

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**XÂY DỰNG ỨNG DỤNG QUẢN LÝ
KHÁCH SẠN – HOMESTAY
BẰNG THƯ VIỆN JAVA SWING**

Sinh viên thực hiện:		
STT	Họ tên	MSSV
1	Hồ Nhất Thống	19522300
2	Nguyễn Phạm Nhật Trường	19522441

TP. HỒ CHÍ MINH – 06/2023

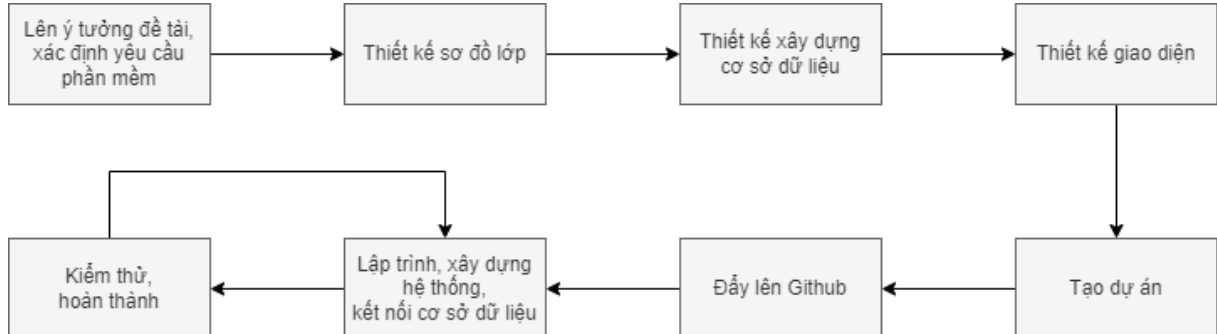
1. GIỚI THIỆU

Đề tài chúng em là về xây dựng hệ thống ứng dụng quản lý khách sạn – homestay. Ứng dụng có thể dùng để quản lý nhiều khách sạn và nhiều homestay của một doanh nghiệp nhờ vào tính chất hai loại nhà ở này khá giống nhau, khác nhau ở chỗ là khách sạn nhiều loại phòng hơn homestay. Mục tiêu chúng em khi hoàn thành đề tài này là ứng dụng có chức năng giúp cho các quản lý, nhân viên của khách sạn có thể quản lý và vận hành khách sạn hiệu quả như quản lý đơn đặt phòng, thông tin khách hàng, dịch vụ, nhân viên, tạo hóa đơn thanh toán. Định hướng đề tài xa hơn nữa là thay vì chỉ thêm dữ liệu phòng ở, quản lý có thể thêm dữ liệu khách sạn hoặc homestay nếu có xây thêm và quản lý chúng. Ứng dụng được xây dựng bằng thư viện Java Swing sử dụng mô hình MVC với sự hỗ trợ của một số phần mềm như công cụ thiết kế sơ đồ lớp StarUML, công cụ thiết kế giao diện Figma, công cụ thiết kế cơ sở dữ liệu MySQL Workbench, công cụ Github hỗ trợ làm việc nhóm và các thư viện hỗ trợ xây dựng ứng dụng ngôn ngữ Java như Java Swing, Jcalendar, MySQL connector. Kết quả đạt được ở hiện tại là chúng em đã hoàn thành ứng dụng với các chức năng có thể giúp nhân viên khách sạn thực hiện công việc cho khách hàng đặt ở và sử dụng dịch vụ nhưng chưa hoàn thành được các chức năng dành cho quản lý như quản lý thông tin nhân viên.

Cam kết: Đề tài chúng em có tham khảo cách tương tác với cơ sở dữ liệu ở phần Model (minh chứng dự án tham khảo ở phần [1] tài liệu tham khảo), còn lại nhóm chúng em cam kết tự thiết kế và xây dựng ạ.

2. XÂY DỰNG HỆ THỐNG

2.1. Quy trình thực hiện



Hình 1: Quy trình thực hiện xây dựng ứng dụng quản lý khách sạn – homestay

- Bước 1: Lên ý tưởng đề tài, xác định yêu cầu phần mềm:

Lên ý tưởng về việc xây dựng ứng dụng về đề tài khách sạn, homestay. Xác định các yêu cầu phần mềm, chức năng cần thiết cho ứng dụng như: màn hình quản lý đơn đặt phòng, dịch vụ, khách hàng, phòng, hóa đơn và các chức năng thao tác cần thiết trên dữ liệu đó như thêm, xóa, sửa, tính tổng hóa đơn theo mã khách hàng.

- Bước 2: Thiết kế sơ đồ lớp:

Thiết kế các đối tượng model, các lớp, các thuộc tính và các hàm xử lý dữ liệu cần thiết của các lớp đó bằng công cụ StarUML.

- Bước 3: Thiết kế xây dựng cơ sở dữ liệu:

Thiết kế và xây dựng các bảng đối tượng của cơ sở dữ liệu quản lý khách sạn homestay, các thuộc tính, các mối quan hệ, ràng buộc bằng công cụ MySQL Workbench.

- Bước 4: Thiết kế giao diện:

Tiếp theo, chúng em dùng công cụ Figma thiết kế bố cục, giao diện cho ứng dụng để thống nhất cách xây dựng phần View khi bắt tay vào lập trình.

- Bước 5: Tạo dự án:

Chúng em sử dụng công cụ NetBeans để tạo dự án sơ bộ theo mô hình MVC sử dụng thư viện Java Swing.

- Bước 6: Đẩy lên Github:

Sau đó, tạo nhóm dự án cho đồ án môn học trên Github để thực hiện việc lưu trữ mã nguồn dự án và thuận lợi cho việc hai thành viên đẩy code lên, quản lý tiến độ. Chúng em sẽ phân chia ra cách phần cần xây dựng và tách nhánh từ nhánh main để code, sau khi code một phần nào đó đã hoàn thành thì merge nhánh đó vào nhánh main.

