

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



THỰC TẬP TỐT NGHIỆP

Đề tài 126

Xây dựng nền tảng và tiêu chuẩn
tương tác 2 chiều giữa chính phủ điện tử
với thiết bị di động của người dân
dựa trên công nghệ Web

GVHD: Thầy: Lương Thế Nhân

Thành viên: Võ Văn Luận 51202065
Trần Văn Tài 51203241

TP. HỒ CHÍ MINH, THÁNG 5/2016



Tóm tắt

Bài báo cáo này trình bày quá trình nghiên cứu và hiện thực đề tài "Xây dựng nền tảng và tiêu chuẩn tương tác 2 chiều giữa chính phủ điện tử với thiết bị di động củaa người dân dựa trên công nghệ Web. Mục tiêu đề tài bao gồm:

- Tìm hiểu các công nghệ Web và ngôn ngữ lập trình cho thiết bị di động
- Tìm hiểu và xây dựng web service theo chuẩn Open311 cho quy trình trên



Mục lục

1 Giới thiệu đề tài	3
1.1 Tổng quan về chính phủ điện tử	3
1.2 Chi tiết về đề tài	5
2 Các dự án tương tự	5
2.1 Trên thế giới	5
2.1.1 New York	5
2.1.2 Singapore	8
2.2 Tại Việt Nam	9
2.2.1 Thành phố Hồ Chí Minh	9
2.2.2 Thành phố Đà Nẵng	10
3 Kiến thức nền	12
3.1 Open311	12
3.1.1 Giới thiệu và định nghĩa	12
3.1.2 GeoReport v2	13
3.2 RESTful Web Service	15
3.2.1 Sử dụng các phương thức HTTP một cách rõ ràng	15
3.2.2 Phi trạng thái	15
3.2.3 URIs hiển thị theo dạng cấu trúc thư mục	16
3.2.4 Biểu diễn tài nguyên	16
3.3 JPA	17
3.4 AngularJS	18
3.4.1 Giới thiệu về AngularJS	18
3.4.2 Các đặc tính của AngularJS	18
3.4.3 Các tính năng cốt lõi của AngularJS	19
3.4.4 Các thành phần của AngularJS	20
3.4.5 Ưu và nhược điểm	20
3.5 Android	21
3.5.1 Giới thiệu về Android	21
3.5.2 Nguyên nhân sự phát triển mạnh mẽ của Android	22
3.5.3 Các tính năng cơ bản của Android	23
4 Hiện thực dự án	24
4.1 Cơ sở dữ liệu	24
4.2 Web Service	25
4.3 Client	26
5 Khó khăn và khắc phục	26
5.1 Xác thực và phân quyền	26
6 Mở rộng đề tài	26
7 Kết luận	27
8 Phụ lục	28



1 Giới thiệu đề tài

1.1 Tổng quan về chính phủ điện tử

Chính phủ điện tử (CPDT) không có một định nghĩa cụ thể nào; trên thế giới có rất nhiều khái niệm của các tổ chức khác nhau về chính phủ điện tử. Sau đây là một số định nghĩa được sử dụng phổ biến:

- **Theo Liên Hợp Quốc:** Chính phủ điện tử được định nghĩa là việc sử dụng Internet và mạng toàn cầu (world-wide-web) để cung cấp thông tin và các dịch vụ của chính phủ tới công dân [1]
- **Theo World Bank:** Chính phủ điện tử đề cập đến việc các cơ quan chính phủ sử dụng các công nghệ thông tin (chẳng hạn như mạng diện rộng, mạng Internet và mạng di động) mà có khả năng chuyển đổi mối quan hệ với công dân, doanh nghiệp và với các cơ quan chính phủ khác. Những công nghệ này có thể phục vụ cho những mục đích khác nhau: cung cấp các dịch vụ tốt hơn, cải thiện sự tương tác với doanh nghiệp, tăng cường quyền lực cho công dân thông qua việc truy cập thông tin hoặc quản lý chính phủ hiệu quả hơn. Các lợi ích mang lại có thể giảm tham nhũng, nâng cao sự minh bạch, thuận tiện hơn, tăng doanh thu và/hoặc giảm chi phí [2]

Theo cách hiểu của nhóm, chính phủ điện tử là việc áp dụng công nghệ thông tin tiên tiến vào việc giao tiếp giữa chính phủ và công dân của mình, đem lại sự nhanh chóng, tiện lợi và minh bạch trong việc cung cấp thông tin và dịch vụ đến công dân.

Việc hiện thực hóa chính phủ điện tử đem đến rất nhiều lợi ích cho cả chính phủ lẫn công dân; nhất là trong bối cảnh hội nhập hiện nay, Internet và di động đã được phổ biến rộng rãi trong các tầng lớp người dân, hầu hết mọi người đều có một lượng kiến thức nhất định về công nghệ thông tin, giúp cho việc sử dụng và thao tác trên ứng dụng trở nên dễ dàng hơn. Trong đó có thể kể đến các lợi ích sau đây:

- Việc thông tin giữa chính phủ và công dân diễn ra một cách nhanh chóng và chính xác, giúp chính phủ có thể ban hành những quyết định kịp thời, giúp công dân nắm được tình hình an ninh, xã hội một cách chi tiết
- Giảm thiểu thời gian, khối lượng công việc và chi phí của các quy trình, thủ tục hành chính, giúp việc quản lý, hoạt động của chính phủ nhanh hơn, hiệu quả hơn
- Về phía các tổ chức, doanh nghiệp, chính phủ điện tử giúp họ làm việc một cách dễ dàng hơn bởi mọi thủ tục, quy trình đều được "số hóa", mỗi công việc được thực hiện một cách chính xác, tin cậy. Nhờ đó, tổ chức và doanh nghiệp hoạt động hiệu quả hơn, nâng cao năng lực cạnh tranh

Tuy nhiên, việc triển khai CPDT vào thực tiễn còn gặp nhiều khó khăn và thiếu sót. Năm 2014, theo Bảng xếp hạng chính phủ điện tử (E-Government Development Index) của Liên Hợp Quốc [4], Việt Nam có chỉ số phát triển CPDT là 0.4704 (đứng hạng thứ 99 trong 193 quốc gia), tuột 16 bậc so với năm 2012 (đứng hạng 83) (Hình 1). Điều này cho thấy rằng tốc độ phát triển và phổ cập chính phủ điện tử ở Việt Nam vẫn còn chậm so với thế giới và các nước trong khu vực.

Những khó khăn và trở ngại trong quá trình xây dựng CPDT ở Việt Nam có thể kể đến sau đây:

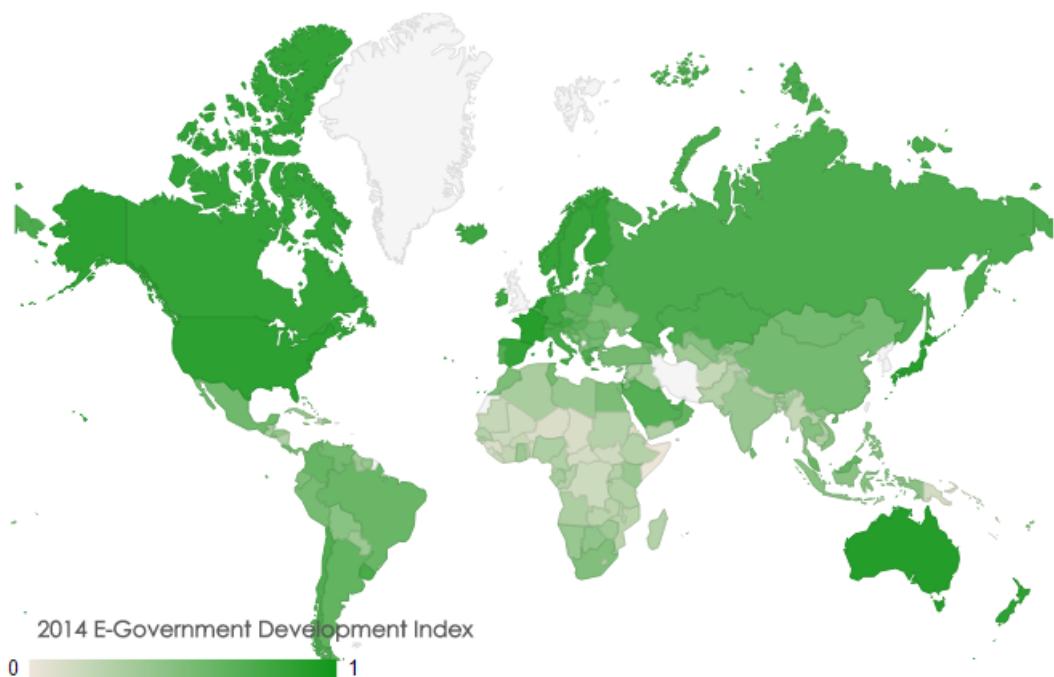


Compare 2014 with: 2012 ▾

Country	Rank 2012	Rank 2014	EGDI 2014	Rank Change
Viet Nam	83	99	0.4705	-16

Hình 1: Chỉ số phát triển CPDT của Việt Nam năm 2014

- Cơ sở hạ tầng công nghệ thông tin ở nước ta còn yếu kém, bất cập từ các dự án công nghệ thông tin
- Trình độ dân trí còn thấp, khó khăn trong việc sử dụng các dịch vụ của công nghệ thông tin (Web, di động,...)
- Trình độ nhận thức và kỹ năng của cán bộ viên chức còn hạn chế, tâm lý không muốn chuyển đổi sang môi trường làm việc mới
- Quy trình nghiệp vụ vẫn chưa ổn định, gây khó khăn cho việc xây dựng một hệ thống toàn diện



Hình 2: Chỉ số phát triển CPDT các quốc gia năm 2014

Hiện nay, các dự án CPDT đang được đẩy mạnh, nhiều cuộc tọa đàm với chủ đề công nghệ thông tin được tổ chức thường xuyên, cho thấy chính phủ ngày càng dành nhiều sự quan tâm đến việc công nghệ hóa các dịch vụ công. Ngày 14/10/2015, Chính phủ ban hành Nghị quyết số



36a/NQ-CP về Chính phủ điện tử, trong đó có một mục tiêu quan trọng là "*Phấn đấu đến hết năm 2016, các bộ, ngành Trung ương có 100% các dịch vụ công được cung cấp trực tuyến ở mức độ cho phép người sử dụng điền và gửi trực tuyến các mẫu văn bản đến cơ quan, tổ chức cung cấp dịch vụ*". Điều đó đã nêu lên được tầm quan trọng của CPDT trong việc phát triển kinh tế, xã hội quốc gia.

