
Capybara Group - Capbr.G

BookaStay

Software Architecture Document

Version 1.0.0.0

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

Revision History

Date	Version	Description	Author
01/12/2024	0.1.0.0	Release document	T.T.Ngan
10/12/2024	0.1.0.1	Edit Layout	L.T.Nghia
13/12/2024	0.1.1.1	Edit the “Introduction” section	L.T.Nghia
13/12/2024	0.1.1.2	Edit the “Architectural Goals ...” section	L.T.Nghia
13/12/2024	0.1.1.3	Edit the “Use Case Model”	T.N.Luong
13/12/2024	0.1.1.4	Edit the “Logical View”	All member
27/12/2024	1.0.0.0	Last edited	All member

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

Table of Contents

- 1. Introduction.....5**
 - 1.1 Purpose.....5
 - 1.2 Scope.....5
 - 1.3 Overview.....5
- 2. Architectural Goals and Constraints.....5**
- 3. Use-Case Model.....6**
- 4. Logical View.....9**
 - 4.1 Back-end.....9
 - 4.1.1 UserModule.....10
 - 4.1.2 AuthModule.....11
 - 4.1.3 RoleModule.....12
 - 4.1.4 HotelModule.....13
 - 4.1.5 RoomModule.....14
 - 4.1.6 BookingModule.....15
 - 4.1.7 ReportModule.....16
 - 4.1.8 ReviewModule.....17
 - 4.1.9 BillModule.....18
 - 4.1.10 PaymentModule.....19
 - 4.1.11 LocationModule.....20
 - 4.1.12 ImageModule.....21
 - 4.1.13 ServiceModule.....22
 - 4.1.14 HelperModule.....23
 - 4.2 Front-end.....23
 - 4.2.1 Package: Pages.....24
 - 4.2.1.1 Component: Home.....24
 - 4.2.1.2 Component: AfterSearch.....25
 - 4.2.1.3 Component: Hotel Details.....25
 - 4.2.1.4 Component: Reserve.....25
 - 4.2.1.5 Component: About us.....26
 - 4.2.1.6 Component: Account.....26
 - 4.2.1.7 Component: Login.....26
 - 4.2.1.8 Component: Register.....26
 - 4.2.1.9 Component: Saved.....27
 - 4.2.1.10 Component: History.....27
 - 4.2.1.11 Component: Feedback.....27
 - 4.2.2 Package: Components.....27
 - 4.2.2.1 Component: SearchBar.....27
 - 4.2.2.2 Component: HotelCard.....28

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

- 4.2.2.3 Component: Filter.....28
 - 4.2.2.4 Component: Stepper.....29
 - 4.2.3 Package: Services.....29
 - 4.2.3.1 Component: ApiService.....29
 - 4.2.4 Package: Redux.....29
 - 4.2.5 Package: Utils.....29
 - 4.2.6 Package: Hooks.....30
 - 4.2.7 Package: Routes.....30
- 5. Deployment.....30**
- 6. Implementation View.....32**
 - 6.1 Back-end.....32
 - 6.2 Front-end.....33

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

Software Architecture Document

1.Introduction

1.1 Purpose

- Tài liệu SAD này giúp người đọc có cái nhìn tổng quan về kiến trúc của toàn bộ hệ thống. Nó cung cấp các mô hình về kiến trúc hệ thống dưới nhiều góc độ khác nhau.
- Tài liệu này được dùng để bao quát và truyền tải các quyết định quan trọng khi chọn kiến trúc phần mềm cho hệ thống.

1.2 Scope

- Tài liệu SAD này được sử dụng trong quá trình phát triển Ứng dụng đặt phòng khách sạn BookaStay trên nền tảng web của dự án nhóm.

1.3 Overview

- Tài liệu SAD này bao gồm một số phần quan trọng như:
 - + Architectural Goals and Constraint: Các mục tiêu và ràng buộc của kiến trúc phần mềm.
 - + Use-Case Model: Chứa mô hình Use-Case.
 - + Logical View: Cung cấp danh sách các components trong hệ thống và mối quan hệ giữa chúng bao gồm các package diagrams và class diagrams.
 - + Deployment: Phương pháp triển khai các thành phần của ứng dụng.
 - + Implementation View: Cấu trúc các thư mục chứa code và cách cài đặt các components.

2.Architectural Goals and Constraints

- Programming Language: NodeJS (NestJs Framework) and Reactjs
- Architectural: Client - Server
 - + Frontend: MVC
 - + Backend: 3-tier (Controller - Service - Repository)
- System Requirements: Web Browser
- Usability Requirements:

Khả năng sử dụng giao diện người dùng phải thân thiện, người dùng dễ thao tác, sử dụng để đặt phòng mà không cần đọc hướng dẫn, đảm bảo vấn đề đa ngôn ngữ.
- Performance Requirements:

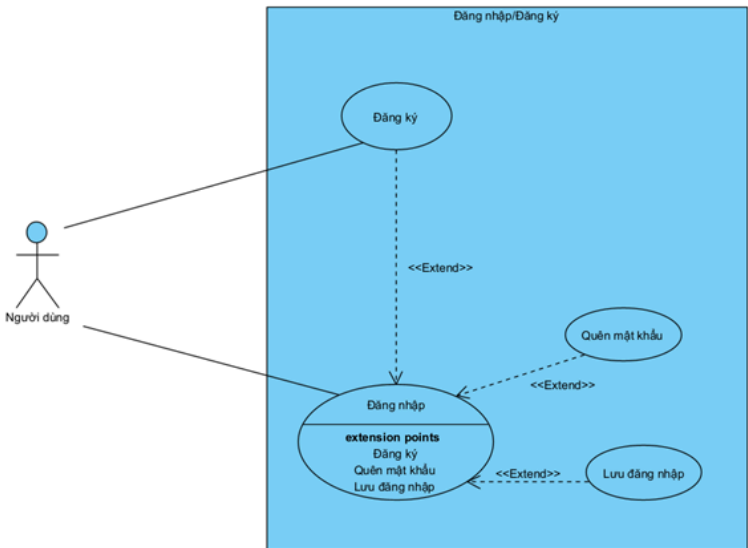
Đòi hỏi về hiệu suất của trang web phải đáp ứng các tiêu chí về thời gian phản hồi nhanh, thời gian phản

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

hồi không vượt quá ngưỡng 3 giây trên mỗi thao tác bất kỳ. Hệ thống đòi hỏi phải đáp ứng được đa người dùng cùng truy cập cùng một lúc, có biện pháp khắc phục trước tình trạng quá tải. Hệ thống cần hoạt động liên tục 24/7 với thời gian chết tối thiểu.

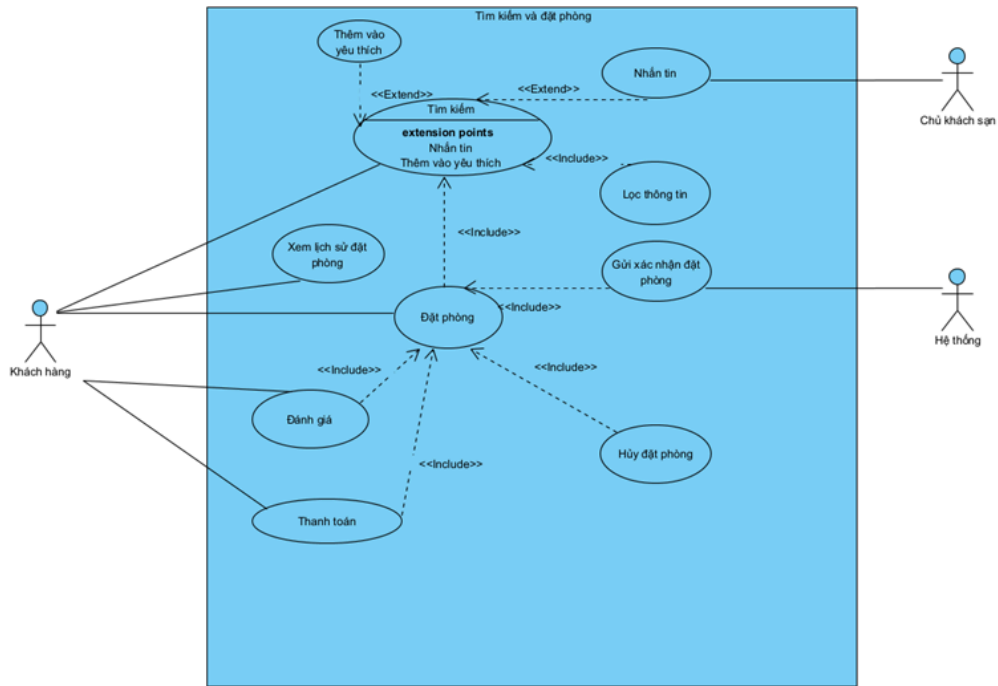
- Development:
Cần có hệ thống sao lưu dữ liệu định kỳ để khôi phục nhanh chóng nếu có sự cố. Cơ sở dữ liệu và các dịch vụ có khả năng mở rộng dễ dàng khi cần.
- Security Requirements:
Dữ liệu khách hàng, đặc biệt là thông tin thanh toán và thông tin cá nhân, phải được mã hóa và bảo vệ. Có biện pháp chống lại các cuộc tấn công phổ biến từ bên ngoài.
- Hỗ trợ tích hợp với các API và dịch vụ bên thứ ba, chẳng hạn như cổng thanh toán và hệ thống quản lý khách sạn (PMS)
- Ràng buộc về hệ thống một số yêu cầu như đáp ứng được trong ngân sách triển khai dự án; về thời gian triển khai quy hoạch dự án, tính cam kết; các công nghệ hỗ trợ tiên tiến, phù hợp khả năng; tuân thủ các quy định pháp lý, điều khoản hợp đồng; ngoài ra, đi kèm theo những hạn chế về cơ sở hạ tầng trong thời gian triển khai, hoặc đáp ứng các kế hoạch phát triển nâng cấp trong tương lai.