- Bước 7: Lập trình, xây dựng hệ thống, kết nối cơ sở dữ liệu:

Sau khi đẩy code lên nhóm dự án trên Github, chúng em bắt tay vào xây dựng hệ thống theo mô hình MVC. Tạo tài khoản, mật khẩu để thực kết nối cơ sở dữ liệu trên MySQL Workbench.

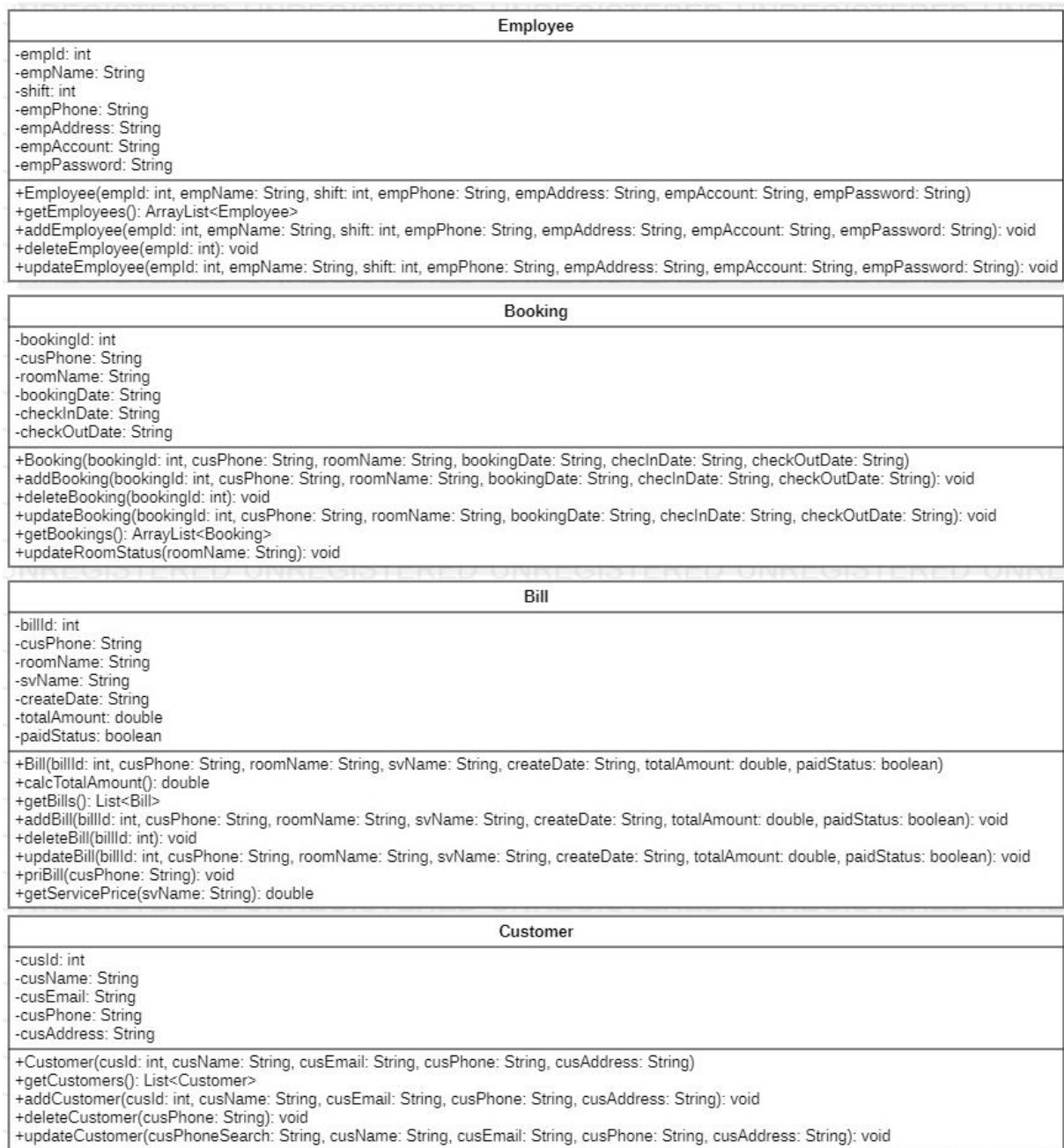
- Bước 8: Kiểm thử, hoàn thành:

Sau khi qua bước xây dựng, chúng em kiểm tra lại ứng dụng xem các chức năng liệu có hoạt động tốt cùng cơ sở dữ liệu không, nếu có sai sót thì quay lại bước 7 để sửa chữa. Nếu ứng dụng đã hoạt động tốt thì hoàn thành sản phẩm.

2.2. Sơ đồ lớp quản lý khách sạn - homestay

HotelHomeStay	Service
<div>-htId: String -htName: String -htAddress: String -roomCount: int -roomList: ArrayList<Room> +HotelHomeStay(htId: String, htName: String, htAddress: String, roomCount: int) +roomSearch(roomId: int): Room</div>	<div>-svId: int -svName: String -svPrice: double +Service(svId: int, svName: String, svPrice: double) +addService(svId: int, svName: String, svPrice: double): void +updateService(svId: int, svName: String, svPrice: double): void +deleteService(svId: int): void +getServices(): void</div>
Room	
<div>-roomId: int -roomName: String -roomType: String -roomRates: double -roomStatus: boolean +Room(roomId: int, roomName: String, roomType: String, roomRates: double, roomStatus: boolean) +addRoom(roomId: int, roomName: String, roomType: String, roomRates: double, roomStatus: boolean): void +deleteRoom(roomId: int): void +updateRoom(roomId: int, roomName: String, roomType: String, roomRates: double, roomStatus: boolean): void +getRooms(): ArrayList<Room></div>	

Hình 2: Sơ đồ lớp khách sạn, phòng, nhân viên



Hình 3: Sơ đồ lớp dịch vụ, đặt phòng, hóa đơn, khách hàng

Chúng em dùng công cụ StarUML để thiết kế sơ đồ lớp, các thuộc tính và các hàm xử lý của các lớp đó. Sau đó tạo code cho phần Model trong mô hình MVC với mục đích để phần Model có thể lấy dữ liệu từ cơ sở dữ liệu, xử lý và lưu trữ dữ liệu.

