

# I. CHIẾN LƯỢC KIỂM THỬ PHẦN MỀM BOOKING HOTEL (BOOKING HOTEL TEST STRATEGY)

## 1. Định danh chiến lược kiểm thử

Mã định danh duy nhất cho chiến lược kiểm thử này là: **BK-TS-001**

## 2. Giới thiệu

- Booking Hotel là một ứng dụng đặt phòng online giúp người dùng dễ dàng đặt phòng từ xa mà không cần trực tiếp đến nơi, tối ưu thời gian, linh hoạt. Đồng thời giúp người quản lý các khách sạn phòng hiệu quả gồm:

- **Quản lý người dùng**
- **Quản lý doanh thu**
- **Quản lý khách sạn**
- **Quản lý phòng**
- **Quản lý đơn đặt**
- **Quản lý đánh giá**

### Mục đích của tài liệu chiến lược kiểm thử

Tài liệu này xác định các mục tiêu kiểm thử cho dự án Booking Hotel gồm các hoạt động kiểm thử, vai trò, trách nhiệm, quy trình, và thực hành kiểm thử. Mục tiêu là đảm bảo các chức năng hoạt động đúng, an toàn, và hiệu quả theo yêu cầu SRS.

## 3. Mục kiểm thử

### Trong phạm vi:

- Đăng nhập và xác thực người dùng (admin, customer).
- Quản lý tài khoản (thêm, sửa, cập nhật tài khoản).
- Đặt phòng khách sạn và quản lý đơn đặt.
- Đánh giá và quản lý đánh giá.
- Quản lý khách sạn.
- Quản lý phòng.
- Quản lý doanh thu.
- Kiểm thử đơn vị (Unit Test).
- Kiểm thử tích hợp (Integration Test).
- Kiểm thử E2E.
- Kiểm thử hiệu suất cơ bản.
- Kiểm thử bảo mật cơ bản (phân quyền theo vai trò).

### Ngoài phạm vi:

- Kiểm thử giao diện người dùng (UI).
- Kiểm thử đơn vị (unit testing) của logic bên trong controller.
- Kiểm thử hiệu suất nâng cao (load testing với số lượng lớn người dùng).

#### **4. Tính năng cần kiểm thử**

- **Đăng nhập:**

- Xác thực người dùng với tên đăng nhập/mã số và mật khẩu.
- Phân luồng giao diện dựa trên vai trò (admin, customer).

- **Quản lý tài khoản:**

- Thêm, sửa, xóa tài khoản customer.
- Tạo ID tự động theo định dạng.

- **Đặt và quản lý đơn đặt:**

- Tạo, cập nhật đơn đặt theo trạng thái.
- Kiểm tra trùng phòng.

- **Đánh giá và quản lý đánh giá:**

- Kiểm tra quyền đánh giá.
- Xem các đánh giá của người dùng và hiển thị

- **Quản lý khách sạn:**

- Thêm, xóa, cập nhật theo tình trạng khách sạn.

- **Quản lý phòng:**

- Thêm, xóa, cập nhật theo tình trạng phòng.

- **Quản lý doanh thu:**

- Thêm, xóa, cập nhật doanh thu theo khách sạn, tháng
- Xuất file doanh thu Excel

- **Hiệu suất cơ bản:**

- Thời gian phản hồi (  $\leq 4s$  )

- **Bảo mật cơ bản:**

- Phân quyền theo vai trò (admin, giáo viên, học sinh).

#### **5. Tính năng không cần kiểm thử**

- Kiểm thử giao diện người dùng (UI) của ứng dụng di động.
- Kiểm thử hiệu suất dưới tải lớn (load testing).
- Kiểm thử bảo mật chuyên sâu (SQL injection, XSS, v.v.).
- Kiểm thử các tính năng chưa triển khai (Role cho lễ tân, phân tích người dùng, Chatbot tư vấn).

## 6. Cách tiếp cận

**Phương pháp kiểm thử:** Agile

**Mô hình kiểm thử:** Mô hình chữ V

**Công cụ kiểm thử:** Jest (Unit Test, Integration Test), Cypress (E2E), Github Action (CI/CD).

**Rủi ro:**

- Yêu cầu không rõ ràng dẫn đến bỏ sót test case.
- Dữ liệu MongoDB không nhất quán.
- Phân quyền không hoạt động đúng.

