

**ĐẠI HỌC PHENIKAA**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN PHENIKAA**  
\*\*\*\*\*



**Báo cáo bài tập lớn Kỹ thuật phần mềm**  
**Đề bài: Quản Lý Quán Internet**

Giảng viên hướng dẫn : Nguyễn Thị Vân  
Lớp học phần : Kỹ thuật phần mềm (COUR01.LT9)  
Sinh viên thực hiện : Nhóm 6

- Đồng Thị Ánh - 23013883
- Đinh Nhật Tân - 23013018
- Bùi Thị Hồng Tươi - 23015124
- Nguyễn Minh Phương - 23015738

Hà Nội, Ngày 20 tháng 6 năm 2025

# MỞ ĐẦU

Ngày nay cùng với sự phát triển nhanh chóng của công nghệ thông tin trên toàn cầu. Chiếc máy tính đã trở thành thân thuộc đối với mỗi con người. Cùng với đó là nhu cầu về mạng Internet là rất cần thiết và không thể tách rời. Điều này khiến cho tốc độ và mật độ các quán Internet tăng lên nhanh chóng. Nhu cầu vào mạng Internet của con người ngày càng cao. Vì thế các quán Internet khi được mở ra đòi hỏi phải có giải pháp hợp lý để quản lý quán Internet một cách tốt nhất. Giải pháp tối ưu nhất là dùng các phần mềm về quản lý quán Internet. Trên cơ sở lý thuyết được học trên lớp và thông tin tìm hiểu về các phần mềm quản lý nói chung và quản lý quán Internet nói riêng. Nhóm chúng em đưa ra bản : “Quản lý quán Internet”.

Để làm ra một sản phẩm phần mềm tốt thì đòi hỏi các bên tham gia dự án phần mềm thực hiện rất nhiều công việc, trong đó việc phân tích thiết kế và xây dựng phần mềm là công việc rất quan trọng mà các nhà phát triển phải thực hiện. Đây là công việc rất quan trọng và khó khăn, ảnh hưởng rất lớn đến việc dự án có được thực hiện tốt hay không. Có phân tích và thiết kế đúng mới giúp cho việc xây dựng phần mềm được tối ưu.

Ngược lại, nếu làm không tốt có thể dẫn đến hậu quả không tốt, nhẹ thì sản phẩm không tối ưu, hoặc dẫn đến thiếu hụt kinh phí, thua lỗ hoặc nghiêm trọng hơn là không thực hiện được dự án. Do đó các nhà phát triển cần thực hiện tốt công việc này. Chúng em thực hiện đề tài “ Quản lý quán Internet ” để áp dụng những kiến thức về phân tích thiết kế phần mềm vào một hệ thống thực tế để hiểu rõ hơn công việc này.

Chúng em xin chân thành cảm ơn sự giúp đỡ của cô đã cung cấp những kiến thức cần thiết để hoàn thành đề tài. Trong quá trình thực hiện đề tài không tránh khỏi những sai sót, chúng em rất mong nhận được sự góp ý và đánh giá của cô.

Sau đây, chúng em xin trình bày về nội dung của bản Báo cáo.

# *Mục lục*

|  |    |
|--|----|
| <b>1. Giới thiệu đề tài</b>                      | 4  |
| <b>2. Phân tích yêu cầu</b>                      | 5  |
| 2.1. Yêu cầu chức năng                           | 5  |
| 2.2. Yêu cầu phi chức năng                       | 5  |
| <b>3. Mô hình Use Case</b>                       | 5  |
| 3.1. Vẽ mô hình Use Case bằng UML                | 6  |
| 3.2. Yêu cầu phi chức năng                       | 6  |
| <b>4. Thiết kế hệ thống</b>                      | 8  |
| 4.1. Kiến trúc hệ thống                          | 8  |
| 4.2. Sơ đồ lớp (Class Diagram)                   | 10 |
| 4.3. Sơ đồ trình tự (Sequence Diagram)           | 11 |
| <b>5. Cơ sở dữ liệu</b>                          | 12 |
| <b>6. Giao diện người dùng (UI Design)</b>       | 13 |
| 6.1. Quản lý khách hàng                          | 13 |
| 6.2. Quản lý phòng                               | 14 |
| 6.3. Quản lý máy                                 | 16 |
| 6.4. Quản lý dịch vụ                             | 18 |
| 6.5. Quản lý thuê máy                            | 19 |
| 6.6. Order                                       | 22 |
| <b>7. Mẫu thiết kế áp dụng (Design Patterns)</b> | 23 |
| 7.1. Model-View-Controller (MVC)                 | 23 |
| 7.2. Data Access Object (DAO)                    | 25 |
| 7.3. Data Transfer Object (DTO)                  | 26 |
| <b>8. Chiến lược kiểm thử</b>                    | 28 |
| 8.1. Chiến lược kiểm thử                         | 28 |
| 8.2. Kịch bản kiểm thử (Test cases)              | 32 |
| <b>9. Triển khai hệ thống</b>                    | 36 |
| 9.1. Hướng dẫn cài đặt và chạy chương trình      | 36 |
| 9.2. Kịch bản kiểm thử (Test cases)              | 36 |
| <b>10. Kết luận và hướng phát triển</b>          | 36 |
| 10.1. Đánh giá kết quả đạt được                  | 36 |
| 10.2. Khó khăn và cách khắc phục                 | 37 |
| 10.3. Hướng dẫn phát triển                       | 37 |
| <b>TỔNG KẾT</b>                                  | 37 |
| <b>11. Tài liệu tham khảo</b>                    | 37 |

# 1. Giới thiệu đề tài

+ **Tên đề tài:** Quản lý quán Internet.

+ **Lý do chọn đề tài:** Trong thời đại công nghệ số, việc quản lý thủ công tại các quán Internet đã trở nên lạc hậu, gây khó khăn trong việc kiểm soát thời gian sử dụng máy, dịch vụ đi kèm và lịch sử giao dịch. Việc xây dựng một hệ thống phần mềm chuyên biệt giúp tự động hóa các nghiệp vụ như đặt máy, thanh toán, quản lý điểm tích lũy, báo cáo doanh thu,... sẽ mang lại hiệu quả cao trong quản lý, giảm sai sót và nâng cao trải nghiệm người dùng. Java Swing được lựa chọn vì là một công cụ phổ biến, mạnh mẽ trong phát triển ứng dụng desktop, phù hợp cho bài toán này.

## Mục tiêu và phạm vi:

+ **Mục tiêu:** Xây dựng phần mềm desktop giúp quản lý toàn diện quán Internet bao gồm khách hàng, máy trạm, dịch vụ, thanh toán và báo cáo.

+ **Phạm vi:**

- Giao diện người dùng phát triển bằng Java Swing.
- Áp dụng mô hình MVC cho phân tách giao diện và logic.
- Lưu trữ dữ liệu sử dụng hệ quản trị cơ sở dữ liệu quan hệ (MySQL).
- Triển khai các tính năng bảo mật ở mức cơ bản như mã hóa thông tin người dùng.

| Thành phần           | Công nghệ   |
|----------------------|---|
| Ngôn ngữ lập trình   | Java  |
| Giao diện người dùng | Java swing  |
| Framework hỗ trợ     | JDBC, MVC Pattern                                     |
| Cơ sở dữ liệu        | MySQL   |
| Công cụ hỗ trợ       | NetBeans / IntelliJ IDEA, MySQL Workbench, Github,... |

## 2. Phân tích yêu cầu

### 2.1 Yêu cầu chức năng:

| Nhóm chức năng          | Mô tả   |
|-------------------------|---|
| Quản lý người dùng      | Đăng ký, đăng nhập, thay đổi thông tin, phân                |
| Quản lý máy             | Kiểm tra máy trống, bật/ tắt máy, theo dõi giờ sử dụng      |
| Quản lý dịch vụ         | Quản lý đồ ăn, nước uống, giờ chơi                          |
| Tích điểm và thanh toán | Tính tiền theo giờ sử dụng, áp dụng tích điểm và thanh toán |
| Xử lý sự cố             | Chuyển máy, ghi nhận lỗi, thông báo sự cố                   |
| Tìm kiếm và lọc         | Tìm kiếm khách hàng, dịch vụ, hóa đơn, máy theo trạng thái  |

### 2.2 Yêu cầu phi chức năng:

| Yêu cầu          | Mô tả   |
|------------------|---|
| Hiệu suất        | Hệ thống phản hồi nhanh, hoạt động ổn định                        |
| Bảo mật          | Mã hóa mật khẩu, kiểm soát quyền truy cập                         |
| Dễ sử dụng       | Giao diện trực quan, phù hợp với mọi người dùng không chuyên      |
| Khả năng mở rộng | Dễ nâng cấp thêm tính năng như đặt máy online, liên hệ mobile app |
| Tính đa nền tảng | Chạy được hết trên các hệ điều hành Windows/linux có Java Runtime |
| Dễ bảo trì       | Cấu trúc mã theo mô hình MVC, dễ sửa lỗi và nâng cấp tính năng    |

