

MỤC LỤC

I. GIỚI THIỆU	5
1. Mục tiêu của project	5
2. Phạm vi và giới hạn	5
2.1. Phạm vi của dự án:	5
2.2. Giới hạn của dự án:	5
3. Công nghệ sử dụng	6
II. PHÂN TÍCH YÊU CẦU	6
1. Yêu cầu chức năng	6
2. Yêu cầu phi chức năng	6
3. Use case diagram và mô tả:	7
3.1. Sơ đồ Use Case:	7
3.2. Mô tả:	7
4. Sequence diagram cho các chức năng chính:	9
III. THIẾT KẾ	10
1. Kiến trúc tổng thể	10
1.1. Tầng Giao diện người dùng (Presentation Layer / UI Layer):	10
1.2. Tầng Logic nghiệp vụ (Business Logic Layer / Application Layer):	10
1.3. Tầng Truy cập dữ liệu (Data Access Layer / DAL):	10
1.4. Tầng Cơ sở dữ liệu (Database Layer / Data Store):	11
2. Giao diện người dùng	11
2.1. HotelManagementSystem	11
2.2. Dashboard	12
2.3. Reception	12

2.4. Admin	13
2.5. Login	13
2.6. AddCustomer	13
2.7. CustomerInfor	14
2.8. AddEmployee	15
2.9. EmployeeInfor	15
2.10. ManagerInfor	16
2.11. AddRoom	16
2.12. Room	17
2.13. SearchRoom	17
2.14. UpdateRoom	18
2.15. UpdateCheckin	18
2.16. Checkout	18
2.17. CheckBil	19
2.18. Revenue	19

IV. CÀI ĐẶT 20

1. Cấu trúc source code	20
1.1. HotelManagementSystem	20
1.2. Dashboard	21
1.3. Admin	22
1.4. Login	23
1.5. Reception	24
1.6. AddCustomer	25
1.7. CustomerInfor	25
1.8. AddEmployee	26

1.9. EmployeeInfor	27
1.10. ManagerInfor	28
1.11. AddRoom	29
1.12. Room	29
1.13. SearchRoom	30
1.14. UpdateRoom	31
1.15. UpdateCheckin	32
1.16. Checkout	33
1.17. CheckBill	34
1.18. Revenue	34
1.19. Connect	35
2. Các thuật toán/kỹ thuật quan trọng	36
2.1. Kết nối và thao tác với cơ sở dữ liệu	36
2.2. Tổ chức giao diện và xử lý sự kiện	36
2.3. Các thuật toán quan trọng	37
2.4. Tối ưu hóa dữ liệu và hiệu suất	37
2.5. Bảo mật và xử lý ngoại lệ	38
3. Code mẫu cho các chức năng chính	38
3.1. Kết nối Cơ sở dữ liệu (Connect.java)	38
3.2. Đăng nhập (Admin.java)	39
3.3. Thêm Khách hàng (AddCustomer.java)	40
3.4. Thanh toán và Trả phòng (Checkout.java)	42
3.5. Thống kê Doanh thu (Revenue.java)	43
4. Hướng dẫn build và chạy chương trình	43
4.1. Cài đặt môi trường phát triển	43

4.3. Cấu hình dự án Java	50
4.4. Cài đặt kết nối MySQL trong Java	51
4.5. Build và chạy chương trình	52
4.6. Kiểm tra và debug	52
V. KIỂM THỬ	52
1. Kế hoạch kiểm thử	53
1.1. Giới thiệu	53
1.2. Mục tiêu kiểm thử	53
1.3. Các giai đoạn kiểm thử	53
1.4. Các loại kiểm thử	54
1.5. Môi trường kiểm thử	55
2. Kết quả kiểm thử	56
2.1. Kết quả theo các giai đoạn kiểm thử	56
2.2. Kết quả theo các loại kiểm thử	56
2.3. Tổng hợp lỗi đã phát hiện	58
3. Demo các chức năng	58
VI. KẾT LUẬN	71
1. Kết quả đạt được	71
2. Hạn chế và hướng phát triển	72
3. Bài học kinh nghiệm	73
TÀI LIỆU THAM KHẢO	74

I. GIỚI THIỆU

1. Mục tiêu của project

Dự án này tập trung phát triển một ứng dụng quản lý khách sạn chạy trên máy tính (sử dụng Java). Mục tiêu chính là tự động hóa và đơn giản hóa mọi hoạt động điều hành hàng ngày của khách sạn, từ việc quản lý thông tin khách hàng, thông tin nhân viên đến việc đặt phòng, phân bổ phòng, và ghi nhận chi tiết nhận/trả phòng và xử lý thanh toán. Phần mềm được thiết kế để dễ sử dụng, giúp bất kỳ ai cũng có thể thêm, xóa, cập nhật dữ liệu và quản lý thông tin một cách hiệu quả, chính xác, đồng thời tối ưu hóa việc sử dụng các tiện ích của khách sạn.

2. Phạm vi và giới hạn

2.1. Phạm vi của dự án:

- Phát triển một ứng dụng quản lý khách sạn trên nền tảng máy tính (desktop-based) sử dụng ngôn ngữ Java.
- Tự động hóa và đơn giản hóa các hoạt động điều hành hàng ngày của khách sạn.
- Quản lý các thông tin và quy trình cốt lõi:
 - Thông tin khách hàng, nhân viên, phòng.
 - Đặt phòng và phân bổ phòng.
 - Ghi nhận chi tiết nhận/trả phòng.
 - Xử lý thanh toán.

2.2. Giới hạn của dự án:

- Đây là ứng dụng chạy trên máy tính (desktop), không phải ứng dụng web hay di động.
- Dự án không được thiết kế để phục vụ các chuỗi khách sạn lớn, đa chi nhánh, mà phù hợp hơn với một khách sạn đơn lẻ hoặc quy mô nhỏ.
- Khả năng tích hợp với các hệ thống bên ngoài (ví dụ: các nền tảng đặt phòng trực tuyến, các hệ thống thanh toán phức tạp, phần mềm kế toán chuyên dụng) có thể bị hạn chế.

3. Công nghệ sử dụng

- Hệ điều hành: Windows 11 Home Single Language
- Front End: Java 24, IntelliJ IDEA 2025.1.1.
- Back End: MySQL 8.0.
- JDBC: MySQL Connector/J 8.0.28
- Giao diện đồ họa: Java API Swing.

II. PHÂN TÍCH YÊU CẦU

1. Yêu cầu chức năng

- Quản lý phòng bởi việc theo dõi tình trạng phòng (trống, đã đặt, đang sử dụng) thông qua danh sách phòng.
- Lưu trữ thông tin cá nhân của khách hàng.
- Theo dõi lịch sử lưu trú và đặt phòng của khách hàng thông qua lịch sử các hóa đơn.
- Xử lý các giao dịch thanh toán tại quầy lễ tân.
- Theo dõi doanh thu qua tổng tiền các hóa đơn.
- Lưu trữ và quản lý thông tin của nhân viên.

2. Yêu cầu phi chức năng

- Yêu cầu về bảo mật hệ thống, bảo vệ thông tin khách hàng, thông tin cửa hàng phải được bảo mật về mật khẩu, các thông tin cá nhân như địa chỉ, số điện thoại, email,...
- Hệ thống cần có thời gian phản hồi nhanh, tối ưu hóa trải nghiệm người dùng.
- Đảm bảo chỉ những nhân viên có quyền hạn mới có thể truy cập vào thông tin.
- Hỗ trợ đa nền tảng, hoạt động trên nhiều thiết bị khác nhau.
- Thiết kế giao diện thân thiện, dễ dàng sử dụng cho nhân viên.
- Yêu cầu hệ thống:

- Phải có một bản sao lưu cơ sở dữ liệu của hệ thống quản lý khách sạn.
- Phải có một framework hỗ trợ Java cho hệ thống.
- Hệ điều hành phải là Windows XP hoặc phiên bản Windows cao hơn.

3. Use case diagram và mô tả:

3.1. Sơ đồ Use Case:



Hình 1. Use case diagram

3.2. Mô tả:

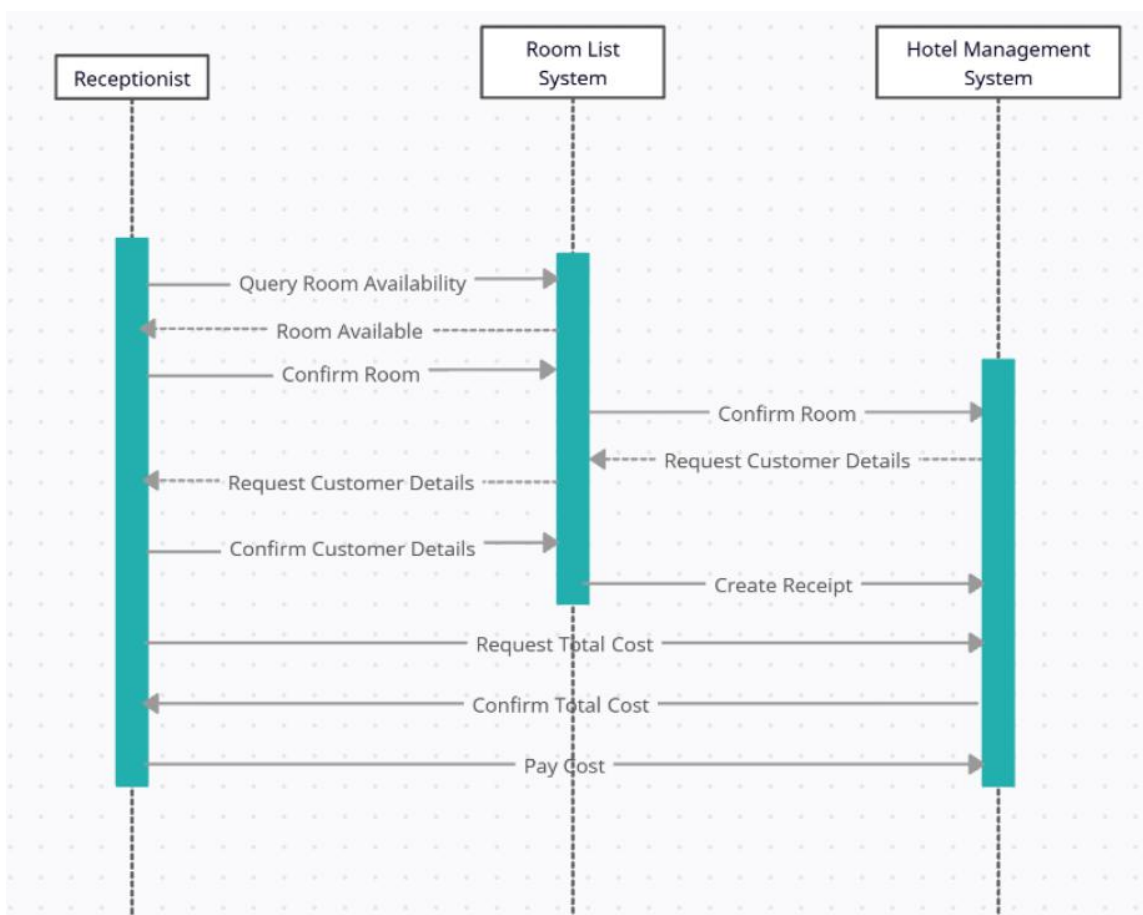
- Các tác nhân (Actors):

- Admin (Quản trị viên): Người quản lý hệ thống, có quyền thực hiện hầu hết các chức năng.
- Lễ tân (Receptionist): Có thể thực hiện một số chức năng nhất định như thêm khách hàng hoặc xem thông tin hóa đơn.

- Các trường hợp sử dụng (Use Cases):

- Add/Update/Delete Customer: Thêm, cập nhật hoặc xóa thông tin khách hàng.
 - Update/Delete Room: Cập nhật hoặc xóa thông tin phòng cụ thể.
 - Check out: Quá trình thanh toán tiền phòng.
 - Watch Bill: Xem thông tin danh sách các hóa đơn.
 - Login: Đăng nhập vào hệ thống.
 - Add Employee: Thêm nhân viên mới vào hệ thống.
 - Add Room: Thêm phòng mới vào hệ thống.
 - Check Revenue: Kiểm tra doanh thu của khách sạn (kiểm tra chi tiết từng hóa đơn, kiểm tra theo tháng, kiểm tra theo ngày).
- **Mối quan hệ:** Các tác nhân tương tác với các trường hợp sử dụng khác nhau, thể hiện quyền hạn và chức năng mà họ có trong hệ thống.

4. Sequence diagram cho các chức năng chính:



Hình 2. Sequence Diagram

III. THIẾT KẾ

1. Kiến trúc tổng thể

- Hệ thống Quản lý Khách sạn (Hotel Management System HMS) thường được thiết kế theo mô hình đa tầng (multi-tier architecture), nhằm phân tách rõ ràng các chức năng và trách nhiệm. Điều này giúp hệ thống dễ phát triển, bảo trì và mở rộng.
- Dưới đây là các thành phần chính trong kiến trúc tổng thể của một HMS:

1.1. Tầng Giao diện người dùng (Presentation Layer / UI Layer):

- Đây là phần mà người dùng (nhân viên lễ tân, quản trị viên) trực tiếp tương tác.
- Bao gồm tất cả các màn hình, nút bấm, bảng hiển thị dữ liệu mà người dùng sử dụng để nhập thông tin, xem kết quả và thực hiện các thao tác.
- Trong dự án, gồm các form sử dụng Java API Swing: AddCustomer, AddEmployee, AddRoom, Admin, CheckBill, Checkout, CustomerInfor, Dashboard, EmployeeInfor, HotelManagementSystem, v.v.).

1.2. Tầng Logic nghiệp vụ (Business Logic Layer / Application Layer):

- Đây là "bộ não" của hệ thống, nơi chứa tất cả các quy tắc nghiệp vụ, quy trình xử lý và các thuật toán.
- Nó nhận yêu cầu từ tầng giao diện người dùng, xử lý logic (ví dụ: kiểm tra phòng trống, tính toán hóa đơn, xác thực người dùng, quản lý nhân viên), và điều phối các tác vụ cần thiết.
- Tầng này không trực tiếp tương tác với cơ sở dữ liệu mà sẽ gọi các phương thức từ tầng truy cập dữ liệu.
- Trong dự án, Dashboard nhận thao tác từ người dùng ở Tầng Giao diện người dùng, thông qua các phương thức xử lý các sự kiện và điều phối đến các tác vụ cần thiết.

1.3. Tầng Truy cập dữ liệu (Data Access Layer / DAL):

- Tầng này có trách nhiệm duy nhất là giao tiếp với cơ sở dữ liệu.

- Nó ẩn đi các chi tiết phức tạp của việc kết nối cơ sở dữ liệu, thực thi các câu lệnh SQL (INSERT, UPDATE, DELETE, SELECT), và ánh xạ dữ liệu từ cơ sở dữ liệu sang các đối tượng hoặc cấu trúc dữ liệu mà tầng logic nghiệp vụ có thể sử dụng.
- Trong dự án, mỗi class chứa các phương thức để tương tác với MySQL, trong đó class Connect có nhiệm vụ tạo môi trường kết nối giữa ứng dụng Java với cơ sở dữ liệu trong MySQL.

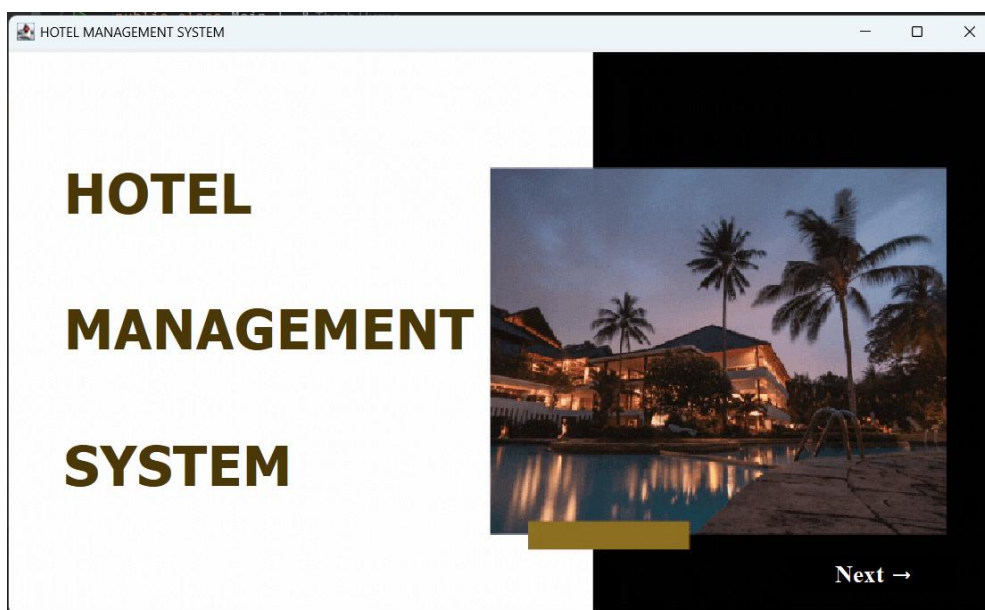
1.4. Tầng Cơ sở dữ liệu (Database Layer / Data Store):

- Đây là nơi lưu trữ tất cả dữ liệu của hệ thống.
- Nó bao gồm các hệ quản trị cơ sở dữ liệu (DBMS) và các cấu trúc dữ liệu vật lý (bảng, quan hệ, chỉ mục, thủ tục lưu trữ, v.v.).
- Trong dự án, cơ sở dữ liệu hotelmanagement trong MySQL có các bảng như HoaDon, KháchHang, Phong, NhanVien, Login, Administrator.

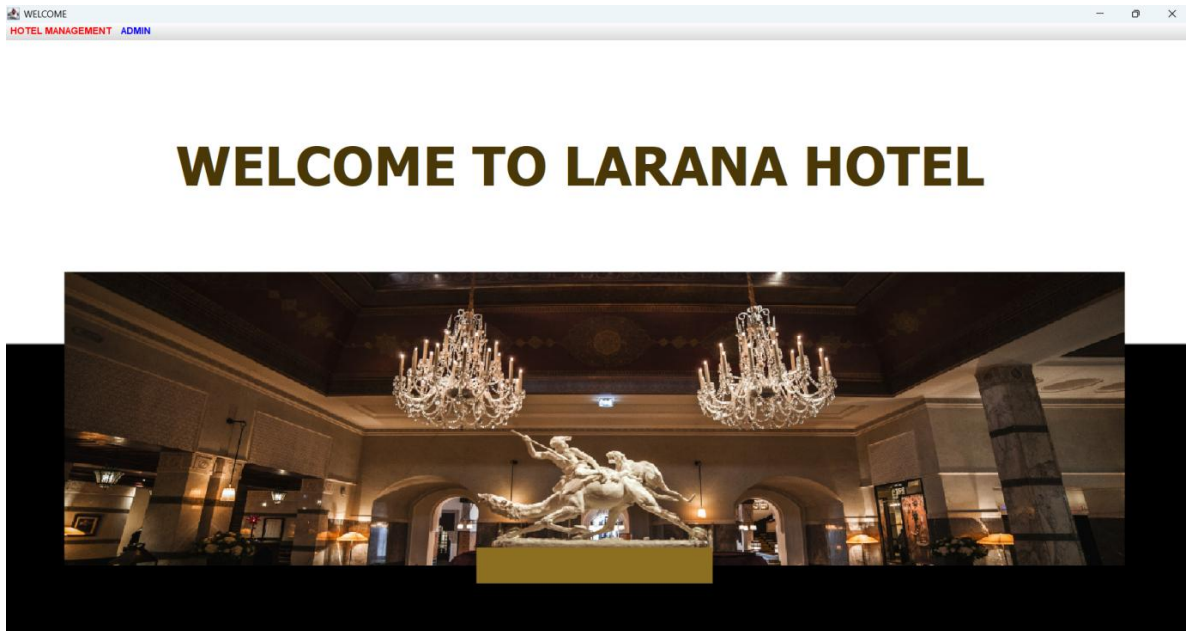
2. Giao diện người dùng

- Các form của dự án sử dụng API Java Swing để xây dựng giao diện đồ họa và Java Database Connectivity (JDBC) để kết nối và tương tác với cơ sở dữ liệu.
- Giao diện gồm 18 form, thực hiện các chức năng khác nhau.

