
Group 25

**Backpacking Travel Website
Software Architecture Document**

Version <1.0>

Backpacking Travel Website	Version: 1.0
Software Architecture Document	Date: 28/11/2022
<sad>	

Revision History

Date	Version	Description	Author
28/11/2022	1.0	Viết phiên bản Software Architecture document đầu tiên	Group 25

Backpacking Travel Website	Version: 1.0
Software Architecture Document	Date: 28/11/2022
<sad>	

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms and Abbreviations	4
1.4	References	4
2.	Architectural Goals and Constraints	4
3.	Use-Case Model	5
4.	Logical View	6
4.1	Model	8
4.2	View	8
4.3	Controller	8
4.4	Google Maps	8
4.5	Google Places	8
4.6	Travel Advisor	9
4.7	Booking.com	9
4.8	VietQR	9
5.	Deployment	9
6.	Implementation View	9

Backpacking Travel Website	Version: 1.0
Software Architecture Document	Date: 28/11/2022
<sad>	

Software Architecture Document

1. Introduction

Tài liệu Software Architecture Document cung cấp cái nhìn tổng quan về kiến trúc phần mềm được áp dụng cho hệ thống Backpacking Travel Website. Cụ thể hơn, đây là website thông tin về du lịch và là bên trung gian cung cấp, đánh giá dịch vụ du lịch để mang đến trải nghiệm tốt hơn cho user.

1.1 Purpose

Tài liệu này cung cấp góc nhìn tổng quan toàn diện về kiến trúc của hệ thống, sử dụng một số mô hình kiến trúc khác nhau để mô tả các khía cạnh khác nhau của hệ thống. Mục đích của tài liệu là nắm bắt và truyền đạt các quyết định về mặt kiến trúc quan trọng đã được thực hiện trên hệ thống.

1.2 Scope

Tài liệu Software Architecture Document chỉ mô tả kiến trúc của hệ thống Backpacking Travel Website. Nó không giải thích cách dịch vụ bên ngoài được thiết kế.

1.3 Definitions, Acronyms and Abbreviations

MVC – Kiến trúc Model View Controller.

1.4 References

Course Registration System SAD:

