

## TÍCH HỢP RESTFUL WEB SERVICE TRÊN NỀN TẢNG DI ĐỘNG XÂY DỰNG HỆ THỐNG WEB KHOA CÔNG NGHỆ THÔNG TIN TRÊN ĐIỆN THOẠI ANDROID

Lương Thị Thảo Hiếu\*, Lê Thanh Cúa, Đào Phương Anh

Trường Đại học Kinh tế - Kỹ thuật Công nghiệp

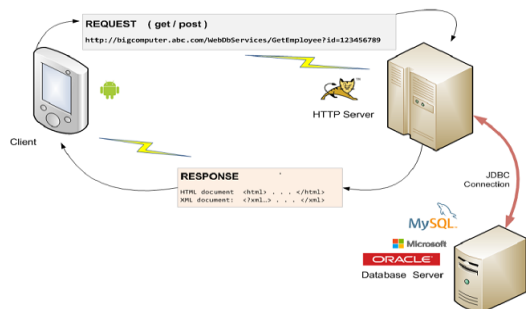
### TÓM TẮT

Ngày nay các thiết bị di động, đặc biệt các thiết bị sử dụng Android đang được sử dụng như thiết bị tiêu thụ Web. Tuy nhiên khi sử dụng di động để duyệt Web gặp nhiều vấn đề: ràng buộc tài nguyên, màn hình nhỏ, dung lượng bộ nhớ thấp, sử dụng kết nối không dây. Vậy lấy dữ liệu thế nào, lưu trữ, biểu diễn ra sao, xây dựng một hệ thống truyền tải thông tin đến người dùng là một vấn đề phức tạp và khó khăn trong quá trình thực thi. Để xây dựng Web thường sử dụng hai loại web service: SOAP hoặc RESTful. Trong bài báo này chúng tôi nghiên cứu kiến trúc mobile web service, sau đó tích hợp RESTful web service, xây dựng hệ thống web quảng bá khoa Công Nghệ Thông Tin trên điện thoại Android.

**Từ khóa:** ANDROID; dịch vụ web di động; RESTful; JSON

### MỞ ĐẦU

Các thiết bị di động (điện thoại thông minh và thiết bị cảm tay) được xem là các thiết bị hạn chế tài nguyên và không phù hợp để sử dụng dịch vụ internet. Các nhà thiết kế di động luôn tập trung mở rộng nâng cấp phần cứng: tăng dung lượng bộ nhớ, tăng cường khả năng tính toán, đồng thời cải thiện hiệu suất các ứng dụng di động. Sự tách biệt giữa giao diện người dùng và dịch vụ logic được cung cấp bởi Web Service là cơ hội mới mang lại dịch vụ internet cho các thiết bị di động. Các ứng dụng chạy trên thiết bị di động, thông qua Web Service, có thể thích nghi với khả năng hạn chế của từng thiết bị.



**Hình 1.** Web service cho ứng dụng đa lớp

Để tích hợp các công nghệ Web Service vào các thiết bị di động tạo thành kiến trúc Mobile Web Service tối ưu, đáp ứng nhu cầu người

sử dụng, cần quan tâm đến tính chặt chẽ của thiết bị này và hệ thống truyền thông di động. Về cơ bản, Web Service được chia thành hai loại chính: RESTful và SOAP based Web Service [1]. Sự phân chia này dựa trên kiểu kiến trúc sử dụng để thực hiện công nghệ.

