

**BỘ GIÁO DỤC VÀ ĐÀO TẠO      VIỆN HÀN LÂM KHOA HỌC VÀ  
CÔNG NGHỆ VIỆT NAM**

**HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM**

---



**Ngô Văn Trung**

**NGHIÊN CỨU LỰA CHỌN CÁC DỊCH VỤ WEB ỨNG DỤNG  
TRONG XÂY DỰNG CÁC HỆ THỐNG HƯỚNG DỊCH VỤ DỰA  
TRÊN MÔ HÌNH ĐỒ THỊ**

**LUẬN VĂN THẠC SĨ NGÀNH MÁY TÍNH**

**Hà Nội – Năm 2022**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO      VIỆN HÀN LÂM KHOA HỌC VÀ  
CÔNG NGHỆ VIỆT NAM**

**HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM**

---



**Ngô Văn Trung**

**NGHIÊN CỨU LỰA CHỌN CÁC DỊCH VỤ WEB ỨNG DỤNG  
TRONG XÂY DỰNG CÁC HỆ THỐNG HƯỚNG DỊCH VỤ DỰA  
TRÊN MÔ HÌNH ĐỒ THỊ**

Chuyên ngành: Hệ thống thông tin

Mã số: 8480104

**LUẬN VĂN THẠC SĨ NGÀNH MÁY TÍNH**

NGƯỜI HƯỚNG DẪN KHOA HỌC

**PGS.TS. Nguyễn Việt Anh**

**Hà Nội – Năm 2022**

## **LỜI CAM ĐOAN**

Nội dung trong luận văn này, tôi xin cam đoan là đúng với nội dung đề cương và nội dung của thầy hướng dẫn đã hướng dẫn giao cho tôi. Các nội dung trong luận văn, và các trích lục, tài liệu đều chính xác., Tôi xin chịu trách nhiệm hoàn toàn nếu xảy ra sai sót.

**Tác giả luận văn**

**NGÔ VĂN TRUNG**

**LỜI CẢM ƠN:**

Với sự chỉ dẫn và hỗ trợ của giáo viên tại Khoa: Công nghệ Thông tin & Viễn thông, Học viện Khoa học và Công nghệ cũng như sự hỗ trợ từ bạn bè và đồng nghiệp, đặc biệt là sự hướng dẫn tận tình của PGS.TS.Nguyễn Việt Anh cộng thêm những nỗ lực, cố gắng bản thân mình, cho đến nay thì đề tài đã được hoàn thành.

Trong quá trình hoàn thiện làm luận văn, học viên cũng đã cố gắng rất nhiều, nhưng do kiến thức và kinh nghiệm còn hạn hẹp, khó tránh khỏi những sai sót, rất mong nhận được lời khuyên của các giáo viên và những chủ đề của thầy cô để luận văn hoàn thiện hơn.

Tôi xin chân thành cảm ơn.

## MỤC LỤC

	Trang
<b>ĐẶT VẤN ĐỀ.....</b>	<b>9</b>
<b>DANH MỤC CÁC KÝ HIỆU .....</b>	<b>11</b>
<b>MỞ ĐẦU.....</b>	<b>12</b>
<b>CHƯƠNG 1: TỔNG QUAN VỀ DỊCH VỤ WEB, KIẾN TRÚC HƯỚNG DỊCH VỤ SOA VÀ MÔ HÌNH ĐỒ THỊ.....</b>	<b>13</b>
<b>1.1. Tổng quan Dịch vụ Web .....</b>	<b>13</b>
<b>1.1.1. Khái niệm Dịch vụ Web .....</b>	<b>14</b>
<b>1.1.2. Mô hình sử dụng Dịch vụ Web.....</b>	<b>14</b>
<b>1.1.3. Kiến trúc của web service .....</b>	<b>15</b>
<b>1.1.4. Các thành phần của web service .....</b>	<b>16</b>
<b>1.1.5. Chất lượng Dịch vụ Web .....</b>	<b>18</b>
<b>1.2. Kiến trúc hướng dịch vụ SOA.....</b>	<b>19</b>
<b>1.2.1. Khái niệm SOA .....</b>	<b>19</b>
<b>1.2.2. Nguyên tắc của SOA.....</b>	<b>20</b>
<b>1.2.3. Tính chất SOA.....</b>	<b>21</b>
<b>1.2.4. Ưu và nhược điểm của SOA. ....</b>	<b>22</b>
<b>1.3. Mô hình đồ thị.....</b>	<b>24</b>
<b>1.3.1. Một số khái niệm về đồ thị.....</b>	<b>25</b>
<b>1.3.2. Biểu diễn đồ thị .....</b>	<b>26</b>
<b>1.3.3. Ứng dụng của đồ thị .....</b>	<b>27</b>
<b>CHƯƠNG 2: LỰA CHỌN DỊCH VỤ WEB TRONG KIẾN TRÚC HƯỚNG DỊCH VỤ DỰA TRÊN MÔ HÌNH ĐỒ THỊ.....</b>	<b>29</b>
<b>2.1. Bài toán lựa chọn dịch vụ web .....</b>	<b>29</b>
<b>2.2. Mô hình toán học bài toán lựa chọn dịch vụ web.....</b>	<b>31</b>
<b>2.3. Một số thuật toán lựa chọn dịch vụ web dựa trên mô hình đồ thị .....</b>	<b>32</b>

2.3.1. Thuật toán BCOV .....	33
2.3.2. Giải thuật Dijkstra cho bài toán đường đi ngắn nhất .....	37
2.4. Lựa chọn dịch vụ web trong hệ thống hướng dịch vụ .....	40
2.4.1. Khách hàng đăng nhập lấy thông tin.....	42
2.4.2. Khách hàng tìm kiếm khách sạn .....	43
2.4.3. Đặt phòng.....	44
<b>CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ. ....</b>	<b>46</b>
3.1. Khảo sát và phân tích hệ thống.....	46
3.1.1. Khảo sát .....	46
3.1.2. Phân tích hệ thống .....	47
3.2. Thiết kế và triển khai hệ thống .....	54
3.2.1. Chức năng đăng nhập.....	54
3.2.2. Chức năng tìm chuyến bay .....	54
3.2.3. Chức năng tìm kiếm khách sạn .....	55
3.2.4. Chức năng đặt vé chuyến bay.....	56
3.2.5. Chức năng tìm kiếm nhanh .....	56
3.3. Đánh giá.....	57
<b>CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TIẾP THEO ....</b>	<b>59</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>61</b>

## ĐẶT VẤN ĐỀ

Kiến trúc hướng dịch vụ SOA là một mô hình kiến trúc mới trong xây dựng các hệ thống phân tán, hiện đang được sử dụng rộng rãi trong các hệ thống phân tán hiện đại. Bản chất của SOA là việc xây dựng hệ thống mới bằng cách kết nối các dịch vụ web hoặc các hệ thống có sẵn. Cách tiếp cận này cho phép doanh nghiệp phân định được rạch ròi tầng ứng dụng, tầng nghiệp vụ, tầng dịch vụ và tầng cơ sở hạ tầng CNTT, tập trung vào phát triển các dịch vụ cốt lõi của mình. Ngoài ra còn cho phép kế thừa các dịch vụ và hệ thống có sẵn, do đó tiết kiệm thời gian, chi phí, giảm rủi ro trong xây dựng hệ thống mới.

Tuy nhiên, với cách tiếp cận sử dụng SOA và dịch vụ web, việc lựa chọn được các dịch vụ web phù hợp là một trong những nhiệm vụ quan trọng cần giải quyết. Với mỗi nhu cầu sử dụng một dịch vụ web có thể có nhiều dịch vụ web có thể thỏa mãn (thực hiện cùng một chức năng), chúng có sự khác nhau về các tiêu chí QoS (Quality of Service), từ đó đặt ra bài toán lựa chọn các dịch vụ web, bản chất là lựa chọn các dịch vụ web thích hợp để tối ưu hóa hàm mục tiêu đề ra. Lựa chọn tập hợp các dịch vụ web phù hợp ảnh hưởng rất lớn đến hiệu suất và chi phí của các hệ thống SOA. Tuy nhiên, mặc dù đã thu hút được nhiều nghiên cứu trong những năm trở lại đây, các phương pháp và thuật toán lựa chọn dịch vụ web được đề xuất vẫn chưa thực sự hiệu quả và cần phải cải tiến để tối ưu hơn nữa quá trình khai thác và sử dụng dịch vụ web.

Trên cơ sở đó, học viên đã chọn đề tài : **“Nghiên cứu lựa chọn dịch vụ web ứng dụng trong xây dựng các hệ thống hướng dịch vụ dựa trên mô hình đồ thị”** Đề tài gồm 4 chương:

Chương 1: Tổng quan về dịch vụ web, kiến trúc hướng dịch vụ SOA và mô hình đồ thị.

Chương 2: Lựa chọn dịch vụ web dựa trên mô hình đồ thị.

Chương 3: Cài đặt thử nghiệm và đánh giá kết quả.

Chương 4: Kết luận và hướng phát triển tiếp theo.

Trong quá trình hoàn thiện làm luận văn, học viên cũng đã cố gắng rất nhiều, nhưng do kiến thức và kinh nghiệm còn hạn hẹp, khó tránh khỏi những sai sót, rất mong nhận được lời khuyên của các thầy cô để luận văn có thể hoàn thiện hơn.

Tôi xin chân thành cảm ơn.

*Hà Nội, ngày 30 tháng 09 năm 2022*

**Học viên thực hiện**

**NGÔ VĂN TRUNG**



## DANH MỤC CÁC KÝ HIỆU

<b>Ký hiệu</b>	<b>Từ Tiếng Anh</b>	<b>Giải thích</b>
WS	Web Service	Dịch vụ web
XML	Extensible Markup Language	Ngôn ngữ Đánh dấu Mở rộng
SOAP	Simple Object Access Protocol	Phương thức truy cập đối tượng đơn giản
WSDL	Web Services Description Language	Ngôn ngữ mô tả dịch vụ web
UDDI	Universal Description, Discovery and Intergration	Một tập các quy tắc đăng ký và tìm kiếm thông tin các Web Service
W3C	World Wide Web Consortium	Chuẩn W3C
SOA	Service-oriented architecture	Kiến trúc hướng dịch vụ
CORBA	Common Object Request Broker Architecture	Kiến trúc môi giới các đối tượng
DCOM	Distributed Component Object Model	Mô hình đối tượng thành phần phân tán
CSDL	Database	Cơ sở dữ liệu
SQL	Structured Query Language	Ngôn ngữ truy vấn

## MỞ ĐẦU

Dịch vụ Web (Web Service) trở nên quen thuộc với người dùng hiện nay trong xu thế phổ biến của các dịch vụ CNTT trên môi trường Internet. Thay vì kiến trúc đơn giản, khá cứng nhắc là client – server, kiến trúc Web rất linh hoạt, phù hợp với khả năng giãn nở (scalability) khi các dịch vụ được nâng cấp về tính năng, mở rộng tái người dùng và đặc biệt là sử dụng phổ biến của công nghệ Cloud. Web Service là các module phần mềm kết nối với các hệ thống khác nhau thông qua các giao diện chuẩn và giao thức mạng TCP/IP. Dịch vụ web rất quan trọng trong việc ứng dụng xây dựng hệ thống hướng dịch vụ ngày càng được thể hiện quan trọng trong quá trình phát triển dịch vụ Công nghệ thông tin hiện nay. Dịch vụ web bao gồm: nhà cung cấp dịch vụ web (Service Provider), người sử dụng dịch vụ web (Service Requester) và người môi giới dịch vụ web (Service Broker).

Nhà cung cấp dịch vụ sẽ phát triển các dịch vụ web của mình, mô tả chúng thông qua ngôn ngữ đặc tả WSDL (Web Services Description Language) và public chúng tới nhà môi giới. Các dịch vụ web ngoài mô tả chức năng thực hiện còn có các thuộc tính chất lượng (QoS – Quality of Service) như giá (Price), thời gian trả lời (Response Time), tính tin cậy (Reliability), tính sẵn sàng (Availability)... Người có nhu cầu sử dụng dịch vụ web sẽ tìm kiếm dịch vụ web cần thiết thông qua người môi giới và lựa chọn dịch vụ web phù hợp với tiêu chí của mình. Sau khi tìm được dịch vụ web cần thiết từ nhà cung cấp dịch vụ, người sử dụng dịch vụ web sẽ kết nối trực tiếp tới nhà cung cấp thông qua giao thức SOAP (Simple Object Access Protocol).

Kiến trúc hướng dịch vụ SOA là một mô hình kiến trúc mới trong xây dựng các hệ thống phân tán, hiện đang được sử dụng rộng rãi trong các hệ thống phân tán hiện đại. Bản chất của SOA là việc xây dựng hệ thống mới bằng cách kết nối các dịch vụ web hoặc các hệ thống có sẵn. Cách tiếp cận này cho phép doanh nghiệp phân định được rạch ròi tầng ứng dụng, tầng nghiệp vụ, tầng dịch

vụ và tầng cơ sở hạ tầng CNTT, tập trung vào phát triển các dịch vụ cốt lõi của mình. Ngoài ra còn cho phép kế thừa các dịch vụ và hệ thống có sẵn, do đó tiết kiệm thời gian, chi phí, giảm rủi ro trong xây dựng hệ thống mới.

Tuy nhiên, với cách tiếp cận sử dụng SOA và dịch vụ web, việc lựa chọn được các dịch vụ web phù hợp là một trong những nhiệm vụ quan trọng cần giải quyết. Với mỗi một nhu cầu sử dụng một dịch vụ web có thể có nhiều dịch vụ web được tìm thấy (thực hiện cùng một chức năng), chúng có sự khác nhau về các tiêu chí QoS, từ đó đặt ra bài toán lựa chọn các dịch vụ web để tối ưu hóa hàm mục tiêu đề ra. Lựa chọn tập hợp các dịch vụ web phù hợp ảnh hưởng rất lớn đến hiệu suất và chi phí của các hệ thống SOA.

Từ đó có thể thấy, việc nghiên cứu các thuật toán lựa chọn các dịch vụ web và các công nghệ liên quan, ứng dụng để xây dựng các hệ thống hướng dịch vụ SOA là rất cấp thiết và có ý nghĩa khoa học.

Đây là một đề tài rất quan trọng nó có tính cấp thiết trong việc xây dựng các hệ thống hướng dịch vụ và được ứng dụng trong thực tiễn rất cao. Việc nghiên cứu dịch vụ web các công nghệ liên quan, đặc biệt là các thuật toán lựa chọn dịch vụ web, ứng dụng trong xây dựng các hệ thống hướng dịch vụ là một việc làm cần thiết và có tính ứng dụng cao. Các kết quả nghiên cứu của đề tài có thể giúp xây dựng các hệ thống phân tán trên nền tảng dịch vụ web và kiến trúc hướng dịch vụ trong thực tế.

## **CHƯƠNG 1: TỔNG QUAN VỀ DỊCH VỤ WEB, KIẾN TRÚC HƯỚNG DỊCH VỤ SOA VÀ MÔ HÌNH ĐỒ THỊ**

### **1.1. Tổng quan Dịch vụ Web**

Kể từ khi Web service xuất hiện đã mở ra cho các nhà phát triển dịch vụ cũng như người làm trong lĩnh vực công nghệ thông tin có hướng tiếp cận và phát triển mới. Web services bản chất là các dịch vụ web. Do Web services là sự kết hợp của nhiều công nghệ khác nhau nên nó giúp cho các nhà phát triển dịch

vụ nói chung và lập trình viên nói riêng có thể sử dụng nhiều nền tảng, nhiều ngôn ngữ khác nhau để phát triển ứng dụng thông qua mạng Internet. Nó không đòi hỏi các nhà phát triển và lập trình viên phải sử dụng công nghệ mới. Đó cũng là điểm khác biệt của web services so với các nền tảng, công nghệ khác, đó chính là khả năng kết hợp các công nghệ đã có như là XML, SOAP, WSDL, UDDI để tạo ra các service, điều này giúp web services có vai trò rất quan trọng trong việc phát triển các dịch vụ hiện nay.