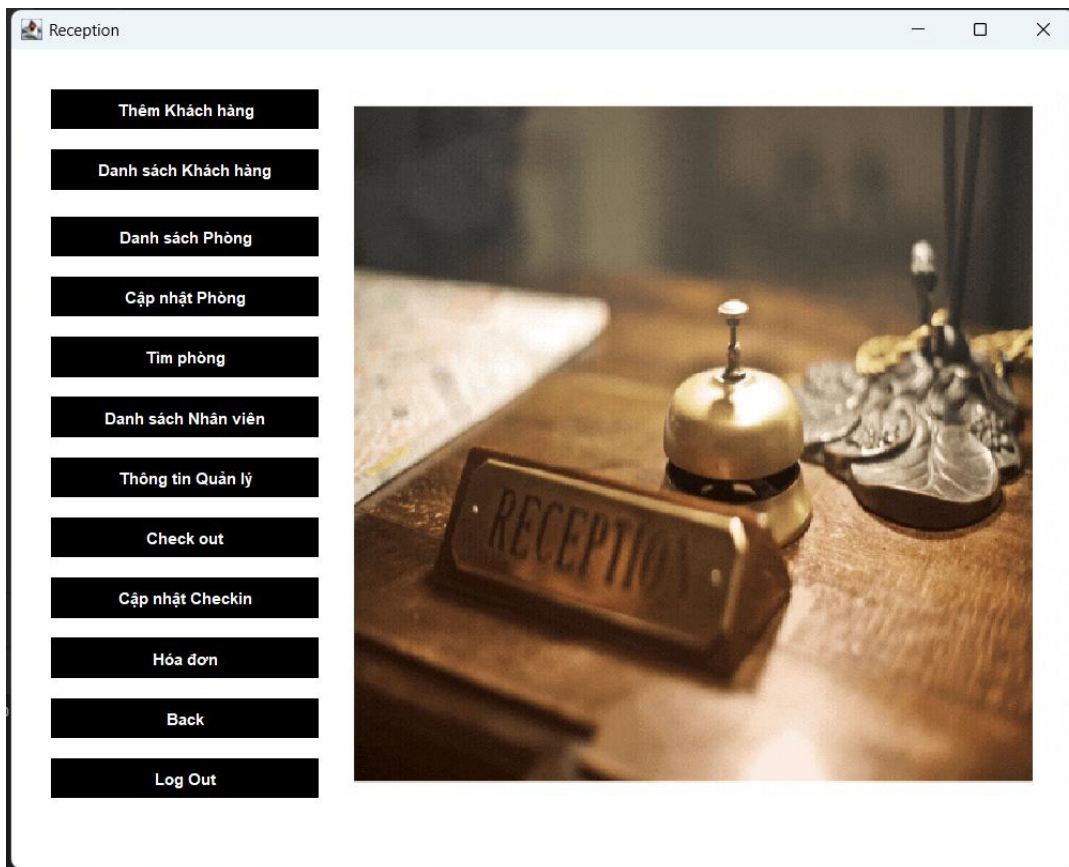
2.1. HotelManagementSystem: Form ban đầu khi mở chương trình.



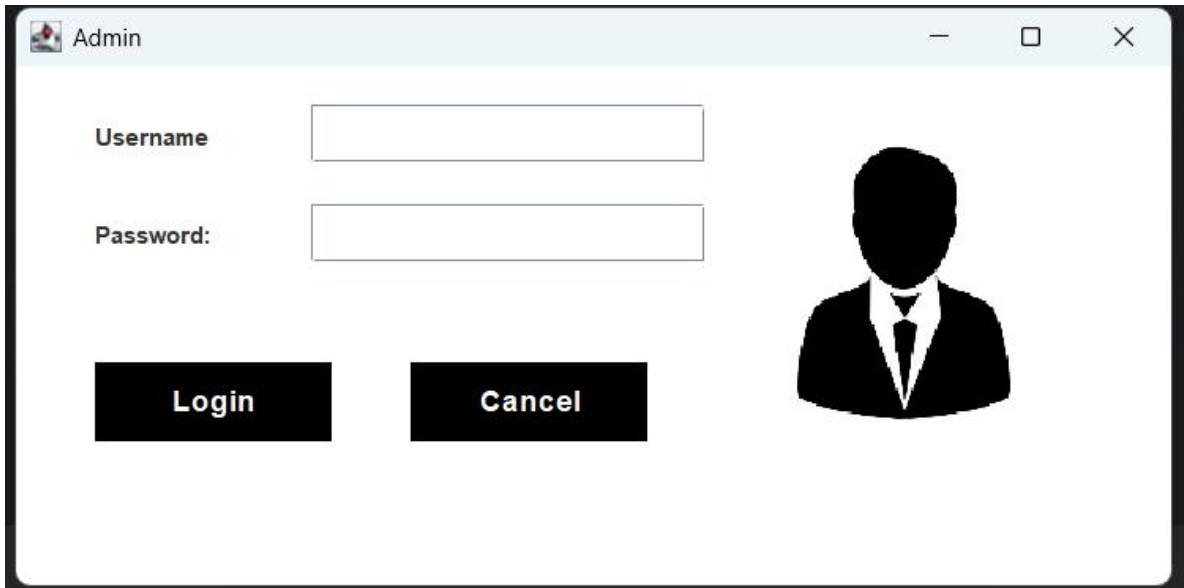
2.2. Dashboard: Form chính sau form HotelManagementSystem.



2.3. Reception: Form hoạt động chính của Lễ tân, bao gồm các button để Lễ tân thao tác và chuyển tiếp đến các tác vụ tương ứng với các button.

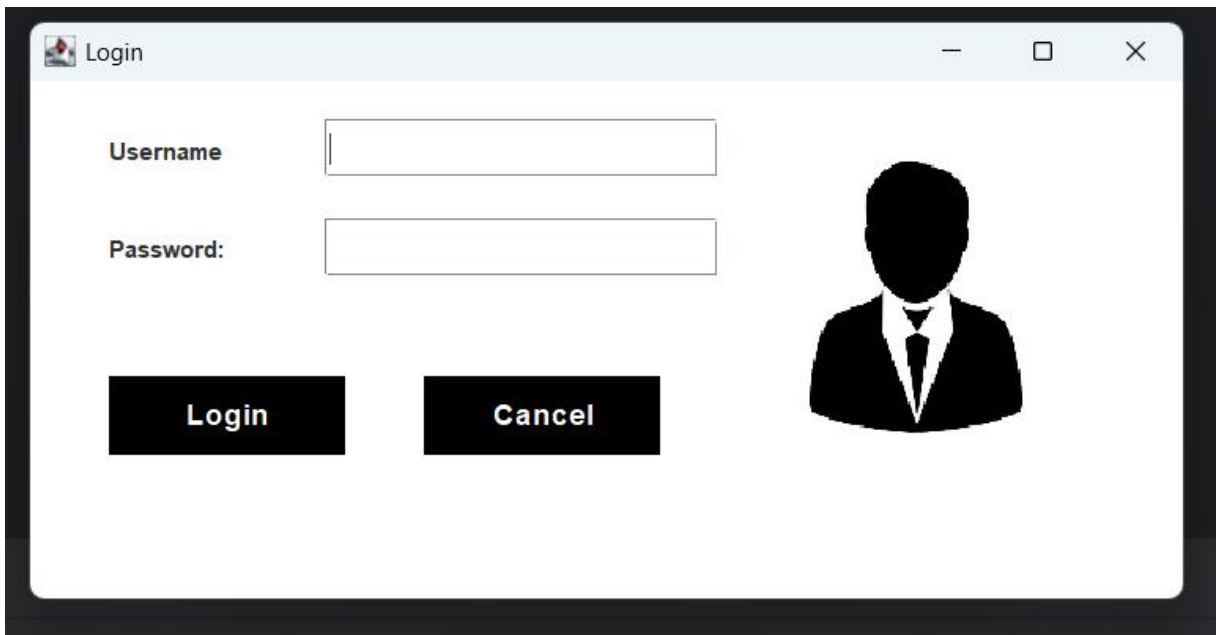


2.4. Admin: Form để đăng nhập vào quyền quản lý của Admin.



A screenshot of a web application window titled "Admin". The window has a light blue header bar with a small icon on the left and standard window controls (minimize, maximize, close) on the right. The main content area is white and contains a login form. On the left, there are two labels: "Username" and "Password:", each followed by a white rectangular input field. Below these fields are two black buttons with white text: "Login" and "Cancel". To the right of the input fields is a black silhouette of a person in a suit and tie.

2.5. Login: Form để Lễ tân đăng nhập vào form Reception.



A screenshot of a web application window titled "Login". The window has a light blue header bar with a small icon on the left and standard window controls (minimize, maximize, close) on the right. The main content area is white and contains a login form. On the left, there are two labels: "Username" and "Password:", each followed by a white rectangular input field. Below these fields are two black buttons with white text: "Login" and "Cancel". To the right of the input fields is a black silhouette of a person in a suit and tie.

2.6. AddCustomer: Form thêm thông tin khách hàng khi check in vào khách sạn.

Add Customer

KHÁCH HÀNG

Số CCCD:

Tên KH:

Giới tính: ☐ Nam ☐ Nữ

Quốc tịch:

Email:

SĐT:

Số phòng:

Ngày vào: 2025/06/02 19:54:53

Đưa trước:

Thêm **Quay lại**

2.7. CustomerInfor: Form để quản lý và lưu trữ thông tin khách hàng.

Customer Information

CCCD	Họ Tên	Giới tính	Quốc tịch	Số điện thoại	Số Phòng	Ngày Nhận	Đưa trước
123456789012	Nguyễn Văn An	Nam	Việt Nam	0912345678	101	2025-05-01T14:00	200000
234567890123	Trần Thị Bình	Nữ	Việt Nam	0923456789	102	2025-05-02T15:00	0
345678901234	Lê Hoàng Cường	Nam	Việt Nam	0934567890	103	2025-05-03T13:30	0
456789012345	Phạm Ngọc Diệp	Nữ	Việt Nam	0945678901	104	2025-05-04T16:00	0
567890123456	Vũ Minh Đức	Nam	Việt Nam	0956789012	105	2025-05-05T14:30	0
678901234567	Hoàng Thị Lan	Nữ	Việt Nam	0967890123	106	2025-05-06T17:00	0
789012345678	Nguyễn Tuấn Hùng	Nam	Việt Nam	0978901234	107	2025-05-07T15:00	0
890123456789	Lý Thị Mai	Nữ	Việt Nam	0989012345	108	2025-05-08T14:00	0
901234567890	Trương Văn Phúc	Nam	Việt Nam	0990123456	109	2025-05-09T16:30	0
012345678901	Đặng Kim Ngân	Nữ	Việt Nam	0901234567	110	2025-05-10T13:00	0

Quay lại **Xóa**

2.8. AddEmployee: Form cho Admin thêm thông tin nhân viên.

Add Employee

Số CCCD

Tên

Tuổi

Giới tính ☐ Nam ☐ Nữ

Chức vụ

Lương

Số điện thoại

Email

Thêm **Quay lại**

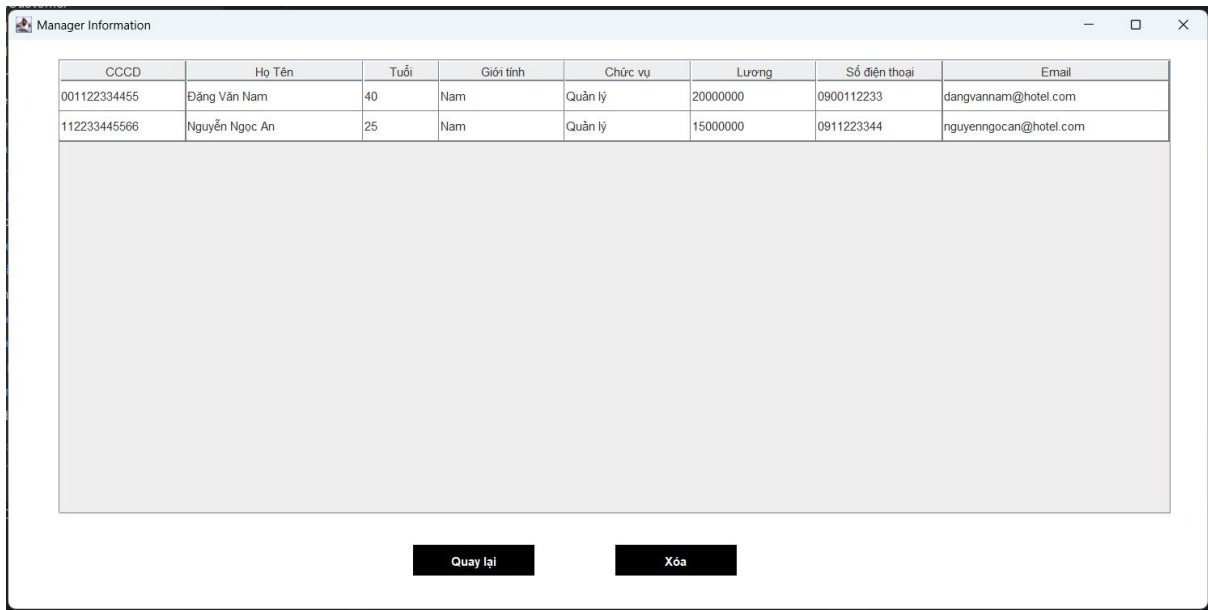
2.9. EmployeeInfor: Form để quản lý và lưu trữ thông tin nhân viên.

Employee Information

CCCD	Họ Tên	Tuổi	Giới tính	Chức vụ	Lương	Số điện thoại	Email
001122334455	Đặng Văn Nam	40	Nam	Quản lý	20000000	0900112233	dangvannam@hotel.com
112233445566	Nguyễn Ngọc An	25	Nam	Quản lý	15000000	0911223344	nguyennngocan@hotel.com
223344556677	Trần Văn Bảo	23	Nam	Lễ tân	12000000	0922334455	tranvanbao@hotel.com
334455667788	Lê Thị Cẩm	25	Nữ	Lễ tân	12000000	0933445566	lethicam@hotel.com
445566778899	Phạm Đức Dũng	27	Nam	Buồng phòng	10000000	0944556677	phamducdung@hotel.com
556677889900	Vũ Thị Em	27	Nữ	Buồng phòng	10000000	0955667788	vuthiem@hotel.com
667788990011	Hoàng Văn Phúc	29	Nam	Đầu bếp	18000000	0966778899	hoangvanphuc@hotel.com
778899001122	Nguyễn Thị Hương	27	Nữ	Đầu bếp	18000000	0977889900	nguyenthihuong@hotel.com
889900112233	Trần Minh Khôi	35	Nam	Bảo vệ	9000000	0988990011	tranminhkhoi@hotel.com
990011223344	Lý Thị Mai	24	Nữ	Kế toán	15000000	0999001122	lythimai@hotel.com

Quay lại **Xóa**

2.10. ManagerInfor: Form để quản lý và lưu trữ thông tin nhân viên có chức vụ là “Quản lý”.

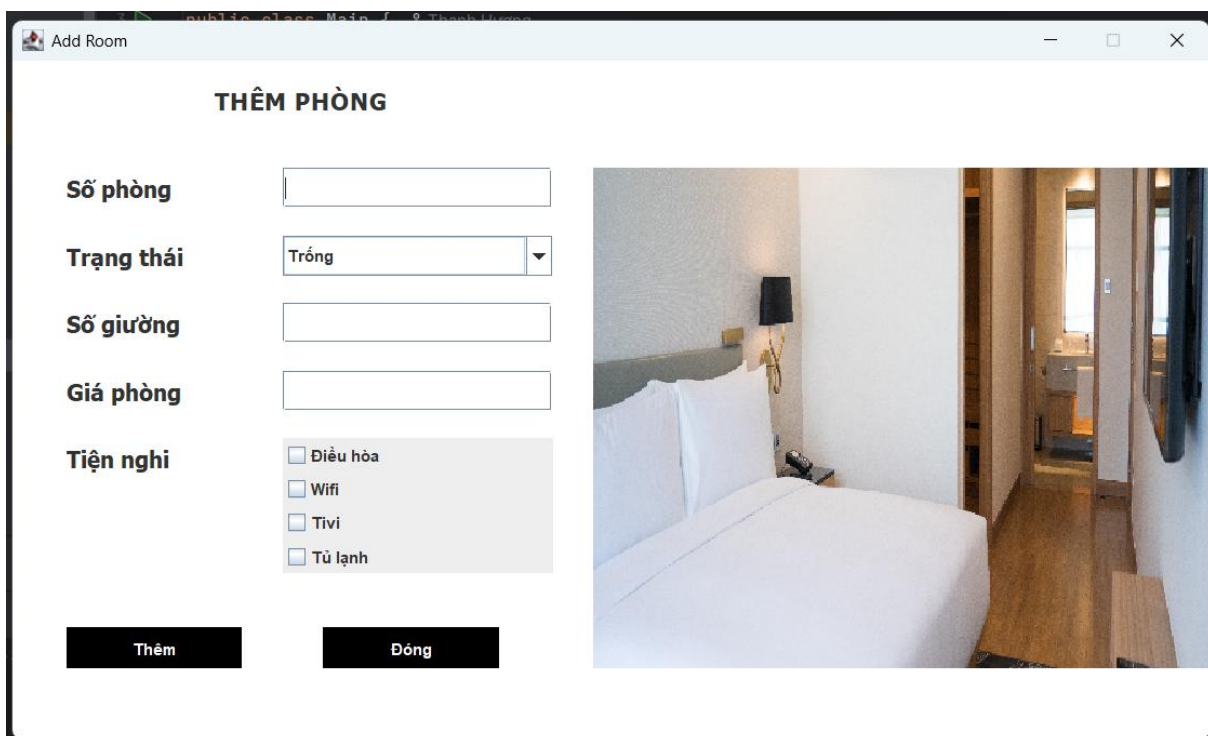


The screenshot shows a web application window titled "Manager Information". It contains a table with the following data:

CCCD	Họ Tên	Tuổi	Giới tính	Chức vụ	Lương	Số điện thoại	Email
001122334455	Đặng Văn Nam	40	Nam	Quản lý	20000000	0900112233	dangvannam@hotel.com
112233445566	Nguyễn Ngọc An	25	Nam	Quản lý	15000000	0911223344	nguyennngocan@hotel.com

Below the table, there are two buttons: "Quay lại" (Back) and "Xóa" (Delete).