1.2 Chi tiết về đề tài

Nhiệm vụ chủ yếu của đề tài là xây dựng kênh tương tác 2 chiều giữa chính phủ và người dân tại địa bàn Thành phố Hồ Chí Minh, dựa trên 2 nền tảng là Web và di động. Trong đó, tương tác 2 chiều trong phạm vi của đề tài có nghĩa là:

- Ứng dụng cho phép người dân phản ánh các sự cố (điện, nước, giao thông,...) cho các cơ quan chức năng để họ có thể khắc phục kịp thời; duyệt xem và bình luận các phản ánh trước đó; xem các thông báo mới của chính phủ
- Chính phủ có thể thông báo đến người dân các tin tức về đời sống, xã hội (lịch cúp điện, công trình giao thông đang thi công,...); thu thập góp ý từ phía công dân, đưa mỗi phản ánh về các cơ quan chức năng chuyên biệt để giải quyết vấn đề, phản hồi nhanh chóng cho người dân

Môi trường làm việc:

- Hệ quản trị cơ sở dữ liệu: MySQL Server
- Ngôn ngữ lập trình: Java, AngularJS, HTML, CSS, Javascript
- Công cụ lập trình: NetBeans, Android Studio, Sublime Text
- Hệ thống quản lý mã nguồn: GitHub

2 Các dự án tương tự

2.1 Trên thế giới

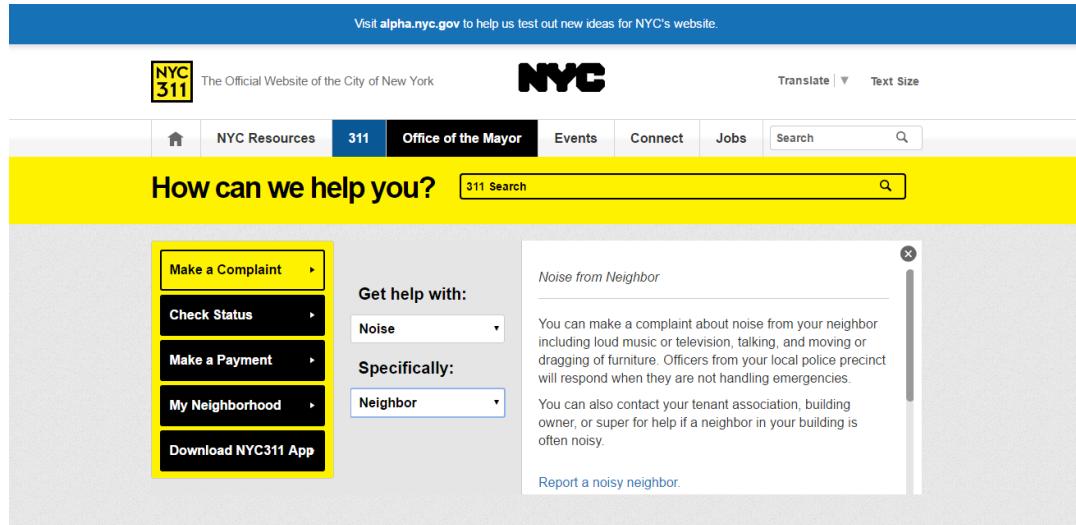
2.1.1 New York

[Nyc.gov](#) là tên miền trang web CPDT của thành phố New York. Hiện có hơn 30,000 trang nội dung và 100 dịch vụ giao dịch online trên trang web này. Giao dịch online bao gồm: mua vé đậu xe, theo dõi tình trạng bất động sản, nhận các chứng chỉ hoặc giấy phép từ chính phủ, hoặc đơn giản hơn là xem lịch đổ rác ở khu phố đang sống... Cổng CPDT của thành phố cập nhật các thông tin khẩn cấp và thông báo đến toàn thể người dân thành phố rất dễ dàng.

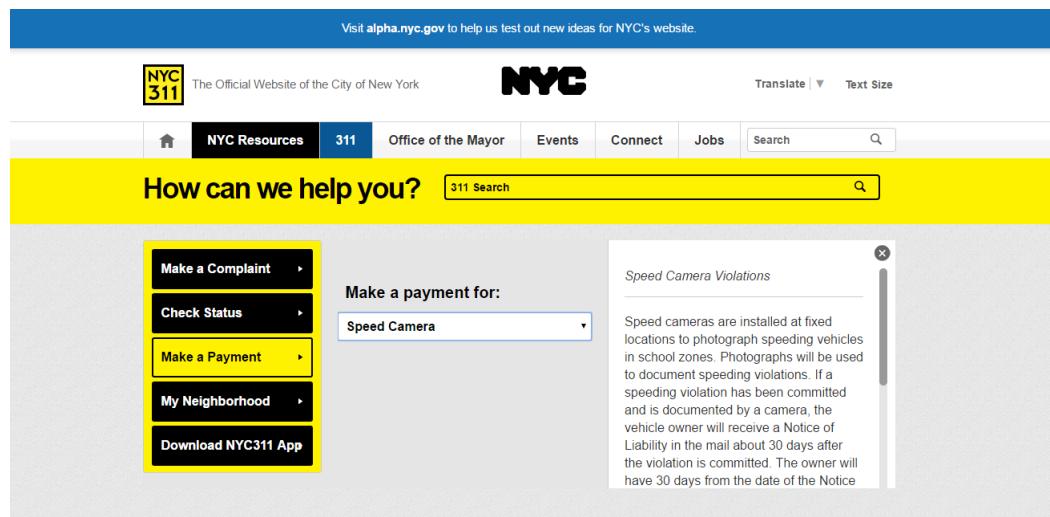


Hình 3: Trang web CPDT [Nyc.gov](#) của thành phố New York

Người dân có thể dễ dàng gửi phản ánh về vấn đề xung quanh như: tiếng ồn, đậu xe không đúng chỗ, xả rác, mất trật tự an ninh... đến chính quyền thành phố từ trang web <http://www1.nyc.gov/311>. Chính quyền sẽ phản hồi và nhanh chóng giải quyết vấn đề. Ngoài ra, người dân cũng có thể thanh toán các dịch vụ công cộng như: vé đậu xe, vé phạt tốc độ, vượt đèn đỏ,... trên trang web này.



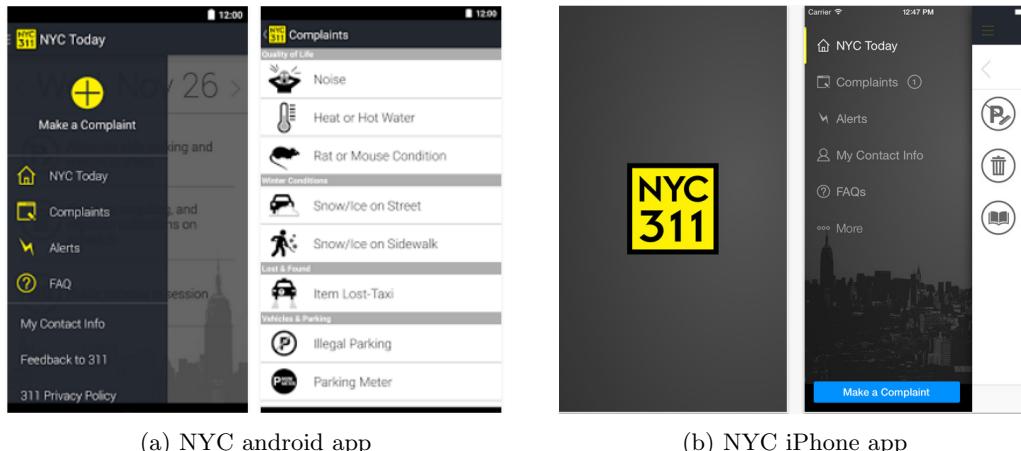
Hình 4: Gửi lời phàn nàn đến chính quyền



Hình 5: Người dân thanh toán phí công cộng

Không chỉ phát triển trên nền tảng web, CPDT ở New York còn phát triển trên các ứng dụng điện thoại di động, đặc biệt là trên hệ điều hành Android và iPhone (Hình 6). Với sự tiện ích và gắn bó bên cạnh người dùng của điện thoại di động, người dân có thể gửi yêu cầu đến chính quyền thành phố nhanh chóng và kịp thời.

Từ sau sự kiện chấn động 11/9, thành phố New York càng tập trung phát triển CPDT để tạo ra kênh thông tin cho người dân thành phố trong các tình huống khẩn cấp. Hiện nay, New York là một trong các thành phố phát triển nhất thế giới về CPDT.