### 1.1.1. Khái niệm Dịch vụ Web

Theo định nghĩa của W3C (World Wide Web Consortium), Web Service là một hệ thống phần mềm được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau thông qua mạng Internet, giao diện chung và sự gắn kết của nó được mô tả bằng XML.[1]

### 1.1.2. Mô hình sử dụng Dịch vụ Web

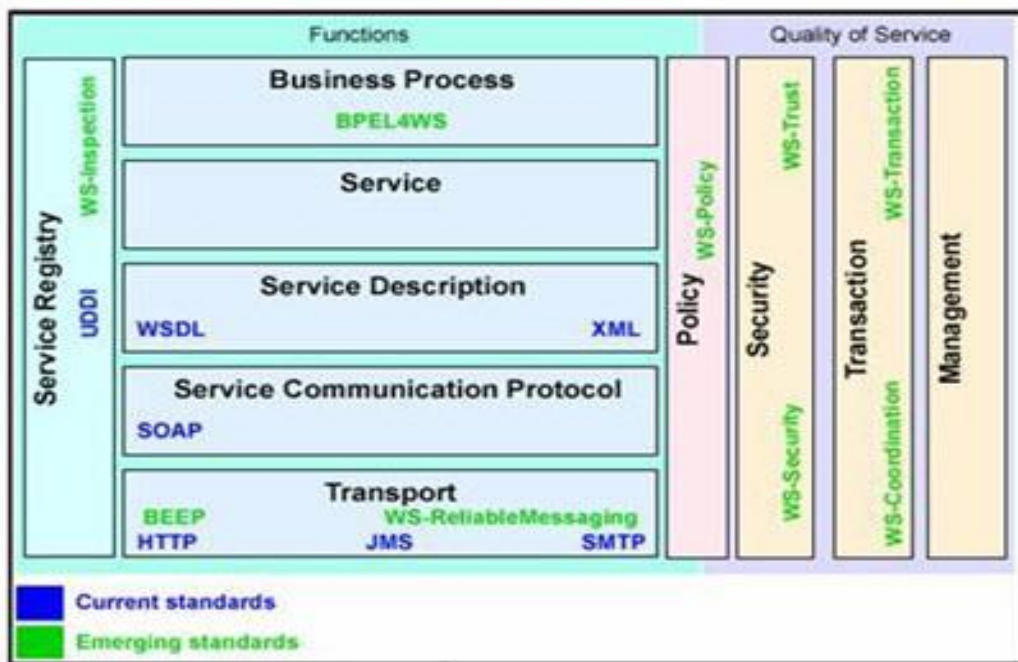


*Hình 1: Kiến trúc của dịch vụ web[1]*

Dựa trên mô hình kiến trúc, dịch vụ web gồm 3 thành phần chính:

- Cung cấp dịch vụ (web service provider)
- Sử dụng dịch vụ (web service consumer)
- Môi giới dịch vụ (web service broker)[1]

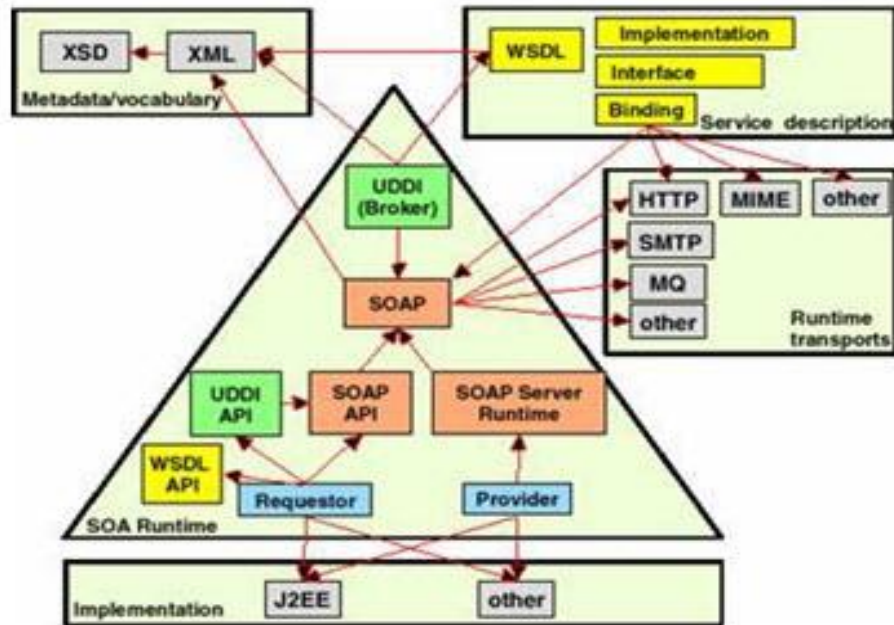
## 1.1.3. Kiến trúc của web service



Hình 2: Mô hình Kiến trúc phân tầng của web service [1]

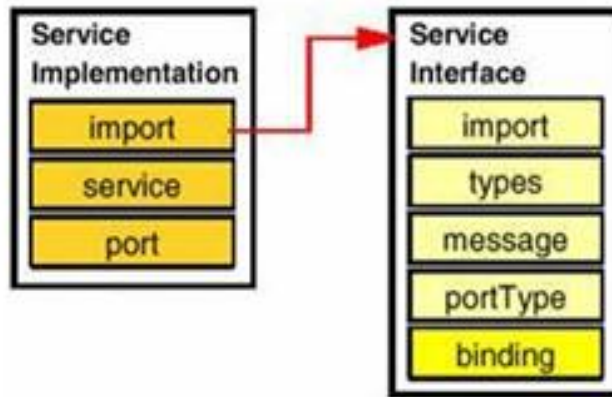
Tầng vận chuyển chính là HTTP (Hypertext Transfer Protocol), SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol), và công nghệ mới nhất là BEEP (Blocks Extensible Exchange Protocol), thực hiện trao đổi thông tin cho các ứng dụng sử dụng mạng.

#### 1.1.4. Các thành phần của web service



Hình 3: Mối quan hệ giữa các thành phần [1]

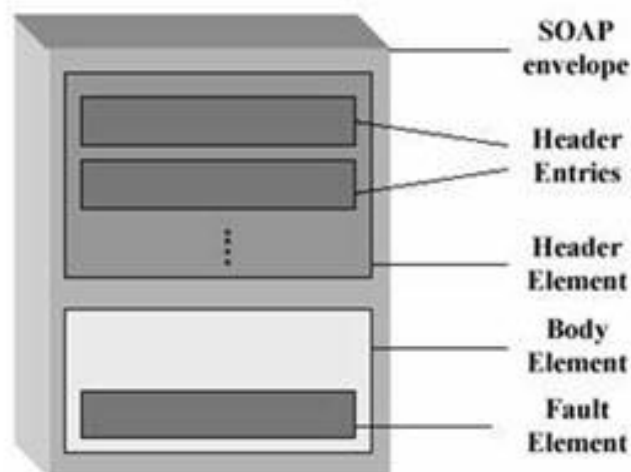
- XML : được gọi là ngôn ngữ mô tả văn bản, các đặc trưng và cấu trúc đều do người sử dụng tự định nghĩa đưa ra mà nó không tuân theo quy ước nào, nó được thể hiện ở các tag XML và là ngôn ngữ chính sử dụng để giao tiếp trong Web service.
- WSDL: là thành phần thể hiện đặc trưng của Web service, được mô tả bằng XML
- WSDL có hai cấu phần: Giao diện, thi hành.



Hình 4: Cấu trúc WSDL [1]

- UDDI : Là thành phần mô tả các tiêu chuẩn kỹ thuật dựa trên Internet.
- SOAP : Là giao thức mô tả giao tiếp có cấu trúc giống như XML và được mã hóa các thành phần chung cho các ứng dụng sử dụng. Hiện nay nó là một thành phần rất quan trọng trong xây dựng các hệ thống phân tán từ nhiều ngôn ngữ , hệ điều hành khác nhau [1].

a. Cấu trúc message dạng SOAP



Hình 5: Cấu trúc message của SOAP [1]

b. Truyền thông

SOAP hỗ trợ hai kiểu truyền thông khác nhau:

- Remote procedure call (RPC)
- Document

c. Mô hình dữ liệu

d. Mã hóa

### 1.1.5. Chất lượng Dịch vụ Web

Mục đích của quá trình tích hợp và lựa chọn dịch vụ web là tìm được tập các dịch vụ web thỏa mãn được các yêu cầu chức năng đặt ra và đảm bảo các tiêu chí chất lượng QoS.

Chất lượng dịch vụ Web là thành phần rất quan trọng cho các nhà cung cấp dịch vụ web nói chung và các nhà phát triển công nghệ thông tin nói riêng. QoS sẽ quyết định đến tính hữu ích, sử dụng của dịch vụ [3] [4], QoS gồm các thành phần

#### 1.1.5.1. Chi phí

Chi phí chất lượng  $C_{ij}$  là số tiền mà bên yêu cầu dịch vụ cần phải trả để sử dụng dịch vụ  $i$  cho việc thực hiện công việc  $j$ .

$$C_{ij}, i \in [1...n], j \in [1...m]$$

#### 1.1.5.2. Thời gian

Thời gian  $t_{ij}$  là thời gian thực hiện của dịch vụ web  $i$  khi xử lý công việc  $j$ , được tính từ thời điểm yêu cầu được gửi tới thời điểm kết quả nhận được:

$$t_{ij}, i \in [1...n], j [1...m]$$

#### 1.1.5.3. Tính sẵn có

Chất lượng sẵn có  $a_{ij}$  là xác suất mà các dịch vụ có thể được truy cập và sử dụng. Đây là tỷ số giữa số lần các dịch vụ đáp ứng yêu cầu và số lượng tổng yêu cầu thực hiện cho các dịch vụ:



$$a_{ij} = \frac{req_{ij}}{tot_{ij}}, \quad tot_{ij} \neq 0, \quad i \in [1...n], \quad j \in [1...m] \quad (1.1)$$

$req_{ij}$  là số lượng yêu cầu thành công để dịch vụ  $i$  sử dụng công việc  $j$ , và  $tot_{ij}$  là tổng số các lần gọi.

#### 1.1.5.4. Danh tiếng

Danh tiếng  $R_{ij}$  là thước đo của sự tin cậy của một định vụ  $i$  cho công việc  $j$ . Nó phụ thuộc vào kinh nghiệm của người dùng bằng cách sử dụng dịch vụ. Người dùng cuối khác nhau có thể có ý kiến khác nhau về cùng một dịch vụ.

Danh tiếng có thể được định nghĩa là việc xếp hạng trung bình cho các dịch vụ của người dùng cuối. Danh tiếng của một dịch vụ nhất định thường được định nghĩa là:

$$q_{rep} = \frac{\sum_{b=1}^N k_b}{N} \quad (1.2)$$

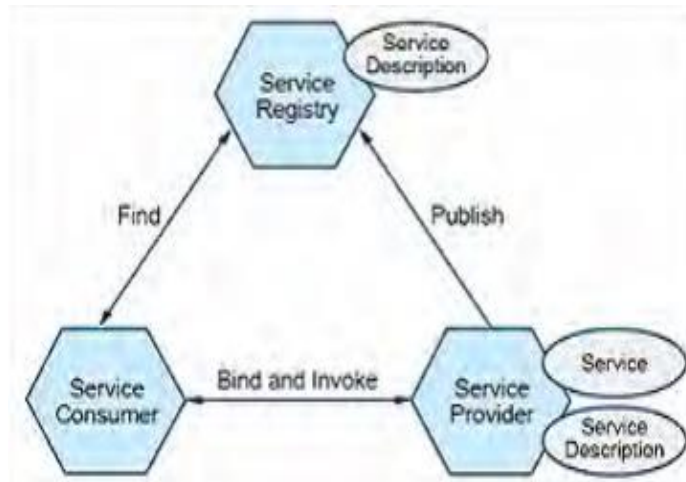
ở đó  $k_b$  là xếp hạng  $b^{th}$  trao cho các dịch vụ và  $N$  là số lần các dịch vụ đã được xếp hạng.

## 1.2. Kiến trúc hướng dịch vụ SOA

### 1.2.1. Khái niệm SOA

Kiến trúc hướng dịch vụ (SOA) là việc thiết kế, xây dựng các phần mềm, chức năng, theo các thành phần, mỗi thành phần là một “dịch vụ có tính kết nối mềm dẻo”, và được truy cập qua mạng. [2]

SOA có ba thành phần :



Hình 6: Sơ đồ cộng tác [7]

Nhà cung cấp (service provider) dịch vụ , lưu trữ thông tin dịch vụ (service registry). Người sử dụng (service consumer).

SOA là giải pháp giải quyết được các hệ thống: phức tạp, không linh hoạt và không ổn định. Triển khai theo mô hình SOA dễ mở rộng, liên kết tốt. Là cơ sở cho việc lựa chọn, sử dụng lại thành phần hiện có.

## 1.2.2. Nguyên tắc của SOA

### 1.2.2.1. Phân định rạch ròi các dịch vụ

Dịch vụ tương tác với nhau qua thành phần giao tiếp. Thành phần giao tiếp sẽ qui định những định dạng giao tiếp: thông điệp nào sẽ được chấp nhận và không được xử lý. Do đó đối tượng bên ngoài có thể truy cập thông tin tới hệ thống.

### 1.2.2.2. Dịch vụ hoạt động

Dịch vụ được triển khai và hoạt động độc lập không phụ thuộc vào một dịch vụ khác, có tính bền vững cao. Các dịch vụ có thể cộng tác lẫn nhau trong trường hợp có dịch vụ khác bị hỏng và chúng có sự an toàn, bảo mật..., thể hiện tính linh hoạt , mềm dẻo trong hệ thống.

### 1.2.2.3. Dịch vụ chia sẻ

Dịch vụ cung cấp thành phần giao tiếp, chia sẻ cấu trúc thông tin, ràng buộc dữ liệu (schema) ra bên ngoài. Thể hiện được hệ thống có tính liên kết và mở rộng.

#### 1.2.2.4. Tương thích với chính sách

Khi dịch vụ này muốn tương tác với dịch vụ khác thì chúng phải đảm bảo chính sách, yêu cầu dịch vụ như: mã hóa, bảo mật...Do vậy, mỗi dịch vụ cần cung cấp công khai yêu cầu và chính sách.

#### 1.2.3. Tính chất SOA

##### 1.2.3.1. Kết nối mềm dẻo

SOA có hai loại kết nối: rời (loose) và chặt (tight). Các thành phần của chúng có sự ràng buộc khác nhau. Hiện nay việc xây dựng các phần mềm đều hướng đến tính liên kết mềm dẻo giữa các thành phần với nhau trong hệ thống. .

##### 1.2.3.2. Tái sử dụng

Các dịch vụ được tái sử dụng lại đó là sự kết hợp lại với nhau theo nhiều mục đích khác nhau. Tái sử dụng lại các dịch vụ loại bỏ những thành phần trùng lặp, làm đơn giản hoá việc vận hành và quản trị hệ thống.

##### 1.2.3.3. Bất đồng bộ

Khi sử dụng kết hợp dịch vụ thì các yêu cầu dịch vụ được đưa vào hàng đợi và xử lý với tốc độ tối ưu. Bên gọi không phải chờ yêu cầu xử lý xong và trả về do đó hệ thống không bị ảnh hưởng việc xử lý trễ và lỗi khi thực thi các dịch vụ bất đồng bộ.

##### 1.2.3.4. Chính sách

Các dịch vụ có một luật kết hợp riêng gọi là các chính sách. Các chính sách được áp dụng cho mỗi dịch vụ khi thiết kế và trong thời gian thực thi. Dẫn đến làm tăng tính tái sử dụng của các dịch vụ. Giúp các nhà phát triển phần mềm

hỗ trợ làm việc với nhau trong suốt thời gian phát triển để cài đặt và kiểm tra những chính sách.

#### 1.2.3.5. Cộng tác

Là khả năng giúp các hệ thống giao tiếp với nhau bằng nhiều nền tảng và ngôn ngữ khác nhau. Các dịch vụ được gọi thông qua các dạng kết nối. Các kết nối gồm các giao thức và định dạng dữ liệu.

#### 1.2.3.6. Dò tìm và giàng buộc

Khi người dùng thực hiện truy vấn và cần đến một dịch vụ dựa trên một số tiêu chuẩn. Người sử dụng chỉ việc hỏi về dịch vụ nào thoả yêu cầu tìm kiếm trên giao diện và hệ thống sẽ đăng ký gửi xuống dữ liệu và dò tìm thông tin thông qua giàng buộc đăng ký, sau đó hệ thống sẽ truy xuất thông tin và trả ra cho người dùng.

#### 1.2.3.7. Hồi phục

Đối với các hệ thống phân tán thì hồi phục là khi hệ thống gặp lỗi, thì hệ thống có khả năng tự hồi phục mà không có sự tác động của con người.

Chúng ta không lường trước được khả năng xảy ra hoạt động, không hoạt động của hệ thống như ứng dụng phức tạp, nhiều thành phần. Khi đó độ tin cậy của hệ thống phụ thuộc rất nhiều vào khả năng phục hồi của phần cứng sau khi bị lỗi, hạ tầng mạng, kiến trúc cũng như tải của hệ thống,...,vv.

### 1.2.4. Ưu và nhược điểm của SOA.

#### 1.2.4.1. Ưu điểm của SOA.

##### a. Tái sử dụng

Các công ty, doanh nghiệp tận dụng rất tốt trong việc sử dụng SOA: khi áp dụng SOA nó giúp doanh nghiệp tận dụng được những tài nguyên sẵn có; nó làm giảm chi phí cho phần kiến trúc, phát triển và giảm chi phí mua phần mềm

mới. giúp các nhà phát triển tiết kiệm được thời gian viết chương trình .Lợi ích của việc sử dụng lại các thành phần dịch vụ trên hệ thống mang lại :

- Giảm tính dư thừa.
- Kế thừa

b. Tính linh hoạt, mềm dẻo.

- Giúp các nhà phát triển kế thừa sử dụng tài nguyên sẵn có trong việc phát triển các dịch vụ mới.
- Tạo tính kết nối tương tác giữa các dịch vụ được linh hoạt hơn.
- Giúp các nhà phát triển tăng khả năng triển khai.

d. Tương lai.