2.11. AddRoom: Form cho Admin thêm thông tin cụ thể của 1 phòng.



The screenshot shows a web application window titled "Add Room". The form is titled "THÊM PHÒNG" (Add Room). It contains the following fields and options:

- Số phòng** (Room Number): Text input field.
- Trạng thái** (Status): Dropdown menu with "Trống" (Empty) selected.
- Số giường** (Number of Beds): Text input field.
- Giá phòng** (Room Price): Text input field.
- Tiện nghi** (Amenities): Checkboxes for "Điều hòa" (Air Conditioning), "Wifi", "Tivi" (TV), and "Tủ lạnh" (Refrigerator).

At the bottom, there are two buttons: "Thêm" (Add) and "Đóng" (Close). To the right of the form is a photograph of a hotel room with a bed, a desk, and a bathroom.

2.12. Room: Form chứa danh sách phòng, bao gồm số phòng, số giường, tiện nghi của phòng, trạng thái phòng (trống, đã đặt, đang dọn dẹp), và giá mỗi phòng.

Số Phòng	Số giường	Tiện nghi	Trạng thái	Giá mặc định
101	2	Điều hòa, Wifi, Tivi	Đã đặt	500000
102	1	Điều hòa, Wifi	Trống	300000
103	3	Điều hòa, Wifi, Tivi, Tủ lạnh	Trống	700000
104	2	Wifi, Tivi	Đã đặt	600000
105	1	Điều hòa	Trống	400000
106	2	Điều hòa, Wifi	Trống	450000
107	3	Điều hòa, Wifi, Tivi	Đã đặt	750000
108	1	Wifi	Đã đặt	350000
109	2	Điều hòa, Tủ lạnh	Đã đặt	550000
110	1	Điều hòa, Wifi, Tivi	Trống	600000

Quay lại Xóa

2.13. SearchRoom: Form cho phép Lễ tân có thể tìm phòng theo số giường và trạng thái của phòng.

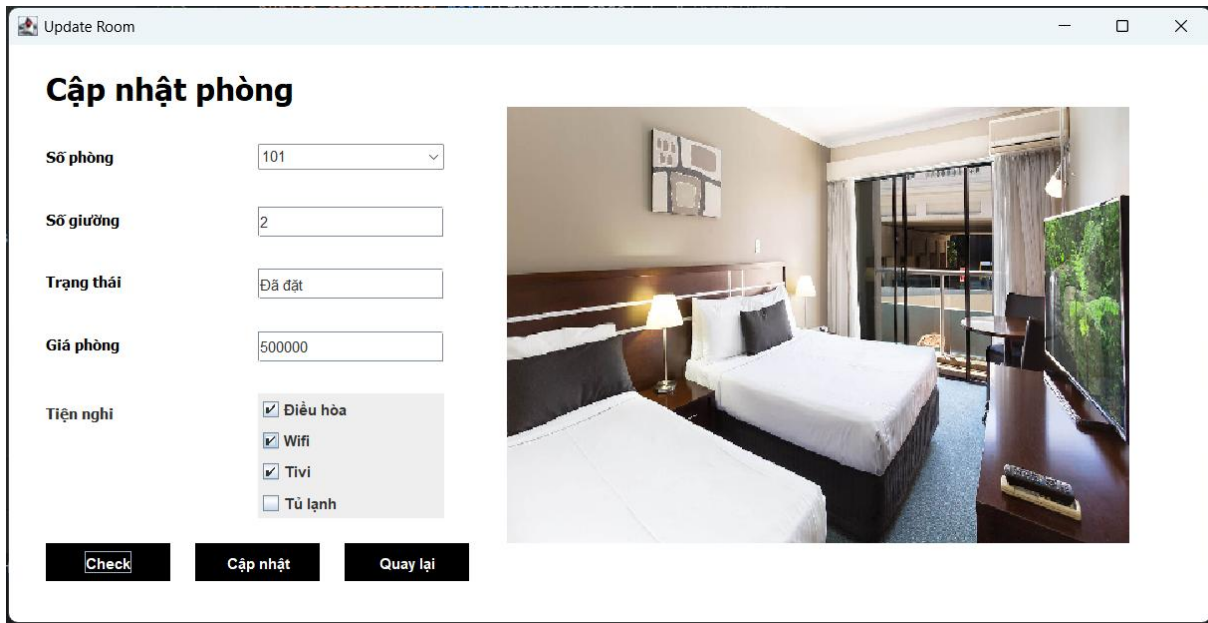
TÌM PHÒNG

Số giường ☐ Chỉ hiển thị phòng Trống

Số Phòng	Số giường	Tiện nghi	Trạng thái	Giá mặc định
101	2	Điều hòa, Wifi, Tivi	Đã đặt	500000
102	1	Điều hòa, Wifi	Trống	300000
103	3	Điều hòa, Wifi, Tivi, Tủ lạnh	Trống	700000
104	2	Wifi, Tivi	Đã đặt	600000
105	1	Điều hòa	Trống	400000
106	2	Điều hòa, Wifi	Trống	450000
107	3	Điều hòa, Wifi, Tivi	Đã đặt	750000
108	1	Wifi	Đã đặt	350000
109	2	Điều hòa, Tủ lạnh	Đã đặt	550000

Tìm Quay lại Reset

2.14. UpdateRoom: Form cho phép Lễ tân cập nhật lại thông tin cụ thể của một phòng.



Cập nhật phòng

Số phòng: 101

Số giường: 2

Trạng thái: Đã đặt

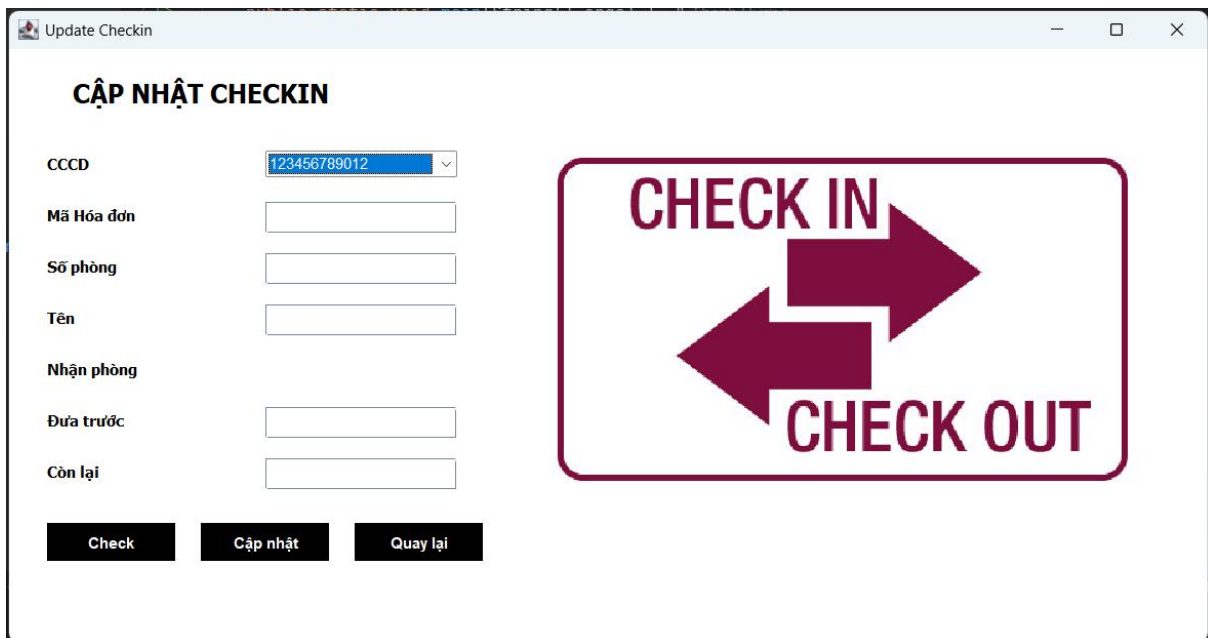
Giá phòng: 500000

Tiện nghi:

- ☒ Điều hòa
- ☒ Wifi
- ☒ Tivi
- ☐ Tủ lạnh

Check **Cập nhật** **Quay lại**

2.15. UpdateCheckin: Form này cho phép Lễ tân cập nhật lại thông tin Checkin của khách hàng, bao gồm số phòng, tên khách hàng, số tiền đưa trước và hiển thị số tiền còn lại phải trả.



CẬP NHẬT CHECKIN

CCCD: 123456789012

Mã Hóa đơn:

Số phòng:

Tên:

Nhận phòng:

Đưa trước:

Còn lại:

Check **Cập nhật** **Quay lại**

CHECK IN

CHECK OUT

2.16. Checkout: Form để Lễ tân thực hiện việc thanh toán và xử lý hóa đơn.

Checkout

CHECKOUT

Số CCCD

567890123456

✓

Số phòng

105

Nhận phòng

2025-05-05 14:30:00

Trả phòng

2025/06/02 19:58:06

Checkout

Quay lại

2.17. CheckBill: Form để Lễ tân xem thông tin các hóa đơn của khách sạn như thông tin thuê phòng của khách hàng, trạng thái hóa đơn (bao gồm cả hóa đơn chưa thanh toán và đã thanh toán) và tổng tiền mỗi hóa đơn.

Statistics Bill

Mã Hóa đơn	CCCD	Số Phòng	Ngày Nhận	Ngày Trả	Trạng Thái	Tổng Tiền
HD001	123456789012	101	2025-05-01T14:00	2025-05-03T12:00	Chưa thanh toán	1,000,000
HD002	234567890123	102	2025-05-02T15:00	2025-05-04T11:00	Đã thanh toán	800,000
HD003	345678901234	103	2025-05-03T13:30	2025-05-05T10:30	Chưa thanh toán	1,400,000
HD004	456789012345	104	2025-05-04T16:00	2025-05-06T12:00	Đã thanh toán	1,200,000
HD005	567890123456	105	2025-05-05T14:30	2025-05-07T11:30	Chưa thanh toán	800,000
HD006	678901234567	106	2025-05-06T17:00	2025-05-08T12:30	Đã thanh toán	900,000
HD007	789012345678	107	2025-05-07T15:00	2025-05-09T10:00	Chưa thanh toán	1,500,000
HD008	890123456789	108	2025-05-08T14:00	2025-05-10T12:00	Đã thanh toán	700,000
HD009	901234567890	109	2025-05-09T16:30	2025-05-11T11:00	Chưa thanh toán	1,100,000
HD010	012345678901	110	2025-05-10T13:00	2025-05-12T10:30	Đã thanh toán	1,200,000

Quay lại

2.18. Revenue: Form thông tin các hóa đơn dành cho Admin, Admin có quyền xem doanh thu của mỗi Hóa đơn, hoặc tổng doanh thu các hóa đơn theo ngày, hoặc tổng doanh thu các hóa đơn theo tháng.

Statistics Revenue

Thống kê theo: Tất cả Hóa đơn

Mã Hóa đơn	CCCD	Số Phòng	Ngày Nhận	Ngày Trả	Tổng Tiền
HD001	123456789012	101	2025-05-01T14:00	2025-05-03T12:00	1,000,000
HD002	234567890123	102	2025-05-02T15:00	2025-05-04T11:00	600,000
HD003	345678901234	103	2025-05-03T13:30	2025-05-05T10:30	1,400,000
HD004	456789012345	104	2025-05-04T16:00	2025-05-06T12:00	1,200,000
HD005	567890123456	105	2025-05-05T14:30	2025-05-07T11:30	800,000
HD006	678901234567	106	2025-05-06T17:00	2025-05-08T12:30	900,000
HD007	789012345678	107	2025-05-07T15:00	2025-05-09T10:00	1,500,000
HD008	890123456789	108	2025-05-08T14:00	2025-05-10T12:00	700,000
HD009	901234567890	109	2025-05-09T16:30	2025-05-11T11:00	1,100,000

Quay lại

IV. CÀI ĐẶT

1. Cấu trúc source code

Gồm 2 packages:

- Package hotelmanagementsystem chứa các class
- Package icon chứa các hình ảnh cần thiết cho GUI

Và class Main để chạy chương trình.

Cụ thể tổ chức các class trong package hotelmanagementsystem:

1.1. HotelManagementSystem: Lớp này extends JFrame và implements Interface ActionListener.

1.1.1. Tổ chức:

- Các trường nhập (JTextField) để thu thập thông tin khách hàng như tên, CCCD, quốc tịch, số điện thoại, email, tiền trả trước.
- Các nút chọn (JRadioButton) để chọn giới tính của khách hàng.
- Một danh sách lựa chọn (Choice) để hiển thị các số phòng đang trống.
- Nhãn (JLabel) hiển thị thời gian khách hàng check-in.
- Hai nút (JButton) để thực hiện chức năng "Thêm" và "Quay lại".

1.1.2. Chức năng:

- Khởi tạo giao diện (Constructor)
 - Thiết lập bố cục, màu sắc, kích thước cửa sổ.
 - Tạo các thành phần giao diện như nhãn, ô nhập, nút chọn.
 - Kết nối với cơ sở dữ liệu để lấy danh sách các phòng trống.
- Xử lý sự kiện (actionPerformed)
 - Khi nhấn "Thêm":
 - Kiểm tra xem số CCCD đã tồn tại trong cơ sở dữ liệu hay chưa.
 - Nếu đã tồn tại, cập nhật thông tin khách hàng.
 - Nếu chưa tồn tại, thêm khách hàng mới vào cơ sở dữ liệu.
 - Cập nhật trạng thái phòng thành "Đã đặt".
 - Tạo hóa đơn thanh toán cho khách hàng.
 - Khi nhấn "Quay lại":
 - Đóng cửa sổ hiện tại và mở giao diện Reception

1.2. Dashboard: Lớp này extends JFrame và implements Interface ActionListener.

1.2.1. Tổ chức:

- Tiêu đề chào mừng (JLabel): Hiển thị dòng chữ "WELCOME TO LARANA HOTEL".
- Hình nền (JLabel với ImageIcon): Hiển thị ảnh nền khách sạn.
- Thanh Menu (JMenuBar): gồm 2 Menu con:
 - HOTEL MANAGEMENT: Chứa menu con RECEPTION.
 - ADMIN: Chứa các menu con ADD EMPLOYEE, ADD ROOM, REVENUE.

1.2.2. Chức năng:

- Khởi tạo giao diện (Constructor):

- Thiết lập cửa sổ ở chế độ toàn màn hình (MAXIMIZED_BOTH).
- Thêm tiêu đề, hình nền, thanh menu và các thành phần giao diện khác.
- Xử lý sự kiện (actionPerformed): Khi người dùng chọn một mục từ menu, chương trình mở một cửa sổ tương ứng:
 - "ADD EMPLOYEE" → Mở giao diện Admin("ADD EMPLOYEE"), là giao diện đăng nhập của admin để dẫn đến giao diện AddEmployee.
 - "ADD ROOM" → Mở giao diện Admin("ADD ROOM"), là giao diện đăng nhập của admin để dẫn đến giao diện AddRoom.
 - "REVENUE" → Mở giao diện Admin("REVENUE"), là giao diện đăng nhập của admin để dẫn đến giao diện Revenue.
 - "RECEPTION" → Mở giao diện Login() của Lễ tân.

1.3. Admin: Lớp này extends JFrame và implements Interface ActionListener.

1.3.1. Tổ chức:

- Trường nhập (JTextField): Nhập tên người dùng.
- Trường nhập mật khẩu (JPasswordField): Nhập mật khẩu.
- Các nút (JButton):
 - "Login": Kiểm tra thông tin đăng nhập.
 - "Cancel": Quay lại Dashboard.

1.3.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu nền, bố cục.
 - Hiện thị trường nhập tài khoản, mật khẩu.
 - Nhận tham số nextForm để quyết định mở giao diện tiếp theo sau khi đăng nhập.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn Login:

- Kiểm tra tên người dùng và mật khẩu từ cơ sở dữ liệu (SELECT * FROM Administrator WHERE username = ? AND password = ?).
- Nếu đúng, chuyển sang giao diện quản lý tương ứng (AddEmployee, AddRoom, Revenue hoặc Dashboard).
- Nếu sai, hiển thị thông báo lỗi.
- Khi nhấn Cancel, quay lại Dashboard.
- Hỗ trợ phím Enter (KeyListener): Cho phép nhấn Enter để thực hiện đăng nhập.

1.4. Login: Lớp này extends JFrame và implements Interface ActionListener.

1.4.1. Tổ chức:

- Trường nhập (JTextField): Nhập tên người dùng.
- Trường nhập mật khẩu (JPasswordField): Nhập mật khẩu.
- Hai nút (JButton):
 - "Login": Kiểm tra thông tin đăng nhập.
 - "Cancel": Quay lại Dashboard

1.4.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu nền, bố cục.
 - Hiển thị trường nhập tài khoản, mật khẩu.
 - Thêm phím Enter để người dùng nhập nhanh mà không cần click chuột.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn "Login":
 - Kiểm tra tên người dùng và mật khẩu từ cơ sở dữ liệu (SELECT * FROM login WHERE username = ? AND password = ?).
 - Nếu đúng, chuyển sang Reception.
 - Nếu sai, hiển thị thông báo lỗi và xóa thông tin nhập.

- Khi nhấn "Cancel", quay lại Dashboard.

1.5. Reception: Lớp này extends JFrame và implements Interface ActionListener.

1.5.1. Tổ chức: Gồm danh sách nút điều hướng đến các chức năng:

- Thêm khách hàng (newCustomer)
- Xem danh sách phòng (rooms)
- Quản lý nhân viên (allEmployees)
- Quản lý khách hàng (customers)
- Xem thông tin quản lý (manager)
- Tìm phòng (searchRoom)
- Cập nhật phòng (roomStatus)
- Cập nhật trạng thái check-in (update)
- Xem hóa đơn (bill)
- Thanh toán (checkout)
- Quay lại Dashboard
- Đăng xuất hệ thống (logOut)

1.5.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu sắc.
 - Hiển thị các nút điều hướng.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn một nút, chương trình đóng giao diện hiện tại và mở lớp tương ứng (AddCustomer, Room, EmployeeInfor, Checkout, v.v.).
 - Khi nhấn "Quay lại", trở về Dashboard.
 - Khi nhấn "Đăng xuất", quay lại màn hình chính của hệ thống.