3.Use-Case Model

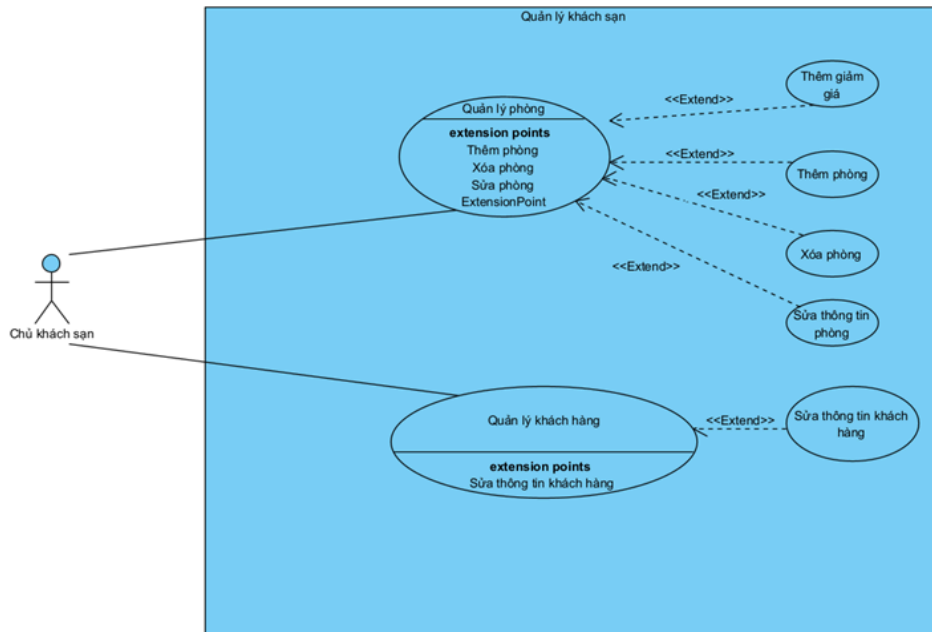


Sơ đồ Use-Case: Hệ thống đăng nhập/dăng ký

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

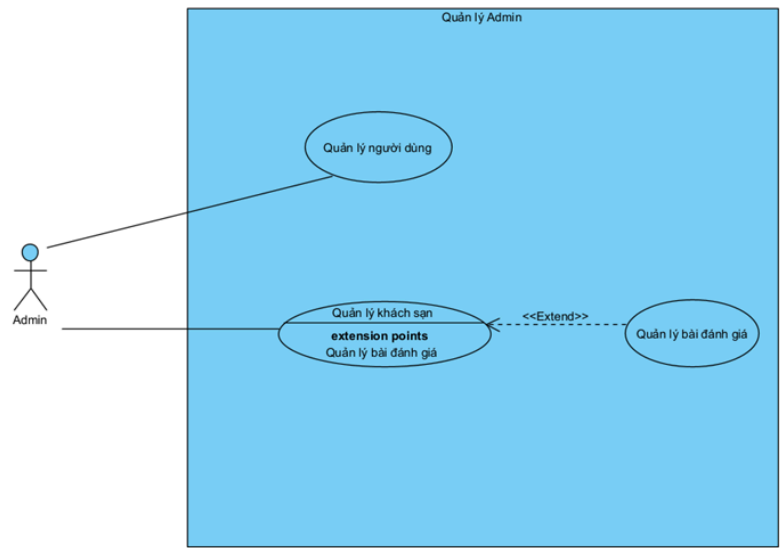


Sơ đồ Use-Case: Hệ thống tìm kiếm và đặt phòng



Sơ đồ Use-Case: Hệ thống quản lý khách sạn

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

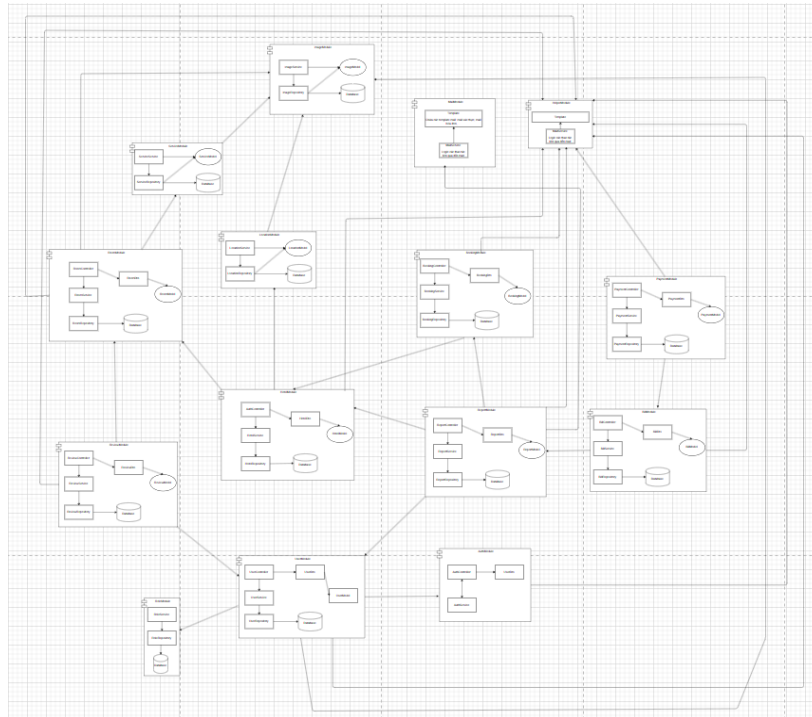
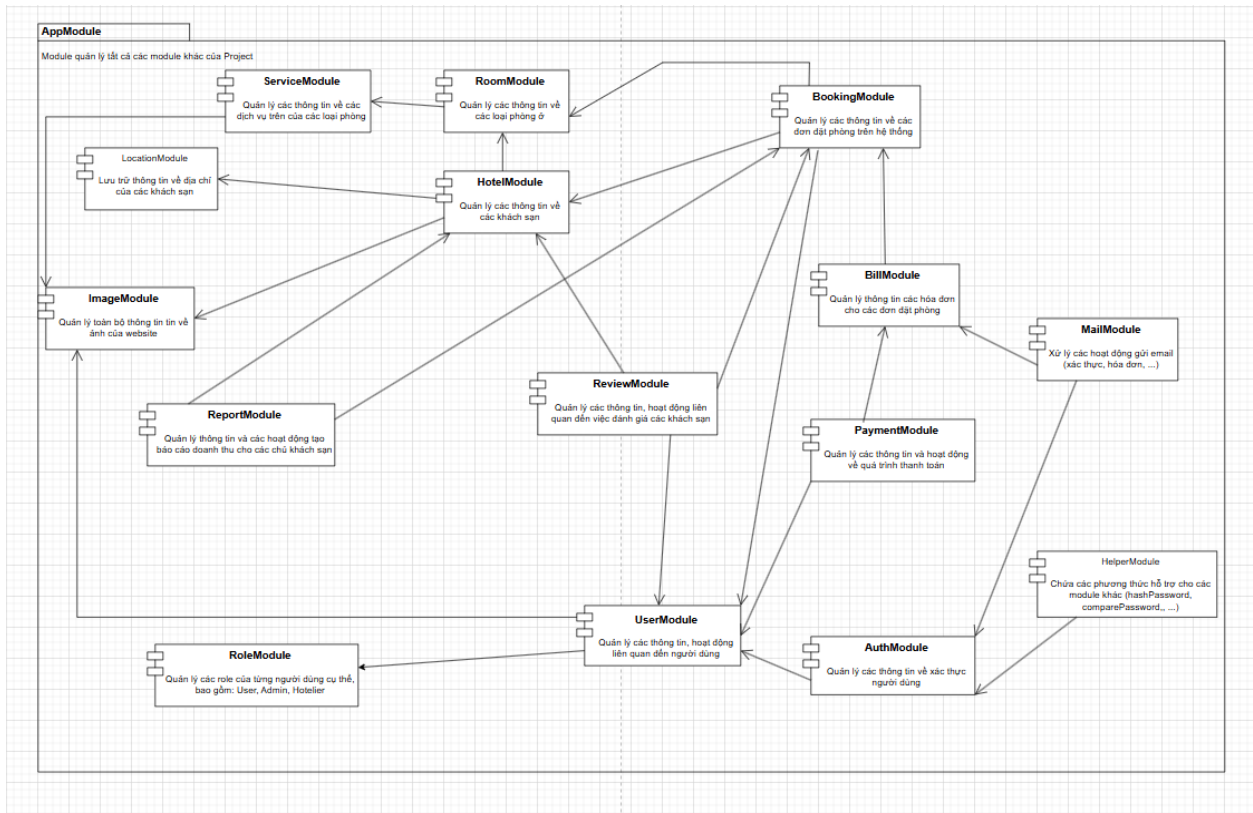


Sơ đồ Use-Case: Hệ thống quản lý của admin

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

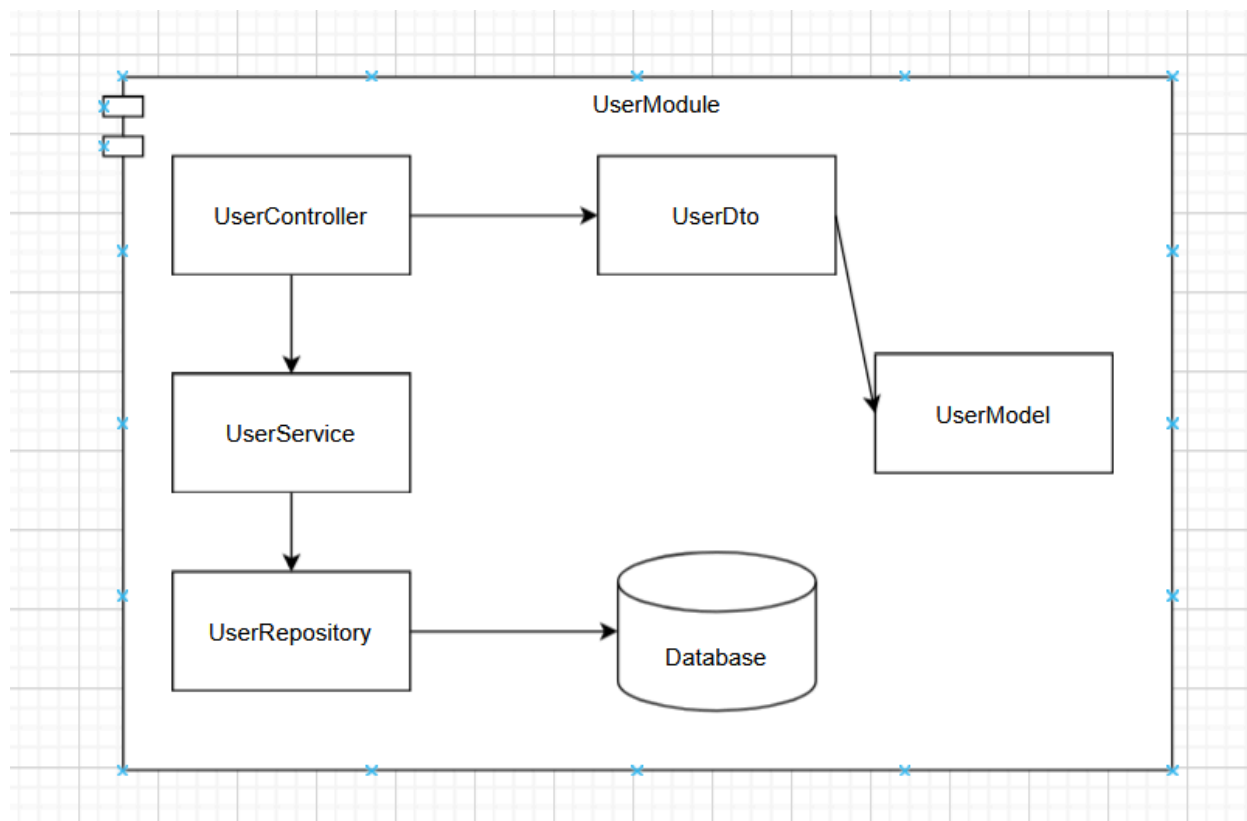
4.Logical View

4.1 Back-end



BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.1 UserModule



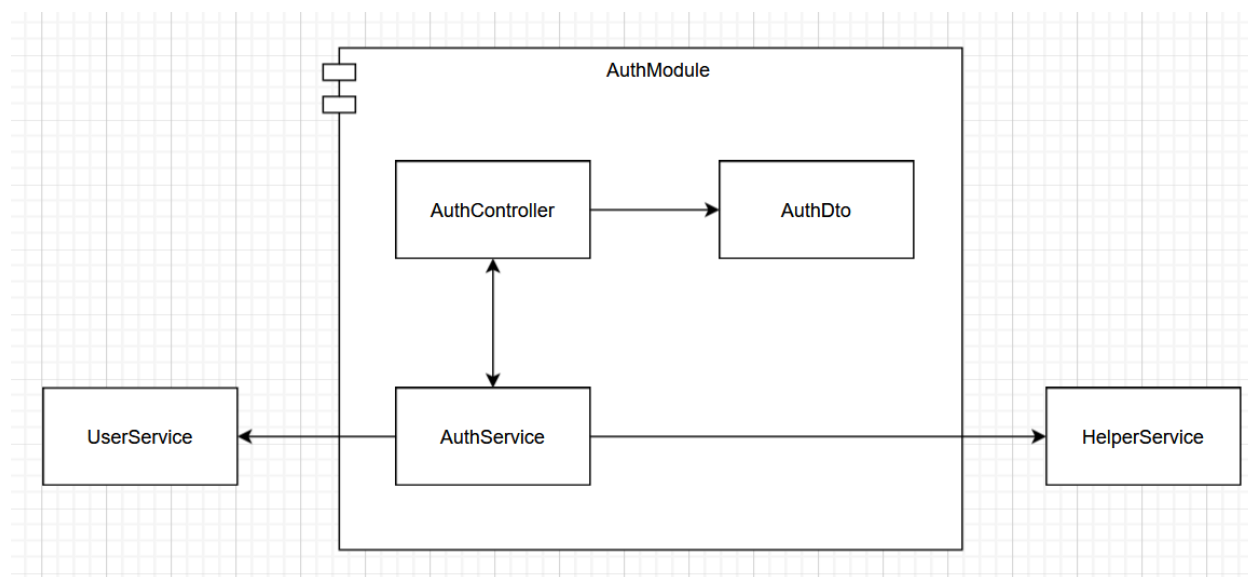
⇒ UserModule thực hiện các thao tác liên quan đến User như đăng nhập, đăng ký, quản lý tài khoản, favourites, ...

Gồm các component con bao gồm: UserController, UserService, UserRepository, UserModel, UserDto

- UserModel: đối tượng ánh xạ table User trong database, phục vụ các thao tác liên quan đến database. Chỉ được sử dụng trong UserRepository
- UserDto (Data transfer object): tạo ra dựa trên UserModel, dùng để truyền tải dữ liệu giữa các component và gửi dữ liệu về client
- UserController: nơi định nghĩa ra các api và nhận dữ liệu request chuyển sang dạng các Dto để truyền cho UserService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- UserService: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các Dto từ Controller, kết hợp với các Service khác như AuthService, HelperService, ... để xử lý dữ liệu. Nếu cần giao tiếp với database thì service sẽ chuyển các Dto về lại UserModel và gọi UserRepository để ghi/nhận dữ liệu với database
- UserRepository: chịu trách nhiệm giao tiếp với database. Tất cả các thao tác dữ liệu liên quan đến database đều được thực hiện qua component này. UserRepository sẽ dùng UserModel để tương tác với database.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.2 AuthModule



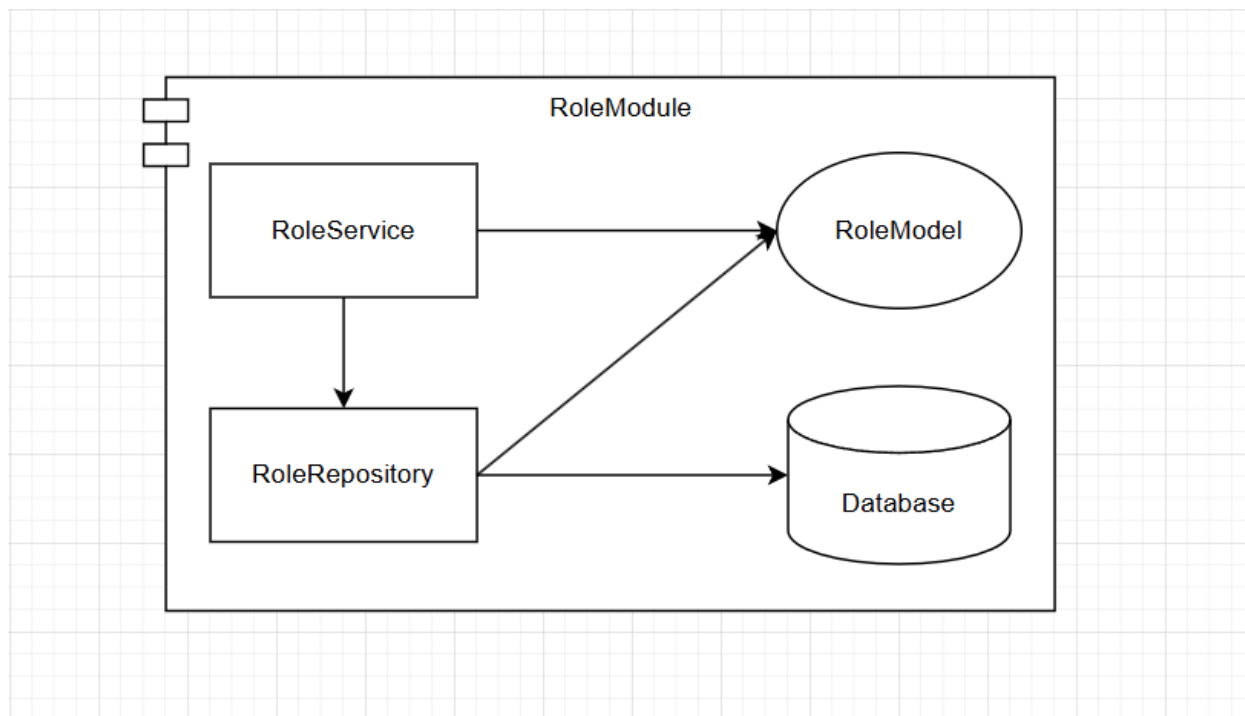
⇒ Module này có vai trò thực hiện xác thực cho ứng dụng. Tất cả các hoạt động liên quan đến xác thực người dùng sẽ được thực hiện ở Module này

Gồm các component con bao gồm: AuthController, AuthService, AuthRepository, AuthDto

- AuthDto (Data transfer object): tạo ra dựa trên UserModel, object dùng để truyền dữ liệu xác thực user giữa các component và với các Module khác
- AuthController: nơi định nghĩa ra các api liên quan đến xác thực người dùng gồm: đăng nhập, đăng ký, quên mật khẩu, ... và nhận dữ liệu request chuyển sang dạng các Dto để truyền cho AuthService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- AuthService: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các Dto từ Controller, kết hợp với các Service khác như UserService, HelperService, ... để xử lý dữ liệu

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.3 RoleModule



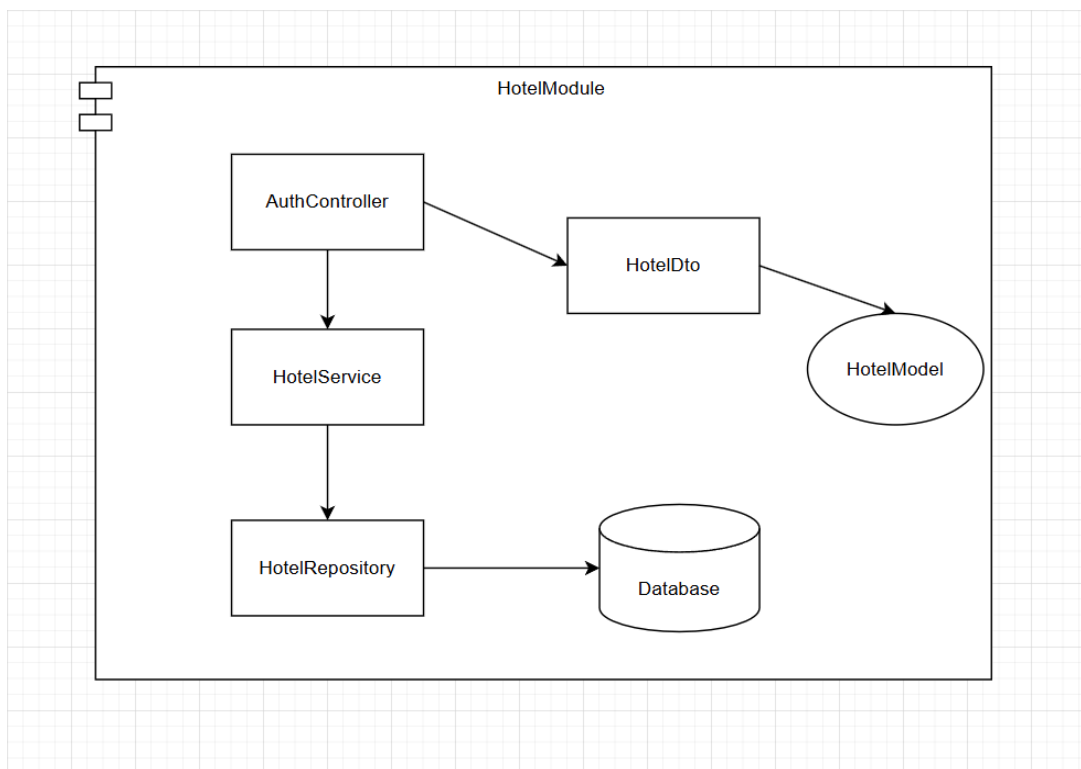
⇒ Module này xử lý việc phân quyền người dùng. Các hoạt động kiểm tra quyền, cấp quyền, quản lý các thao tác nào được thực hiện bởi quyền nào sẽ được thực hiện ở đây