Hỗ trợ nhà phát triển dịch vụ có quy trình nghiệp vụ linh hoạt đáp ứng theo yêu cầu khách hàng.

e. Thiết bị và nền tảng.

Kiến trúc hướng dịch vụ cung cấp một tầng giao tiếp đặc biệt rất linh hoạt và mềm dẻo thông qua các Service. Tương tích với tất cả các thiết bị đầu cuối như : điện thoại di động, tablet, laptop, máy tính , trình duyệt...vv,. Giúp cho các công ty phát triển được kiến trúc đa nền tảng, tiết kiệm chi phí.

f. Mở rộng và cung cấp.

Như đã được đề cập ở trên do kiến trúc hướng dịch vụ hoạt động độc lập các thành phần, do vậy ta có thể tăng khả năng mở rộng các thành phần bằng cách thêm các thể hiện cho dịch vụ.

#### 1.2.4.2. Nhược điểm của SOA

a. Tăng Overhead

Mỗi khi một dịch vụ tương tác với các dịch vụ khác, xác nhận hoàn toàn mọi tham số đầu vào diễn ra. Điều này làm tăng thời gian phản ứng và tải máy,

và do đó làm giảm hiệu suất tổng thể.

#### b. Quản lý dịch vụ phức tạp

Dịch vụ này cần phải đảm bảo rằng thông điệp đã được cung cấp một cách kịp thời. Nhưng như các dịch vụ giữ trao đổi thông điệp để thực hiện nhiệm vụ, số lượng những tin nhắn này có thể đi vào hàng triệu thậm chí đối với một ứng dụng duy nhất. Điều này đặt ra một thách thức lớn đối với quản lý như một số dân rất lớn của dịch vụ.

#### c. Chi phí đầu tư cao

Thực hiện các SOA đòi hỏi một sự đầu tư trả trước lớn bằng các phương tiện công nghệ, phát triển, và nguồn nhân lực.

#### d. SOA không được khuyến cáo cho các loại ứng dụng sau đây

Đồng nhất: Thực hiện SOA cho các ứng dụng sử dụng công nghệ của một nhà cung cấp đơn lẻ sẽ không hiệu quả. Ví dụ, nếu một ứng dụng được xây dựng trong Java, sau đó nó sẽ được tốt hơn để sử dụng phương pháp của Java hơn là sử dụng HTTP để truyền thông giữa các thành phần.

GUI-Based: SOA sẽ không phù hợp cho các ứng dụng với các chức năng giao diện, ví dụ một ứng dụng bản đồ thao tác. Các ứng dụng như vậy đòi hỏi năng trao đổi dữ liệu, do đó sẽ làm tăng sự phức tạp của các ứng dụng nếu SOA được sử dụng.

Real-time: SOA không phải là mong muốn được sử dụng với thời gian đáp ứng nghiêm ngặt được thực thi kể từ khi dịch vụ giao tiếp không đồng bộ.

Stand-alone: Nó sẽ là vô ích để đầu tư vào SOA cho các ứng dụng không được phân phối độc lập, mà không đòi hỏi yêu cầu và phản ứng dựa trên các cuộc gọi.

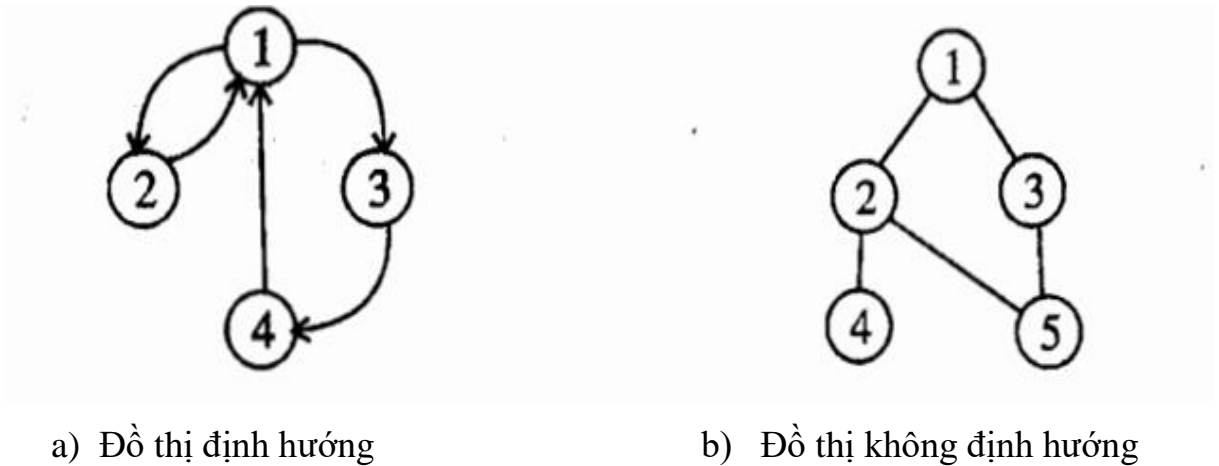
### 1.3. Mô hình đồ thị

### 1.3.1. Một số khái niệm về đồ thị

#### 1.3.1.1. Đồ thị

Một đồ thị  $G(V,E)$  bao gồm một tập hợp hữu hạn  $V$  các nút, hay đỉnh và một tập hợp hữu hạn  $E$  các cặp đỉnh mà ta gọi là cung.

Nếu  $(v_1, v_2)$  là cặp đỉnh thuộc  $E$  thì ta nói : có một cung nối  $v_1$  và  $v_2$ . Nếu cùng  $(v_1, v_2)$  khác với cung  $(v_2, v_1)$  thì ta có một đồ thị định hướng. Lúc đó  $(v_1, v_2)$  được gọi là cung định hướng  $v_1, v_2$ . Nếu thứ tự các nút trên cung không được coi trọng thì ta gọi đồ thị không hướng.



Hình 7: đồ thị

#### 1.3.1.2. Đồ thị định hướng dịch vụ web

Gọi  $G$  là một Đồ thị có hướng được xác định bởi  $(S, E)$  trong đó,  $S$  là tập hợp các dịch vụ web được xác định bởi các đầu vào và đầu ra của nó, và  $E$  là tập các cạnh được định hướng, chẳng hạn như một cạnh liên tục được ký hiệu là  $\overrightarrow{(S_1, S_2)}$  có nghĩa là một dịch vụ  $S_1$  có thể được cấu tạo với  $S_2$  liên quan đến các đầu vào và đầu ra.

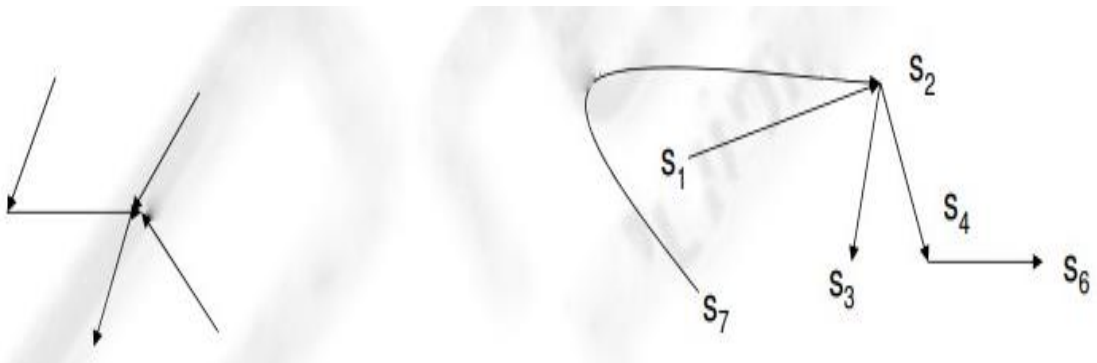
#### 1.3.1.3. Thành phần tuyến tính

Gọi  $G$  là một đồ thị có hướng được xác định bởi  $(S, E)$ , thành phần được gọi là tuyến tính nếu mỗi nút chỉ có một cạnh đi ra,  $\forall S_i, S_j \in S, \exists e \in E \mid e =$

$(\overrightarrow{S_i, S_j})$ ,  $j \geq 1$  và  $|e| = 1$ , trong đó  $|e|$  nghĩa là bản số của các cạnh giữa  $S_i$ ,  $S_j$ . Ví dụ trong Hình 8 (a), tất cả các dịch vụ chỉ có một cạnh gửi đi.

#### 1.3.1.4. Thành phần phi tuyến tính

Gọi  $G$  là một đồ thị có hướng được xác định bởi  $(S, E)$ , thành phần được gọi là phi tuyến tính nếu với mỗi nút có nhiều hơn một cạnh đi  $\forall S_i, S_j \in S, \exists e \in E | e = (\overrightarrow{S_i, S_j}), |e| > 1$ . Ví dụ trong Hình 8 (b), dịch vụ  $S_2$  có hai cạnh đi ra



(a) Thành phần tuyến tính

(b) Thành phần không tuyến tính

Hình 8: Ví dụ về biểu đồ thành phần dịch vụ web.[5]

#### 1.3.2. Biểu diễn đồ thị

##### 1.3.2.1 Ma trận liên kề hay ma trận kề

Ma trận kề là một mảng 2 chiều gồm các đỉnh  $V \times V$ . Mỗi hàng và cột đại diện cho một đỉnh. Nếu giá trị của bất kỳ phần tử nào  $a[i][j]$  là 1, nó biểu thị rằng có một cạnh nối đỉnh  $i$  và đỉnh  $j$ .

##### 1.3.2.2 Danh sách kề.

Danh sách kề biểu thị một biểu đồ dưới dạng một mảng các danh sách được liên kết. Chỉ số của mảng đại diện cho một đỉnh và mỗi phần tử trong danh sách liên kết của nó đại diện cho các đỉnh khác tạo thành một cạnh với đỉnh.



### 1.3.3. Ứng dụng của đồ thị

- Trong khoa học máy tính, đồ thị được sử dụng để biểu diễn luồng tính toán.
- Bản đồ của Google sử dụng đồ thị để xây dựng hệ thống định vị GPS, giao thông, trong đó giao điểm của hai (hoặc nhiều) đường được coi là một đỉnh và đường nối hai đỉnh được coi là một cạnh, do đó hệ thống điều hướng của chúng dựa trên thuật toán để tính toán ngắn nhất đường đi giữa hai đỉnh.
- Trong mạng xã hội Facebook, người dùng được coi là đỉnh và nếu họ là bạn bè thì sẽ có một cạnh chạy giữa họ. Thuật toán đề xuất bạn bè của Facebook sử dụng lý thuyết đồ thị. Facebook là một ví dụ về đồ thị vô hướng.
- Các trang Web được coi là các đỉnh. Có một cạnh từ trang u sang trang khác v nếu có một liên kết của trang v trên trang u. Đây là một ví dụ về đồ thị có hướng. Đó là ý tưởng cơ bản đằng sau thuật toán xếp hạng trang của Google.
- Trong hệ điều hành, chúng ta bắt gặp đồ thị phân bổ tài nguyên, mà trong đó, mỗi tiến trình và tài nguyên được coi là các đỉnh. Các cạnh được rút ra từ các tài nguyên tới tiến trình được cấp phát hoặc từ tiến trình yêu cầu đến tài nguyên được yêu cầu. Nếu điều này dẫn đến sự hình thành của một chu trình thì sẽ xảy ra bế tắc.

## 1.4. Một số khái niệm tương quan đồ thị và dịch vụ web.

### 1.4.1. Đồ thị có hướng cho các dịch vụ web

Giả sử một nhà cung cấp dịch vụ web duy trì một biểu đồ định hướng được gọi là biểu đồ toàn cục trong đó mỗi đỉnh đại diện cho một dịch vụ và một cạnh đi có nghĩa là một thành phần có thể có giữa hai dịch vụ do nhà cung cấp xuất bản (đầu ra của dịch vụ với đầu vào của một số dịch vụ). Ý tưởng đằng sau

cách tiếp cận thứ hai là, xem xét chuyển đổi tối thiểu với chi phí tốt nhất (không sử dụng tất cả các chuyển đổi tối thiểu).

#### 1.4.2 Truy vấn đối tượng có định hướng.

Một truy vấn  $Q$  được gọi là có định hướng và được ký hiệu là  $\vec{Q}$  nếu các nguyên tử tạo thành  $Q$  được sắp xếp theo thứ tự.

Chúng ta hãy xem xét truy vấn sau đây. "Tôi muốn đi từ Paris đến Barcelona vào ngày 3 tháng 3, tìm một chỗ ở ở đó trong một tuần (từ ngày 3 đến ngày 10 tháng 3) và thuê một chiếc xe hơi". Chúng tôi có thể tìm thấy một chuyến bay hoặc dịch vụ đường sắt và dịch vụ khách sạn, v.v. Tuy nhiên, chúng tôi không thể chấp nhận những gì khách sạn cung cấp dịch vụ cho mình trước khi tìm đến dịch vụ chuyến bay. Vì vậy, các dịch vụ được khám phá có thể tuân theo một thứ tự để thực hiện truy vấn.

#### 1.4.2. Tỷ lệ bao phủ

Tỷ lệ bao phủ được tính bằng tỷ lệ giữa số lượng dịch vụ web được tạo ra trong thành phần trên tổng số dịch vụ được cung cấp trong tập hợp bao gồm tốt nhất:  $CR = \frac{|\text{compos-web-services}|}{|\text{tổng số dịch vụ}|}$ , ở đây  $||$  có nghĩa là bản số.

## **CHƯƠNG 2: LỰA CHỌN DỊCH VỤ WEB TRONG KIẾN TRÚC HƯỚNG DỊCH VỤ DỰA TRÊN MÔ HÌNH ĐỒ THỊ**

Mô hình SOA có rất nhiều ưu điểm, tuy nhiên tồn tại một bài toán rất quan trọng trong SOA mà vẫn chưa được giải quyết triệt để, đó là bài toán lựa chọn dịch vụ web tối ưu.

Do thời gian nghiên cứu hạn hẹp, ở chương này tôi xin tập chung vào trình bày các nội dung chính sau :

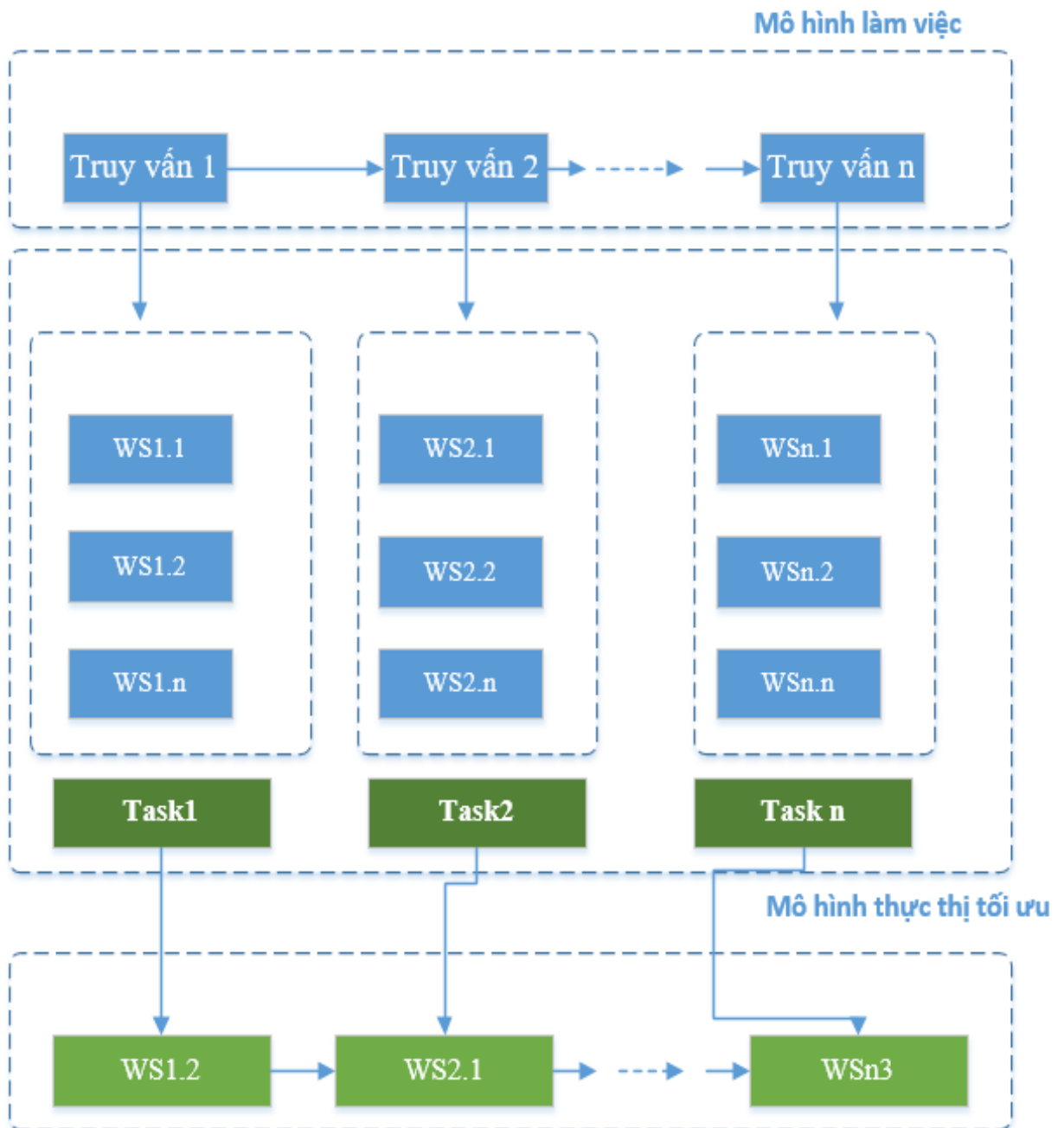
- Bài toán lựa chọn dịch vụ web.
- Mô hình toán học của bài toán lựa chọn dịch vụ web.
- Thuật toán Bcov, thuật toán Dijkstra áp dụng cho bài toán lựa chọn dịch vụ web.
- Lựa chọn dịch vụ web trong hệ thống hướng dịch vụ

### **2.1. Bài toán lựa chọn dịch vụ web**

Dịch vụ web đóng một vai trò quan trọng trong việc lựa chọn dịch vụ web. Trên thực tế, trong cuộc sống thực, việc phát hiện ra các dịch vụ web phù hợp với truy vấn của người dùng là điều không dễ dàng. Vì vậy, việc lựa chọn các dịch vụ web đáp ứng truy vấn là cần thiết.