Hình 6: Ứng dụng CPDT của New York trên hệ điều hành Android và iPhone

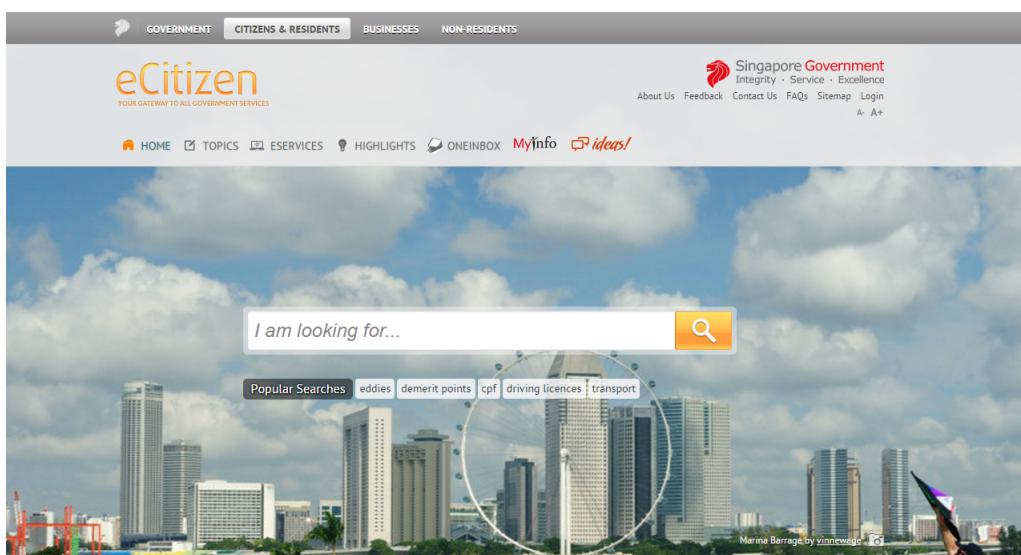


2.1.2 Singapore

Là một trong những quốc gia tiên phong về CPDT, Singapore bắt tay vào CPDT với các kế hoạch CNTT tổng thể ở cấp quốc gia và tin học hóa quản lý hành chính kể từ những năm 80 của thế kỷ 20. Singapore nổi tiếng về mô hình "một cửa" trong dịch vụ công và đã triển khai thành công CPDT. Singapore rất nỗ lực trong việc đầu tư phát triển CPDT và đã đạt được nhiều thành tựu quan trọng, xếp hạng cao trong các đánh giá toàn cầu về CPDT:

- Trong cuộc khảo sát của Liên Hiệp Quốc về CPDT:
 - Đứng thứ 2 về chỉ số tham gia CPDT
 - Đứng thứ 10 về chỉ số phát triển CPDT
- Đứng thứ nhất trong cuộc khảo sát hằng năm về CPDT của trường đại học Waseda

CPDT ở Singapore nhận được sự tin tưởng của người dân từ khi được triển khai. 93% người dân hài lòng và cảm thấy có ích về những thông tin được đăng trên trang web của chính phủ, 94% bằng lòng với chất lượng dịch vụ CPDT. Hiện nay, có 4 cổng CPDT và hơn 400 websites chi nhánh giải quyết từng vấn đề cụ thể ở Singapore. Không chỉ vậy, <https://www.ecitizen.gov.sg> là cổng CPDT riêng để người dân có thể truy cập các thông tin và dịch vụ điện tử của chính phủ.



Hình 7: Trang web CPDT [ecitizen.gov.sg](https://www.ecitizen.gov.sg) của thành phố Singapore

Người dân dễ dàng truy cập vào tất cả thông tin và dịch vụ chính phủ. Đồng thời, trang [ecitizen.gov.sg](https://www.ecitizen.gov.sg) cũng liên kết đến các websites chi nhánh để cung cấp chi tiết thông tin hay dịch vụ người dân cần. Ngoài ra, CPDT ở Singapore còn có hệ thống cung cấp giấy phép, chứng chỉ kinh doanh điện tử, mang lại nhiều lợi ích về thời gian và chi phí:

- Tự động cập nhật tình trạng qua tin nhắn hoặc email
- Giảm 90% thời gian làm thủ tục
- Giảm 50% thời gian nhập dữ liệu



2.2 Tại Việt Nam

2.2.1 Thành phố Hồ Chí Minh

Trang web hochiminhcity.gov.vn là cổng thông tin chính phủ điện tử ở tại Thành phố Hồ Chí Minh. Trang web với hơn 7000 lượt truy cập mỗi ngày, cung cấp các tin tức, hoạt động nổi bật của thành phố, các văn bản chỉ đạo điều hành, văn bản quy phạm pháp luật, các báo cáo thống kê và tình hình kinh tế - xã hội.

Ngoài ra trang web còn cho phép công dân gửi góp ý về các quy định, thủ tục hành chính, những vướng mắc cụ thể trong việc thực hiện các quy định hành chính; người dân cũng có thể đề xuất với chính quyền các phương án xử lý những phản ánh nêu trên hoặc có sáng kiến ban hành mới quy định hành chính liên quan đến hoạt động kinh doanh, đời sống nhân dân.



Hình 8: Trang web CPDT hochiminhcity.gov.vn của thành phố Hồ Chí Minh

Tuy nhiên, trang web vẫn còn nhiều bất cập và thiếu sót, trang góp ý (Hình 9) còn đơn giản, chưa đi sâu vào chi tiết các mảng, chưa phân hóa ra thành các mục chuyên biệt rõ ràng; trang web không hiển thị cho người xem các phản ánh trước đó, do đó người dân không biết được tình trạng cũng như các cơ quan chức năng nào đang xử lý phản ánh, góp ý, không tạo được kênh tương tác giữa chính phủ và công dân.



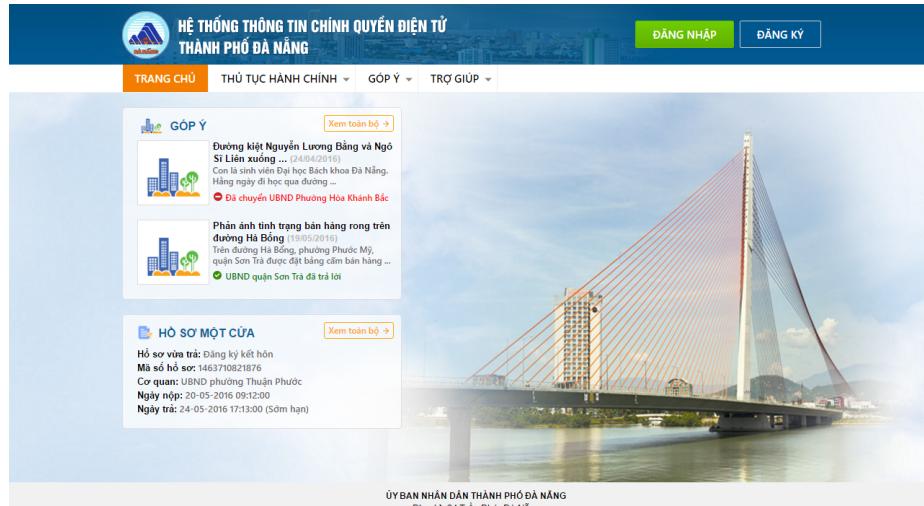
The screenshot shows a web page with a blue header bar. On the left, there's a sidebar with links like 'GIỚI THIỆU', 'HỆ THỐNG VĂN BẢN', 'LẤY Ý KIẾN GÓP Y', and 'BẢO CÁO, THỐNG KÊ'. The main content area has a form titled 'THÔNG TIN GÓP Y' with fields for 'Họ và tên: (*)', 'Email: (*)', 'Chủ đề: (*)', 'Địa chỉ: (*)', 'Số điện thoại: (*)', 'Nội dung: (*)', 'File đính kèm: Choose File [No file chosen]', and 'Mã kiểm tra:'. Below the form is a button 'Gửi Góp ý | Lần sau'. To the right, there are two columns: 'BẢN ĐỒ THÀNH PHỐ' showing a map of HCMC districts and 'CHỈ ĐẠO, ĐIỀU HÀNH' with links to maps and administrative information. At the top right, there are language and search buttons.

Hình 9: Trang góp ý của hochiminhcity.gov.vn

2.2.2 Thành phố Đà Nẵng

Vào tháng 8/2014, ông Phạm Kim Sơn, Giám đốc Sở Thông tin & Truyền thông công bố chia sẻ miễn phí nền tảng Chính phủ điện tử **eGov Framework** cho các địa phương. **eGov Framework** là giải pháp nền tảng về Chính phủ điện tử được chuyển giao từ Chính phủ Hàn Quốc.

Cùng với sự tư vấn của các chuyên gia ở Hoa Kỳ, Đà Nẵng đã phối hợp phát triển thêm nền tảng Chính phủ điện tử phù hợp với điều kiện địa phương trong suốt 2 năm với chi phí bỏ ra khoảng 630.000 USD, cho ra đời hệ thống thông tin chính quyền điện tử Thành phố Đà Nẵng tại đường dẫn egov.danang.gov.vn.



Hình 10: Trang web CPDT egov.danang.gov.vn của Thành phố Đà Nẵng



Trang web cho phép công dân, doanh nghiệp sau khi đăng nhập bằng tài khoản đã được đăng ký và xác thực bởi cơ quan nhà nước có thể thực hiện các dịch vụ hành chính công một cách đơn giản và thuận tiện. Công dân, doanh nghiệp quản lý được hồ sơ cá nhân (thông tin cá nhân, giấy tờ số) và toàn bộ các giao dịch với cơ quan nhà nước thông qua việc nộp, theo dõi tình hình xử lý và nhận kết quả xử lý hồ sơ thực hiện dịch vụ hành chính công trực tiếp tại bộ phận một cửa hoặc trực tuyến tại Hệ thống thông tin chính quyền điện tử TP Đà Nẵng. Ngoài ra, công dân và doanh nghiệp còn dễ dàng khai thác các dịch vụ công.

