

Design of a Location and VOD-Based Campus Guide

Jaegeol Yim

*Department of Computer Engineering, Dongguk University at Gyeongju
38066 Korea
yim@dongguk.ac.kr*

Abstract

The mobile campus guide is a location-based mobile application. Making use of the sensors installed on a smart phone, it recognizes the building in the picture taken by the phone camera and plays a video related to the building. This paper enhances the campus guide with watching status recognition. If a user holds still his smart phone aiming at a building, then the enhanced campus guide identifies the building aimed at and starts playing a video related to the building. The player is interactive, i.e., if the user touches a hotspot (an object on the screen in which the user is interested), the player pauses the current video and starts playing another video related to the hotspot.

Keywords: *Mobile App, LBS, Watching Status, Building recognition, Interactive Player*

1. Introduction

As the computing power and storage capacity of smart phones are becoming more powerful and larger, many new smart phone applications appear in the market every day. Since a smart phone is one of the most useful and popular personal belongings, a smart phone is the most suitable device on which a context aware service is installed. Context aware systems try to recognize who, where, when, and why the user is in order to provide important information to the user [1].

As an example of context aware service, the authors of [2] introduced a campus guide mobile app. using the GPS and the compass installed on the phone, it determines where the user is located and which direction the camera is aiming at. When the camera takes a picture, it recognizes which building is in the picture using the GPS value and the camera direction. In the process of recognition, it refers to the electronic map. An electronic map is a list of lines representing boundaries of the buildings on the campus. Once it recognizes the building on the photo, it starts playing a video related to the building.

This paper enhances the campus guide with watching status recognition. We define watching status as follows: Watching status is defined by the status in which a user holds still his smart phone aiming at a building. In addition to watching status recognition, we enhance the campus guide with interactive player introduced in [3]. The interactive player allows the user to touch the screen while watching a video. If the user touch the screen, the player identifies the touched object and starts playing a new video that is associated with the object.

2. Related Works

The campus guide is a location based service, context aware service, mobile app, and a video on demand (VOD) service. There are many VOD references [4-9]. Development of a mobile VOD system is introduced in [4]. After investigating functional user requirements of other VOD systems, the authors of [4] concluded that a mobile VOD system should provide the services shown in Figure 1 and introduced their database design as shown in Figure 2. The VOD system they implemented consists of a video streaming management system, database management system and a mobile application

(client) as shown in Figure 3. The process of the client mobile application is described in Figure 4. The mobile application starts with invoking 'Directory Service' to get a list of genres, and then invokes 'Video Service' to select a video and obtain the metadata associated with the video. Finally, it accesses the streaming server with the URL of the video to play it.

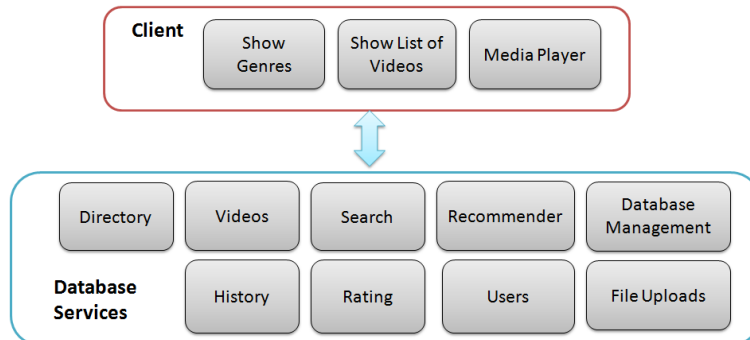


Figure 1. The Functional Requirements of A VOD [4]

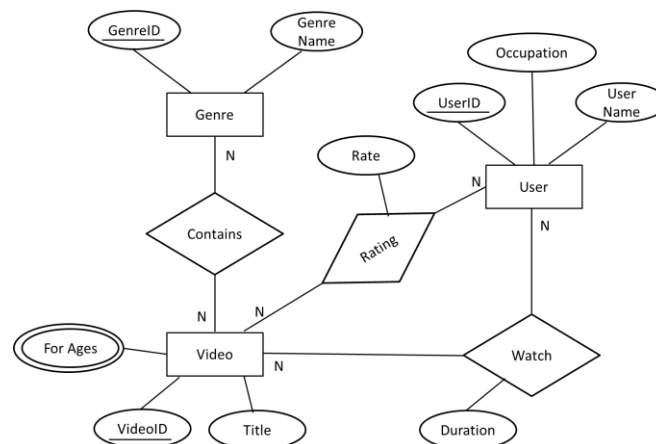


Figure 2. An E-R Diagram of the Database System for the VOD [4]