Lựa chọn các dịch vụ đã có trên hệ thống là một phần của SOA. Các hệ thống hiện nay việc lựa chọn dịch vụ web đang là một giải pháp cho việc sử dụng lại các dịch vụ đã có để tạo một dịch vụ mới tốt hơn.

Lựa chọn dịch vụ web là quá trình lựa chọn các dịch vụ đã tồn tại (dịch vụ web thành phần) để tạo thành một dịch vụ “web phức hợp” hoặc phối hợp các dịch vụ web để thực hiện một công việc phức tạp hơn. Các dịch vụ web thành phần còn được gọi là dịch vụ web đơn hay dịch vụ web nguyên tử.



*Hình 9: Mạng dịch vụ web.*

**Một số ví dụ của bài toán lựa chọn dịch vụ web:**

**Bài toán 1:** Cho một truy vấn của người dùng đang tìm kiếm người dịch dịch vụ cung cấp văn bản dịch từ tiếng Pháp sang tiếng Thái, đầu vào truy vấn là tiếng

Pháp và đầu ra là tiếng Thái. Khi đó nhà cung cấp dịch vụ web mô tả một tập hợp các dịch vụ sau:

1. Bản dịch từ tiếng Pháp sang tiếng Anh (SF –E)
2. Bản dịch từ tiếng Anh sang tiếng Đức (SE-G)
3. Dịch từ tiếng Anh sang tiếng Trung (SE-C)
4. Dịch từ tiếng Trung sang tiếng Thái ... (SC-T)

(SE – C), (SC – T)}. Thành phần này được gọi là tuyến tính. Thành phần này được thực hiện thông qua sự phù hợp giữa đầu ra và đầu vào của dịch vụ

**Bài toán 2:** Xét truy vấn khách hàng tìm kiếm một chỗ ở du lịch. Khi đó nhà cung cấp dịch vụ web, cung cấp các dịch vụ web sau:

- Câu lạc bộ du lịch có tên khách hàng và thẻ tín dụng làm đầu vào cung cấp dịch vụ đặt vé máy bay và đặt chỗ ở.
- Câu lạc bộ giải trí, có đặt chỗ ở làm đầu vào cung cấp dịch vụ bơi lội-giải trí.
- Cho thuê ô tô, có dịch vụ đặt vé máy bay làm đầu vào (thuê ô tô, thuê thuyền).

Nhà cung cấp không thể tìm thấy các dịch vụ đáp ứng các ràng buộc của người yêu cầu, vì vậy câu trả lời là một thành phần của các dịch vụ web sau: Câu lạc bộ du lịch + (Câu lạc bộ giải trí + cho thuê xe hơi), thành phần này được gọi là phi tuyến tính

## 2.2. Mô hình toán học bài toán lựa chọn dịch vụ web

Khi người dùng thực hiện gửi truy vấn, trong đó yêu cầu thực thi đồng thời  $N$  dịch vụ web  $\{S_1, S_2, \dots, S_N\}$ . Đối với mỗi dịch vụ web  $S_k$ ,  $k = 1, 2, \dots, N$ ,

xây dựng tương ứng một mạng  $G_k$  từ các dịch vụ web. Từ  $N$  mạng riêng biệt này tiến hành xây dựng mạng đồ thị. Ký hiệu  $G = (V, A)$ , với  $V$  – Tập hợp các nút mạng,  $A$  – Tập hợp các cung của mạng. Cung nối hai nút mạng  $i$  và  $j$  ký hiệu  $(i, j)$ .

Chúng ta định nghĩa hàm “nguyên tử”  $x_{ij}^k(t)$  cho cung  $(i, j)$  của mạng  $G_k$  xác định trong khoảng thời gian  $[0, T_x]$ , với  $T_x$  là thời điểm hoàn thành tất cả các dịch vụ web  $S_k$  theo công thức sau.

$$x_{ij}^k(t) = \begin{cases} 1, & t \in [T_0^k(i, j), T_1^k(i, j)), \\ 0 & \text{otherwise,} \end{cases} \quad (2.1)$$

$$T_1^k(i, j) - T_0^k(i, j) = T_{\text{exec}}(i, j)$$

$T_0^k(i, j)$ ,  $T_1^k(i, j)$  là thời điểm bắt đầu và kết thúc việc sử dụng dịch vụ web đơn  $(i, j)$ ,  $T_{\text{exec}}(i, j)$  là thời gian thực hiện dịch vụ web đơn  $(i, j)$ .

Các dịch vụ web đơn - tương ứng với các cung của một đường đi đầy đủ, bao gồm  $m$  nút mạng  $i_1, i_2 \dots i_m$  sẽ được xử lý theo trình tự, điều này được thể hiện bằng ràng buộc.

$$T_1^k(i_n, i_{n+1}) \leq T_0^k(i_{n+1}, i_{n+2}) \quad (2.2)$$

với  $n = 1, 2, \dots, m - 2$ .

Để biểu diễn đường đi đầy đủ  $C$  trên mạng  $G_k$  chúng ta sử dụng mô hình bài toán vận tải, vận chuyển 1 đơn vị sản phẩm trong mạng với điều kiện cân bằng trên các nút. Trên mỗi cung ta định nghĩa một biến logic  $u_{ij}^k$ .

$$u_{ij}^k = \begin{cases} 0, & x_{ij}^k(t) = 0, t \in [0, T_x] \\ 1, & \text{otherwise.} \end{cases} \quad (2.3)$$

### 2.3. Một số thuật toán lựa chọn dịch vụ web dựa trên mô hình đồ thị

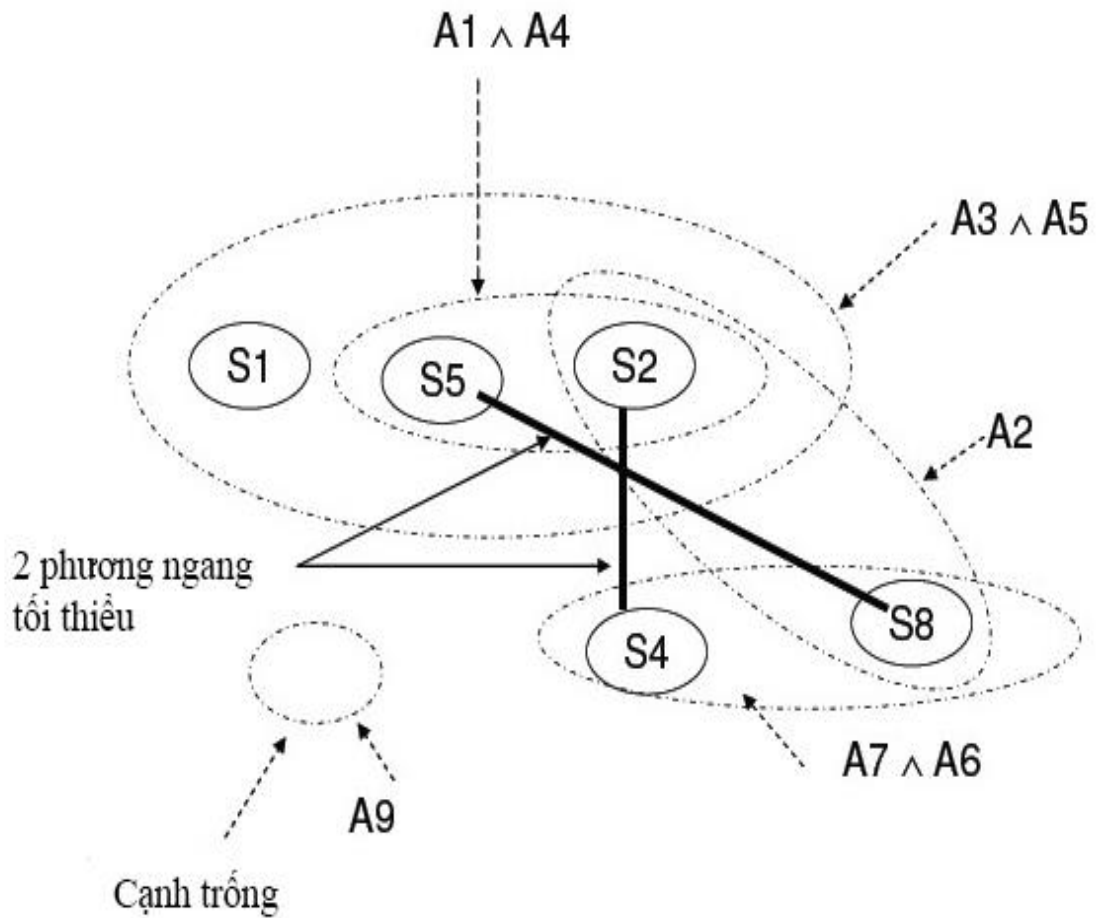
Hiện nay, có rất nhiều giải thuật để xử lý trên đồ thị, nhưng do thời gian nghiên cứu còn hạn hẹp, tôi xin trình bày 2 giải thuật sau :

### 2.3.1. Thuật toán BCOV

Đó là một cách tiếp cận dựa trên siêu đồ thị. Ý tưởng chính cơ bản của thuật toán là:

Cho một truy vấn  $Q$  và một tập hợp các dịch vụ  $S$ , vấn đề khám phá bao gồm việc tìm một tập con của  $S$  được gọi là "bìa" của  $Q$  chứa càng nhiều thông tin chung với  $Q$  càng tốt và càng ít thông tin bổ sung càng tốt liên quan đến  $Q$ . Các dịch vụ và truy vấn được mô tả bằng sự kết hợp của các nguyên tử. Các dịch vụ được phát hiện thông qua việc xây dựng một siêu đồ thị  $(\Sigma, \Gamma)$  trong đó  $\Sigma$  là một tập hợp các đỉnh tương ứng với mô tả các dịch vụ và mỗi liên hợp trong truy vấn trở thành một cạnh của  $\Gamma$  trong siêu đồ thị.[5]

Giả sử có một truy vấn được xác định bởi kết hợp sau  $Q = A1 \wedge A2 \wedge A3 \wedge A4 \wedge A5 \wedge A6 \wedge A7 \wedge A8 \wedge A9$  và một tập hợp các dịch vụ  $S1, S2, S3, \dots, Sn$ . Trong Hình 10 được mô tả siêu đồ thị tương ứng và bìa tốt nhất của truy vấn  $Q$ , được biểu diễn bằng hai phương ngang tối thiểu:  $(S2 \wedge S4), (S5 \wedge S8)$ .



Hình 10: Ví dụ về mô hình siêu đồ thị của các dịch vụ web.[5]

### Nội dung của giải thuật Bcov:

Ý tưởng đằng sau nội dung được đề xuất đưa ra theo các bước sau:

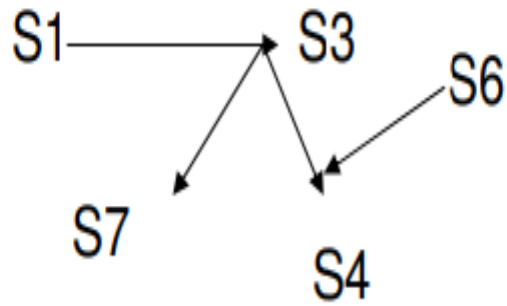
- Đối với mỗi chuyển ngang tối thiểu được cung cấp bởi thuật toán Bcov, chúng tôi xây dựng một đồ thị định hướng trong đó mỗi nút tương ứng với một dịch vụ và cạnh là các mối quan hệ giữa đầu vào và đầu ra của dịch vụ
- Sau khi phát hiện ra các dịch vụ nguyên tử là các nút cô lập trong biểu đồ, chúng tôi sắp xếp các dịch vụ của phép chuyển ngang tối thiểu liên quan đến các nguyên tử truy vấn.



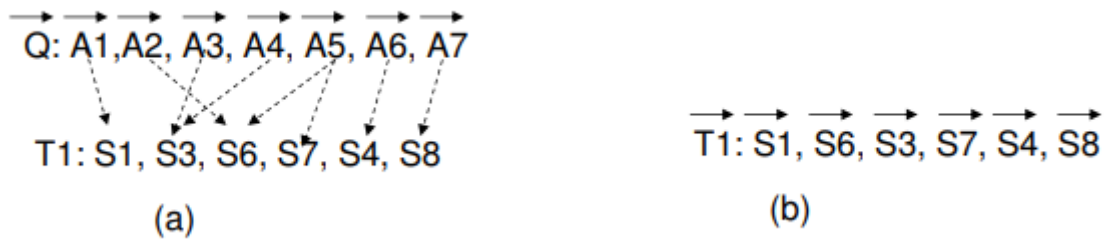
- Các dịch vụ nguyên tử bị loại bỏ khỏi chuyển đổi. Tìm trong biểu đồ định hướng, một đường dẫn (thành phần tuyến tính) hoặc đồ thị con (thành phần không tuyến tính) giữa dịch vụ đầu tiên và dịch vụ cuối cùng (của tập hợp có thứ tự) có tính đến tất cả các dịch vụ trung gian.
- Tính toán chi phí truy xuất bằng cách chỉ xem xét các dịch vụ đã tổng hợp liên quan đến truy vấn. Chi phí được đưa ra bởi tỷ lệ bao phủ.

Giả sử, một truy vấn được mô tả bởi các nguyên tử sau:  $\vec{Q} = \vec{A}_1, \vec{A}_2, \vec{A}_3, \vec{A}_4, \vec{A}_5, \vec{A}_6, \vec{A}_7$  và tập hợp các phương ngang tối thiểu được cung cấp bởi thuật toán Bcov:  $\{T1, T2, T3\}$  trong đó  $T1 = \{S1, S3, S6, S7, S4, S8\}$ ,  $T2 = \{S1, S2, S7, S9, S10, S11\}$ ,  $T3 = \{S1, S2, S12, S13, S20\}$ . Đối với  $T1, T2, T3$ , xây dựng một đồ thị tương ứng trong đó các đầu vào khớp với các đầu ra. Hình 3 đưa ra một ví dụ về đồ thị định hướng liên quan đến  $T1$ . Chúng tôi nhận thấy trong biểu đồ, nó có thể tồn tại một thành phần phi tuyến tính cho  $S3$ .

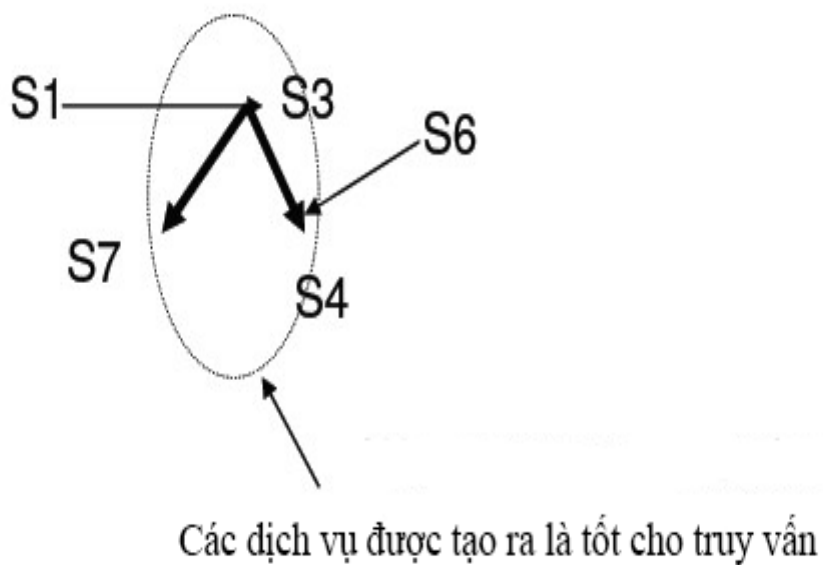
Hình 12 (a) cho thấy ánh xạ giữa các nguyên tử truy vấn và các dịch vụ được cung cấp trong  $T1$ . Hình 12 (b) cung cấp tập hợp có thứ tự sau khi ánh xạ. Cuối cùng, mục đích là tìm một đường dẫn (hoặc một đồ thị con) bắt đầu từ  $S1$  đến  $S4$  bao gồm  $S6, S3, S7$  trên đồ thị định hướng ( $S8$  là một dịch vụ nguyên tử). Con đường duy nhất mà chúng tôi có thể tìm thấy là một đồ thị con là thành phần phi tuyến ( $S3, S4$ ) và ( $S3, S7$ ). Bây giờ tính toán tỷ lệ bao gồm:  $CR = \frac{3+1}{6} = \frac{2}{3}$ . Phép tính tương tự được thực hiện cho tất cả các phương ngang nhỏ nhất và cuối cùng chọn một phép tính có tỷ lệ tốt nhất.