1.6. AddCustomer: Lớp này extends JFrame và implements Interface ActionListener.

1.6.1. Tổ chức:

- Các trường nhập (JTextField) để thu thập thông tin như tên, CCCD, quốc tịch, điện thoại, email, đặt cọc.
- Hai nút chọn (JRadioButton) để xác định giới tính.
- Danh sách lựa chọn (Choice) để chọn số phòng trống từ cơ sở dữ liệu.
- Nhãn (JLabel) hiển thị thời gian check-in.
- Hai nút (JButton) "Thêm" và "Quay lại".

1.6.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu sắc, bố cục.
 - Hiển thị tiêu đề, các trường nhập liệu và ảnh minh họa.
 - Kết nối cơ sở dữ liệu để lấy danh sách phòng trống.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn nút "Thêm":
 - Kiểm tra CCCD:
 1. Nếu CCCD đã tồn tại, cập nhật thông tin khách hàng.
 2. Nếu CCCD chưa tồn tại, thêm khách hàng mới vào hệ thống.
 - Cập nhật trạng thái phòng thành "Đã đặt".
 - Thêm hóa đơn mới cho khách hàng.
 - Khi nhấn nút "Quay lại", chương trình chuyển về màn hình Reception.

1.7. CustomerInfor: Lớp này extends JFrame và implements Interface ActionListener.

1.7.1. Tổ chức:

- Bảng dữ liệu (JTable): Hiển thị thông tin khách hàng từ cơ sở dữ liệu.

- Thanh cuộn (JScrollPane): Giúp xem danh sách nếu quá dài.
- Hai nút chức năng (JButton):
 - "Quay lại": Quay về giao diện Reception.
 - "Xóa": Xóa thông tin khách hàng khỏi hệ thống.

1.7.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu nền và kích thước bảng.
 - Kết nối với cơ sở dữ liệu để lấy danh sách khách hàng (ResultSet).
 - Hiển thị dữ liệu với thư viện DbUtils.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn nút "Quay lại", chương trình đóng cửa sổ hiện tại và mở Reception.
 - Khi nhấn nút "Xóa":
 - Kiểm tra xem khách hàng đã được chọn chưa.
 - Hiện thị hộp thoại xác nhận trước khi xóa.
 - Xóa khách hàng khỏi cơ sở dữ liệu và cập nhật giao diện bảng.

1.8. AddEmployee: Lớp này extends JFrame và implements ActionListener.

1.8.1. Tổ chức:

- Trường nhập (JTextField): Tên, CCCD, tuổi, lương, số điện thoại, email.
- Nút chọn (JRadioButton): Xác định giới tính (Nam/Nữ).
- Danh sách chọn (JComboBox): Chọn chức vụ nhân viên.
- Hai nút (JButton):
 - "Thêm": Xác nhận và lưu thông tin nhân viên vào cơ sở dữ liệu.
 - "Quay lại": Quay về giao diện Dashboard.

1.8.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu nền, bố cục.
 - Hiện thị các trường nhập liệu và hình ảnh nhân viên.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn nút "Thêm":
 - Kiểm tra đầu vào (Tên không được trống, email hợp lệ).
 - Lưu nhân viên vào cơ sở dữ liệu (insert into NhanVien).
 - Hiện thị thông báo thành công.
 - Khi nhấn "Quay lại": Quay về màn hình Dashboard.

1.9. EmployeeInfor: Lớp này extends JFrame và implements Interface ActionListener.

1.9.1. Tổ chức:

- Bảng (JTable): Hiện thị danh sách nhân viên với các thông tin như CCCD, tên, tuổi, giới tính, chức vụ, lương, số điện thoại, email.
- Thanh cuộn (JScrollPane): Giúp xem danh sách dài hơn.
- Hai nút chức năng (JButton):
 - "Quay lại": Quay về giao diện Reception.
 - "Xóa": Xóa nhân viên khỏi hệ thống.

1.9.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu nền, kích thước bảng.
 - Kết nối với cơ sở dữ liệu để lấy danh sách nhân viên (ResultSet).
 - Hiện thị dữ liệu bằng thư viện DbUtils.
- Xử lý sự kiện (actionPerformed):

- Khi nhấn nút "Quay lại", chương trình đóng cửa sổ hiện tại và mở Reception.

- Khi nhấn nút "Xóa":

- Kiểm tra xem nhân viên đã được chọn chưa.
- Hiện thị hộp thoại xác nhận trước khi xóa.
- Xóa nhân viên khỏi cơ sở dữ liệu và cập nhật bảng.

1.10. ManagerInfor: Lớp này extends JFrame và implements Interface ActionListener.

1.10.1. Tổ chức:

- Bảng (JTable): Hiện thị danh sách các quản lý, bao gồm CCCD, tên, tuổi, giới tính, chức vụ, lương, số điện thoại, email.
- Thanh cuộn (JScrollPane): Giúp xem danh sách nếu dài.
- Hai nút chức năng (JButton):
 - "Quay lại": Quay về giao diện Reception.
 - "Xóa": Xóa quản lý khỏi hệ thống.

1.10.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu nền, bảng.
 - Kết nối cơ sở dữ liệu để lấy danh sách các quản lý (ResultSet).
 - Hiện thị dữ liệu bằng thư viện DbUtils.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn nút "Quay lại", chương trình đóng cửa sổ hiện tại và mở Reception.
 - Khi nhấn "Xóa":
 - Kiểm tra xem quản lý đã được chọn chưa.

- Hiển thị hộp thoại xác nhận.
- Xóa quản lý khỏi cơ sở dữ liệu và cập nhật bảng.

1.11. AddRoom: Lớp này extends JFrame và implements Interface ActionListener.

1.11.1. Tổ chức:

- Trường nhập (JTextField): Số phòng, số giường, giá phòng.
- Danh sách chọn (JComboBox): Trạng thái phòng (Trống, Đã đặt, Đang dọn dẹp).
- Các ô chọn (JCheckBox): Tiện nghi như Điều hòa, WiFi, Tivi, Tủ lạnh.
- Hai nút (JButton):
 - "Thêm": Lưu phòng vào hệ thống.
 - "Đóng": Quay về Dashboard.

1.11.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập bố cục, màu sắc, hình ảnh minh họa.
 - Tạo các trường nhập và danh sách chọn.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn nút "Thêm":
 - Thu thập dữ liệu từ giao diện.
 - Lưu phòng vào cơ sở dữ liệu (INSERT INTO Phong).
 - Hiển thị thông báo thành công.
 - Khi nhấn "Đóng": Quay về màn hình Dashboard.

1.12. Room: Lớp này extends JFrame và implements Interface ActionListener.

1.12.1. Tổ chức:

- Bảng (JTable): Hiển thị thông tin phòng gồm số phòng, số giường, tiện nghi, trạng thái, giá.
- Thanh cuộn (JScrollPane): Giúp xem danh sách nếu dài.

- Hai nút chức năng (JButton):
 - "Quay lại": Quay về giao diện Reception.
 - "Xóa": Xóa phòng khỏi hệ thống.

1.12.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu nền.
 - Hiển thị hình ảnh minh họa của khách sạn.
 - Kết nối cơ sở dữ liệu để lấy danh sách phòng (ResultSet).
 - Hiển thị dữ liệu bằng thư viện DbUtils.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn nút "Quay lại", chương trình đóng cửa sổ hiện tại và mở Reception.
 - Khi nhấn "Xóa":
 - Kiểm tra xem phòng đã được chọn chưa.
 - Hiển thị hộp thoại xác nhận.
 - Xóa phòng khỏi cơ sở dữ liệu và cập nhật bảng.

1.13. SearchRoom: Lớp này extends JFrame và implements ActionListener.

1.13.1. Tổ chức:

- Trường nhập (JTextField): Nhập số giường để tìm kiếm phòng.
- Checkbox (JCheckBox): Chọn chỉ hiển thị phòng Trống.
- Bảng (JTable): Hiển thị kết quả tìm kiếm gồm số phòng, số giường, tiện nghi, trạng thái, giá.
- Ba nút (JButton):
 - "Tìm": Thực hiện tìm kiếm.
 - "Reset": Xóa dữ liệu nhập và hiển thị toàn bộ danh sách phòng.

- "Quay lại": Quay về giao diện Reception.

1.13.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu sắc.
 - Hiển thị các trường nhập liệu, bảng danh sách phòng.
 - Tải dữ liệu phòng từ cơ sở dữ liệu.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn "Tìm":
 - Xây dựng truy vấn SQL dựa trên dữ liệu nhập.
 - Lọc phòng theo số giường và trạng thái (Trống) nếu được chọn.
 - Hiển thị kết quả lên bảng.
 - Khi nhấn "Reset", xóa dữ liệu nhập và hiển thị toàn bộ phòng.
 - Khi nhấn "Quay lại", quay về Reception.

1.14. UpdateRoom: Lớp này extends JFrame và implements Interface ActionListener.

1.14.1. Tổ chức:

- Danh sách chọn (Choice): Hiển thị số phòng có trong hệ thống.
- Trường nhập (JTextField): Cập nhật trạng thái phòng, số giường, giá.
- Danh sách tiện nghi (JCheckBox): Điều hòa, Wifi, Tivi, Tủ lạnh.
- Ba nút (JButton):
 - "Check": Hiển thị thông tin phòng hiện tại.
 - "Cập nhật": Lưu thông tin đã chỉnh sửa.
 - "Quay lại": Quay về Reception.

1.14.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu nền.

- Tải danh sách số phòng từ cơ sở dữ liệu.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn "Check":
 - Lấy thông tin phòng từ cơ sở dữ liệu.
 - Hiện thị trạng thái, số giường, giá.
 - Hiện thị tiện nghi đã có.
 - Khi nhấn "Cập nhật":
 - Thu thập dữ liệu mới từ giao diện.
 - Cập nhật thông tin phòng vào cơ sở dữ liệu (UPDATE Phong).
 - Hiện thị thông báo thành công.
 - Khi nhấn "Quay lại", quay về Reception.

1.15. UpdateCheckin: Lớp này extends JFrame và implements Interface ActionListener.

1.15.1. Tổ chức:

- Danh sách chọn (Choice): Hiện thị danh sách khách hàng chưa thanh toán.
- Trường nhập (JTextField): Hiện thị và cho phép cập nhật số phòng, tên khách hàng, tiền đặt cọc, số tiền còn lại.
- Các nhãn (JLabel): Hiện thị ngày nhận phòng.
- Ba nút (JButton):
 - "Check": Kiểm tra thông tin khách hàng.
 - "Cập nhật": Lưu thông tin cập nhật.
 - "Quay lại": Trở về Reception.

1.15.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu nền.

- Hiển thị danh sách khách hàng chưa thanh toán.
- Kết nối cơ sở dữ liệu để tải thông tin khách hàng.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn "Check":
 - Lấy thông tin khách hàng từ cơ sở dữ liệu.
 - Hiển thị số phòng, tên, tiền đặt cọc.
 - Tính số tiền còn lại dựa trên giá phòng.
 - Khi nhấn "Cập nhật":
 - Lưu thông tin cập nhật vào cơ sở dữ liệu (UPDATE KhachHang, UPDATE HoaDon).
 - Hiển thị thông báo thành công.
 - Khi nhấn "Quay lại", quay về Reception.

1.16. Checkout: Lớp này extends JFrame và implements Interface ActionListener.

1.16.1. Tổ chức:

- Danh sách chọn (Choice): Hiển thị danh sách khách hàng chưa thanh toán.
- Các nhãn (JLabel): Hiển thị số phòng, ngày nhận phòng, ngày trả phòng.
- Hai nút (JButton):
 - "Checkout": Cập nhật trạng thái phòng và hóa đơn.
 - "Quay lại": Trở về Reception.

1.16.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu nền.
 - Hiển thị các trường thông tin và tải danh sách khách hàng.
- Xử lý sự kiện (actionPerformed):
 - Khi nhấn "Checkout":

- Cập nhật trạng thái phòng thành Trống.
- Ghi lại thời gian trả phòng vào hóa đơn (UPDATE HoaDon).
- Hiện thị thông báo thành công.
- Khi nhấn "Quay lại", quay về Reception.

1.17. CheckBill: Lớp này extends JFrame và implements Interface ActionListener.

1.17.1. Tổ chức:

- Bảng (JTable): Hiện thị danh sách hóa đơn gồm mã hóa đơn, CCCD, số phòng, ngày nhận, ngày trả, trạng thái thanh toán, tổng tiền.
- Thanh cuộn (JScrollPane): Giúp xem danh sách dài hơn.
- Nút "Quay lại" (JButton): Trở về Reception.

1.17.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu nền.
 - Hiện thị bảng danh sách hóa đơn.
 - Kết nối cơ sở dữ liệu để tải dữ liệu hóa đơn (ResultSet).
- Xử lý sự kiện (actionPerformed): Khi nhấn "Quay lại", quay về Reception.

1.18. Revenue: Lớp này extends JFrame và implements Interface ActionListener.

1.18.1. Tổ chức:

- Danh sách chọn (JComboBox): Lọc doanh thu theo tất cả hóa đơn, ngày, tháng.
- Bảng (JTable): Hiện thị thông tin mã hóa đơn, số phòng, CCCD, ngày nhận trả, tổng tiền.
- Nút "Quay lại" (JButton): Trở về Reception.

1.18.2. Chức năng:

- Khởi tạo giao diện (Constructor):
 - Thiết lập cửa sổ, màu nền.

- Tải dữ liệu hóa đơn từ cơ sở dữ liệu (ResultSet).
- Hiện thị bảng danh sách hóa đơn.
- Xử lý sự kiện (actionPerformed):
 - Khi chọn lọc theo Ngày, chương trình:
 - Tính tổng doanh thu mỗi ngày từ hóa đơn đã thanh toán.
 - Hiện thị kết quả lên bảng.
 - Khi chọn lọc theo Tháng, chương trình:
 - Tính tổng doanh thu mỗi tháng từ hóa đơn đã thanh toán.
 - Hiện thị kết quả lên bảng.
 - Khi chọn tất cả hóa đơn, hiển thị toàn bộ danh sách hóa đơn.
 - Khi nhấn "Quay lại", trở về Dashboard.

1.19. Connect: Lớp này có nhiệm vụ kết nối hệ thống quản lý khách sạn với cơ sở dữ liệu MySQL.

1.19.1. Tổ chức:

- Khai báo thuộc tính:
 - Connection c: Kết nối với cơ sở dữ liệu.
 - Statement s: Thực thi truy vấn SQL.
- Hàm khởi tạo (Constructor):
 - Tải driver MySQL (Class.forName("com.mysql.cj.jdbc.Driver")).
 - Thiết lập kết nối (DriverManager.getConnection(...)).
 - Tạo Statement để thực thi SQL (c.createStatement()).
 - Xử lý ngoại lệ:
 - ClassNotFoundException: Nếu thiếu driver JDBC.
 - SQLException: Nếu không kết nối được với MySQL.

1.19.2. Chức năng:

- Kết nối hệ thống với MySQL.
- Cho phép thực thi các truy vấn SQL.
- Xử lý lỗi kết nối an toàn.

2. Các thuật toán/kỹ thuật quan trọng

2.1. Kết nối và thao tác với cơ sở dữ liệu

- Kết nối MySQL bằng JDBC (Connect.java):
 - Sử dụng DriverManager để thiết lập kết nối với cơ sở dữ liệu.
 - Dùng Statement và PreparedStatement để thực thi truy vấn SQL.
 - Xử lý lỗi kết nối (SQLException) giúp duy trì độ ổn định của hệ thống.
- Tương tác với dữ liệu bằng SQL:
 - Truy vấn dữ liệu (SELECT) để hiển thị danh sách khách hàng, phòng, nhân viên, hóa đơn.
 - Thêm mới (INSERT), cập nhật (UPDATE), và xóa (DELETE) dữ liệu linh hoạt.
 - Kết hợp bảng (JOIN) để lấy dữ liệu liên quan giữa khách hàng, phòng và hóa đơn.

2.2. Tổ chức giao diện và xử lý sự kiện

- Sử dụng Swing để tạo GUI:
 - Các lớp như AddCustomer, Reception, Checkout, Revenue đều dùng JFrame, JButton, JLabel, JTable, JTextField, JRadioButton, JComboBox, v.v.
 - JScrollPane giúp hiển thị danh sách dài mà vẫn dễ dàng tương tác.
 - Choice hỗ trợ chọn dữ liệu từ danh sách động.
- Xử lý sự kiện bằng ActionListener:
 - Xử lý các nút bấm (Button) để điều hướng và thao tác với dữ liệu.
 - Lắng nghe sự kiện nhập (KeyListener) để hỗ trợ nhập nhanh bằng phím Enter.

- ComboBox Listener giúp cập nhật danh sách dựa trên lựa chọn của người dùng.

2.3. Các thuật toán quan trọng

- Cập nhật trạng thái phòng:
 - Khi khách hàng đặt phòng, trạng thái phòng được đổi từ "Trống" → "Đã đặt".
 - Khi khách trả phòng, trạng thái phòng đổi từ "Đã đặt" → "Trống".
- Tính toán tổng tiền hóa đơn (Revenue.java):
 - Công thức:
`SELECT SUM(DATEDIFF(NgayTra, NgayNhan) * GiaMacDinh) AS 'Tổng Doanh Thu'`
`FROM HoaDon JOIN Phong ON HoaDon.SoPhong = Phong.SoPhong`
`WHERE HoaDon.TrangThai = 'Đã thanh toán'`
 - DATEDIFF(NgayTra, NgayNhan) giúp tính số ngày khách thuê phòng.
 - Nhóm dữ liệu theo ngày/tháng (GROUP BY) để thống kê doanh thu.
- Kiểm tra và cập nhật thông tin khách hàng (AddCustomer.java):
 - Trước khi thêm khách hàng, kiểm tra CCCD đã tồn tại chưa.
 - Nếu tồn tại, cập nhật thông tin thay vì tạo bản ghi mới.
 - COUNT(*) giúp kiểm tra số lượng bản ghi khách hàng.

2.4. Tối ưu hóa dữ liệu và hiệu suất

- DbUtils.resultSetToTableModel(rs) giúp hiển thị dữ liệu từ SQL nhanh chóng lên JTable.
- Format tiền tệ (FORMAT(value, 0)) giúp hiển thị số tiền dễ đọc.
- Cấu trúc chương trình theo mô hình MVC:
 - Model: Connect.java xử lý dữ liệu.

- View: Các lớp JFrame hiển thị giao diện.
- Controller: ActionListener điều khiển hoạt động hệ thống.

2.5. Bảo mật và xử lý ngoại lệ

- Sử dụng PreparedStatement thay vì Statement để tránh SQL Injection.
- Xử lý lỗi bằng try-catch:
 - Nếu kết nối database thất bại, hiển thị thông báo cho người dùng thay vì crash hệ thống.
 - Nếu không tìm thấy khách hàng hoặc phòng, hiển thị "N/A" thay vì báo lỗi.
- Ẩn mật khẩu (JPasswordField) trong Admin.java và Login.java.