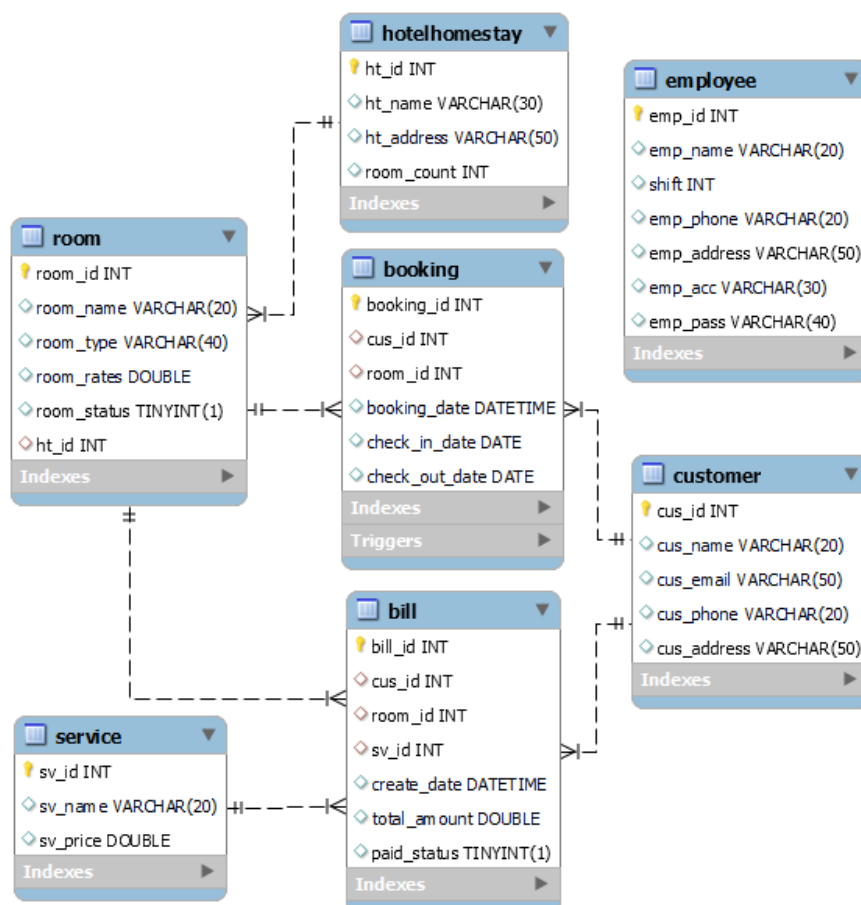
Các lớp bao gồm:

- HotelHomeStay: Lớp chứa thuộc tính và hành động để truy xuất, xử lý dữ liệu về khách sạn hoặc homestay. Lớp này chứa nhiều dữ liệu về các phòng.
- Room: Lớp chứa thuộc tính và hành động để truy xuất, xử lý dữ liệu về phòng ở. Có thể xem trạng thái của phòng để chọn cho khách đặt.

- Customer: Lớp chứa thuộc tính và hành động để truy xuất, xử lý dữ liệu về khách hàng. Lưu dữ liệu khách hàng tạo sự tiện lợi cho khách sạn nếu khách hàng tiếp tục sử dụng khách sạn trong tương lai.
- Employee: Lớp chứa thuộc tính và hành động để truy xuất, xử lý dữ liệu về nhân viên. Giúp nhân viên biết ca làm việc và cho người quản lý khách sạn có thể quản lý thông tin nhân viên.
- Service: Lớp chứa thuộc tính và hành động để truy xuất, xử lý dữ liệu về dịch vụ cho khách hàng lựa chọn sử dụng.
- Booking: Lớp chứa thuộc tính và hành động để truy xuất, xử lý dữ liệu về đơn đặt phòng, khách hàng có thể đặt trước và nhờ nhân viên đặt giúp hoặc đặt phòng trực tiếp tại quầy tiếp tân.
- Bill: Lớp chứa thuộc tính và hành động để truy xuất, xử lý dữ liệu về hóa đơn, giúp nhân viên có thể thêm chi tiết hóa đơn khi khách hàng sử dụng phòng hay các dịch vụ. Sau đó khi khách trả phòng, nhân viên sẽ dùng chức năng tính tổng hóa đơn và xuất tổng tiền khách phải thanh toán.

Hầu hết các lớp đều có các hàm xử lý khởi tạo lớp, lấy danh sách, thêm xóa hoặc sửa các lớp đó.