Gồm các component con bao gồm: RoleController, RoleService, RoleRepository, RoleModel, RoleDto

- RoleModel: đối tượng ánh xạ table Role trong database, phục vụ các thao tác liên quan đến database. Chỉ được sử dụng trong RoleRepository
- RoleDto (Data transfer object): tạo ra dựa trên RoleModel, dùng để truyền tải dữ liệu giữa các component và gửi dữ liệu về client
- RoleController: nơi định nghĩa ra các api và nhận dữ liệu request chuyển sang dạng các Dto để truyền cho RoleService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- RoleService: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các Dto từ Controller, kết hợp với các Service khác như AuthService, HelperService, ... để xử lý dữ liệu. Nếu cần giao tiếp với database thì service sẽ chuyển các Dto về lại RoleModel và gọi RoleRepository để ghi/nhận dữ liệu với database
- RoleRepository: chịu trách nhiệm giao tiếp với database. Tất cả các thao tác dữ liệu liên quan đến database đều được thực hiện qua component này. RoleRepository sẽ dùng RoleModel để tương tác với database

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.4 HotelModule



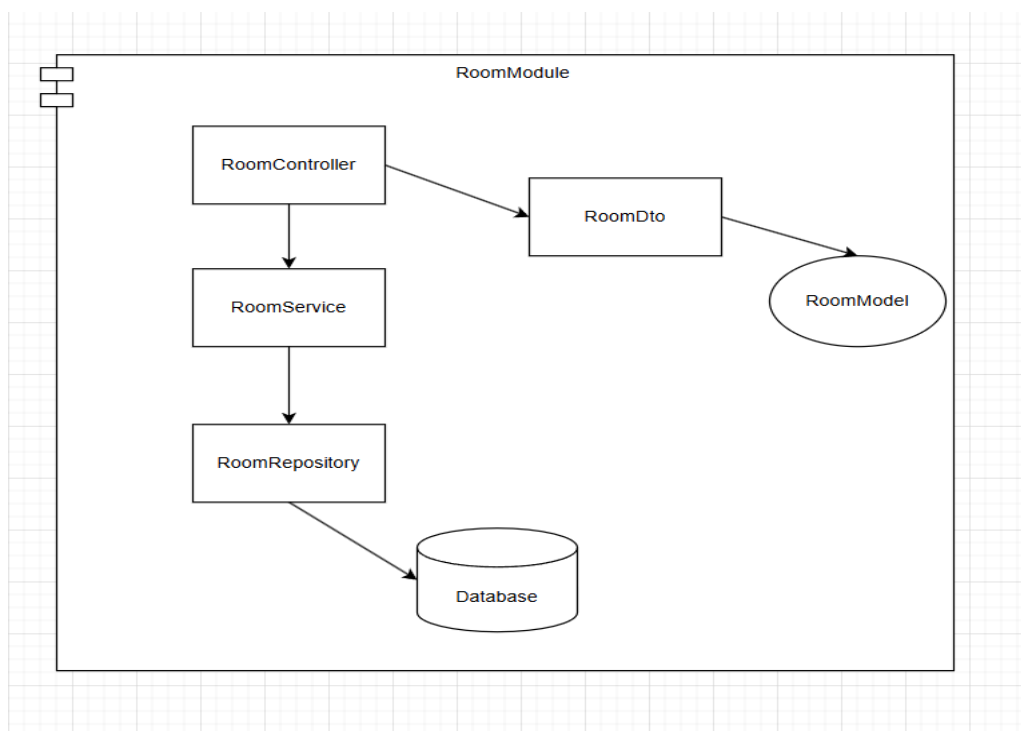
⇒ Module này thực hiện các hoạt động liên quan đến các Hotel của ứng dụng. Tất cả các hoạt động thêm/sửa/xóa hotel đều được thực hiện trong module này

Gồm các component con bao gồm: HotelController, HotelService, HotelRepository, HotelModel, HotelDto

- HotelModel: đối tượng ánh xạ table Hotel trong database, phục vụ các thao tác liên quan đến database. Chỉ được sử dụng trong HotelRepository
- HotelDto (Data transfer object): tạo ra dựa trên HotelModel, dùng để truyền tải dữ liệu giữa các component và gửi dữ liệu về client
- HotelController: nơi định nghĩa ra các api và nhận dữ liệu request chuyển sang dạng các Dto để truyền cho HotelService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- HotelService: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các Dto từ Controller, kết hợp với các Service khác như HelperService, UserService... để xử lý dữ liệu. Nếu cần giao tiếp với database thì service sẽ chuyển các Dto về lại HotelModel và gọi HotelRepository để ghi/nhận dữ liệu với database
- HotelRepository: chịu trách nhiệm giao tiếp với database. Tất cả các thao tác dữ liệu liên quan đến database đều được thực hiện qua component này. HotelRepository sẽ dùng HotelModel để tương tác với database

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.5 RoomModule



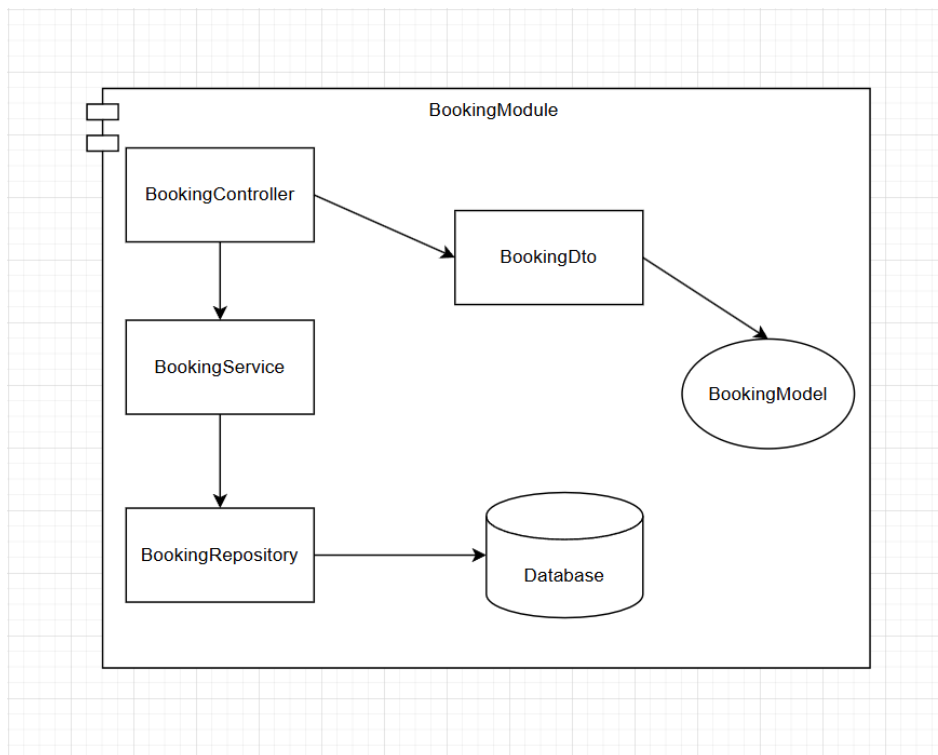
⇒ Module này xử lý các hoạt động về phòng ở của ứng dụng. Liên kết với các Module khác bao gồm Service, Image, Hotel, Booking, ... để xử lý các thao tác, thông tin về các phòng ở

Gồm các component con bao gồm: RoomController, RoomService, RoomRepository, RoomModel, RoomDto

- RoomModel: đối tượng ánh xạ table Room trong database, phục vụ các thao tác liên quan đến database. Chỉ được sử dụng trong RoomRepository
- RoomDto (Data transfer object): tạo ra dựa trên RoomModel, dùng để truyền tải dữ liệu giữa các component và gửi dữ liệu về client
- RoomController: nơi định nghĩa ra các api và nhận dữ liệu request chuyển sang dạng các Dto để truyền cho RoomService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- RoomService: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các Dto từ Controller, kết hợp với các Service khác như HotelService, HelperService, ... để xử lý dữ liệu. Nếu cần giao tiếp với database thì service sẽ chuyển các Dto về lại RoomModel và gọi RoomRepository để ghi/nhận dữ liệu với database
- RoomRepository: chịu trách nhiệm giao tiếp với database. Tất cả các thao tác dữ liệu liên quan đến database đều được thực hiện qua component này. RoomRepository sẽ dùng RoomModel để tương tác với database.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.6 BookingModule



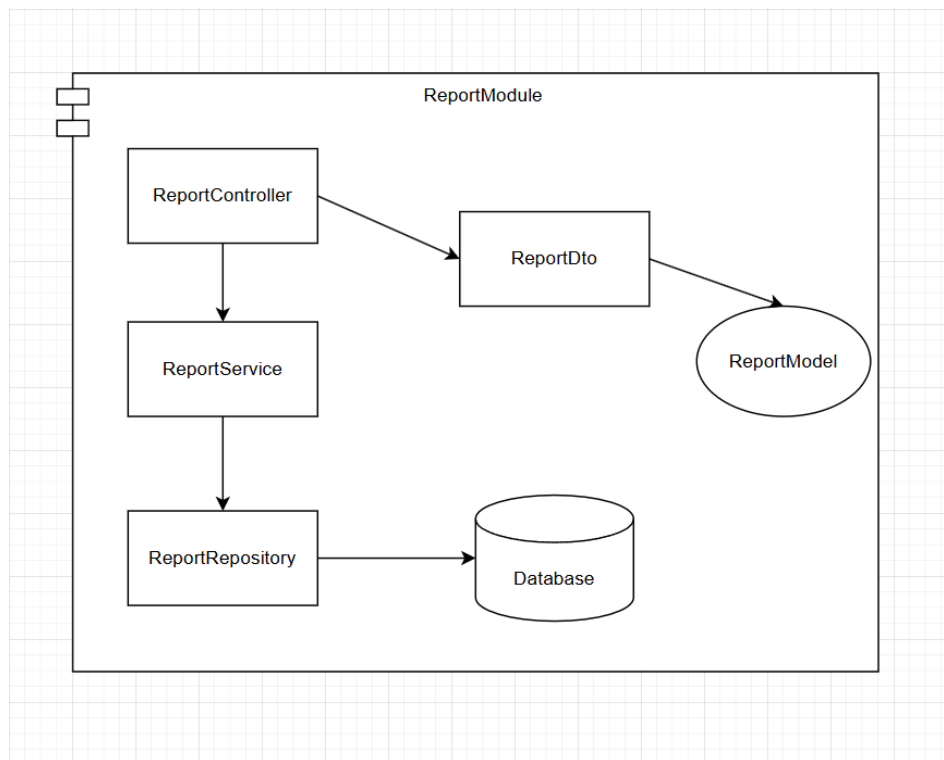
⇒ Module này xử lý các hoạt động về việc đặt phòng của ứng dụng. Liên kết với các Module khác bao gồm Hotel, Bill, Payment... để xử lý các thao tác, thông tin về các đơn đặt phòng của khách hàng

Gồm các component con bao gồm: BookingController, BookingService, BookingRepository, BookingModel, BookingDto

- BookingModel: thực thể ánh xạ table Booking trong database, phục vụ các thao tác liên quan đến database. Chỉ được sử dụng trong BookingRepository
- BookingDto (Data transfer object): tạo ra dựa trên BookingModel, dùng để truyền tải dữ liệu giữa các component và gửi dữ liệu về client
- BookingController: nơi định nghĩa ra các api và nhận dữ liệu request chuyển sang dạng các Dto để truyền cho BookingService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- BookingService: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các Dto từ Controller, kết hợp với các Service khác như BillService, PaymentService, HelperService, ... để xử lý dữ liệu. Nếu cần giao tiếp với database thì service sẽ chuyển các Dto về lại BookingModel và gọi BookingRepository để ghi/nhận dữ liệu với database
- BookingRepository: chịu trách nhiệm giao tiếp với database. Tất cả các thao tác dữ liệu liên quan đến database đều được thực hiện qua component này. BookingRepository sẽ dùng BookingModel để tương tác với database

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.7 ReportModule



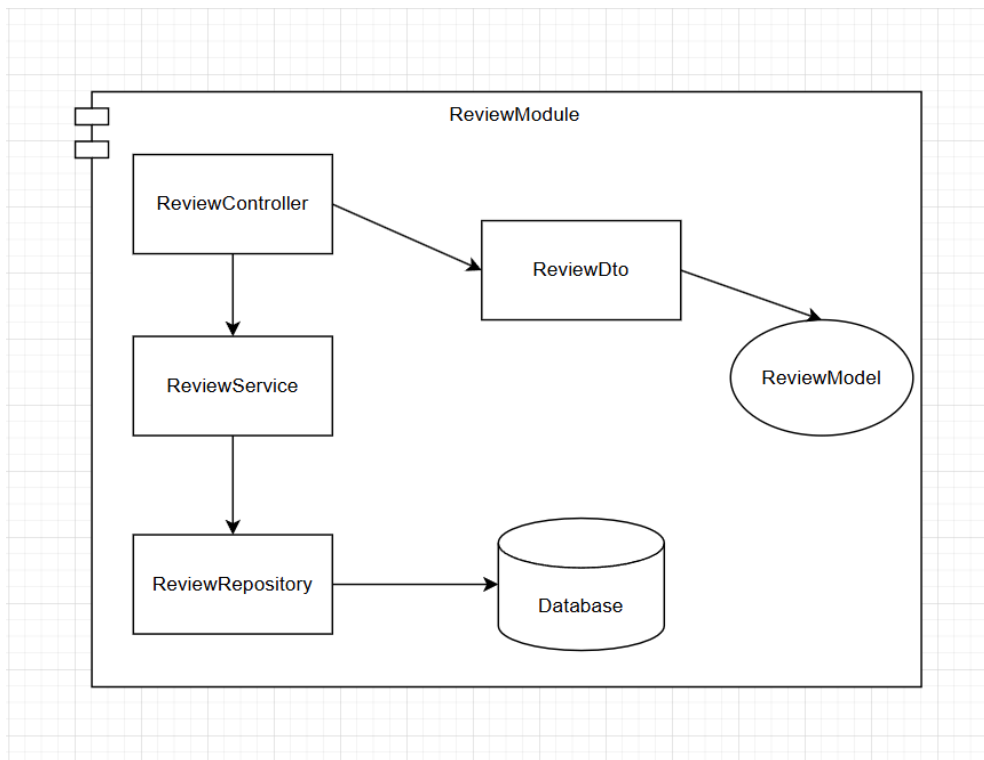
⇒ Module này xử lý các hoạt động về các báo cáo tài chính cho các khách sạn của ứng dụng. Liên kết với các Module khác bao gồm Hotel, Payment... để xử lý các thao tác, thông tin về các báo cáo cho khách hàng