## 3. Mô hình Use Case

### 3.1 Vẽ mô hình Use Case bằng UML



### 3.2 Mô tả từng Use Case chính

| Tên                | Actor         | Mô tả   | Dòng sự kiện   |
|--------------------|---------------|---|--|
| Quản lý khách hàng | Người quản lý | Chức năng thể hiện người quản lý quản lý các thông tin của người dùng như: tên, giờ chơi, dịch vụ,... | 1. Người quản lý truy cập vào mục “Quản lý khách hàng”.<br>2. Hệ thống hiển thị danh sách tất cả các thông tin của khách hàng hiện tại.<br>3. Người quản lý có thể chọn các thao tác: <ul style="list-style-type: none"> <li>- Thêm khách hàng mới.</li> <li>- Sửa thông tin khách hàng.</li> <li>- Xóa khách hàng.</li> </ul> |

|                 |               |  |   |
|-----------------|---------------|--|---|
|                 |               |  | <p>4. Hệ thống lưu thay đổi.</p> <p>5. Thông báo (không) thành công .</p>   |
| Quản lý phòng   | Người quản lý | Chức năng thể hiện giá tiền phòng qua đó thấy được chất lượng loại phòng đó                  | <p>1. Người quản lý truy cập vào mục “ Quản lý phòng”.</p> <p>2. Hệ thống hiển thị danh sách tất cả các phòng hiện tại.</p> <p>3. Người quản lý có thể chọn các thao tác:</p> <ul style="list-style-type: none"> <li>- Thêm phòng mới.</li> <li>- Chỉnh sửa thông tin phòng.</li> <li>- Xóa phòng.</li> </ul> <p>4. Thay đổi được cập nhật và hiển thị lên danh sách phòng.</p> |
| Quản lý dịch vụ | Người quản lý | Chức năng giúp người quản lý quản lý danh mục đồ ăn, nước uống hoặc các dịch vụ khác đi kèm. | <p>1. Người quản lý truy cập vào mục “ Quản lý dịch vụ”.</p> <p>2. Hệ thống hiển thị danh sách dịch vụ.</p> <p>3. Người quản lý có thể chọn các thao tác:</p> <ul style="list-style-type: none"> <li>- Thêm dịch vụ mới.</li> <li>- Chỉnh sửa thông tin dịch vụ.</li> <li>- Xóa dịch vụ.</li> </ul> <p>4. Thay đổi được lưu và cập nhật.</p>                                    |
| Thuê máy        | Khách hàng    | Cho phép khách hàng chọn máy tính để bắt đầu dịch vụ   | <p>1. Khách hàng đến quán và thuê máy.</p> <p>2. Khách chọn máy từ những máy đang trống.</p> <p>3. Hệ thống ghi nhận thời gian bắt đầu và thay đổi trạng thái máy thái máy thành “đang sử dụng”.</p>  |

|            |               |  |  |
|------------|---------------|--|--|
| Order      | Khách hàng    | Cho phép khách hàng gọi đồ ăn, đồ uống khi đang chơi.                          | <ol style="list-style-type: none"> <li>1. Từ giao diện người dùng, khách chọn mục “Order”.</li> <li>2. Chọn tên dịch vụ, nhập số lượng.</li> <li>3. Xác nhận order.</li> <li>4. Hệ thống lưu đơn hàng, gắn với mã máy đang sử dụng.</li> </ol>   |
| Thanh toán | Người quản lý | Chức năng giúp người quản lý tính tổng chi phí khách đã sử dụng và thanh toán. | <ol style="list-style-type: none"> <li>1. Người quản lý truy cập tính năng “Thanh toán”.</li> <li>2. Hệ thống tính thời gian bắt đầu – thời gian kết thúc, tính tổng tiền dựa trên giá phòng, giá giờ.</li> <li>3. Hệ thống cộng thêm các dịch vụ đã order (nếu có).</li> <li>4. Tổng hợp hóa đơn và hiển thị.</li> <li>5. Khách hàng thanh toán và hệ thống cập nhật trạng thái máy thành “trống”.</li> </ol> |

## 4. Thiết kế hệ thống

### 4.1. Kiến trúc hệ thống

- Hệ thống được xây dựng theo mô hình Layered kết hợp với MVC, giúp tách biệt rõ ràng giữa giao diện, nghiệp vụ và xử lý dữ liệu.

- Presentation Layer (MVC)
- Business Logic Layer
- Data Access Layer

#### *Presentation Layer*

- Giao diện người dùng, chia thành các module:
  - Quản lý khách hàng
  - Quản lý máy



- Quản lý phòng
- Quản lý dịch vụ
- Quản lý thuê máy
- Order dịch vụ

**Công nghệ:** Java Swing, JavaFX

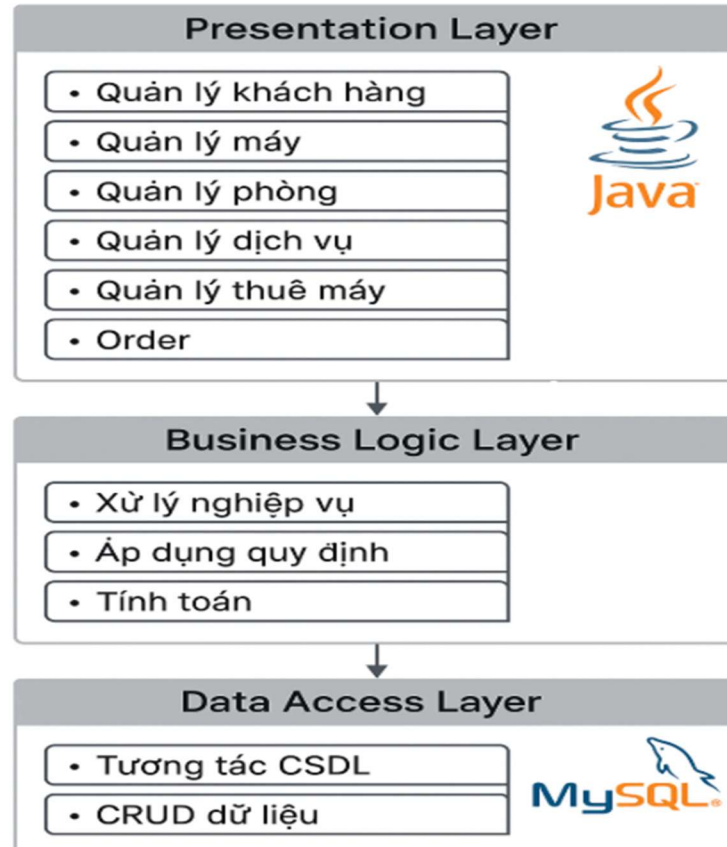
### ***Business Logic Layer***

- Xử lý các nghiệp vụ quan trọng:
  - Kiểm tra ràng buộc dữ liệu.
  - Tính toán (thời gian thuê, chi phí).
  - Áp dụng quy định quán internet (ví dụ: giờ hoạt động, tính phí dịch vụ).

### ***Data Access Layer***

- Kết nối và tương tác với cơ sở dữ liệu MySQL
- Thực hiện các chức năng CRUD (Thêm, Xóa, Chỉnh sửa,.....)
- Quản lý kết nối an toàn, hiệu quả.