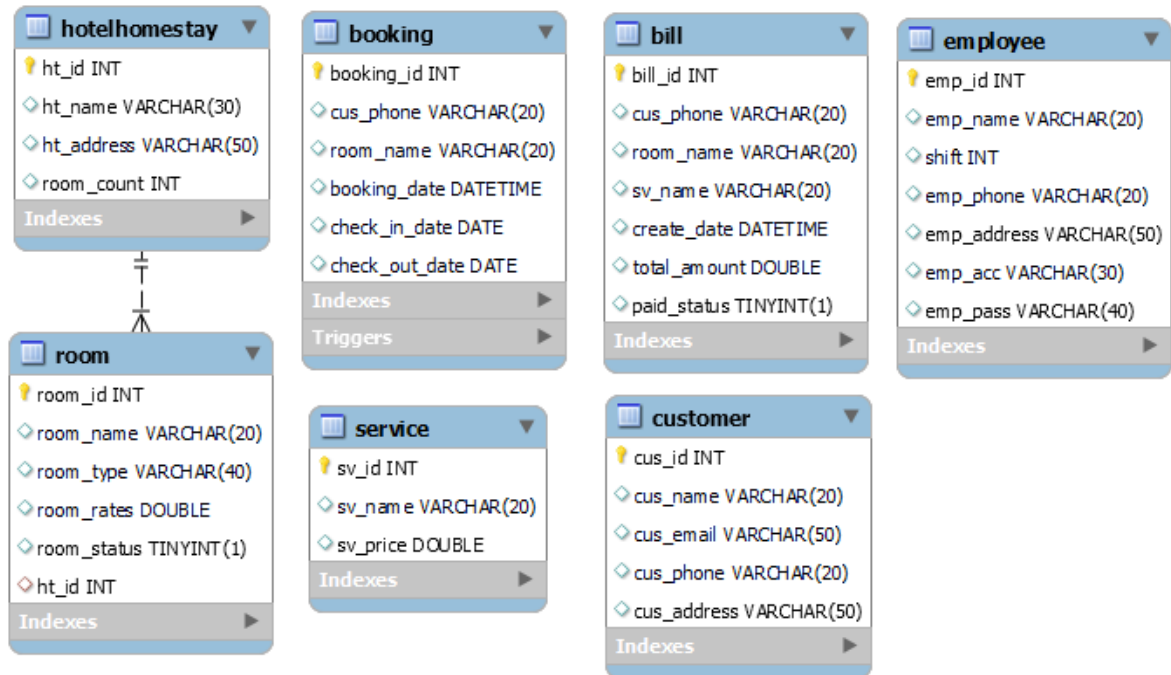
2.3. Cơ sở dữ liệu



Hình 4. Mô hình quan hệ cơ sở dữ liệu quản lý khách sạn – homestay (cũ)

Trên đây là mô hình quan hệ cơ sở dữ liệu khi chúng em cho tạo nhập các đơn đặt phòng hay hóa đơn bằng mã của khách hàng, dịch vụ và phòng. Các phần tìm kiếm cũng tìm bằng mã nên khi sử dụng ứng dụng gây bất tiện, khó ghi nhớ mã (ID).

Dưới đây là mô hình quan hệ cơ sở dữ liệu mới, chúng em thiết kế lại để có thể tạo nhập đặt phòng hoặc hóa đơn bằng tên phòng, tên dịch vụ hay số điện thoại khách hàng, tìm sửa xóa phòng bằng tên phòng, tính tổng hóa đơn theo số điện thoại khách hoặc tìm sửa xóa khách hàng bằng số điện thoại, giúp thuận tiện hơn. Sau khi sửa đổi, chúng em nhận thấy thuộc tính mã phòng, mã khách hàng, mã dịch vụ trong bảng đặt phòng và bảng hóa đơn chưa cần thiết nên chúng em đã bỏ bớt.



Hình 5. Mô hình quan hệ cơ sở dữ liệu quản lý khách sạn – homestay (mới)

Cơ sở dữ liệu quản lý khách sạn homestay bao gồm các quan hệ:

- HOTELHOMESTAY (ht_id, ht_name, ht_address, room_count)

Quan hệ HOTELHOMESTAY lưu trữ thông tin về khách sạn, chỗ ở gồm các thuộc tính: mã khách sạn homestay, tên, địa chỉ, số lượng phòng của khách sạn.

- ROOM (room_id, room_name, room_type, room_rates, room_status, ht_id)

Quan hệ ROOM lưu trữ thông tin về phòng ở gồm các thuộc tính: mã phòng, tên, loại phòng, giá phòng, trạng thái. Mỗi phòng đều thuộc một khách sạn và có mã khách sạn đó. Phòng có 4 loại là Single bed room (1 giường) giá 100.000đ/ngày, Twin bed room (2 giường) giá 250.000đ/ngày, Double bed room (1 giường dành cho 2 người) giá 250.000đ/ngày, Triple bed room (3 giường nhỏ hoặc 1 giường 2 người + 1 giường nhỏ) giá 300.000đ/ngày.

- CUSTOMER (cus_id, cus_name, cus_email, cus_phone, cus_address)

Quan hệ CUSTOMER lưu trữ thông tin về khách hàng gồm các thuộc tính: mã khách hàng, họ tên, email, số điện thoại, địa chỉ khách hàng.

- EMPLOYEE (emp_id, emp_name, shift, emp_phone, emp-address, emp_acc, emp_pass)

Quan hệ EMPLOYEE lưu trữ thông tin về nhân viên gồm các thuộc tính: mã nhân viên, tên, ca làm việc, số điện thoại, địa chỉ, tên đăng nhập ứng dụng và mật khẩu. Mỗi nhân viên trực bàn đều có tài khoản để sử dụng ứng dụng với quyền nhân viên.

- SERVICE (sv_id, sv_name, sv_price)

Quan hệ SERVICE lưu trữ thông tin về dịch vụ gồm các thuộc tính: mã dịch vụ, tên, giá dịch vụ. Dịch vụ này người quản lý có thể thêm xóa sửa dịch vụ để cho khách hàng sử dụng. Bao gồm các dịch vụ như bữa ăn 40.000đ/bữa, đưa đón 5000đ/km, hồ bơi 10.000đ/giờ.

- BOOKING (booking_id, booking_date, check_in_date, check_out_date, cus_phone, room_name)

Quan hệ BOOKING lưu trữ thông tin đặt phòng gồm các thuộc tính: mã đặt phòng, ngày giờ đặt, ngày nhận phòng, ngày trả phòng. Mỗi mục đặt phòng được một khách hàng yêu cầu nên sẽ có số điện thoại khách hàng đó và tên phòng khách hàng đặt.

- BILL (bill_id, create_date, total_amount, paid_status, cus_phone, room_name, sv_name)

Quan hệ BILL lưu trữ thông tin về hóa đơn gồm các thuộc tính: mã hóa đơn, ngày tạo, tổng tiền phải thanh toán, trạng thái thanh toán. Mỗi hóa đơn được một khách hàng thanh toán nên sẽ có số điện thoại khách hàng đó, tên phòng đã đặt và tên dịch vụ đã sử dụng.