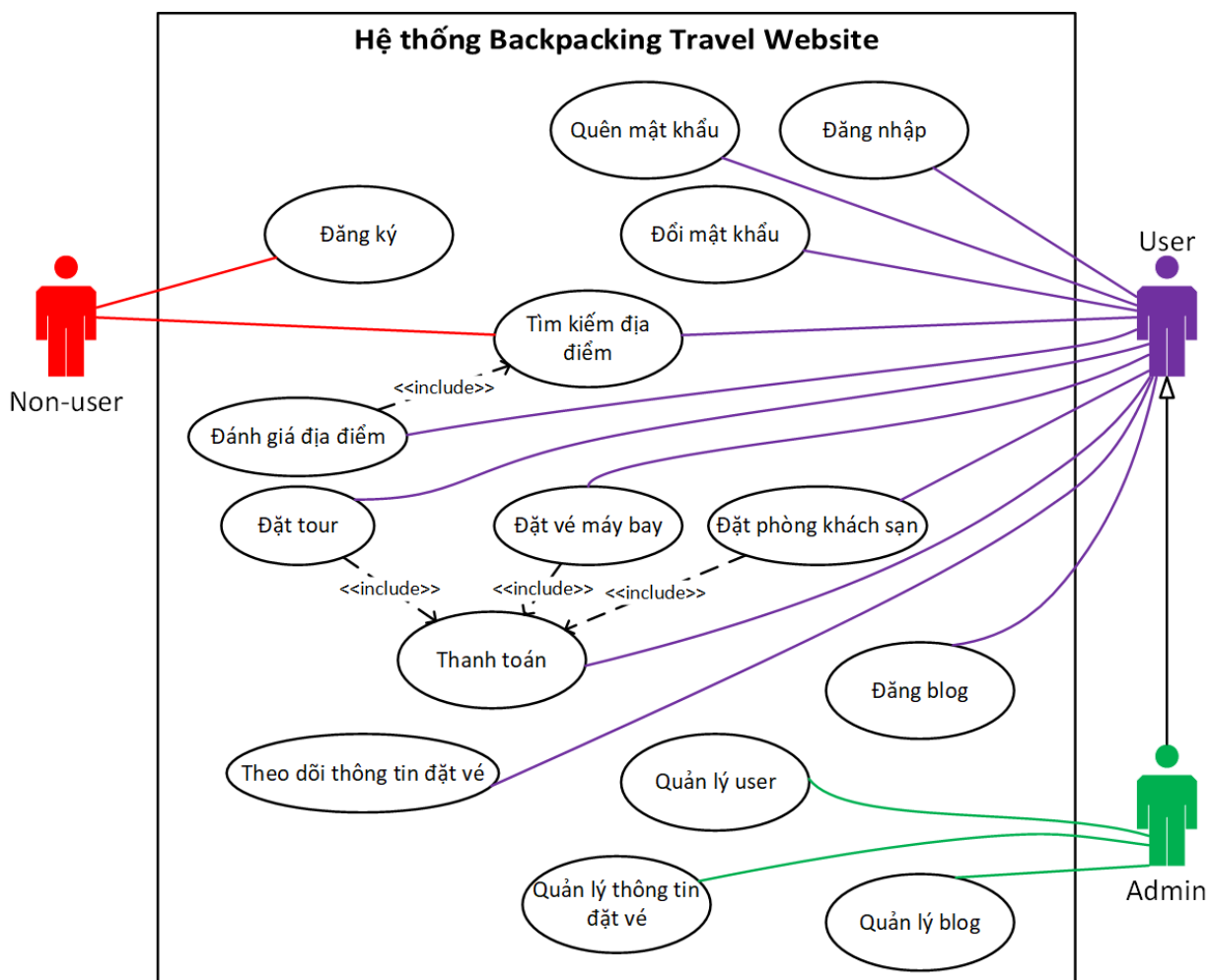
<https://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm>

2. Architectural Goals and Constraints

- Ngôn ngữ lập trình: HTML, CSS, Javascript, Python.
- Môi trường thực thi: Web.
- Tính dễ sử dụng: Có cửa sổ help, có Navigation.