### 6.1. Tiêu chí thực hiện khâu phân tích và lập kế hoạch kiểm thử

- Danh sách các tính năng trong phạm vi được xác định và ưu tiên.
- Tài liệu SRS và mã nguồn API được kiểm soát trong quy trình thay đổi.

### 6.2. Tiêu chí kết thúc pha phân tích và lập kế hoạch kiểm thử

- Test case được viết và đánh giá ngang hàng.
- Tài liệu kiểm thử được phê duyệt.
- Ước lượng thời gian kiểm thử được thống nhất.
- Danh sách tính năng kiểm thử được ưu tiên.

### 6.3. Tiêu chí bắt đầu pha kiểm thử

- Môi trường kiểm thử (máy chủ ExpressJS, MongoDB) được thiết lập.
- Dữ liệu kiểm thử được nạp sẵn vào MongoDB.
- Các API đã được triển khai trên môi trường kiểm thử.
- Các công cụ kiểm thử (Postman, MongoDB Compass) sẵn sàng.

### 6.4. Tiêu chí kết thúc pha kiểm thử

- Báo cáo kiểm thử hoàn thành.
- Tất cả test case đã được thực thi với tỷ lệ pass đạt yêu cầu.
- Các lỗi nghiêm trọng (S1, S2) đã được sửa và kiểm tra lại.
- Không có lỗi ở trạng thái chưa giải quyết.

### 6.5. Quản lý sự thay đổi

- Quản lý build đảm bảo không có thay đổi mã nguồn trong quá trình kiểm thử.
- Mọi thay đổi (sửa lỗi, cập nhật API) phải được thông báo trước và ghi nhận trong hệ thống quản lý phiên bản (ví dụ: Git).
- Nhóm kiểm thử được thông báo về phiên bản API đang kiểm thử.

## 7. Tiêu chí đánh giá test case 'Pass/Fail'

Mỗi test case được đánh giá dựa trên:

- Mã trạng thái HTTP, nội dung phản hồi, và thời gian phản hồi khớp với yêu cầu SRS.
- Dữ liệu trong MongoDB nhất quán sau mỗi request.
- Không có lỗi nghiêm trọng (S1, S2) trong quá trình thực thi.

### Bảng mức độ nghiêm trọng của lỗi

Mức độ	Định nghĩa	Số lượng tối đa cho phép
S1	Hệ thống sập, mất dữ liệu, không thể khắc phục	0
S2	Lỗi chức năng chính, kết quả sai	High
S3	Lỗi nhỏ, không ảnh hưởng lớn	Medium
S4	Lỗi giao diện, thẩm mỹ	Low
S5	Không áp dụng (yêu cầu mới, nhiệm vụ phát triển)	Tham chiếu

Tổng số lỗi tối đa ở trạng thái mở/ chưa giải quyết được chấp nhận để phát hành sẽ do Quản lý dự án quyết định.

## 8. Các bản phân phối

- **Kế hoạch kiểm thử:** Quy định phạm vi, phương pháp, tài nguyên, và lịch trình kiểm thử.
- **Test case:** Chi tiết điều kiện tiên quyết, bước thực hiện, kết quả mong đợi/thực tế.
- **Báo cáo tiến độ định kỳ:** Cập nhật trạng thái kiểm thử.
- **Báo cáo lỗi:** Ghi nhận lỗi trong công cụ quản lý lỗi (ví dụ: Jira).
- **Báo cáo tóm tắt kiểm thử:** Tổng hợp kết quả kiểm thử.

## 9. Nhiệm vụ kiểm thử

- **Trong phạm vi đầy đủ:** Kiểm thử chức năng và kiểm thử tích hợp.
- **Trong phạm vi một phần:** Kiểm thử bảo mật cơ bản, kiểm thử hiệu suất cơ bản.
- **Ngoài phạm vi:** Kiểm thử giao diện người dùng, kiểm thử hiệu suất nâng cao, kiểm thử bảo mật chuyên sâu.

## 10. Nhu cầu môi trường và cơ sở hạ tầng

### Phần cứng

- **Môi trường kiểm thử:**

- o Test-A: <URL máy chủ ExpressJS local/cloud>

- o Test-B: <URL máy chủ dự phòng>

- **MongoDB:** Cơ sở dữ liệu được thiết lập trên máy chủ kiểm thử.

### Phần mềm

- **Postman:** Công cụ kiểm thử API.

- **MongoDB Compass:** Kiểm tra dữ liệu MongoDB.

- **Jira:** Quản lý lỗi.