### **Sơ đồ kiến trúc tổng quan**

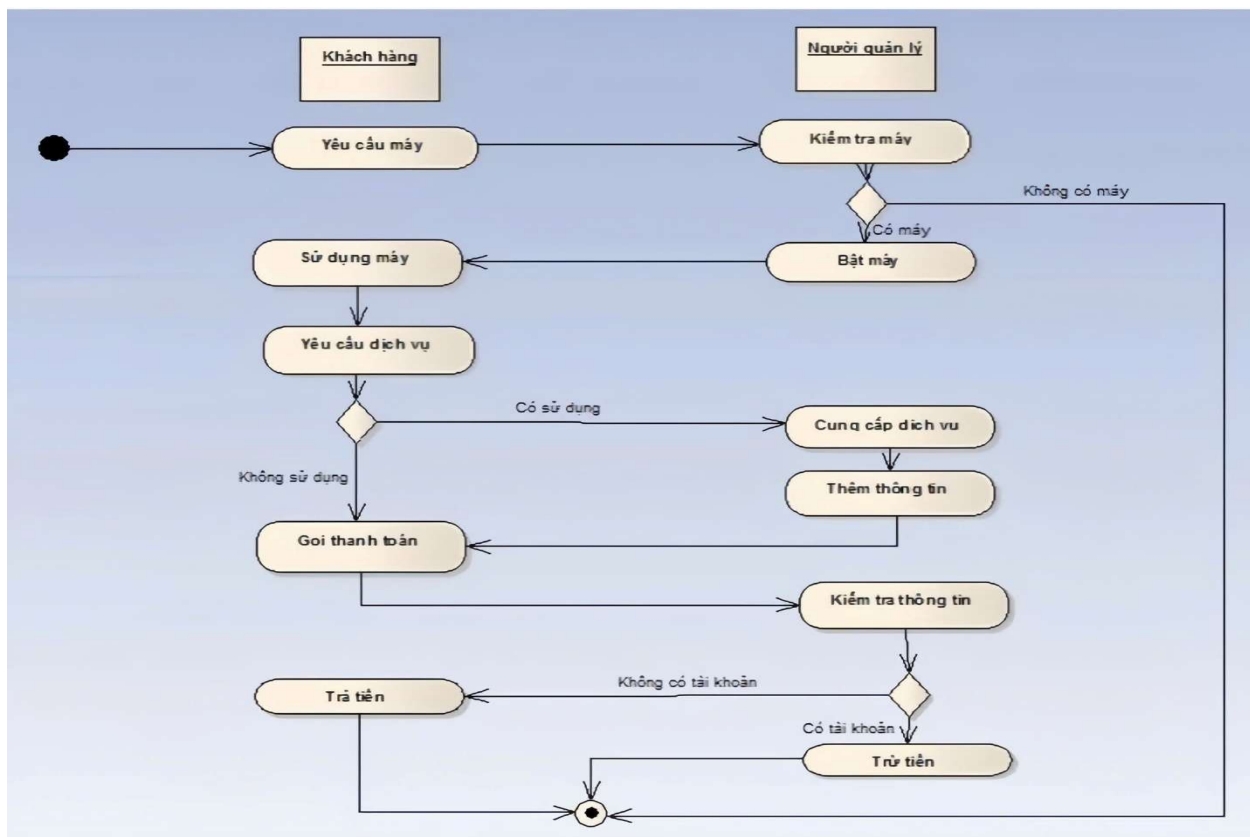


## 4.2. Sơ đồ lớp (Class Diagram)

### a. Lớp chính và quan hệ

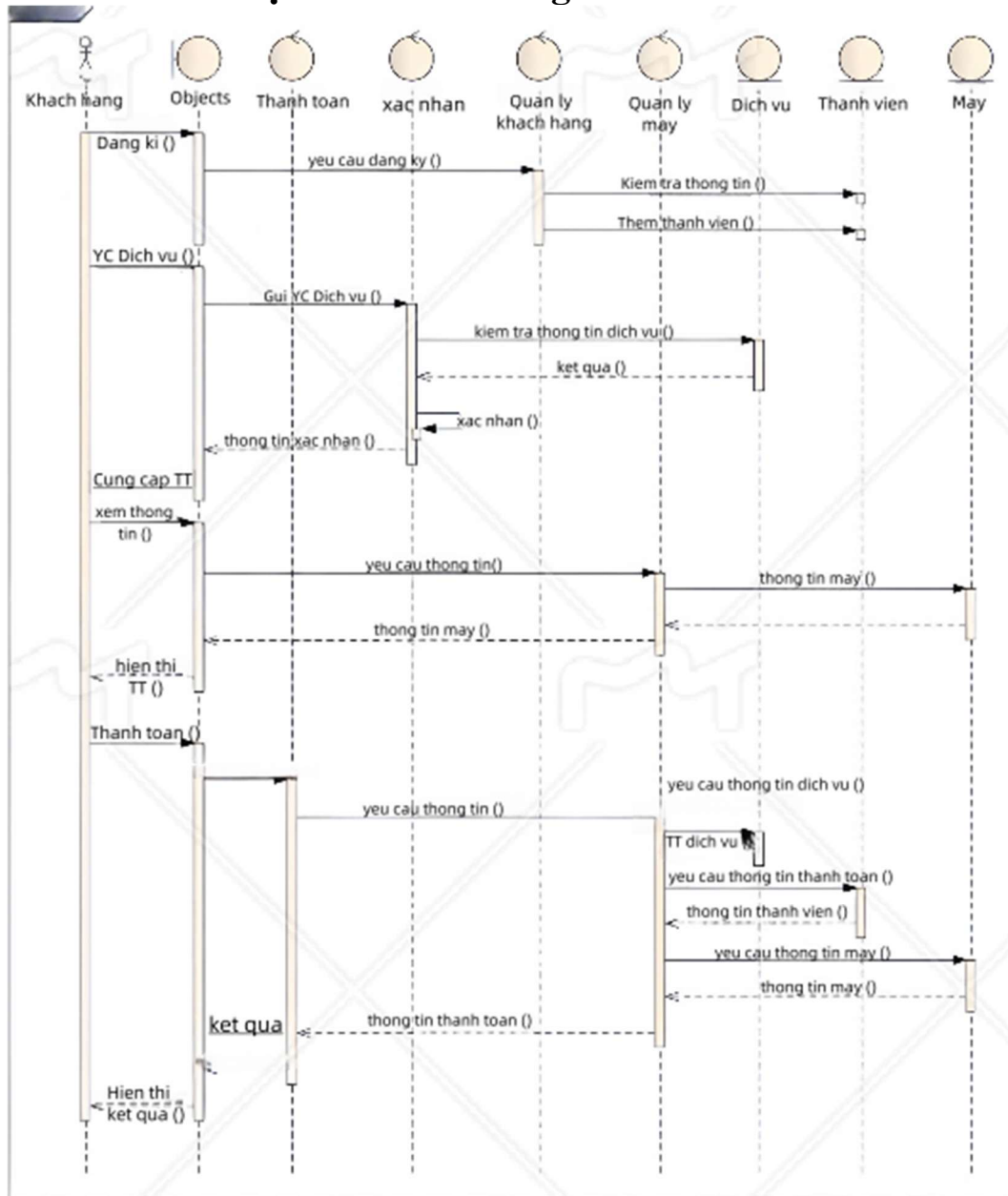
| Lớp              | Thuộc tính chính  | Phương thức chính                             | Quan hệ                                   |
|------------------|---|---|---|
| <b>KhachHang</b> | id, ten, sdt, diaChi, email   | them(), sua(), xoa(),<br>timKiem()            | -   |
| <b>May</b>       | id, tenMay, trangThai, phong  | them(), sua(), xoa(),<br>capNhatTrangThai()   | Thuộc <b>Phong</b>                        |
| <b>Phong</b>     | id, tenPhong, danhSachMay   | them(), sua(), xoa(),<br>timMay()             | Chứa <b>May</b>                           |
| <b>DichVu</b>    | id, tenDichVu, gia  | them(), sua(), xoa(),<br>timKiem()            | -   |
| <b>Thue May</b>  | id, khachHang, may,<br>thoiGianBatDau, thoiGianKetThuc,<br>tongTien | batDauThue(),<br>ketThucThue(),<br>tinhTien() | Liên kết<br><b>KhachHang &amp; May</b>    |
| <b>Order</b>     | id, khachHang, dichVu, soLuong,<br>tongTien                         | themOrder(),<br>tinhTienOrder()               | Liên kết<br><b>KhachHang &amp; DichVu</b> |

### b. Sơ đồ Class Diagram



### 4.3. Sơ đồ trình tự (Sequence Diagram)

#### a. Biểu đồ trình tự của khách hàng



#### b. Biểu đồ trình tự của quản lý



## 6. Giao diện người dùng (UI Design)

### 6.1. Quản lý khách hàng

| Mã khách hàng | Tên khách hàng    | Số điện thoại | Email               | Điểm tích lũy |
|---------------|-------------------|---------------|---------------------|---------------|
| 1             | Nguyễn Minh Ph... | 23015338      | nguyenphuonggth...  | 0             |
| 2             | Đông Thị Ánh      | 23013883      | donganh@gmail....   | 0             |
| 3             | Đình Nhật Tân     | 23013018      | Nhattan@gmail.c...  | 0             |
| 4             | Bùi Thị Hồng Tươi | 23015124      | hongtuoi@gmail.c... | 0             |

Mã khách hàng

Tên khách hàng

Số điện thoại

Email

Điểm tích lũy

| Chức năng                     | Dữ liệu hiển thị   | Cơ chế điều hướng   |
|-------------------------------|--|---|
| Hiển thị danh sách khách hàng | - Bảng bên phải hiển thị danh sách khách hàng:<br>+ Mã khách hàng<br>+ Tên khách hàng<br>+ Số điện thoại<br>+ Email<br>+ Điểm tích lũy | - Truy cập tab Quản lý khách hàng trên thanh menu<br>→ Bảng dữ liệu tự động tải<br>→ Có thể chọn dòng để thao tác sửa/xóa |
| Thêm khách hàng mới           | - Nhập thông tin ở các ô bên trái:<br>+ Mã khách hàng  | - Điền đầy đủ các thông tin<br>→ Nhấn Thêm mới  |

|                          |  |   |
|--------------------------|--|---|
|                          | + Tên<br>+ SĐT<br>+ Email<br>+ Điểm tích lũy (nếu có)<br>→ Nhấn nút Thêm mới   | → Dữ liệu được thêm vào bảng và lưu vào hệ thống (database/file)  |
| Chọn khách hàng          | Khi chọn 1 dòng trong bảng:<br>→ Dữ liệu của dòng đó được load ngược lên form bên trái                                       | - Click vào dòng bất kỳ<br>→ Dữ liệu sẽ đổ ngược vào các ô nhập liệu<br>→ Có thể nhấn Sửa hoặc Xóa tiếp theo    |
| Sửa thông tin khách hàng | - Chọn dòng trong bảng (tên, số điện thoại,...) → thông tin sẽ được load sang các ô bên trái<br>→ Chỉnh sửa rồi nhấn nút Sửa | - Chọn khách<br>→ Thay đổi thông tin trong form<br>→ Nhấn Sửa<br>→ Cập nhật thông tin tương ứng trong danh sách |
| Xóa khách hàng           | - Chọn dòng cần xóa trong bảng<br>→ Nhấn nút Xóa   | - Hệ thống sẽ xóa dòng tương ứng trong bảng<br>→ Có thể hiển thị hộp thoại xác nhận (nếu có)                    |

## 6.2.Quản lí phòng

Quản lý khách hàng

Quản lý phòng

Quản lý máy

Quản lý dịch vụ

Quản lý thuê máy

Order

Mã phòng

Tên phòng

Mô tả

Thêm mới

Sửa

Xóa

| Mã phòng | Tên phòng    | Mô tả   |
|----------|--------------|---------|
| 1        | Phòng thường | Giá rẻ  |
| 2        | Phòng đắt    | Giá đắt |

| Chức năng                | Dữ liệu hiển thị  | Cơ chế điều hướng   |
|--------------------------|---|---|
| Hiển thị danh sách phòng | <ul style="list-style-type: none"> <li>Danh sách các phòng đã có trong hệ thống</li> <li>Gồm: Mã phòng, Tên phòng, Mô tả</li> </ul>   | <ul style="list-style-type: none"> <li>Nhấn vào tab "Quản lý phòng" để xem danh sách phòng.</li> </ul>  |
| Thêm phòng mới           | <ul style="list-style-type: none"> <li>Khi người dùng điền đầy đủ thông tin ở 3 ô trên và nhấn nút "Thêm mới", hệ thống sẽ: <ul style="list-style-type: none"> <li>Kiểm tra trùng mã phòng</li> <li>Thêm dữ liệu vào bảng hiển thị</li> <li>Ghi vào CSDL hoặc file</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>Người dùng điền đầy đủ các ô: mã, tên, mô tả → nhấn "Thêm mới".</li> <li>Nếu thành công: dữ liệu hiện trong bảng bên phải.</li> <li>Nếu lỗi: popup cảnh báo hiển thị.</li> </ul> |

|                     |   |   |
|---------------------|---|---|
| Sửa thông tin phòng | <ul style="list-style-type: none"> <li>- Khi nhấp chọn một hàng trong bảng, dữ liệu phòng sẽ được hiển thị vào các ô nhập liệu để chỉnh sửa.</li> <li>- Sau khi sửa xong, nhấn nút "Sửa" để lưu lại.</li> </ul>                       | <ul style="list-style-type: none"> <li>- Click chọn dòng cần chỉnh sửa → hệ thống điền dữ liệu vào các ô.</li> <li>- Người dùng chỉnh sửa → nhấn "Sửa" để cập nhật dữ liệu mới vào bảng và hệ thống.</li> </ul> |
| Xóa phòng           | <ul style="list-style-type: none"> <li>- Khi chọn một dòng và nhấn "Xóa", hệ thống sẽ:</li> <li>+ Hiện hộp thoại xác nhận (nếu có)</li> <li>+ Xóa phòng khỏi bảng hiển thị</li> <li>+ Gỡ khỏi hệ thống lưu trữ (file/CSDL)</li> </ul> | <ul style="list-style-type: none"> <li>- Click chọn dòng trong bảng → nhấn "Xóa".</li> <li>- Sau khi xóa, dữ liệu phòng đó không còn hiển thị trong bảng</li> </ul>   |

