International Journal of u- and e- Service, Science and Technology
Vol.8, No. 7 (2015), pp.313-320
http://dx.doi.org/10.14257/ijunesst.2015.8.7.32

Design and Implementation of Campus Information System with Android and Web Dual-mode Based on MVC Pattern

Feng Jian[1], Ren Jian[2], Fan Ren-Yi[3] and Lei Jing[4]

[1]*College of Computer Science & Technology, Xi'an University of Science and Technology, Xi'an 710054, China*
[2]*Alibaba Group, Hangzhou 311121, China*
[3]*Kingsoft Corporation Limited, Zhuhai 519015, China*
[4]*YY Inc., Guangzhou 510655, China*
fengjian@xust.edu.cn

Abstract

Mobile application for campus is an integral part of the construction of wisdom campus. CampusService - a campus information service system is designed, and is comprised of three parts, Android client, Web client and server, which are all designed according to MVC pattern. C/S architecture is adopted between the Android client and the server, B/S architecture is adopted between the Web client and the server, and JSON is used as a unified data interface for accomplishing the interactions. The architectures of three parts are described, and key technologies are represented, including database access optimization, Android display, transaction management and authentication, among which database access optimization technology has two respects: using C3P0 to maintain connection pool and prefetching data based on their popularity. According to testing, the system implements functions of user management, intramural announcements, interaction with friends, activity participation, and campus maps, and it has good performance. The system provides an efficient platform for real-time communication among teachers, students and school authority.

Keywords: *Wisdom campus, MVC, Android, Dual-mode, Prefetching*

1. Introduction

In recent years, the world has entered a phase of vigorous development of mobile Internet. Tablet PCs, smart phones and other kind of mobile intelligent terminals became increasingly popular, traditional desktop applications began to shift to mobile devices.

On this background, mobile wisdom campus system came into being. It mainly refers to applications using wireless network technology in the campus. Based on intelligent terminals, 3G networks, Wi-Fi and other mobile technology, it can provide a variety of mobile information services which are convenient for students, teachers and administrators to use anytime and anywhere. In 2008, Stanford University began to construct mobile campus system. Through it, students can check course catalog, school maps, game information and information about people in campus directly on mobile intelligent terminals supporting iOS system. Duke University, Carnegie Mellon University and other universities have also launched their own mobile applications. In China, Fudan University had built a mobile campus application "iFudan" in 2009 [1], and was closely followed by other universities.

In typical smart phone systems, Android occupies large market share due to its openness and excellent services provided by Google, has a huge space for development. To settle problems encountered in school life conveniently, CampusService - an campus information service system is implemented in the paper. By using the system, the

ISSN: 2005-4246 IJUNESST

efficiency of information dissemination in campus and the construction of campus intelligent are improved.

2. System Design

2.1. System Target

CampusService is an Android and Web dual-mode information service system; it provides Android client for teachers and students to receive information released by school authority and to interact with each other; and provides Web client for the school authority to publish information or organize activities, and to collect feedback from teachers and students. At the same time, as an information service system, it needs to have fast response and high stability.

According to resource accumulation and user demand of campus informatization in Xi'an University of Science and Technology, main functions of CampusService are set as follows:

Table 1. System Functions

Functions	Description
Fresh news	To provide function of user attention, and receive dynamic of attended users
Public announcement	To provide function of publishing, receiving, reading and searching for campus bulletin
Interaction on line	To provide function of friends' management, allowing users to view basic information of friends, upload pictures, and interact in real-time
Activities participating	To provide function of publishing activities and user participation
Campus map	To provide function of viewing campus maps, checking news of the unit at which one is currently locate, and positioning and navigation services in campus

2.2. System Architecture

The physical structure of the system is shown in Figure 1.