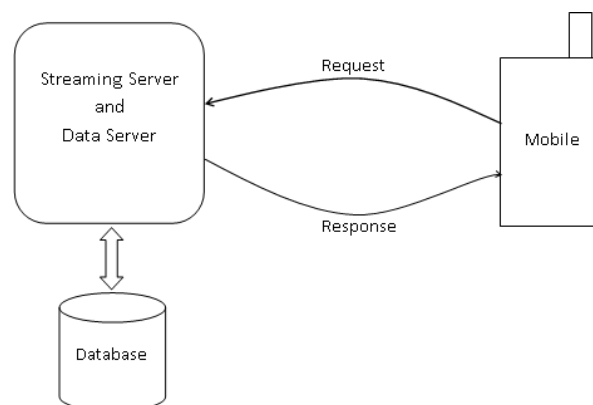


Figure 3. A Schematic Diagram of the VOD System [4]

The interactive player is comprised of three classes as shown in Figure 5. MediaDirectory: Referring to the database, MediaDirectory shows the list of genre names.

MediaList: Given a genre name, it lists all video files that are classified into that genre.

MyMediaPlayer: Plays a video from video streaming. When a hotspot is touched, it finds the URL of the video associated with the hotspot, referring to the metadata. Then it recursively invokes itself with the URL. When the Back Key clicked, it invokes itself with the URL of the back video view so that it continues playing the back video.

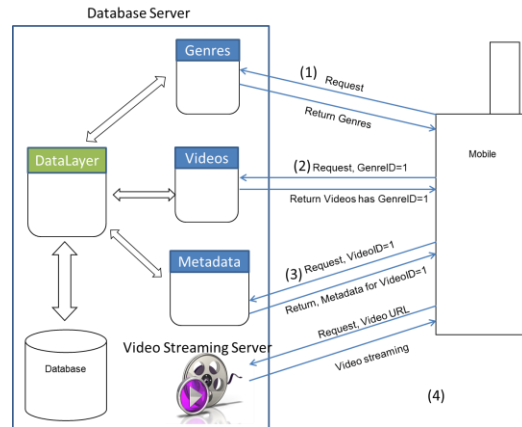


Figure 4. Mobile VOD Service Scenario

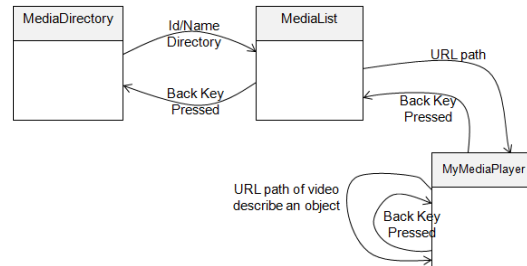


Figure 5. Main Components of the Interactive Player

A campus guide has already been developed [2]. The service scenario of the guide is summarized as follows [2]:

- 1) When users download the application, they have to provide their personal information to the system so that the system stores it in the database.
- 2) The first thing the application does, whenever the user executes it, is rendering the campus map.
- 3) The application displays the picture whenever the user takes a picture and also recognizes which building is taken.
- 4) The application overlays a text on the picture. The text describes important information related to the picture.
- 5) If the user clicks the “play video” button, then it plays the video associated with the recognized building.
- 6) When playing the video is done or the user quits playing, it goes back to rendering the campus map step.

3. Design of a Campus Guide

We are designing a campus guide that recognizes watching status, identifies the building the user is watching, and plays a video related to the building. While playing the video, if a user touches a hotspot on the screen, it stops playing the video and starts playing a new one that is related to the hotspot. This is a kind of VOD service. Implementation of a VOD service system is introduced in [4]. Development of an interactive player is introduced in [3] and development of a campus guide is introduced in

[2]. Besides, a mobile application recognizing not moving status is introduced in [10]. Making use of these developed systems we can quickly implement the enhanced campus guide.