- **Git:** Quản lý phiên bản mã nguồn.

### Cơ sở hạ tầng

- Kết nối mạng ổn định cho máy chủ kiểm thử và client Postman.

- Kho lưu trữ test case: .

### 11. Ma trận trách nhiệm

Hoạt động	Quản lý sản phẩm	Quản lý phát triển	Quản lý kiểm thử	Kỹ sư kiểm thử
Cung cấp tài liệu kỹ thuật	X	X		
Lập kế hoạch và ước lượng kiểm thử	X	X		
Đánh giá và phê duyệt kế hoạch kiểm thử	X	X	X	
Tài liệu kiểm thử	X	X		
Chuẩn bị và thực thi kiểm thử	X			
Thiết lập môi trường kiểm thử	X			
Kiểm soát thay đổi môi trường kiểm thử	X	X		
Cung cấp API đã kiểm thử đơn vị	X			
Sửa lỗi và trả lại để kiểm tra lại	X			
Kiểm soát thay đổi sản phẩm	X	X	X	
Báo cáo kiểm thử định kỳ	X	X		
Báo cáo tóm tắt kiểm thử	X			

### 12. Nhu cầu nhân sự và đào tạo

#### Nhân sự

- 1 Quản lý kiểm thử: 50% thời gian cho lập kế hoạch.
- 2 Kỹ sư kiểm thử: 100% thời gian cho thực thi kiểm thử.

#### Đào tạo

- Đào tạo sử dụng Postman và viết test script.
- Hướng dẫn kiểm tra dữ liệu MongoDB bằng MongoDB Compass.

### 13. Lịch trình và kế hoạch tài nguyên

#### Kế hoạch nhóm

- Nhóm kiểm thử duy trì kế hoạch công việc, ghi nhận phân công nhiệm vụ và thời gian thực hiện.
- Lịch trình được lưu tại: .

#### Lịch kiểm thử

Giai đoạn	Thời gian	Hoạt động
Lập kế hoạch	1 ngày	Xác định test case, thiết lập môi trường
Thiết kế test case	2 ngày	Tạo Postman Collection và test script
Thực thi kiểm thử	4 ngày	Chạy test case, kiểm tra dữ liệu MongoDB
Báo cáo	1 ngày	Tổng hợp kết quả, báo cáo lỗi

### 14. Rủi ro và phương án dự phòng

STT	Rủi ro	Chiến lược giảm thiểu	Tác động
1	Tri hoãn cung cấp API từ nhóm phát triển	Quản lý sản phẩm và phát triển thông báo sớm, điều chỉnh phạm vi	Cao
2	Lỗi nghiêm trọng cần thời gian sửa lâu	Quản lý phát triển ưu tiên sửa lỗi, đảm bảo thời gian kiểm tra lại	Cao
3	Nhóm kiểm thử thiếu thông tin từ nhóm phát triển	Đảm bảo các nhóm sẵn sàng liên lạc trong giai đoạn kiểm thử	Trung bình
4	Yêu cầu không rõ ràng dẫn đến bỏ sót test case	Đối chiếu SRS với mã nguồn, làm rõ với nhóm phát triển	Trung bình
5	Dữ liệu MongoDB không nhất quán	Kiểm tra dữ liệu trước/sau mỗi test case, sử dụng MongoDB Compass	Thấp

### 15. Phê duyệt

Những người sau cần phê duyệt chiến lược kiểm thử này:

Phê duyệt bởi	Phê duyệt
Quản lý kiểm thử	
Trưởng phòng kiểm thử	
Quản lý sản phẩm	
Quản lý phát triển	
Quản lý dự án	



## II. KẾ HOẠCH KIỂM THỬ PHẦN MỀM BOOKING HOTEL (BOOKING HOTEL TEST PLANNING)

### 1. Mục tiêu kiểm thử

- Đảm bảo tất cả các chức năng của ứng dụng hoạt động đúng theo yêu cầu đặc tả phần mềm (SRS) đã nêu ra
- Xác minh tính đúng đắn của dữ liệu đầu vào/ đầu ra và tính toàn vẹn dữ liệu trong MongoDB
- Đảm bảo thời gian phản hồi đáp ứng yêu cầu phi chức năng
- Kiểm tra quyền dựa theo vai trò
- Phát hiện các lỗi trong xử lý dữ liệu không hợp lệ và các trường hợp cạnh.