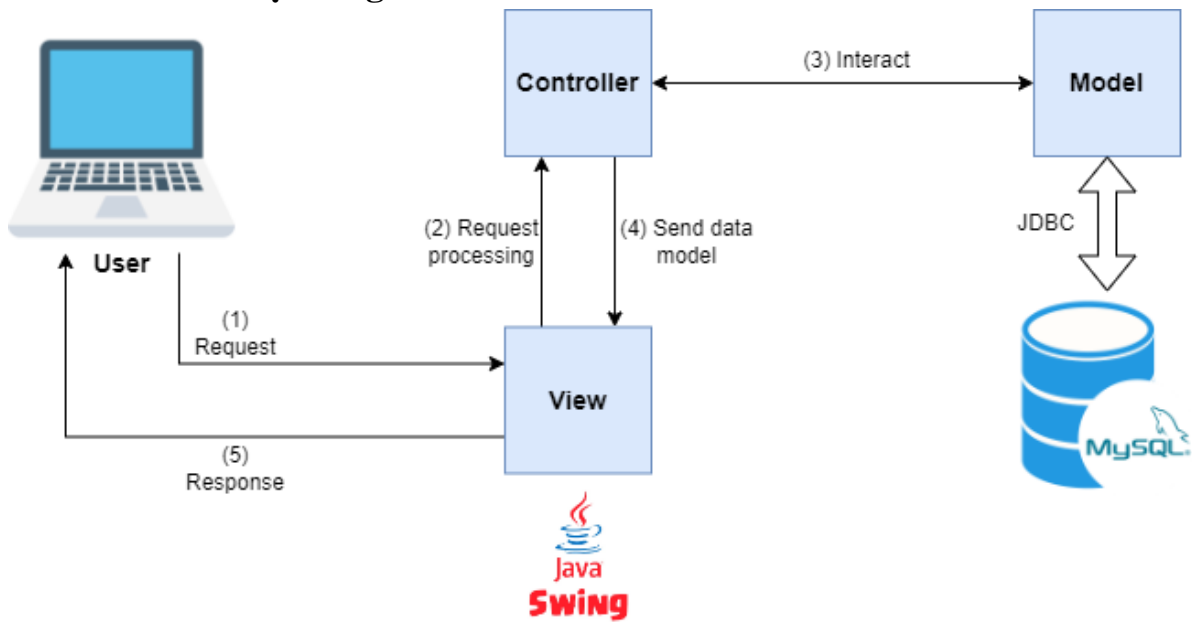
Các ràng buộc trong cơ sở dữ liệu:

- Trong thông tin đặt phòng (BOOKING), ngày nhận phòng (check_in_date) của khách hàng phải bé hơn hoặc bằng ngày trả phòng (check_out_date).
- Nếu trong thông tin của phòng (ROOM) có trạng thái (room_status) là true, nghĩa là đã có người đặt thì không được đặt phòng đó cho khách khác.

Chức năng:

- Thêm - xóa - sửa khách hàng, nhân viên, phòng, dịch vụ, đơn đặt, hóa đơn.
- Tính tổng hóa đơn của khách hàng.

2.4. Kiến trúc hệ thống



Hình 6: Ứng dụng quản lý khách sạn homestay theo mô hình MVC

Mô hình MVC là kiến trúc thiết kế phần mềm phổ biến bao gồm 3 thành phần:

- Model: có chức năng truy xuất, quản lý, xử lý và lưu trữ dữ liệu từ cơ sở dữ liệu MySQL thông qua giao thức JDBC (Java Database Connectivity).
- View: có chức năng hiển thị giao diện, các components, dữ liệu cho người dùng để người dùng tương tác. View không xử lý logic mà chỉ hiển thị dữ liệu nhận từ Model. Trong đề tài này, chúng em sử dụng thư viện Java Swing để thiết kế phần View. Java Swing là một thư viện hỗ trợ thiết kế giao diện.
- Controller: có chức năng điều khiển tương tác giữa Model và View. Controller là nơi nhận yêu cầu từ View và truy cập, xử lý các yêu cầu với dữ liệu từ Model rồi cập nhật đến Model, sau đó hiển thị qua View.

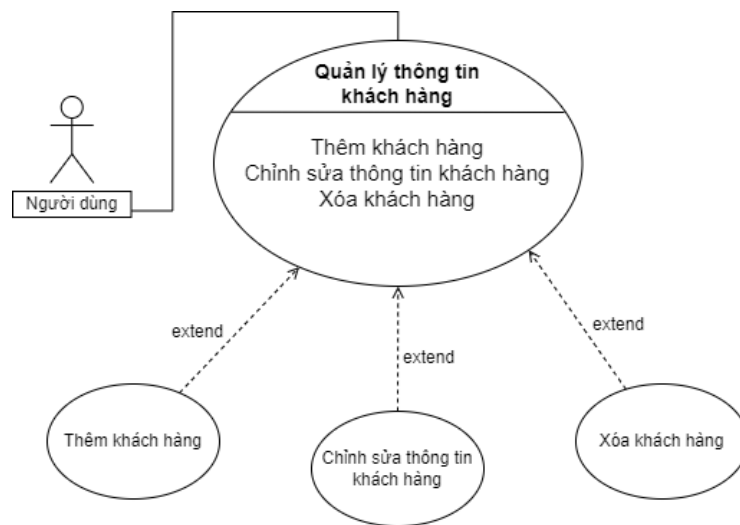
Ưu điểm của kiến trúc này là giúp tách các phần xử lý logic, dữ liệu và giao diện ra riêng biệt, code các phần sẽ dễ đọc và sửa chữa hơn. Nhược điểm là việc tách riêng các phần có thể tăng độ dài dòng phức tạp của code đối với người mới tiếp cận mô hình, tăng số lượng tệp tin và mất nhiều thời gian để bảo trì.

Quy trình kiến trúc hệ thống hoạt động:

- Đầu tiên, người dùng sẽ tương tác, gửi yêu cầu qua giao diện (View).
- Tiếp theo, View gửi yêu cầu cho Controller, Controller nhận yêu cầu.
- Controller tương tác với Model để xử lý yêu cầu.
- Model truy xuất, xử lý và cập nhật dữ liệu từ cơ sở dữ liệu thông qua giao thức JDBC.
- Controller tương tác lấy kết quả từ Model và gửi dữ liệu cho View.
- Cuối cùng, View trả về kết quả cho người dùng qua giao diện.