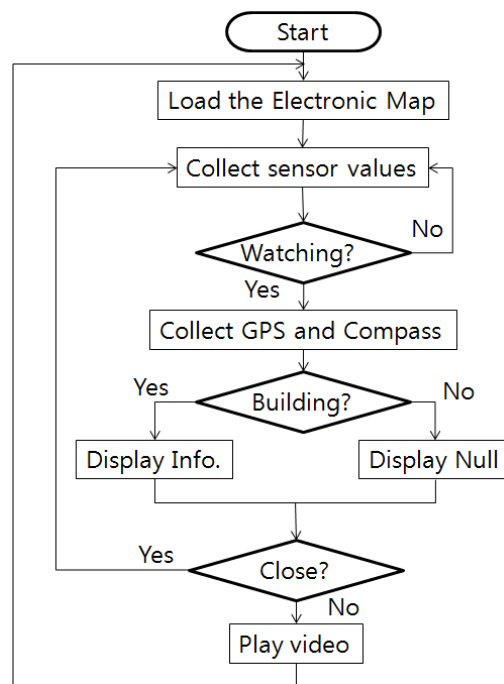


Figure 6. The Process of Our Enhanced Campus Guide

Only one problem we have to solve is watching status recognition. Watching status is defined by the status in which a user holds still his smart phone aiming at a building. When we hold a Galaxy S (an Android smart phone) aiming at a building, the pitch reads about -90 (between -90 to -120). Therefore, if a user is in watching status, then the pitch value should be about -90 and in 'not moving' status. The reverse may not be true, but it ('pitch is -90' and 'not moving' yet the user is not watching) extremely rarely happens in practice. So, we apply the following rule to detect watching status:

$$|pitch + 90| < 30 \text{ and 'not moving'} \text{ implies 'in watching status'} \quad (1)$$

The process of our enhanced campus guide is described in Figure 6. It collects sensor values after loading the electronic map. Then, it determines whether the user is in watching status or not by using (eq. 1). If it is not in watching status, then it goes back to the sensor value collection step after sleeping 100 milliseconds, otherwise it moves to the next step. In the next step, it takes a picture and identifies the building in the photo. If the recognition is successful, then it displays the picture and multimedia information associated with the building. After that, if the user clicks video play button then it starts playing video, otherwise it goes back to the 'collect sensor values' step.

4. Implementation

Implementation of our system is straight forward. As the streaming server, we can install a Darwin streaming server or a Wowza streaming engine. MySQL or Oracle can be installed as the database server. Each of the functional requirements can be implemented as a web service using Netbeans IDE. For the web server, Windows Internet Information Server (IIS) can be used. The structure of the database of our VOD system is shown in Figure 7. Part of the RESTful web services are shown in the class diagram provided by IDE in Figure 8.