Hình 11: đồ thị định hướng cho đường ngang T1. [5]



Hình 12: (a) các nguyên tử thuộc dịch vụ che tốt nhất T1, (b) phủ tốt nhất được sắp xếp liên quan đến các nguyên tử truy vấn. [5]



Hình 13: Các dịch vụ được tổng hợp tốt trong biểu đồ định hướng.[5]

### Thuật toán Bcov :

Yêu cầu: Một tập BS: tỷ lệ chuyển đổi tối thiểu của truy vấn Q do BCOV cung cấp.

Đảm bảo: Một tập WBS: bao gồm tốt nhất của truy vấn Q.

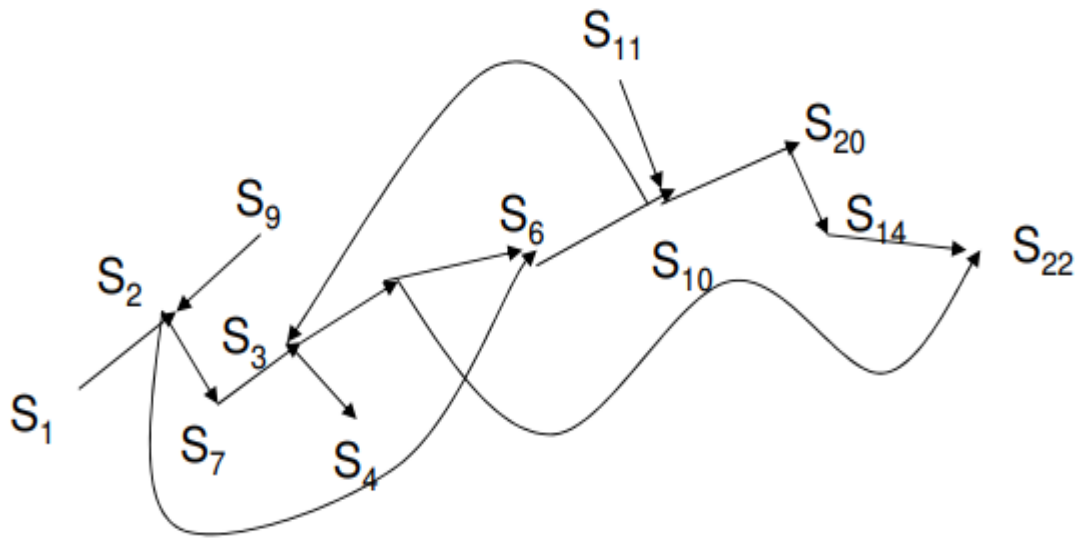
- 1: **for all** tập bìa  $SC_i \in BS$  **do**
- 2: Xây dựng đồ thị định hướng  $OG_i$  for  $SC_i$
- 3: Thứ tự  $SC_i$  liên quan đến thứ tự của các nguyên tử truy vấn.
- 4: **for all**  $S_j \in SC_i$  **do** { // Tìm kiếm các dịch vụ web nguyên tử }
- 5: **if**  $S_j$  là nguyên tử **then**
- 6: Dịch vụ có thể được xem xét theo bất kỳ thứ tự nào trong trang bìa tốt nhất.
- 7: **else**
- 8: Nó được tạo ra và giữ lại.
- 9: **end if**
- 10: **end for**
- 11: Tìm một đường dẫn (cho bố cục tuyến tính hoặc một đồ thị con cho bố cục phi tuyến tính) cho biết từ dịch vụ đầu tiên đến dịch vụ cuối cùng trong OG.
- 12: Tính toán tỷ lệ bao phủ  $CR_i$
- 13: **end for**
- 14: Chọn bìa tốt nhất liên quan đến tỷ lệ che phủ tốt nhất

### 2.3.2. Giải thuật Dijkstra cho bài toán đường đi ngắn nhất[5][6]

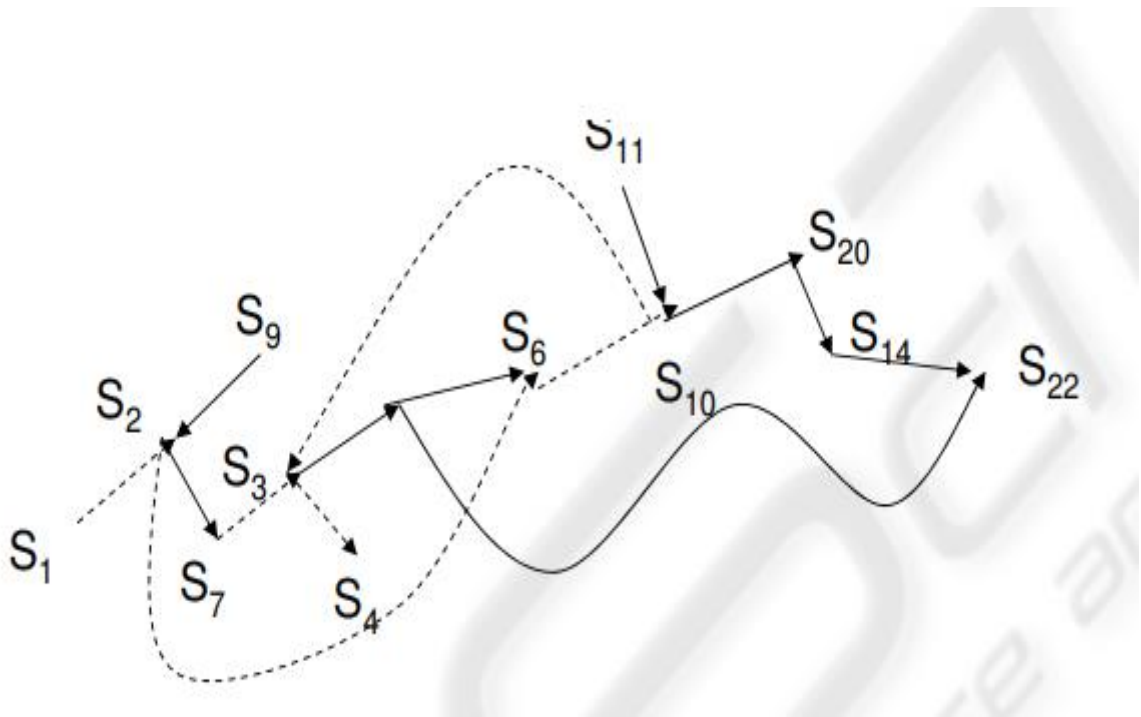
Nó được xác định như sau: Cho đồ thị  $G = (S, E)$  và  $e \in E$ , chúng ta liên kết với mỗi cạnh một trọng số  $l(e) \geq 0$  (trong trường hợp của chúng ta, chúng ta giả sử tất cả các cạnh có trọng số bằng 1) chúng ta gọi nó là độ dài của cạnh  $u$ . Tìm một đường đi có tâm cao  $\mu$ , bắt đầu từ một cạnh  $S_i$  đến một cạnh khác  $S_j$ , chẳng hạn như tổng độ dài  $l(\mu) = \sum_{e \in \mu} l(e)$  càng nhỏ càng tốt.

Có nhiều thuật toán xử lý bài toán đường đi ngắn nhất, tôi sử dụng thuật toán được phát triển bởi thuật toán Dijkstra [6], ký hiệu nó bằng Dijkstra  $(G, s)$ , mà cho trước một đồ thị  $G$  tìm tất cả các đường đi ngắn nhất từ  $s$  đến mỗi đỉnh khác trong đồ thị. Và Đường ngắn nhất  $(G, s, t)$  sử dụng Dijkstra để tìm đường đi ngắn nhất từ  $s$  đến  $t$

Giả định rằng đường ngang T1 là lớp phủ tốt nhất của phương ngang tối thiểu và nhà cung cấp duy trì một biểu đồ định hướng toàn cầu động cho thành phần của các dịch vụ web được mô tả trong hình trên. Mục đích bây giờ là tìm một đường dẫn hoặc một đồ thị con cho mỗi dịch vụ liên tiếp của  $\overline{T1}$ : Đường ngắn nhất  $(G, S1, S6)$ , Đường ngắn nhất  $(G, S6, S3)$ , Đường ngắn nhất  $(G, S3, S7)$ , Đường ngắn nhất  $(G, S7, S4)$ . S8 coi như như một nguyên tử lỗi. Cuối phép tính, ta nhận được một đồ thị con toàn cầu đáp ứng truy vấn bằng cách soạn các dịch vụ không được cung cấp bởi Bcov. Vì vậy, đồ thị con gây ra giữa S1 và S6 là một đường được xác định bởi  $(S1, \overline{S2}, S6)$  (nó là một thành phần tuyến tính). Thực hiện quy trình tương tự cho phần còn lại của cặp dịch vụ. Đồ thị con thu được được mô tả hình trên. Hai dịch vụ S2 và S10 đã được giới thiệu trong đề xuất bên cạnh các dịch vụ được cung cấp bởi thuật toán Bcov.



Hình 14: Biểu đồ định hướng toàn cầu.[5]



Hình 15: Thành phần hoàn thành truy vấn.[5]

Thuật toán ngắn nhất SCA:

Yêu cầu: Chuyển khoản tối thiểu do Bcov cung cấp:

T

### Biểu đồ định hướng toàn cầu GOG

Đảm bảo: Một đồ thị con là một thành phần có thể có bao gồm tất cả các dịch vụ liên quan đến mức tối thiểu cho chuyên ngang.

- 1: **for all** dịch vụ liên tiếp  $S_i, S_{i+1}$  2 T do
- 2: Thực hiện **shortestPath**(GOG,  $S_i, S_{i+1}$ )
- 3: **end for**
- 4: Cung cấp một đồ thị con bao gồm các dịch vụ đáp ứng truy vấn.

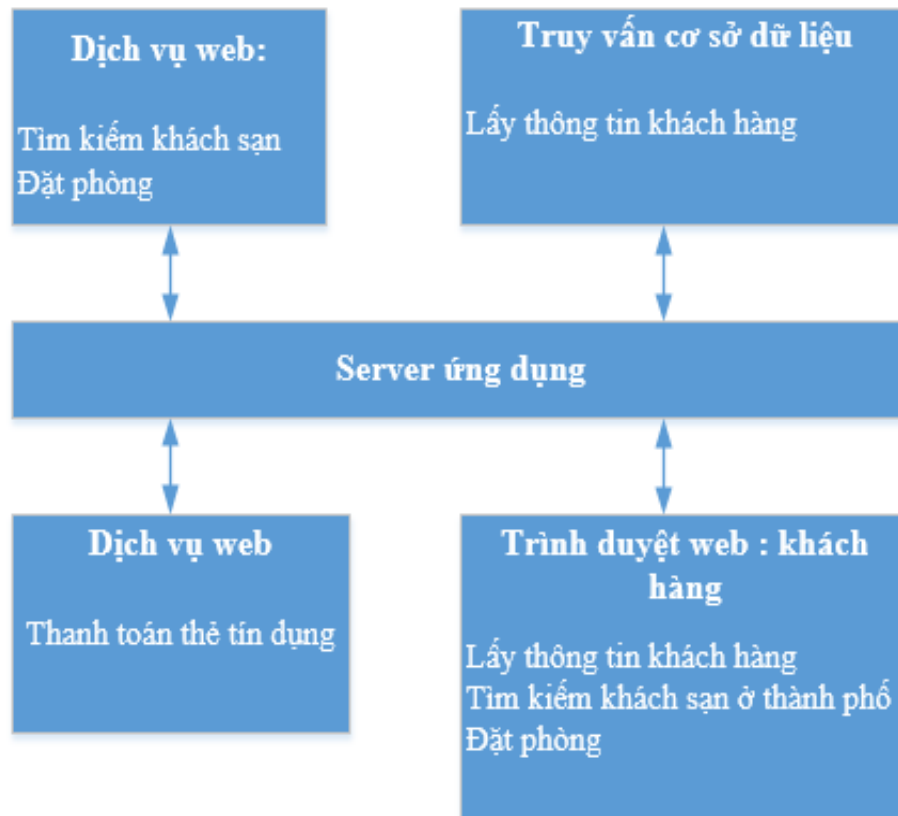
Độ phức tạp của giải thuật Dijkstra trên đồ thị có  $n$  nút và  $m$  cạnh là  $O(n^2)$ . Trong đó  $O(n^2)$  cho lựa chọn nút và  $O(m)$  cho các cập nhật sai lệch. Mặc dù nó là phức tạp tốt nhất có thể cho đồ thị dày đặc, nhưng có thể được cải thiện đáng kể đối với đồ thị thưa thớt.

Như vậy việc khám phá lựa chọn các dịch vụ web thông qua 2 thuật toán trên dựa trên lý thuyết mô hình đồ thị, được xây dựng dựa trên giải thuật Bcov ban đầu được cung cấp cho việc khám phá các dịch vụ web động. Phương pháp đầu tiên sử dụng tất cả các số chuyển đổi tối thiểu được cung cấp bởi thuật toán Bcov và cố gắng tìm ra tỷ lệ bao trùm của biểu đồ định hướng tốt nhất. Tuy nhiên, phương thức thứ hai chỉ sử dụng đường đi tối thiểu với chi phí tốt nhất và cố gắng tìm đường dẫn ngắn nhất giữa tất cả các dịch vụ được truy vấn trong biểu đồ toàn cầu.

#### 2.4. Lựa chọn dịch vụ web trong hệ thống hướng dịch vụ

Xem xét bài toán dưới đây về lựa chọn dịch vụ web và SOA, Bài toán đặt phòng khách sạn qua web. Khách hàng đăng nhập vào hệ thống để biết thông tin của mình.

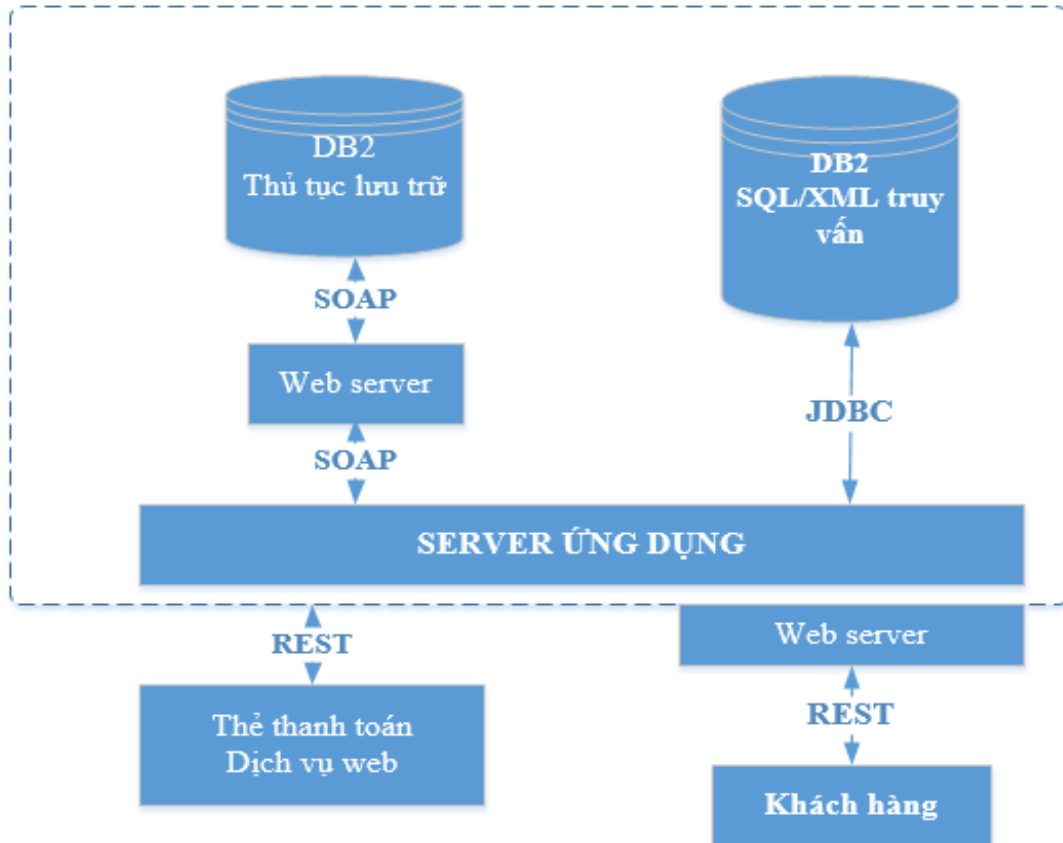
- Tìm kiếm các thành phố có sẵn các khách sạn để đặt giá.
- Khách hàng chọn một khách sạn rồi đặt phòng



*Hình 16: Tính năng hệ thống*

Khách hàng thực hiện các cuộc gọi REST từ máy tính của mình đến server máy chủ., đầu phía máy chủ có thể :

- Kết nối tới cơ sở dữ liệu trực tiếp để lấy truy vấn thông tin khách hàng.
- Thực hiện các Messages SOAP tới cơ sở dữ liệu.
- Gọi REST thực hiện giao dịch bằng thẻ tín dụng bên ngoài như PayPal thông qua nhà cung cấp dịch vụ.

