Efficient Retrieval of GPS information through cloud computing

*[1]Jefferson Andrew Prince, *[2]Mohan.V, *[3]Kumarasamy.N
*[1]andreuprince@gmail.com, *[2]mohanjaya449@gmail.com, #[3]skvkumaran@skpec.in
Department of Information Technology
SKP Engineering College

Abstract— — ‘‘Mobile tourism’’ represents a relatively new trend in the field of tourism and involves the use of mobile devices as electronic tourist guides. While much of the underlying technology is already available, there are still open challenges with respect to design, usability, portability, and functionality and implementation aspects. Most existing ‘‘mobile tourism’’ solutions either represent of-theshelf applications with rigidly defined content or involve portable devices with networking capabilities that access tourist content with the requirement of constant airtime, i.e., continuous wireless network coverage. This paper presents the design and implementation issues of a ‘‘mobile tourism’’ research prototype, which brings together the main assets of the two aforementioned approaches. Namely, it enables the creation of portable tourist applications with rich content that matches user preferences. The users may download these personalized applications (optimized for their specific device’s model) either directly to their mobile device or first to a PC and then to a mobile terminal (through infrared or bluetooth). Thereafter, network coverage is not further required as the applications execute in standalone mode and may be updated when the user returns online.

Keywords—web services, XML, GPS, Cloud Computing

Introduction

Wireless access through mobile devices adds to the Internet connection the element of ‘‘portability’’, i.e., connection with no time or geographical limitations, by devices with high penetration to the public [1]; tourists are amongst this technology-oriented public. Hence, a growing body of commercial and research initiatives that incorporates Electronic tourist guide functionality into mobile devices have been reported [4]. However these technologies in general have had a limited success; this is due to the lack of an in-depth study of the special characteristics of tourism, which can draw implications for the design of mobile tourist applications [6]. In the remainder of this section, several commercial and research projects with respect to mobile tourist guides are being reviewed.

The computing processes are managed to solve the Limited computing power of some mobile devices. Appropriate frameworks were chosen and best third-party libraries avoid any unnecessary processes. A multithreaded model is used to put some heavy and time-consuming tasks into the queue of tasks to be executed in the background.

The processor executes the tasks in the background when it is free and it prioritizes the main queue to execute user interactions first. Mobile devices have also limited amount of memory [3]. By monitoring memory peak usage during the development stage memory usage efficiency was improved. The application downloads an XML file and parses the file in the memory. A library that is the fastest of all of the current methods and efficient in memory usage was chosen. The last limitation is bandwidth and high cost of wireless connections [5].

A small-embedded database on the mobile device was designed that stores data to use when it is necessary, for instance when tourists are not connected to the Internet. A caching system and image store was developed for local storage of a mobile device to reduce the number of requests to the back-end level and save data traffic. It costs less from users perspective when they try to access to the Same images by their own 3G/LTE network. This approach also gives a better functionality when tourists are in slow public networks and offline mode.

A location-based mobile application for cultural tourism in Malaysia was designed and developed by using a cloud based platform to find out where the tourists are, where they are looking at by using heading technology or digital compass, calculate distances between current position and places, display only nearby attractions, give direction to tourists, and let them share visited places with other people.

This article presents the analysis and design of a city tourist guide system and also its implementation in a prototype system, the myMytileneCity guide. Our prototype includes a database-enabled tourist web site wherein, on a first stage, tourists planning to visit the city of Mytilene (Lesvos Island, Greece) choose the content that interests them (lodging, sightseeing, entertainment, etc.); based on that chosen content, the system automatically generates a custom application which can operate on their mobile phone or PDA. On a second stage, the users may download their application either directly to their mobile device or first to a PC and then to a mobile terminal (through infrared or bluetooth).[2] In contrast to i-mode/ WAP-enabled applications that presuppose continuous connection to the service provider’s network, the-MytileneCity guide does not pose such requirement. That is, following its installation, the application is fully functional with no extra charge, even in places where the connection to the mobile network is not feasible. Moreover, in contrast to existing applications, the myMytileneCity guide incorporates a ‘‘push model’’. Namely, it enables mobile users to be notified and update their personalized application as soon as new content or services, which the user has registered to, is added or updated by the web site's administrator. Our prototype implementation is based on Java 2 Micro Edition (J2ME) which represents an ideal platform for developing powerful and portable applications operating in asynchronous mode.

II. DESIGN AND DEVELOPMENT

The front-end level was designed and developed for the End-user mobile devices, which are the Apple mobile devices programmed on iOS platforms, the middle-ware level provides a Web service to generate XML output from the relational database to communicate with the mobile application. Finally, the back-end level that is built on AWS cloud platform to provide three services, Apache Web server on Amazon Elastic Computing Cloud (EC2), MySQL database server that is deployed on Amazon Relational Database Service (RDS), and a cloud storage that uses Amazon Simple Storage Service (S3), to have a flexible, dynamically scalable, and secure network infrastructure.