### 6.3. Quản lý máy

| Mã máy | Trạng thái | Thông số | Giá theo giờ | Phòng        |
|--------|------------|----------|--------------|--------------|
| 1      | Rảnh       | i7 16Gb  | 25000.0      | Phòng đắt    |
| 2      | Không rảnh | i5 8GB   | 10000.0      | Phòng thường |
| 3      | Rảnh       | i9 32GB  | 30000.0      | Phòng đắt    |



| <b>Chức năng</b>       | <b>Dữ liệu hiển thị</b>   | <b>Cơ chế điều hướng</b>  |
|------------------------|---|---|
| Hiển thị danh sách máy | <ul style="list-style-type: none"> <li>- Bảng bên phải hiển thị danh sách các máy đã thêm vào hệ thống, với cột: <ul style="list-style-type: none"> <li>+ Mã máy (ID duy nhất)</li> <li>+ Trạng thái (Rảnh / Không rảnh)</li> <li>+ Thông số (CPU, RAM)</li> <li>+ Giá theo giờ</li> <li>+ Phòng (liên kết với phòng đang đặt máy)</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>- Người dùng vào tab "Quản lý máy" trên thanh điều hướng.</li> <li>- Dữ liệu hiển thị tự động từ hệ thống hoặc file lưu trữ.</li> </ul>                      |
| Nhập thông tin máy     | <ul style="list-style-type: none"> <li>- Các ô nhập bên trái gồm: <ul style="list-style-type: none"> <li>+ Mã máy (ID máy)</li> <li>+ Trạng thái ( Rảnh / Không rảnh)</li> <li>+ Thông số (mô tả: CPU, RAM, ổ cứng...)</li> <li>+ Đơn giá/h (tính theo giờ)</li> <li>+ Phòng (chọn phòng chứa máy)</li> </ul> </li> </ul>                                 | <ul style="list-style-type: none"> <li>- Nhập đầy đủ thông tin → chuẩn bị cho việc thêm mới hoặc sửa máy.</li> <li>- Các lựa chọn dropdown giúp đảm bảo tính nhất quán dữ liệu nhập vào.</li> </ul> |
| Thêm máy mới           | <ul style="list-style-type: none"> <li>- Khi người dùng nhập đầy đủ các trường và nhấn nút "Thêm mới", hệ thống sẽ: <ul style="list-style-type: none"> <li>+ Kiểm tra trùng mã</li> <li>+ Lưu thông tin máy vào bảng hiển thị và hệ thống lưu trữ (file/CSDL)</li> </ul> </li> </ul>  | <ul style="list-style-type: none"> <li>- Người dùng nhấn "Thêm mới" sau khi điền đủ thông tin.</li> <li>- Dữ liệu được cập nhật vào danh sách bên phải.</li> </ul>                                  |
| Sửa thông tin máy      | <ul style="list-style-type: none"> <li>- Click dòng cần sửa → thông tin tự đổ vào các ô nhập.</li> <li>- Người dùng chỉnh lại và nhấn Sửa để cập nhật.</li> </ul>   | <ul style="list-style-type: none"> <li>- Click dòng → chỉnh sửa → nhấn Sửa để lưu thay đổi.</li> </ul>  |
| Xóa máy                | <ul style="list-style-type: none"> <li>- Chọn dòng cần xóa và nhấn Xóa → dữ liệu bị gỡ khỏi bảng và hệ thống.</li> </ul>  | <ul style="list-style-type: none"> <li>- Chọn dòng → nhấn Xóa. Có thể có hộp xác nhận trước khi xóa.</li> </ul>   |

## 6.4.Quản lý dịch vụ

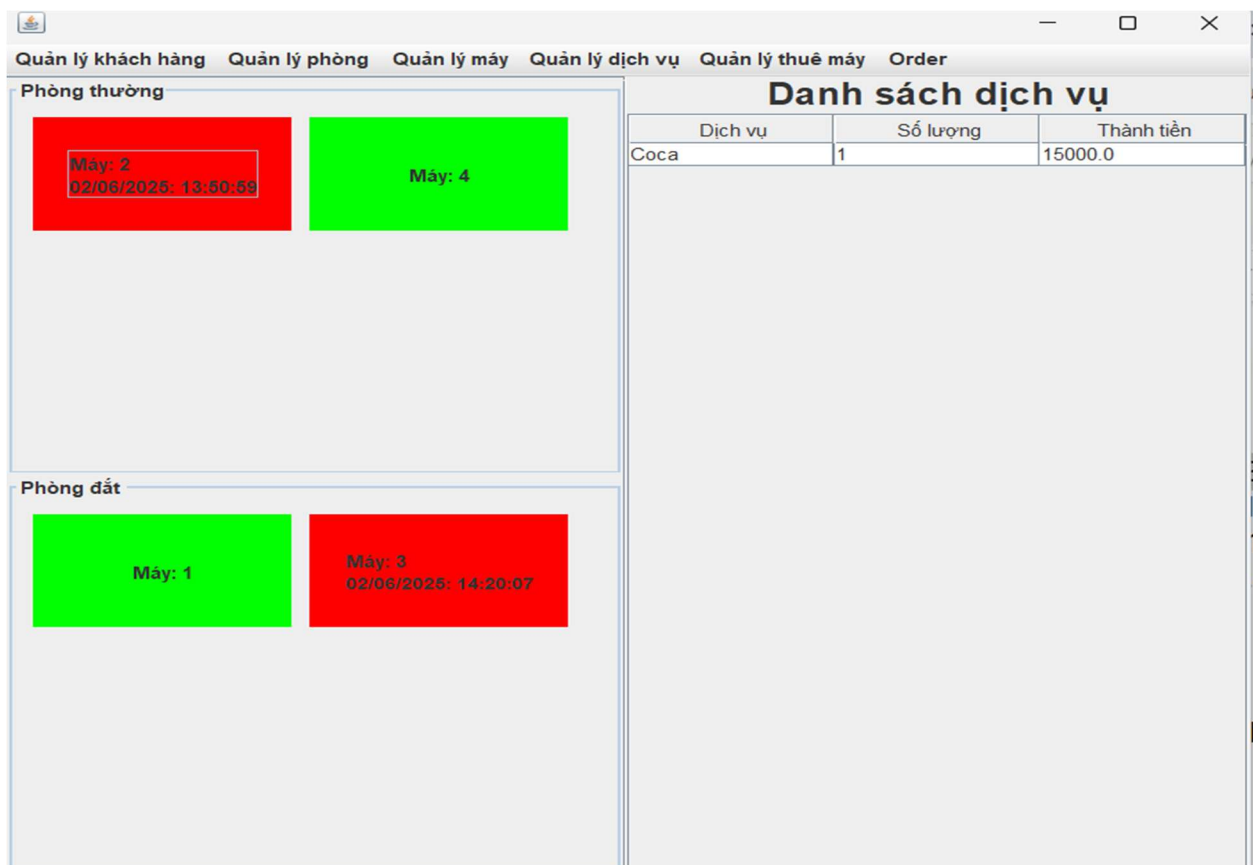
| Mã dịch vụ | Tên dịch vụ | Đơn giá |
|------------|-------------|---------|
| 1          | Coca        | 15000.0 |
| 2          | Pepsi       | 10000.0 |
| 3          | Cơm         | 25000.0 |

| Chức năng                  | Dữ liệu hiển thị   | Cơ chế điều hướng  |
|----------------------------|--|--|
| Hiển thị danh sách dịch vụ | Danh sách các dịch vụ đã được thêm vào gồm:<br>+ Mã dịch vụ<br>+ Tên dịch vụ<br>+ Đơn giá            | - Tự động hiển thị ngay trong bảng bên phải sau khi mở tab “Quản lý dịch vụ” hoặc sau khi thêm/sửa dữ liệu.    |
| Thêm mới dịch vụ           | Nhập đầy đủ 3 trường:<br>+ Mã dịch vụ (VD: 1)<br>+ Tên dịch vụ (VD: Coca)<br>+ Đơn giá (VD: 15000.0) | - Nhập dữ liệu vào các ô bên trái → Nhấn Thêm mới → Dữ liệu được lưu vào hệ thống và hiển thị ở bảng bên phải. |
| Sửa thông tin dịch vụ      | - Khi chọn dòng cần sửa, dữ liệu hiển thị lại vào các ô nhập liệu.                                   | - Click chọn dòng dịch vụ → sửa thông tin trong các ô → Nhấn Sửa để cập nhật lại dòng đã chọn trong bảng       |

|             |   |  |
|-------------|---|--|
| Xóa dịch vụ | - Dòng được chọn sẽ bị xóa khỏi danh sách hiển thị và hệ thống lưu trữ. | - Chọn dòng cần xóa → Nhấn Xóa → Hiện hộp thoại xác nhận → đồng ý thì dòng bị xóa khỏi bảng. |
|-------------|---|--|

