

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
Nhóm thực hiện: A01 – 03
Sinh viên thực hiện: Bùi Trọng Văn – 2213915
Phạm Ngọc Long – 2211894
Nguyễn Trần Yến Nhi – 2312506
Bùi Trọng Hiên – 2310993
Dặng Ngọc Bảo Trâm – 2213568

Thành phố Hồ Chí Minh, Tháng 2 - 2026



Mục lục

1	Giới thiệu	2
2	Phân công	2
3	Quy trình Review	3
4	Tóm tắt kết quả	4
5	Một số lỗi khác không nằm trong Checklist	4
5.1	Lỗi syntax trong khối catch	4
5.2	Lỗi type mismatch trong phương thức setStatusId	5
5.3	Lỗi so sánh chuỗi bằng toán tử !=	6
5.4	Lỗi không đóng tài nguyên đúng cách	6
5.5	Lỗi mã hóa cứng thông tin nhạy cảm	7
6	Kết luận	7



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	Reviewed files:	1448	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	
	Validated files: files reviewed by Phạm Ngọc Long		
Phạm Ngọc Long 2211894	Reviewed files:	1452	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	
	Validated files: files reviewed by Bùi Trọng Văn		
Nguyễn Trần Yến Nhi 2312506	Reviewed files:	1419	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	
	Validated files: files reviewed by Bùi Trọng Hiển		

Thành viên	Files	LOC	% đóng góp
Bùi Trọng Hiển 2310993	Reviewed files:	1516	100%
	com/ohs/models/Prescription.java	426	
	com/ohs/services/PrescriptionService.java	250	
	com/ohs/models/Admin.java	840	
	Validated files: files reviewed by Đặng Ngọc Bảo Trâm		
Đặng Ngọc Bảo Trâm 2213568	Reviewed files:	1395	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	
	Validated files: 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 273 lỗi, được tóm tắt qua bảng dưới đây:

Nhóm mục tiêu (Objective)	Số lượng lỗi
I. Defect Objective	219
II. Ambiguity Objective	21
III. Redundance Objective	33
Tổng cộng	273

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

- **Nhóm lỗi I.9 - Comment:** Chiếm 74 lỗi. Các vi phạm chủ yếu tập trung vào thiếu Javadoc cho các phương thức, lớp quan trọng hoặc comment không được viết đầy đủ.
- **Nhóm lỗi I.1 - Variable and Constant Declaration:** Chiếm 57 lỗi. Các lỗi chủ yếu là tên biến không tuân theo convention, kiểu dữ liệu của biến không phù hợp, hoặc biến được khai báo nhưng không được khởi tạo đúng cách.
- **Nhóm lỗi I.7 Control Flow:** Chiếm 23 lỗi. Các vấn đề tiêu biểu là các ngoại lệ không được xử lý đúng cách, hoặc các khối điều khiển trống (null body control structure) không được chú thích rõ ràng.

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 syntax trong khối catch

- **Tệp tin:** `/ohs/services/DoctorService.java`

- **Vị trí:** Dòng 86 - 90
- **Mô tả lỗi:** Lỗi cú pháp, dư dấu ngoặc nhọn } trong khối catch.

Mã nguồn hiện tại

```
1 try {  
2     // some code  
3 } catch (SQLException ex) { } // extra closing brace  
4 } finally {  
5     close(rs, preparedStatement, con);  
6 }
```

Đề xuất sửa đổi

```
1 try {  
2     // some code  
3 } catch (SQLException ex) {  
4     // handle exception  
5 } finally {  
6     close(rs, preparedStatement, con);  
7 }
```

5.2 Lỗi type mismatch trong phương thức setStatusId

- **Tệp tin:** /ohs/models/Status.java
- **Vị trí:** Dòng 24 - 25
- **Mô tả lỗi:** Lỗi biên dịch, sai kiểu dữ liệu trong phép gán một String cho một biến kiểu int và ngược lại.

Mã nguồn hiện tại

```
1 public void setStatusId(String statusId) {  
2     this._status = statusId;  
3     this.statusId = this._status;  
4 }
```

Đề xuất sửa đổi

```
1 public void setStatusId(String statusId) {  
2     this.statusId = statusId;  
3 }
```

5.3 Lỗi so sánh chuỗi bằng toán tử !=

- **Tệp tin:** /ohs/services/DoctorService.java
- **Vị trí:** Dòng 240 - 241
- **Mô tả lỗi:** Lỗi sai ngữ nghĩa, sử dụng toán tử != để so sánh chuỗi, dẫn đến kết quả không chính xác. Nên sử dụng phương thức .equals() để so sánh nội dung chuỗi.

Mã nguồn hiện tại

```
1 if (rs.getString("symptoms") != "") // wrong syntax
2     appointment.setSymptoms(rs.getString("symptoms"));
```

Đề xuất sửa đổi

```
1 if (!rs.getString("symptoms").equals("")) // correct syntax
2     appointment.setSymptoms(rs.getString("symptoms"));
```

5.4 Lỗi không đóng tài nguyên đúng cách

- **Tệp tin:** AdminService.java
- **Vị trí:** Dòng 498
- **Mô tả lỗi:** Tài nguyên PreparedStatement và Connection không được đóng đúng cách trong khối finally, có thể gây rò rỉ tài nguyên (resource leak).

Mã nguồn hiện tại

```
1 finally {
2     close(rs, preparedStatement, con);
3 }
```

Đề xuất sửa đổi

Đảm bảo kiểm tra null trước khi đóng tài nguyên.

```
1 finally {
2     if (rs != null) rs.close();
3     if (preparedStatement != null) preparedStatement.close();
4     if (con != null) con.close();
5 }
```

5.5 Lỗi mã hóa cứng thông tin nhạy cảm

- **Tệp tin:** MailService.java
- **Vị trí:** Dòng 36 – 37
- **Mô tả lỗi:** Thông tin nhạy cảm như tên người dùng và mật khẩu được mã hóa cứng trong mã nguồn, gây rủi ro bảo mật nghiêm trọng nếu mã nguồn bị lộ.

Mã nguồn hiện tại

```
1 final String userName = jdbcUtility.getPropertyValue("email");  
2 final String appPassword =  
    jdbcUtility.getPropertyValue("appPassword");
```

Đề xuất sửa đổi

```
1 final String userName = System.getenv("EMAIL_USERNAME");  
2 final String appPassword = System.getenv("EMAIL_APP_PASSWORD");
```

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 273 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 comment hay khai báo biến, 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.