2.5. Đặc tả use case

2.5.1. Quản lý thông tin khách hàng

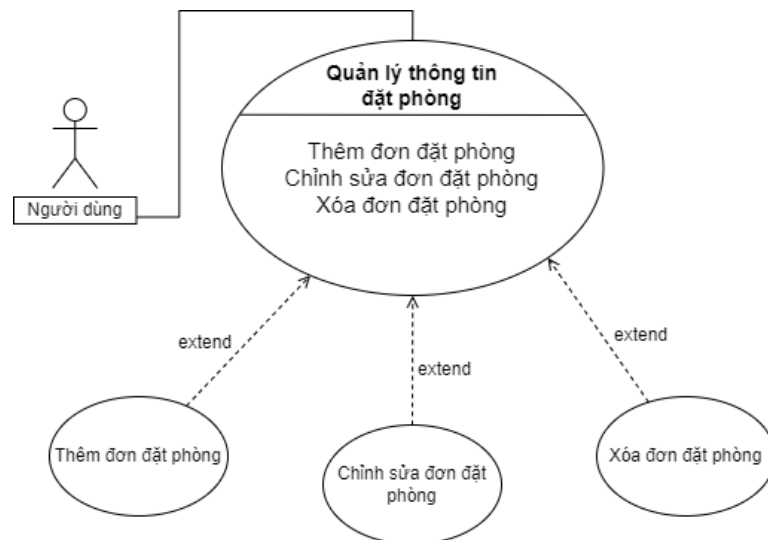


Hình 7: Use case quản lý thông tin khách hàng

Quản lý thông tin khách hàng	
Tên use case	Quản lý thông tin khách hàng
Tóm tắt	Người dùng có thể thêm, sửa, xóa bằng các nút chức năng Add, Delete, Update.
Người thực hiện	Người quản lý, người dùng nhân viên.
Điều kiện tiền xử lý	Nhập các trường thông tin khách hàng muốn thêm, sửa, xóa.
Sau xử lý	Trả về thông báo và danh sách khách hàng đã cập nhật.
Dòng sự kiện chính	1/ Vào trang đăng nhập ứng dụng. 2/ Chọn nút Customer. 3/ Nhập các thông tin yêu cầu và ấn nút.
Dòng sự kiện khác	Nếu thêm, sửa, xóa không thành công sẽ trả về thông báo “Unsuccessful ...!”.

Bảng 1: Mô tả Use case quản lý thông tin khách hàng

2.5.2. Quản lý thông tin đặt phòng

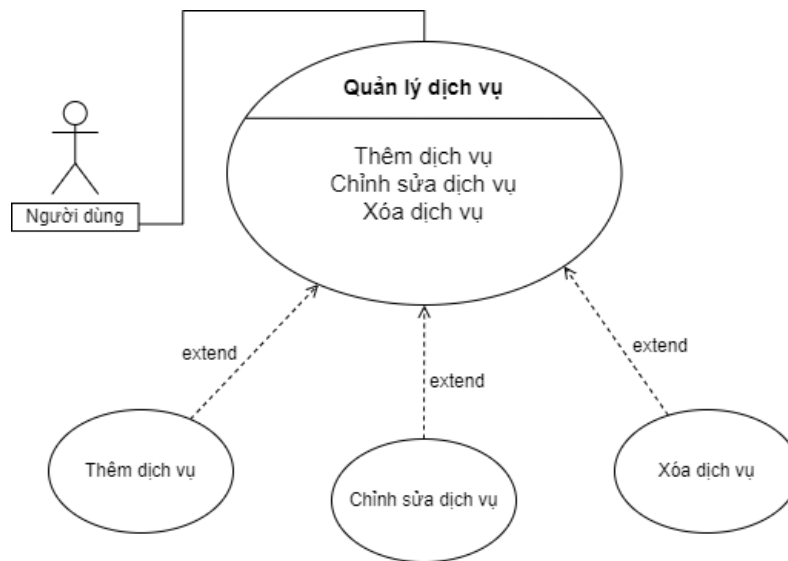


Hình 8: Use case quản lý thông tin đặt phòng

Quản lý thông tin đặt phòng	
Tên use case	Quản lý thông tin đặt phòng
Tóm tắt	Người dùng có thể thêm, sửa, xóa bằng các nút chức năng Add, Delete, Update.
Người thực hiện	Người quản lý, người dùng nhân viên.
Điều kiện tiền xử lý	Nhập các trường thông tin cần thiết của đơn đặt phòng muốn thêm, sửa, xóa.
Sau xử lý	Trả về thông báo và danh sách đặt phòng đã được cập nhật.
Dòng sự kiện chính	1/ Vào trang đăng nhập ứng dụng. 2/ Nhập các thông tin yêu cầu và ấn nút.
Dòng sự kiện khác	Nếu thêm, sửa, xóa không thành công sẽ trả về thông báo “Unsuccessful ...!”.