A. Frontend Application

- The front-end level is a location-based mobile tourism application for Android mobile devices, to explore nearby cultural tourism places.

- It shows some information in annotation callout such as name of the place, category of the place (we defined three categories: museums, natural parks, and historical places), calculated distance between the tourist's current location and the selected place on the map, and a photo of that place.

When the application displays the places on the map the title, description, photo URL, latitude, and longitude are already in the memory and stored in the SQLite for further offline usage. Although calculating the distances between the current location and the DOM elements and the filtering operations are done, still one more operation is needed so when the tourist taps on an annotation the photo of a place inside the annotation callout can be seen. By using "img URL" from each DOM element the image can be downloaded from the given URL to display it inside the callout. Before simply requesting the data from the stored URL address, there are some concerns that should be considered. The actions that need to be computed by mobile processor And files that need to be downloaded from Internet are the most expensive acts that can lead to poor functionality by being slow, not responsive to the user interactions, and also unexpected crashes.

To avoid this, images are downloaded from the online cloud storage server every time a tourist taps on any annotation on a map, instead of downloading the Images all at once at the outset. However, downloading needs to be in the way that waiting time does not block any user interaction or Graphic User Interface (GUI) on a device. A multithreaded model was used to put some heavy and time-consuming tasks into the queue of tasks to be executed in the background.Therefore, the processor executes the tasks in the background when it is free and it prioritizes the main queue first to execute user interactions first. Additionally, this method of only downloading the selected place makes the application launch faster by reducing the initialization time. There is another benefit that data traffic is saved on both tourist's side and our side.

Tourists who are not using public Internet need to buy a package plan or other services in order to connect to the Internet. Downloading so many images knowing that tourists may not use them all is wasting data. Downloading unnecessary content is not an efficient usage of the cloud storage. To keep requests as efficient as possible and avoid unnecessary request to the cloud storage server a cache system was used where the downloaded image will be stored on a local storage on mobile device. Next time the tourist requests that image by tapping on an annotation, it first checks the disk to see if there is any image is saved with the same name, if so will load the image from local storage and not request to the cloud storage. As it shown in Fig.1, there is a detail disclosure button with blue color that navigates tourists to the next view where they can find more information. Perhaps repeating some information from previous view such as a same image but enlarged to be more visible, a distance to that place, and category of the place would be helpful just in case if they missed it or they need to see again.

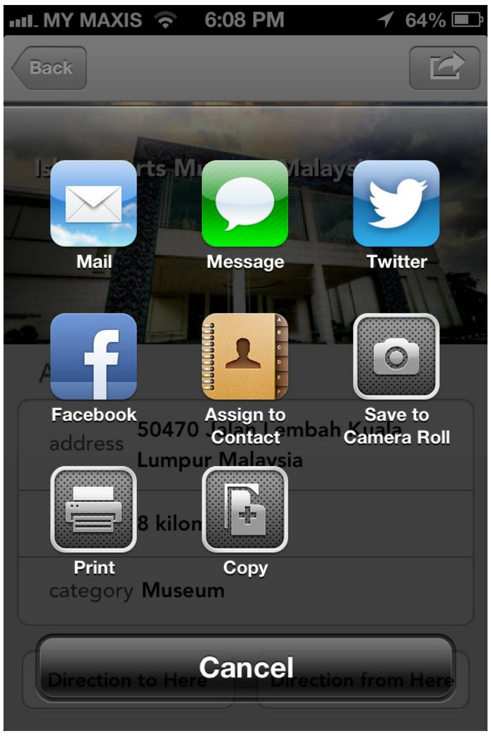


Figure 1. (a) Front end

B. Data exchange Model

This level is responsible to exchange data between the Location-based mobile tourism application in front-end level and the relational database server deployed on Amazon RDS. As it shown in Fig. 6, the application sends a simple request via http and the middle-ware level tries to query the data from the MySQL database server and after reading the data from database, it generates XML output so later front-end level can parse and make DOM elements.