- Độ tin cậy: thời gian thất bại (timeout) là 10s.
- Tính dễ bảo trì và mở rộng: Code có comment, tránh lặp code.

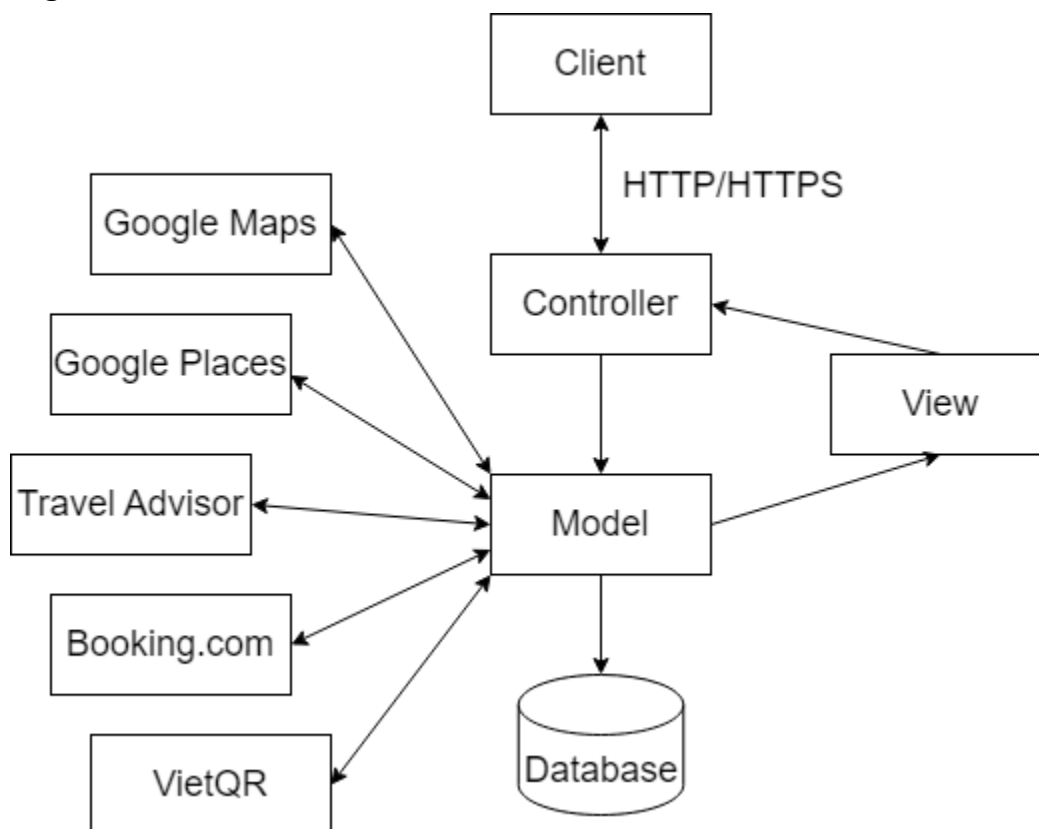
Backpacking Travel Website	Version: 1.0
Software Architecture Document	Date: 28/11/2022
<sad>	