Gồm các component con bao gồm: ReportController, ReportService, ReportRepository, ReportModel, ReportDto

- **ReportModel**: thực thể ánh xạ table Report trong database, phục vụ các thao tác liên quan đến database. Chỉ được sử dụng trong ReportRepository
- **ReportDto** (Data transfer object): tạo ra dựa trên ReportModel, dùng để truyền tải dữ liệu giữa các component và gửi dữ liệu về client
- **ReportController**: nơi định nghĩa ra các api và nhận dữ liệu request chuyển sang dạng các Dto để truyền cho ReportService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- **ReportService**: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các Dto từ Controller, kết hợp với các Service khác như HotelService, RoomService, HelperService, ... để xử lý dữ liệu. Nếu cần giao tiếp với database thì service sẽ chuyển các Dto về lại ReportModel và gọi ReportRepository để ghi/nhận dữ liệu với database
- **ReportRepository**: chịu trách nhiệm giao tiếp với database. Tất cả các thao tác dữ liệu liên quan đến database đều được thực hiện qua component này. ReportRepository sẽ dùng ReportModel để tương tác với database.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.8 ReviewModule



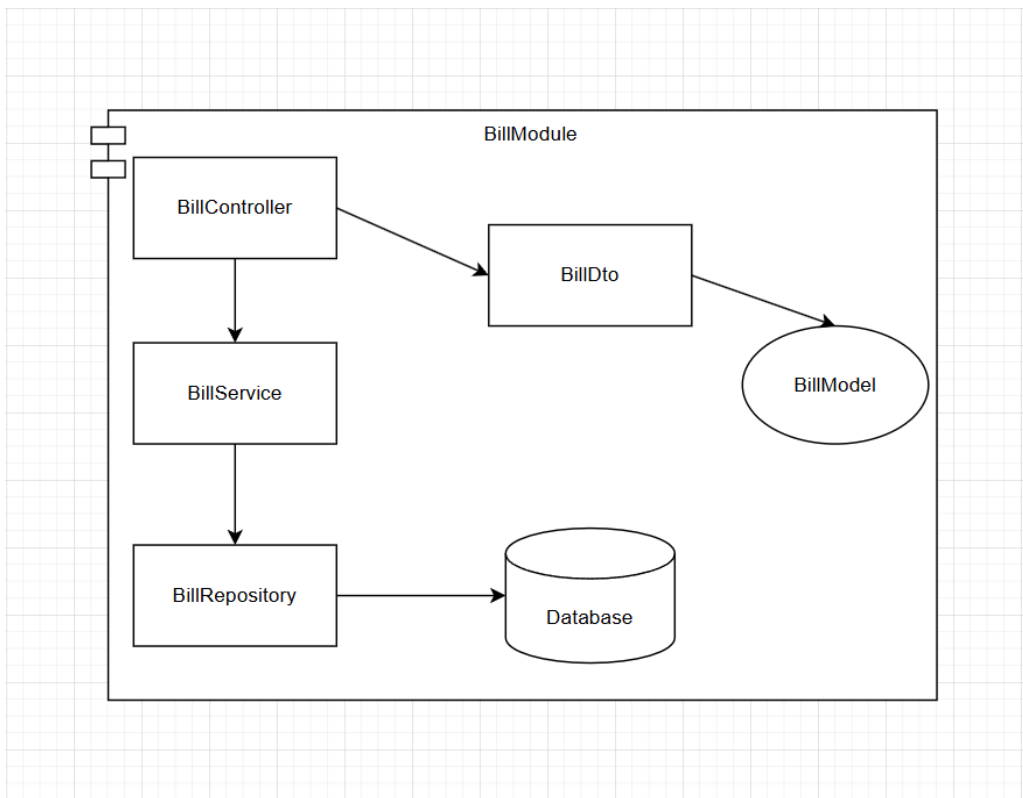
⇒ Module này xử lý các hoạt động về các đánh giá khách sạn của ứng dụng. Liên kết với các Module khác bao gồm Hotel, Room, ... để xử lý các thao tác, thông tin về các đánh giá của khách hàng

Gồm các component con bao gồm: ReviewController, ReviewService, ReviewRepository, ReviewModel, ReviewDto

- ReviewModel: thực thể ánh xạ table Review trong database, phục vụ các thao tác liên quan đến database. Chỉ được sử dụng trong ReviewRepository
- ReviewDto (Data transfer object): tạo ra dựa trên ReviewModel, dùng để truyền tải dữ liệu giữa các component và gửi dữ liệu về client
- ReviewController: nơi định nghĩa ra các api và nhận dữ liệu request chuyển sang dạng các Dto để truyền cho ReviewService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- ReviewService: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các Dto từ Controller, kết hợp với các Service khác như HotelService, RoomService, HelperService, ... để xử lý dữ liệu. Nếu cần giao tiếp với database thì service sẽ chuyển các Dto về lại ReviewModel và gọi ReviewRepository để ghi/nhận dữ liệu với database
- ReviewRepository: chịu trách nhiệm giao tiếp với database. Tất cả các thao tác dữ liệu liên quan đến database đều được thực hiện qua component này. ReviewRepository sẽ dùng ReviewModel để tương tác với database.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.9 BillModule



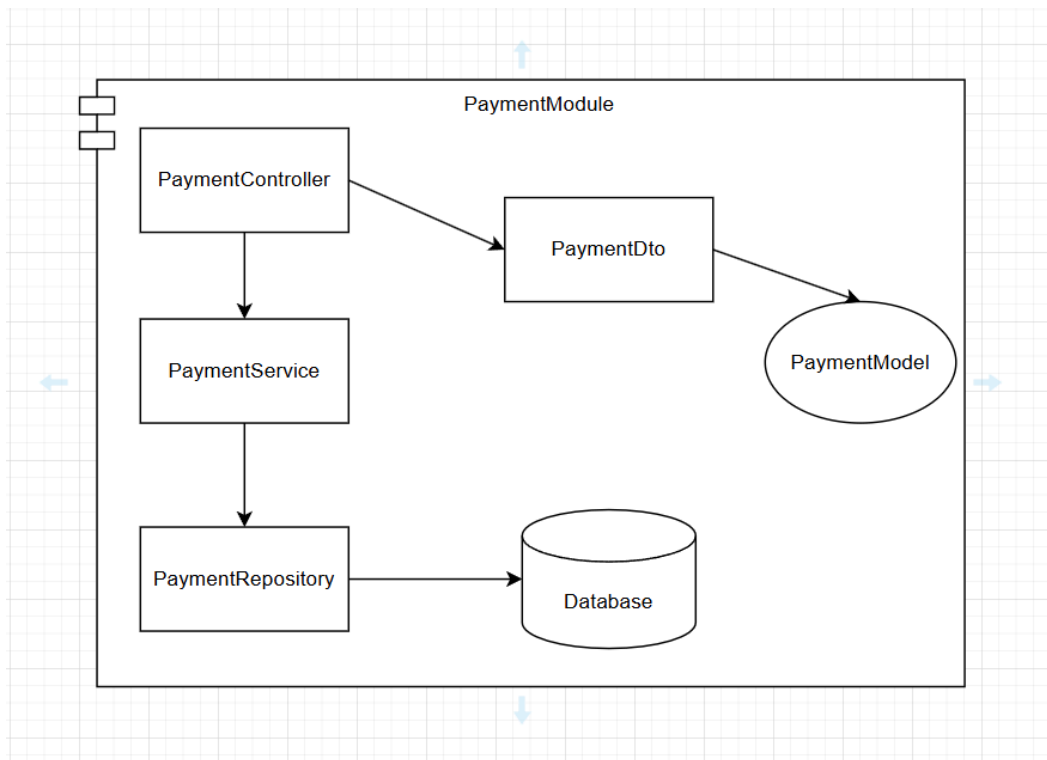
⇒ Module này xử lý các hoạt động về các hóa đơn cho các đơn đặt phòng của ứng dụng. Liên kết với các Module khác bao gồm Hotel, Room, ... để xử lý các thao tác, thông tin về các hóa đơn cho các đơn đặt phòng

Gồm các component con bao gồm: BillController, BillService, BillRepository, BillModel, BillDto

- BillModel: thực thể ánh xạ table Bill trong database, phục vụ các thao tác liên quan đến database. Chỉ được sử dụng trong BillRepository
- BillDto (Data transfer object): tạo ra dựa trên BillModel, dùng để truyền tải dữ liệu giữa các component và gửi dữ liệu về client
- BillController: nơi định nghĩa ra các api và nhận dữ liệu request chuyển sang dạng các Dto để truyền cho BillService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- BillService: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các Dto từ Controller, kết hợp với các Service khác như HotelService, RoomService, HelperService, ... để xử lý dữ liệu. Nếu cần giao tiếp với database thì service sẽ chuyển các Dto về lại BillModel và gọi BillRepository để ghi/nhận dữ liệu với database
- BillRepository: chịu trách nhiệm giao tiếp với database. Tất cả các thao tác dữ liệu liên quan đến database đều được thực hiện qua component này. BillRepository sẽ dùng BillModel để tương tác với database.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.10 PaymentModule



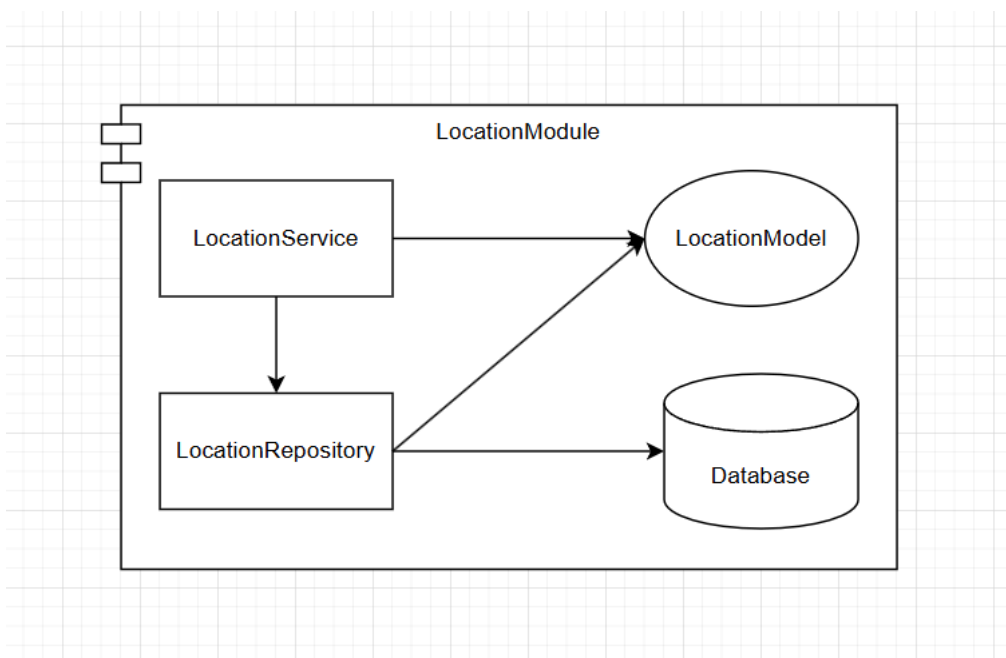
⇒ Module này xử lý các hoạt động về thanh toán cho các đơn đặt phòng của ứng dụng. Liên kết với các Module khác bao gồm Bill, Hotel, Room, ... để xử lý các thao tác, thông tin về thanh toán cho các đơn đặt phòng

Gồm các component con bao gồm: PaymentController, PaymentService, PaymentRepository, PaymentModel, PaymentDto

- PaymentModel: thực thể ánh xạ table Payment trong database, phục vụ các thao tác liên quan đến database. Chỉ được sử dụng trong PaymentRepository
- PaymentDto (Data transfer object): tạo ra dựa trên PaymentModel, dùng để truyền tải dữ liệu giữa các component và gửi dữ liệu về client
- PaymentController: nơi định nghĩa ra các api và nhận dữ liệu request chuyển sang dạng các Dto để truyền cho PaymentService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- PaymentService: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các Dto từ Controller, kết hợp với các Service khác như BillService, HotelService, RoomService, HelperService, ... để xử lý dữ liệu. Nếu cần giao tiếp với database thì service sẽ chuyển các Dto về lại PaymentModel và gọi PaymentRepository để ghi/nhận dữ liệu với database
- PaymentRepository: chịu trách nhiệm giao tiếp với database. Tất cả các thao tác dữ liệu liên quan đến database đều được thực hiện qua component này. PaymentRepository sẽ dùng PaymentModel để tương tác với database.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.11 LocationModule



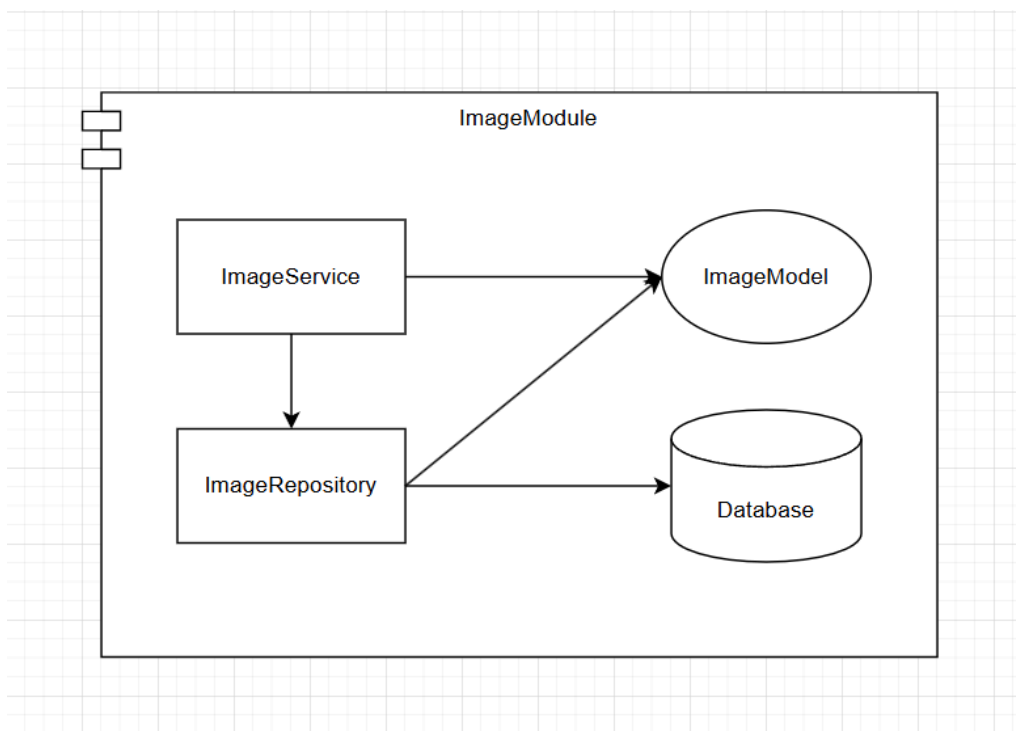
⇒ Module này quản lý thông tin về các địa chỉ của khách sạn. Mọi thao tác liên quan đến địa chỉ như tìm kiếm, thêm/sửa/xóa địa chỉ đều được thực hiện trong Module này.