3. Code mẫu cho các chức năng chính

3.1. Kết nối Cơ sở dữ liệu (Connect.java): Kết nối hệ thống với MySQL để thực hiện truy vấn SQL

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class Connect {

    public Connection c;

    public Statement s;

    public Connect() {

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");
```

```

        c
        =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hotelmanagementsystem?
useSSL=false&serverTimezone=UTC",

        "root", "1234");

s = c.createStatement();

} catch (ClassNotFoundException | SQLException e) {

    throw new RuntimeException("Database connection failed", e);

}

}

}

```

3.2. Đăng nhập (Admin.java): Xác thực người dùng trước khi truy cập hệ thống.

```

@Override

public void actionPerformed(ActionEvent event) {

    if (event.getSource() == login) {

        String user = username.getText();

        String pass = new String(password.getPassword());

        try {

            Connect c = new Connect();

            String query = "SELECT * FROM Administrator WHERE
username = ? AND password = ?";

            PreparedStatement pstmt = c.c.prepareStatement(query);

            pstmt.setString(1, user);

            pstmt.setString(2, pass);

            ResultSet rs = pstmt.executeQuery();

            if (rs.next()) {

```

```

        this.setVisible(false);

        new Dashboard();

    } else {

        JOptionPane.showMessageDialog(null, "Sai tên đăng nhập
hoặc mật khẩu.");

    }

    } catch (Exception e) {

        e.printStackTrace();

    }

    } else if (event.getSource() == cancel) {

        this.setVisible(false);

        new Dashboard();

    }

    }

```

3.3. Thêm Khách hàng (AddCustomer.java): Lưu thông tin khách hàng mới vào hệ thống.

```

@Override

public void actionPerformed(ActionEvent event) {

    if (event.getActionCommand().equals("Thêm")) {

        String cccd = jTextFieldCCCD.getText();

        String hoTen = jTextFieldName.getText();

        String gioiTinh = rmale.isSelected() ? "Nam" : rfemale.isSelected() ?
        "Nữ" : null;

        String quocTich = jTextFieldCountry.getText();

        String email = jTextFieldEmail.getText();

```



```

String sdt = jTextFieldPhone.getText();

String soPhong = choiceRoom.getSelectedItem();

String checkin = jLabelCheckinTime.getText();

String duaTruoc = jTextFieldDeposit.getText();

try {

    Connect conn = new Connect();

    String insertQuery = "INSERT INTO KhachHang VALUES
(? , ? , ? , ? , ? , ? , ?)";

    PreparedStatement pstmt = conn.c.prepareStatement(insertQuery);

    pstmt.setString(1, cccd);

    pstmt.setString(2, hoTen);

    pstmt.setString(3, gioiTinh);

    pstmt.setString(4, quocTich);

    pstmt.setString(5, email);

    pstmt.setString(6, sdt);

    pstmt.setString(7, duaTruoc);

    pstmt.executeUpdate();

    JOptionPane.showMessageDialog(null, "Thêm Khách hàng thành
công!");

} catch (Exception e) {

    JOptionPane.showMessageDialog(null, "Error: " + e.getMessage());

}

}

}

```

3.4. Thanh toán và Trả phòng (Checkout.java): Cập nhật thông tin đặt phòng của khách hàng sau khi trả phòng.

@Override

```
public void actionPerformed(ActionEvent event) {  
    if (event.getSource() == checkout) {  
        String queryUpdateRoom = "UPDATE Phong SET TrangThai='Trống'  
WHERE SoPhong = ?";  
        String queryUpdateBill = "UPDATE HoaDon SET NgayTra=? WHERE  
SoPhong = ? AND TrangThai=N'Chưa thanh toán";  
        try {  
            Connect c = new Connect();  
            PreparedStatement pstmtUpdateRoom =  
c.c.prepareStatement(queryUpdateRoom);  
            pstmtUpdateRoom.setString(1, jtext_room.getText());  
            pstmtUpdateRoom.executeUpdate();  
            PreparedStatement pstmtUpdateBill =  
c.c.prepareStatement(queryUpdateBill);  
            pstmtUpdateBill.setString(1, jlabel_checkout.getText());  
            pstmtUpdateBill.setString(2, jtext_room.getText());  
            pstmtUpdateBill.executeUpdate();  
            JOptionPane.showMessageDialog(null, "Checkout thành công!");  
            this.setVisible(false);  
            new Reception();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```

    } else {

        this.setVisible(false);

        new Reception();

    }

}

```

3.5. Thống kê Doanh thu (Revenue.java): Tính tổng doanh thu theo tháng từ hóa đơn khách sạn.

```

if (choiceRoom.getSelectedItem().equals("Tháng")) {

    try {

        Connect c = new Connect();

        ResultSet rs = c.s.executeQuery("SELECT YEAR(HoaDon.NgayTra)
AS 'Năm', MONTH(HoaDon.NgayTra) AS 'Tháng', "

            + "FORMAT(SUM(DATEDIFF(HoaDon.NgayTra,
HoaDon.NgayNhan) * Phong.GiaMacDinh), 0) AS 'Tổng Doanh Thu' "

            + "FROM HoaDon JOIN Phong ON HoaDon.SoPhong =
Phong.SoPhong WHERE HoaDon.TrangThai = 'Đã thanh toán' "

            + "GROUP BY YEAR(HoaDon.NgayTra),
MONTH(HoaDon.NgayTra) ORDER BY Năm, Tháng;");

        jTable.setModel(DbUtils.resultSetToTableModel(rs));

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, "Error: " + e.getMessage());

    }

}

```

4. Hướng dẫn build và chạy chương trình

4.1. Cài đặt môi trường phát triển: Chuẩn bị các công cụ sau:

- JDK (Java Development Kit):

- Tải và cài đặt phiên bản mới nhất từ trang chính thức của Oracle hoặc OpenJDK.
- Kiểm tra phiên bản đã cài đặt bằng lệnh `java -version`.
- MySQL Server:
 - Tải và cài đặt MySQL từ trang chính thức.
 - Cấu hình tài khoản mặc định (root) và mật khẩu.
- MySQL Connector/J:
 - Đây là thư viện JDBC giúp Java kết nối với MySQL.
 - Tải tệp `mysql-connector-java.jar` và lưu vào thư mục dự án.
- IDE để phát triển Java: Có thể sử dụng IntelliJ IDEA, Eclipse, hoặc NetBeans.

4.2. Cấu hình cơ sở dữ liệu MySQL

- Mở MySQL Workbench hoặc Command Line, nhập các lệnh SQL sau để tạo database:

```
create database hotelmanagementsystem;
```

```
use hotelmanagementsystem;
```

```
create table login (username varchar(25), password varchar(25));
```

```
insert into login values("reception", "1234");
```

```
create table administrator (username varchar(25), password varchar(25));
```

```
insert into administrator values("admin", "1234");
```

```
CREATE TABLE NhanVien(
```

```
    CCCD VARCHAR(50) primary key,
```

```
    HoTen NVARCHAR(100),
```

```
    Tuoi INT,
```

```
    GioiTinh NVARCHAR(10),
```

```
    ChucVu NVARCHAR(50),
```

```

Luong INT,

SDT_NV VARCHAR(15),

Email_NV VARCHAR(100)

);

INSERT INTO NhanVien (CCCD, HoTen, Tuoi, GioiTinh, ChucVu, SDT_NV,
Email_NV, Luong)

VALUES

('112233445566', N'Nguyễn Ngọc An', 25, N'Nam', N'Quản lý', '0911223344',
'nguyenngocan@hotel.com', 15000000),

('223344556677', N'Trần Văn Bảo', 23, N'Nam', N'Lễ tân', '0922334455',
'tranvanbao@hotel.com', 12000000),

('334455667788', N'Lê Thị Cẩm', 25, N'Nữ', N'Lễ tân', '0933445566',
'lethicam@hotel.com', 12000000),

('445566778899', N'Phạm Đức Dũng', 27, N'Nam', N'Buồng phòng', '0944556677',
'phamducdung@hotel.com', 10000000),

('556677889900', N'Vũ Thị Em', 27, N'Nữ', N'Buồng phòng', '0955667788',
'vuthiem@hotel.com', 10000000),

('667788990011', N'Hoàng Văn Phúc', 29, N'Nam', N'Bếp', '0966778899',
'hoangvanphuc@hotel.com', 18000000),

('778899001122', N'Nguyễn Thị Hương', 27, N'Nữ', N'Bếp', '0977889900',
'nguyenthihuong@hotel.com', 18000000),

('889900112233', N'Trần Minh Khôi', 35, N'Nam', N'Bảo vệ', '0988990011',
'tranminhkhoi@hotel.com', 9000000),

('990011223344', N'Lý Thị Mai', 24, N'Nữ', N'Kế toán', '0999001122',
'lythimai@hotel.com', 15000000),

('001122334455', N'Đặng Văn Nam', 40, N'Nam', N'Quản lý', '0900112233',
'dangvannam@hotel.com', 20000000);

```

CREATE TABLE Phong(

SoPhong NVARCHAR(20) NOT NULL primary key,

SoGiuong INT NOT NULL,

TienNghì NVARCHAR(100) NOT NULL,

TrangThai NVARCHAR(50) NOT NULL,

GiaMacDinh INT NOT NULL

);

INSERT INTO Phong (SoPhong, SoGiuong, TienNghì, TrangThai, GiaMacDinh)

VALUES

('101', 2, N'Điều hòa, Wifi, Tivi', N'Đã đặt', 500000),

('102', 1, N'Điều hòa, Wifi', N'Trống', 300000),

('103', 3, N'Điều hòa, Wifi, Tivi, Tủ lạnh', N'Trống', 700000),

('104', 2, N'Wifi, Tivi', N'Đã đặt', 600000),

('105', 1, N'Điều hòa', N'Trống', 400000),

('106', 2, N'Điều hòa, Wifi', N'Trống', 450000),

('107', 3, N'Điều hòa, Wifi, Tivi', N'Đã đặt', 750000),

('108', 1, N'Wifi', N'Đã đặt', 350000),

('109', 2, N'Điều hòa, Tủ lạnh', N'Đã đặt', 550000),

('110', 1, N'Điều hòa, Wifi, Tivi', N'Trống', 600000);

CREATE TABLE KháchHang(

CCCD VARCHAR(50) NOT NULL primary key,

HoTen NVARCHAR(100) NOT NULL,

GioiTinh NVARCHAR(10) NOT NULL,

QuocTich NVARCHAR(50),

Email_KH VARCHAR(100),

```

SDT_KH VARCHAR(15),

DuaTruoc INT    default (0)

);

INSERT INTO KhachHang (CCCD, SDT_KH, GioiTinh, HoTen, QuocTich,
Email_KH)

VALUES

('123456789012', '0912345678', N'Nam', N'Nguyễn Văn An',N'Việt Nam',
'nguyenvanan@gmail.com'),

('234567890123', '0923456789', N'Nữ', N'Trần Thị Bình',N'Việt Nam',
'tranthibinh@gmail.com'),

('345678901234', '0934567890', N'Nam', N'Lê Hoàng Cường', N'Việt Nam',
'lehoangcuong@gmail.com'),

('456789012345', '0945678901', N'Nữ', N'Phạm Ngọc Diệp', N'Việt Nam',
'phamngocdiep@gmail.com'),

('567890123456', '0956789012', N'Nam', N'Vũ Minh Đức',N'Việt Nam',
'vuminhduc@gmail.com'),

('678901234567', '0967890123', N'Nữ', N'Hoàng Thị Lan', N'Việt Nam',
'hoangthilan@gmail.com'),

('789012345678', '0978901234', N'Nam', N'Nguyễn Tuấn Hùng', N'Việt Nam',
'nguyentuanhung@gmail.com'),

('890123456789', '0989012345', N'Nữ', N'Lý Thị Mai', N'Việt Nam',
'lythimai@gmail.com'),

('901234567890', '0990123456', N'Nam', N'Trương Văn Phúc', N'Việt Nam',
'truongvanphuc@gmail.com'),

('012345678901', '0901234567', N'Nữ', N'Đặng Kim Ngân',N'Việt Nam',
'dangkimngan@gmail.com');

CREATE TABLE HoaDon(

```

MaHD VARCHAR(10) NOT NULL,
CCCD VARCHAR(50) NOT NULL ,
SoPhong NVARCHAR(20) NOT NULL,
NgayNhan DATETIME,
NgayTra DATETIME,
TrangThai NVARCHAR(50) NOT NULL DEFAULT N'Chưa thanh toán'

);

ALTER TABLE HoaDon

ADD CONSTRAINT FK_HoaDon_KhachHang

FOREIGN KEY (CCCD) REFERENCES KhachHang(CCCD) ON DELETE
CASCADE;

ALTER TABLE HoaDon

ADD CONSTRAINT FK_HoaDon_Phong

FOREIGN KEY (SoPhong) REFERENCES Phong(SoPhong) ON DELETE
CASCADE;

INSERT INTO HoaDon (MaHD, CCCD, SoPhong, NgayNhan, NgayTra, TrangThai)

VALUES

('HD001', '123456789012', '101', '2025-05-01 14:00:00', '2025-05-03 12:00:00',
N'Chưa thanh toán'),

('HD002', '234567890123', '102', '2025-05-02 15:00:00', '2025-05-04 11:00:00',
N'Đã thanh toán'),

('HD003', '345678901234', '103', '2025-05-03 13:30:00', '2025-05-05 10:30:00',
N'Chưa thanh toán'),

('HD004', '456789012345', '104', '2025-05-04 16:00:00', '2025-05-06 12:00:00',
N'Đã thanh toán'),

('HD005', '567890123456', '105', '2025-05-05 14:30:00', '2025-05-07 11:30:00',
N'Chưa thanh toán'),

('HD006', '678901234567', '106', '2025-05-06 17:00:00', '2025-05-08 12:30:00',
N'Đã thanh toán'),

('HD007', '789012345678', '107', '2025-05-07 15:00:00', '2025-05-09 10:00:00',
N'Chưa thanh toán'),

('HD008', '890123456789', '108', '2025-05-08 14:00:00', '2025-05-10 12:00:00',
N'Đã thanh toán'),

('HD009', '901234567890', '109', '2025-05-09 16:30:00', '2025-05-11 11:00:00',
N'Chưa thanh toán'),

('HD010', '012345678901', '110', '2025-05-10 13:00:00', '2025-05-12 10:30:00',
N'Đã thanh toán');

-hàm tạo MaHD tự động

DELIMITER //

CREATE FUNCTION GenerateMaHD() RETURNS VARCHAR(10)

DETERMINISTIC

BEGIN

DECLARE new_id INT;

DECLARE new_mahd VARCHAR(10);

-Lấy số lượng hóa đơn hiện có

SELECT COUNT(*) + 1 INTO new_id FROM HoaDon;

-Tạo mã hóa đơn mới bằng cách ghép tiền tố 'HD' với số lượng hóa đơn

SET new_mahd = CONCAT('HD', LPAD(new_id, 3, '0'));

RETURN new_mahd;

END //

DELIMITER ;

4.3. Cấu hình dự án Java: Nếu sử dụng IntelliJ IDEA hoặc Eclipse, thực hiện các bước sau:

- Tạo một Project Java
 - Chọn Java SE làm cấu hình ban đầu.
 - Đặt tên dự án là hotelmanagementsystem.
- Thêm thư viện MySQL Connector/J
 - Trên IntelliJ IDEA: File → Project Structure → Libraries → Add MySQL Connector/J.
 - Trên Eclipse: Build Path → Add External JARs → Chọn tệp mysql-connector-java.jar.
- Cấu trúc thư mục dự án
 - Các tệp mã nguồn Java sẽ nằm trong thư mục src.
 - Các thư viện sẽ được lưu trong thư mục lib.
 - Ví dụ về cấu trúc thư mục:

HotelManagementSystem/

```
|— src/
|   |— hotelmanagementsystem
|   |   |— Connect.java
|   |   |— Dashboard.java
|   |   |— Reception.java
|   |   |— AddCustomer.java
|   |   |— Checkout.java
|   |   |— Revenue.java
|   |   |— ...
|— lib/
```

| |—— mysql-connector-java.jar

4.4. Cài đặt kết nối MySQL trong Java

- Mở tệp Connect.java và đảm bảo đoạn mã sau được sử dụng để kết nối Java với MySQL:

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

public class Connect {

    public Connection c;

    public Statement s;

    public Connect() {

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

            c =

DriverManager.getConnection("jdbc:mysql://localhost:3306/hotelmanagementsystem?

useSSL=false&serverTimezone=UTC",

                            "root", "1234");

            s = c.createStatement();

        } catch (ClassNotFoundException | SQLException e) {

            throw new RuntimeException("Database connection failed", e);

        }

    }

}
```

- Thay đổi thông tin "root", "1234" theo tài khoản MySQL của bạn.

4.5. Build và chạy chương trình: Có hai cách để build và chạy chương trình: sử dụng Terminal hoặc IDE.

- Dùng IntelliJ IDEA hoặc Eclipse
 - Mở dự án → src/hotelmanagementsystem/Main.java.
 - Thêm MySQL Connector/J vào thư viện dự án.
 - Nhấn Run hoặc Shift + F10 để khởi chạy chương trình.
- Dùng Terminal hoặc Command Line
 - Biên dịch tất cả các file Java: `javac -d bin src/hotelmanagementsystem/*.java`
 - Chạy chương trình: `java -cp "bin:lib/mysql-connector-java.jar" hotelmanagementsystem.Main`

4.6. Kiểm tra và debug: Nếu gặp lỗi, kiểm tra các nguyên nhân sau:

- Không tìm thấy driver MySQL: Kiểm tra lại mysql-connector-java.jar đã được thêm đúng chưa.
- Không kết nối được với MySQL:
 - Kiểm tra MySQL Server đang chạy.
 - Kiểm tra tài khoản MySQL (root, 1234).
- Truy vấn SQL bị lỗi:
 - Kiểm tra bảng đã được tạo đúng bằng lệnh SHOW TABLES;.
 - Kiểm tra cấu trúc bảng bằng lệnh DESCRIBE KháchHang;.
- Sử dụng System.out.println() để kiểm tra lỗi trong từng bước.

V. KIỂM THỬ

1. Kế hoạch kiểm thử

1.1. Giới thiệu

- Kế hoạch kiểm thử phác thảo các hoạt động kiểm thử cần thiết để đảm bảo chất lượng, độ tin cậy và hiệu suất của Hệ thống Quản lý Khách sạn (Hotel Management System).
- Mục tiêu là xác định và khắc phục các lỗi trước khi hệ thống được triển khai, đảm bảo trải nghiệm tốt nhất cho người dùng và hoạt động kinh doanh trơn tru.

1.2. Mục tiêu kiểm thử

- Xác minh rằng tất cả các chức năng của hệ thống hoạt động đúng theo yêu cầu nghiệp vụ.
- Đảm bảo hệ thống đáp ứng các tiêu chí về hiệu suất, bảo mật và khả năng sử dụng.
- Phát hiện và báo cáo tất cả các lỗi và khiếm khuyết để chúng có thể được khắc phục.
- Đảm bảo tính toàn vẹn và nhất quán của dữ liệu.
- Xác nhận rằng hệ thống dễ sử dụng và thân thiện với người dùng.