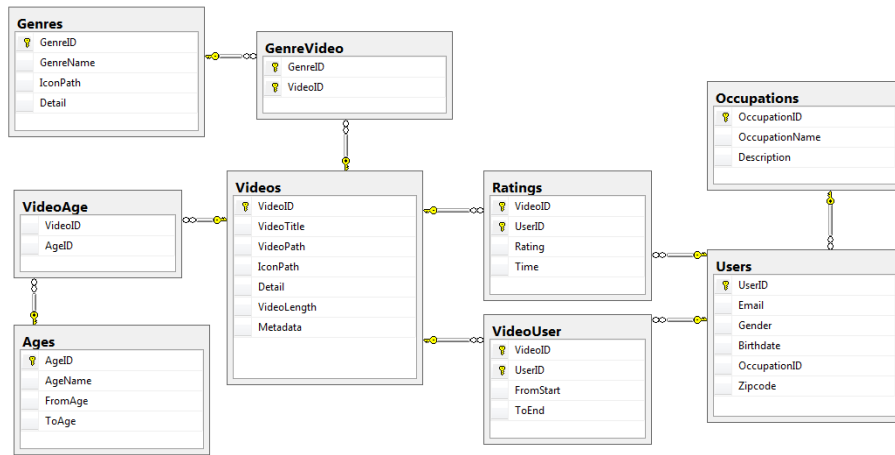


Figure 7. The Structure of the Database System

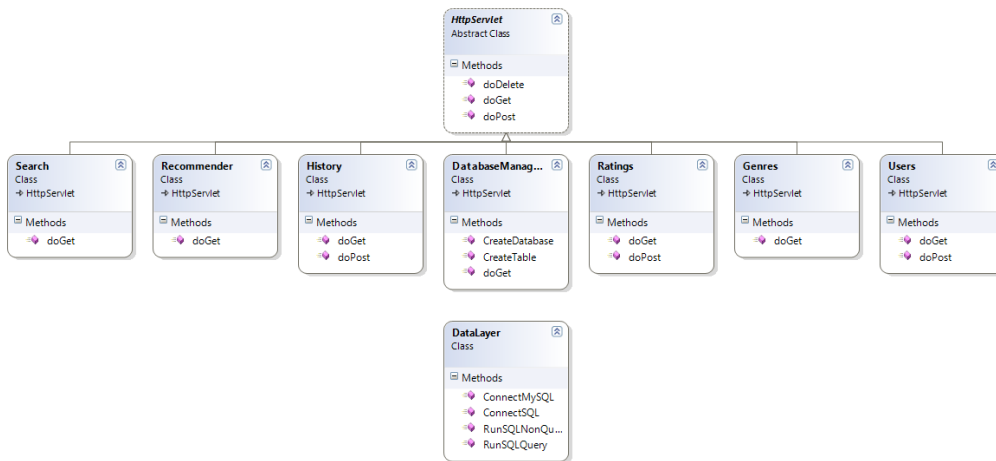


Figure 8. The Implemented Restful Web Services

On an android application, we use HttpClient, HttpGet, HttpResponse, HttpPost to make a request to a RESTful web service. For example, we use HttpGet to activate ‘Directory Service’ to obtain a list of genre names. In order to activate ‘Rating Service’ or ‘History Service’, we use HttpPost. The following is the sentence to activate ‘Create Database Service’ to create a database named TaoCSDL2: `http://localhost:8080/InteractiveMediaPlayer/DatabaseManagement?MethodName=createDatabase&Parameters=TaoCSDL2`

Similarly, we use the following sentence to activate ‘Create Table Service’ to create a table named TbNew that consists of ID and Contents:

`http://localhost:8080/InteractiveMediaPlayer/DatabaseManagement?MethodName=createTable&Parameters=TbNew|$|ID|$|int|$|Contents|$|nvarchar(100)|$|`

Typical test results of ‘Create Database Service’ and ‘Create Table Service’ are shown in Figure 9. Those results show that both ‘TaoCSDL2’ and ‘TbNew’ already exist and suggest using another name.

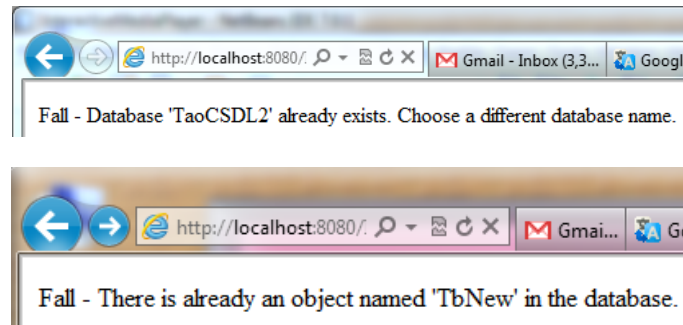


Figure 9. Typical Test Results of 'Database Management Service'

The first step of the process executed by the proposed campus guide is drawing the campus map. We need a Web service that returns a list of all lines in a given electronic map. Using this Web service, the mobile app can easily draw the campus map on the user interface.

At the second step, it collects sensor values. Accelerometer, gyroscope, light, magnetic_field, orientation, pressure, proximity, temperature, rotation_vector, gravity, and linear_accelerometer are the sensors installed on a smart phone.

At the third step, using the equation 1, it determines whether the user is in watching status or not. If it is not watching status then it goes back to the second step, otherwise it moves on to the next step.

At the fourth step, it collects GPS and compass data. At the fifth step, it identifies the building the user is watching with the algorithm described in [2]. The result of the fifth step is displayed on the screen at the sixth step. At this moment, the user can choose to go back to the second step or to go on to the next step to watch a video related to the building. A video can be associated with a metadata file. In a metadata file, the size of the window, interesting objects and occurrences of objects are described.

5. Experiments

The moving or not moving status recognition process collects accelerometer value every 100 milliseconds and puts them in array S . By investigating recent 10 accelerometer values, *i.e.*, $S[0]$ to $S[9]$, it determines whether it is moving status or not. The process calculates differences between $S[i]$ and $S[i-1]$: $d[i] = S[i]-S[i-1]$. Then the process counts the number of $d[i]$ which is less than a threshold $th2$. If the count is greater than another threshold $th1$ then the process concludes that the user is in 'not moving' status. Therefore, the accuracy of the process depends on the values of $th1$ and $th2$. We tested the process with various values of $th1$ and $th2$. The process perfectly recognizes 'not moving' status when $th1=0.2$ and $th2=0.5$ (50%). If it is in 'not moving' status and $|\text{pitch} + 90| < 30$ then it concludes that the user is in "watching status" [10].

The accuracy of the building recognition process is discussed in [2]. With the test results discussed in the references [2, 10], we can expect that the building recognition process will perfectly recognizes the building the user is watching. A screenshot of the Android app is shown in Figure 10.



