International Journal of Electronic Commerce Studies
Vol.6, No.1, pp.127-136, 2015
doi: 10.7903/ijecs.1399

UBIQUITOUS JOB RECOMMENDATION SYSTEM FOR GRADUATES IN TAIWAN

Chi-Tsai Yeh
Shih Chien University
200 University Road, Neimen, Kaohsiung 84550, Taiwan
yehchitsai@mail.kh.usc.edu.tw

Tzuo-Ming Chen
Shih Chien University
200 University Road, Neimen, Kaohsiung 84550, Taiwan
jchen@mail.kh.usc.edu.tw

ABSTRACT

The number of graduates with bachelor degrees has increased rapidly from 117,430 in 2000 to 232,448 in 2011. It is evident that the recruitment market may be out of balance due the lack of a positive correlation with the employment rate. Based on a survey of the National Youth Commission (NYC) in 2010, 47.5% of youth employees received their jobs through media advertisement and 35.26% through the introduction of relatives, friends, etc. The rates via introductions are less than those via advertisement, since the former provides lower risk and higher reliability for the employers. This paper proposes a job recommendation system that aims to assist academia in promoting graduates to appropriate employers. With usability in mind, the system provides a ubiquitous framework with web service technologies and can be applied easily to web and mobile applications. A job recommendation prototype was implemented on the Android and iPhone mobile platforms. Through these systems, users could efficiently seek out and discover appropriate graduates and then recommend them to employers.

Keywords: Web Service, Mobile Development, Career Counseling, Recommendation System, E-Portfolio

1. INTRODUCTION

Although the number of baccalaureate graduates has been increasing[1], relative unemployment rates have remained relatively high. Due to the sheer number of graduates in recent decades, institutes of higher education have asked post-baccalaureates to prepare more easily manageable electronic portfolios (E-portfolio). Yancey[2] proposed of E-portfolios to track student learning. Regardless of university sanctioned career advocacy programs, there are still several unsolved problems with this system in the search for employment.

First, the shortcomings of an E-portfolio must be addressed. It may be able to reflect a student's academic progress, but an E-portfolio in itself may not be able to attract employers looking for employees for their companies. The other problem is that universities emphasize an analysis of career intention, instead of providing actual recommendations or job matching. According to the 2010 survey of NSC[3], 47.5% of youth employees received their jobs through media advertisement and 35.26% through the introduction of the relatives and friends. Therefore, the recommendation of relatives or friends is the next best channel in the job search. Companies often encourage their employees to recommend candidates for new positions. This is partly due to the low monetary cost of such a method relative to other methods of recruitment. Also, the candidates may have a better understanding about the company by communicating through acquaintances.

Our research proposes a ubiquitous job recommendation system (UJRS) to efficiently allow graduates to search for jobs. The system would perform the following tasks:

- Provide a collection of student E-portfolios to efficiently assist other users in finding employment and employee candidates.
- Teachers/alumni of the university play an important role in introducing the graduates to employers through this system.
- Using a standardized assessment model, it would appraise its own performance or collect performance data for future analysis, improvement, and accuracy.
- There would be an implementation of a credibility rating system. This set of data would rely on weighted votes. Such feedback would be used to deter unqualified recommendations from teachers/alumni.
- Considering the convenience of operation, the system provides versatile accessibility. The system will be optimized for various platforms such as mobile devices and personal computers.

The organization of this paper is as follows: Section 2 describes the technologies of web and mobile development. Section 3 explains the system architecture and the concept of its design. Section 4 expresses the occurrence of possible system problems and demonstrates a system prototype. Finally, conclusions are provided.

2. RELATED WORK

The World Wide Web Consortium (W3C) defines a "Web service" as a software system designed to support interoperable machine-to-machine interaction over a network, typically conveyed using hypertext transfer protocol (HTTP)[4]. Representational state transfer (REST) is one such web service architecture for distributed systems. Roy Fielding[5, 6] introduced and defined the REST architectural style developed by the W3C Technical Architecture Group (TAG). REST-style architectures conventionally consist of clients and servers. Clients initiate requests to servers; servers process requests and return appropriate responses. Requests and responses are built around the transfer of representations of resources. A resource can essentially be any addressable, coherent, or meaningful concept. A representation of a resource is typically a document that captures the current or intended state of a resource[7]. JavaScript Object Notation (JSON)[8] is a text-based open standard designed for human-readable data interchange. This paper uses JSON as a representation of a resource in REST.

The following platform providers have dominated the mobile development market in the first quarter of 2013: Google with its Android OS (74.4%), Apple with its iOS (18.2%), RIM with its Blackberry OS (3%), and Microsoft with its Windows Phone OS family (2.9%)[9]. Developers have often faced cross-platform complications through the use of the PhoneGap framework.