1.3. Các giai đoạn kiểm thử

- Kiểm thử đơn vị (Unit Testing):
 - Mục đích: Kiểm tra từng thành phần hoặc module riêng lẻ của mã nguồn (ví dụ: các phương thức trong Connect, AddCustomer, Revenue,...).
 - Thực hiện bởi người lập trình.
 - Mục tiêu: Đảm bảo mỗi đơn vị mã hoạt động chính xác độc lập.
- Kiểm thử tích hợp (Integration Testing):
 - Mục đích: Kiểm tra sự tương tác và giao tiếp giữa các module hoặc thành phần khác nhau (ví dụ: AddCustomer giao tiếp với Connect, Login giao tiếp với Reception,...).
 - Thực hiện bởi người lập trình.

- Mục tiêu: Phát hiện lỗi giao diện, lỗi truyền dữ liệu giữa các module.
- Kiểm thử hệ thống (System Testing):
 - Mục đích: Kiểm tra toàn bộ hệ thống như một thể thống nhất, đảm bảo nó đáp ứng các yêu cầu chức năng và phi chức năng.
 - Thực hiện bởi người lập trình.
 - Mục tiêu: Xác minh luồng nghiệp vụ end-to-end, hiệu suất, bảo mật, khả năng tương thích.

1.4. Các loại kiểm thử

1.4.1. Kiểm thử chức năng (Functional Testing):

- Kiểm thử đăng nhập: Đăng nhập thành công/thất bại với các tài khoản khác nhau (admin, lễ tân).
- Kiểm thử quản lý khách hàng: Thêm, sửa, xóa; kiểm tra ràng buộc dữ liệu (CCCD duy nhất, định dạng email/SĐT).
- Kiểm thử quản lý phòng: Thêm, sửa, xóa, tìm kiếm phòng; cập nhật trạng thái phòng (trống, đã đặt).
- Kiểm thử checkin/checkout: Quy trình nhận phòng, trả phòng cho khách hàng.
- Kiểm thử thanh toán/hóa đơn: Tạo hóa đơn, tính toán tổng tiền phòng, cập nhật trạng thái thanh toán, trạng thái hóa đơn.
- Kiểm thử quản lý nhân viên: Thêm, sửa, xóa thông tin nhân viên.

1.4.2. Kiểm thử phi chức năng (Non-Functional Testing):

- Kiểm thử bảo mật:
 - Kiểm tra quyền truy cập dựa trên vai trò (admin có quyền gì, lễ tân có quyền gì).
 - Kiểm tra lỗ hổng SQL Injection (mặc dù đã dùng parameterized queries, vẫn nên kiểm tra).
- Kiểm thử khả năng sử dụng (Usability Testing):

- Đánh giá tính thân thiện, dễ học, dễ sử dụng của giao diện người dùng.
- Kiểm tra luồng người dùng có hợp lý không.
- Kiểm thử tương thích (Compatibility Testing):
 - Kiểm tra hệ thống trên các phiên bản Windows khác nhau (Windows 10, Windows 11).
 - Kiểm tra với các phiên bản MySQL khác nhau.
- Kiểm thử hồi quy (Regression Testing):
 - Mục đích: Đảm bảo rằng các thay đổi hoặc sửa lỗi mới không gây ra lỗi mới hoặc làm hỏng các chức năng hiện có.
 - Thực hiện: Chạy lại các bộ kiểm thử đã có sau mỗi lần thay đổi mã nguồn.

1.5. Môi trường kiểm thử

- Phần cứng:
 - Máy tính xách tay
 - Hệ điều hành – Windows 11 Home Single Language
 - Bộ xử lý: 13th Gen Intel(R) Core(TM) i7-13620H
 - RAM 16.00 GB
 - Hệ điều hành 64-bit, bộ xử lý dựa trên x64
 - Độ phân giải màn hình 1920 x 1080
 - Bàn phím và chuột
- Phần mềm:
 - JDK phiên bản 24.0.1.
 - MySQL Server phiên bản 8.0.
 - MySQL Connector/J phiên bản 8.0.28.
 - IntelliJ IDEA.

2. Kết quả kiểm thử

Hệ thống Quản lý Khách sạn đã vượt qua hầu hết các bài kiểm thử chức năng và phi chức năng quan trọng. Một số lỗi nhỏ đã được phát hiện và ghi nhận, trong đó phần lớn đã được khắc phục. Hệ thống cho thấy khả năng hoạt động ổn định trên môi trường kiểm thử đã định.

2.1. Kết quả theo các giai đoạn kiểm thử

- Kiểm thử đơn vị (Unit Testing): Tất cả các module và phương thức cốt lõi đã được kiểm thử đơn vị thành công. Các lỗi cú pháp và logic cơ bản đã được phát hiện và sửa chữa ở giai đoạn này.
- Kiểm thử tích hợp (Integration Testing): Sự tương tác giữa các module đã được xác minh. AddCustomer kết nối thành công với cơ sở dữ liệu thông qua Connect, CheckBill đưa ra các thông tin chính xác của Hóa đơn, dữ liệu được lưu trữ và truy xuất đúng cách. Luồng dữ liệu giữa các form (ví dụ: Login đến Reception) hoạt động chính xác. Một số lỗi nhỏ liên quan đến việc truyền dữ liệu giữa các form đã được phát hiện và xử lý.
- Kiểm thử hệ thống (System Testing): Hệ thống tổng thể hoạt động đúng theo luồng nghiệp vụ end-to-end. Các quy trình chính như nhận phòng, trả phòng, tạo hóa đơn diễn ra suôn sẻ.

2.2. Kết quả theo các loại kiểm thử

2.2.1. Kiểm thử chức năng (Functional Testing):

- Kiểm thử đăng nhập:
 - Đăng nhập thành công với tài khoản "admin" và "lễ tân" hợp lệ.
 - Đăng nhập thất bại khi nhập sai tên đăng nhập/mật khẩu, hệ thống hiển thị thông báo lỗi phù hợp.
- Kiểm thử quản lý khách hàng:
 - Thêm khách hàng mới: Thành công, dữ liệu được lưu chính xác. Ràng buộc CCCD duy nhất hoạt động đúng.
 - Sửa thông tin khách hàng: Thành công, dữ liệu được cập nhật đúng.

- Xóa khách hàng: Thành công.
- Kiểm thử quản lý phòng:
 - Thêm, sửa, xóa phòng: Thành công.
 - Cập nhật trạng thái phòng: Hoạt động chính xác (ví dụ: từ “Trống” sang “Đã đặt” khi có khách nhận phòng).
- Kiểm thử check-in/check-out:
 - Quy trình nhận phòng: Khách hàng được gán phòng trống, thông tin được lưu, trạng thái phòng cập nhật.
 - Quy trình trả phòng: Khách hàng được thanh toán, phòng chuyển trạng thái “Đã đặt” sang “Trống”.
- Kiểm thử thanh toán/hóa đơn:
 - Tạo hóa đơn: Thành công.
 - Tính toán tổng tiền phòng: Chính xác dựa trên số ngày ở và giá phòng mặc định.
 - Cập nhật trạng thái thanh toán/hóa đơn: Hoạt động đúng.
- Kiểm thử quản lý nhân viên:
 - Thêm, xóa thông tin nhân viên: Thành công.

2.2.2. Kiểm thử phi chức năng (Non-Functional Testing):

- Kiểm thử bảo mật:
 - Quyền truy cập dựa trên vai trò: Đã xác minh rằng tài khoản admin có đầy đủ quyền, tài khoản lễ tân bị hạn chế các chức năng quản trị (ví dụ: không thể thêm nhân viên/phòng).
 - Lỗ hổng SQL Injection: Không phát hiện lỗ hổng SQL Injection khi thử nghiệm với các chuỗi độc hại, do việc sử dụng parameterized queries.
- Kiểm thử khả năng sử dụng (Usability Testing):

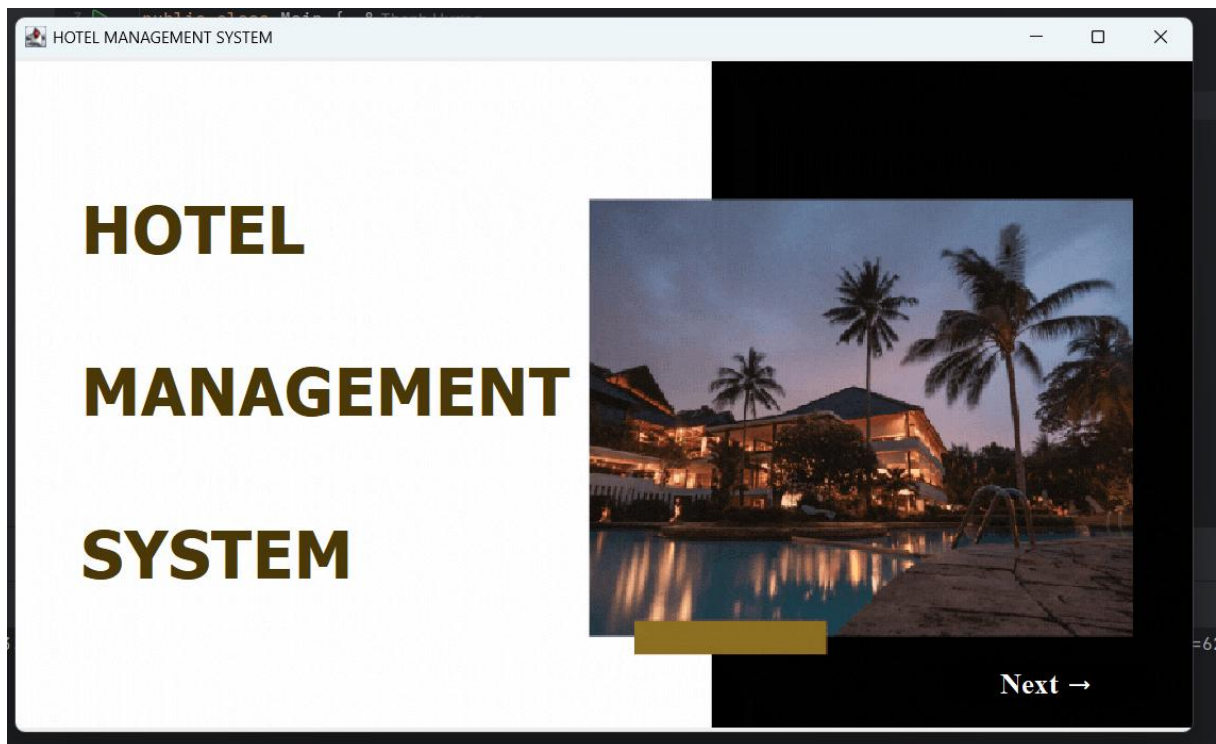
- Giao diện người dùng: Đánh giá chung là thân thiện, dễ học. Các nút và trường nhập liệu rõ ràng.
- Kiểm thử tương thích (Compatibility Testing):
 - Hệ điều hành: Hệ thống hoạt động ổn định trên Windows 10 và Windows 11.
 - MySQL: Hệ thống kết nối và hoạt động bình thường với MySQL Server phiên bản 5.5 và 8.0.
- Kiểm thử hồi quy (Regression Testing):
 - Sau các lần sửa lỗi và cập nhật, các bộ kiểm thử hồi quy đã được chạy lại. Không phát hiện lỗi mới hoặc chức năng hiện có bị ảnh hưởng.

2.3. Tổng hợp lỗi đã phát hiện

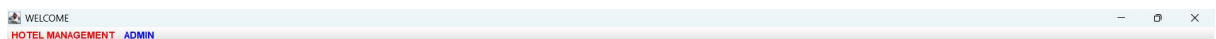
- Tổng số lỗi phát hiện: 5
- Lỗi nghiêm trọng (Critical): 0
- Lỗi lớn (Major): 1 (Lỗi tính toán tổng hóa đơn ban đầu do sai logic join trong SQL, đã khắc phục)
- Lỗi nhỏ (Minor): 4 (Lỗi hiển thị font chữ nhỏ ở một số trường, lỗi căn chỉnh giao diện nhỏ, đã khắc phục)
- Trạng thái lỗi: Tất cả các lỗi đã được khắc phục và xác nhận lại.

3. Demo các chức năng

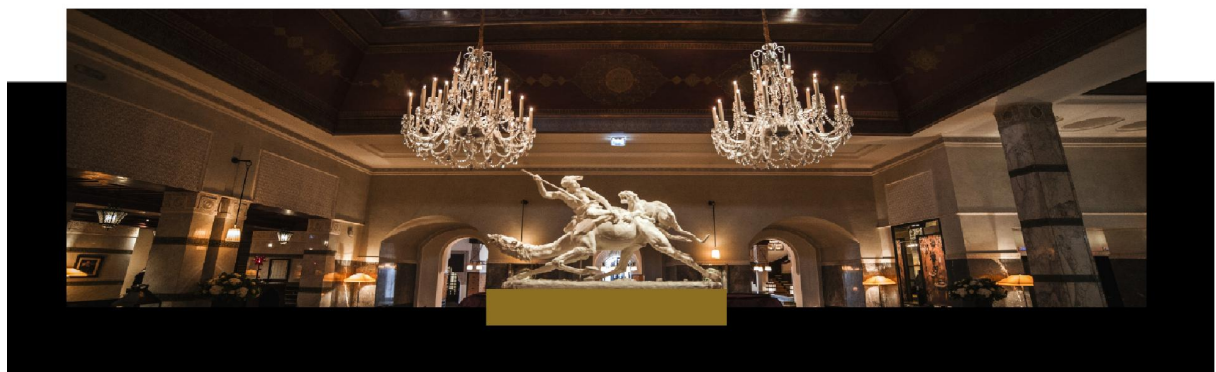
- Khi chạy chương trình, hiển thị form HotelManagementSystem đầu tiên. Sau đó nhấn nút “Next” để chuyển sang form Dashboard.



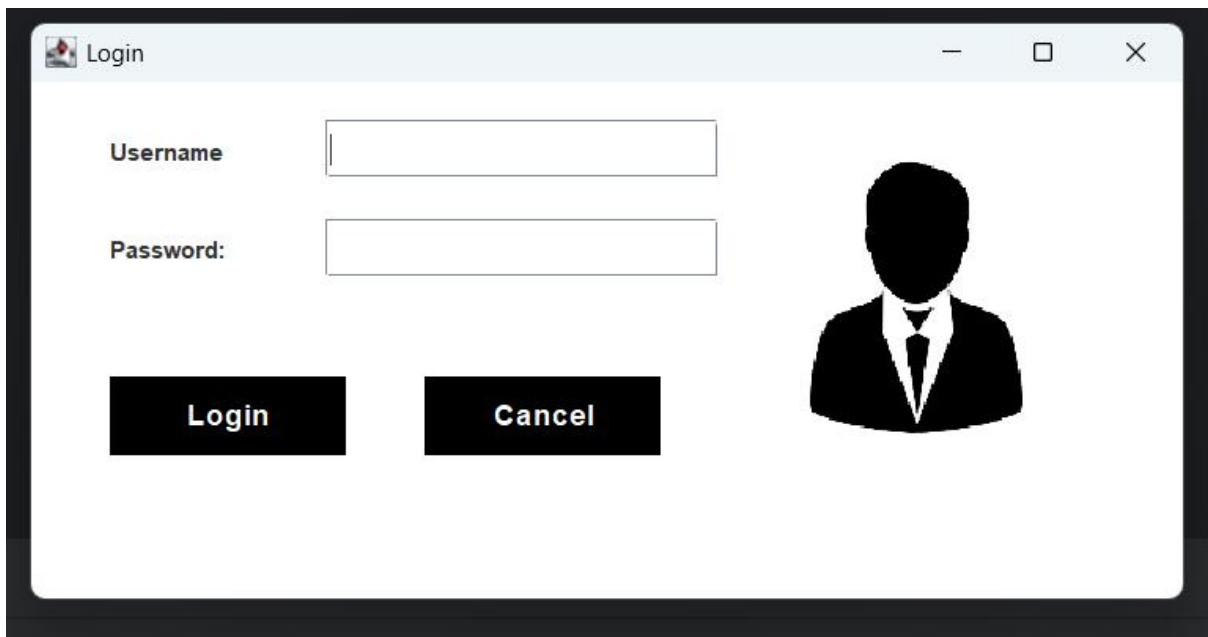
- Dashboard được hiển thị với hai menu: “Hotel Management” và “Admin”.



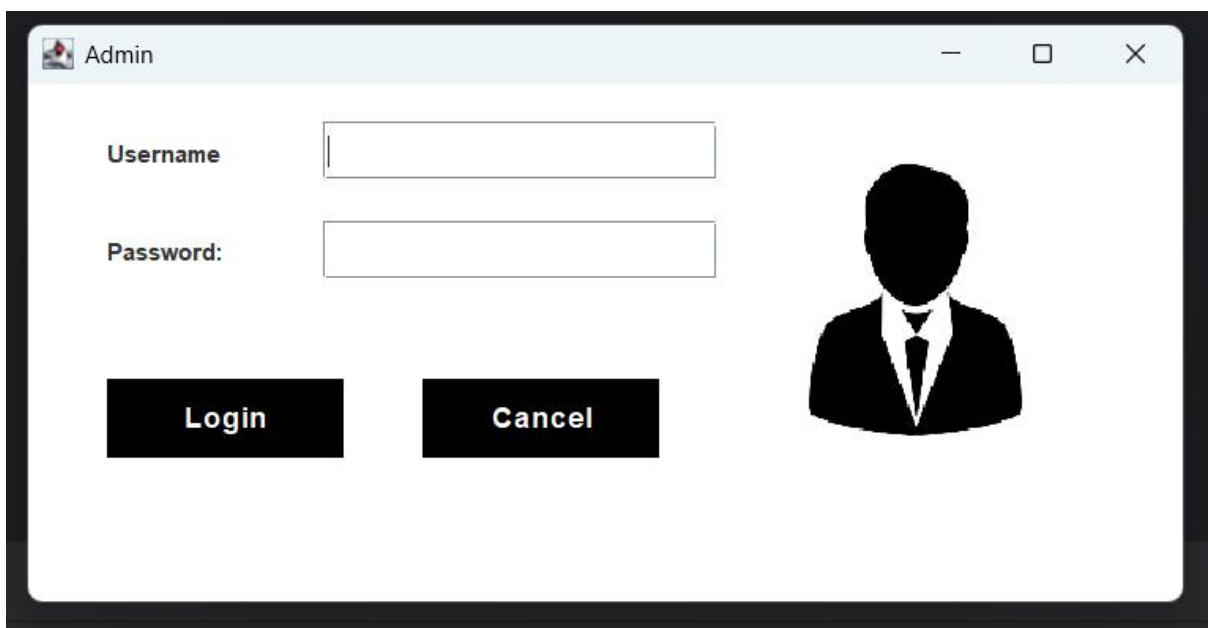
WELCOME TO LARANA HOTEL



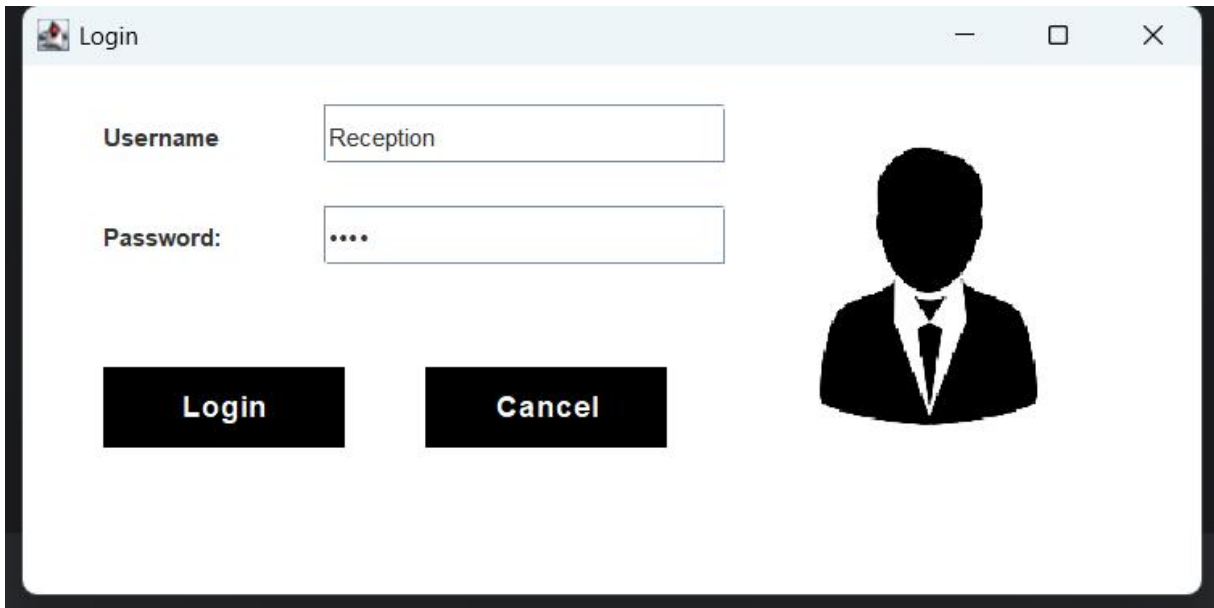
- Nhấn vào menu “Hotel Management” sẽ dẫn tới form Reception của Lễ tân, khi đó sẽ yêu cầu đăng nhập vào form Login của Lễ tân.



