không phải văn bản (sticker, ảnh) và các lệnh không xác định (/help, v.v.).
- **Xử lý Lỗi:** Triển khai các khối try...except xung quanh tất cả các lệnh gọi API (sheet.append_row()), ngăn bot bị sập (crash) khi Google API tạm thời gián đoạn.

2.8 Day 09: AI v1.4 (Learning & Suggestion Engine)

Đây là một nâng cấp lớn, biến bot từ bị động (reactive) sang chủ động (proactive).

- **Database Gợi ý:** Một sheet mới, Content_DB, được tạo để lưu trữ các tài nguyên học tập (từ khóa, link, text, suggestion_id).
- **Suggestion Engine:** Một hàm get_suggestion_engine() mới được viết, có khả năng quét tin nhắn của người dùng để tìm từ khóa khớp với Content_DB.

- **Learning (Học tập):** Khi người dùng đánh giá ""hoặc ""một gợi ý, bot sẽ cập nhật lại cột Rating_Score của gợi ý đó trong Content_DB. Các gợi ý có điểm cao hơn sẽ được ưu tiên trong các lần sau.
- **Thách thức:** Đã phát hiện và sửa một bug (lỗi) nghiêm trọng trong logic phân tích callback_data (do xử lý sai ký tự gạch dưới _ trong suggestion_id).

2.9 Day 10: AI Feedback Collector v2.0 (Tái cấu trúc)

Ghi nhận một vấn đề nghiêm trọng về hiệu suất: Các lệnh gọi API gspread (đặc biệt là update_cell cho "Learning" ở Day 09) rất chậm và không đáng tin cậy.

- **Giải pháp:** Tái cấu trúc (refactor) hệ thống. Chuyển từ Google Sheets sang CSDL SQLite cục bộ.
- **Module v2.0:** Xây dựng một file Python độc lập db_collector.py, chứa class CollectorV2. Module này đóng gói (encapsulate) tất cả các tương tác CSDL (setup, log_message, log_feedback, get_all_content, update_suggestion_score).
- **Di chuyển dữ liệu:** Viết một script migrate.py (chạy một lần) để chuyển dữ liệu từ sheet Content_DB sang bảng content_db trong file aimentor.db.
- **Kết quả:** Bot chính được cập nhật để import CollectorV2. Các lệnh gọi API gspread được thay thế bằng các lệnh gọi hàm SQLite cục bộ, giúp tăng tốc độ phản hồi và độ tin cậy lên đáng kể.

3 Kết luận và Hướng phát triển Tương lai

Sau 10 ngày, dự án đã phát triển thành công từ một nguyên mẫu (prototype) thu thập dữ liệu đơn giản (v1.0) thành một hệ thống v2.0 mạnh mẽ, có trạng thái, có khả năng học hỏi, và sử dụng kiến trúc CSDL SQLite hiệu suất cao.

Các bước phát triển tiếp theo (từ Day 11 trở đi) sẽ tập trung vào:

1. **Day 11 (Smart Scheduler):** Triển khai một bộ lập lịch thông minh (ví dụ: APScheduler) để bot có thể chủ động gửi tin nhắn (ví dụ: nhắc nhở học tập) thay vì chỉ chờ người dùng.
2. **Day 13 (AI Mentor v3.0):** Huấn luyện một mô hình AI/ML thực thụ (thay vì rule-based) dựa trên tập dữ liệu đã được dán nhãn (labeled data) thu thập từ Feedback_Log.

Tài liệu

- [1] The python-telegram-bot Team. (2024). *Python Telegram Bot Library*. <https://python-telegram-bot.org/>
- [2] Burnash, Anton. (2022). *gspread - Google Sheets Python API*. <https://gspread.readthedocs.io/>
- [3] SQLite Consortium. (2024). *SQLite Documentation*. <https://www.sqlite.org/docs.html>