**Báo cáo chi tiết cho dự án ứng dụng Flash Card bằng Java Web**

**1. Case Study**

Hệ thống "Flash Card" là ứng dụng web giúp người dùng học và luyện tập từ vựng, ngữ pháp ngôn ngữ qua phương pháp thẻ học. Người dùng có thể tạo các bộ thẻ riêng, luyện tập ghi nhớ và đánh giá tiến độ học tập. 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**

* Mỗi người dùng đăng ký tài khoản riêng.
* Người dùng có thể tạo, sửa, xóa bộ thẻ và thẻ học.
* Lịch sử luyện tập được lưu lại để đánh giá tiến độ.
* Hệ thống giới hạn kích thước tối đa từng bộ thẻ để đảm bảo hiệu suất.

**2. Database Design**

**Logical Design**

* Bảng User: user\_id, username, password\_hash, email, created\_at
* Bảng Deck (Bộ thẻ): deck\_id, user\_id (FK), title, description, created\_at
* Bảng Card (Thẻ học): card\_id, deck\_id (FK), front\_content, back\_content, created\_at
* Bảng Practice\_History: history\_id, user\_id (FK), card\_id (FK), status, practiced\_at

**Physical Design**

* Sử dụng MySQL với các kiểu dữ liệu phù hợp: INT cho id, VARCHAR cho text, TIMESTAMP cho thời gian.
* Thiết lập khoá chính, khoá ngoại và các index để tối ưu truy vấn.

**3. System Design**

**Kiến trúc MVC**

* Model: Đại diện cho dữ liệu và logic nghiệp vụ, tương tác với database qua DAO.
* View: JSP hoặc Thymeleaf để xây dựng giao diện người dùng.
* Controller: Servlet hoặc Spring Controller xử lý yêu cầu từ user, điều phối giữa Model và View.

**Các diagram (đính kèm)**

* Deployment Diagram: Mô tả triển khai ứng dụng trên server web và database.
* Class Diagram: Các lớp chính như User, Deck, Card, PracticeHistory, Controller, DAO.
* Sequence Diagram: Minh họa luồng xử lý khi người dùng đăng nhập và luyện tập thẻ.
* State Chart: Trạng thái của một thẻ học (Mới, Đã học, Đã nhớ, Cần ôn lại).

**Công nghệ mới**

* Sử dụng Spring Boot để nhanh chóng xây dựng ứng dụng web.
* Sử dụng JPA/Hibernate để quản lý ORM, giúp tương tác database hiệu quả.
* Thymeleaf làm engine hiển thị tương tác người dùng.

**Coding Convention**

* Đặt tên biến, phương thức rõ ràng, theo chuẩn camelCase.
* Mỗi class có header comment mô tả chức năng.
* Code có comment giải thích những đoạn phức tạp.
* Tuân thủ mô hình MVC rõ ràng, tránh để lẫn business logic trong View.

**4. Screenshot và Sitemap**

* Đính kèm ảnh màn hình trang đăng nhập, trang chính (danh sách bộ thẻ), màn hình luyện tập.
* Sitemap thể hiện mối liên kết giữa các trang: Login -> Dashboard -> Deck Management -> Card Practice.

**5. Conclusion and Discussion**

* Ưu điểm: Ứng dụng dễ dùng, thiết kế rõ ràng, hỗ trợ luyện tập hiệu quả.
* Nhược điểm: Giao diện còn đơn giản, chưa hỗ trợ đa ngôn ngữ hoặc các tính năng nâng cao.
* Học được: Kiến trúc MVC, ứng dụng Spring Boot, kỹ năng thiết kế database và lập trình web.
* Cải tiến tương lai: Thêm chức năng chia sẻ bộ thẻ, tích hợp AI hỗ trợ luyện tập cá nhân hóa, cải tiến UI/UX.

**Tài liệu kèm theo:**

* Toàn bộ code Java web (đã comment), cấu trúc thư mục, file cấu hình.
* Slide trình bày nội dung và demo hệ thống.