- Nhấn vào menu “Admin” sẽ hiện ra các menu item của admin, người dùng muốn đi đến form tương ứng với từng menu item thì sẽ được yêu cầu đăng nhập vào form Admin.



- Trong mỗi form đăng nhập, người dùng sẽ được yêu cầu nhập thông tin tài khoản và mật khẩu. Sau đó, nhấn “Login” (hoặc nhấn phím Enter) để truy cập vào các form tương ứng.

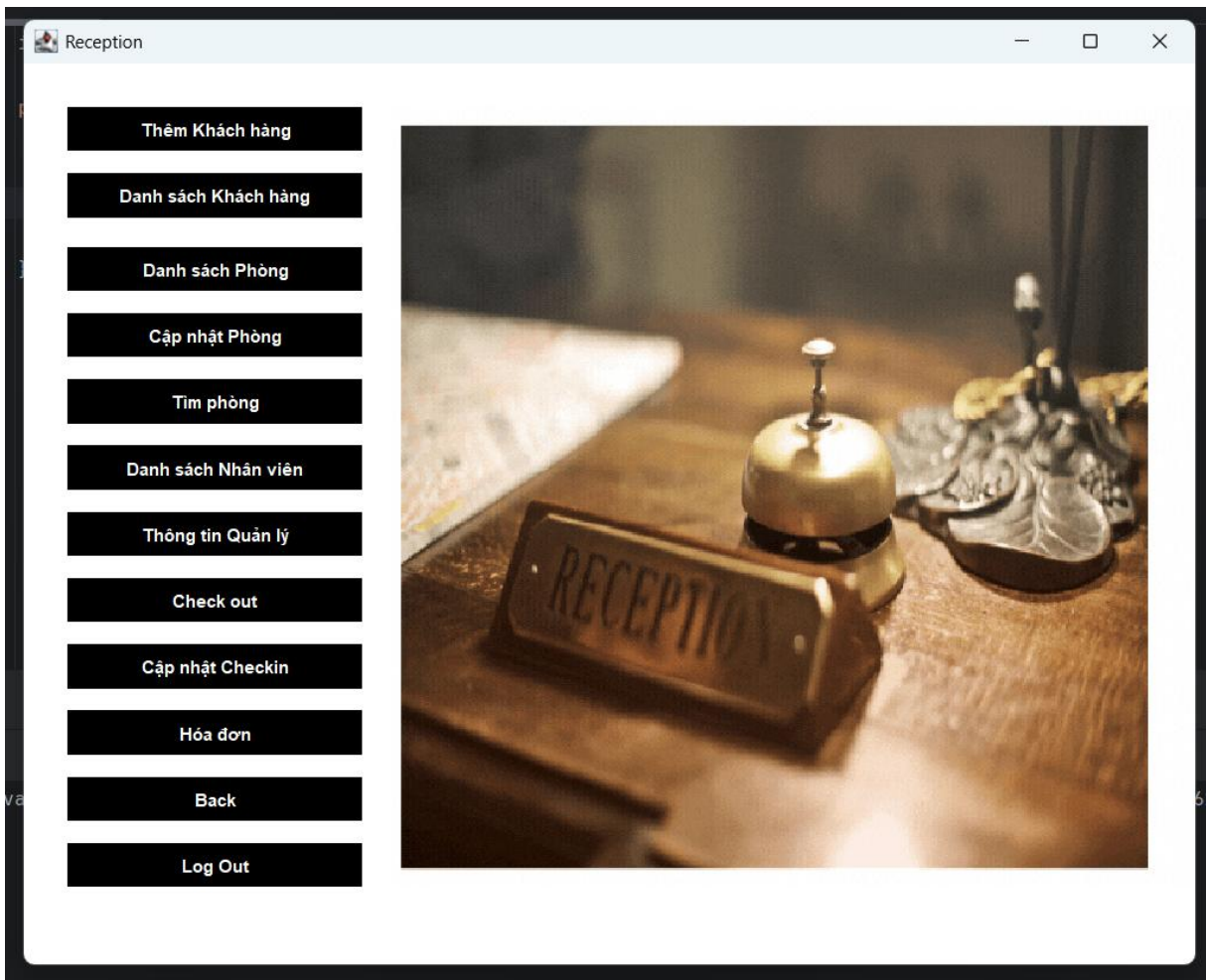


Username: Reception

Password:

Login Cancel

- Khi Lễ tân đã đăng nhập thành công thì chương trình sẽ hiện form Reception, gồm các Menu (là các button) để Lễ tân thực hiện các thao tác.



Thêm Khách hàng

Danh sách Khách hàng

Danh sách Phòng

Cập nhật Phòng

Tìm phòng

Danh sách Nhân viên

Thông tin Quản lý

Check out

Cập nhật Checkin

Hóa đơn

Back

Log Out

- Thêm thông tin khách hàng: Click vào button “Thêm Khách hàng” ở form Reception, nhập thông tin khách hàng (CCCD, tên, giới tính, quốc tịch, email, số điện thoại); đồng thời, chọn số phòng và nhập vào số tiền đưa trước để làm thủ tục nhận phòng. Khi nhập đủ thông tin, click button “Thêm” để thêm thông tin khách hàng.

- Quản lý thông tin khách hàng: Click vào button “Danh sách Khách hàng” ở form Reception, sẽ hiện danh sách khách hàng trong bảng JTable. Khi muốn xóa thông tin 1 khách hàng nào đó, Lẽ tân click vào 1 hàng trong bảng, nhấn “Xóa” để xóa thông tin khách hàng đã chọn.

Customer Information

CCCD	Họ Tên	Giới tính	Quốc tịch	Số điện thoại	Số Phòng	Ngày Nhận	Đưa trước
123456789012	Nguyễn Văn An	Nam	Việt Nam	0912345678	101	2025-05-01T14:00	200000
234567890123	Trần Thị Bình	Nữ	Việt Nam	0923456789	102	2025-05-02T15:00	0
345678901234	Lê Hoàng Cường	Nam	Việt Nam	0934567890	103	2025-05-03T13:30	0
456789012345	Phạm Ngọc Diệp	Nữ	Việt Nam	0945678901	104	2025-05-04T16:00	0
567890123456	Vũ Minh Đức	Nam	Việt Nam	0956789012	105	2025-05-05T14:30	0
0123456789951	Nguyễn Văn An	Nam	Việt Nam	0951234567	105	2025-06-05T22:45:0	250000
678901234567	Hoàng Thị Lan	Nữ	Việt Nam	0967890123	106	2025-05-06T17:00	0
789012345678	Nguyễn Tuấn Hùng	Nam	Việt Nam	0978901234	107	2025-05-07T15:00	0
890123456789	Lý Thị Mai	Nữ	Việt Nam	0989012345	108	2025-05-08T14:00	0
901234567890	Trương Văn Phúc	Nam	Việt Nam	0990123456	109	2025-05-09T16:30	0
012345678901	Đặng Kim Ngân	Nữ	Việt Nam	0901234567	110	2025-05-10T13:00	0

Xác Nhận
? Bạn có chắc chắn muốn xóa không?
Yes No

Quay lại Xóa

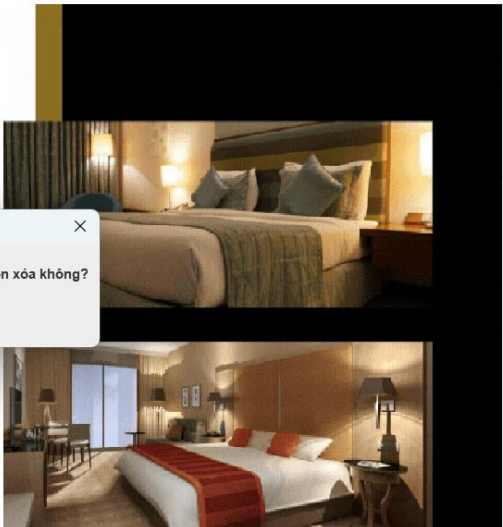
- Quản lý phòng: Click vào button “Danh sách Phòng” ở form Reception, danh sách phòng sẽ hiển thị với các thông tin chi tiết (số phòng, số giường, trạng thái, tiện nghi, giá mặc định). Khi muốn xóa phòng, tương tự như form CustomerInfor, người dùng nhấn button Xóa để xóa phòng không còn sử dụng.

Room

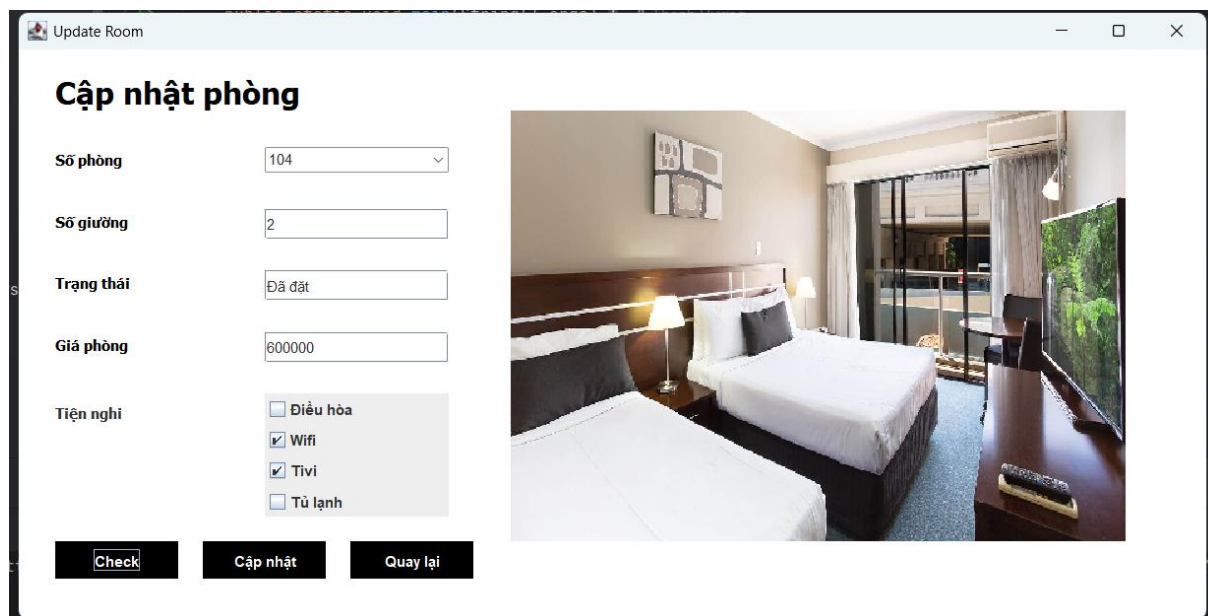
Số Phòng	Số giường	Tiện nghi	Trạng thái	Giá mặc định
101	2	Điều hòa, Wifi, Tivi	Đã đặt	500000
102	1	Điều hòa, Wifi	Trống	300000
103	3	Điều hòa, Wifi, Tivi, Tủ lạnh	Trống	700000
104	2	Wifi, Tivi	Đã đặt	600000
105	1	Điều hòa	Đã đặt	400000
106	2	Điều hòa, Wifi	Trống	450000
107	3	Điều hòa, Wifi, Tivi	Đã đặt	650000
108	1	Wifi	Đã đặt	200000
109	2	Điều hòa, Tủ lạnh	Đã đặt	550000
110	1	Điều hòa, Wifi, Tivi	Trống	350000

Xác Nhận
? Bạn có chắc chắn muốn xóa không?
Yes No

Quay lại Xóa



- Cập nhật phòng: Click vào button “Cập nhật Phòng” ở form Reception, người dùng chọn số phòng muốn cập nhật, nhấn “Check” để lấy thông tin phòng, nếu muốn sửa đổi, thực hiện sửa thông tin cần sửa rồi nhấn “Cập nhật” để lưu thông tin.



Cập nhật phòng

Số phòng: 104

Số giường: 2

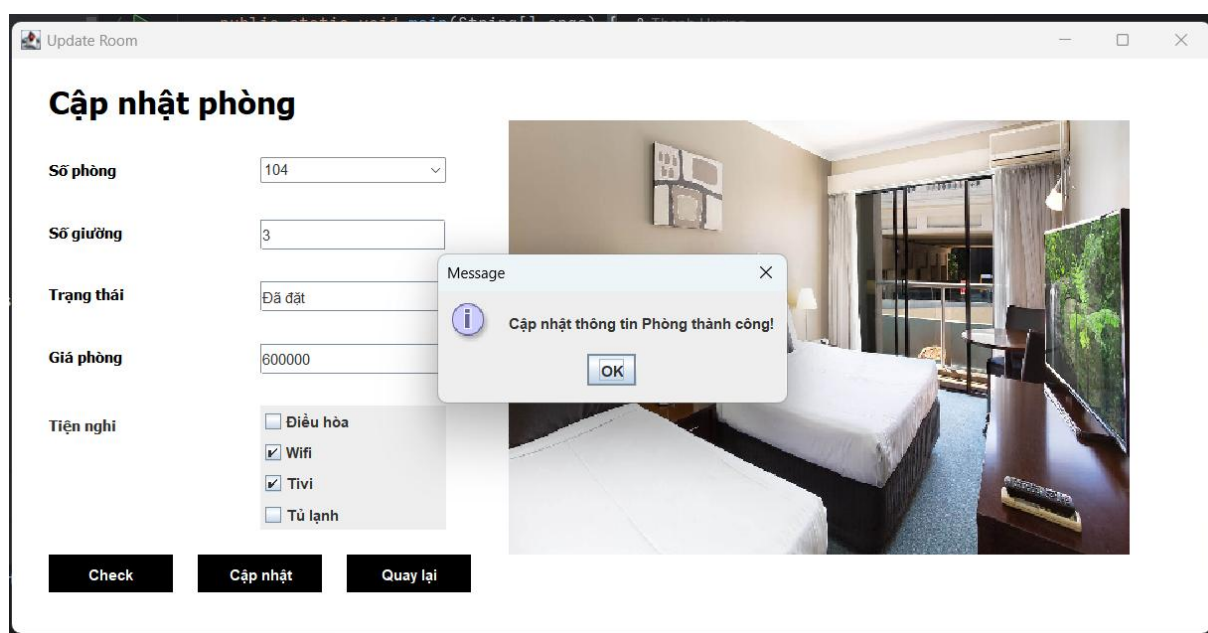
Trạng thái: Đã đặt

Giá phòng: 600000

Tiện nghi:

- ☐ Điều hòa
- ☒ Wifi
- ☒ Tivi
- ☐ Tủ lạnh

Buttons: Check, Cập nhật, Quay lại



Cập nhật phòng

Số phòng: 104

Số giường: 3

Trạng thái: Đã đặt

Giá phòng: 600000

Tiện nghi:

- ☐ Điều hòa
- ☒ Wifi
- ☒ Tivi
- ☐ Tủ lạnh

Buttons: Check, Cập nhật, Quay lại

Message: Cập nhật thông tin Phòng thành công! OK

- Tìm phòng: Click vào button “Tìm phòng” ở form Reception, nhập số giường và trạng thái để tìm kiếm phòng. Khi đã nhập đủ thông tin, nhấn “Tìm” để hiển thị kết quả tìm kiếm. Nhấn “Reset” để trả về bảng ban đầu.

Search Room

TÌM PHÒNG

Số giường ☒ Chỉ hiển thị phòng Trống

SoPhong	SoGiuong	TienNghie	TrangThai	GiaMacDinh
106	2	Điều hòa, Wifi	Trống	450000

Tim Quay lại Reset

Search Room

TÌM PHÒNG

Số giường ☐ Chỉ hiển thị phòng Trống

Số Phòng	Số giường	Tiện nghi	Trang thái	Giá mặc định
101	2	Điều hòa, Wifi, Tivi	Đã đặt	500000
102	1	Điều hòa, Wifi	Trống	300000
103	3	Điều hòa, Wifi, Tivi, Tủ lạnh	Trống	700000
104	3	Wifi, Tivi	Đã đặt	600000
105	1	Điều hòa	Đã đặt	400000
106	2	Điều hòa, Wifi	Trống	450000
107	3	Điều hòa, Wifi, Tivi	Đã đặt	750000
108	1	Wifi	Đã đặt	350000
109	2	Điều hòa, Tủ lạnh	Đã đặt	550000

Tim Quay lại Reset

- Quản lý Nhân viên: Click vào button “Danh sách Nhân viên” ở form Reception, sẽ hiển thị danh sách nhân viên trong bảng JTable. Tương tự như form thông tin khách hàng, cho phép xóa khi người dùng chọn 1 hàng trong bảng.