## 6.5.Quản lý thuê máy

### a.Thuê máy



| Chức năng               | Dữ liệu hiển thị  | Cơ chế điều hướng   |
|-------------------------|---|---|
| Hiển thị trạng thái máy | - Danh sách máy chia theo khu vực: "Phòng thường", "Phòng đắt". | - Nhấn vào máy màu xanh để bắt đầu thuê máy<br>- Nhấn vào máy màu đỏ để |

|                           |  |  |
|---------------------------|--|--|
|                           | <ul style="list-style-type: none"> <li>- Máy màu xanh: chưa thuê</li> <li>- Máy màu đỏ: đang thuê + thời gian bắt đầu thuê</li> </ul>  | kết thúc phiên thuê và thanh toán  |
| Quản lý phân loại máy     | <ul style="list-style-type: none"> <li>- Các máy được chia theo khu vực: Phòng thường và Phòng đắt.</li> <li>- Giao diện chia khung để dễ quản lý khu vực.</li> </ul>                                  | - Không điều hướng chỉ là phân nhóm hiển thị trực tiếp trong giao diện chính |
| Hiển thị dịch vụ đi kèm   | <ul style="list-style-type: none"> <li>- Bảng bên phải hiển thị danh sách dịch vụ khách đã dùng trong phiên (VD: Coca).</li> <li>- Gồm các cột: Dịch vụ, Số lượng, Thành tiền.</li> </ul>              | - Nhấn vào tab "Order" để thêm dịch vụ vào đơn thuê đang hoạt động           |
| Ghi nhận thời gian thuê   | <ul style="list-style-type: none"> <li>- Khi click máy màu xanh → hệ thống ghi lại thời gian bắt đầu thuê và chuyển sang màu đỏ.</li> <li>- Dữ liệu được cập nhật trực tiếp trên giao diện.</li> </ul> | Giao diện tự động cập nhật, không chuyển trang                               |
| Kết thúc thuê & tính tiền | <ul style="list-style-type: none"> <li>- Khi click máy đang thuê (màu đỏ) → thực hiện thanh toán theo thời gian thuê + dịch vụ đã dùng.</li> <li>- Tính toán và tích điểm</li> </ul>                   | - Popup/tự động mở cửa sổ thanh toán   |

## **b. Thanh toán**

Quản lý khách hàng

Quản lý phòng

Quản lý máy

Quản lý dịch vụ

Quản lý thuê máy

Order

Phòng thường

Máy: 2  
02/06/2025: 13:50:59

Máy: 4

Danh sách dịch vụ

| Dịch vụ | Số lượng | Thành tiền |
|---------|----------|------------|
| Coca    | 1        | 15000.0    |

Phòng đặt

Máy: 1

Mã thuê máy

4

Tổng tiền

40364.0

Mã khách hàng

Check

Số điểm tích lũy

0

Số điểm sử dụng

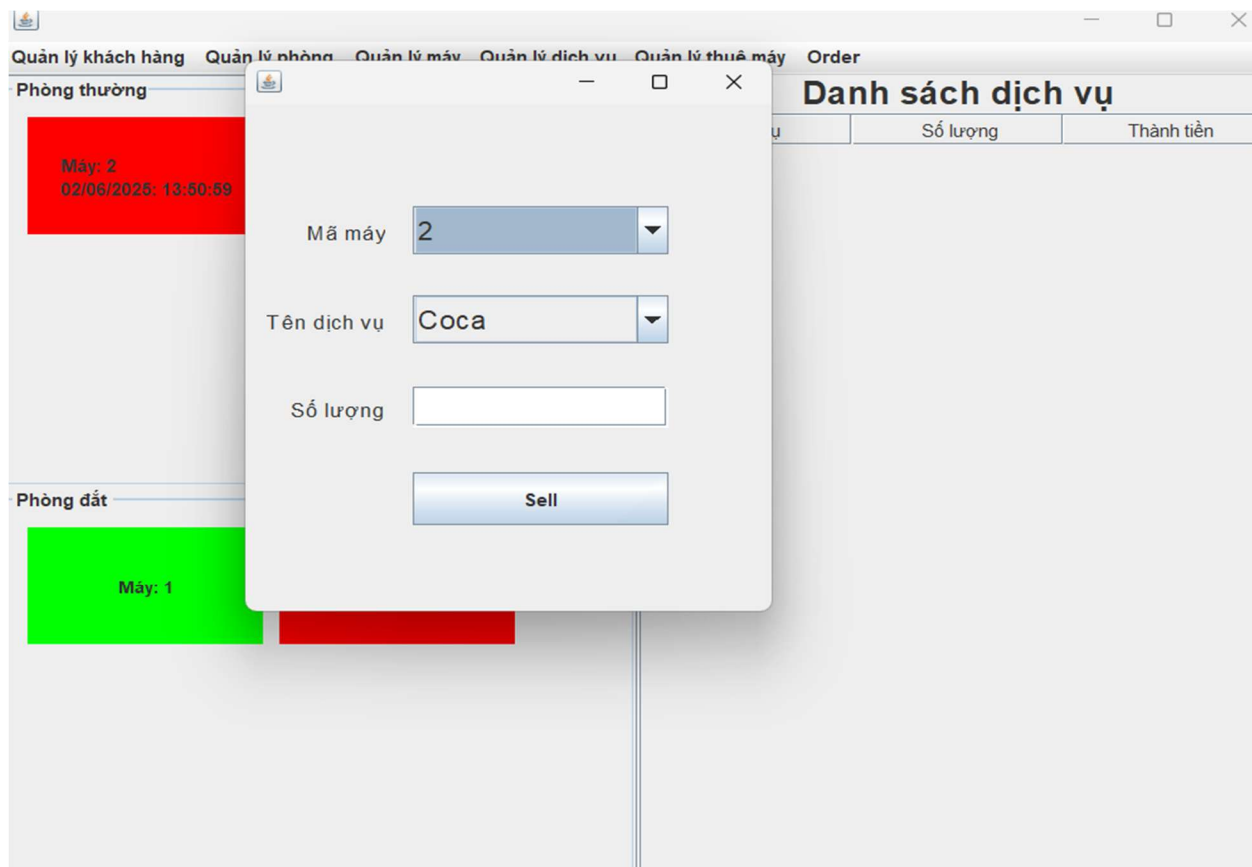
0

Thanh toán

| Chức năng            | Dữ liệu hiển thị  | Cơ chế điều hướng   |
|----------------------|---|---|
| Xác định mã thuê máy | - Trường "Mã thuê máy" hiển thị số hiệu máy đã chọn (ví dụ: 4)      | - Hệ thống tự động điền từ máy vừa click (màu đỏ)         |
| Hiển thị tổng tiền   | - Trường "Tổng tiền" hiển thị số tiền phải thanh toán (VD: 40364.0) | - Tự động tính dựa trên thời gian thuê và dịch vụ sử dụng |
| Nhập mã khách hàng   | - Nhập tay vào ô "Mã khách hàng"                                    | - Nhấn Check để tra thông tin điểm tích lũy               |

|                        |   |  |
|------------------------|---|--|
| Kiểm tra điểm tích lũy | <ul style="list-style-type: none"> <li>- Sau khi nhấn "Check", 2 ô được cập nhật:</li> <li>+ "Số điểm tích lũy"</li> <li>+ "Số điểm sử dụng" (có thể nhập vào để trừ tiền)</li> </ul> | <ul style="list-style-type: none"> <li>- Có thể nhập điểm vào ô "Số điểm sử dụng" để giảm tổng tiền thanh toán</li> </ul>  |
| Thực hiện thanh toán   | <ul style="list-style-type: none"> <li>- Nhấn nút Thanh toán để xác nhận giao dịch</li> </ul>   | <ul style="list-style-type: none"> <li>- Sau khi thanh toán thành công:</li> <li>- Máy sẽ chuyển sang màu xanh</li> <li>- Giao diện popup tự đóng lại</li> </ul> |

## 6.6. Order



| Chức năng       | Dữ liệu hiển thị  | Cơ chế điều hướng   |
|-----------------|---|---|
| Đặt món (Order) | <ul style="list-style-type: none"> <li>- Khi nhấn vào nút "Order" tại máy đang hoạt động:</li> <li>+ Hiển thị bảng nhỏ dạng popup gồm các nhóm món: thức uống, đồ ăn, combo...</li> <li>+ Mỗi món hiển thị: tên món, đơn giá, nút chọn số lượng nhập trực tiếp</li> <li>+ Tổng tiền cập nhật theo thời gian thực (số lượng * đơn giá)</li> <li>+ Nút xác nhận để lưu đơn hàng hoặc thoát để hủy thao tác</li> </ul> | <ul style="list-style-type: none"> <li>- Nhấn vào tab "Order" tại máy đang sử dụng</li> <li>- Hệ thống mở bảng nhỏ ngay tại vị trí máy</li> <li>- Khi bấm xác nhận, đơn hàng được lưu lại kèm thông tin máy và cập nhật trong danh sách dịch vụ đi kèm</li> <li>- Có thể quay lại để sửa đơn nếu chưa thanh toán</li> </ul> |

## 7. Mẫu thiết kế áp dụng ( Design Patterns)

### 7.1. Model-View-Controller (MVC)

#### - Mô tả

- + MVC là một kiến trúc phần mềm được sử dụng để tổ chức và quản lý mã nguồn trong quá trình phát triển ứng dụng, kiến trúc tách biệt ứng dụng thành ba thành phần:
  - Model – quản lý dữ liệu và logic nghiệp vụ, tương tác trực tiếp với cơ sở dữ liệu hoặc dịch vụ lưu trữ, không quan tâm đến cách hiển thị dữ liệu.
  - View – Chịu trách nhiệm hiển thị giao diện và xuất dữ liệu từ Model, không chứa logic nghiệp vụ, chỉ có logic trình bày (layout, format).
  - Controller – Nhận và xử lý các yêu cầu (request) từ người dùng, gọi Model để cập nhật hoặc truy vấn dữ liệu, sau đó chọn View phù hợp để phản hồi, giữ vai trò “điều phối”, đảm bảo Model và View hoạt động đồng bộ.
- + Giúp che giấu chi tiết triển khai từng phần, giảm phụ thuộc chặt chẽ giữa giao diện, xử lý nghiệp vụ và lưu trữ dữ liệu.

#### - Vai trò

##### + Tách biệt trách nhiệm

Model chỉ lo lưu trữ, truy vấn và cập nhật dữ liệu, View chỉ lo hiển thị,

Controller chỉ lo điều hướng và xử lý tương tác, không chứa logic lưu trữ hay trình bày.

+ **Trừu tượng hóa giao diện**

View gọi Controller qua các hành động (action) mà không cần biết dữ liệu bên trong, Controller gọi Model qua API như findAll(), save()... mà không cần quan tâm cách lưu.

+ **Giảm phụ thuộc**

Thay đổi giao diện (HTML, template) không ảnh hưởng Model hoặc Controller, thay đổi logic nghiệp vụ chỉ cần sửa Model/Controller, không động View.

