Đại học Quốc Gia – Đại học Bách Khoa

**Khoa khoa học và kĩ thuật máy tính**

**🙞∙∙∙☼∙∙∙🙜**

****

báo cáo Bài tập Lớn 2

Kiểm Tra phần mềm

Nhóm L02 – 03

Giảng viên hướng dẫn: PGS.TS Bùi Hoài Thắng

|  |  |  |
| --- | --- | --- |
| **Sinh viên thực hiện** | **Mã số sinh viên** | **Điểm số** |
| La Quốc Anh | 1710465 |  |
| Hà Hải Thiên Sơn | 1811193 |  |
| Nguyễn Tiến Đăng Khoa | 1832026 |  |
| Trịnh Duy Hưng | 1913652 |  |

Mục lục

1. Lời cảm ơn 3
2. Giới thiệu 4
3. Phân công nhiệm vụ 4
4. Đặc tả ứng dụng

1. Functional requirements

2. Non-functional requirements

1. Kiểm tra các yêu cầu chức năng

1.

2.

3.

4.

5.

1. Kiểm tra các yêu cầu phi chức năng

1. 5

2. 7

3. 8

4. 9

5. 10

1. Tổng kết 11
2. **Lời cảm ơn**

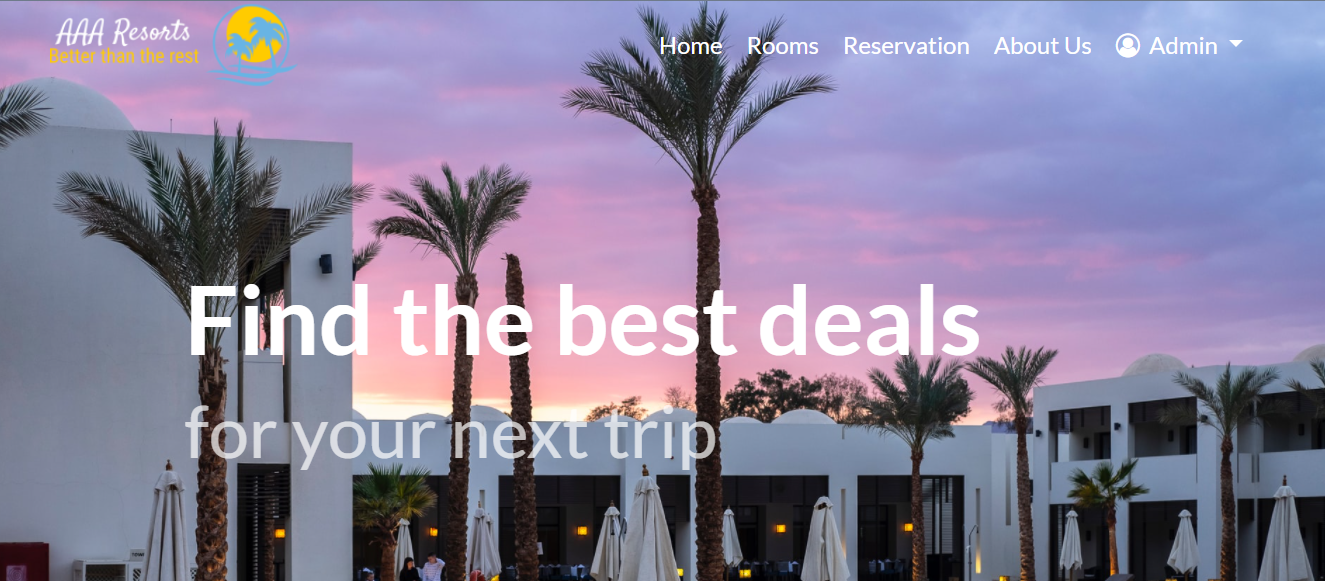
Thân chào thầy và các bạn sinh viên, đây là bài tập lớn môn Kiểm tra Phần mềm với sự hướng dẫn của thầy Phó giáo sư tiến sĩ Bùi Hoài Thắng.

Chúng em sẽ cố gắng cung cấp các ý về các bài toán của đề tài một cách dễ hiểu nhất. Thay mặt lớp, nhóm xin chân thành cảm ơn thầy đã chỉ dạy và hướng dẫn nhiệt tình trong học kì vừa qua.

1. **Giới thiệu**

Trong bài tập lớn này, nhóm sử dụng các phương pháp Black Box Testing để kiểm tra ứng dụng quản lí khách sạn được cung cấp.

Link project: <https://github.com/alandsilva26/hotel-management-php>



1. **Phân công nhiệm vụ**

Part A : Trịnh Duy Hưng – 1913652 – 100%

1. **Đặc tả ứng dụng**

Phần này sẽ bao gồm một số yêu cầu chức năng và yêu cầu phi chức năng của hệ thống mà nhóm tìm kiếm, phân tích và quyết định sẽ kiểm thử với black-box testing.

1. **Functional requirements**

Dưới đây là một số functional requirements mà nhóm đã tìm hiểu trên ứng dụng:

* Là một người sử dụng, tôi muốn đặt phòng..
* Là một người sử dụng, tôi muốn nhận hóa đơn khi thanh toán.
* Là một người sử dụng, tôi muốn đăng ký tài khoản.
* Là một người quản lí, tôi muốn xóa phòng đã được đặt.
* Là một người quản lí, tôi muốn chỉnh sửa thông tin của phòng.

1. **Non-funtional requirements**

Dưới đây là một số non-functional requirements mà nhóm đã tìm kiếm được:

* Thời gian đáp ứng yêu cầu của hệ thống chậm nhất là 5s.
* Người dùng chỉ có thể đặt 2 phòng nếu không đăng nhập.
* Tài khoản sẽ bị khóa nếu nhập sai thông tin đăng nhập quá 5 lần.
* Khi người dùng thực hiện thanh toán online, hệ thống không được phép lưu trữ thông tin thẻ credit/ debit của người dùng.
* Mỗi ảnh đính kèm cập nhật ảnh phòng có dung lượng tối đa 2MB.

1. **Kiểm tra yêu cầu chức năng**

Các kĩ thuật Black-box testing để sinh testcase cho các yêu cầu chức năng của nhóm là:

* Boundary Value Testing
* Equivalence Class Partitioning
* Decision Table
* Use Cases Testing.

1. **Chức năng đặt phòng**
   1. **Boundary Value Testing**

Chức năng đặt phòng cho phép người dùng đặt và sử dụng dịch vụ của khách sạn.

* 1. **Equivalence Class Partitioning**
  2. **Decision Table**
  3. **Use Cases Testing**

1. **Chức năng thanh toán**
2. **Chức năng đăng ký tài khoản**
3. **Chức năng xóa phòng**
4. **Chức năng chỉnh sửa thông tin của phòng**
5. **Kiểm tra yêu cầu phi chức năng**
6. **Tổng kết**

Trong bài báo cáo lần này, nhóm đã thực hiện kiểm tra lỗi trên 5 file code mẫu. Từng đoạn code có những đặc thù riêng nhưng nếu thiết kế một checklist review code một cách hợp lý thì chúng ta có thể kiểm tra lỗi một cách dễ dàng hơn, hạn chế trong việc sửa lỗi và tránh được những lỗi không đáng có. Nhóm đã sử dụng checklist code review của TMA Solution nhưng đây chỉ là bảng rút gọn, vì vậy có thể còn nhiều sai sót mà nhóm chưa tìm ra. Tuy nhiên có thể hiểu được phần nào về công việc kiểm tra lỗi code.