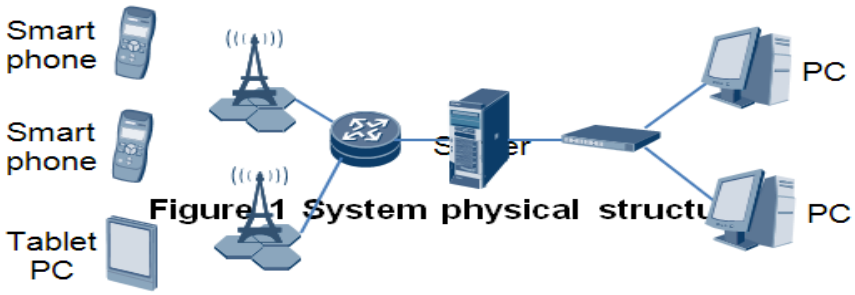


Figure 1 System physical structu

Figure 1. System Physical Structure

CampusService is Java EE project software based on both B/S and C/S mode, including three components: Android client, Web client and server. Android client achieves the same functionality as Web client, and the two kinds of clients use unified data interface JSON to interact with the server, so that the server can handle requests from both clients in a unified way. Another reason for selecting JSON to realize data exchange is to reduce the flow of data transmission. The system uses four-layer architecture, containing presentation layer in client side, presentation layer in server side, service logic layer in server-side, data model layer in server-side, shown in Figure 2.

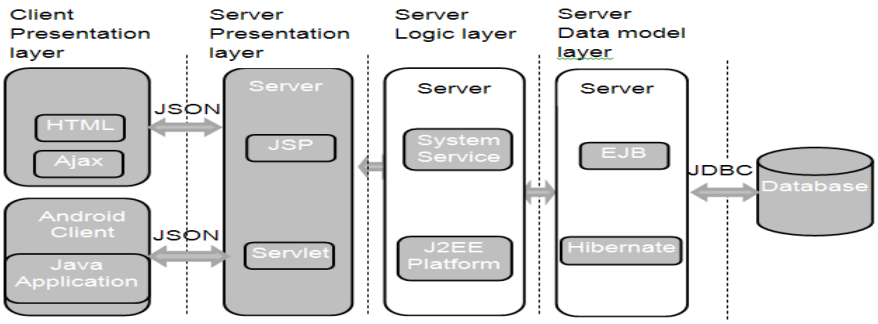


Figure 2. System Architecture

C/S architecture is used for interaction between Android client and the server, and B/S architecture is used for interaction between Web client and the server, so the overall design uses a combination of the two architectures - namely extension of B/S architecture [2]. The following are main idea of design of the server, Android client and Web client.

2.3. Server Architecture

Server processes and responds to requests from Android client and Web client, manages and preserves system data and application data. Its design pattern is MVC + DAO, uses struts2 to implement a hierarchical structure and Hibernate to realize data persistence, and uses Tomcat as Web server and MySQL as database, shown in Figure 3.

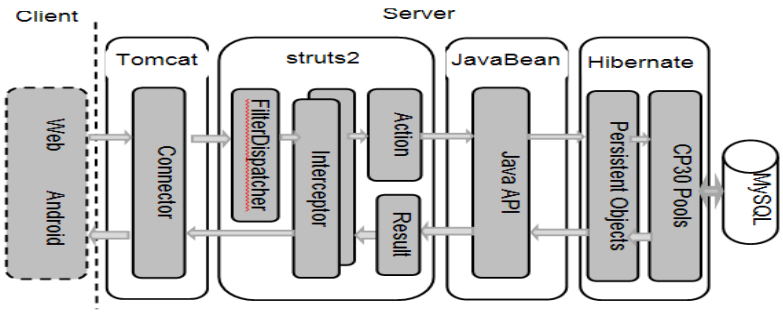


Figure 3. Server Architecture

The server implements a layered architecture using MVC pattern.