Gồm các component con bao gồm: LocationController, LocationService, LocationRepository, LocationModel

- LocationModel: thực thể ánh xạ table Location trong database, phục vụ các thao tác liên quan đến database. Chỉ được sử dụng trong LocationRepository
- LocationDto (Data transfer object): vì LocationModel chỉ gồm các trường dữ liệu đơn giản và không cần luân chuyển nhiều nên không cần có thêm Dto cho Module này, tối ưu hóa được cấu trúc project
- LocationController: nơi định nghĩa ra các api và nhận dữ liệu request để truyền cho LocationService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- LocationService: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các thông tin từ Controller, kết hợp với các Service khác như HotelService, HelperService, ... để xử lý dữ liệu. Nếu cần giao tiếp với database thì service sẽ gọi LocationRepository để ghi/nhận dữ liệu với database
- LocationRepository: chịu trách nhiệm giao tiếp với database. Tất cả các thao tác dữ liệu liên quan đến database đều được thực hiện qua component này. LocationRepository sẽ dùng LocationModel để tương tác với database.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.12 ImageModule



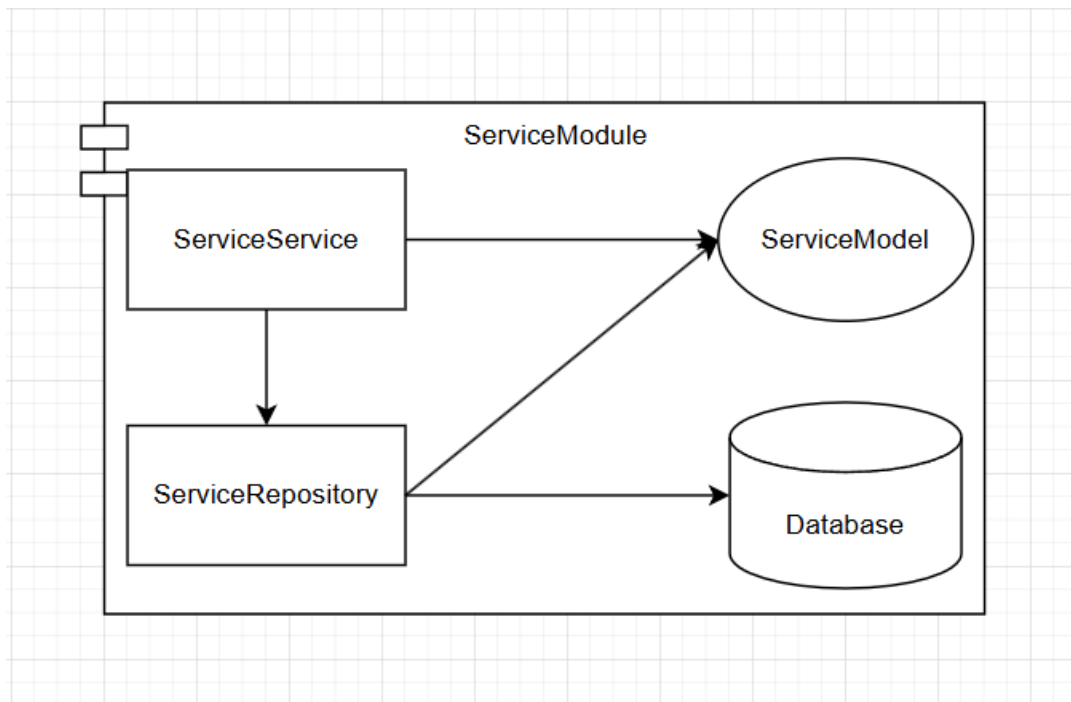
⇒ Module này quản lý thông tin và các thao tác liên quan đến hình ảnh của website. Mọi thao tác liên quan đến ảnh như tìm kiếm, thêm/xóa ảnh đều được thực hiện trong Module này.

Gồm các component con bao gồm: ImageController, ImageService, ImageRepository, ImageModel

- ImageModel: thực thể ánh xạ table Image trong database, phục vụ các thao tác liên quan đến database. Chỉ được sử dụng trong ImageRepository
- ImageDto (Data transfer object): vì ImageModel chỉ gồm các trường dữ liệu đơn giản và không cần luân chuyển nhiều nên không cần có thêm Dto cho Module này, giúp source code ngắn gọn và dễ quản lý hơn
- ImageController: nơi định nghĩa ra các api và nhận dữ liệu request để truyền cho ImageService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- ImageService: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các thông tin từ Controller. Vì module này chịu trách nhiệm xử lý tất cả những gì liên quan đến ảnh nên hầu như sẽ liên kết đến tất cả các Module và Service khác để xử lý dữ liệu. Nếu cần giao tiếp với database thì service sẽ gọi ImageRepository để ghi/nhận dữ liệu với database
- ImageRepository: chịu trách nhiệm giao tiếp với database. Tất cả các thao tác dữ liệu liên quan đến database đều được thực hiện qua component này. ImageRepository sẽ dùng ImageModel để tương tác với database.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.1.13 ServiceModule



⇒ Module này quản lý thông tin và các thao tác liên quan đến dịch vụ của các phòng ở. Mọi thao tác liên quan đến ảnh như tìm kiếm, thêm/sửa/xóa các dịch vụ đều được thực hiện trong Module này. Mặc định Module này sẽ cung cấp sẵn một số dịch vụ cho chủ khách sạn chọn ứng khách sạn của họ. Nếu có dịch vụ khác thì chủ khách sạn sẽ thực hiện thêm thủ công và hệ thống cũng sẽ lưu vào database để phục vụ cho các lần thêm tiếp theo

Gồm các component con bao gồm: ServiceController, ServiceService, ServiceRepository, ServiceModel

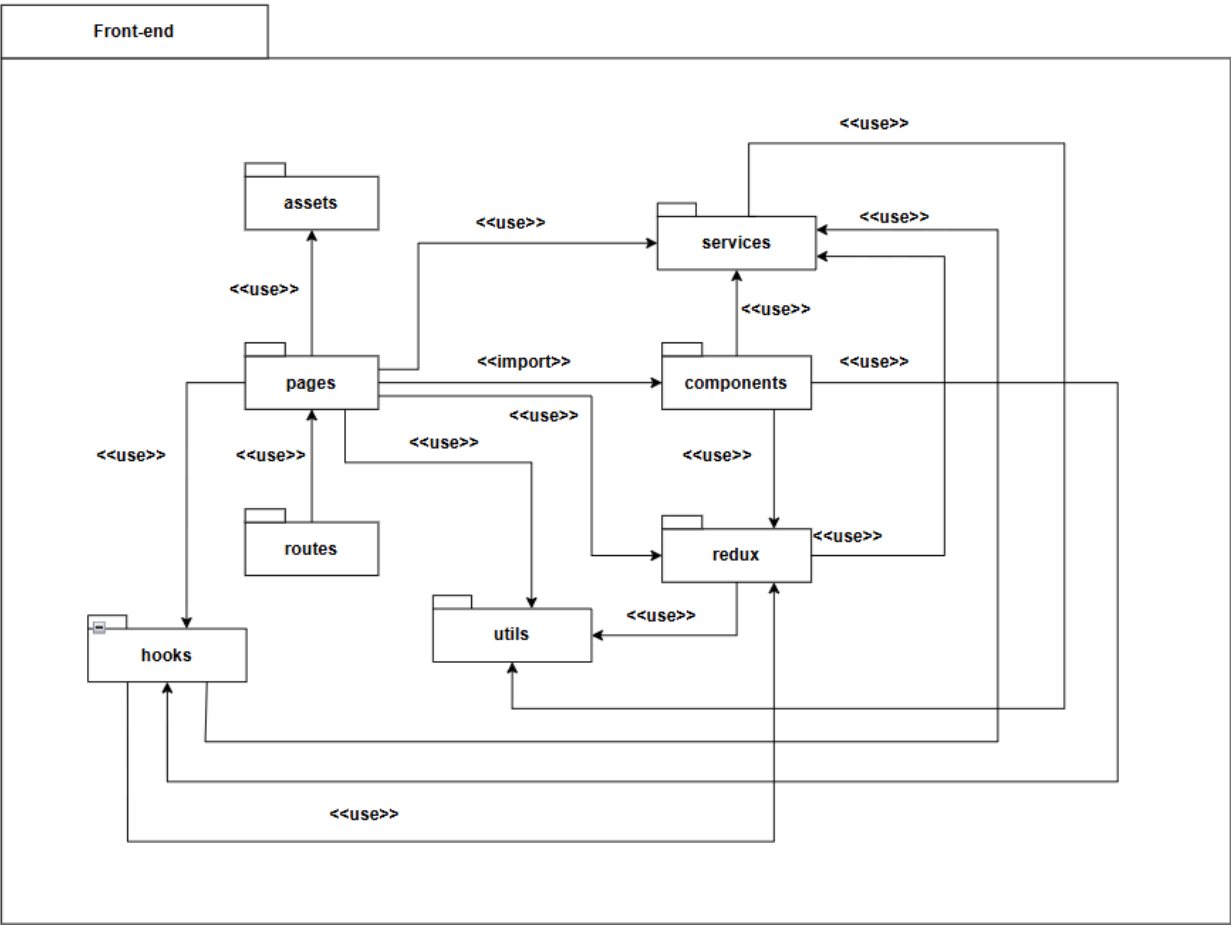
- ServiceModel: thực thể ánh xạ table Service trong database, phục vụ các thao tác liên quan đến database. Chỉ được sử dụng trong ServiceRepository
- ServiceDto (Data transfer object): vì ServiceModel chỉ gồm các trường dữ liệu đơn giản và không cần luân chuyển nhiều nên không cần có thêm Dto cho Module này, giúp source code ngắn gọn và dễ quản lý hơn
- ServiceController: nơi định nghĩa ra các api và nhận dữ liệu request để truyền cho ServiceService xử lý. Sau đó nhận kết quả trả về từ các Service, đưa vào response và gửi phản hồi cho client
- ServiceService: nơi chứa các hàm xử lý business logic cho các api được định nghĩa ở Controller. Component này nhận các thông tin từ Controller, kết hợp với các Service khác như HotelService, HelperService, ... để xử lý dữ liệu. Nếu cần giao tiếp với database thì service sẽ gọi ServiceRepository để ghi/nhận dữ liệu với database
- ServiceRepository: chịu trách nhiệm giao tiếp với database. Tất cả các thao tác dữ liệu liên quan đến database đều được thực hiện qua component này. ServiceRepository sẽ dùng ServiceModel để tương tác với database

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

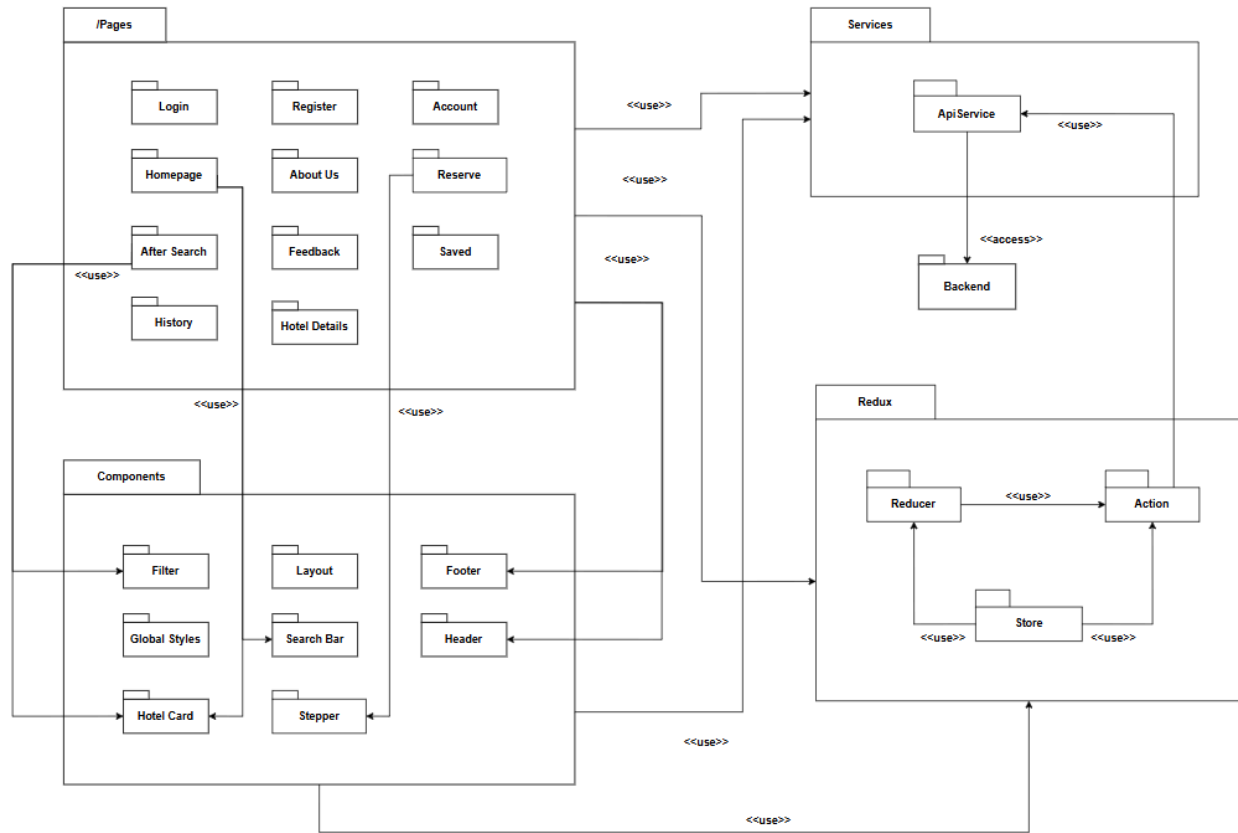
4.1.14 HelperModule

⇒ Module này là Module hỗ trợ cho các Module khác, không giao tiếp với database hay liên quan đến api nên không có các object model (dto), controller và cả repository. Module này chỉ gồm một component duy nhất là HelperService, chứa các function hỗ trợ thực thi logic cho các Module khác như các hàm hashPassword, comparePassword, getProfit (tính lợi nhuận cho các báo cáo), exportFile (convert báo cáo tài chính sang các định dạng khác nhau như excel, pdf, ...)

4.2 Front-end



BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	



4.2.1 Package: Pages

Package "Pages" chịu trách nhiệm quản lý các trang chính của ứng dụng, nơi hiển thị giao diện người dùng. Mỗi trang trong package này thường là một phần lớn của ứng dụng, có thể chứa nhiều components con và thực hiện các hành động, như hiển thị thông tin, xử lý sự kiện, và tương tác với API.

4.2.1.1 Component: Home

- Trách Nhiệm: Hiển thị trang chủ của ứng dụng, bao gồm các phần tử như slider ảnh, thanh tìm kiếm, các điểm đến nổi bật, danh sách khách sạn đề xuất. Quản lý trạng thái và logic điều hướng các phần tử trong trang chủ.
- Kết nối với các component khác:
 - + **Home** sử dụng **SearchBar** và **HotelCard** để hiển thị UI.
 - + **Home** sử dụng **ApiServices** để lấy dữ liệu từ các API cần thiết cho các phần tử trong trang. (Danh sách khách sạn recommend)

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.2.1.2 Component: AfterSearch

- Trách nhiệm: Thực hiện việc tìm kiếm / lọc và hiển thị kết quả tìm kiếm /lọc ra một danh sách thông qua dữ liệu được gửi từ SearchBar / Filter.
- Kết nối với các component khác:
 - + **AfterSearch** sử dụng component **SearchBar** để hiển thị thanh tìm kiếm và thực hiện tìm kiếm lại khách sạn theo thông tin người dùng nhập lại.
 - + Sử dụng Component **Filter** để hiển thị ra bộ lọc và nhận các tiêu chí lọc từ **Filter** để thực hiện lọc và hiển thị ra kết quả đúng với tiêu chí lọc.
 - + Sử dụng **ApiServices** để gửi về cho Backend dữ liệu tìm kiếm và các tiêu chí lọc. Đồng thời nhận lại dữ liệu sau khi tìm kiếm / lọc để hiển thị danh sách kết quả ra màn hình.

