

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



## KIỂM TRA PHẦN MỀM

---

Bài tập lớn 1

# Code Review

---

Giáo viên hướng dẫn: PGS. TS. Bùi Hoài Thắng  
Sinh viên thực hiện: Bùi Trọng Văn – 2213915  
Phạm Ngọc Long – 2211894  
Nguyễn Trần Yến Nhi – 2312506  
Bùi Trọng Hiên – 2310993  
Đặng Ngọc Bảo Trâm – 2213568

Thành phố Hồ Chí Minh, Tháng 2 - 2026



## Mục lục

<b>1</b>	<b>Giới thiệu</b>	<b>2</b>
<b>2</b>	<b>Phân công</b>	<b>2</b>
<b>3</b>	<b>Quy trình Review</b>	<b>3</b>
<b>4</b>	<b>Tóm tắt kết quả</b>	<b>4</b>
<b>5</b>	<b>Một số lỗi khác không nằm trong Checklist</b>	<b>4</b>
5.1	Lỗi [] . . . . .	4
5.2	Lỗi [] . . . . .	5
<b>6</b>	<b>Kết luận</b>	<b>5</b>



## 1 Giới thiệu

Báo cáo này trình bày kết quả đánh giá kỹ thuật mã nguồn (Technical Code Review) cho dự án **OHS**. Dựa trên bộ *Java Code Review Checklist*, quá trình kiểm thử tập trung rà soát từ khai báo biến, cấu trúc điều khiển đến các lỗ hổng bảo mật như SQL Injection.

Quá trình review các tệp tin Java trong thư mục `com/` được thực hiện kết hợp giữa đánh giá thủ công và công cụ AI để đảm bảo tính khách quan. Để tối ưu độ chính xác, nhóm áp dụng quy trình kiểm chứng chéo (Cross-validation): mỗi file mã nguồn sau khi được một thành viên review sẽ được một thành viên khác thẩm định lại trước khi tổng hợp kết quả cuối cùng.

## 2 Phân công

Thành viên	Files	LOC	% đóng góp
Bùi Trọng Văn 2213915	<b>Reviewed files:</b>	<b>1448</b>	100%
	<code>com/ohs/utils/JDBCConnectionManager.java</code>	64	
	<code>com/ohs/utils/JDBCUtility.java</code>	66	
	<code>com/ohs/services/OtpService.java</code>	28	
	<code>com/ohs/services/MD5.java</code>	33	
	<code>com/ohs/services/AdminService.java</code>	1257	
	<b>Validated files:</b> files reviewed by Phạm Ngọc Long		
Phạm Ngọc Long 2211894	<b>Reviewed files:</b>	<b>1452</b>	100%
	<code>com/ohs/models/Appointment.java</code>	626	
	<code>com/ohs/services/AppointmentService.java</code>	261	
	<code>com/ohs/models/Patients.java</code>	236	
	<code>com/ohs/services/PatientService.java</code>	329	
	<b>Validated files:</b> files reviewed by Bùi Trọng Văn		
Nguyễn Trần Yến Nhi 2312506	<b>Reviewed files:</b>	<b>1419</b>	100%
	<code>com/ohs/models/Doctors.java</code>	403	
	<code>com/ohs/services/DoctorService.java</code>	512	
	<code>com/ohs/services/LoginService.java</code>	394	
	<code>com/ohs/models/Status.java</code>	43	
	<code>com/ohs/services/StatusService.java</code>	67	
	<b>Validated files:</b> files reviewed by Bùi Trọng Hiển		

Thành viên	Files	LOC	% đóng góp
Bùi Trọng Hiển 2310993	<b>Reviewed files:</b>	<b>1516</b>	100%
	com/ohs/models/Prescription.java	426	
	com/ohs/services/PrescriptionService.java	250	
	com/ohs/models/Admin.java	840	
	<b>Validated files:</b> files reviewed by Đặng Ngọc Bảo Trâm		
Đặng Ngọc Bảo Trâm 2213568	<b>Reviewed files:</b>	<b>1395</b>	100%
	com/ohs/services/ApiService.java	102	
	com/ohs/models/Department.java	73	
	com/ohs/services/DepartmentService.java	88	
	com/ohs/services/MailService.java	116	
	com/ohs/models/Users.java	1016	
	<b>Validated files:</b> files reviewed by Nguyễn Trần Yến Nhi		

### 3 Quy trình Review

Nhóm thực hiện quy trình kiểm thử qua 4 giai đoạn:

- Phân chia các file Java trong thư mục com/ cho từng thành viên đảm bảo khối lượng > 300 LOC/người.
- Đối chiếu mã nguồn với bộ Checklist để phát hiện lỗi logic, định dạng và bảo mật.
- Sử dụng công cụ AI để quét nhanh các lỗi tiềm ẩn và hỗ trợ viết các đề xuất sửa đổi.
- Mỗi thành viên kiểm tra lại kết quả của thành viên khác để đảm bảo tính khách quan và xác thực các lỗi tìm thấy trước khi đưa vào file Excel.

Nhóm đã tích hợp AI trong quá trình review như sau:

- Công cụ: Google Gemini / ChatGPT (Phiên bản miễn phí).
- Cách thức áp dụng: Nhóm cung cấp đoạn mã nguồn kèm các tiêu chí từ Checklist để AI phân tích lỗi kỹ thuật; đồng thời, công cụ này cũng được sử dụng như một trợ lý ngôn ngữ để hiệu chỉnh các mô tả lỗi và đề xuất sửa đổi sao cho súc tích, chuyên nghiệp hơn.