- Based on Tomcat, presentation layer is responsible for receiving JSON stream sent by clients and passing to struts2, and turning the results from struts2 into JSON stream and returning back to Android or Web client;
- In struts2, uses FilterDispatcher as controller to filter requests from clients, and then after transcoding, authentication and other pretreatment by multiple interceptors, passes requests to related action, and the action calls for JavaBean (as Model) to complete business logic treatment; corresponding actions are defined for each functional module, and each JavaBean class corresponds to a specific action to handle the business logic;
- Transaction management object in Hibernate completes CRUD operations in database; C3P0 architecture is used to take over data source connection.
- Business processing results are returned to the presentation layer in recursive order.

Data table designed in the server including: ①User table. Contains student number (work number), nickname, user groups, passwords and other user information. Where the student number (work number) is the only identity of user; ②News table. It is data sheet for news issued by user, contains news ID, author ID, release time, contents, comments,

pictures and other information; ③Activity table. It is data sheet for activity information, contains activities and their registration information; and ④Place table. It is designed to save address information, stores locations and the latest news related to locations.

2.3. Android Client

Using MVC design pattern, Android client designs specifically to a layered structure, dividing into presentation layer, control layer and model layer, shown in Figure 4.

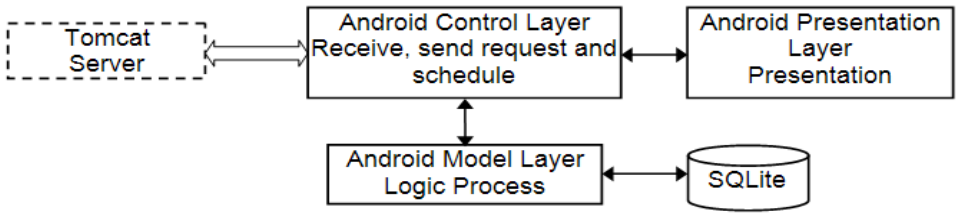


Figure 4 Android Client Architecture

Presentation layer consists of several Activities and XML layout files, interacts with users through a visual interface, and completes display of information. Model layer uses GSON packet parsing server provided by Google to transfer JSON data, to achieve parsing and processing of data; uses SQLite library coming with Android to complete storage of local data. Control layer is a bridge between the presentation layer and the model layer to control flow of data. Specific process is: Android client receives message sent by the server through the control layer, classifies the message and sends to the model layer after sorting to complete logic process. In accordance with the needs, part of data is stored in SQLite, and part of data is used to change and display interface of presentation layer through the incident response; at the same time, after processing and treating in the model layer, users' requests from presentation layer are forwarded to the control layer, and the latter sends requests to the server.

2.4. Web Client

The main work of Web client is to interact with the server, and update web pages. It uses jQuery framework in Ajax which is using JavaScript for scripting language [3], uses JSON as data format while interacts with the server. The whole processes of basic interaction are: loads jQuery when the web pages are loaded; user operate interface and jQuery binds events according to the operations; reads JSON data from the server, parses and processes JSON data to promote partial refresh of the Web pages.

3. Key Technologies

3.1. Optimization of Database Access

In the process of handling business logic, frequent database operations are needed. When the concurrent accesses are very high, operation of the database will become a performance bottleneck. To improve the performance of the server under high concurrency, multi-layered optimization is used for database access.

3.1.1. Establishing a Connection Pool with C3P0: Each access to database in traditional JDBC needs to go through a series process including building data source, processing transactions, releasing data sources, and this causes waste of resources. To settle this problem, DAO layer in CampusService uses open source framework C3P0 to establish and maintain a database connection pool, and the connection pool maintenances many

active physical connections. When the system requires data access operations, takes a connection directly from the pool to use, does not release the connection after the operation is completed but put the connection back to the pool for later use to improve response rate of the server. Adjusts the configuration of C3P0 according to stress testing results of Webbench test software.

3.1.1. Prefetching Web Pages based on their Popularity: In order to improve hit rate of Web pages and to further improve the response speed of the server, prefetching policy based on popularity of Web pages is designed.