### 2. Phạm vi kiểm thử

- Trong phạm vi:
  - Kiểm thử đơn vị Unit Test
  - Kiểm thử tích hợp Integration Test
  - Kiểm thử E2E
  - Tích hợp CI/CD tự động
- Ngoài phạm vi:
  - Kiểm tra tất cả các endpoint trong các file controller được cung cấp
  - Kiểm thử giao diện người dùng
  - Kiểm tra hiệu suất cơ bản
  - Kiểm thử các chức năng chính của ứng dụng

### 3. Các sản phẩm kiểm thử

- Bộ sưu tập Jest, chứa các test case unit test và integration test
- Bộ sưu tập Cypress kiểm thử giao diện và E2E
- Bộ github workflow hỗ trợ CI/CD

### 4. Lịch trình kiểm thử

## 5. Các test case

Dưới đây là bộ test case cho các chức năng chính của ứng dụng bao gồm request, dữ liệu đầu vào và kết quả mong đợi.

TC ID	Mô tả	Kết quả mong đợi
TC-01	Đăng nhập thành công với vai trò Customer	Hệ thống trả về 200 OK, token hợp lệ, role = Customer
TC-02	Đăng nhập thành công với vai trò Admin	Hệ thống trả về 200 OK, token hợp lệ, role = Admin
TC-03	Thiếu trường dữ liệu khi đăng nhập	Hệ thống trả về 400 Bad Request, thông báo lỗi trường bị thiếu
TC-04	Đăng nhập với tài khoản sai dữ liệu	Hệ thống trả về 404 Not Found, thông báo lỗi hợp lệ
TC-05	Đăng nhập với mật khẩu sai dữ liệu	Hệ thống trả về 401 Unauthorized, thông báo lỗi hợp lệ
TC-06	Tạo khách sạn thành công	Hệ thống trả về 201 Created, dữ liệu khách sạn lưu vào DB
TC-07	Tạo khách sạn thiếu trường dữ liệu	Hệ thống trả về 400 Bad Request, thông báo lỗi hợp lệ
TC-08	Lấy tất cả khách sạn	Hệ thống trả về 200 OK, danh sách khách sạn
TC-09	Lấy khách sạn theo ID	Hệ thống trả về 200 OK, đúng dữ liệu khách sạn theo ID
TC-10	Lấy khách sạn không tồn tại	Hệ thống trả về 404 Not Found
TC-11	Chỉnh sửa khách sạn thành công	Hệ thống trả về 200 OK, dữ liệu khách sạn được cập nhật
TC-12	Chỉnh sửa khách sạn không tồn tại	Hệ thống trả về 404 Not Found
TC-13	Xoá khách sạn tồn tại	Hệ thống trả về 200 OK, khách sạn bị xoá khỏi DB
TC-14	Xoá khách sạn không tồn tại	Hệ thống trả về 404 Not Found
TC-15	Thêm phòng thành công	Hệ thống trả về 201 Created, dữ liệu phòng lưu vào DB
TC-16	Thêm phòng thiếu trường	Hệ thống trả về 400 Bad Request, thông báo lỗi
TC-17	Thêm phòng trong khách sạn không tồn tại	Hệ thống trả về 404 Not Found
TC-18	Lấy danh sách tất cả các phòng	Hệ thống trả về 200 OK, danh sách phòng