Employee Information

CCCD	Họ Tên	Tuổi	Giới tính	Chức vụ	Lương	Số điện thoại	Email
001122334455	Đặng Văn Nam	40	Nam	Quản lý	20000000	0900112233	dangvannam@hotel.com
112233445566	Nguyễn Ngọc An	25	Nam	Quản lý	15000000	0911223344	nguyennngocan@hotel.com
223344556677	Trần Văn Bảo	23	Nam	Lễ tân	12000000	0922334455	tranvanbao@hotel.com
334455667788	Lê Thị Cẩm	25	Nữ	Lễ tân	12000000	0933445566	lethicam@hotel.com
445566778899	Phạm Đức Dũng	27	Nam	Buồng phòng	10000000	0944556677	phamducdung@hotel.com
556677889900	Vũ Thị Em	27	Nữ	Buồng phòng	10000000	0955667788	vuthiem@hotel.com
667788990011	Hoàng Văn Phúc	29	Nam	Buồng phòng	10000000	0966778899	hoangvanphuc@hotel.com
778899001122	Nguyễn Thị Hương	27	Nữ	Buồng phòng	10000000	0977889900	nguyenthithuong@hotel.com
889900112233	Trần Minh Khôi	35	Nam	Buồng phòng	10000000	0988990011	tranminhkhoi@hotel.com
990011223344	Lý Thị Mai	24	Nữ	Buồng phòng	10000000	0999001122	lythimai@hotel.com

Xác Nhận
? Bạn có chắc chắn muốn xóa không?
Yes No

Quay lại Xóa

- Quản lý danh sách Nhân viên là Manager: tương tự như quản lý nhân viên.

Manager Information

CCCD	Họ Tên	Tuổi	Giới tính	Chức vụ	Lương	Số điện thoại	Email
001122334455	Đặng Văn Nam	40	Nam	Quản lý	20000000	0900112233	dangvannam@hotel.com
112233445566	Nguyễn Ngọc An	25	Nam	Quản lý	15000000	0911223344	nguyennngocan@hotel.com

Xác Nhận
? Bạn có chắc chắn muốn xóa không?
Yes No

Quay lại Xóa

- Cập nhật thông tin Checkin: Click vào button “Cập nhật Checkin” ở form Reception để cập nhật thông tin check-in của khách hàng. Khi Lễ tân đã chọn được số CCCD của khách hàng, nhấn “Check” để hiện thông tin, sau đó nhập vào thông tin cần cập nhật rồi nhấn “Cập nhật” để lưu thông tin mới.

CẬP NHẬT CHECKIN

CCCD: 123456789012

Mã Hóa đơn: HD001

Số phòng: 101

Tên: Nguyễn Văn An

Nhận phòng: 2025-05-01 14:00:00

Đưa trước: 200000

Còn lại: 300000

Buttons: Check, Cập nhật, Quay lại

Graphic: CHECK IN (right arrow), CHECK OUT (left arrow)

CẬP NHẬT CHECKIN

CCCD: 123456789012

Mã Hóa đơn: HD001

Số phòng: 101

Tên: Nguyễn Văn An

Nhận phòng: 2025-05-01 14:00:00

Đưa trước: 250000

Còn lại: 300000

Buttons: Check, Cập nhật, Quay lại

Message: Cập nhật thông tin Checkin thành công! OK

Graphic: CHECK IN (right arrow), CHECK OUT (left arrow)

- Checkout: Click vào button “Checkout” ở form Reception để nhập thông tin thanh toán cho khách hàng. Lễ tân chọn số CCCD của khách hàng, các thông tin liên quan như số phòng và thời gian vào sẽ được hiển thị, khi đó nhấn “Checkout” để thực hiện thủ tục trả phòng.

Checkout

CHECKOUT

Số CCCD

789012345678

✓

Số phòng

107

Nhận phòng

2025-05-07 15:00:00

Trả phòng

2025/06/05 23:01:54

Checkout

Quay lại

- Kiểm tra các hóa đơn: Click vào button “Hóa đơn” ở form Reception để hiển thị thông tin hóa đơn của khách hàng. Lễ tân được phép kiểm tra trạng thái hóa đơn và tổng tiền của từng hóa đơn.

Statistics Bill

Mã Hóa đơn	CCCD	Số Phòng	Ngày Nhận	Ngày Trả	Trạng Thái	Tổng Tiền
HD001	123456789012	101	2025-05-01T14:00	2025-05-03T12:00	Chưa thanh toán	1,000,000
HD002	234567890123	102	2025-05-02T15:00	2025-05-04T11:00	Đã thanh toán	800,000
HD003	345678901234	103	2025-05-03T13:30	2025-05-05T10:30	Chưa thanh toán	1,400,000
HD004	456789012345	104	2025-05-04T16:00	2025-05-06T12:00	Đã thanh toán	1,200,000
HD005	567890123456	105	2025-05-05T14:30	2025-05-07T11:30	Chưa thanh toán	800,000
HD011	0123456789951	105	2025-06-05T22:45:09		Chưa Thanh toán	
HD006	678901234567	106	2025-05-06T17:00	2025-05-08T12:30	Đã thanh toán	900,000
HD007	789012345678	107	2025-05-07T15:00	2025-05-09T10:00	Chưa thanh toán	1,500,000
HD008	890123456789	108	2025-05-08T14:00	2025-05-10T12:00	Đã thanh toán	700,000
HD009	901234567890	109	2025-05-09T16:30	2025-05-11T11:00	Chưa thanh toán	1,100,000
HD010	012345678901	110	2025-05-10T13:00	2025-05-12T10:30	Đã thanh toán	1,200,000

Quay lại

- Thêm thông tin Nhân viên: Khi click vào menu Admin thuộc form Dashboard, admin chọn vào menu item “Add Employee” và thực hiện đăng nhập để dẫn tới form nhập thông tin nhân viên mới, bao gồm các thông tin cụ thể như số CCCD, họ tên, giới tính, chức vụ, số điện thoại,... Sau đó, nhấn “Thêm” để lưu thông tin nhân viên.

Add Employee

Số CCCD

Tên

Tuổi

Giới tính ☒ Nam

Chức vụ

Lương

Số điện thoại

Email

Message
 Thêm Nhân viên mới thành công!

- Thêm thông tin phòng: Khi click vào menu Admin thuộc form Dashboard, admin chọn vào menu item “Add Room” và thực hiện đăng nhập để dẫn tới form để nhập thông tin phòng mới. Khi đã nhập đủ thông tin, nhấn “Thêm” để lưu thông tin.

Add Room

THÊM PHÒNG

Số phòng

Trạng thái

Số giường

Giá phòng

Tiện nghi

- ☒ Điều hòa
- ☒ Wifi
- ☒ Tivi
- ☐ Tủ lạnh

Message
 Thêm Phòng mới thành công!

- **Thống kê doanh thu:** Khi click vào menu Admin thuộc form Dashboard, admin chọn vào menu item “Revenue” và thực hiện đăng nhập để dẫn tới form hiển thị doanh thu. Admin có thể chọn thống kê doanh thu theo từng hóa đơn, tổng doanh thu theo ngày hoặc tháng.

Thống kê theo: Tất cả Hóa đơn

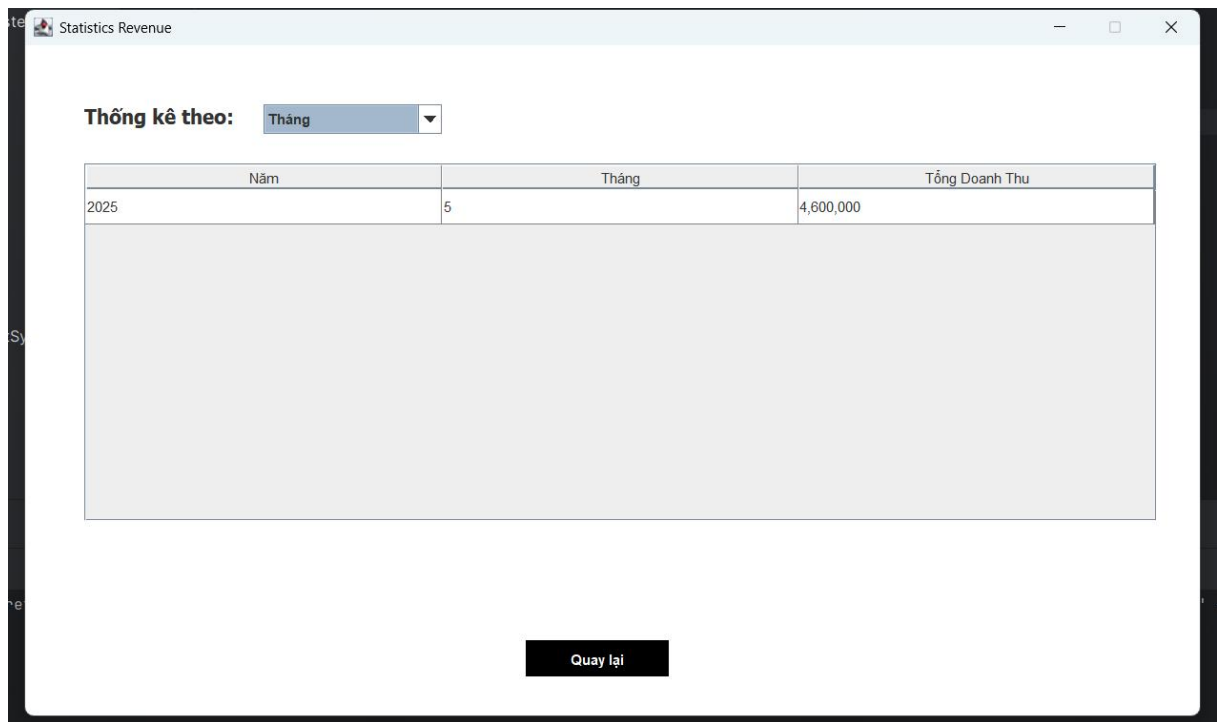
Mã Hóa đơn	CCCD	Số Phòng	Ngày Nhận	Ngày Trả	Tổng Tiền
HD001	123456789012	101	2025-05-01T14:00	2025-05-03T12:00	1,000,000
HD002	234567890123	102	2025-05-02T15:00	2025-05-04T11:00	600,000
HD003	345678901234	103	2025-05-03T13:30	2025-05-05T10:30	1,400,000
HD004	456789012345	104	2025-05-04T16:00	2025-05-06T12:00	1,200,000
HD005	567890123456	105	2025-05-05T14:30	2025-05-07T11:30	800,000
HD006	678901234567	106	2025-05-06T17:00	2025-05-08T12:30	900,000
HD007	789012345678	107	2025-05-07T15:00	2025-05-09T10:00	1,500,000
HD008	890123456789	108	2025-05-08T14:00	2025-05-10T12:00	700,000
HD009	901234567890	109	2025-05-09T16:30	2025-05-11T11:00	1,100,000

Quay lại

Thống kê theo: Ngày

Ngày	Tổng Doanh Thu
2025-05-04	600,000
2025-05-06	1,200,000
2025-05-08	900,000
2025-05-10	700,000
2025-05-12	1,200,000

Quay lại



VI. KẾT LUẬN

1. Kết quả đạt được

Dựa trên các mục tiêu đã đề ra và quá trình kiểm thử, dự án Hệ thống Quản lý Khách sạn đã đạt được những kết quả chính sau:

- Tự động hóa và Đơn giản hóa Quy trình: Hệ thống đã thành công trong việc chuyển đổi các hoạt động quản lý khách sạn hàng ngày từ thủ công sang máy tính hóa, giúp giảm đáng kể thời gian và công sức.
- Hệ thống có khả năng quản lý và duy trì hồ sơ khách hàng, phòng và nhân viên. Các chức năng thêm, sửa, xóa thông tin hoạt động ổn định.
- Các quy trình nghiệp vụ chính như nhận phòng, trả phòng, phân bổ phòng và xử lý thanh toán được thực hiện mượt mà, đảm bảo tính toàn vẹn dữ liệu.
- Giao diện người dùng được thiết kế trực quan, dễ sử dụng, giúp người dùng dễ dàng thao tác và làm quen với hệ thống.
- Hệ thống hoạt động ổn định trên môi trường Windows và MySQL đã định. Đặc biệt, việc sử dụng truy vấn tham số hóa đã giúp hệ thống chống chịu tốt trước các SQL Injection, đảm bảo an toàn dữ liệu.

- Hệ thống đã đáp ứng được hầu hết các yêu cầu chức năng và phi chức năng cơ bản đã đề ra trong kế hoạch kiểm thử

2. Hạn chế và hướng phát triển

2.1. Hạn chế:

Mặc dù đã đạt được các mục tiêu quan trọng, dự án vẫn còn một số hạn chế và tiềm năng để phát triển trong tương lai:

- Đây là một ứng dụng desktop, giới hạn khả năng truy cập (không có phiên bản web hoặc di động).
- Hệ thống hiện tại phù hợp với khách sạn quy mô nhỏ. Khả năng mở rộng cho các chuỗi khách sạn lớn, đa chi nhánh cần được xem xét và phát triển thêm.
- Hệ thống chưa tích hợp với các nền tảng bên ngoài như các trang đặt phòng trực tuyến (OTA), hệ thống thanh toán điện tử phức tạp, hoặc phần mềm kế toán chuyên dụng.
- Tính năng nâng cao:
 - Chưa có module quản lý kho hàng (thực phẩm, đồ dùng khách sạn).
 - Chưa có tính năng quản lý quan hệ khách hàng (CRM) chuyên sâu hoặc chương trình khách hàng thân thiết.
 - Các phân tích dữ liệu còn cơ bản, cần phát triển chuyên sâu hơn về doanh thu, hiệu suất phòng, v.v.
 - Chưa có tính năng định giá phòng động theo thời điểm hoặc nhu cầu.
 - Chưa linh động trong việc xóa, sửa thông tin khách hàng, nhân viên, phòng.
 - Chưa phân luồng một cách tối ưu cho các quyền hoạt động trên hệ thống của lễ tân và admin.

2.2. Hướng phát triển:

- Phát triển phiên bản web hoặc ứng dụng di động để tăng cường khả năng truy cập và quản lý từ xa.

- Tích hợp với các nền tảng đặt phòng trực tuyến phổ biến, cổng thanh toán điện tử và phần mềm kế toán.
- Xây dựng module quản lý khách hàng thân thiết, lịch sử giao dịch để cá nhân hóa trải nghiệm khách hàng.
- Phát triển các dashboard và báo cáo phân tích dữ liệu nâng cao để hỗ trợ ra quyết định chiến lược.
- Thêm module quản lý hàng hóa, vật tư trong khách sạn.
- Liên tục tối ưu hóa mã nguồn và cơ sở dữ liệu để đảm bảo hiệu suất cao khi dữ liệu tăng trưởng.

3. Bài học kinh nghiệm

- Việc hiểu rõ và chi tiết các yêu cầu nghiệp vụ từ đầu là cực kỳ quan trọng để xây dựng một hệ thống đáp ứng đúng nhu cầu người dùng.
- Một thiết kế cơ sở dữ liệu mạnh mẽ, có tính nhất quán và rõ ràng về mối quan hệ giữa các bảng là nền tảng cho sự ổn định và khả năng mở rộng của hệ thống. Các vấn đề về schema cần được giải quyết triệt để từ sớm.
- Việc thực hiện kiểm thử ở mọi giai đoạn (đơn vị, tích hợp, hệ thống) và các loại kiểm thử khác nhau (chức năng, phi chức năng) giúp phát hiện lỗi sớm, giảm chi phí sửa lỗi và nâng cao chất lượng sản phẩm cuối cùng.
- Luôn ưu tiên các biện pháp bảo mật ngay từ giai đoạn thiết kế và mã hóa, đặc biệt là việc sử dụng truy vấn tham số hóa để chống lại các lỗ hổng phổ biến như SQL Injection.
- Việc ghi nhận, theo dõi và xử lý lỗi một cách có hệ thống, cùng với việc thu thập phản hồi từ người dùng/kiểm thử viên, là chìa khóa để cải thiện sản phẩm.
- Đối với các dự án lớn, sự phối hợp nhịp nhàng giữa các thành viên trong nhóm là yếu tố quyết định thành công.
- Lĩnh vực công nghệ thông tin luôn thay đổi, việc không ngừng học hỏi các công nghệ, công cụ và phương pháp mới là cần thiết để nâng cao chất lượng dự án.

TÀI LIỆU THAM KHẢO

[1] Patel, Pratik. *Java Database Programming with JDBC*. Coriolis, The Coriolis Group, 1996.

[2] Meet Ahir, Amin Shaikh. “Hotel Management System: Enhancing Hospitality through Digital Transformation”. *International Journal of Innovative Research in Science Engineering and Technology*, tập 14, số 4, 2025, tr. 3693 - 3695.

[3] <https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html>

(Oracle, Lesson: JDBC Basics), truy cập ngày 01/06/2025.

[4] https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

(OWASP Cheat Sheet Series, SQL Injection Prevention Cheat Sheet), truy cập ngày 01/06/2025.

[5] <https://techmaster.vn/posts/36976/huong-dan-su-dung-jdbc-ket-noi-co-so-du-lieu-trong-java>

(TeachMaster, 2022, Hướng dẫn sử dụng JDBC kết nối cơ sở dữ liệu trong Java), truy cập ngày 01/06/2025.

[6] <https://www.geeksforgeeks.org/java-database-connectivity-with-mysql/>

(GeeksForGeeks, 2023, Java Database Connectivity with MySQL), truy cập ngày 01/06/2025.

[7] <https://github.com/kunaltyagi9/Hotel-Management-System>

(Github, 2020, Hotel-Management-System), truy cập ngày 01/06/2025.

[8] <https://codestar.vn/tai-lieu-dac-ta-srs-trong-phan-tich-yeu-cau/>

(CodeStar, Tài liệu đặc tả SRS trong phân tích yêu cầu), truy cập ngày 03/06/2025.

[9] <https://fr.scribd.com/document/794783797/Phan-tich-thi%E1%BA%BFt-k%E1%BA%BF-h%E1%BB%87-th%E1%BB%91ng-thong-tin-qu%E1%BA%A3n-li-khach-s%E1%BA%A1n>

(Scribd, 2024, Phân tích thiết kế hệ thống thông tin quản lý khách sạn), truy cập ngày 03/06/2025.

[10] <https://fr.scribd.com/document/866820394/Yeu-C%E1%BA%A7u-Phi-Ch%E1%BB%A9c-N%C4%83ng-C%E1%BB%A7a-H%E1%BB%87-Th%E1%BB%91ng-Qu%E1%BA%A3n-L%C3%BD-Khach-S%E1%BA%A1n>

(Scribd, 2025, Yêu Cầu Phi Chức Năng Của Hệ Thống Quản Lý Khách Sạn), truy cập ngày 03/06/2025.

[11] <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

(Visual Paradigm, What is Sequence Diagram?), truy cập ngày 03/06/2025.

[12] <https://fr.scribd.com/document/744642830/Hotel-Management-System-A-GUI-using-JAVA-SWING>

(Scribd, 2024, Hotel Management System: A GUI Using JAVA SWING), truy cập ngày 03/06/2025.

[13] <https://iviettech.vn/category/blog/phan-tich-thiet-ke-he-thong>

(IViettech, Phân tích thiết kế hệ thống), truy cập ngày 04/06/2025.