PhoneGap[10] enables software programmers to build applications for mobile devices using the common web languages such as JavaScript, hypertext markup language 5 (HTML5), and cascading style sheets 3 (CSS3) instead of device-specific languages such as Objective-C for iOS. The resulting applications are a hybrid of native languages and web-based languages. Ghatol and Yogesh[11] expressed the application architecture of PhoneGap. PhoneGap Build[12] can create several packages for different platforms automatically. Because of this, developers can write applications once and run them on any mobile device.

The abundance and popularity of social media sites floods users with huge volumes of information. Social recommendation systems (SRSs)[13, 14] aim to filter the most attractive and relevant content to social media users. Content-based and collaborative filtering are fundamental approaches and are widely adopted for building SRSs. Guy et al.[15] demonstrated a recommendation system that provides users with recommendations of people to invite into their explicit enterprise social network. The recommendations are based on aggregated information collected from various sources across the organization. Eirinaki, etc.[16] thought that inter-user trust was an essential quality for such interactions, and the recommendation of useful content and trusting users was crucial for all the members of the network. They introduced a framework for handling users trust levels in SRS based on a reputation mechanism. The UJRS depends on the trust and reputation of the teacher/mentor to recommend graduates to employers.

3. ARCHITECTURE OF THE PROPOSED UJRS

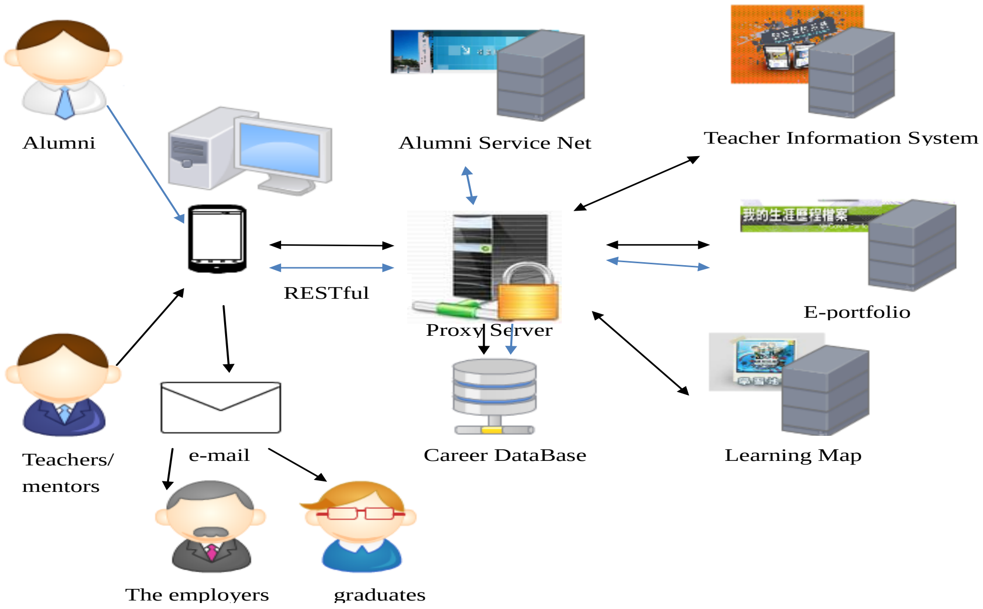

Figure 1. Architecture of the proposed UJRS.

In addition to the ubiquitous and cross-platform accessibility of the RESTful data exchange protocol, UJRS reduces excess data transmission and client processing in ways such as parsing HTML tags. All the corresponding information is stored on separate servers: the Teacher Information System, the Alumni Service Net, the Learning Map and the E-portfolio as shown in Figure 1. This system provides the following services:

- User Authentication: This UJRS is limited the teachers and alumni of a university. The users are each asked to authenticate their account through the Teacher Information System and Alumni Service Net.
- Data retrieval: The users publish their data in the E-portfolio and Learning Map servers. The UJRS proxy server collects the

corresponding data from the respective servers and stores them in the career database to increase query speed.

- Candidates Search: All the specialties and certificates will be tagged for search.
- Recommendation: The teachers/mentors will be able to write anonymous comments about the graduates or electronically recommend them to employers.
- Feedback: The employers could give feedback on their experience by providing various recommendations and specifics.