Bảng 2: Mô tả Use case quản lý thông tin đặt phòng

2.5.3. Quản lý dịch vụ

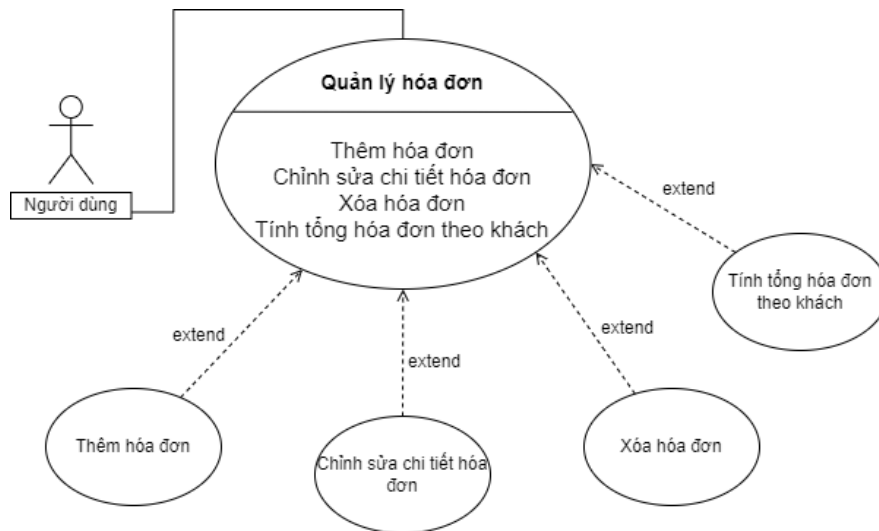


Hình 9: Use case quản lý dịch vụ

Quản lý dịch vụ	
Tên use case	Quản lý thông tin dịch vụ
Tóm tắt	Người dùng có thể thêm, sửa, xóa bằng các nút chức năng Add, Delete, Update.
Người thực hiện	Người quản lý, người dùng nhân viên.
Điều kiện tiên xử lý	Nhập các trường thông tin cần thiết của dịch vụ muốn thêm, sửa, xóa.
Sau xử lý	Trả về thông báo và danh sách dịch vụ đã cập nhật.
Dòng sự kiện chính	1/ Vào trang đăng nhập ứng dụng. 2/ Chọn nút Service. 3/ Nhập các thông tin yêu cầu và ấn nút.
Dòng sự kiện khác	Nếu thêm, sửa, xóa không thành công sẽ trả về thông báo “Unsuccessful ...!”.

Bảng 3: Mô tả Use case quản lý dịch vụ

2.5.4. Quản lý hóa đơn



Hình 10: Use case quản lý hóa đơn

Quản lý chi tiết hóa đơn	
Tên use case	Quản lý chi tiết hóa đơn
Tóm tắt	Người dùng có thể thêm, sửa, xóa bằng các nút chức năng Add, Delete, Update.
Người thực hiện	Người quản lý, người dùng nhân viên.
Điều kiện tiên xử lý	Nhập các trường thông tin cần thiết của hóa đơn muốn thêm, sửa, xóa.
Sau xử lý	Trả về thông báo và danh sách hóa đơn đã được cập nhật.
Dòng sự kiện chính	1/ Vào trang đăng nhập ứng dụng. 2/ Chọn nút Bill. 3/ Nhập các thông tin yêu cầu và ấn nút.
Dòng sự kiện khác	Nếu thêm, sửa, xóa không thành công sẽ trả về thông báo “Unsuccessful ...!”.

Bảng 4: Mô tả Use case quản lý thông tin đặt phòng

3. MỞ RỘNG THÊM

Các phần làm mở rộng thêm cho dự án:

- Phần quản lý phòng, chuyển tìm kiếm bằng mã phòng sang bằng tên phòng vì tên phòng dễ ghi nhớ hơn.
- Phần quản lý khách hàng, chuyển tìm kiếm khách hàng bằng mã sang bằng số điện thoại để thuận tiện hơn.
- Phần quản lý đặt phòng, chỉnh sửa chỗ tạo đơn đặt phòng ở phần nhập mã phòng và mã khách sang thành nhập tên phòng và số điện thoại khách hàng.

- Phân quản lý hóa đơn, chỗ tạo hóa đơn sửa nhập mã khách hàng, mã phòng, mã dịch vụ sang thành nhập số điện thoại khách hàng, tên phòng và tên dịch vụ. Tính tổng hóa đơn theo số điện thoại thay vì mã của khách hàng.
- Thêm phần View thống kê doanh thu theo năm và phần Controller để xử lý màn hình thống kê doanh thu.

4. KẾT LUẬN

Trong thời gian 2 tuần, với tuần thứ nhất chúng em đã lên ý tưởng đề tài, thiết kế giao diện, thiết kế sơ đồ lớp, xây dựng cơ sở dữ liệu. Tuần thứ hai chúng em dành thời gian lập trình ứng dụng và viết báo cáo. Hiện nhóm chúng em đã xây dựng được một ứng dụng có khả năng phục vụ cho nghiệp vụ của nhân viên tiếp tân khách sạn trong phạm vi duy nhất một khách sạn hoặc homestay, ứng dụng có màn hình đăng nhập, màn hình quản lý đơn đặt phòng, quản lý phòng, khách hàng, dịch vụ và hóa đơn. Cùng với các chức năng cơ bản đáp ứng được quản lý và vận hành khách sạn - homestay như thêm, sửa hoặc xóa thông tin khách hàng, phòng, đơn đặt phòng, dịch vụ, hóa đơn, tính tổng tiền hóa đơn của khách hàng. Những điều chưa đạt được là ứng dụng chỉ mới hỗ trợ được công việc của nhân viên, chưa hỗ trợ được nghiệp vụ của người quản lý khách sạn như quản lý, thêm sửa xóa khách sạn, các loại homestay và nhân viên. Bên cạnh đó, trong quá trình thực hiện đồ án chúng em đã nhận ra những thiếu sót của mình trong việc quản lý thời gian thực hiện đồ án, chúng em đã trì hoãn khoảng một tuần, kém hiệu quả khi làm việc nhóm vì chủ yếu chỉ trao đổi qua tin nhắn. Hướng khắc phục của chúng em là lên kế hoạch đặt mục tiêu cho các thành viên tập trung thực hiện đồ án, sử dụng công cụ quản lý dự án để tiết kiệm thời gian, làm việc nhóm trực tiếp nhiều hơn.

TÀI LIỆU THAM KHẢO

- [1] Dự án tham khảo. Link: https://github.com/nguyendathuyhoang/java_IS216.M22_1?fbclid=IwAR0gJRRI6BRzWQPO91WNexBwdGMcXzLn8nOQ_Eb4eVOX53im59Z_qyqNkBM (12/6/2023).
- [2] LANIT. Link: <https://lanit.com.vn/mo-hinh-mvc-trong-java-la-gi.html> (1/6/2023).
- [3] VietTuts. Link: <https://viettuts.vn/java-jdbc> (29/5/2023).
- [4] Toedter.com. Link: <https://toedter.com/jcalendar/> (22/5/2023).
- [5] W3Schools. Link: <https://www.w3schools.com/MySQL/default.asp> (1/6/2023).