4.2.1.3 Component: Hotel Details

- Trách nhiệm: Hiển thị thông tin chi tiết của một khách sạn, bao gồm các thông tin như tên khách sạn, hình ảnh, mô tả, tiện nghi, đánh giá, giá cả, và nút đặt phòng. Cho phép người dùng xem các đánh giá và tương tác với khách sạn (lưu, đặt phòng).
- Kết nối với các component khác:
 - + Sử dụng **ApiServices** để lấy dữ liệu chi tiết của khách sạn từ backend dựa trên ID khách sạn.
 - + Kết nối với **Saved** để lưu khách sạn vào danh sách yêu thích.
 - + Điều hướng đến trang **Reveser** sau khi người dùng nhấn nút đặt phòng.

4.2.1.4 Component: Reserve

- Chịu trách nhiệm quản lý quy trình đặt phòng, bao gồm các bước như nhập thông tin người dùng, chọn phương thức thanh toán và hiển thị thông tin chi tiết đơn hàng.
- Trách nhiệm:
 - + **Quản lý Quy trình Đặt phòng:** Component này chứa một bước điều hướng giúp người dùng di chuyển qua các bước khác nhau như nhập thông tin cá nhân, phương thức thanh toán, và xác nhận thông tin.
 - + Xử lý **Form:** Sử dụng formik để quản lý và xác thực dữ liệu đầu vào của người dùng, bao gồm tên, email, số điện thoại, và yêu cầu đặc biệt.
 - + Cung cấp lựa chọn cho phương thức thanh toán.
 - + Hiển thị thông tin chi tiết của khách sạn và các phòng sẽ đặt. Đồng thời là tổng hợp lại giá phải trả.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

- Kết nối với các component khác:
 - + Sử dụng component **Stepper** để hiển thị progress bar cho biết người dùng đang ở bước nào của quá trình đặt phòng.
 - + Sử dụng **ApiServices** để gửi thông tin đặt phòng về cho server. Sau đó nhận lại và hiển thị thông báo cho người dùng (Thành công / Thất bại).

4.2.1.5 Component: About us

- Trách nhiệm: Hiển thị thông tin về ứng dụng, đội ngũ phát triển, bao gồm các phần tử như tiêu đề, mô tả, hình ảnh minh họa.
- Kết nối với các component khác: Không có kết nối đặc biệt với các component khác, chỉ sử dụng Header và Footer

4.2.1.6 Component: Account

- Trách nhiệm: Quản lý thông tin tài khoản người dùng, bao gồm hiển thị thông tin cá nhân, chỉnh sửa và cài đặt tài khoản (thay đổi mật khẩu).
- Kết nối với các component khác:
 - + Sử dụng **ApiServices** để lấy và cập nhật dữ liệu tài khoản.
 - + Có thể điều hướng đến trang **Saved, History**

4.2.1.7 Component: Login

- Trách nhiệm: Cho phép người dùng đăng nhập vào ứng dụng. Xử lý xác thực thông tin đăng nhập và quản lý trạng thái đăng nhập.
- Kết nối với các component khác:
 - + Sử dụng **ApiServices** để xác thực thông tin đăng nhập.

4.2.1.8 Component: Register

- Trách nhiệm: Cung cấp giao diện cho người dùng đăng ký tài khoản mới. Xử lý việc nhập thông tin như tên, email, mật khẩu, và xác thực dữ liệu trước khi gửi yêu cầu tạo tài khoản.
- Kết nối với các component khác:
 - + Sử dụng **ApiServices** để gửi thông tin đăng ký tài khoản lên server.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.2.1.9 Component: Saved

- Trách nhiệm: Hiển thị danh sách các khách sạn mà người dùng đã lưu lại để xem hoặc đặt phòng sau này.
- Kết nối với các component khác:
 - + Sử dụng **ApiServices** để lấy dữ liệu danh sách khách sạn đã lưu của người dùng.
 - + Sử dụng **HotelCard** để hiển thị danh sách khách sạn.
 - + Có thể điều hướng đến trang **Hotel Details** để xem chi tiết khách sạn

4.2.1.10 Component: History

- Trách nhiệm: Hiển thị lịch sử đặt phòng của người dùng, bao gồm các thông tin như tên khách sạn, ngày đặt, trạng thái đặt phòng.
- Kết nối với các component khác:
 - + Sử dụng **ApiServices** để lấy dữ liệu lịch sử đặt phòng từ backend.
 - + Sử dụng **HotelCard** để hiển thị danh sách khách sạn.

4.2.1.11 Component: Feedback

- Trách nhiệm: Quản lý thông tin tài khoản người dùng, bao gồm hiển thị thông tin cá nhân, chỉnh sửa và cài đặt tài khoản (thay đổi mật khẩu).
- Kết nối với các component khác:
 - + Sử dụng **ApiServices** để gửi dữ liệu phản hồi từ người dùng lên server

4.2.2 Package: Components

4.2.2.1 Component: SearchBar

- Trách nhiệm: Có trách nhiệm nhận input từ người dùng và xử lý các sự kiện tìm kiếm. Khi người dùng nhập vào từ khóa và nhấn nút tìm kiếm, **SearchBar** sẽ gửi dữ liệu tìm kiếm cho **AfterSearch** và chuyển hướng đến **AfterSearch** (thông qua **navigate** của **react-router-dom**).
- Kết nối với các component khác:
 - + **SearchBar** -> **AfterSearch**: Sau khi nhận dữ liệu tìm kiếm từ SearchBar, AfterSearch sẽ xử lý và hiển thị kết quả.
 - + Ngoài ra SearchBar còn sử dụng các tiện ích từ **Utils** để phục vụ cho các việc như chuyển đổi định dạng ngày để phục vụ cho hiển thị UI.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.2.2.2 Component: HotelCard

- Đảm nhận việc hiển thị thông tin chi tiết của khách sạn bao gồm tên, địa chỉ, ảnh, giá, đánh giá và số lượt đánh giá.
- Trách nhiệm:
 - + **Hiển thị thông tin khách sạn:** Bao gồm tên, địa chỉ, hình ảnh, giá và đánh giá của khách sạn.
 - + **Quản lý yêu thích:** Cho phép người dùng đánh dấu khách sạn vào mục yêu thích.
 - + Chuyển hướng đến trang **HotelDetail** khi người dùng bấm vào nút **Book Now**.
- Kết nối với các component khác:
 - + **HotelCard sử dụng Redux:** Để lấy thông tin tỷ giá và tiền tệ hiện tại từ store.
 - + **HotelCard sử dụng Utils:** Sử dụng các hàm convertCurrency và formatCurrency để tính toán và hiển thị giá sao cho phù hợp với định dạng tiền tệ mà người dùng chọn (được lấy từ Redux).
 - + Nhận thông tin của khách sạn từ các component khác như **Home, AfterSearch, Saved,..** thông qua **props** để hiển thị thông tin của khách sạn.

4.2.2.3 Component: Filter

- Trách nhiệm: cung cấp các bộ lọc để người dùng lọc các sản phẩm theo giá, điểm đánh giá và xếp hạng sao.
 - + **Giá:** Hiển thị thanh trượt để người dùng chọn mức giá, sau đó chuyển đổi giá theo loại tiền tệ hiện tại và gửi thông tin này đến component cha thông qua hàm handleFilter
 - + **Đánh giá:** Cung cấp các checkbox cho các mức đánh giá khác nhau (tốt, rất tốt, xuất sắc, dễ chịu), cho phép người dùng lọc theo điểm đánh giá.
 - + **Xếp hạng sao:** Cho phép người dùng chọn các mức xếp hạng sao (từ 1 sao đến 5 sao) để lọc sản phẩm.
- Kết nối với các component khác:
 - + **Filter sử dụng Redux** để lấy thông tin tiền tệ và tỷ giá hối đoái.
 - + Sử dụng useDebounce (của Package Hooks) để giảm tần suất gọi API khi người dùng thay đổi bộ lọc.
 - + Kết nối với các component khác thông qua **props.handleFilter** để gửi thông tin của các tiêu chí lọc.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

4.2.2.4 Component: Stepper

- Trách nhiệm: hiển thị progress bar cho các bước trong một quy trình (steps)
 - + Nhận một danh sách stepsConfig, mỗi phần tử trong danh sách đại diện cho một bước với tiêu đề và số thứ tự. Các bước được đánh dấu là "đang thực hiện" hoặc "hoàn thành" tùy thuộc vào currentStep và trạng thái isComplete.
 - + Tính toán các khoảng cách (margins) giữa các bước để đảm bảo thanh tiến trình được căn chỉnh đúng.
- Kết nối với các component khác:
 - + **Stepper** nhận các **props** như **currentStep**, **setCurrentStep**, **isComplete**, và **setIsComplete** để theo dõi và thay đổi trạng thái của quy trình từ component cha, cho phép các bước và tiến trình được cập nhật khi người dùng di chuyển qua các bước.

4.2.3 Package: Services

4.2.3.1 Component: ApiService

- Trách nhiệm: Giao tiếp với các API (GET, POST, PUT, DELETE) để lấy và gửi dữ liệu từ/to server. Nó xử lý các yêu cầu HTTP và trả về kết quả cho các component hoặc service khác
- Kết nối với các component khác:
 - + Các component gọi **ApiServices** để lấy dữ liệu và sử dụng trong UI

4.2.4 Package: Redux

Quản lý trạng thái toàn cục của ứng dụng, lưu trữ và cập nhật dữ liệu dựa trên các action và reducer. Nó giúp các component và service có thể truy xuất và thay đổi trạng thái mà không cần truyền dữ liệu qua nhiều lớp component.

4.2.5 Package: Utils

Cung cấp các hàm dùng chung như định dạng ngày tháng năm, chuỗi, chuyển đổi dữ liệu, kiểm tra điều kiện, kiểm tra tính hợp lệ cho input,... cho các thành phần khác của ứng dụng (VD: SearchBar sẽ dùng hàm formatDate để định dạng ngày giờ cho check-in và check-out). Đồng thời giúp giảm thiểu việc lặp lại mã nguồn đối với các hàm có thể tái sử dụng.

BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

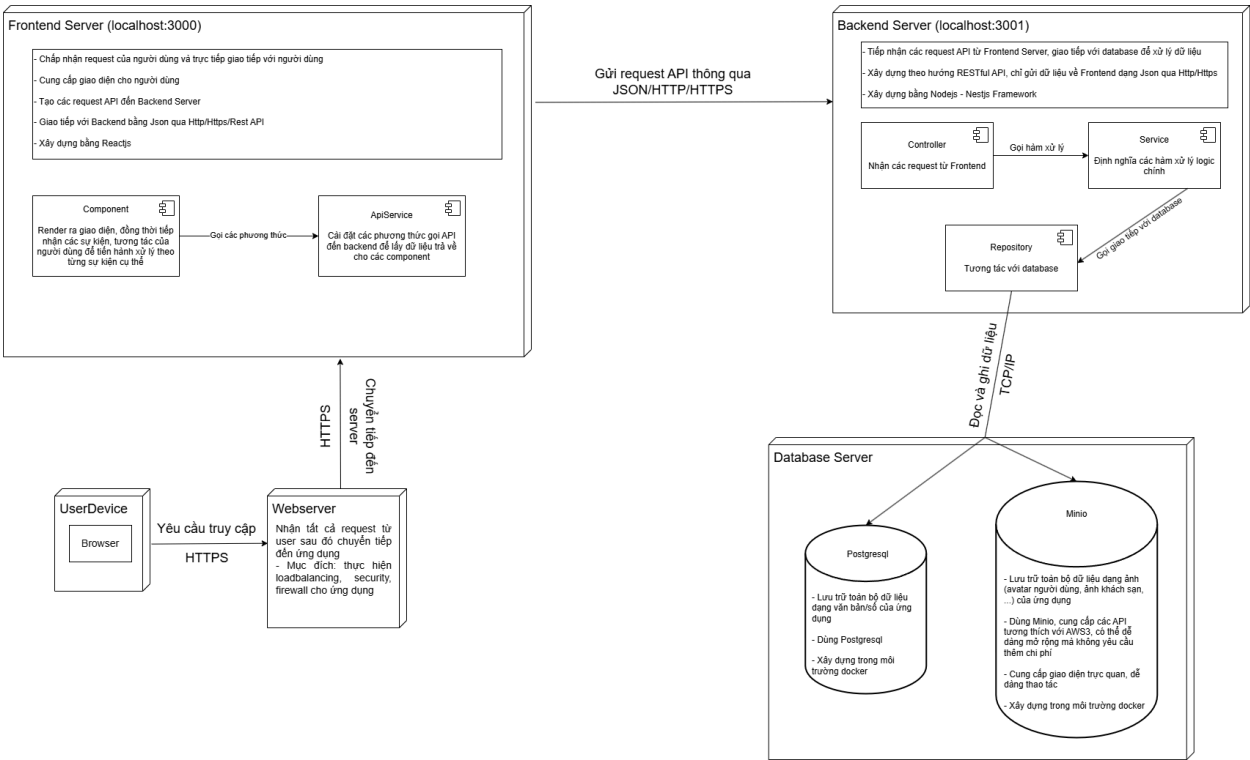
4.2.6 Package: Hooks

Package Hook có trách nhiệm quản lý các hooks — các hàm hoặc tính năng đặc biệt mà giúp xử lý các trạng thái (state) và hiệu ứng phụ (side effects) trong ứng dụng.

4.2.7 Package: Routes

Package Routes trong ứng dụng có trách nhiệm chính là điều hướng người dùng giữa các màn hình (pages) khác nhau trong hệ thống. Đảm nhận việc quản lý các URL và chuyển hướng người dùng đến các trang tương ứng khi có yêu cầu từ người dùng hoặc ứng dụng.