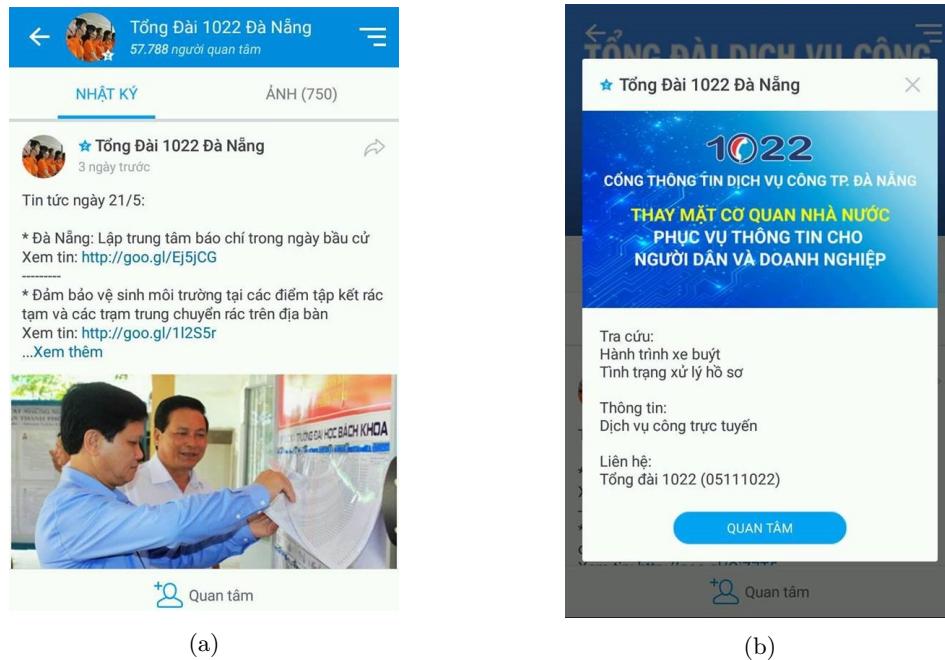
Cán bộ sử dụng Hệ thống thông tin chính quyền điện tử để tiếp nhận hồ sơ trực tuyến, tiếp nhận hồ sơ trực tiếp, luân chuyển và xử lý hồ sơ, trả kết quả hồ sơ và thực hiện thống kê, báo cáo tùy theo vai trò được phân công.

The screenshot shows a feedback form on a website. The main form includes fields for Name, Email (with an error message 'Bạn không được để trống trường này.'), Phone, Address, Content, and attachments. To the right are several sidebar modules: 'GỬI GÓP Ý PHẢN ÁNH' (Send Feedback), 'HƯỚNG DẪN SỬ DỤNG' (Usage Guide), 'THỐNG KÊ BÁO CÁO' (Statistics Report), 'Tổng đài hỗ trợ 0511 1022' (Support Hotline 0511 1022), and 'BÌNH LUẬN MỚI NHẤT' (Newest Comments) which lists a comment from 'Hoai Duc'.

Hình 11: Trang góp ý của egov.danang.gov.vn

Không những thế, chính quyền Đà Nẵng còn mở thêm cổng thông tin "Quản lý đô thị Đà Nẵng: Tiện nghi - Xanh - Sạch - Đẹp" trên mạng xã hội Facebook tiếp nhận các phản ánh của người dân về các vấn đề an ninh đô thị, môi trường, v.v

Ngoài ra, Đà Nẵng còn cung cấp "Tổng đài dịch vụ công 1022" trên mạng xã hội Zalo (Hình 12). Tổng đài chính thức đi vào hoạt động ngày 15/12/2015, cung cấp các tin tức nổi bật hàng ngày tại Đà Nẵng, cho phép người dân tra cứu hành trình xe buýt, tình trạng nộp hồ sơ đăng ký kinh doanh. Trong tương lai, Tổng đài trên Zalo còn dự kiến cung cấp nhiều loại thông tin hữu ích khác tới người dân như tình hình bão lũ và hướng dẫn phòng tránh; lịch tiếp dân; lịch cắt điện, cắt nước; văn bản, chính sách mới của thành phố, điểm thi tốt nghiệp THCS, THPT, Đại học...



Hình 12: Ứng dụng CPDT của Đà Nẵng trên Zalo

Việc sử dụng các mạng xã hội như Facebook và Zalo vào mô hình CPDT giúp cho thông tin có độ phủ lớn, đến với người dân rất nhanh, phản ứng của chính quyền kịp thời. Điều này cho thấy Đà Nẵng đã dẫn đầu xu hướng CPDT tại Việt Nam, tiên phong trong việc triển khai chính phủ điện tử trên nền tảng OTT, đón đầu cả xu thế sử dụng điện thoại di động ngày càng phổ biến trong xã hội.

3 Kiến thức nền

3.1 Open311

3.1.1 Giới thiệu và định nghĩa

Hiện nay, dựa vào sự phát triển của CNTT người dân có thể gửi những yêu cầu, nguyện vọng của bản thân đến chính quyền thông qua email. Một email báo cáo việc một miếng đá lát trên đường Nguyễn Huệ bị vỡ được gửi đến một nhân viên chính quyền. Người này sẽ lấy đoạn email và lưu vào cơ sở dữ liệu, sau đó gửi đến cho ai đó cần quan tâm giải quyết vấn đề này.

Thay vì vậy, nếu báo cáo về miếng đá bị vỡ được gửi trực tiếp từ người dân đến ban ngành quản lý việc sửa chữa miếng đá vỡ ấy, sẽ giảm thiểu rất nhiều chi phí quản lý. Đồng thời, thông báo người dân khi vấn đề của họ đã được giải quyết. Vấn đề bức thiết là phải tạo ra một giao thức để người dân có thể gửi yêu cầu dịch vụ tới chính quyền, và nhận thông báo về tiến trình giải quyết vấn đề đó. Ý tưởng này đã được nhóm kỹ sư tại OpenPlans quan tâm và họ đã xây dựng một dự án có tên Open311.

Open311 là một phương thức mà người dân có thể thông qua nó để gửi các yêu cầu dịch vụ vào



danh sách các việc cần làm của chính quyền địa phương, và là cách thức để người dân biết được tiến trình xử lý của chính quyền nhanh chóng và dễ dàng.

3.1.2 GeoReport v2

GeoReport v2 là một API của Open311 để xây dựng những ứng dụng có thể xem và báo cáo những vấn đề đến chính quyền. Một số thành phố đang sử dụng chuẩn GeoReport v2:

- Helsinki, Finland
- Bonn, Germany
- Lamia, Greece
- Chicago, Illinois
- Toronto, Ontario
- San Francisco, California
- Washington D.C
- Boston, Massachusetts
- Baltimore, Maryland
- Bloomington, Indiana
- New Haven, Connecticut

Tiếp theo, nhóm xin giới thiệu một số thông tin cơ bản về chuẩn GeoReport v2

Định dạng ngày/giờ

Mọi ô dữ liệu kiểu ngày/giờ phải định dạng theo một tập con chung của chuẩn ISO 8601. Thông tin về múi giờ phải có trong định dạng.

Ví dụ: 1994-11-05T08:15:30-05:00 nghĩa là November 5, 1994, 8:15:30 am, US Eastern Standard Time

Mã hóa

GeoReport yêu cầu sử dụng UTF-8. Mọi kí tự trả về ở dạng XML, JSON, hoặc bất kỳ một định dạng khác phải được mã hóa dưới dạng UTF-8. Những HTTP headers phải thêm thuộc tính encoding = "UTF-8". Mọi kí tự nhận được bởi dịch vụ từ client sẽ được đưa về UTF-8 và được giải mã một cách phù hợp.

Hỗ trợ định dạng

Định dạng có thể sử dụng trong GeoReport v2 là XML hoặc JSON. HTTP content-type headers của mỗi định dạng sẽ là:

XML: Content-Type: text/xml; charset=utf-8

JSON: Content-Type: application/json; charset=utf-8



Các phương thức API

- **GET Service List** cung cấp danh sách các yêu cầu dịch vụ và các mã code tương ứng của từng loại dịch vụ.
- **GET Service Definition** định nghĩa các thuộc tính tương ứng với một mã dịch vụ.
- **POST Service Request** gửi một yêu cầu dịch vụ từ client lên server.
- **GET Service Request** lấy thông tin về tiến trình hiện tại của yêu cầu: đang xem xét hoặc đã giải quyết...
- **GET Service Requests** lấy thông tin của nhiều yêu cầu dịch vụ.

Các thuộc tính cần có trong cơ sở dữ liệu

Với chuẩn GeoReport v2, một số thuộc tính sau bắt buộc phải có trong cơ sở dữ liệu:

- **jurisdiction_id** là định danh độc nhất của từng thành phố, dùng để phân biệt các thành phố trong trường hợp hiện thực cho nhiều thành phố.
- **service_code**: Mã code quy định loại dịch vụ
- **service_name**: Tên có thể đọc được của dịch vụ
- **description**: Mô tả ngắn gọn về dịch vụ, yêu cầu
- **lat**: Vĩ độ vị trí xảy ra vấn đề cần gửi báo cáo, yêu cầu lên chính quyền
- **long**: Kinh độ vị trí xảy ra vấn đề cần gửi báo cáo, yêu cầu lên chính quyền
- **address**: người dùng nhập địa chỉ, hoặc mô tả địa điểm
- **address_id**: ID của địa chỉ
- **status**: tình trạng hiện tại của yêu cầu (đang xem xét, đã giải quyết)
- **agency_responsible**: cơ quan chịu trách nhiệm giải quyết yêu cầu
- **requested_datetime**: thời điểm yêu cầu được gửi
- **updated_datetime**: thời điểm yêu cầu được cập nhật
- **expected_datetime**: dự tính yêu cầu sẽ được giải quyết xong vào thời điểm này
- **email**: địa chỉ email của người gửi yêu cầu
- **device_id**: ID độc nhất của thiết bị gửi yêu cầu
- **account_id**: ID độc nhất của tài khoản người dùng gửi yêu cầu
- **first_name**: tên của người gửi yêu cầu
- **last_name**: họ của người gửi yêu cầu
- **phone**: số điện thoại của người gửi yêu cầu
- **media_url**: đường dẫn đến ảnh, clip... mô tả về yêu cầu



3.2 RESTful Web Service

REST (Representational State Transfer) định nghĩa các quy tắc kiến trúc thiết kế Web services chú trọng vào tài nguyên hệ thống, các tài nguyên này được chuyển tải thông qua giao thức HTTP. REST được giới thiệu lần đầu tiên vào năm 2000 bởi Roy Fielding trong luận án của ông "Architectural Styles and the Design of Network-based Software Architectures".