+ **Hỗ trợ phát triển song song**

Front-end (View) và back-end (Model/Controller) có thể làm việc tách biệt.

+ **Dễ dàng bảo trì & mở rộng**

Bổ sung tính năng mới chỉ cần thêm Controller và View tương ứng, thay đổi nguồn dữ liệu chỉ cần điều chỉnh trong Model.

## - Mã ví dụ

*// Lấy danh sách dịch vụ từ Model (DAO) và hiển thị lên bảng (View)*

*List dịchVus = dichVuDao.getListDichVu();*

*for (DichVu c : dịchVus) {*

*Object[] row = {*

*c.getMaDichVu(),*

*c.getTenDichVu(),*

*c.getGiaDichVu()*

*};*

*model.addRow(row);*

*}*

*jTable1.setModel(model);*

*// Xử lý sự kiện thêm mới dịch vụ: Controller tạo đối tượng Model và cập nhật View*

*if (dichVuDao.add(new DichVu(0, txt\_tendv.getText(),*

*Double.parseDouble(txt\_dongia.getText())))) {*

*LoadData(); // Cập nhật lại bảng sau khi thêm*

*JOptionPane.showMessageDialog(rootPane, "Thêm thành công"); }*

→ Khi form khởi tạo, nó sử dụng phương thức getListDichVu() (Model) để lấy dữ liệu và điền vào jTable1 (View). Khi người dùng nhập thông tin và bấm nút “Thêm mới”, đoạn code xử lý sự kiện sẽ tạo DichVu(0, tên, giá) và gọi dichVuDao.add(...)



để lưu Model, sau đó gọi LoadData() để làm mới bảng hiển thị (View) Các tương tác này minh chứng rõ vai trò Model (lưu trữ và thao tác dữ liệu), View (hiển thị giao diện) và Controller (xử lý sự kiện và điều phối Model-View) trong dự án.

## 7.2. Data Access Object (DAO)

### + *Mô tả*

- DAO là một mẫu thiết kế thuộc nhóm Structural/Architectural, chuyên trách việc tương tác với nguồn dữ liệu (CSDL, file, dịch vụ ngoài,...). DAO thường là lớp (hoặc interface và các implement) chỉ chứa các phương thức truy vấn, thêm, sửa, xóa dữ liệu (CRUD), không chứa logic nghiệp vụ.
- Có thể hiểu DAO giống như “cầu nối” giữa lớp nghiệp vụ (service/business layer) và lớp lưu trữ dữ liệu, giúp che giấu chi tiết triển khai (SQL, ORM, API gọi ngoài...) và đảm bảo tính nhất quán khi truy cập nguồn dữ liệu.

### + *Vai trò*

- **Tách biệt truy xuất dữ liệu**

DAO gom gọn tất cả các thao tác với cơ sở dữ liệu (INSERT, UPDATE, DELETE, SELECT), tách riêng khỏi lớp nghiệp vụ, giúp giảm sự phụ thuộc giữa hai tầng.

- **Trừu tượng hóa nguồn dữ liệu**

Khai báo các phương thức như findById(), save(), delete(),... mà không để lộ SQL hay API cụ thể, cho phép dễ dàng thay đổi backend (ví dụ chuyển từ MySQL sang MongoDB) mà không ảnh hưởng đến các lớp khác.

- **Quản lý kết nối và giao dịch**

DAO chịu trách nhiệm mở/đóng kết nối, bắt và xử lý transaction, rollback khi lỗi để đảm bảo tính toàn vẹn dữ liệu.

- **Hỗ trợ unit testing**

Nhờ DAO được trừu tượng qua interface, ta có thể dễ dàng mock hoặc stub để kiểm thử các lớp service mà không cần kết nối thật tới CSDL.

- **Tối ưu hiệu năng**

Tại đây có thể triển khai kỹ thuật như query batching, lazy loading, caching... để cải thiện tốc độ và giảm tải cho cơ sở dữ liệu.

- **Dễ dàng bảo trì và mở rộng**

Khi cần thay đổi schema, điều chỉnh query hay bổ sung nguồn dữ liệu mới (ví dụ gọi API ngoài), chỉ cần sửa/extend trong DAO mà không cần chạm tới business logic.

**- Mã ví dụ:**

```
public class DichVuDao {
    private static final String DICHVU_FILE = "src/main/java/XMLFile/DichVu.xml";
    private List<DichVu> listDichVu;

    public DichVuDao() {
        this.listDichVu = readListDichVu();
        if (listDichVu == null) listDichVu = new ArrayList<>();
    }

    public void writeListDichVu(List<DichVu> dichVus) {
        FileUtils.writeXMLToFile(DICHVU_FILE, new DichVuXML(dichVus));
    }

    public List<DichVu> readListDichVu() {
        DichVuXML xmlData = (DichVuXML) FileUtils.readXMLFile(DICHVU_FILE,
            DichVuXML.class);
        return (xmlData != null) ? xmlData.getDichVuList() : new ArrayList<>();
    }
}
```

→ DichVuDao khởi tạo danh sách từ file XML. Phương thức writeListDichVu gọi FileUtils để lưu lại. Lớp DAO đảm nhận việc định dạng và quản lý lưu trữ (ưu tiên thao tác với lớp DichVuXML), còn phần giao diện sẽ chỉ gọi DAO này khi cần đọc/ghi dữ liệu.

### 7.3. Data Transfer Object (DTO)

**- Mô tả:**

- DTO (Data Transfer Object) là mẫu thiết kế thuộc nhóm Structural, dùng để truyền dữ liệu giữa các lớp, tầng hoặc hệ thống.

- DTO thường là lớp đơn giản (POJO) chỉ chứa thuộc tính (fields), constructor, getters/setters, không chứa logic xử lý
- Có thể hiểu DTO giống như một "chiếc hộp dữ liệu" dùng để mang dữ liệu đi từ nơi này đến nơi khác trong chương trình.

## - Vai trò:

- **Tách biệt dữ liệu**  
 DTO giúp chia tách rõ ràng dữ liệu cần truyền giữa các lớp hoặc module khác nhau trong ứng dụng, như giữa giao diện người dùng và lớp nghiệp vụ, tránh phụ thuộc chặt chẽ.
- **Truyền dữ liệu**  
 Dùng để chuyển dữ liệu giữa các thành phần như UI, dịch vụ và cơ sở dữ liệu.
- **Xác thực dữ liệu**  
 DTO có thể chứa logic hoặc cấu trúc để kiểm tra tính hợp lệ của dữ liệu trước khi truyền đi, giúp duy trì tính toàn vẹn và tránh lỗi.
- **Phản ánh cấu trúc dữ liệu**  
 DTO định nghĩa rõ ràng cấu trúc dữ liệu mà các thành phần sử dụng chung, đảm bảo mọi bên đều hiểu và xử lý dữ liệu theo cùng một chuẩn.
- **Cải thiện khả năng tái sử dụng mã**  
 DTO có thể dùng lại trong nhiều phần của ứng dụng, giảm việc phải viết lại các định nghĩa dữ liệu và logic xử lý tương tự.
- **Tăng cường khả năng bảo trì ứng dụng**  
 Nhờ tách riêng dữ liệu ra khỏi logic nghiệp vụ, DTO giúp ứng dụng dễ bảo trì và nâng cấp hơn, vì thay đổi dữ liệu ít ảnh hưởng đến các lớp khác.
- **Hỗ trợ lập mô hình dữ liệu**  
 DTO giúp mô hình hóa dữ liệu và các quy tắc liên quan, đảm bảo dữ liệu được tổ chức và quản lý chặt chẽ theo yêu cầu nghiệp vụ.
- **Giảm thiểu sự phụ thuộc**  
 DTO giúp giảm sự liên kết trực tiếp giữa các lớp hoặc module, tạo

sự độc lập và linh hoạt hơn trong việc phát triển và mở rộng ứng dụng.

## - Mã Ví Dụ

```
// Trong FormPhong.java - sự kiện nút Thêm mới
if (phongDao.add(new Phong(
    0,
    txt_tenphong.getText(),
    txt_mota.getText()
))) {
    LoadData();
    JOptionPane.showMessageDialog(rootPane, "Thêm thành công");
}
```

→ Thêm mới phòng (FormPhong ⇒ PhongDao): Trong FormPhong.java, khi người dùng bấm “Thêm mới”, mã tạo một đối tượng DTO Phong mới từ giá trị nhập liệu và truyền nó cho PhongDao để lưu. Ở đây new Phong(0, txt\_tenphong.getText(), txt\_mota.getText()) là tạo DTO chứa thông tin phòng mới. Phương thức PhongDao.add(Phong phong) sẽ nhận DTO này, gán mã (maPhong) và ghi vào file XML. Quá trình truyền đối tượng giữa Form và DAO thể hiện rõ vai trò của DTO.

## 8. Chiến lược kiểm thử

### 8.1. Chiến lược kiểm thử

#### a. Kiểm thử đơn vị ( Unit test)

- Unit test là kiểm tra các đơn vị chức năng nhỏ nhất của mã nguồn. Với phần mềm này, unit test tập trung vào các phương thức của lớp DAO và xử lý dữ liệu (ví dụ: hàm thêm, sửa, xóa khách hàng trong cơ sở dữ liệu, tính toán giá thuê máy, ...) để đảm bảo từng thành phần hoạt động đúng. Mỗi chức năng riêng lẻ được viết kịch bản kiểm thử riêng, có thể dùng Unit để tự động hóa kiểm thử.
- **Mục tiêu:** Xác nhận từng thành phần nhỏ nhất (hàm, phương thức, lớp) hoạt động đúng như thiết kế mà không phụ thuộc vào các thành phần khác.
- **Phương pháp áp dụng:** Các lập trình viên viết kịch bản kiểm thử (test case) tự động cho từng class/module. Ví dụ, kiểm thử đơn vị được áp dụng

cho các class như CustomerManager (thêm, sửa, tìm kiếm khách hàng), RoomManager (quản lý phòng: khai báo, đặt phòng), MachineManager (quản lý máy: bật/tắt, gán máy cho phòng), ServiceManager (thêm sửa dịch vụ, giá thành), v.v. Mỗi phương thức của class cần có ít nhất một test case hợp lệ và các tình huống sai (negative case) để kiểm tra phản hồi lỗi.