- Kiểm soát kết quả: Nhóm không chấp nhận hoàn toàn phản hồi từ AI. Mọi gợi ý đều được đối chiếu lại với logic thực tế của dự án và kiến thức chuyên môn để loại bỏ các lỗi do AI suy diễn sai ngữ cảnh.

## 4 Tóm tắt kết quả

Sau khi rà soát các tệp nguồn trong thư mục `com/`, nhóm đã tổng hợp và phân loại các lỗi dựa trên bộ *Java Code Review Checklist*. Tổng cộng phát hiện được **[Tổng số]** lỗi, được tóm tắt qua bảng dưới đây:

Nhóm mục tiêu (Objective)	Số lượng lỗi	Tỷ lệ (%)
I. Defect Objective	[Số]	[%]
II. Ambiguity Objective	[Số]	[%]
III. Redundance Objective	[Số]	[%]
<b>Tổng cộng</b>	<b>[Tổng]</b>	<b>100%</b>

Dưới đây là phân tích chi tiết các nhóm lỗi có tần suất xuất hiện cao nhất:

- **Nhóm lỗi [Mã số - Tên nhóm, VD: I.1 Variable Declaration]:** Chiếm [%] tổng số lỗi. Các vi phạm chủ yếu tập trung vào [ví dụ: quy tắc đặt tên camelCase, biến chưa được khởi tạo].
- **Nhóm lỗi [Mã số - Tên nhóm, VD: I.9 Comment]:** Chiếm [%] tổng số lỗi. Phổ biến nhất là tình trạng [ví dụ: thiếu Javadoc cho các phương thức quan trọng hoặc comment không cập nhật theo code].
- **Nhóm lỗi [Mã số - Tên nhóm, VD: I.7 Control Flow]:** Chiếm [%] tổng số lỗi. Các vấn đề tiêu biểu gồm [ví dụ: thiếu câu lệnh default trong cấu trúc switch-case hoặc vòng lặp tiềm ẩn nguy cơ vô hạn].

## 5 Một số lỗi khác không nằm trong Checklist

Trong phần này, nhóm trích dẫn một số lỗi tiêu biểu phát hiện được trong quá trình review thư mục `com/` mà không nằm trong *Java Code Review Checklist*. Mỗi lỗi sẽ bao gồm mô tả, vị trí, mã nguồn hiện tại và đề xuất sửa đổi cụ thể.

### 5.1 Lỗi []

- **Tệp tin:** `[Tên_file].java`

- **Vị trí:** Dòng [Số dòng]
- **Mô tả lỗi:** [Mô tả chi tiết lỗi, ví dụ: Thiếu câu lệnh default trong switch-case dẫn đến rủi ro không xử lý được các giá trị ngoại lệ.]

#### Mã nguồn hiện tại

```
1 switch (status) {  
2     case 1: ... break;  
3     case 2: ... break;  
4 }
```

#### Đề xuất sửa đổi

```
1 switch (status) {  
2     case 1: ... break;  
3     case 2: ... break;  
4     default:  
5         throw new IllegalStateException();  
6 }
```

## 5.2 Lỗi []

- **Tệp tin:** [Tên\_file].java
- **Mô tả lỗi:** Sử dụng cộng chuỗi trực tiếp trong truy vấn SQL, gây nguy cơ bảo mật nghiêm trọng.

#### Mã nguồn hiện tại

```
1 String sql = "SELECT * FROM users  
2             WHERE id = " + id;
```

#### Đề xuất sửa đổi

```
1 String sql = "SELECT * FROM users  
2             WHERE id = ?";  
3 PreparedStatement pstmt =  
4     conn.prepareStatement(sql);  
5 pstmt.setInt(1, id);
```

## 6 Kết luận

Quá trình thực hiện đánh giá kỹ thuật cho dự án **OHS** đã hoàn thành đúng mục tiêu và tiến độ đề ra. Thông qua việc đối chiếu nghiêm ngặt với bộ *Java Code Review Checklist*, nhóm đã đạt được các kết quả sau:



- **Đảm bảo chất lượng:** Xác định và phân loại thành công [Tổng số] lỗi từ mức độ trình bày đến các rủi ro bảo mật nghiêm trọng.
- **Tuân thủ quy trình:** Đảm bảo mỗi thành viên đều hoàn thành khối lượng công việc trên 300 LOC và thực hiện kiểm chứng chéo đầy đủ để loại bỏ sai sót chủ quan.
- **Giá trị thực tiễn:** Các đề xuất sửa đổi (Fixing Suggestions) đã cung cấp giải pháp cụ thể để nâng cao tính bảo trì, hiệu suất và độ bảo mật của hệ thống.

Nhìn chung, dù mã nguồn vẫn còn tồn tại một số hạn chế về [ví dụ: Layout hoặc Comment], nhưng các vấn đề cốt lõi về logic và bảo mật đã được nhận diện rõ ràng. Việc thực hiện nghiêm túc các đề xuất sửa chữa trong tệp tin đính kèm sẽ giúp mã nguồn của dự án **OHS** đạt chuẩn chuyên nghiệp và sẵn sàng cho các giai đoạn phát triển tiếp theo.