SOAP viết tắt của Simple Object Access Protocol- giao thức truy cập đối tượng đơn giản- đây là công nghệ hướng đối tượng, định nghĩa các giao thức dùng để trao đổi thông điệp dựa trên XML. Năm 2000 Fieding cho ra đời nguyên lý REST. REST-Representational State Transfer - công nghệ hướng tài nguyên, chứa một tập các chuẩn thiết kế, định nghĩa cách thức đặc biệt sử dụng các chuẩn web như HTTP và URI. REST đã trở thành công nghệ thực thi phổ biến để phát triển Web Service. REST triển khai độc lập, đơn giản, gọn nhẹ, khả năng tương tác cao và gần như thay thế SOAP. Trong bài báo này chúng tôi giới thiệu một số khái niệm Web Service, so sánh XML với JSON, so sánh hai công nghệ thiết kế Web Service: SOAP based và RESTful. Nghiên cứu, lựa chọn kiến trúc mobile web service, đề xuất mô hình hệ thống, mô hình ứng dụng xây dựng hệ thống web quảng cáo khoa Công Nghệ Thông Tin (CNTT) trên nền tảng Android, phục vụ nhu cầu tra cứu thông tin sinh viên, thông tin

\* Tel: 0942 160880, Email: ltthieu@uneti.edu.vn

tuyển sinh, quảng cáo khoa CNTT, lịch cá nhân giảng viên,..

MỘT SỐ KHÁI NIỆM

**Khái niệm 2.1.** Web service (dịch vụ web) là phương thức trao đổi giữa hai hay nhiều thiết bị điện tử trên môi trường Word Wide Web. Dịch vụ web “là một hệ thống phần mềm được thiết kế để hỗ trợ giao tiếp máy với máy trên mạng”, nó cung cấp thông tin thô, khó hiểu với người dùng, chính vì vậy được sử dụng bởi tầng ứng dụng. Các ứng dụng sẽ chế biến dữ liệu thô trước khi trả về cho người dùng cuối. Các hệ thống khác có thể giao tiếp với Web Service bằng cách sử dụng tín hiệu SOAP, kết hợp với HTTP và XML hoặc JSON. Web Service là nền tảng trung gian và thường dưới dạng text, có thể phát triển, chạy và truy cập dựa trên công nghệ phức tạp.

**Phân loại theo cách sử dụng:** Web service sử dụng máy chủ cố định và được tiêu thụ bởi máy trạm gọi là Station Web Service, trong khi đó các dịch vụ được tiêu thụ bởi thiết bị mobile gọi là Mobile Web Service.

Mobile Web Service được triển khai trên thiết bị di động qua môi trường mạng không dây đã giải quyết được những thách thức lớn về hạn chế tài nguyên và đặc tính truy cập băng thông không dây. Những thách thức này tổng kết lại như sau:

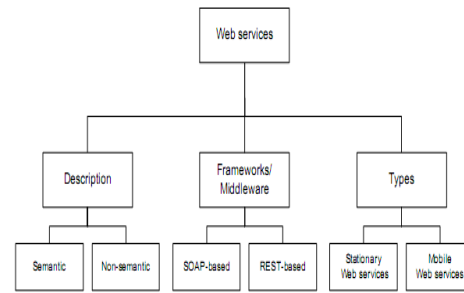
Hạn chế tài nguyên: Hạn chế màn hình hiển thị

Kết nối gián đoạn: Các thiết bị di động thường xuyên thay đổi thao tác thực hiện trên mạng và kết quả là bất kỳ dịch vụ nào cũng có thể trở nên tạm thời không tiếp cận được. Điều này đặt ra một thách thức lớn cho việc cung cấp các dịch vụ web tin cậy trong môi trường mạng không dây thay đổi liên tục.

Tính địa chỉ: Thiết bị di động luôn thay đổi điểm gắn kết vào mạng khi chúng định vị lại.

Khả năng mở rộng: Mobile Web Service không thực hiện tốt trong trường hợp có sự tương tranh truy cập dịch vụ web.

Tài nguyên không đồng nhất: Việc cung cấp một dịch vụ web mobile thực hiện độc lập và xuyên suốt là một vấn đề phức tạp.