5.Deployment



BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

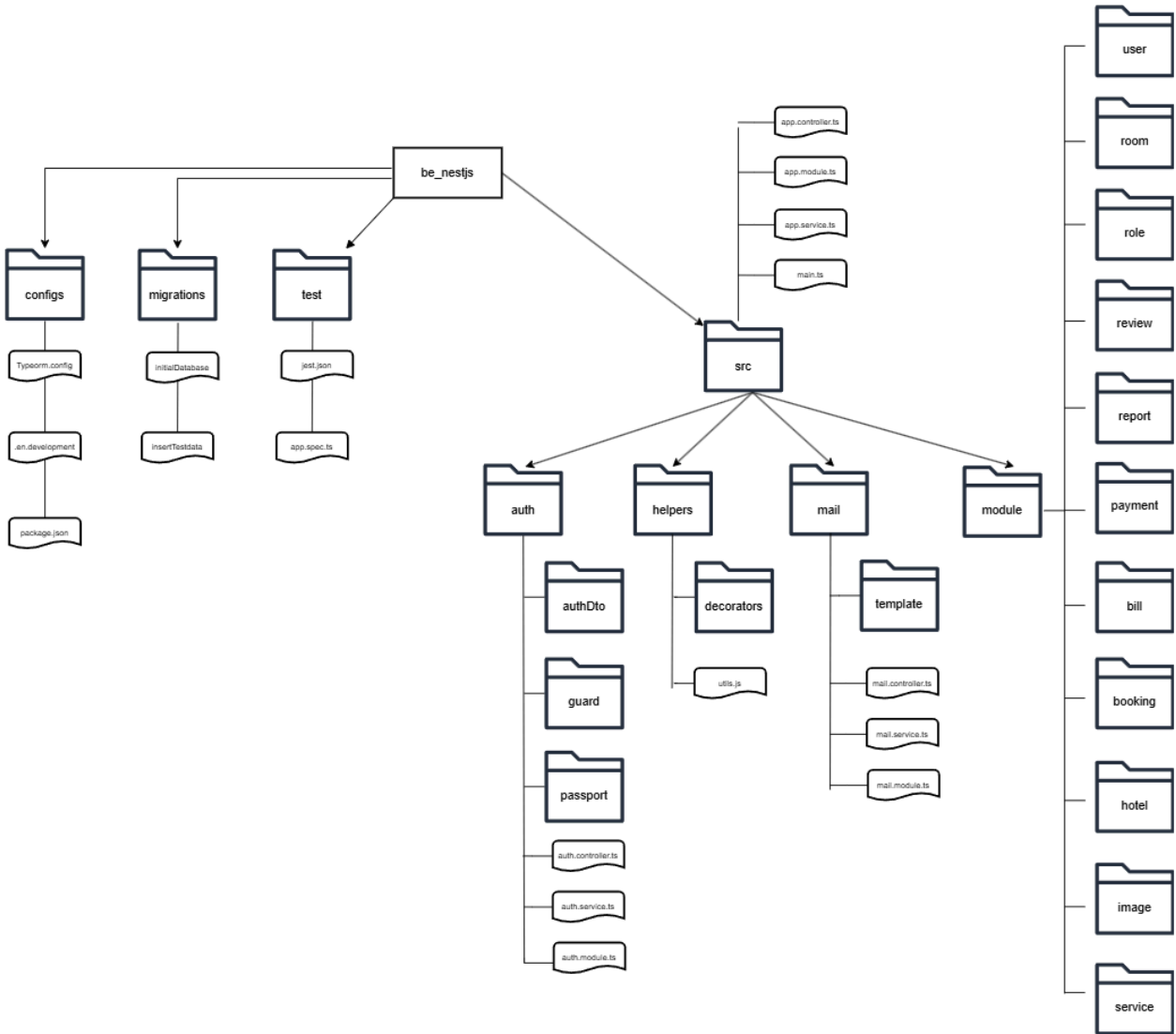
- Ứng dụng thực tế sẽ được triển khai trên 3 thiết bị. Là 3 máy chủ bao gồm:
 - + Frontend: máy chủ này sẽ chạy server frontend ở port 3000 để phục vụ trang web cho người dùng. Đồng thời đây cũng là nơi tiếp nhận các request, tương tác và, là nơi giao tiếp chính với người dùng. Máy chủ sẽ tiếp nhận yêu cầu người dùng, xử lý hoặc gửi nó đến server backend xử lý sau đó lấy thông tin hiển thị lại cho người dùng
 - + Backend: là nơi chạy server backend ở port 3001. Server này là nơi chứa các logic xử lý chính của ứng dụng. Nó tiếp nhận các request từ server frontend, giao tiếp với database để đọc/ghi dữ liệu, xử lý nó và gửi kết quả lại cho server frontend hiển thị cho người dùng
 - + Database: đây là nơi lưu trữ dữ liệu chính của ứng dụng. Mọi thông tin dữ liệu về của ứng dụng bao gồm hình ảnh hay dữ liệu người dùng đều sẽ được lưu trữ ở đây

⇒ Việc triển khai trên 3 máy chủ khác nhau đảm bảo tính ổn định của ứng dụng khi phục vụ trong thời gian lâu dài. Nó cũng giúp cải thiện tính đơn giản trong quá trình bảo trì hệ thống

- Để tiện cho quá trình demo báo cáo cuối kì, nhóm chỉ thực hiện demo chạy cả 3 server trên cùng một máy của mỗi thành viên
- Thực tế sẽ có thêm một máy chủ webserver đứng trước ứng dụng, nhận tất cả request từ user đến người dùng để thực hiện load balancing, security cho ứng dụng. Tuy nhiên trong phạm vi project môn học, nhóm quyết định không thực hiện phần này mà chỉ tập trung vào vận hành server thực tế