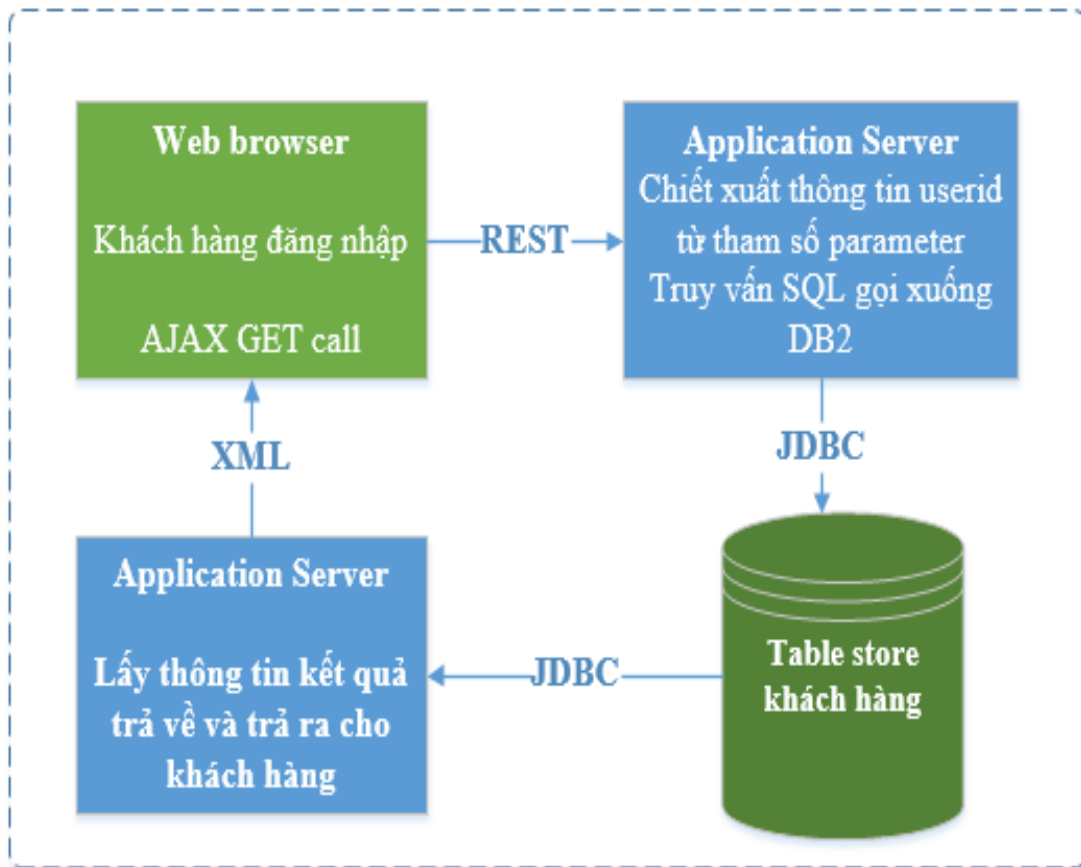


*Hình 17: Kiến trúc mức thiết kế*

#### 2.4.1. Đăng nhập

Khách hàng đăng nhập để lấy thông tin của mình.

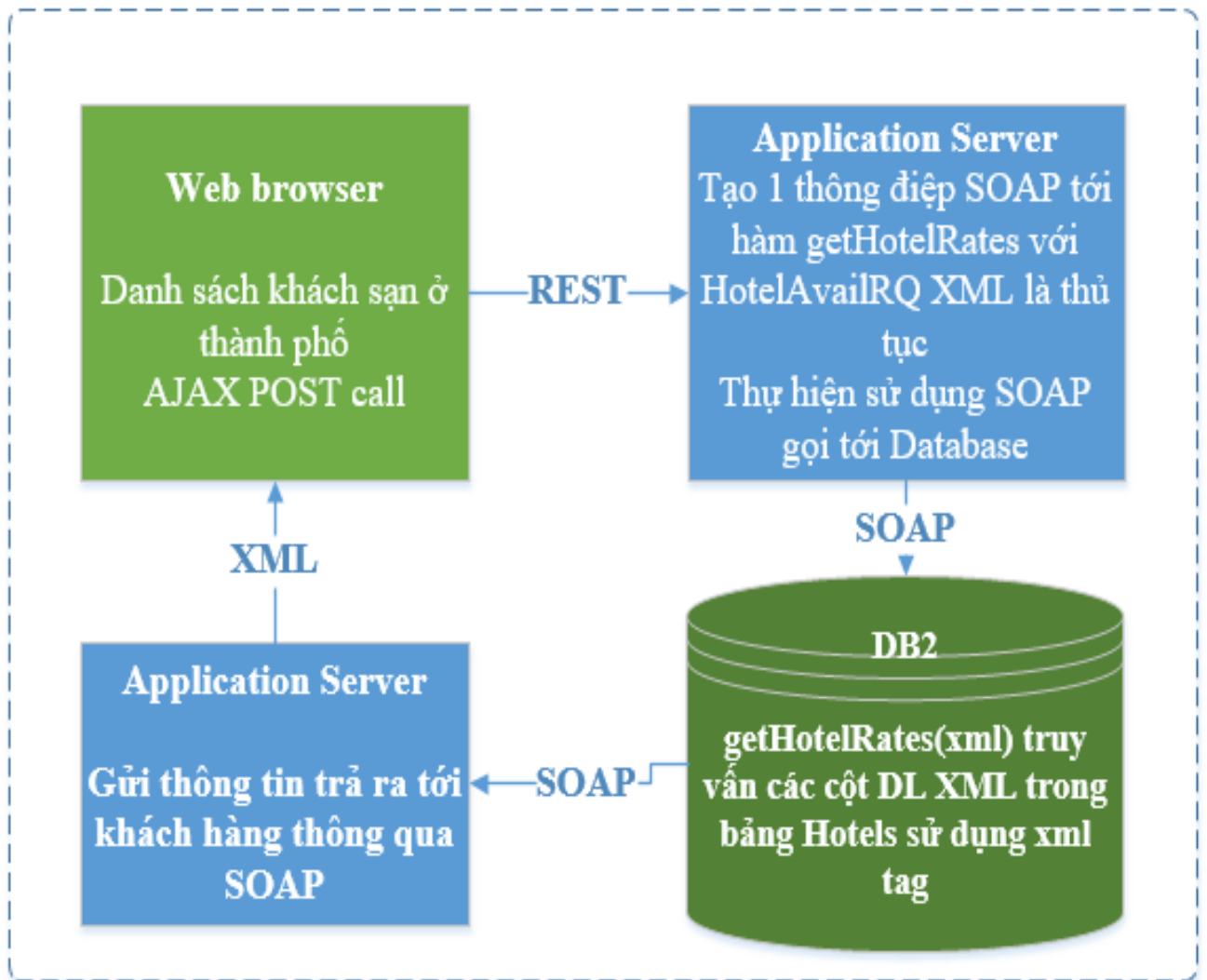




*Hình 18: Khách hàng đăng nhập và lấy thông tin*

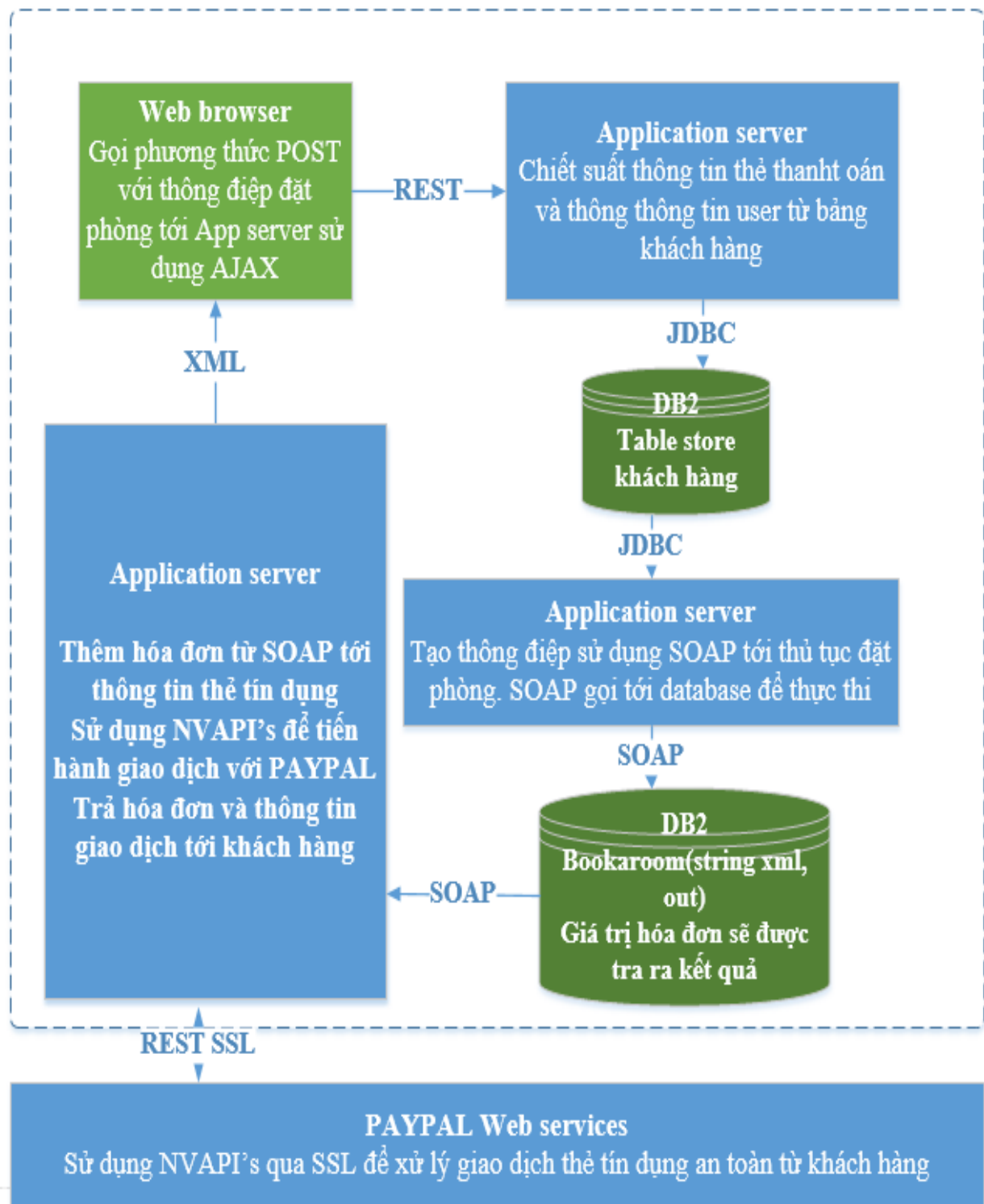
Thông tin khách hàng, là tập hợp dữ liệu thể hiện dưới dạng tài liệu XML, lưu trữ thể hiện ở các cột XML. Thẻ tín dụng được lưu trữ dưới dạng tài liệu XML, trong đó nó được bảo mật và mã hóa dưới dạng nhị phân.

#### 2.4.2. Khách hàng tìm kiếm khách sạn



Hình 19: Danh sách khách sạn và giá phòng

### 2.4.3. Đặt phòng

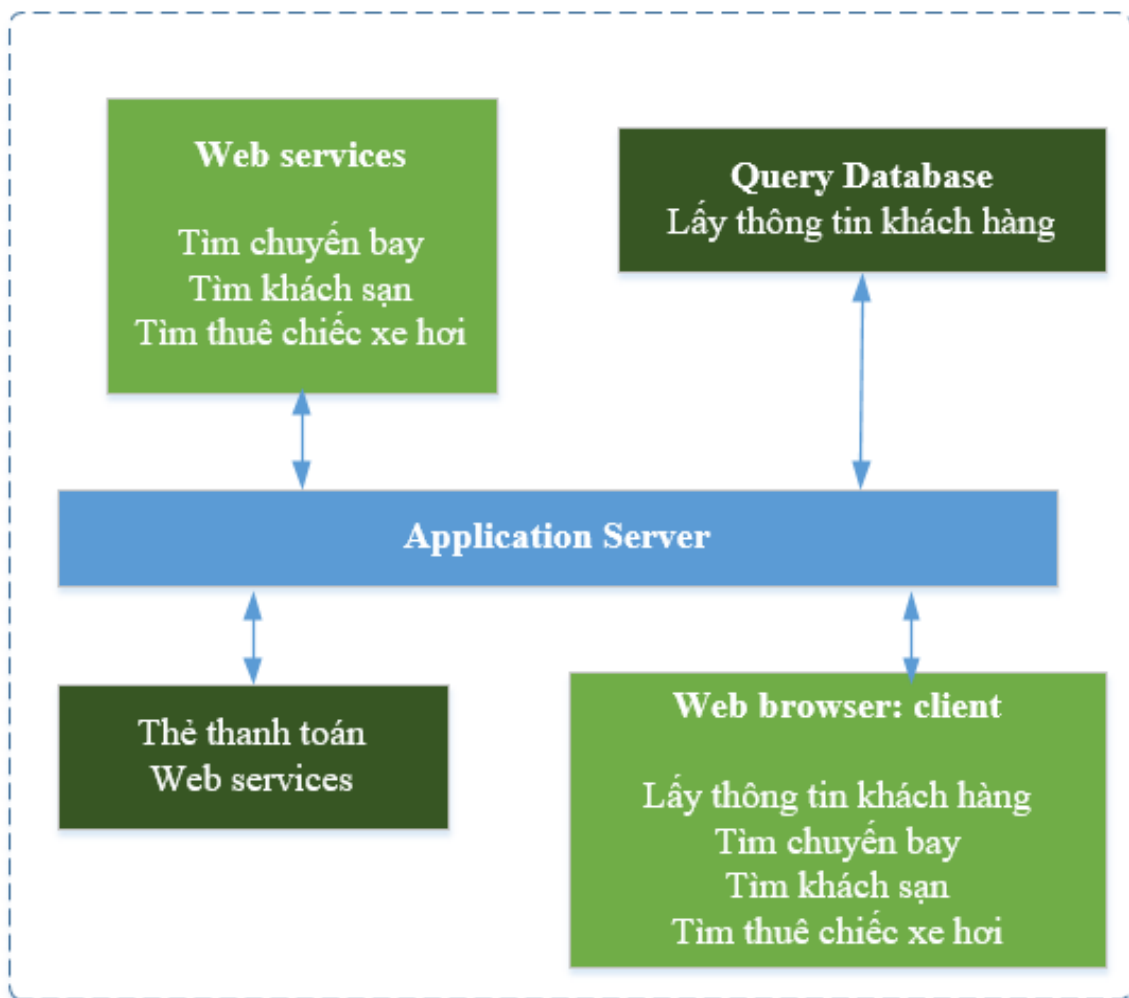


Hình 20: Đặt phòng

### CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ.

#### XÂY DỰNG HỆ THỐNG "TÌM THÔNG TIN TOUR DU LỊCH" SỬ DỤNG CÁCH TIẾP CẬN HƯỚNG DỊCH VỤ

##### 3.1. Khảo sát và phân tích hệ thống



Hình 21 : Kiến trúc mức tính năng

##### 3.1.1. Khảo sát

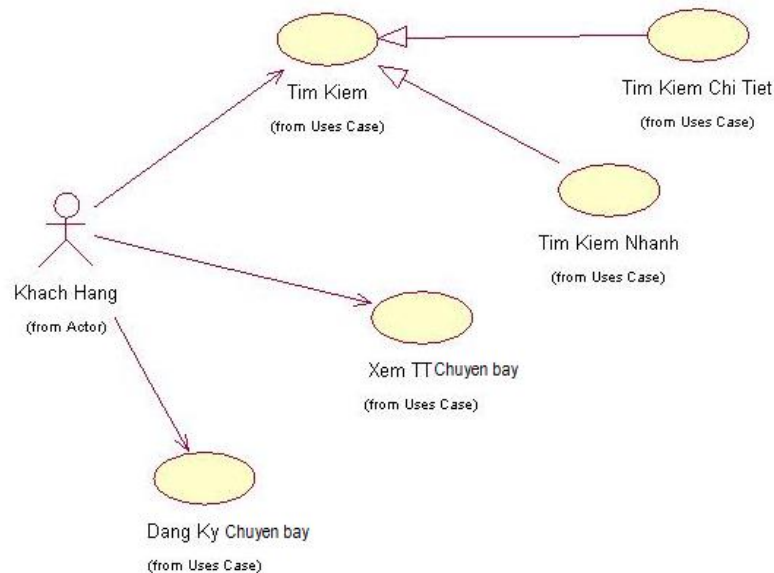
Nhu cầu tìm kiếm thông tin và giao dịch trên mạng ngày càng chiếm vị trí quan trọng. Người dùng đang hướng đến những dịch vụ tiện ích hơn, việc kết hợp những dịch vụ riêng lẻ càng tạo nên tiện ích cho người sử dụng, trong bài

tập tìm kiếm thông tin chuyến du lịch này mình triển khai kiến trúc hướng dịch vụ SOA bằng cách kết hợp các dịch vụ tìm chuyến bay, khách sạn và thuê xe.