Hình 2. Tổng quan Web service

**Khái niệm 2.2**

JSON (Java Script Object Notation) [3]. Trước đây, XML được sử dụng rộng rãi cho việc truyền dữ liệu trong web service. Tuy nhiên sử dụng XML để tải trọng dữ liệu xảy ra các vấn đề sau: biểu diễn XML hơi nặng khi xử lý. Việc sử dụng các thẻ TAG đóng mở quá nhiều, không cần thiết, làm nặng quá trình tải nạp trang. XML khó phân tích cú pháp. Để phân tích tài liệu XML sử dụng hai cách chính: DOM (Document Object Model) và SAX (Simple API for XML), tuy nhiên trên mobile sử dụng DOM và SAX không thuận tiện vì cả hai phương pháp này sử dụng tính toán và cần nhiều bộ nhớ. Trong những năm gần đây, một định dạng trao đổi dữ liệu khác đã được đưa vào sử dụng phổ biến, đó là JSON (Java Script Object Notation). JSON có đặc tính nhẹ, sử dụng text để mô tả, trao đổi dữ liệu độc lập ngôn ngữ. JSON định nghĩa một tập hợp các luật có cấu trúc cho việc biểu diễn dữ liệu trên di động. Tương tự XML, JSON là chuẩn mở sử dụng text-based để biểu diễn dữ liệu và sử dụng các kí tự như cặp ngoặc "[{}]", dấu ":" và dấu ",". Dữ liệu được biểu diễn sử dụng các cặp key/value.

```

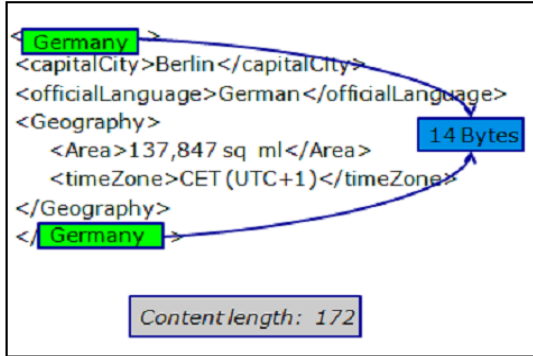
{
  "Germany": {
    "capitalCity": "Berlin",
    "officialLanguage": "German",
    "Geography": {
      "Area": "137,847 sq ml",
      "timeZone": "CET (UTC+1)"
    }
  }
}
    
```

Germany " : 7 Bytes

Content length: 123

Hình 3. Định dạng JSON

Cấu trúc của JSON: là một định dạng thông điệp trộn lẫn, chứa nhiều cấu trúc dữ liệu khác nhau như: integer, objects, boolean.



Hình 4. Định dạng XML

Qua hai hình cho thấy định dạng dữ liệu JSON có chiều dài nội dung nhỏ hơn 28.5% so với XML. Lý do của sự sai khác này do lặp lại nội dung trong mô tả XML, ví dụ cần thêm 7 byte để mô tả từ “Germany”. Điều này gây tiêu tốn bộ nhớ, tăng thời gian xử lý, và dĩ nhiên không phù hợp với thiết bị di động hạn chế tài nguyên.

**Khái niệm 2.3** Web service tuân thủ kiến trúc SOAP và giao thức SOAP gọi là SOAP Based Web Service. Web service xây dựng trên nguyên lý REST gọi là Restful Web Service. So sánh giữa hai công nghệ này như sau:

**SOAP based:** Được thiết kế để thực hiện các tương tác với các hệ thống từ xa. Trong thiết kế này các nhà cung cấp dịch vụ và khách hàng cần thiết lập một sự hiểu biết chung của cú pháp dịch vụ và các thao tác thực hiện. Mỗi web service soap based có giao diện riêng và được mô tả bởi Web Services Description Language (WSDL). Sử dụng XML để truyền thông điệp. Điểm mạnh của giao thức truyền thông điệp này là có khả năng làm việc trong môi trường không đồng nhất và độc lập thiết bị. Tuy nhiên tồn tại một số hạn chế:

**Phức tạp:** Để triển khai dịch vụ SOAPbased cần nhiều kinh nghiệm do độ phức tạp giao thức. Thêm vào đó các yêu cầu được viết bằng ngôn ngữ tự nhiên gây mất thời gian, không phù hợp với sự hạn chế tài nguyên của thiết bị di động.