3. Use-Case Model



Backpacking Travel Website	Version: 1.0
Software Architecture Document	Date: 28/11/2022
<sad>	

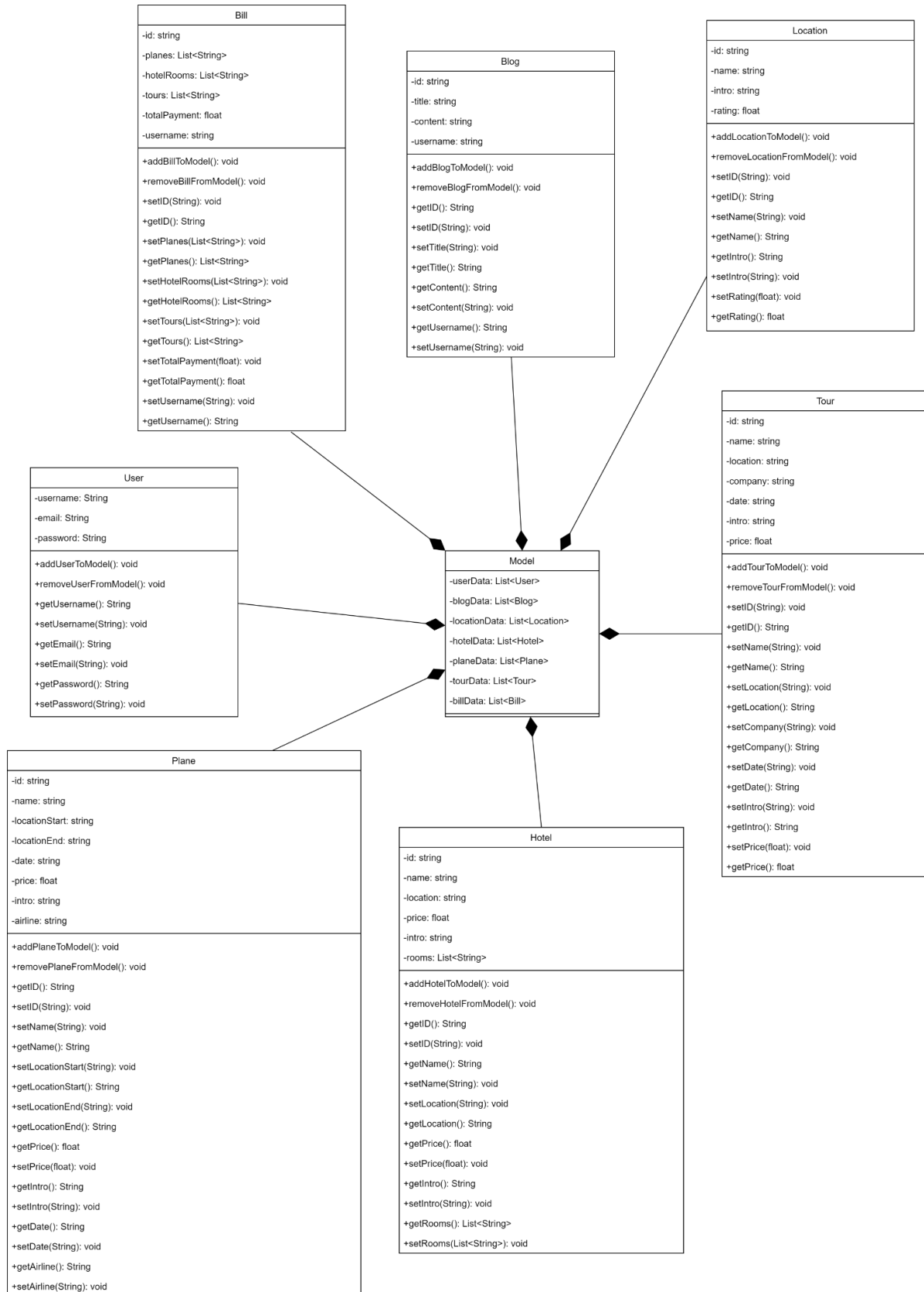
4. Logical View



4.1 Model

- Nhiệm vụ: Model là component trung tâm của MVC, là cầu nối giữa View và Controller. Nó quản lý dữ liệu, logic và các ràng buộc của web. Nó là cấu trúc dữ liệu động của web, độc lập với giao diện người dùng.
- Ngôn ngữ sử dụng: Django/Python.
- Class diagram:

Backpacking Travel Website	Version: 1.0
Software Architecture Document	Date: 28/11/2022
<sad>	



Backpacking Travel Website	Version: 1.0
Software Architecture Document	Date: 28/11/2022
<sad>	

- Mô tả class diagram:

Class	Description
User	<p>Có các thuộc tính:</p> <ul style="list-style-type: none"> - username: tên đăng nhập - email - password: mật khẩu <p>Có các phương thức:</p> <ul style="list-style-type: none"> - addUserToModel(): Thêm dữ liệu user vào cơ sở dữ liệu - removeUserFromModel(): Loại bỏ dữ liệu user khỏi cơ sở dữ liệu - getter và setter với mỗi thuộc tính trong User
Blog	<p>Có các thuộc tính:</p> <ul style="list-style-type: none"> - id - title: Tiêu đề blog - content: Nội dung blog - username: Tên username viết blog <p>Có các phương thức:</p> <ul style="list-style-type: none"> - addBlogToModel(): Thêm dữ liệu blog vào cơ sở dữ liệu - removeBlogFromModel(): Loại bỏ dữ liệu blog khỏi cơ sở dữ liệu - getter và setter với mỗi thuộc tính trong Blog
Location	<p>Có các thuộc tính:</p> <ul style="list-style-type: none"> - id - name: Tên địa điểm - intro: Giới thiệu địa điểm - rating: Đánh giá <p>Có các phương thức:</p> <ul style="list-style-type: none"> - addLocationToModel(): Thêm dữ liệu địa điểm vào cơ sở dữ liệu - removeLocationFromModel(): Loại bỏ dữ liệu địa điểm khỏi cơ sở dữ liệu - getter và setter với mỗi thuộc tính trong Location
Hotel	<p>Có các thuộc tính:</p> <ul style="list-style-type: none"> - id - name: Tên khách sạn - location: Địa chỉ - price: Giá phòng - intro: Giới thiệu khách sạn - rooms: Danh sách phòng <p>Có các phương thức:</p> <ul style="list-style-type: none"> - addHotelToModel(): Thêm dữ liệu khách sạn vào cơ sở dữ liệu - removeHotelFromModel(): Loại bỏ dữ liệu khách sạn khỏi cơ sở dữ liệu - getter và setter với mỗi thuộc tính trong Hotel