- Khách hàng đăng nhập lấy thông tin và dùng tài khoản để tìm kiếm và thanh toán.
- Khách hàng tìm thông tin dựa vào địa điểm du lịch, tìm kiếm nhanh danh sách các chuyến bay, khách sạn, dịch vụ thuê xe trong chuyến du lịch đó.

### 3.1.2. Phân tích hệ thống

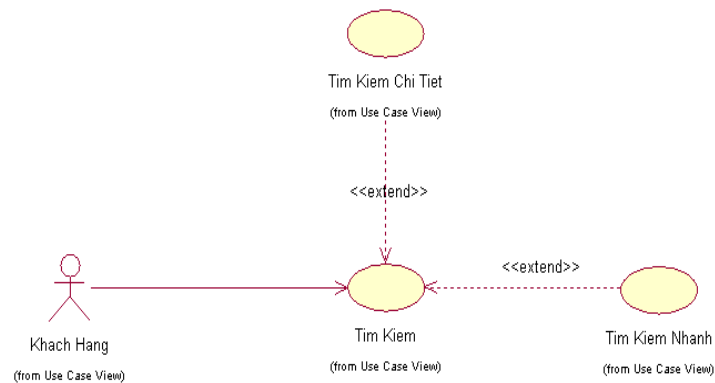
#### Sơ đồ use case tổng quát



Hình 22: Sơ đồ use case tổng quát

#### 3.1.2.1. Chức năng tìm kiếm thông tin chuyến bay

##### a. Sơ đồ use case tìm kiếm chuyến bay



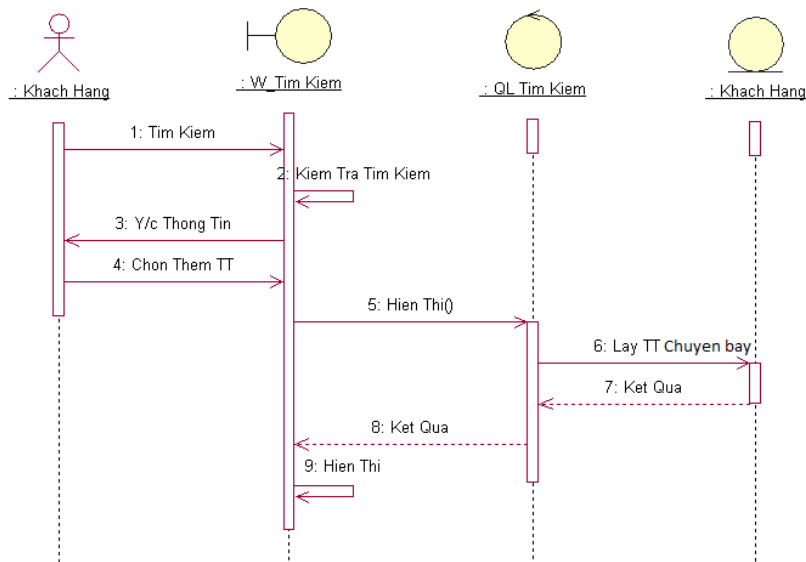
Hình 23: Sơ đồ use case tìm kiếm thông tin chuyến bay

### Đặc tả use case "Khách hàng tìm kiếm thông tin chuyến bay"

Tên Use case	Tìm kiếm thông tin chuyến bay
Actor	Khách hàng
Mô tả	Khách hàng chọn chức năng tìm kiếm thông tin chuyến bay
Luồng sự kiện chính	<p>Actor đăng nhập.</p> <p>Actor chọn chức năng Tìm kiếm chuyến bay.</p> <p>Actor điền thông tin chuyến bay cần tìm.</p> <p>Hệ thống hiển thị màn hình thông tin chuyến bay.</p> <p>Extend Use Case DangNhap</p> <p>Extend Use Case Timkiemthongtinchuyenbay</p>
Luồng sự kiện phụ	<p>5. Đặt vé thành công, Actor có thể tìm kiếm thông tin Khách sạn, và đặt phòng.</p> <p>6. Actor ấn nút thoát.</p>
<Extend Use Case>	<p>DangNhap</p> <p>Actor chọn chức năng đăng nhập.</p> <p>Kiểm tra thông tin không rỗng.</p>

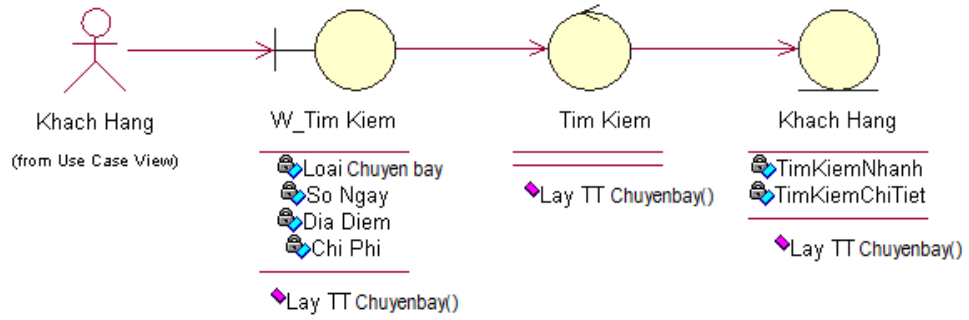
	<p>Actor nhấn button đăng nhập.</p> <p>Hiện thị thông tin khách hàng khi đăng nhập thành công.</p> <p>Thông tin khách hàng có thể dùng để xác nhận mua vé.</p> <p>Rẽ nhánh 1:</p> <ol style="list-style-type: none"> <li>1. Kiểm tra thông tin bị rỗng.</li> <li>2. Lập lại bước 1 luồng sự kiện chính.</li> </ol> <p>Rẽ nhánh 2:</p> <ol style="list-style-type: none"> <li>1. Thông tin bị trùng.</li> <li>2. Lập lại bước 1 luồng sự kiện chính.</li> </ol> <p>Rẽ nhánh 3:</p> <ol style="list-style-type: none"> <li>1. Kiểm tra thông tin khách hàng hợp lệ.</li> <li>2. Kiểm tra CSDL.</li> </ol>
--	---

### b. Biểu đồ trình tự tìm kiếm thông tin chuyến bay



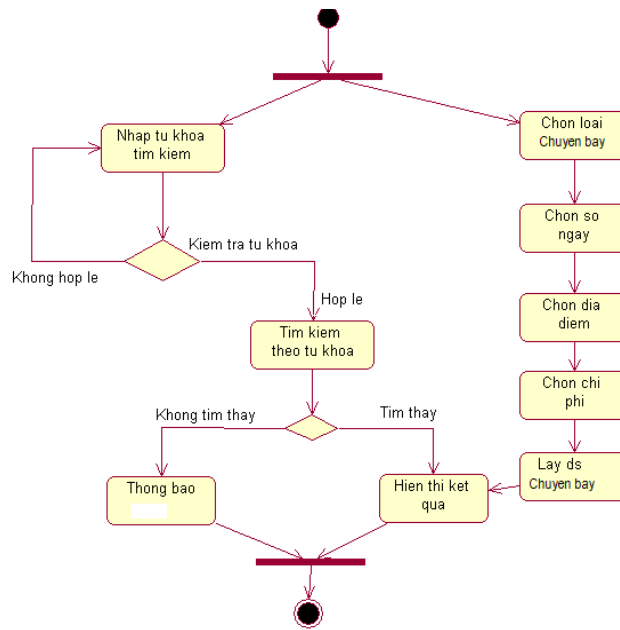
Hình 24: Biểu đồ trình tự tìm kiếm thông tin chuyến bay

### c. Sơ đồ lớp tìm kiếm thông tin chuyến bay



Hình 25: Sơ đồ lớp tìm kiếm thông tin chuyến bay

d. Biểu đồ hoạt động tìm kiếm thông tin chuyến bay

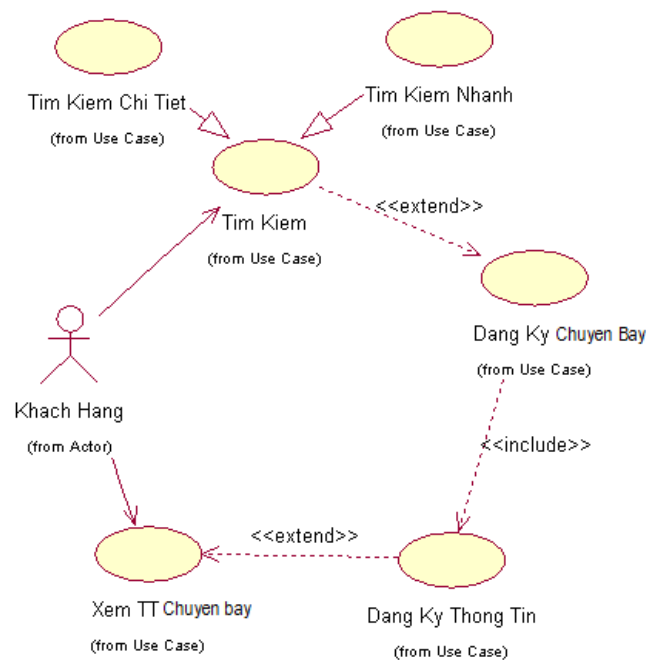


Hình 26: Biểu đồ hoạt động tìm kiếm thông tin chuyến bay

3.1.2.2. Chức năng Đặt vé chuyến bay

a. Sơ đồ Use case đặt vé chuyến bay





Hình 27: Hình Use Case Đặt vé chuyến bay

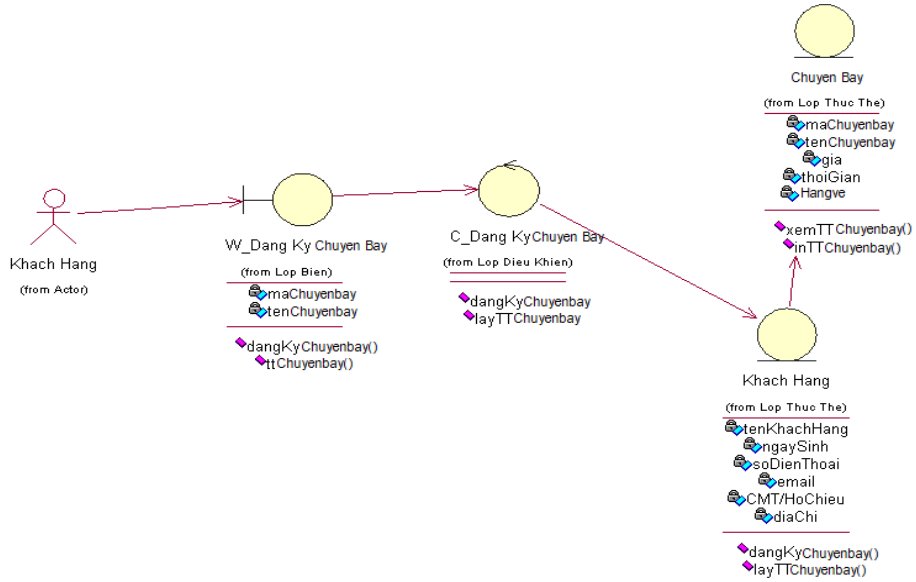
### Đặc tả Use Case" Đặt vé chuyến bay"

Tên Use case	Đặt vé chuyến bay
Actor	Khách hàng
Mô tả	Khách hàng sau khi tìm kiếm thông tin chuyến bay, có thể tiến hành đặt vé, dựa vào thông tin của mình, và thanh toán.
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Actor đăng nhập.</li> <li>2. Actor chọn chức năng Tìm kiếm chuyến bay.</li> <li>3. Hệ thống hiển thị màn hình thông tin chuyến bay.</li> </ol> Extend Use Case DangNhap Extend Use Case Timkiemthongtinchuyenbay Extend Use Case Dangkychuyenbay
Luồng sự kiện phụ	<ol style="list-style-type: none"> <li>4. Đặt vé thành công, Actor có thể tìm kiếm thông tin Khách sạn, và đặt phòng.</li> <li>5. Actor ấn nút thoát.</li> </ol>

<p>&lt;Extend Use Case&gt;</p>	<p>Timkiemthongtinchuyenbay</p> <p>6. Actor nhập thông tin chuyến bay.  7. Kiểm tra thông tin không rỗng.  8. Actor nhấn button tìm chuyến bay.  9. Hiển thị thông tin chuyến bay.  10. Thoát.</p> <p>Rẽ nhánh 1:</p> <ol style="list-style-type: none"> <li>1. Kiểm tra thông tin bị rỗng.</li> <li>2. Lập lại bước 1 luồng sự kiện chính.</li> </ol> <p>Rẽ nhánh 2:</p> <ol style="list-style-type: none"> <li>1. Thông tin bị trùng.</li> <li>2. Lập lại bước 1 luồng sự kiện chính.</li> </ol> <p>Rẽ nhánh 3:</p> <ol style="list-style-type: none"> <li>1. Kiểm tra thông tin chuyến bay hợp lệ.</li> <li>2. Kiểm tra CSDL.</li> </ol>
<p>&lt;Extend Use Case&gt;</p>	<p>Dangkychuyenbay</p> <p>Actor chọn thông tin chuyến ba từ danh sách chuyến bay.  Hiển thị thông tin chuyến bay lên Form.  Actor chọn button Đặt vé.  Actor xem thông tin mình + thông tin chuyến bay.  Thanh toán.  Hiển thị thông tin thành công.  Actor nhấn nút thoát.  Hủy màn hình Đặt vé.</p> <p>Rẽ nhánh :</p> <ol style="list-style-type: none"> <li>1. Kiểm tra thông tin bị rỗng.</li> </ol>

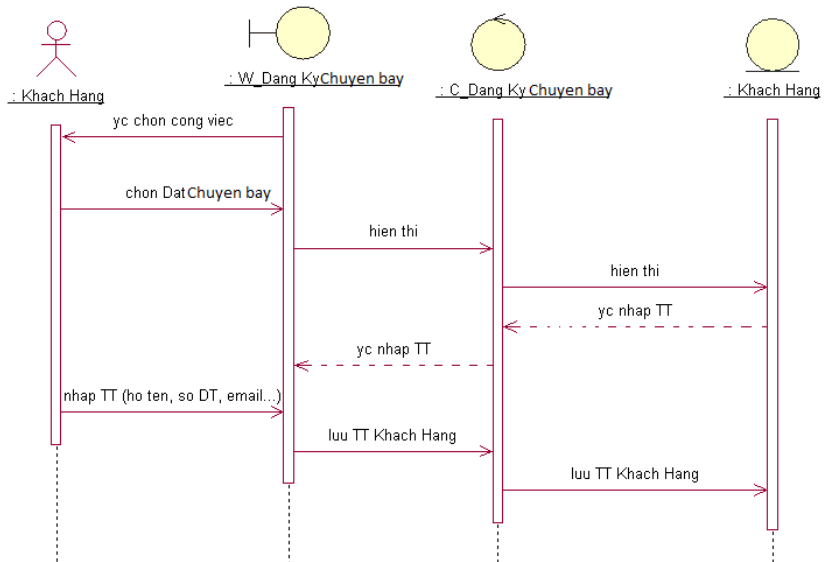
2. Lập lại bước 1 luồng sự kiện chính.