**Khả năng truy cập, giao diện hạn chế**

**Khả năng tương tác kém:** Mỗi web service có giao diện riêng.

**REST based:** RESTful Web Service (RESTful web API) là web API sử dụng HTTP xây dựng trên kiến trúc REST. Restful web service thu hút được sự quan tâm từ cộng đồng web do đơn giản, linh hoạt và khả năng tương tác cao. Các dịch vụ thiết kế theo hướng này khai thác tài nguyên web qua địa chỉ URI duy nhất. Người dùng truy cập tài nguyên thông qua URI. Hướng tiếp cận này phù hợp với các thiết bị di động, nó tách biệt giao diện người dùng với lưu trữ dữ liệu đồng thời cải tiến sự linh hoạt của giao diện trên các nền tảng khác nhau và đơn giản hóa các thành phần server bằng cách chuyển chúng thành phi trạng thái.

RESTful Web Service nhẹ, dễ mở rộng và bảo trì, đưa ra tài nguyên thông qua URI và sử dụng 4 phương thức của HTTP để tạo, khôi phục, cập nhật, xóa tài nguyên. [4]

Một số lợi ích của công nghệ này như sau:

**Tính tương tác:** RESTful kế thừa tính tương tác của HTTP.

**Tính định địa chỉ:** Mỗi nguồn tài nguyên có địa chỉ URI duy nhất và người dùng có thể lấy thông qua kết nối giữa các tài nguyên.

**Phi trạng thái:** Những yêu cầu trong tiếp cận RESTful là độc lập.

Bảng 1. So sánh giữa SOAP-based và REST-based

Đặc tính	SOAP based	REST based
Kiến trúc	Hướng dịch vụ	Hướng tài nguyên
Giao thức	Bất kỳ	HTTP
Mô tả	WSDL	Không theo chuẩn
Mô hình dữ liệu	Ẩn	Hiện
Biểu diễn dữ liệu	XML	XML, JSON
An toàn	Dựa trên dịch vụ web	Dựa trên HTTP

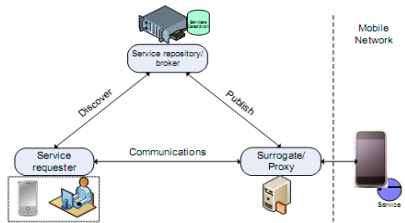
**Giao diện duy nhất:** Hai máy khách HTTP bất kỳ có thể giao tiếp trực tiếp với nhau mà không cần cấu hình đặc biệt nào. Trái lại

SOAP cần cả hai bên server và client phải đồng ý và thống nhất chung một số phương thức, kiểu dữ liệu và mô hình định địa chỉ.

**KIẾN TRÚC MOBILE WEB SERVICE [2]**

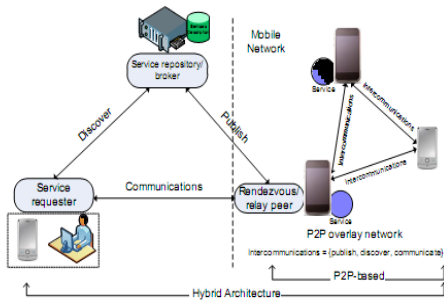
**Kiến trúc Proxy-based**

Đây là kiến trúc dễ tiếp cận nhất, tránh được nhiều thách thức về việc đối mặt với sự thực thi web service trong điều kiện ràng buộc tài nguyên. Proxy thường là một máy high-end được gắn cố định với một mạng. Từ đó về lý thuyết nó có băng thông không giới hạn để giảm thiểu việc sử dụng băng thông trong thiết bị di động, đủ năng lực xử lý để giảm tải các thiết bị hạn chế tài nguyên và thực hiện quá trình sử dụng tài nguyên chuyên sâu.