Backpacking Travel Website	Version: 1.0
Software Architecture Document	Date: 28/11/2022
<sad>	

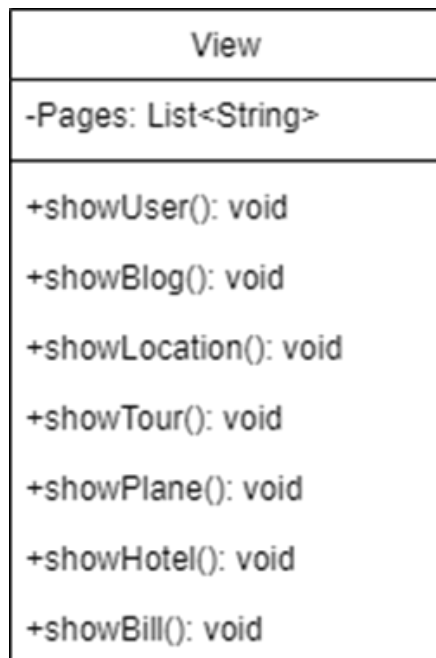
Plane	<p>Có các thuộc tính:</p> <ul style="list-style-type: none"> - id - name: Tên chuyến bay - locationStart: Điểm khởi hành - locationEnd: Điểm đến - date: Ngày bay - price: Giá vé - intro: Thông tin thêm - airline: Đường bay <p>Có các phương thức:</p> <ul style="list-style-type: none"> - addPlaneToModel(): Thêm dữ liệu chuyến bay vào cơ sở dữ liệu - removePlaneFromModel(): Loại bỏ dữ liệu chuyến bay khỏi cơ sở dữ liệu - getter và setter với mỗi thuộc tính trong Plane
Tour	<p>Có các thuộc tính:</p> <ul style="list-style-type: none"> - id - name: Tên tour - location: Tên địa điểm - company: Tên công ty tổ chức tour - date: Ngày đi tour - intro: Giới thiệu tour - price: Giá vé tour <p>Có các phương thức:</p> <ul style="list-style-type: none"> - addTourToModel(): Thêm dữ liệu tour vào cơ sở dữ liệu - removeTourFromModel(): Loại bỏ dữ liệu tour khỏi cơ sở dữ liệu - getter và setter với mỗi thuộc tính trong Tour
Bill	<p>Có các thuộc tính:</p> <ul style="list-style-type: none"> - id - planes: Thông tin các chuyến bay cần thanh toán - hotelRooms: Thông tin các phòng khách sạn cần thanh toán - tours: Thông tin các tour cần thanh toán - totalPayment: Tổng chi phí cần thanh toán - username: Tên user thanh toán <p>Có các phương thức:</p> <ul style="list-style-type: none"> - addBillToModel(): Thêm dữ liệu hoá đơn vào cơ sở dữ liệu - removeBillFromModel(): Loại bỏ dữ liệu hoá đơn khỏi cơ sở dữ liệu - getter và setter với mỗi thuộc tính trong Tour

4.2 View

- Nhiệm vụ: View chịu trách nhiệm hiển thị giao diện cho người dùng thông qua hiển thị dữ liệu từ Model. Nó là bất kỳ dạng thông tin nào mà có thể hiển thị cho người dùng. View cung cấp các biểu diễn trực quan của dữ liệu.