Việc cụ thể hóa một Web service REST sẽ tuân thủ theo bốn nguyên tắc thiết kế sau:

- Sử dụng các phương thức HTTP một cách rõ ràng
- Phi trạng thái
- URIs hiển thị theo dạng cấu trúc thư mục
- Biểu diễn tài nguyên

3.2.1 Sử dụng các phương thức HTTP một cách rõ ràng

Một đặc tính quan trọng của dịch Web service RESTful là sử dụng một cách rõ ràng các phương thức HTTP theo cách một giao thức được xác định bởi RFC 2616. Ví dụ HTTP GET được xác định như là một phương thức sinh ra số liệu được sử dụng có chủ đích bởi các ứng dụng người dùng để thu thập tài nguyên, dữ liệu từ một máy chủ, hoặc thực thi một truy vấn mà máy chủ sẽ tìm kiếm và phản hồi cùng với một gói thông tin tương thích.

Dưới đây là các phương thức HTTP được sử dụng phổ biến trong kiến trúc REST:

- **GET** - truy xuất một tài nguyên
- **PUT** - thay đổi trạng thái một tài nguyên hoặc để cập nhật nó
- **POST** - tạo mới một tài nguyên trên máy chủ
- **DELETE** - huỷ bỏ hoặc xoá một tài nguyên

3.2.2 Phi trạng thái

Trong RESTful web service, server không lưu lại trạng thái của client; mỗi khi gửi một request lên server thì client phải đóng gói thông tin đầy đủ để server có thể hiểu được. Sự giới hạn này được gọi là phi trạng thái trong RESTful web service.

Lợi ích:

- Web service xử lý các yêu cầu của client một cách độc lập
- Web service không cần phải lưu lại trạng thái trước đó của client, giúp hệ thống dễ phát triển, bảo trì và mở rộng hơn
- Bản thân HTTP cũng là giao thức phi trạng thái, do đó RESTful web service sẽ hoạt động hiệu quả và ăn khớp với các giao thức của HTTP

Bất lợi: gia tăng lượng thông tin cần truyền tải giữa client và server



3.2.3 URIs hiển thị theo dạng cấu trúc thư mục

Đặc tính thứ ba về RESTful web service là về URIs. REST Web service URIs phải được thể hiện một cách trực quan và dễ đoán. Hãy nghĩ về URIs như một "*self-documenting interface*" - một dạng tài liệu trong đó cần rất ít hoặc thậm chí không cần các giải thích và tham khảo để một nhà phát triển có thể hiểu được những gì nó trả tới hoặc rút trích ra được các thông tin liên quan.

Có một cách đơn giản để có thể đạt được điều đó là thể hiện URIs theo dạng cấu trúc thư mục - trong đó URIs có thứ bậc, bắt nguồn từ một đường dẫn đơn giản, và có thể phân nhánh tại các nút con thể hiện nội dung của một dịch vụ xác định.

Một ví dụ đơn giản như một đường dẫn URIs sau đây: <http://www.myservice.org/discussion/2008/12/10/{topic}>. Phần đầu tiên trong đường dẫn là năm có bốn chữ số, phần thứ hai là ngày có hai chữ số, và phần thứ ba là tháng có hai chữ số. Có vẻ hơi rắc rối khi giải thích theo cách này, nhưng đây là cách thể hiện đơn giản, dễ hiểu và dễ đoán. Nhà phát triển có thể suy ra được ngày tháng các *topic* được đăng lên chuyên mục *discussion*. Con người và máy móc có thể dễ dàng sinh ra các cấu trúc URIs giống như vậy vì chúng dựa trên các nguyên tắc nhất định.

Một vài lưu ý khi nói đến cấu trúc địa chỉ của một REST Web service URIs:

- Giấu các đuôi tài liệu mở rộng của file (.jsp, .php, .asp), vì vậy bạn có thể chèn thêm một số thứ mà không cần thay đổi địa chỉ URLs.
- Sử dụng chữ thường cho tất cả các ký tự
- Thay thế các khoảng trắng bằng gạch chân hoặc gạch nối
- Tránh sử dụng các chuỗi query
- Thay vì sử dụng mã *404 Not Found*, luôn luôn phản hồi về tài nguyên hoặc một trang mặc định

3.2.4 Biểu diễn tài nguyên

Kiến trúc REST xem các loại nội dung như là một dạng tài nguyên. Các tài nguyên này có thể là tập tin văn bản, các trang HTML, hình ảnh, video hoặc các loại dữ liệu động. REST server đơn giản chỉ cung cấp quyền truy cập vào các tài nguyên cho REST client, client sẽ tiến hành truy cập và chỉnh sửa trực tiếp trên các tài nguyên. Một vài loại MIME thông dụng được sử dụng bởi dịch vụ RESTful được thể hiện trong bảng sau

MIME-Type	Content-Type
JSON	application/json
XML	application/xml
XHTML	application/xhtml+xml

REST của không áp đặt bất kỳ hạn chế nào về định dạng của một tài nguyên. Một client có thể yêu cầu dưới định dạng JSON, một khách hàng khác có thể yêu cầu dưới định dạng XML của cùng một tài nguyên cho server. Trách nhiệm của các REST server là trả về cho các client các tài nguyên trong các định dạng mà client hiểu.



3.3 JPA

Java Persistence API hay JPA là một đặc tả Java cho việc ánh xạ giữa các đối tượng Java tới cơ sở dữ liệu quan hệ sử dụng công nghệ phổ biến là ORM (Object Relational Mapping). JPA API cung cấp đầy đủ các công cụ cho phép người lập trình có thể tạo cơ sở dữ liệu một cách đơn giản và nhanh chóng. JPA API có thể dùng để persist một đối tượng business (POJO) vào trong cơ sở dữ liệu hoặc lấy dữ liệu từ cơ sở dữ liệu và ánh xạ ra các đối tượng business một cách đơn giản. Điều này làm giảm bớt gánh nặng viết code trong việc quản lý các đối tượng quan hệ, một lập trình viên khi sử dụng "*JPA Provider*" framework có thể dễ dàng tương tác với các instance trong cơ sở dữ liệu.

JPA là một đặc tả của Sun, ra đời cùng với bản đặc tả J2EE 5. JPA không phải là một sản phẩm và không thể nó như thành phần persistence. Nó cần phải có một cài đặt ORM để hoạt động và persist các đối tượng Java. Các framework ORM có thể sử dụng cho JPA như Hibernate, Toplink, Open JPA,...

Ngày nay đa số các nhà cung cấp đều hỗ trợ cài đặt JPA cho framework persistence của họ. Vì vậy, người lập trình có thể lựa chọn nhà cung cấp tốt nhất tùy thuộc yêu cầu ứng dụng của mình.

Sau đây ta cùng tìm hiểu một số khái niệm của JPA

Entity: là các đối tượng persistence thể hiện một mẩu tin trong cơ sở dữ liệu. Entity chỉ là các lớp POJO đơn giản, dễ dàng tạo. Dưới đây là một số đặc điểm của một Entity:

- Entity có thể tương tác với cơ sở dữ liệu quan hệ
- Entity được xác định thông qua một định danh (tương đương với khóa chính trong table của cơ sở dữ liệu quan hệ)
- Entity hỗ trợ giao tác (transaction)
- Entity hỗ trợ kế thừa giống như những lớp thường khác

EntityManager: là một giao diện (interface) cung cấp các API cho việc tương tác với các Entity. Một số chức năng cơ bản của EntityManager như:

- Persist: phương thức này dùng để lưu một thực thể mới tạo vào cơ sở dữ liệu
- Merge: dùng để cập nhật trạng thái của entity vào cơ sở dữ liệu
- Remove: xóa một thực thể hiện của entity

EntityManagerFactory: được dùng để tạo ra một thể hiện của EntityManager

Việc giới thiệu JPA vào trong đặc tả J2EE 5 là một bước tiến lớn trong việc đơn giản hóa các quy trình phát triển ứng dụng. JPA đơn giản hóa mô hình thực thể dữ liệu và cộng thêm một số tính năng mới mà phiên bản EJB trước (EJB 2.0) không có. Giờ đây người lập trình có thể ánh xạ trực tiếp các đối tượng persistence với cơ sở dữ liệu quan hệ. JPA có thể sử dụng bên ngoài container, điều này không dễ thực hiện trong EJB 2.1. Bạn cũng có thể sử dụng JPA trong các ứng dụng swing.