**Hình 5. Kiến trúc proxy-based**

Đồng thời phục vụ di động trong trường hợp ngắt kết nối và đáp ứng nhu cầu truy cập cao trong khi vẫn duy trì độ trễ hợp lý. Trong kiến trúc này chứa thiết bị mobile hosting Web service kết nối với máy high-end đóng vai trò như một proxy. Proxy biểu diễn điểm cuối của web service tới máy khách. Kiến trúc này giải quyết nhiều vấn đề thách thức liên quan đến mobile web service, ví dụ: Proxy có khả năng lưu trữ lớn, phục vụ một lượng lớn các máy client liên tục. Nó cũng giải quyết tính không đồng nhất của các thiết bị di động khác nhau.



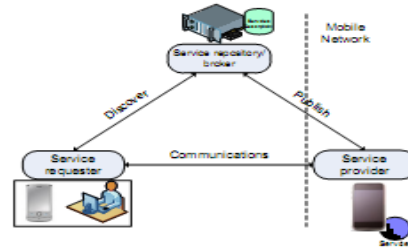
**Hình 6. Kiến trúc P2P**

**Kiến trúc P2P**

Đây là kiến trúc phân tán. Kiến trúc này dựa trên cơ chế quảng cáo mạng P2P-Peer to peer để xuất bản và khai thác web service. Cơ chế này quản lý tính di động của các node, quản lý vị trí và ràng buộc thông tin của web service.

**Kiến trúc bất đối xứng**

Kiến trúc này được thiết kế đặc biệt với mục đích tương tác các dịch vụ mobile có chu kỳ sống dài, cho phép các dịch vụ chạy không đồng bộ và độc lập. Trong kiến trúc này client yêu cầu dịch vụ và chờ phản hồi mà không khóa thiết bị trong suốt thời gian thực hiện.



**Hình 7. Kiến trúc bất đối xứng**

**XÂY DỰNG HỆ THỐNG WEB KHOA CÔNG NGHỆ THÔNG TIN TRÊN NỀN TẢNG ANDROID**

**Xây dựng hệ thống**

Xuất phát từ nhu cầu thực tế trường Đại học Kinh tế Kỹ thuật Công nghiệp đang duy trì hệ thống website, lưu trữ thông tin về trường, thông tin tuyển sinh, thông tin sinh viên... Vào những thời kỳ cao điểm sinh viên cần xem lịch thi, xem điểm, lượng truy cập trang web quá đông gây ra quá tải.

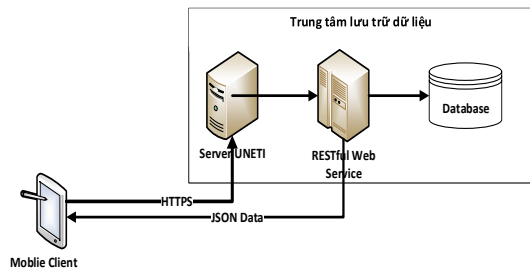
Ý tưởng: Xây dựng hệ thống web trên điện thoại di động, song song với web của trường.

Trong 3 kiến trúc mobile web service trình bày trên, chúng tôi lựa chọn kiến trúc proxy-based, đưa vào xây dựng hệ thống mobile web phù hợp với hiện trạng cơ sở vật chất tại khoa CNTT. Đề xuất giải pháp tích hợp module Restful web service trên kiến trúc proxy-based, nhận tham số từ thiết bị di động, thực hiện yêu cầu dịch vụ và trả về kết quả cho thiết bị di động dưới dạng dữ liệu JSON,

loại bỏ quá trình xử lý XML nặng trên thiết bị di động, hệ thống luôn đảm bảo yêu cầu truy cập thông tin liên tục trong điều kiện số lượng người truy cập lớn. Hệ thống sử dụng lập trình android studio 2.3 triển khai phía client, phía server tích hợp module RESTful web service, module này lấy thông tin từ web nhà trường, lưu cơ sở dữ liệu vào máy chủ đặt tại khoa CNTT.