- Ngôn ngữ sử dụng: HTML, CSS, Javascript.
- Class diagram:

Backpacking Travel Website	Version: 1.0
Software Architecture Document	Date: 28/11/2022
<sad>	



- Mô tả class diagram:

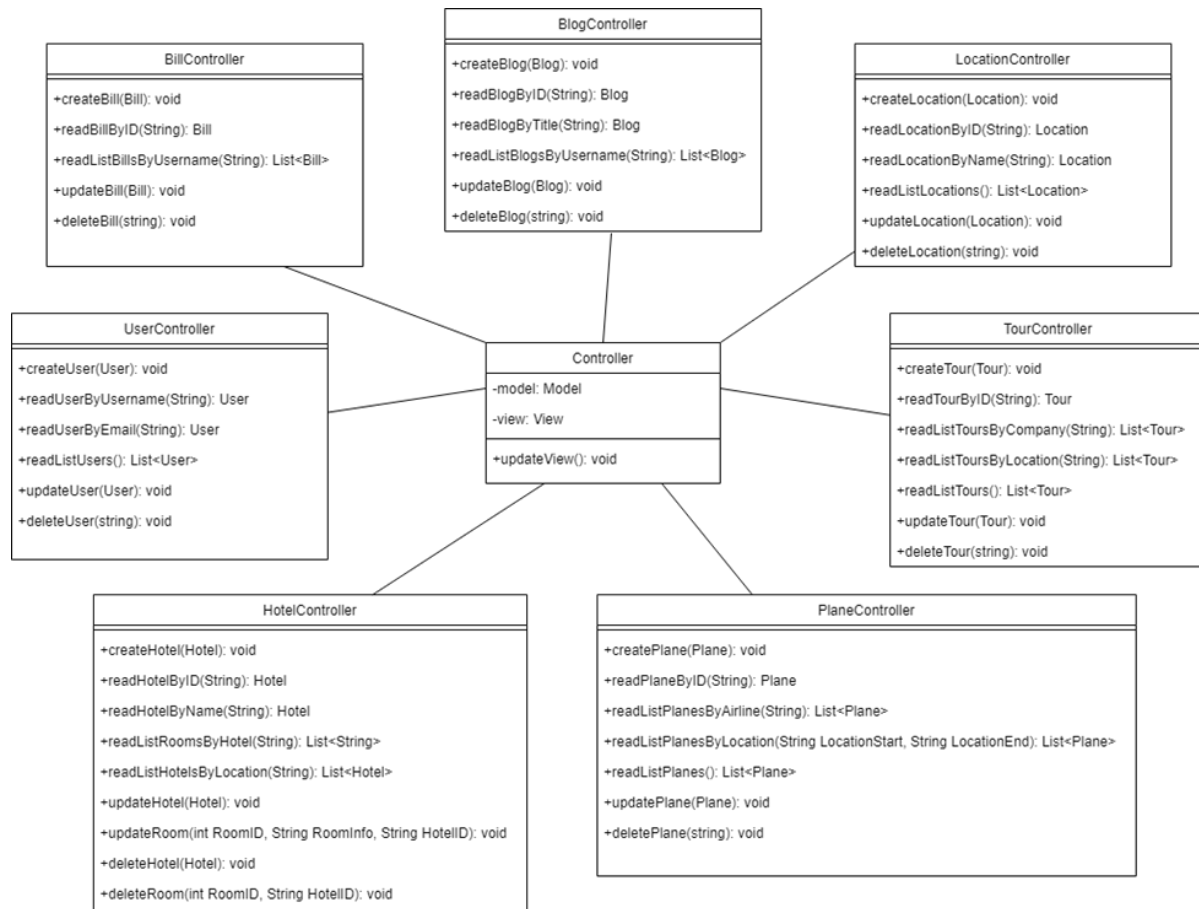
Properties	Description
Pages: Các trang hiển thị	Các phương thức hiển thị nội dung các trường dữ liệu trong cơ sở dữ liệu

4.3 Controller

- Nhiệm vụ: Controller nhận yêu cầu, xử lý và chuyển thành câu lệnh cho View hoặc Model để cập nhật tương ứng. Nó kiểm soát việc tương tác giữa Model và View. Controller hoạt động như một giao diện giữa các Model, View và thiết bị đầu vào được liên kết.