Một số tính năng của JPA:



- JPA hỗ trợ pluggable, tức là có thể sử dụng nhiều nhà hàng cung cấp thứ ba như Hibernate hay Toplink.
- Hỗ trợ annotation
- Giảm bớt số lớp yêu cầu cho việc phát triển persistence
- Không cần phải viết các mô tả triển khai trong xml. Annotation dựa trên metadata đã hỗ trợ trong các ứng dụng JPA
- Đã chuẩn hóa ORM và dễ dàng phát triển hơn
- JPA hỗ trợ truy vấn động và tĩnh
- Nhiều IDE hỗ trợ phát triển ứng dụng JPA và có thể tự động sinh code ánh xạ từ cơ sở dữ liệu thành các entity và ngược lại

3.4 AngularJS

3.4.1 Giới thiệu về AngularJS

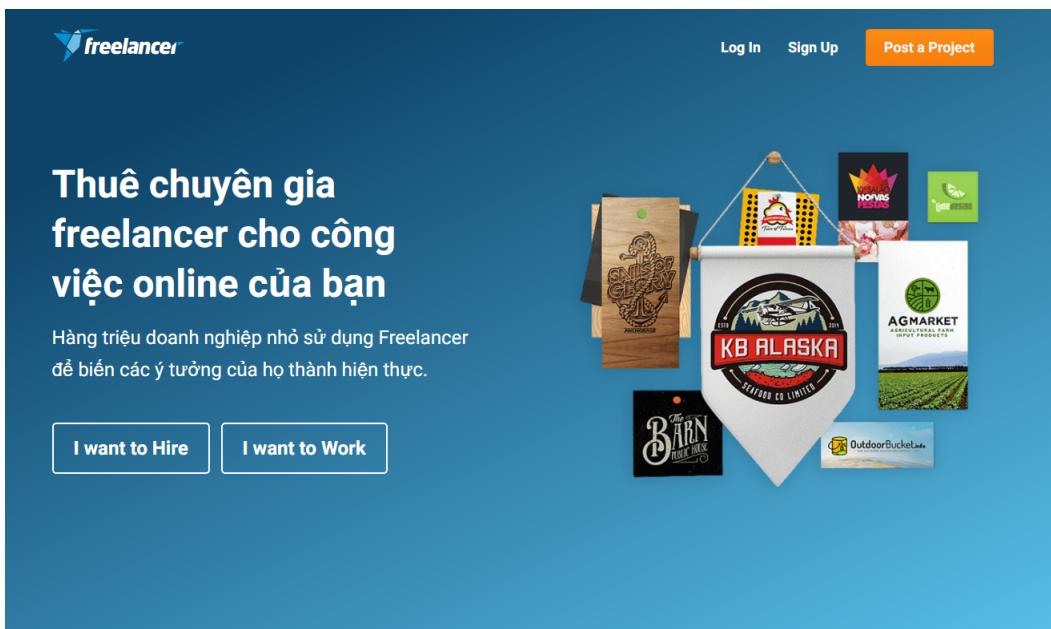
AngularJS là một framework có cấu trúc cho các ứng dụng web động. AngularJS cho phép lập trình viên sử dụng HTML như là ngôn ngữ mẫu và cho phép bạn mở rộng cú pháp của HTML để diễn đạt các thành phần trong ứng dụng một cách rõ ràng và súc tích. Hai tính năng cốt lõi: **Data binding** và **Dependency injection** của AngularJS loại bỏ phần lớn code dư thừa.

AngularJS được bắt đầu từ năm 2009, do lập trình viên Misko Hevery tại Google tạo ra. Misko Hevery và nhóm lúc này đang tham gia vào 1 dự án của Google tên là Google Feedback. Với AngularJS, Misko đã rút ngắn số dòng code front-end từ 17000 dòng còn chỉ khoảng 1500. Với sự thành công đó, đội ngũ của dự án Google Feedback quyết định phát triển AngularJS theo hướng mã nguồn mở.

Hiện nay, nhiều websites nổi tiếng được xây dựng từ AngularJS như <https://www.paypal.com>, <https://www.freelancer.com/>... có với giao diện đẹp, thân thiện với người dùng (Hình 13).

3.4.2 Các đặc tính của AngularJS

- AngularJS là một framework phát triển mạnh mẽ dựa trên JavaScript để tạo các ứng dụng RICH Internet Application (RIA).
- AngularJS cung cấp cho lập trình viên những tùy chọn để viết các ứng dụng client-side trong mô hình MVC (Model View Controller) một cách rõ ràng.
- Các ứng dụng được viết bởi AngularJS tương thích với nhiều phiên bản trình duyệt web. AngularJS tự động xử lý mã JavaScript để phù hợp với mỗi trình duyệt.
- AngularJS có mã nguồn mở, miễn phí hoàn toàn, được sử dụng bởi hàng ngàn lập trình viên trên thế giới. Nó hoạt động dưới giấy phép Apache License version 2.0.



Hình 13: Giao diện website <https://www.freelancer.com/> được xây dựng từ AngularJS

3.4.3 Các tính năng cốt lõi của AngularJS

- **Data-binding:** Nó tự động đồng bộ hóa dữ liệu giữa thành phần model và view.
- **Scope:** Là những đối tượng hướng đến model, nó hoạt động như là cầu nối giữa controller và view.
- **Controller:** Đây là những tính năng của AngularJS mà được giới hạn tối một scope cụ thể.
- **Service:** AngularJS hoạt động với một vài dịch vụ (service) có sẵn, ví dụ \$http để tạo XMLHttpRequests. Nó là các singleton object mà được khởi tạo duy nhất một lần trong ứng dụng.
- **Filter:** Nó lựa chọn (hay là lọc) các tập con từ tập item trong các mảng và trả về các mảng mới.
- **Directive:** Directive là các marker trong các phần tử DOM (như các phần tử, thuộc tính, css và nhiều hơn thế). Nó có thể dùng để tạo các thẻ HTML riêng phục vụ những mục đích riêng. AngularJS có những directive có sẵn như ngBind, ngModel...
- **Template:** Là các rendered view với các thông tin từ controller và model. Nó có thể được sử dụng trong các file riêng biệt (ví dụ như index.jsp) hoặc nhiều view với một trang sử dụng "partials".
- **Routing:** Là khái niệm của sự chuyển dịch qua lại các view.
- **Model View Whatever:** MVC là một mô hình thiết kế để phân chia các ứng dụng thành nhiều phần khác nhau (gọi là Model, View và Controller), một phần sử dụng với một nhiệm



vụ nhất định. AngularJS không triển khai MVC theo cách truyền thống, mà gắn liền hơn với Model-View-ViewModel. Nhóm phát triển AngularJS đã đặt tên vui cho mô hình này là Model View Whatever.

- **Deep Linking:** Cho phép bạn mã hóa trạng thái các ứng dụng trên địa chỉ URL để nó có thể được bookmark. Các ứng dụng có thể được phục hồi lại từ các địa chỉ URL với cùng một trạng thái.
- **Dependency Injection:** AngularJS có sẵn một hệ thống con dependency injection để giúp các lập trình viên tạo ra các ứng dụng dễ phát triển, dễ hiểu và kiểm tra.

3.4.4 Các thành phần của AngularJS

AngularJS framework có thể được chia thành ba phần chính sau:

- **ng-app:** directive này định nghĩa và liên kết một ứng dụng AngularJS tới HTML.
- **ng-model:** directive này gắn kết giá trị của dữ liệu ứng dụng AngularJS đến các điều khiển đầu vào HTML.
- **ng-bind:** directive này gắn kết dữ liệu ứng dụng AngularJS đến các thẻ HTML.

3.4.5 Ưu và nhược điểm

Ưu điểm của AngularJS

- AngularJS cung cấp khả năng tạo ra các *Single Page Application* một cách rất rõ ràng và dễ dàng để duy trì.
- AngularJS cung cấp khả năng *Binding* tới HTML do đó giúp người dùng cảm giác linh hoạt, thân thiện.
- AngularJS hỗ trợ *Unit Test*.
- AngularJS sử dụng *Dependency Injection*.
- AngularJS cung cấp khả năng tái sử dụng các component (thành phần).
- Với AngularJS, lập trình viên sẽ viết ít code hơn, với nhiều chức năng hơn.
- Với AngularJS, view là thành phần trong trang HTML thuần, trong khi controller được viết bởi JavaScript với quá trình xử lý chức năng logic.

Và trên tất cả, ứng dụng AngularJS có thể chạy trên hết các trình duyệt web, trên các nền tảng Android và iOS.

Nhược điểm của AngularJS

Mặc dù AngularJS có thể kể đến rất nhiều các ưu điểm, nhưng đến thời điểm này, nó vẫn có một số điểm yếu sau:

- Ứng dụng được viết bởi AngularJS không an toàn. Phải có các tính năng bảo mật và xác thực phía server sẽ giúp ứng dụng trở nên an toàn hơn.
- Không thể hiển thị nếu người dùng vô hiệu hóa JavaScript.

3.5 Android

3.5.1 Giới thiệu về Android

Android là một hệ điều hành mã nguồn mở và là một hệ điều hành dựa trên Linux cho các thiết bị mobile như điện thoại thông minh và máy tính bảng. Ban đầu Android được phát triển bởi Công ty Android với sự hỗ trợ tài chính từ Google, sau đó được Google mua lại vào năm 2005.