TC-19	Lấy danh sách phòng theo Khách sạn	Hệ thống trả về 200 OK, danh sách phòng thuộc khách sạn đó
TC-20	Lấy phòng theo ID	Hệ thống trả về 200 OK, thông tin đúng phòng
TC-21	Xoá phòng tồn tại	Hệ thống trả về 200 OK, phòng bị xoá khỏi DB
TC-22	Xoá phòng không tồn tại	Hệ thống trả về 404 Not Found
TC-23	Chỉnh sửa thông tin phòng	Hệ thống trả về 200 OK, dữ liệu phòng được cập nhật
TC-24	Lấy danh sách đơn đặt	Hệ thống trả về 200 OK, danh sách đơn đặt
TC-25	Đặt phòng thành công	Hệ thống trả về 201 Created, đơn đặt lưu DB
TC-26	Đặt phòng đã Booking	Hệ thống trả về 409 Conflict, thông báo phòng đã đặt
TC-27	Lấy danh sách đơn theo người dùng	Hệ thống trả về 200 OK, danh sách đơn thuộc user
TC-28	Chỉnh sửa thông tin thanh toán	Hệ thống trả về 200 OK, thông tin thanh toán được cập nhật
TC-29	Lấy đơn đặt theo ID	Hệ thống trả về 200 OK, thông tin đơn đúng
TC-30	Huỷ đơn đặt phòng thành công	Hệ thống trả về 200 OK, trạng thái đơn = Cancelled
TC-31	Lấy danh sách tất cả người dùng	Hệ thống trả về 200 OK, danh sách users
TC-32	Sửa mật khẩu (hợp lệ)	Hệ thống trả về 200 OK, mật khẩu được thay đổi
TC-33	Sửa mật khẩu (không hợp lệ)	Hệ thống trả về 400 Bad Request, thông báo lỗi
TC-34	Chỉnh sửa thông tin người dùng tồn tại	Hệ thống trả về 200 OK, dữ liệu được cập nhật
TC-35	Chỉnh sửa thông tin người dùng không tồn tại	Hệ thống trả về 404 Not Found
TC-36	Lấy người dùng theo ID	Hệ thống trả về 200 OK, dữ liệu đúng
TC-37	Thêm đánh giá thành công	Hệ thống trả về 201 Created, đánh giá lưu DB
TC-38	Thêm đánh giá không đủ quyền	Hệ thống trả về 403 Forbidden
TC-39	Lấy danh sách đánh giá	Hệ thống trả về 200 OK, danh sách đánh giá
TC-40	Chỉnh sửa đánh giá thành công	Hệ thống trả về 200 OK, dữ liệu đánh giá được cập nhật

TC-41	Chỉnh sửa đánh giá không đủ quyền	Hệ thống trả về 403 Forbidden
TC-42	Xoá đánh giá thành công	Hệ thống trả về 200 OK, đánh giá bị xoá
TC-43	Xoá đánh giá không có quyền	Hệ thống trả về 403 Forbidden
TC-44	Lấy đánh giá theo phòng	Hệ thống trả về 200 OK, danh sách đánh giá của phòng
TC-45	Lấy doanh thu theo khách sạn	Hệ thống trả về 200 OK, dữ liệu doanh thu đúng
TC-46	Lấy doanh thu theo tháng	Hệ thống trả về 200 OK, dữ liệu doanh thu đúng
TC-47	Lấy danh sách doanh thu	Hệ thống trả về 200 OK, danh sách doanh thu
TC-48	Lấy danh sách tất cả đơn đặt	Hệ thống trả về 200 OK, danh sách đơn đặt
TC-49	Đặt phòng thành công (trùng TC-25)	Hệ thống trả về 201 Created
TC-50	Đặt phòng không thành công	Hệ thống trả về 400 Bad Request hoặc 409 Conflict
TC-51	Cập nhật thanh toán thành công	Hệ thống trả về 200 OK, dữ liệu thanh toán cập nhật
TC-52	Cập nhật thanh toán không thành công	Hệ thống trả về 400 Bad Request
TC-53	Lấy danh sách đơn đặt theo ID	Hệ thống trả về 200 OK, dữ liệu đúng
TC-54	Hủy đơn đặt	Hệ thống trả về 200 OK, trạng thái đơn = Cancelled

## 6. Thực thi kiểm thử

- Thiết lập:
  - Mock dữ liệu kiểm thử
  - Viết mã nguồn kiểm thử cho từng test case
  - Cài đặt Jest, Cypress
- Các bước thực thi:
  - Viết mã nguồn thực thi kiểm thử
  - Chạy ứng dụng
  - Kiểm tra dữ liệu pass/ fail

- Tiêu chí pass/ fail:
  - Pass: Mã trạng thái, nội dung phản hồi, thời gian phản hồi khớp với yêu cầu
  - Fail: Phản hồi sai, sai mã trạng thái, thời gian vượt quá yêu cầu, hoặc dữ liệu không nhất quán
- Báo cáo lỗi: Ghi lại lỗi trong báo cáo

## **7. Môi trường kiểm thử**

- **Máy chủ:** ExpressJS hoặc cloud với MongoDB
- **Dữ liệu:** Dữ liệu được tích hợp trực tiếp với mã nguồn kiểm thử
- **Mã nguồn:** Xây dựng mã nguồn testing

## **8. Kết quả mong đợi**

- Tất cả testcase pass với thời gian phản hồi đúng yêu cầu
- Thời gian phản hồi trung bình đáp ứng yêu cầu phi chức năng
- Không có lỗi bảo mật
- Dữ liệu MongoDB nhất quán