PHỤ LỤC PHÂN CÔNG NHIỆM VỤ

STT	Thành viên	Nhiệm vụ
1	Hồ Nhất Thống	<ul style="list-style-type: none">– Thiết kế giao diện Figma phần quản lý Service, Employee, Room, Booking.– Thiết kế sơ đồ lớp trên StarUML.– Xây dựng ứng dụng phần Model, View, Controller của Service, Employee, Room.– Làm slide thuyết trình.– Thuyết trình.– Phần mở rộng: Xây dựng thêm Controller và View của Statistic (thống kê doanh thu).
2	Nguyễn Phạm Nhật Trường	<ul style="list-style-type: none">– Thiết kế giao diện Figma phần Login, Customer, Bill.– Thiết kế xây dựng cơ sở dữ liệu MySQL.– Xây dựng ứng dụng phần Model, View, Controller của Booking, Customer, Bill, Login.– Viết nội dung báo cáo word, pdf và vẽ hình ảnh quy trình, mô hình.– Thuyết trình.– Phần mở rộng: Chỉnh sửa các phần tìm kiếm, tạo nhập bằng ID sang tìm kiếm bằng tên hoặc số điện thoại.

PHỤ LỤC DEMO

Chạy sản phẩm các bước sau:

Bước 1: Tại dự án.

- Mở file Main.java ở packpage View của dự án.
- Nhấn chuột phải chọn run file.

Bước 2: Đăng nhập.

- Khi màn hình đăng nhập mở ra, nhập user name là “nhanvienca1”, password là “123456789@Ca1”.
- Nhấn nút Login.

Bước 3: Quản lý khách hàng.

- Màn hình quản lý khách hàng hiện ra, nhập các trường cần thiết để thêm khách hàng, sau đó ấn nút Add.
- Muốn cập nhật khách hàng, nhập số điện thoại khách ở ô search phone number. Tiếp theo, nhập các trường cần thiết rồi ấn nút Update.
- Muốn xóa khách hàng, nhập số điện thoại khách ở ô search phone number rồi ấn nút Delete.

Bước 4: Quản lý đặt phòng.

- Nhấn nút Booking trên menu để qua màn hình quản lý đặt phòng.
- Khi thêm đặt phòng, nhập các thông tin vào các ô ở form Booking.
- Ấn nút Add để thêm đặt phòng, nếu ô Room name nhập tên của phòng có trạng thái true tức là có người đang đặt hoặc nhập ngày check in lớn hơn ngày check out thì không thể thêm đặt phòng.
- Sau khi thêm đơn đặt một phòng, phòng đó sẽ tự chuyển trạng thái từ false sang true để báo hiệu đã có người đặt.
- Để cập nhật đơn đặt, ta nhập mã đơn đặt vào ô Booking ID search và nhập các thông tin cập nhật đầy đủ, sau đó ấn nút Update.
- Để xóa, ta nhập mã đơn đặt vào ô Booking ID search và ấn nút Delete.

Bước 5: Quản lý phòng.

- Ta nhấn nút Room trên menu để qua màn hình quản lý phòng.
- Khi thêm phòng, nhập các thông tin vào các ô ở form Room Information. Ở checkbox Not available, ta nhấn vào để cài đặt trạng thái là true cho biết phòng đang có lịch đặt hoặc có người ở và ngược lại để trống là trạng thái false. Ấn nút Add để thêm phòng.
- Để cập nhật phòng, ta nhập tên phòng vào ô search room name và nhập các thông tin cập nhật đầy đủ, sau đó ấn nút Update.
- Để xóa, ta nhập tên phòng vào ô search room name và ấn nút Delete.

Bước 6: Quản lý dịch vụ.

- Ta nhấn nút Service trên menu để qua màn hình quản lý dịch vụ.
- Khi thêm dịch vụ, nhập các thông tin vào các ô ở form Service Details và ấn nút Add để thêm dịch vụ.
- Để cập nhật dịch vụ, ta nhập mã dịch vụ vào ô Service ID search và nhập các thông tin cập nhật đầy đủ, sau đó ấn nút Update.
- Để xóa, ta nhập mã dịch vụ vào ô Service ID search và ấn nút Delete.

Bước 7: Quản lý hóa đơn.

- Ta nhấn nút Bill trên menu để qua màn hình quản lý hóa đơn.
- Khi thêm hóa đơn, nhập các thông tin vào các ô ở form Bill create và ấn nút Add để thêm hóa đơn.
- Để cập nhật hóa đơn, ta nhập mã hóa đơn vào ô Bill ID search và nhập các thông tin cập nhật đầy đủ, sau đó ấn nút Update.
- Để xóa, ta nhập mã hóa đơn vào ô Bill ID search và ấn nút Delete.
- Để hiện thông báo tính tổng tiền cho khách hàng, ta nhập số điện thoại khách vào ô Customer Phone và ấn nút Calculate.
- Thông báo số điện thoại và số tiền cần thanh toán của khách hàng hiện lên. Sau khi ấn Ok của thông báo thì trạng thái thanh toán ở các hóa đơn có số điện thoại của khách hàng đó từ false sang true tức là đã thanh toán và phòng của khách hàng đó chuyển trạng thái từ true sang false nghĩa là khách đã trả phòng.

Bước 8: Xem doanh thu.

- Ta nhấn nút Statistics góc dưới trái danh sách hóa đơn của màn hình quản lý hóa đơn để qua màn hình thống kê doanh thu.
- Khi màn hình thống kê doanh thu hiện lên, ta nhập năm muốn tính doanh thu vào ô Enter year và ấn nút Confirm.
- Số doanh thu sẽ hiện bên dưới ô Total revenue of the year. Ấn Exit để trở lại màn hình quản lý hóa đơn.