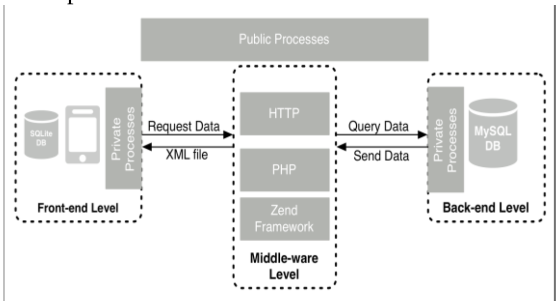


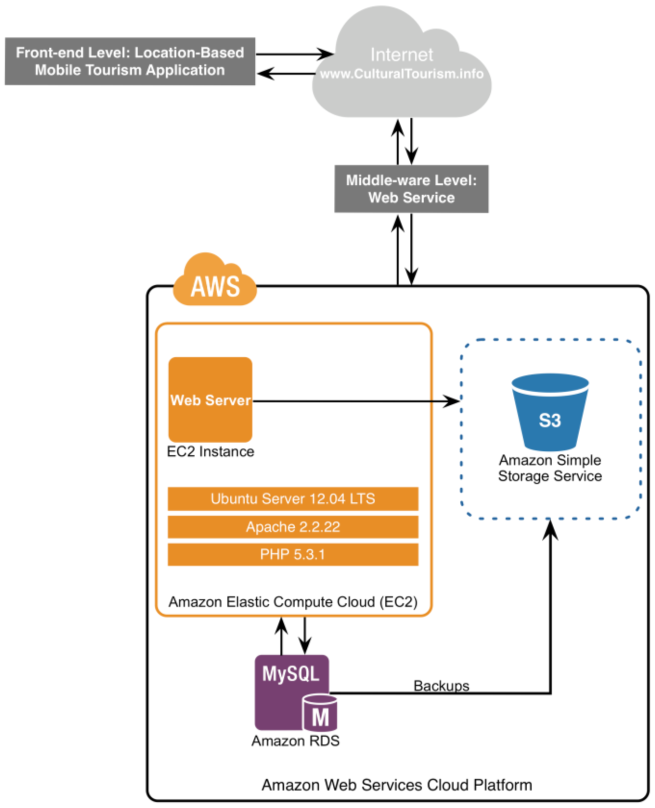
Figure 2. Data exchange model provided by web service written in
PHP framework

- ❑ This level is responsible to exchange data between the location-based mobile tourism application in front-end level and the relational database server deployed on Amazon RDS.
- ❑ The application sends a simple request via http and the middle-ware level tries to query the data from the MySQL database server and after reading the data from database, it generates XML output so later front-end level can parse and make DOM elements.

C. Cloud Platform

In back-end level the servers and the network infrastructure were built as seen in Fig. 7 by using Amazon Web Services (AWS) cloud platform. Amazon Elastic Cloud Computing (EC2) is used as a virtual server in the cloud to run the Apache Web server with resizable compute capacity. Amazon Relational Database Service (RDS) was used to have a relational MySQL database server in the cloud to host the database. Finally the content was hosted on Amazon Simple Storage Service (S3) to provide fast and scalable storage in the cloud for storing the media files.

- ❑ The back-end level that is built on AWS cloud platform provides services such as Apache Web server on Amazon Elastic Computing Cloud (EC2), MySQL database server on Amazon Relational Database Service (RDS), and a Cloud storage that uses Amazon Simple Storage Service (S3), giving flexible, auto scalable, and secure servers to provide Required services.
- ❑ The application is also caching images on the local storage and storing some data in SQLite database on the mobile device itself to provide some offline activities.
- ❑ This process decreases the number of sending requests to the servers, in order to save some data traffic.



Front-end Level: Location-Based Mobile Tourism Application
Internet
www.CulturalTourism.info
Middle-ware Level: Web Service
AWS
Web Server
EC2 Instance
Ubuntu Server 12.04 LTS
Apache 2.2.22
PHP 5.3.1
Amazon Elastic Compute Cloud (EC2)
S3
Amazon Simple Storage Service
MySQL
M
Amazon RDS
Backups
Amazon Web Services Cloud Platform

Figure 3. Back-end level architecture on Amazon Web Services cloud Platform

CONCLUSION

In conclusion, the usage of mobile phones as offline mobile tourist guides requires further research. The myMytileneCity Guide (developed at our laboratory and presented herein) addresses the shortcomings of existing systems We designed and developed a location-based mobile tourism application for Android platforms as a front-end level which displays nearby cultural tourism attraction.We used a Web service to generate XML output from our relational database Amazon MySQL RDS to exchange data with our mobile application in middle-ware level. We use Amazon Web Services cloud platform to take advantage of cloud computing and cloud storage.

REFERENCES

[1].M. Kenteris, D. Gavalas, and D. Economou, "An innovative mobile electronic tourist guide application," *Personal and ubiquitous computing*, vol. 13, pp. 103-118, Feb 2009.

[2] G. Chen and D. Kotz, "A survey of context-aware mobile computing
research," Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College2000.

[3] M. Ebner, C. Stickel, and J. Kolbitsch, "iPhone/iPad human interface design," *HCI in Work and Learning, Life and Leisure*, pp. 489-492, 2010
.
[4] Apple Developer, (2012). *Xocde IDE*, [Online], Available: https://developer.apple.com/xcode/,

[5] R. Wenderlich, (2010). *How to choose the best XML parser for your iPhone project*, [Online], Available: http://www.raywenderlich.com/553/how-to-chose-the-best-xmlparser- for-your-iphone-project, January 2013.

[6] T. Bradley, (2009). *TBXML*, [Online], Available: http://www.tbxml.co.uk, August 2012.

[7] Apple Developer, (2012). *Core Location Framework Reference*,[Online], Available: https://developer.apple.com/library/ios/documentation/CoreLocation/
reference/CoreLocation_Framework/CoreLocation_Framework.pdf,September 2012.