**Thực nghiệm và kết quả**

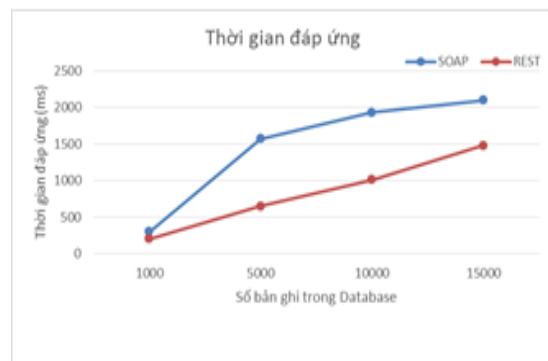
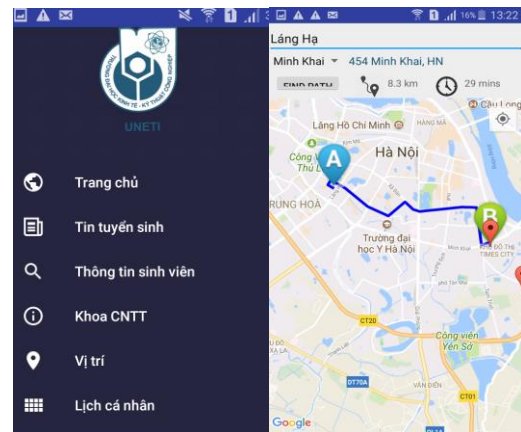
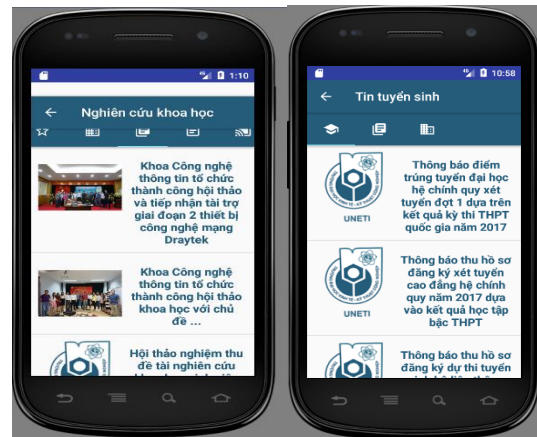
Chúng tôi xây dựng hoàn chỉnh hệ thống web khoa CNTT trên điện thoại android với một số chức năng chính sau: Giới thiệu trường, khoa, thông tin tuyển sinh, định vị vị trí kết hợp chỉ đường, tìm khoảng cách ngắn nhất từ vị trí hiện tại đến trường. Hệ thống có thêm chức năng tra cứu thông tin sinh viên, chức năng này hỗ trợ sinh viên, phụ huynh có thể biết điểm, tình trạng học tập của con em mình tại trường.



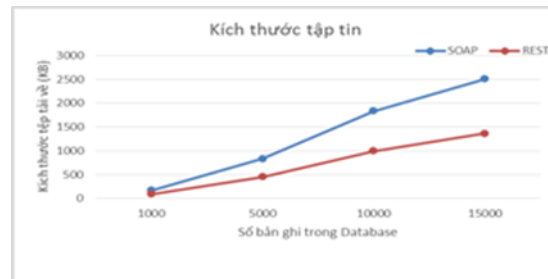
**Hình 8. Mô hình hệ thống**

Người dùng sử dụng thiết bị android, thực hiện kết nối không dây truy cập đến máy chủ đặt tại khoa CNTT, trên máy chủ này tích hợp module RESTful web service lấy thông tin trực tiếp từ web site nhà trường. Việc sử dụng máy chủ với cấu hình cao, khả năng lưu trữ lớn, phục vụ một lượng lớn người dùng truy cập web nhanh và hiệu quả, dữ liệu được cập nhật liên tục.