- Ngôn ngữ sử dụng: Django/Python.
- Class diagram:

Backpacking Travel Website	Version: 1.0
Software Architecture Document	Date: 28/11/2022
<sad>	



- Mô tả class diagram:

Class	Description methods
UserController	<ul style="list-style-type: none"> - createUser: Tạo user. - readUserByUsername: Đọc user bằng tên user - readUserByEmail: Đọc user bằng email - readListUser: Đọc danh sách user - updateUser: Cập nhật user - deleteUser: Xóa user
BlogController	<ul style="list-style-type: none"> - createBlog: Tạo blog. - readBlogById: Đọc blog bằng ID - readBlogByTitle: Đọc blog bằng tiêu đề - readListBlogsByUsername: Đọc danh sách blog bằng tên user - updateBlog: Cập nhật blog - deleteBlog: Xóa user
LocationController	<ul style="list-style-type: none"> - createLocation: Tạo địa điểm - readLocationById: Đọc địa điểm bằng ID - readLocationByName: Đọc địa điểm bằng tên địa điểm - readListLocation: Đọc danh sách địa điểm

Backpacking Travel Website	Version: 1.0
Software Architecture Document	Date: 28/11/2022
<sad>	

	<ul style="list-style-type: none"> - updateLocation: Cập nhật địa điểm - deleteLocation: Xóa địa điểm
HotelController	<ul style="list-style-type: none"> - createHotel: Tạo khách sạn - readHotelByID: Đọc khách sạn bằng ID - readHotelByName: Đọc khách sạn bằng tên khách sạn - readListRoomsByHotel: Đọc danh sách phòng bằng tên khách sạn - readListHotelsByLocation: Đọc danh sách khách sạn bằng tên địa điểm - updateHotel: Cập nhật khách sạn - updateRoom: Cập nhật phòng khách sạn - deleteHotel: Xóa khách sạn - deleteRoom: Xóa phòng khách sạn
PlaneController	<ul style="list-style-type: none"> - createPlane: Tạo chuyến bay - readPlaneByID: Đọc chuyến bay bằng ID - readListPlanesByAirline: Đọc danh sách chuyến bay bằng đường bay - readListPlanesByLocation: Đọc danh sách chuyến bay bằng địa điểm - readListPlanes: Đọc danh sách chuyến bay - updatePlane: Cập nhật chuyến bay - deletePlane: Xóa chuyến bay
TourController	<ul style="list-style-type: none"> - createTour: Tạo tour - readTourByID: Đọc tour bằng ID - readListToursByCompany: Đọc danh sách tour bằng tên công ty tổ chức - readListToursByLocation: Đọc danh sách tour bằng tên địa điểm - readListTours: Đọc danh sách tour - updateTour: Cập nhật tour - deleteTour: Xóa tour
BillController	<ul style="list-style-type: none"> - createBill: Tạo hoá đơn - readBillByID: Đọc hoá đơn bằng ID - readListBillsByUsername: Đọc danh sách hoá đơn bằng tên user - updateBill: Cập nhật hoá đơn - deleteBill: Xóa hóa đơn

4.4 Google Maps

- Nhiệm vụ: Dùng API để hiển thị bản đồ.
- Ngôn ngữ sử dụng: Django/Python (HTTP Request).

4.5 Google Places

- Nhiệm vụ: Dùng API để lấy dữ liệu về địa điểm khách sạn.
- Ngôn ngữ sử dụng: Django/Python (HTTP Request).

4.6 Travel Advisor

- Nhiệm vụ: Dùng API để truy vấn giá chuyến bay, đặt phòng khách sạn, các địa điểm thu hút...
- Ngôn ngữ sử dụng: Django/Python (HTTP Request).

4.7 Booking.com

- Nhiệm vụ: Dùng API lấy dữ liệu để phục vụ việc đặt vé máy bay, phòng khách sạn.

Backpacking Travel Website	Version: 1.0
Software Architecture Document	Date: 28/11/2022
<sad>	

- Ngôn ngữ sử dụng: Django/Python (HTTP Request).

4.8 VietQR

- Nhiệm vụ: Dùng API lấy dữ liệu danh sách ngân hàng phục vụ việc thanh toán.
- Ngôn ngữ sử dụng: Django/Python (HTTP Request).

5. Deployment

6. Implementation View