b. Sơ đồ class đặt vé chuyến bay



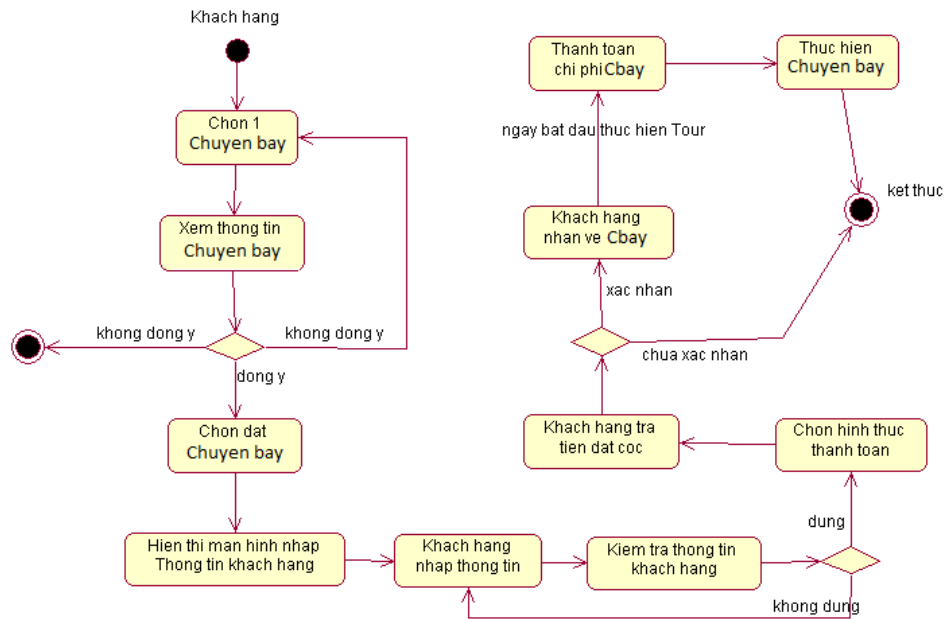
Hình 28: Sơ đồ class Đặt vé chuyến bay

c. Biểu đồ trình tự đặt vé chuyến bay



Hình 29: Biểu đồ trình tự đặt vé chuyến bay

d. Biểu đồ hoạt động đặt vé chuyến bay

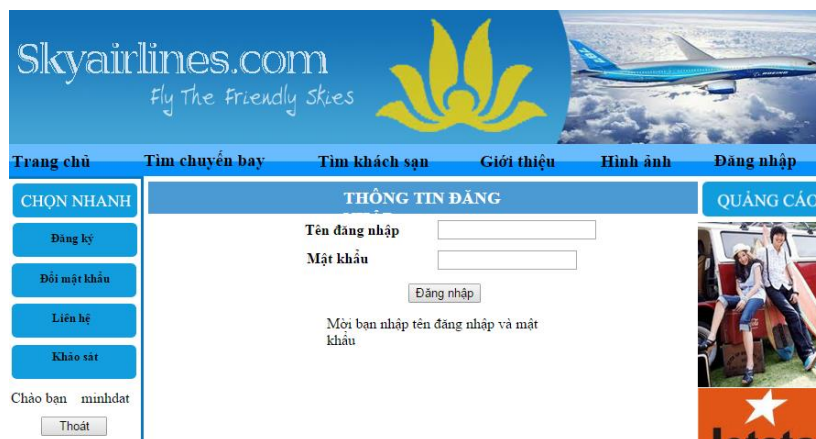


Hình 30: Biểu đồ hoạt động Đặt vé chuyến bay

## 3.2. Thiết kế và triển khai hệ thống

### 3.2.1. Chức năng đăng nhập

- Nhập tên đăng nhập và mật khẩu vào textbox.
- Ấn button\_dn đồng thời kiểm tra thông tin đúng, đăng nhập thành công.

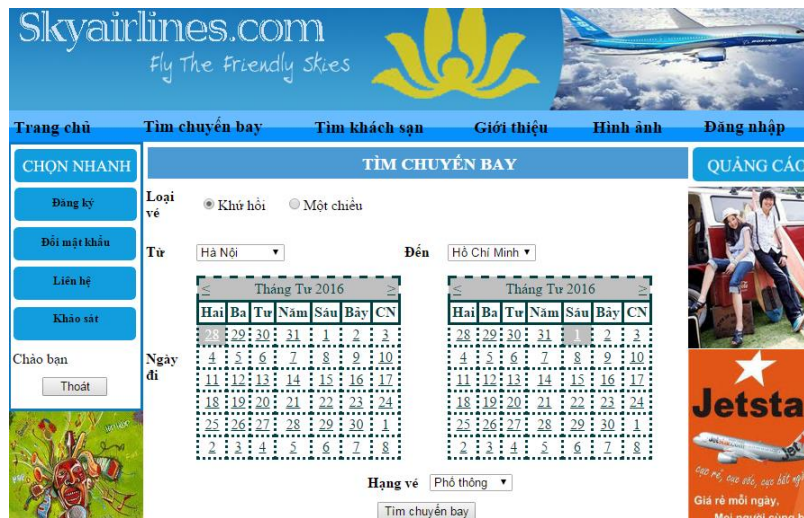


Hình 31: Giao diện chức năng đăng nhập

### 3.2.2. Chức năng tìm chuyến bay

Thực hiện tìm kiếm thông tin chuyến bay:

- Chọn radiobutton khứ hồi, một chiều.
- Chọn DropDownList nơi đi và nơi đến.
- Chọn Calendar ngày khởi hành.
- Chọn DropDownList hạng vé.



Hình 7: Chức năng tìm chuyến bay

### 3.2.3. Chức năng tìm kiếm khách sạn

- Chọn DropDownList\_Tp thành phố cần tìm khách sạn.
- Chọn DropDownList\_HangKs



Hình 8: Chức năng tìm khách sạn

### 3.2.4. Chức năng đặt vé chuyến bay

- Trước tiên thực hiện tìm kiếm thông tin chuyến bay:
  - ✓ Chọn radiobutton khứ hồi, một chiều.
  - ✓ Chọn DropDownList nơi đi và nơi đến.
  - ✓ Chọn Calendar ngày khởi hành.
  - ✓ Chọn DropDownList hạng vé.
- Chọn chuyến bay cần đặt vé từ GridView.
- Ấn nút Đặt vé.

TÌM CHUYẾN BAY

Loại vé  Khứ hồi  Một chiều

Từ  Đến

Ngày đi

Tháng Năm 2016						
Hai	Ba	Tur	Năm	Sáu	Bảy	CN
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Ngày đi

Tháng Năm 2016						
Hai	Ba	Tur	Năm	Sáu	Bảy	CN
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Hạng vé

KẾT QUẢ TÌM KIẾM

mach	noidi	noiden	ngaydi	giodi	giaphothong	giathuonggia	choPT	choTG
HTS01	Hà Nội	Hồ Chí Minh	Monday	6h00	700	1000	20	5
STH01	Hồ Chí Minh	Hà Nội	Friday	6h00	700	1000	28	8

Hình 34: Chức năng đặt vé chuyến bay

### 3.2.5. Chức năng tìm kiếm nhanh

- Chức năng hỗ trợ khách hàng tìm kiếm nhanh danh sách chuyến bay và khách sạn và dịch vụ thuê xe dựa vào nơi đi và nơi đến.

- Tư vấn khách hàng tổng chi phí chuyến đi.

TÌM NHANH CHUYẾN BAY VÀ KHÁCH SẠN, DỊCH VỤ XE

Khởi hành

**Từ**  **Đến**

KẾT QUẢ TÌM KIẾM

Danh sách chuyến bay

macb	noidi	noiden	ngaydi	giodi	giaphothong	giathuonggia	choPT	choTG
HTS01	Hà Nội	Hồ Chí Minh	Monday	6h00	700	1000	20	5
HTS02	Hà Nội	Hồ Chí Minh	Sunday	9h00	700	1000	10	15

Thông tin dịch vụ khách sạn

maks	thanhpho	hang	ten	diachi	gia
HCM1	Hồ Chí Minh	5 sao	New World	72- Lê Lai- Phường Bến Thành- Quận 1	300
HCM2	Hồ Chí Minh	4 sao	New Pacific	9,11- Kỳ Đồng- Phường 9- Quận 3	240
HCM3	Hồ Chí Minh	3 sao	Continental	225- Hai Bà Trưng- Phường 6- Quận 3	170
HCM4	Hồ Chí Minh	Dưới 2 sao	Asian Ruby 2	28- Trương Định- Quận 1	100
HCM5	Hồ Chí Minh	5 sao	The Reverie Saigon	22-36 Nguyen Hue Boulevard, Quận 1	300

Dịch vụ thuê xe du lịch

maxe	thanhpho	hangxe	tenxe	socho	gia
Ôtô_HCM1	Hồ Chí Minh	Toyota	Fortuner 2.7V (4x4)	7 chỗ	120
Ôtô_HCM2	Hồ Chí Minh	Toyota	Hiace Động cơ xăng	16 chỗ	150
Ôtô_HCM4	Hồ Chí Minh	Toyota	Toyota Coaster 35 chỗ	35 chỗ	300
Ôtô_HCM6	Hồ Chí Minh	Toyota	Camry 2.5Q	5 chỗ	110
Ôtô_HCM7	Hồ Chí Minh	Toyota	Toyota 86	4 chỗ	100
Ôtô_HCM8	Hồ Chí Minh	Rolls Royce	Rolls Royce wraith	4 chỗ	1000
Ôtô_HCM9	Hồ Chí Minh	Mercedes	Mercedes C200 2016	4 chỗ	500

Tổng chi phí để đến Tp Hồ Chí Minh với những dịch vụ tốt nhất là 2300 USD

Tổng chi phí để đến Tp Hồ Chí Minh với những dịch vụ phổ thông là 900 USD

Hình 35: Giao diện chức năng tìm kiếm nhanh

### 3.3. Đánh giá

Chức năng tìm kiếm nhanh nhiều dịch vụ như tìm kiếm thông tin chuyến bay, tìm kiếm khách sạn, tìm kiếm dịch vụ thuê xe du lịch,... từ đó tổng hợp

thông tin về giá để tư vấn cho khách hàng các gói dịch vụ phù hợp với từng đối tượng. Việc xây dựng dịch vụ web triển khai theo mô hình hướng đối tượng SOA đã tạo nên những ưu điểm:

- Với việc lựa chọn nhiều dịch vụ sẽ giúp người sử dụng tìm kiếm nhanh chóng thông tin về chuyến du lịch.
- Tiết kiệm thời gian tìm kiếm những thông tin rời rạc cho người sử dụng, qua đó việc lựa chọn nhiều dịch vụ kết hợp những thuật toán ưu rút ra những tư vấn về giá, giúp người dùng dễ dàng so sánh tổng chi phí chuyến đi cho phù hợp. Một cách tiếp cận SOA cho nhà cung cấp dịch vụ.

Hệ thống sau khi hoàn thành được sử dụng để đánh giá mô hình và thuật toán đề xuất.

Tham số dùng để đánh giá là thời gian lựa chọn các dịch vụ và chi phí sử dụng các dịch vụ. Chi phí thực tế cho các dịch vụ web chưa được sử dụng, mà các dịch vụ chỉ được gán chi phí thực nghiệm với các trọng số khác nhau, với mục đích đánh giá hiệu quả của phương pháp đề xuất, do đó chưa quan tâm tới đơn vị của chi phí.

Kết quả thử nghiệm được thể hiện trong bảng dưới đây:

Lần thử nghiệm	Số dịch vụ thử nghiệm	Thời gian thực hiện (ms)		Chi phí dịch vụ	
		Phương pháp thủ công	Phương pháp đề xuất	Phương pháp thủ công	Phương pháp đề xuất
1	10	3325	450	26	24
2	15	6532	673	34	34
3	20	14578	682	45	40
4	25	30467	793	68	56
5	30	60569	875	124	78

*Kết quả đánh giá thực nghiệm*



Từ kết quả so sánh có thể thấy rằng phương pháp đề xuất cho hiệu quả đáng kể về thời gian lựa chọn và chi phí sử dụng các dịch vụ web so với khi sử dụng cách tiếp cận thông thường. Về mặt lý thuyết, nếu sử dụng phương pháp vét cạn có thể tìm được phương án cho chi phí nhỏ nhất, tuy nhiên thời gian lựa chọn sẽ là làm số mũ, do đó với số lượng lớn các dịch vụ web thành phần sẽ không khả thi khi áp dụng thực tế.

#### **CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TIẾP THEO**

Lựa chọn dịch vụ web ngày càng được sử dụng phổ biến trong việc giải quyết các bài toán phức tạp. Với khả năng vượt trội của mình, Web Service đã đáp ứng mạnh mẽ các quy trình nghiệp vụ phức tạp của các nhà phát triển dịch vụ. Người dùng sử dụng dịch vụ có thể lựa chọn dịch vụ web thích hợp dựa trên truy vấn của mình là việc làm cần thiết được trình bày nội dung chính của đề tài.

- Luận văn đã trình bày khái quát về cách tiếp cận kiến trúc hướng dịch vụ SOA, Cách xây dựng và triển khai Web Services dựa trên các thành phần SOAP, WSDL, UDDI.
- Luận văn nghiên cứu bài toán lựa chọn dịch vụ web và các thuật toán lựa chọn dịch vụ web dựa trên mô hình đồ thị, cụ thể là 2 thuật toán Bcov và Dijkstra.
- Luận văn đã đưa ra được các khái niệm cơ bản, các ví dụ thực tiễn về lựa chọn dịch vụ web phù hợp với hướng nghiên cứu cho bài toán lựa chọn dịch vụ web.
- Thông qua hệ thống tìm thông tin tour du lịch triển khai theo mô hình SOA, cho thấy nhiều ưu điểm cho người sử dụng dịch vụ về tính tiện lợi, tiết kiệm thời gian, tư vấn về dịch vụ, và nhiều lợi thế của mô hình SOA trong kinh doanh.

Do thời gian nghiên cứu còn hạn hẹp nên luận văn không tránh khỏi những thiếu sót: Luận văn mới chỉ được thực hiện đối với các Web Services ở localhost. Để có thể triển khai được mới môi trường bên ngoài gắn với thực tế

hơn thì cần áp dụng ở môi trường Internet.

Nhận thấy rõ bài toán lựa chọn dịch vụ web là bài toán rất cấp thiết do vậy trong tương lai sẽ tiếp tục phát triển đề tài và khắc phục tìm kiếm ra những thuật toán tối ưu hơn nữa để từ đó áp dụng vào sản xuất thực tiễn đáp ứng nhu cầu nghiệp vụ phức tạp của các doanh nghiệp.

Em xin chân thành cảm ơn!

## TÀI LIỆU THAM KHẢO

- [1] Ethan Cerami, "Web Services Essentials", Feb 2002.
- [2] Prentice Hall, "Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services", 2004.
- [3] Liangzhao Zeng, Boualem Benatallah, Anne H.H. Ngu, Marlon Dumas, Jayant Kalagnanam, Henry Chang, QoS-A ware Middleware for Web Services Composition, Pages [10 - 12].
- [4] Daniela Barreiro Claro, Patrick Albers and Jin-Kao Hao, Web services composition, Pages [208 - 211].
- [5] Benbernou S., Benslimane D., Said Hacid M., Kheddouci H. and Tari K, "A GRAPH-BASED APPROACH FOR WEB SERVICES COMPOSITION", 2007.
- [6] E.J.Dijkstra (1959). A note on two problems in connection with graphs. *Numerische Math*, 1:269–171
- [7] Mc Graw Hill, "SOA-Based Enterprise Integration", 2009.
- [8] Robert Hablutzel, "Developing Web Services with Java APIs for XML", 2005.
- [9] Richard Seroter, "SOA Patterns with BizTalk Server", 2009.
- [10] Deitel, "Visual C Sharp 2005 How to Program", 2005.
- [11] B. Benatallah and F. Casati, editors. *Distributed and Parallel Database, Special issue on Web Services*. Kluwer Academic Publishers, 2002.
- [12] F. Curbera et al. *Unraveling the Web Services: an Introduction to SOAP, WSDL, and UDDI*. *IEEE Internet Computing*, Mar/Apr issue 2002.
- [13] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng. *Quality Driven Web Services Composition*. In *Proceedings of the 12th international conference on World Wide Web (WWW)*, Budapest, Hungary. ACM Press, May 2003 .
- [14] Benatallah.B, Hacid.M.S, Leger.A, Rey.C, and Toumani.F (2005a). *On automating web services discovery*. *VLDB Journal*, 14(1):84–96.

- [15] Yuan.Y, Li.Z, and Sun.W (2006). *Graph search based approach for BPEL4WS Test Generation*. In *IEEE International Conference on Software Engineering Advances ICSEA'06*.
- [16] Hamadi, R. and Benatallah, B. (2003). *A petri net-based model for web service composition*. In *ADC*, pages 191–200
- [17] Medjahed, B., Bouguettaya, A., and Elmagarmid, A. K. (2003). *Composing web services on the semantic web*. *VLDB J.*, 12(4):333–351 .
- [18] Kruskal jr., J. B. : *On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem*. *Proc. Amer. Math. Soc.* 7, 48-50 (1956).
- [19] Berge, C. : *Théorie des graphes et ses applications*, pp. 68-69. Paris : Dunod 1958 .
- [20] Achraf Karray, Rym Teyeb and Maher Ben Jemaa, “A Heuristic Approach for Web-Service Discovery and Selection” In: *International Journal of Computer Science & Information Technology (IJCSIT'13)*, April 2013, vol 5, p 107-119.
- [21] David Jorgensen, *Developing .NET Web Services with XML*, 2002.
- [22] Dieter Jungnickel , *Graphs, Networks and Algorithms*, 2000.
- [23] Wirth, N.: *Algorithms + Data Structures = Programs*. Prentice Hall, Englewood Cliffs, N.J. (1976) .