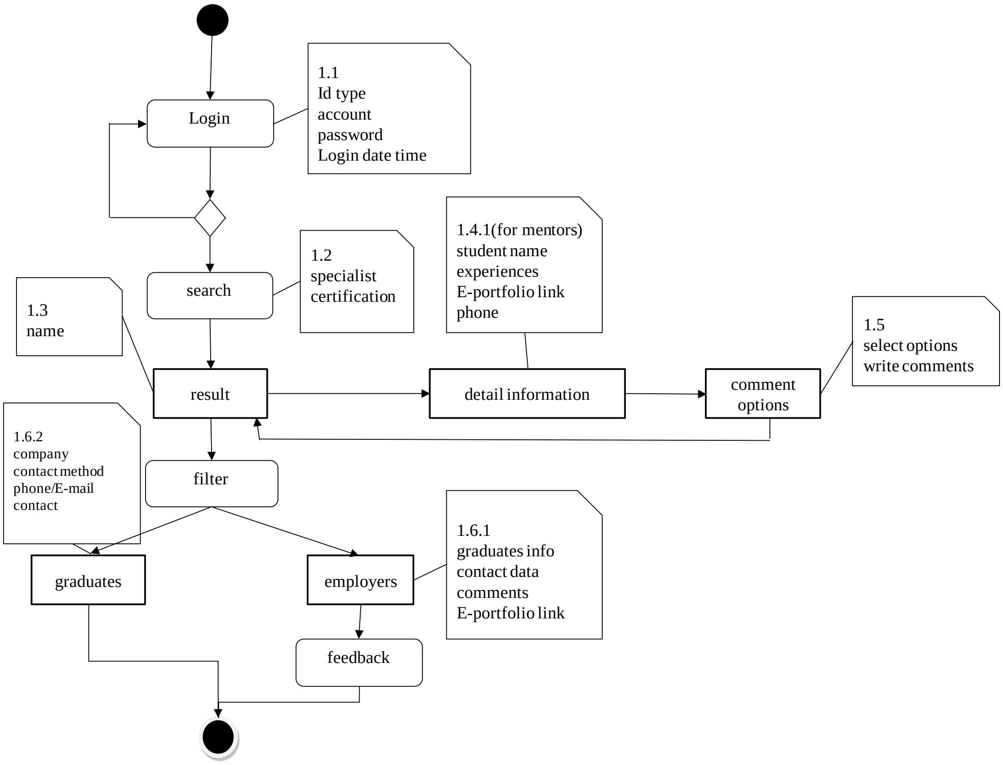

Figure 2. System activity diagram of the UJRS.

Figure 2 shows the activity diagram of UJRS. To utilize the system, a user can operate the system in the following steps.

Step 1: Authenticate the user and verify the login.
Step 2: Use search criteria based on student attributes to find qualified users.
Step 3: List the qualifying students and highlight a specific student for further examination, depending on a reviewer’s status (instructor or counselor).
Step 4: A reviewer can make some comments on a student on an opinion window viewing his or her information.

Step 5: Either further inspect the querying results or repeat Step 2 in query window.
Step 6: Send specified information to both a student and an employer.

4. UJRS PROBLEMS AND PROTOTYPE

4.1 UJRS Problems

During the development of the UJRS system, the designers should resolve the following problems:

- User Authentication: Consider possible shortcomings in the response time and flexibility, since this UJRS sets up a proxy server to authenticate users. The mobile users/clients send the account/password to the proxy server that forwards the data to the appropriate web sites to validate the users.
- Information Fetch: In order to speed up the process of data load, the open information of the E-portfolio server will be pre-fetched to the local database. Students can decide to publish or not publish their E-portfolio.
- Cross Platform: REST proposes a solution to improve cross-platform performance. It divides the job to two sides, user side and server side. The user side processes the web display and the server side the contents themselves.

4.2 UJRS Prototype

Both Android and iOS operating systems are referenced in UJRS design. An iOS virtual machine environment was used to test the system. Although mobile device performance does not match PC performance, we applied the “Sencha Touch Framework”[17] to enhance the interface design. We also used RESTful concepts to reduce both data transfer and redundant web processing. Components of UJRS development are shown in Table 1 in detail.

Table 2 compares our design and a web based design. Traditionally, a web browser user watches one page through a succession of steps. The first step is to retrieve web data via HTTP consisting of an HTML page, JavaScript files, CSS files, etc. The web browser then parses the corresponding files to render onscreen. The first login page of the UJRS would transfer about 928K bytes via the Internet using web-based technology. Our proposed design will store related files on local devices and transfer the required JSON encoded data via Internet, thus reducing the amount of network traffic relative to web-based methods.