- **Công cụ:** Sử dụng framework kiểm thử đơn vị (như JUnit, NUnit, pytest, v.v.) để chạy tự động các test case. Kết quả test chỉ rõ hàm/phương thức nào lỗi, giúp sửa chữa ngay khi phát triển.

- **Ví dụ cụ thể:** Với class CustomerManager, viết test cho phương thức addCustomer(), kiểm tra việc thêm khách mới với dữ liệu hợp lệ và cảnh báo khi trùng mã khách hàng. Với ServiceManager, kiểm thử hàm tính tổng tiền dịch vụ (calculateServiceTotal()), đưa vào một loạt dịch vụ và kiểm tra kết quả trả về đúng với bảng giá.

## **b. Kiểm thử tích hợp (Integration Test)**

- Integration Test là kiểm tra sự tương tác giữa các module của hệ thống. Cụ thể, cần kiểm thử luồng dữ liệu giữa lớp DAO và giao diện người dùng (GUI) – ví dụ: thêm khách hàng qua form giao diện và xác nhận dữ liệu được lưu vào CSDL đúng, hay khi thanh toán từ GUI thì hóa đơn được tạo đúng trong DAO. Mục tiêu là phát hiện lỗi phát sinh do tích hợp không đúng giữa các bộ phận.

- **Mục tiêu:** Đánh giá sự phối hợp giữa các module hoặc tính năng để đảm bảo chúng tương tác đúng. Tập trung vào giao diện và luồng dữ liệu qua lại giữa các bộ phận đã hoạt động riêng lẻ tốt ở cấp đơn vị.

- **Kịch bản áp dụng:** Kiểm thử các trường hợp nghiệp vụ điển hình khi các module kết hợp với nhau. Ví dụ, quy trình thuê máy kèm dịch vụ là tình huống tích hợp: khi khách hàng thuê một máy (thông qua RentalManager), hệ thống phải cập nhật trạng thái máy (bằng MachineManager), cập nhật thông tin thuê trong CustomerManager, và khi khách order dịch vụ (bằng OrderServiceManager), giá dịch vụ được cộng vào hóa đơn và trừ tồn kho (nếu có). Kiểm thử tích hợp sẽ chạy các luồng này để đảm bảo dữ liệu lưu trữ và xử lý liên tục, không có xung đột.

- **Các bước thử:** Ví dụ, một test tích hợp có thể gồm: tạo mới khách, đăng nhập thuê phòng với máy tính cụ thể, tiếp tục order một số dịch vụ,

sau đó tính toán thanh toán. Kết thúc test, kiểm tra dữ liệu trong cơ sở dữ liệu (hoặc các module) như: máy được đánh dấu đang sử dụng, dịch vụ đã giảm số lượng tồn kho, hóa đơn tổng tiền đúng.

- **Công cụ:** Có thể sử dụng bộ khung kiểm thử tích hợp hoặc test scripts tự động kiểm tra toàn bộ luồng. Ngoài ra, tích hợp kiểm thử API hay giao diện người dùng (GUI) cũng đảm bảo các module giao tiếp đúng.

### c. Kiểm thử chức năng (Functional Test)

- Functional Test là kiểm tra các chức năng chính của phần mềm theo yêu cầu. Các test case ở mức này sẽ tập trung vào các luồng nghiệp vụ: đăng nhập hệ thống, quản lý khách hàng (thêm, sửa, xóa), quản lý máy tính, quản lý dịch vụ, thuê máy, order dịch vụ, thanh toán... Mỗi chức năng chính được thực hiện đầy đủ các trường hợp kiểm thử dương và kiểm thử biên (nhập sai, thiếu thông tin, v.v.) để đảm bảo hệ thống hoạt động đúng như mong đợi.

- **Mục tiêu:** Kiểm tra hoạt động của tính năng tương ứng với yêu cầu đã đưa ra. Là kiểm thử hộp đen, chỉ quan tâm vào kết quả đầu ra so với đặc tả mà không cần xem mã nguồn bên trong.

- **Phương pháp áp dụng:** Tập trung vào từng chức năng chính của phần mềm một cách tổng thể: quản lý khách hàng, phòng, máy, dịch vụ, chức năng thuê máy và order dịch vụ. Ví dụ, kiểm thử chức năng đăng ký khách hàng sẽ kiểm tra xem sau khi nhập thông tin, khách hàng có thực sự được thêm vào danh sách không. Kiểm thử chức năng thuê máy sẽ mô phỏng toàn bộ quy trình: chọn phòng, chọn máy, xác nhận thuê, kiểm tra thông báo thành công và cập nhật đúng trên giao diện.

- **Ví dụ cụ thể:** Với chức năng Order dịch vụ, tạo kịch bản thử: khách hàng trong phòng đang thuê vào phần đặt dịch vụ, chọn một số loại dịch vụ, xem tổng tiền hiện lên đúng. Thực thi các tình huống như đặt dịch vụ lớn hơn tồn kho để kiểm tra hệ thống báo lỗi. Các kết quả báo cáo doanh thu, tổng doanh thu cũng là các trường hợp chức năng quan trọng cần kiểm thử.

- **Trình tự thực hiện:** Từng chức năng sẽ có bộ test case tương ứng, chạy tuần tự như một người dùng cuối: đăng nhập, quản lý khách hàng (thêm/sửa/xóa), quản lý phòng/máy (khai báo, phân công), tạo đơn thuê

máy, ghi nhận dịch vụ kèm theo, và in hóa đơn. Mỗi bước cần so khớp với đặc tả yêu cầu của phần mềm.

#### **d. Kiểm thử chấp nhận (Acceptance Test)**

- Acceptance Test là do người dùng cuối hoặc bên liên quan thực hiện, nhằm xác minh phần mềm đáp ứng yêu cầu thực tế. Bằng cách mô phỏng các kịch bản sử dụng thực tế (kịch bản từ đăng nhập cho đến thanh toán khi thuê máy và order dịch vụ), chúng ta kiểm tra xem giao diện và chức năng có thuận tiện và đúng như người dùng mong đợi không. Ví dụ, người dùng sẽ kiểm tra quy trình đăng nhập, tìm kiếm và thêm khách hàng, tiến hành thuê máy và thanh toán, đảm bảo phần mềm phản hồi chính xác và dễ sử dụng.
- **Mục tiêu:** Xác nhận hệ thống đáp ứng đầy đủ yêu cầu nghiệp vụ và tiêu chí của khách hàng/sử dụng thực tế trước khi đi vào sử dụng. Đây là bước kiểm thử cuối cùng, đảm bảo phần mềm có thể chấp nhận được.
- **Phương pháp áp dụng:** Dựa trên kịch bản của người dùng hoặc chủ quán Internet, thực hiện các tình huống sử dụng thật tế: từ tiếp nhận khách (đăng ký thông tin), thuê máy và order dịch vụ, đến thanh toán và ghi nhận doanh thu. Đảm bảo các chức năng chính (quản lý khách, phòng, máy, dịch vụ, thuê máy, order dịch vụ) liên kết trơn tru và không còn lỗi nghiêm trọng.
- **Ví dụ cụ thể:** Tổ chức chạy thử toàn bộ quy trình: nhân viên đăng ký khách mới, khách thuê máy ở một phòng có máy tính trống, trong quá trình thuê khách order thêm dịch vụ (đồ ăn, nước uống), sau đó tính tiền và thanh toán. Kết thúc, kiểm tra báo cáo tổng hợp để đảm bảo tổng doanh thu đúng bằng tổng tiền của thuê máy cộng dịch vụ. Các vấn đề về giao diện hoặc trải nghiệm người dùng (như thao tác đơn giản, thông báo rõ ràng) cũng được đánh giá.
- **Tiêu chí đánh giá:** Phần mềm chỉ được chấp nhận khi mọi tính năng chính hoạt động đúng yêu cầu nghiệp vụ và user thử nghiệm (hoặc khách hàng thực) hài lòng với kết quả. Nếu có lỗi phát sinh, cần sửa chữa và lặp lại kiểm thử đến khi đạt yêu cầu.

## 8.2. Kịch bản kiểm thử (Test cases)

### a. Đăng Nhập

| Chức năng | Mô tả kiểm thử                        | Input                                 | Expected Output                                   | Kết quả (Pass/Fail) |
|-----------|---------------------------------------|---------------------------------------|---|---------------------|
| Đăng nhập | Đăng nhập với thông tin hợp lệ        | Tài khoản: "admin", Mật khẩu: "123"   | Đăng nhập thành công, chuyển đến giao diện chính  | Pass                |
| Đăng nhập | Đăng nhập với tài khoản không tồn tại | Tài khoản: "unknown", Mật khẩu: "123" | Hiển thị thông báo lỗi “Tên đăng nhập không đúng” | Fail                |
| Đăng nhập | Đăng nhập với mật khẩu sai            | Tài khoản: "admin", Mật khẩu: "wrong" | Hiển thị thông báo lỗi “Mật khẩu không đúng”      | Fail                |
| Đăng nhập | Để trống thông tin đăng nhập          | Tài khoản: "", Mật khẩu: ""           | Hiển thị cảnh báo yêu cầu nhập đầy đủ thông tin   | Fail                |