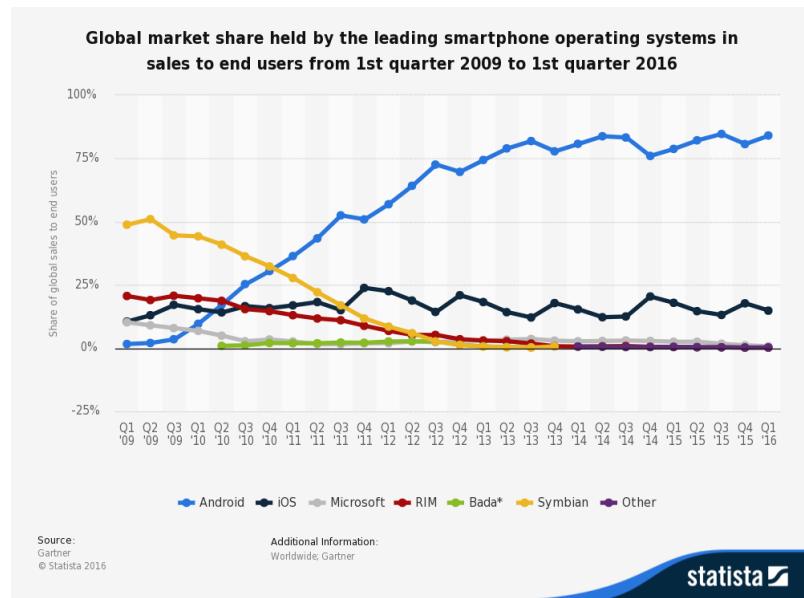
Android đưa ra một phương pháp thống nhất để phát triển ứng dụng cho các thiết bị di động, nghĩa là các lập trình viên chỉ cần phát triển Android, và các ứng dụng khác có thể chạy trên các thiết bị khác nhau mà đã được trang bị Android.

Phiên bản beta của Android Software Development Kit (SDK) được công bố bởi Google vào năm 2007, trong khi phiên bản thương mại đầu tiên Android 1.0 được công bố 9/2008. Kể từ tháng 4 năm 2009, phiên bản Android được phát triển, đặt tên theo chủ đề bánh kẹo và phát hành theo thứ tự bảng chữ cái: Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, Kitkat, Lollipop, và phiên bản hiện tại Marshmallow.



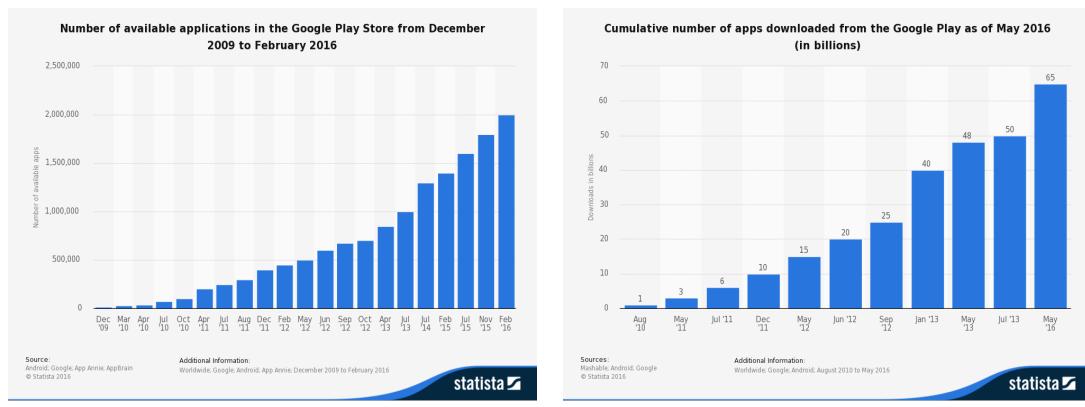
Hình 14: Các phiên bản của hệ điều hành Android tính đến thời điểm hiện tại

Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị, bằng một loại ngôn ngữ lập trình Java có sửa đổi. Chính vì vậy, Android có một tốc độ mở rộng và phát triển không tưởng, chiếm thị phần lớn nhất trong hệ thống điện thoại thông minh toàn cầu (84.1% vào quý 1 năm 2016) [7] (Hình 15)



Hình 15: Thị phần điện thoại thông minh phân loại theo hệ điều hành

Vào tháng 2 năm 2016, có khoảng 2 triệu ứng dụng trên Android [8], và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 65 tỉ lượt tải [9] (Hình 16)



(a) Số ứng dụng trên Android

(b) Số lượt tải ứng dụng trên Android

Hình 16: Số lượt tải và số lượng ứng dụng Android giai đoạn 2009-2016

3.5.2 Nguyên nhân sự phát triển mạnh mẽ của Android

- **Mã nguồn mở:** Bởi vì Android là mã nguồn mở nên lập trình viên dễ dàng tiếp cận với Android và thử nghiệm các công nghệ mới vào Android.
- **Mở rộng tầm ảnh hưởng của cộng đồng và lập trình viên**



- **Tăng hiệu quả marketing:** Dựa vào sự hỗ trợ từ Google Play, marketing cho ứng dụng trên Android có nhiều lợi thế hơn, dễ dàng quảng cáo ứng dụng đến người dùng trên toàn cầu.
- **Kết hợp các ứng dụng:** Android có tính linh động cao và có hỗ trợ để kết hợp với các ứng dụng trên các nền tảng khác.
- **Giảm chi phí phát triển:** Các công cụ để lập trình Android hoàn toàn miễn phí, và Google chỉ thu một khoản phí nhỏ để đưa ứng dụng Android lên Android Market. Việc xây dựng và phát triển một ứng dụng Android gần như không tốn chi phí.
- **Tỉ lệ thành công cao.**
- **Môi trường lập trình tốt:** Android cung cấp nhiều công cụ hỗ trợ tốt, SDK và các công nghệ hấp dẫn với lập trình viên để giúp họ sử dụng và phát triển ứng dụng.

3.5.3 Các tính năng cơ bản của Android

Android là một hệ điều hành đầy sức mạnh và hỗ trợ nhiều tính năng tuyệt vời, danh sách các tính năng cơ bản của Android được nhóm liệt kê trong Bảng 1.

Đặc điểm	Miêu tả
Giao diện đẹp	Màn hình Android OS cơ bản cung cấp một giao diện người dùng đẹp và có tính thẩm mỹ cao
Tính kết nối	GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC và WiMAX
Lưu trữ	SQLite, một cơ sở dữ liệu quan hệ gọn nhẹ, được sử dụng cho mục đích lưu trữ dữ liệu
Hỗ trợ Media	H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, và BMP
Tin nhắn	SMS và MMS
Trình duyệt Web	Dựa trên thiết bị WebKit mã nguồn mở, đi kèm với thiết bị V8 JavaScript của Chrome hỗ trợ HTML5 và CSS3
Multi-touch	Android hỗ trợ cho multi-touch mà đã được tạo ban đầu có sẵn cho các Handset
Đa nhiệm	Người dùng có thể nhảy từ một tác vụ tới tác vụ khác và nhiều ứng dụng có thể chạy đồng thời cùng một lúc
Widget tùy chỉnh	Widgets có thể thay đổi kích cỡ, vì thế người dùng có thể mở rộng để hiển thị nhiều nội dung hơn, hoặc thu nhỏ để tiết kiệm không gian
Đa ngôn ngữ	Hỗ trợ text đơn hướng và song hướng
GCM	Google Cloud Messaging (GCM) là một dịch vụ cho phép lập trình viên gửi thông điệp dữ liệu ngắn tới người dùng trên thiết bị Android, mà không cần một Sync Solution
Wi-Fi Direct	Một công nghệ cho phép các ứng dụng dò tìm và ghép cặp một cách trực tiếp, thông qua một kết nối peer-to-peer
Android Beam	Một công nghệ dựa trên NFC phổ biến cho phép người dùng chia sẻ tức thì, chỉ cần kích hoạt NFC của hai điện thoại với nhau