Figure 7. A Screenshot of the Campus Guide. It Detects Watching Status and Takes a Picture of the Building Automatically

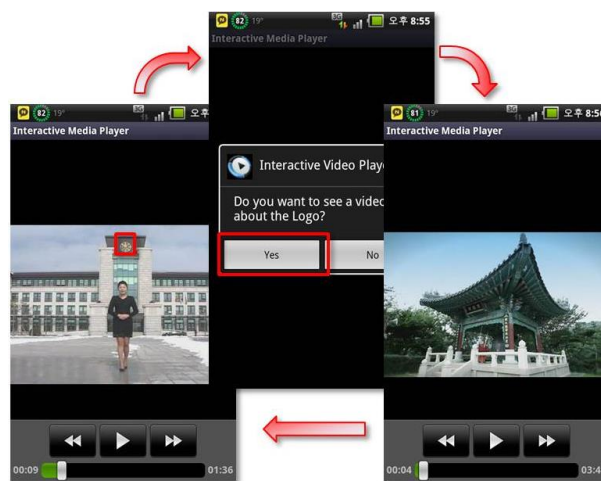


Figure 8. Screenshots off Interactive Player

If the user clicks 'video button' on the screen in Figure 7, it starts playing a video related to the building. The first image of Figure 8 is a screenshot of the video. If the user touches the logo on the screen, then the app shows buttons so that the user can choose to watch a new video about the logo or not (the second image of Figure 8). If the user chooses to watch a video about the logo, then it starts playing the video about the logo (the third image of Figure 8). After playing the video about the logo, it goes back to play the video related to the main building.

6. Conclusion

The campus guide Android app introduced in [2] recognizes the building in the picture when the user takes a picture of it. The app introduced in this paper recognizes the building the user is watching even before the user takes a picture of it. Another novel feature of this app is that its video player is interactive in that users can touch screen in order to watch another video that is germane to the object that is touched.

Acknowledgments

This research was supported by the Dongguk University Research Fund of 2016, by the Ministry of Education (NRF-2011-0006942), and by the Ministry of Knowledge Economy (10037393).

References

- [1] J. Zhou and J. Riecki, "Context-Aware Pervasive Service Composition", Proceedings of the International Conference on Intelligent Systems, Modelling and Simulation (ISMS), Liverpool, United Kingdom, (2010).
- [2] J. Yim, J. Joo, G. Lee and K. Shim, "Design and Implementation of a Smart Campus Guide Android App," Applied Mathematics & Information Sciences An International Journal, vol. 8, no. 1L, (2014), pp. 47-53.
- [3] G. Lee, J. Yim and C. Han, "Implementation of an Indoor-Location-Based Interactive VoD Player Using Web Services," Information, vol. 16, no. 2B, (2013), pp. 1309-1316.
- [4] J. Yim, G. Lee and T. Le, "Design and Implementation of VOD Database System," International Journal of Multimedia and Ubiquitous Engineering, vol. 7, no. 2, (2012), pp. 503-508.
- [5] K. Shirahama and K. Uehara, "A Novel Topic Extraction Method based on Bursts in Videos Streams," IJHIT, vol. 1, no. 3, (2008), pp. 21-32.
- [6] B. Chow, "A Very Low Bit Rate Video Coding for Ubiquitous Services," IJHIT, vol. 1, no. 4, (2008), pp. 1-8.
- [7] B. Kang, "A Review on Image and Video processing," IJMUE, vol. 2, no. 2, (2007), pp. 49-64.
- [8] A. Mishra, "Quality of Experience Based Policy Control Framework for RTP Based Video and Non-Voice Media Sessions," IJFGCN, vol. 4, no. 2, (2011), pp. 81-90.
- [9] L. Lee, H. Chang, G. Tao and S. Yang, "A Load Balanced PC-Cluster for Video-On-Demand Server Systems," IJGDC, vol. 1, no. 1, (2008), pp. 63-70.
- [10] J. Yim, G. Lee, K. Shim and T. Le, "A Method of determining whether a smart-phone is moving," Journal of Korea Multimedia Society, vol. 15, no. 5, (2012), pp. 632-638.
- [11] Darwin Streaming Server, (2012) <http://dss.macosforge.org>

Author



Jaegeol Yim, received an MSc and PhD in Computer Science from the University of Illinois at Chicago, in 1987 and 1990, respectively. He is a Professor in the Department of Computer Engineering at Dongguk University at Gyeongju Korea. His current research interests include Petri net theory and its applications, location-based services, AI systems, and multimedia systems. He has published more than 50 journal papers, 100 conference papers (mostly written in the Korean Language), and several undergraduate textbooks.