### b. Quản lý khách hàng

| Chức năng       | Mô tả kiểm thử                         | Input   | Expected Output  | Kết quả (Pass/Fail) |
|-----------------|--|---|--|---------------------|
| Thêm khách hàng | Thêm khách hàng mới với dữ liệu hợp lệ | Tên: "Nguyễn Văn A", SĐT: "09876543", Địa chỉ email: "vana@gmail.com" | Khách hàng mới được lưu và hiển thị trong danh sách khách hàng | Pass                |
| Thêm Khách hàng | Thêm khách hàng với thông tin thiếu    | Tên: "", SĐT: "09876543", Địa chỉ email "vana@gmail.com"              | Hiển thị cảnh báo “Vui lòng nhập đầy đủ thông tin khách hàng”  | Fail                |
| Sửa             | Cập nhật                               | Chọn khách hàng ID=1,   | Thông tin khách  | Pass                |



|                |                                     |   |  |      |
|----------------|-------------------------------------|---|--|------|
| khách hàng     | thông tin khách hàng hợp lệ         | sửa Tên thành "Nguyễn Văn C", lưu           | hàng được cập nhật   |      |
| Sửa khách hàng | Cập nhật khách hàng với dữ liệu sai | Chọn khách hàng ID=1, sửa SĐT thành "", lưu | Hiển thị cảnh báo “Vui lòng nhập số điện thoại ”                 | Fail |
| Xóa khách hàng | Xóa khách hàng tồn tại              | Chọn khách hàng ID=2, nhấn Xóa              | Khách hàng bị xóa khỏi danh sách, thông báo thành công           | Pass |
| Xóa Khách hàng | Xóa khách hàng không tồn tại        | Chọn khách hàng ID=999 (không có), nhấn Xóa | Hiển thị cảnh báo “Khách hàng không tồn tại” hoặc không thay đổi | Fail |

### c. Quản lý máy

| <b>Chức năng</b> | <b>Mô tả kiểm thử</b>                  | <b>Input</b>  | <b>Expected Output</b>                                | <b>Kết quả (Pass/Fail)</b> |
|------------------|--|---|---|----------------------------|
| Thêm máy         | Thêm máy tính mới với thông tin hợp lệ | Trạng thái “rảnh”, thông số “I9 32GB”, giá “ 50000”, phòng “ Phòng đất” | Máy tính mới được lưu và hiển thị trong danh sách máy | Pass                       |
| Sửa máy          | Cập nhật thông tin máy hợp lệ          | Chọn máy mã “1” sửa tên giá thành “20000”                               | Thông tin máy được cập nhật đúng                      | Pass                       |
| Xóa máy          | Xóa thông tin máy tồn tại              | Chọn máy mã “2”, nhấn Xóa   | Máy tính bị xóa khỏi danh sách                        | Pass                       |

#### d. Quản lý dịch vụ

| <b>Chức năng</b> | <b>Mô tả kiểm thử</b>            | <b>Input</b>                                       | <b>Expected Output</b>  | <b>Kết quả (Pass/Fail)</b> |
|------------------|----------------------------------|--|---|----------------------------|
| Thêm dịch vụ     | Thêm dịch vụ mới hợp lệ          | Tên “Cafe”, đơn giá “50000”                        | Dịch vụ mới được lưu và hiển thị trong danh sách                      | Pass                       |
| Thêm dịch vụ     | Thêm dịch vụ với thông tin thiếu | Tên “ ”, đơn giá “30000” hoặc tên “Mỳ”, đơn giá “” | Hiển thị cảnh báo “Vui lòng nhập tên dv” hoặc “Vui lòng nhập đơn giá” | Fail                       |
| Sửa dịch vụ      | Cập nhật dịch vụ hợp lệ          | Chọn dịch vụ “Cafe”, sửa đơn giá thành “55000”     | Dịch vụ được cập nhật, giá mới hiển thị đúng                          | Pass                       |
| Xóa dịch vụ      | Xóa dịch vụ tồn tại              | Chọn dịch vụ “Pepsi”, nhấn Xóa                     | Dịch vụ bị xóa khỏi danh sách   | Pass                       |

#### e. Thuê máy

| <b>Chức năng</b> | <b>Mô tả kiểm thử</b>                 | <b>Input</b>   | <b>Expected Output</b>                            | <b>Kết quả (Pass/Fail)</b> |
|------------------|---------------------------------------|--|---|----------------------------|
| Thuê máy         | Thuê máy thành công                   | Khách hàng ID=1, Chọn máy "PC03", thời gian: 2 giờ   | máy "PC03" chuyển trạng thái sang Đang sử dụng    | Pass                       |
| Thuê máy         | Thuê máy đã có người sử dụng          | Khách hàng ID=2, Chọn máy "PC03" (đang sử dụng)      | Hiển thị thông báo lỗi “Máy đang được sử dụng”    | Fail                       |
| Thuê máy         | Thuê máy với khách hàng không tồn tại | Khách hàng ID=999, Chọn máy "PC04", thời gian: 1 giờ | Hiển thị thông báo lỗi “Khách hàng không tồn tại” | Fail                       |

## f. Order dịch vụ

| Chức năng     | Mô tả kiểm thử                        | Input   | Expected Output                                   | Kết quả (Pass/Fail) |
|---------------|---------------------------------------|---|---|---------------------|
| Order dịch vụ | Thuê máy thành công                   | Khách hàng ID=1, Chọn máy "PC03", thời gian: 2 giờ    | Máy "PC03" chuyển trạng thái sang Đang sử dụng    | Pass                |
| Thuê máy      | Thuê máy đã có người sử dụng          | Khách hàng ID=2, Chọn máy "PC03" (đang sử dụng)       | Hiển thị thông báo lỗi “Máy đang được sử dụng”    | Fail                |
| Thuê máy      | Thuê máy với khách hàng không tồn tại | Khách hàng ID =999, Chọn máy "PC04", thời gian: 1 giờ | Hiển thị thông báo lỗi “Khách hàng không tồn tại” | Fail                |

## g. Thanh toán

| Chức năng  | Mô tả kiểm thử                          | Input                                      | Expected Output  | Kết quả (Pass/Fail) |
|------------|---|--|--|---------------------|
| Thanh toán | Thanh toán cho ca thuê chỉ có thời gian | Thuê máy ID=2(2 giờ), không order dịch vụ  | Tính tiền đúng (2 giờ * đơn giá giờ), thanh toán                 | Pass                |
| Thanh toán | Thanh toán cho ca thuê có order dịch vụ | Thuê máy ID=2 (2 giờ), đã order dịch vụ x2 | Tính tiền đúng (phí giờ + 2*phí dịch vụ), tạo hóa đơn thanh toán | Pass                |
| Thanh toán | Thanh toán khi không có ca thuê nào     | Chưa thuê máy                              | Hiển thị thông báo “Không có ca thuê để thanh toán”              | Fail                |
| Thanh toán | Thanh toán đã được thực hiện            | Máy ID=2 (đã thanh toán)                   | Thông báo “Đã thanh toán”  | Pass                |

## 9. Triển khai hệ thống

### 9.1 Hướng dẫn cài đặt và chạy chương trình

#### \* Cài đặt môi trường

- Java JDK (phiên bản 8 trở lên).
- Apache Maven.
- IDE hỗ trợ Java như NetBeans, IntelliJ IDEA, hoặc Eclipse.

#### \* Mở dự án

- Giải nén file zip.
- Mở thư mục QuanLyQuanInternet/QuanLyQuanInternet trong IDE.
- Nếu dùng NetBeans thì mở trực tiếp bằng File > Open Project.
- Nếu dùng IntelliJ hoặc Eclipse, import dưới dạng Maven project.

#### \* Chạy chương trình

- Tìm file có phương thức main (thường là lớp giao diện hoặc lớp điều khiển chính) và nhấn **Run**.
- Trong file README.md có thể có thêm hướng dẫn cụ thể về lớp main.

### 9.2 Môi trường triển khai

- Loại triển khai: Ứng dụng chạy local.

## 10. Kết luận và hướng phát triển

### 10.1. Đánh giá kết quả đạt được

Nhóm đã hoàn thành hệ thống “Quản lý quán Internet” với đầy đủ chức năng cơ bản như: quản lý máy trạm, người dùng, thời gian sử dụng và tính tiền tự động. Giao diện đơn giản, dễ thao tác, phù hợp với nhân viên và quản lý. Về kỹ thuật, nhóm áp dụng mô hình MVC, lập trình hướng đối tượng bằng Java và công cụ Maven. Việc sử dụng GitHub cũng giúp nhóm phối hợp hiệu quả.

Nhìn chung, sản phẩm đã đáp ứng được các yêu cầu cốt lõi mà đề tài đặt ra, thể hiện được tính thực tế, ứng dụng, cũng như sự trưởng thành trong tư duy lập trình và làm việc nhóm của các thành viên.

## 10.2. Khó khăn và cách khắc phục

- Thiếu kinh nghiệm xây dựng hệ thống hoàn chỉnh → Nhóm tham khảo phần mềm thực tế để bổ sung kiến thức.
- Làm quen Maven và tổ chức project → Thảo luận nhóm, chia sẻ tài liệu để học nhanh.
- Giao diện còn thô sơ → Sử dụng Swing và cải tiến dựa trên phản hồi người dùng.
- Phân công chưa đồng đều → Điều chỉnh và phân chia lại công việc hợp lý.

## 10.3. Hướng phát triển

- **Nâng cấp giao diện** với JavaFX, thêm chế độ sáng/tối.
- **Tích hợp AI**: gợi ý giờ cao điểm, cảnh báo sử dụng bất thường.
- **Tính năng mới**: thông kê doanh thu, nạp tiền điện tử, thông báo đến máy khách.
- **Hỗ trợ đa nền tảng**: xây dựng bản web/mobile, đồng bộ dữ liệu online.
- **Tăng cường bảo mật**: phân quyền theo cấp, mã hóa dữ liệu người dùng.

## TỔNG KẾT

Đồ án không chỉ là một bài tập học thuật mà còn là cơ hội giúp nhóm tiếp cận gần hơn với thực tế phát triển phần mềm. Qua quá trình thực hiện, nhóm đã cải thiện rõ rệt khả năng phân tích vấn đề, kỹ năng lập trình, làm việc nhóm và đặc biệt là sự linh hoạt trong giải quyết khó khăn. Với những định hướng phát triển đã nêu, nhóm tin rằng hệ thống "Quản lý quán Internet" hoàn toàn có thể trở thành một phần mềm hoàn chỉnh, ứng dụng được trong thực tiễn.

## 11. Tài liệu tham khảo

<https://cloud.z.com/vn/news/mvc>

[aws.amazon.com](https://aws.amazon.com)

<https://www.slideshare.net/slideshow/phn-tch-v-thit-k-h-thng-qun-l-qun-internet/239257042>