So sánh sự thực hiện của hệ thống web di động được tích hợp Restful Web Service với hệ thống sử dụng SOAP, dựa trên thang đo thời gian tính từ khi yêu cầu đến khi dữ liệu phản hồi gọi tắt là “thời gian đáp ứng” và kích thước tệp tải về cho kết quả sau:



**Hình 9. Thời gian đáp ứng giữa REST/SOAP**



**Hình 10. Kích thước file tải về giữa REST/SOAP**

Đồ thị trên cho thấy khi sử dụng mobile để truy cập dữ liệu, với yêu cầu một lượng dữ liệu như nhau, REST cần ít thời gian để phản hồi dữ liệu hơn SOAP. Điều đó có nghĩa là hệ thống sử dụng REST phản hồi thông tin nhanh hơn SOAP. Đồng thời cùng tải một lượng bản ghi từ cơ sở dữ liệu, nhưng REST trả về file có kích thước file nhỏ hơn SOAP, điều này làm giảm không gian lưu trữ và hoàn toàn phù hợp thiết bị di động.

#### KẾT LUẬN

Qua nghiên cứu về kiến trúc mobile web service, các vấn đề xử lý dữ liệu trong điều kiện ràng buộc tài nguyên, chúng tôi đề xuất mô hình hệ thống, tích hợp module Restful Web Service phía server, kết hợp xử lý dữ liệu JSON phía client, xây dựng hoàn chỉnh ứng dụng web thông tin trong môi trường các thiết bị android với các lợi ích sau: hệ thống truy cập thông tin nhanh, giảm không gian lưu trữ, tiện lợi: biết được đầy đủ thông tin về khoa, trường, thông tin sinh viên, thông tin tuyển sinh, thông tin này luôn được cập nhật.

Hiện tại, khung mạng di động được đề xuất mới xây dựng một số chức năng chính: Giới thiệu trường, khoa, tra cứu thông tin sinh viên, thông tin đào tạo nổi bật, tìm đường và chỉ thực hiện trên nền tảng thiết bị android. Trong tương lai chúng tôi sẽ tích hợp một số module ứng dụng khác: như trao đổi thông tin trực tuyến, thực hiện cuộc gọi... và triển khai đa nền tảng.

#### TÀI LIỆU THAM KHẢO

1. COBÂRZAN, A. (2010). "Consuming Web Services on Mobile Platforms". *Informatica Economică*, vol. 14, pp 98-105.
2. Ei Ei Thu, T. N. (2015). "Developing Mobile Application Framework By Using RESTful Web Service with JSON Parser". *Conference: The Ninth International Conference on Genetic and Evolutionary Computing, ICGEC\_2015At: Myanmar*, pp 1-7.
3. Murtaza, S. (2011). "Implementation and Evaluation of a JSON Binding for Mobile Web Services with IMS Integration Support". *XR-EE-LCN*, pp 1-55.
4. Khalid Elgazzar, P. M. (2014). "Mobile Web Services: State of the Art and Challenges". *International Journal of Advanced Computer Science and Applications*, pp 1-16.

#### ABSTRACT

#### INTERGRATING RESTFUL WEB SERVICE ON MOBILE PLATFORM, BUILDING WEB SYSTEM OF INFORMATION TECHNOLOGY FACULTY ON ANDROID DEVICE

Luong Thi Thao Hieu\*, Le Thanh Cua, Dao Phuong Anh  
*University of Economic and Technical Industry*

Now a day, mobile devices, specially devices using Android were commonly recognized as Web consumers. However, when building web system on mobile have faced to many problem: resource constraints, display screen, storage capacity, using wire less connection....So how to caching data, store and represent on mobile, building mobile information system is complicate and difficult implementation. Web service can be built by using two separate ways: standard SOAP based and RESTful web service. In this paper, research some mobile web service architect, we intergrate RESTful web service with mobile application to make mobile web service, then building web system of UNETI's information technology faculty on mobile device.

**Keyword:** ANDROID; mobile web service; RESTful; JSON

Ngày nhận bài: 19/6/2018; Ngày phản biện: 18/7/2018; Ngày duyệt đăng: 31/8/2018

\* Tel: 0942 160880, Email: ltthieu@uneti.edu.vn