Bảng 1: Các tính năng cơ bản của hệ điều hành Android



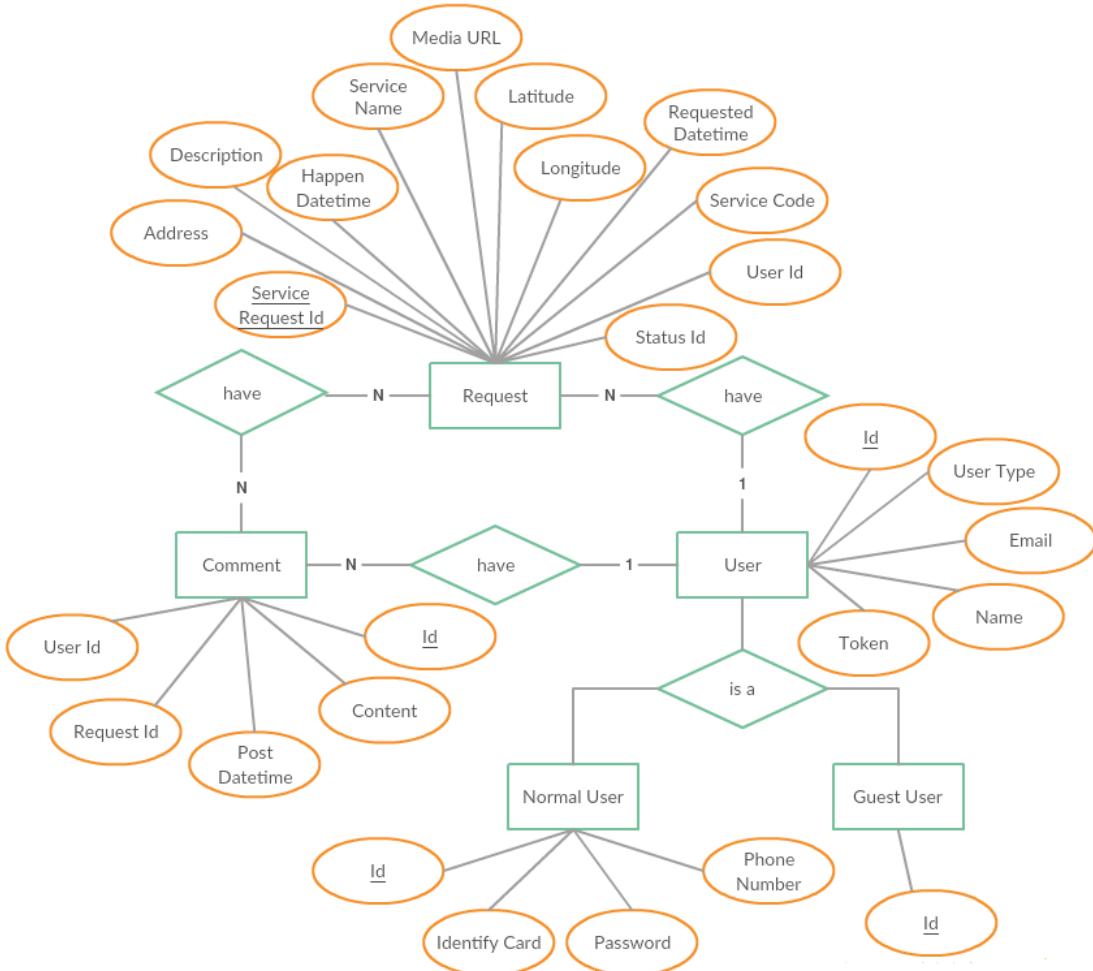
4 Hiện thực dự án

4.1 Cơ sở dữ liệu

Từ những phân tích về các chức năng của ứng dụng, chúng em xác định cơ sở dữ liệu sẽ bao gồm bảng User quản lý các tài khoản và thông tin người dùng, bảng Request và Comment lưu trữ góp ý và bình luận của người dùng. Lược đồ Entity - Relationship giữa các thực thể trong ứng dụng được mô tả ở Hình 17, bao gồm:

- **Request:** Là thực thể mô tả một góp ý của người dân đến chính quyền, các thuộc tính của thực thể này bao gồm tên, mã số, địa chỉ, mô tả, hình ảnh, vị trí, id người gửi, v.v
- **Comment:** Là thực thể mô tả bình luận trong về một góp ý nào đó, có thể có nhiều bình luận trong một góp ý, các thuộc tính của thực thể bao gồm id của bình luận, thời gian gửi, nội dung và thông tin người gửi
- **User:** Là thực thể chứa thông tin của một người dùng, đây là thực thể tổng quát hóa, bao gồm hai thực thể con là *người dùng thông thường (normal user)* và *khách (guest)*. Các thuộc tính của thực thể này bao gồm id, loại người dùng, email, tên và token của tài khoản
- **Normal User:** Là thực thể chứa thông tin của người dùng thông thường, được thừa kế từ thực thể User, bao gồm các thuộc tính id, số CMND, mật khẩu và số điện thoại
- **Guest User:** Là thực thể chứa thông tin của khách, được thừa kế từ thực thể User, chỉ có một thuộc tính là id

Tuy nhiên nhóm không tạo trước cơ sở dữ liệu mà sử dụng chức năng tạo ra cơ sở dữ liệu động trong lần chạy đầu tiên của ứng dụng. Việc này được thực hiện thông qua JPA API, các bảng trong cơ sở dữ liệu tương ứng với các JPA Entity đã được hiện thực trước đó.



Hình 17: Mô hình Entity - Relationship của đề án

4.2 Web Service

Với từng thực thể nói trên, chúng em đã hiện thực các Web Service tương ứng:

Request

Comment

GuestUser



Tên	Phương thức	Đối số	Mô tả
find	GET	id của Request	Lấy thông tin của một thực thể Request theo id
findAll	GET		Lấy thông tin của mọi Request có trong cơ sở dữ liệu
create	POST	Request mới	Thêm một Request mới vào cơ sở dữ liệu
edit	PUT	- id của Request - Request mới	Cập nhật một Request theo id của nó
remove	DELETE	id của Request	Xóa một Request theo id

Tên	Phương thức	Đối số	Mô tả
find	GET	id của Comment	Lấy thông tin của một thực thể Comment theo id
findAll	GET		Lấy thông tin của mọi Comment có trong cơ sở dữ liệu
create	POST	Comment mới	Thêm một Comment mới vào cơ sở dữ liệu
edit	PUT	- id của Comment - Comment mới	Cập nhật một Comment theo id của nó
remove	DELETE	id của Comment	Xóa một Comment theo id

NormalUser

Xác thực

Để quản lý quá trình xác thực người dùng khi đăng nhập, nhóm đã hiện thực thêm một Web Service có chức năng xác thực người dùng:

4.3 Client

Trong giai đoạn TTTN, nhóm đang hiện thực client trên nền tảng web.

5 Khó khăn và khắc phục

5.1 Xác thực và phân quyền

Java Web Token

6 Mở rộng đề tài

Hiện tại nhóm chỉ mới phát triển ứng dụng trên nền Web.

Trong giai đoạn luận văn tốt nghiệp, nhóm sẽ hoàn thành các nhiệm vụ sau:

- Hoàn chỉnh giao diện web cho ứng dụng



Tên	Phương thức	Đối số	Mô tả
create	POST	GuestUser mới	Thêm một GuestUser mới vào cơ sở dữ liệu

Tên	Phương thức	Đối số	Mô tả
find	GET	id của NormalUser	Lấy thông tin của một thực thể NormalUser theo id
findAll	GET		Lấy thông tin của mọi NormalUser có trong cơ sở dữ liệu
create	POST	NormalUser mới	Thêm một NormalUser mới vào cơ sở dữ liệu
edit	PUT	- id của NormalUser - NormalUser mới	Cập nhật một NormalUser theo id của nó
remove	DELETE	id của NormalUser	Xóa một NormalUser theo id

- Tiếp tục tìm hiểu về cách làm việc của CPDT, thủ tục hành chính của các bên liên quan để hoàn thiện các luồng làm việc của ứng dụng
- Sửa chữa các lỗi còn tồn tại
- Phát triển ứng dụng trên nền tảng di động
- Viết báo cáo về đề tài

7 Kết luận

Dề án Chính phủ điện tử hiện nay là cơ hội lớn cho ngành công nghệ thông tin Việt Nam. Nhận thấy tầm quan trọng của nó, chúng em hy vọng với đề tài này, chúng em có thể góp sức vào công cuộc phát triển và áp dụng CPDT vào thực tiễn ở địa bàn Thành phố Hồ Chí Minh

Sau cuối, nhóm xin cảm ơn thầy cô đã hỗ trợ giải đáp các khó khăn và vướng mắc, giúp đỡ chúng em hoàn thiện sản phẩm.



Tên	Phương thức	Đối số	Mô tả
authenticationUser	POST	- email - password	Xác thực người dùng theo email và password

8 Phụ lục

Tài liệu

- [1] United Nation (13/01/2010). *A General Framework for E-Government: Definition - Maturity Challenges, Opportunities, and Success.* Truy cập 18/05/2016 tại <http://www.unpan.org/Library/MajorPublications/UNEGovernmentSurvey/PublicEGovernanceSurveyintheNews/tabid/651/mctl/ArticleView/ModuleId/1555/articleId/20840/Default.aspx>
- [2] World Bank (19/05/2015). *Information & Communication Technologies: e-Government.* Truy cập 18/05/2016 tại <http://www.worldbank.org/en/topic/ict/brief/e-government>
- [3] United Nation (2014). *Data Center: 2014 E-Government Development Index.* Truy cập 18/05/2016 tại <https://publicadministration.un.org/egovkb/en-us/Data-Center>
- [4] Ministry of Finance Singapore (4/9/2013). *E-Government in Singapore.* Truy cập 23/05/2016 tại <http://workspace.unpan.org/sites/Internet/Documents/UNPAN90601.pdf>
- [5] Open311 Documents. Truy cập tại <http://www.open311.org/>
- [6] AngularJS Documents. Truy cập tại <https://docs.angularjs.org/>
- [7] Thị phần điện thoại thông minh. Truy cập tại <http://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>
- [8] Số lượng ứng dụng trên Android. Truy cập tại <http://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>
- [9] Số lần tải ứng dụng trên Android. Truy cập tại <http://www.statista.com/statistics/281106/number-of-android-app-downloads-from-google-play/>
- [10] Tên tác giả, năm xuất bản. *Tên sách.* Lần xuất bản. Thành phố: Nhà xuất bản.



Phân công công việc

STT	Tên	Nhiệm vụ	Dánh giá
1	Võ Văn Luận	Xây dựng database, model, router	100%
2	Trần Văn Tài	Thiết kế trang đăng nhập, đăng ký, chỉnh sửa thông tin, giao diện	100%

Danh sách hình vẽ

1	Chỉ số phát triển CPDT của Việt Nam năm 2014	4
2	Chỉ số phát triển CPDT các quốc gia năm 2014	4
3	Trang web CPDT Nyc.gov của thành phố New York	6
4	Gửi lời phàn nàn đến chính quyền	6
5	Người dân thanh toán phí công cộng	7
6	Ứng dụng CPDT của New York trên hệ điều hành Android và iPhone	7
7	Trang web CPDT ecitizen.gov.sg của thành phố Singapore	8
8	Trang web CPDT hochiminhhcity.gov.vn của thành phố Hồ Chí Minh	9
9	Trang góp ý của hochiminhhcity.gov.vn	10
10	Trang web CPDT egov.danang.gov.vn của Thành phố Đà Nẵng	10
11	Trang góp ý của egov.danang.gov.vn	11
12	Ứng dụng CPDT của Đà Nẵng trên Zalo	12
13	Giao diện website https://www.freelancer.com/ được xây dựng từ AngularJS	19
14	Các phiên bản của hệ điều hành Android tính đến thời điểm hiện tại	21
15	Thị phần điện thoại thông minh phân loại theo hệ điều hành	22
16	Số lượt tải và số lượng ứng dụng Android giai đoạn 2009-2016	22
17	Mô hình Entity - Relationship của đề án	25