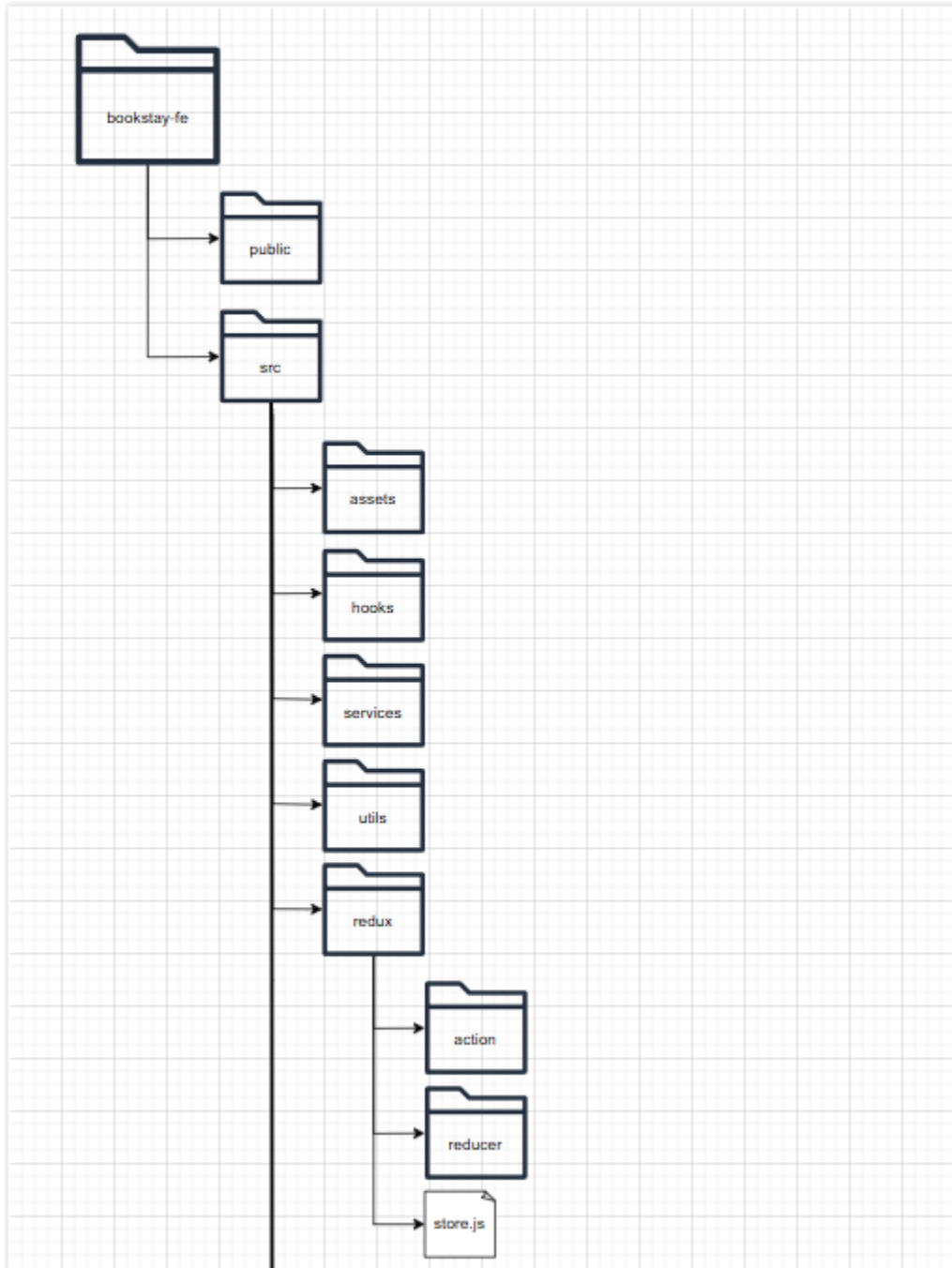
6.Implementation View

6.1 Back-end

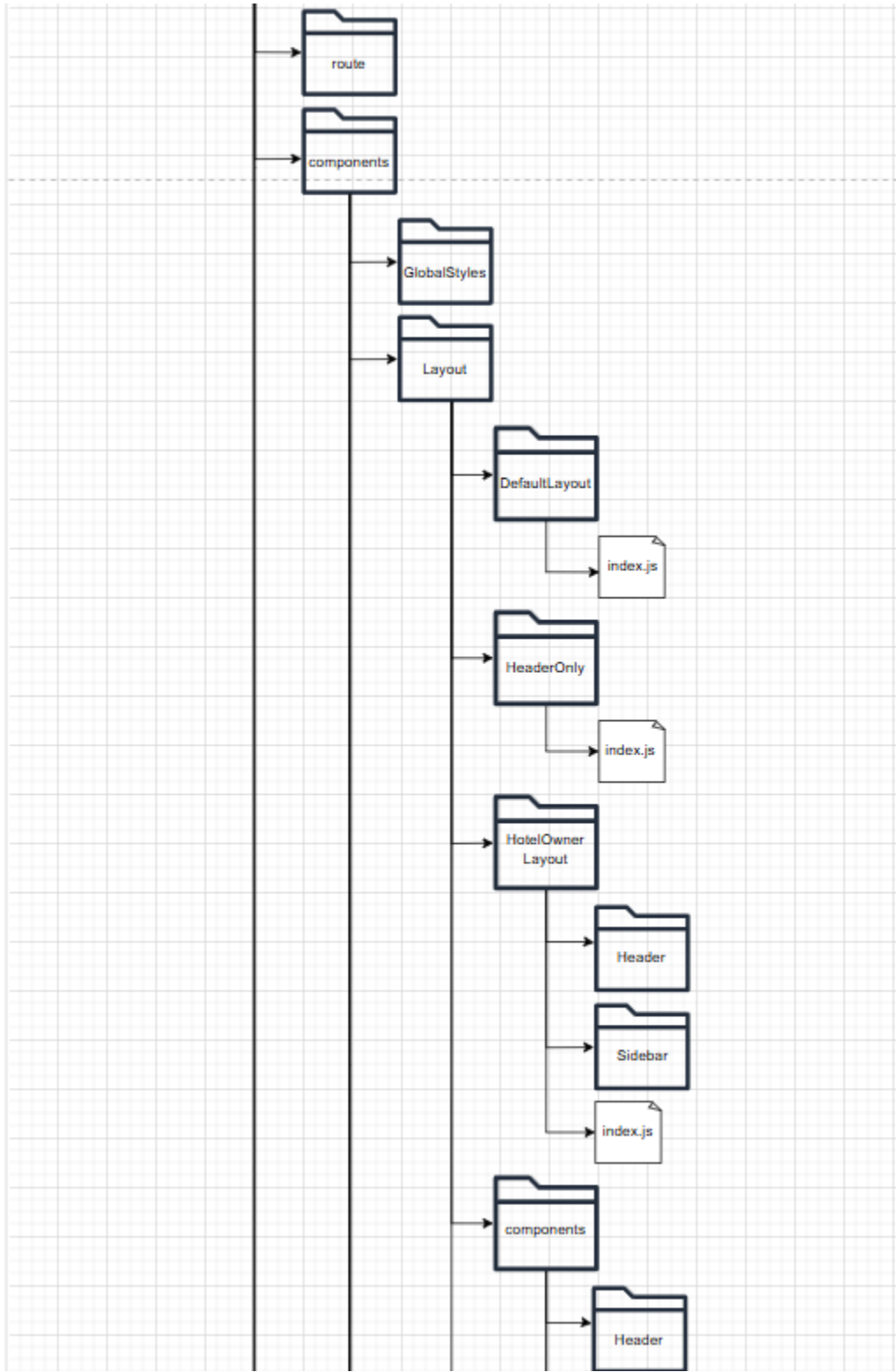


BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

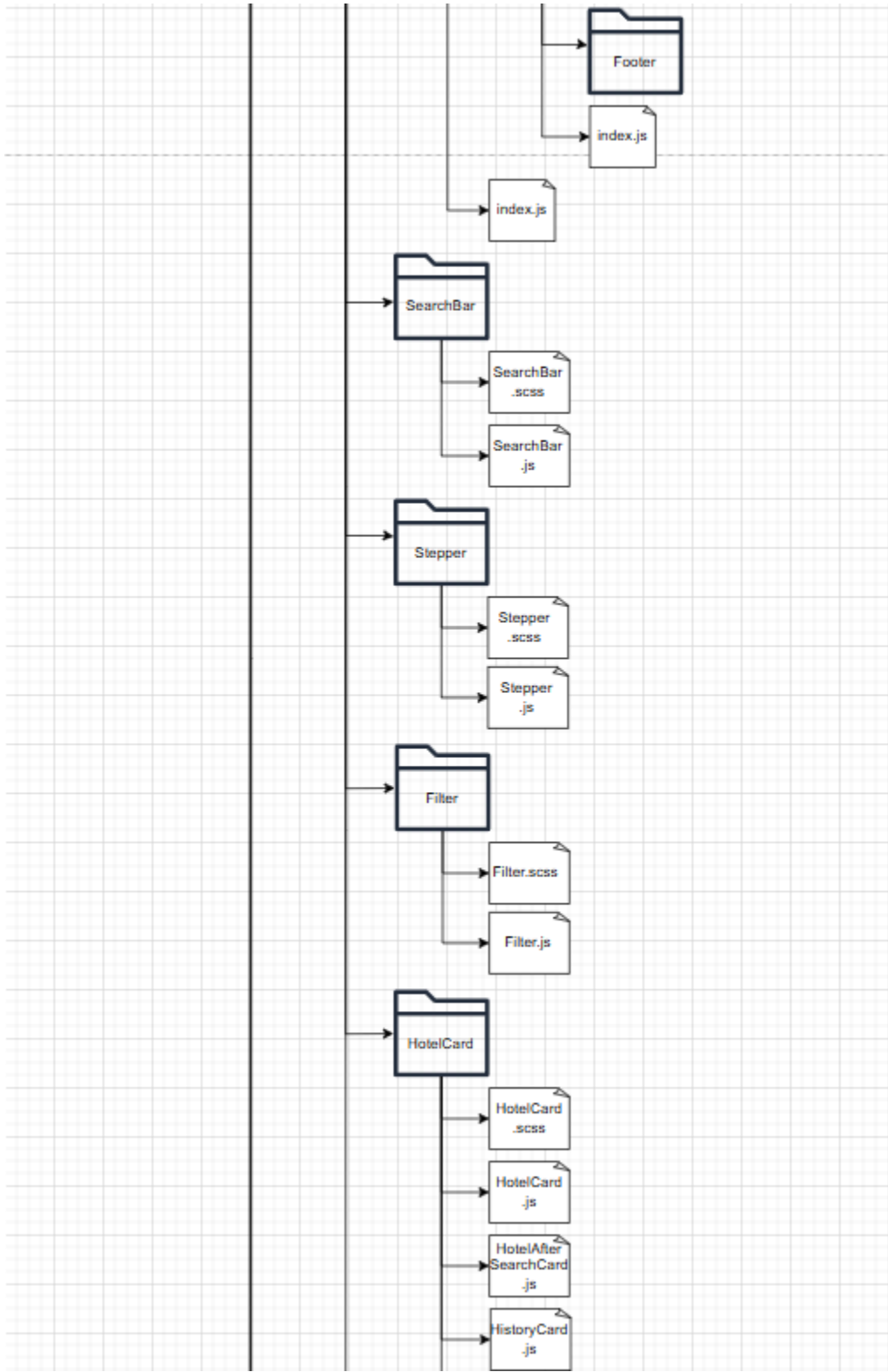
6.2 Front-end



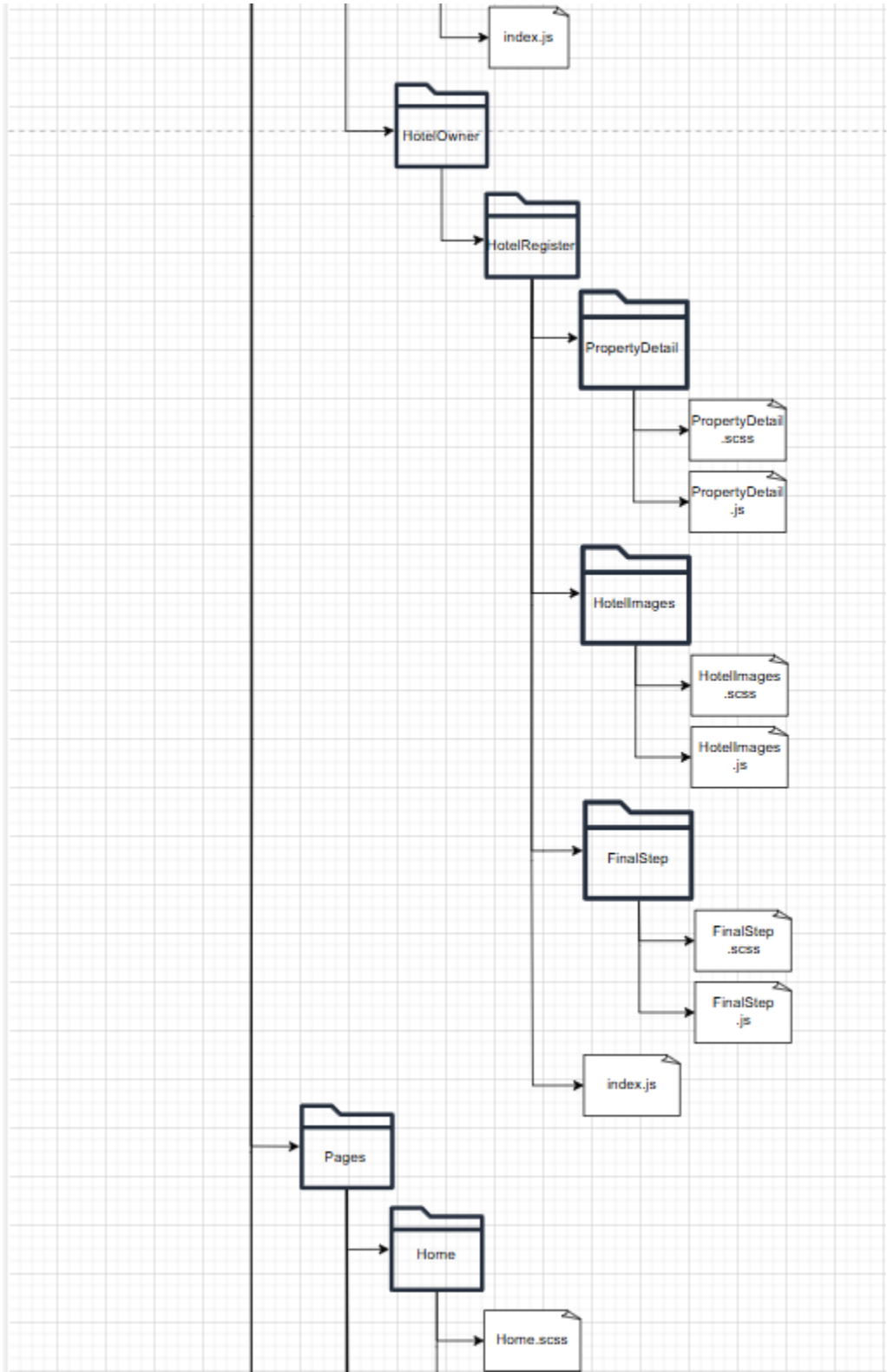
BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	



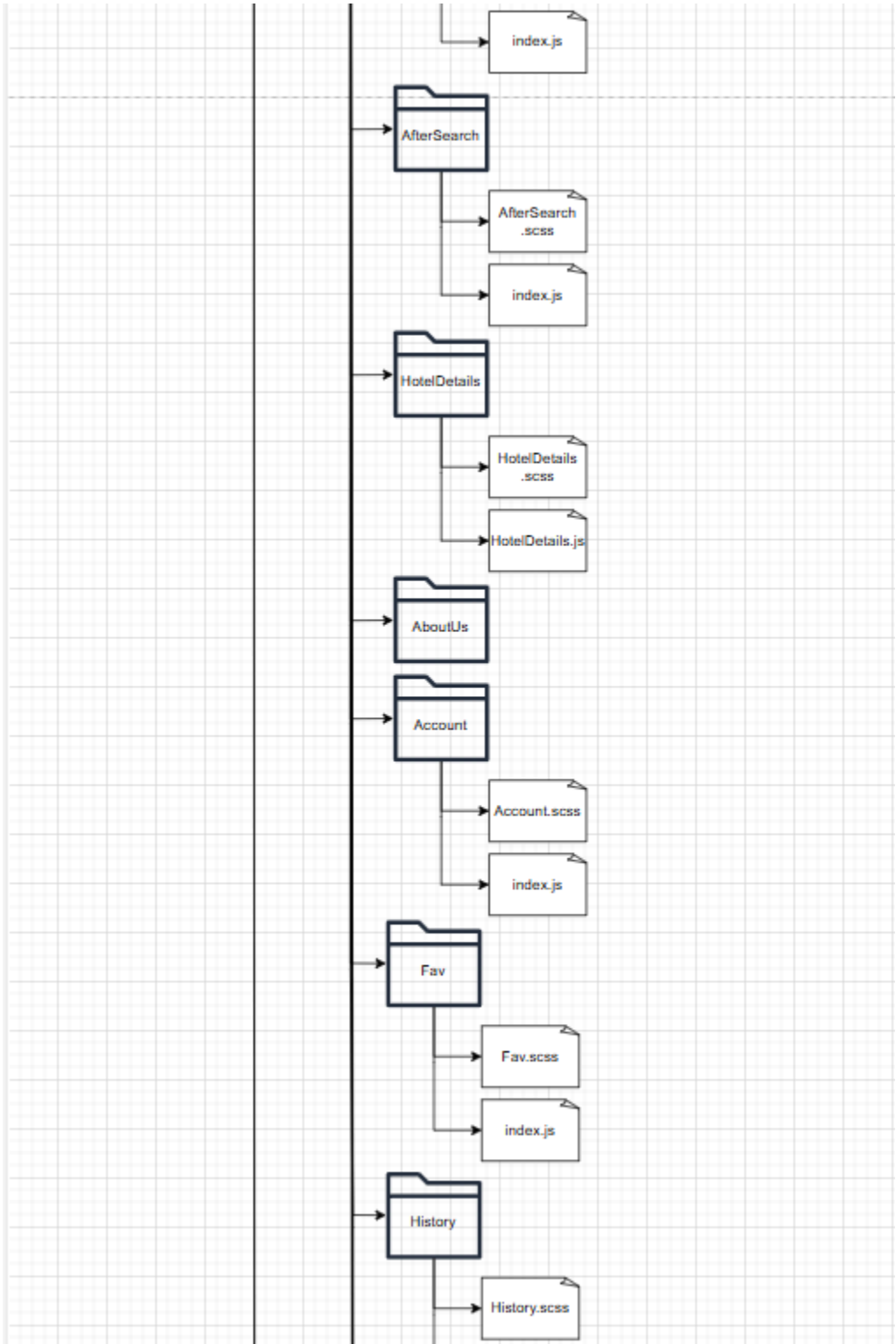
BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	



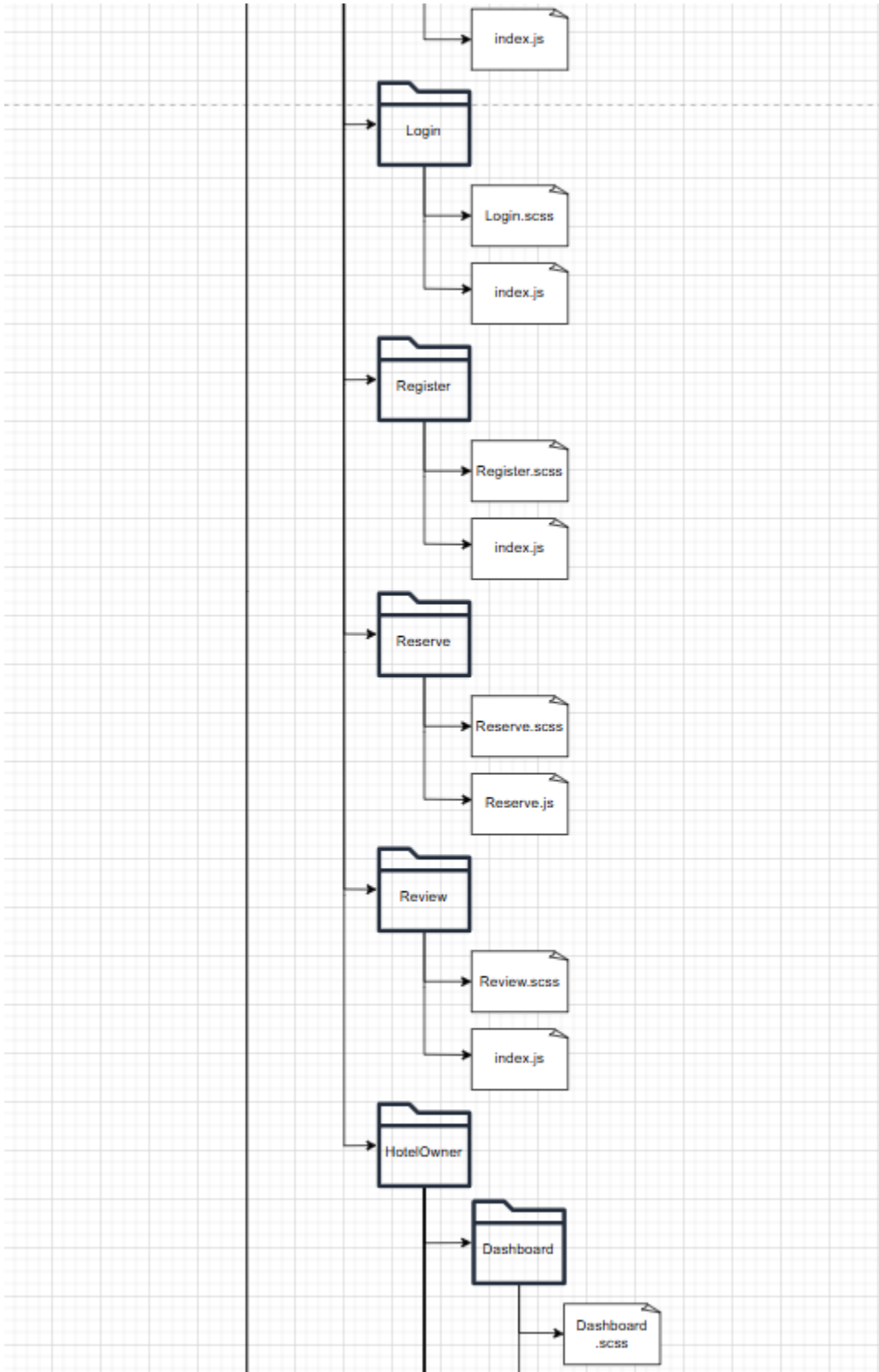
BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	



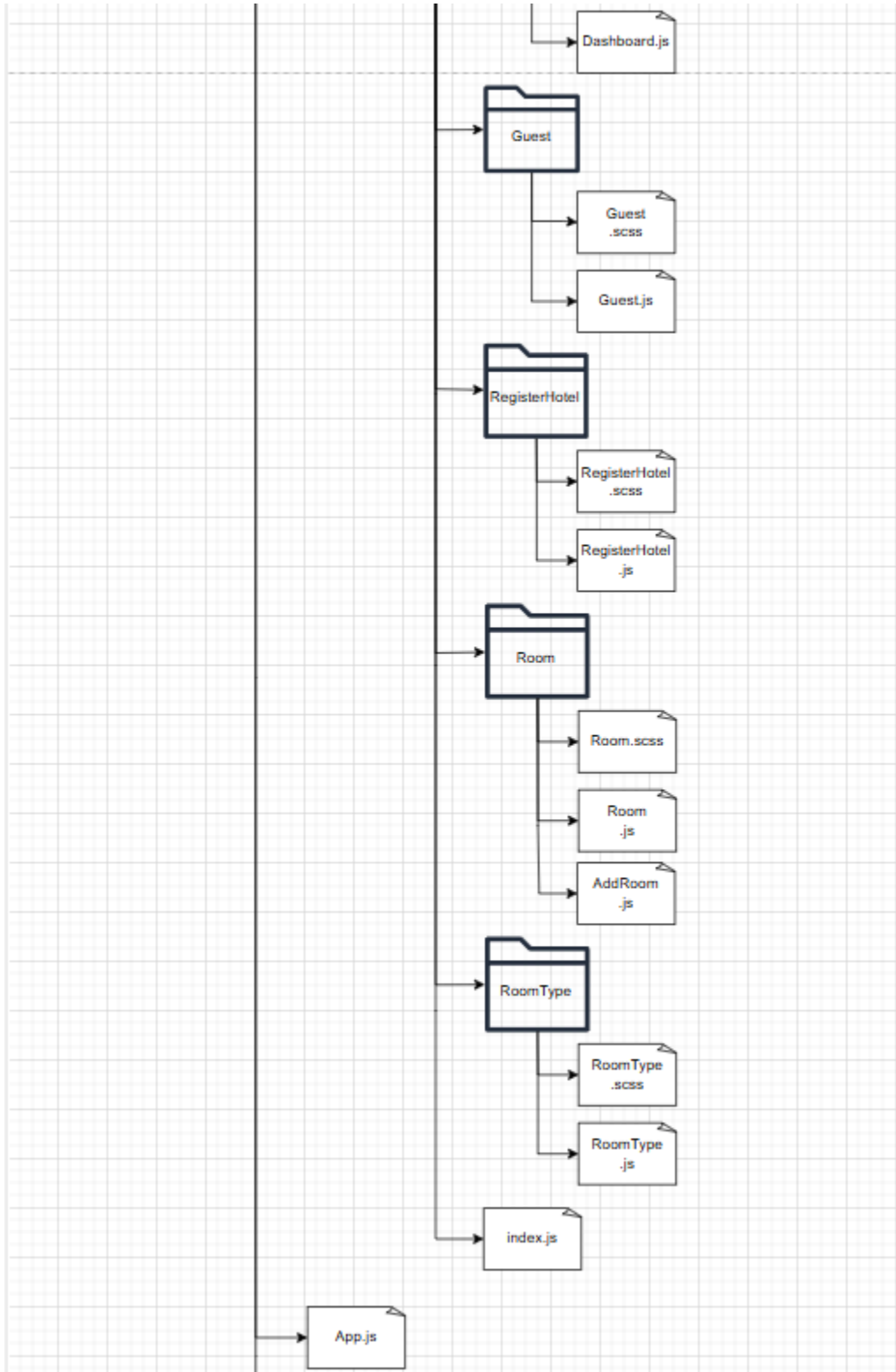
BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	



BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	



BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	



BookaStay	Version: 1.0.0.0
Software Architecture Document	Date: 27/12/2024
DOC-PR813717	

