

Báo cáo Chi tiết Thiết kế Ứng dụng Flash Card

1. Case Study

Mô tả Hệ thống

Hệ thống "Flash Card" là một ứng dụng web, được xây dựng theo mô hình MVC (Model-View-Controller) trên nền tảng Java Servlet/JSP, nhằm hỗ trợ người dùng tạo, quản lý và luyện tập với các bộ thẻ học (flashcards) theo chủ đề (Category) riêng biệt. Hệ thống hỗ trợ quản lý tài khoản, lưu trữ dữ liệu cá nhân và cung cấp giao diện thân thiện để tối ưu trải nghiệm học tập.

Business Rules (Quy tắc Nghiệp vụ)

- Quản lý Tài khoản:** Mỗi người dùng phải đăng ký tài khoản và đăng nhập để truy cập tính năng cá nhân.
- Phân cấp:** Dữ liệu được tổ chức theo phân cấp **1-nhiều** (One-to-Many): Người dùng → Chủ đề (Category) → Bộ thẻ (Deck) → Thẻ học (Card).
- CRUD:** Người dùng có quyền Tạo, Đọc, Sửa, Xóa (CRUD) đối với Chủ đề, Bộ thẻ và Thẻ học của mình.
- Luyện tập:** Hệ thống hỗ trợ luyện tập bằng cách hiển thị thẻ ngẫu nhiên hoặc tuần tự.
- Trạng thái phiên:** Hệ thống sử dụng Session để duy trì ID của Chủ đề và Bộ thẻ hiện tại, phục vụ cho việc điều hướng "Quay lại" và "Tiếp theo" mượt mà (stateless navigation).

2. Database Design

Logical & Physical Design

Cơ sở dữ liệu bao gồm 4 bảng chính, phản ánh mối quan hệ dựa trên sơ đồ ERD .

Tên Bảng (Table)	Khóa Chính (PK)	Khóa Ngoại (FK)	Kiểu dữ liệu	Mô tả
Users	user_id	-	INT	Khóa chính, ID người dùng.
		-	VARCHAR	email, password, name
Categories	category_id	user_id	INT, VARCHAR	Chủ đề, liên

				kết đến User sở hữu.
Decks	deck_id	category_id	INT, VARCHAR	Bộ thẻ, liên kết đến Category.
		-	VARCHAR	deck_name, description
Cards	card_id	deck_id	INT, VARCHAR	Thẻ học, liên kết đến Deck.
		-	VARCHAR	front, back, example

3. System Design

Kiến trúc (Architecture)

Ứng dụng tuân thủ nghiêm ngặt mô hình **Model-View-Controller (MVC)**:

- **Controller (Servlets)**: Xử lý yêu cầu HTTP (doGet/doPost), điều phối luồng, gọi DAO, quản lý Session/Scope và chọn View (JSP).
 - Ví dụ: CategoryServlet, DeckServlet, ManageDeckServlet.
- **Model (DAO & Java Beans)**: Chứa dữ liệu (Java Beans: User, Deck, Card) và logic tương tác với database (DAO) sử dụng JDBC.
- **View (JSP & JSTL)**: Hiển thị giao diện người dùng dựa trên dữ liệu từ Controller.

Công nghệ Sử dụng (Stack)

- **Backend Core**: Java Servlet 5.0+ (Jakarta EE)
- **View Layer**: JavaServer Pages (JSP) và JSTL (Standard Tag Library)
- **Data Access**: JDBC (Java Database Connectivity)
- **Web Container**: Apache Tomcat 10.1+

Coding Convention

- **Đặt tên**: Sử dụng camelCase cho biến, phương thức; PascalCase cho tên lớp.
- **Chức năng**: Tuân thủ mô hình MVC rõ ràng, không lẫn lộn logic nghiệp vụ trong View.
- **Comment**: Code có comment giải thích các đoạn logic phức tạp hoặc các phương thức chính.

4. Sơ đồ Trang (Sitemap)

Sitemap thể hiện luồng điều hướng và cách các Servlet sử dụng Session để duy trì trạng thái.

1. **/login / /register**
2. **/homePage** (Dashboard)
3. **/CategoryDirect** (Hiển thị danh sách Categories)
 - **/category?id={ID}** (Chọn Category: Lưu ID vào Session)
 - **/category** (Nút Quay lại: Lấy ID từ Session)
 - **/deck?id={ID}** (Chọn Deck: Lưu ID Deck vào Session)
 - **/deck?action=next/prev** (Điều hướng thẻ: Lấy ID Deck & Index từ Session)
 - **/manageDeck?id={ID}** (Quản lý thẻ: Lấy ID Deck từ Request/Session)

5. Kết luận và Cải tiến Tương lai

Ưu điểm: Kiến trúc MVC rõ ràng, đã tối ưu hóa việc sử dụng Session để điều hướng, mang lại trải nghiệm người dùng mượt mà hơn.

Cải tiến Tương lai: Tích hợp logic luyện tập thẻ nâng cao (như spaced repetition), cải tiến UI/UX di động, và thêm chức năng chia sẻ bộ thẻ.