Table 1. Development tools for smart phones and proxy server

Mobile devices (client side)	Host OS	Windows 7	Mac OS
	Mobile phone OS	Android 2.3, Android 4.0	iPhone 6.0
	IDE tools	Eclipse for Java	Xcode
	Verification platforms	HTC Z710e, Samsung i8150 Galaxy W	iPhone 6.0 simulator
	Development software	CSS, HTML, Sencha Touch, PhoneGap	
	Version control	Tortoise SVN、Eclipse SVN	
Proxy server (server side)	Development software	PHP、Apache	
	Database	My SQL 5.5.24	
	Version control	Subversive SVN	

Table 2. Comparison of web-based and proposed design on login data traffic

	HTML	Javascript			CSS			data	Sum
Web-based	login	jquery	Sencha touch	my	jquery	Sencha touch	my	0	928K
	5K	93K	568K	9K	72K	179k	2K		
Our design	Local files and data								0K

5. CONCLUSIONS

This research integrates information from the learning-map and E-portfolio systems, which can be referred to by employers. Due to security reasons, faculty members and researchers are the only users of the current system. We hope to cooperate with companies so that they may conduct preliminary evaluations for this recommendation system. Meanwhile, students will be encouraged to build their E-portfolios and understand their utility. This system allows users to take advantage of the convenience of mobile devices while supporting our recommendation system. By the nature of this technology, both students and employers will mutually benefit.

6. ACKNOWLEDGMENT

This work was supported by College of Business and Information, Shih Chien University under contracts USC-101-05-05006 and USC-102-05-05001.

7. REFERENCES

[1] Statistical Summaries of Ministry of Education, *Number of Schools, Teachers, Staff, Classes, Students and Graduates (1950-2012).* Retrieved on June 1, 2013, from http://english.moe.gov.tw/ct.asp?xItem=14530&ctNode=11432&mp=1

.

[2] K.B. Yancey, Electronic portfolios: Emerging practices in student, faculty, and institutional learning. In Barbara Cambridge (Ed.), *Digitized student portfolios* (p.15-30). Washington, DC: American Association for Higher Education, 2001.

[3] Survey report of the National Youth Commission (NYC), Survey report on youth employment. Retrieved on April 10, 2011, from http://open.nat.gov.tw/OpenFront/gpnet_detail.jspx?gpn=1009902711#d, 2010.

[4] W3C, *Web Services Glossary*, Retrieved on November 21, 2013, from http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice.

[5] R.T. Fielding, *Architectural styles and the design of network-based software architectures*. Diss. University of California, 2000.

[6] R.T. Fielding, and N.T. Richard, Principled design of the modern Web architecture. *ACM Transactions on Internet Technology (TOIT)*, 2(2), p115-150, 2002.

[7] Wikipedia, *Representational state transfer*. Retrieved on November 21, 2013, from http://en.wikipedia.org/wiki/REST.

[8] Google, *Using JSON with Google data APIs*. Retrieved on July 3, 2013, from http://code.google.com/apis/gdata/json.html.

[9] Gartner, *Worldwide mobile phone sales*. Retrieved on May 20, 2013, from http://www.gartner.com/newsroom/id/2482816.

[10] PhoneGap, *Easily create apps using the web technologies you know and love: HTML, CSS, and JavaScript*. Retrieved on April 10, 2012, from http://phonegap.com/.

[11] R. Ghatol, and P. Yogesh, *Beginning PhoneGap*. New York: Apress Media, 2012.

[12] PhoneGap Build, *Take the pain out of developing mobile apps*. Retrieved on April 7, 2012, from https://build.phonegap.com/.

[13] K. Irwin, R.L. Michael, and Hao Ma, Introduction to social recommendation. *Paper presented at the 19th International Conference on World Wide Web,* Raleigh, North Carolina, USA, April 26-30, 2010. http://dx.doi.org/10.1145/1772690.1772927.

[14] I. Guy, and C. David, Social recommender systems. In G. Ghinita and K. Punera (Eds.), *Proceedings of the 20th International Conference Companion on World Wide Web* (p283-284). New York, NY, USA: ACM Press, 2011. http://dx.doi.org/10.1145/1963192.1963312.

[15] I. Guy, R. Inbal, and W. Eric, Do you know? Recommending people to invite into your social network. In C. Conati and M. Bauer (Eds.), *Proceedings of the 14th International Conference on Intelligent User Interfaces* (p77-86). New York, NY, USA: ACM Press, 2009. http://dx.doi.org/10.1145/1502650.1502664.

[16] M. Eirinaki, M.D. Louta, and I. Varlamis, A trust-aware system for

personalized user recommendations in social networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(4), p409-421, 2013. http://dx.doi.org/10.1109/TSMC.2013.2263128.

[17] Sencha, *Sencha touch*. Retrieved on July 4, 2012, from http://www.sencha.com/products/touch/.