Based on spatial locality principle of Web access, through the analysis of current and historical page requests, Web prefetching techniques proactively anticipate Web pages which may be browsed next by user, and fetch these Web pages to the local cache in advance so that users only need to access these pages from the local cache. A typical Web prefetching technique is PPM [4], it is a context model. The basic idea of PPM is to describe the user' access pattern by Markov forecast tree, and to predict future user access via sequence matching. The model has a lot of extended studies [5, 6], and PPM model based on the popularity is used for reference in this paper [6]. A new Web prefetching model based on popularities of Web page is proposed. Popularity of Web page is calculated according to many factors, including connection relationship between Web pages, freshness, grade level, and frequency of access. The model prefetches Web pages which have the highest popularities to the buffer of connection pool in database to response to user requests fast.

The model consists of the following two parts:

① To calculate popularity of a Web page. Influencing factors to popularity include:

- link relationship l: access order of related pages;
- freshness f: publishing time of Web message;
- grade level d: publisher level;
- access frequency v: accessed times of Web page.

Corresponding weight are given to these factors respectively, namely w_l, w_f, w_d and w_v, and then calculates popularity p of the Web page by weighted summation formula (1).

$$p=l*w_l+f* w_f+d*w_d+v*w_v \quad (1)$$

② To generate set of prefetching Web pages:

Class PageRankNote is defined to obtain a set of pages needed to be prefetched. The class includes five properties, namely newsId, nextPage, freshness, classify, visitCount, to represent Web page ID, the most relevant next page, freshness, grade level and page access frequency respectively. Generation processes of Top-N page set are as follows:

- According to user clicks, checks whether Web page needed by user is in the buffer whether connection pool or not. If it is, returns the Web page directly and recalculates the popularity of PageRankNote instance of the page;
- If the page required is not in the buffer, fetches the page from the database and returns back to the user, and then generates PageRankNote instance for the page and calculates its popularity;
- Compares calculated popularity to popularity of the existing pages, if it is in Top-N, then keeps it in the buffer;
- Otherwise, removes it from the buffer.

In particular implementation, in order to find the corresponding PageRankNote instance of Web page quickly, a hash table is used, where Key is newsId and Value is the corresponding instance; configures corresponding interceptors of struts2 framework, so that when the user requests a Web page, the requests can all be put to the interceptors; if the Web page requested by the user belongs to Top-N, returns the requested Web page directly from the interceptors.

3.2. Compatible Screen Display in Android

In the design of Android client, the problem of mismatch between the different models of the screens should to be resolved. These guidelines should be followed: ① Wrap_content, fill_parent and dp should be adopted when sets the size of controls in layout file. Accordingly, using sp to define the size of text in order to better adapt to the screen size. Specific pixel values do not appear in the code of the program; ② Do not use AbsoluteLayout (obsoleted in Android1.5), RelativeLayout should be used; ③ Provide images with suitable size for different screens.

3.3. Transaction Management

In the process of handling the business logic, a business often needs to involve a number of changes to the file server or database records. If an exception occurs during the processing and one step cannot be performed, then data disunity will occur to affect the stability of the system. Hibernate itself does not have the ability of transaction management, but delegates to the underlying JDBC or JTA to implement transaction management and scheduling capabilities. The system implements transaction management based on JDBC Transaction. During the transaction process, calls the essionFactory.openSession() to return a session, and then calls session.beginTransAction() and session.commit() to pack multiple operations into a group. When an exception occurs, makes restitution operation through rollback technology to ensure atomicity of transactions.

3.4. Authentication

To ensure safety, each operation of business logic should be authenticated. The system uses interceptor of struts2 to achieve a unified authentication. In configuration file struts.xml, defines <interceptor name="CheckUser" class="interceptor.CheckUser"/> and adds it to interceptor stack of struts2. When the user request arrives, CheckUser interceptor intercepts identity field to authenticate, and if authentication is successful then amputates the identity field and sends it to Action, or directly returns validation failure.

4. Testing

To test the functionality and performance of the system, a network test environment is built, in which the server is Windows 2003 + Tomcat 6.0 + MySQL 5.1; version of Android in intelligent terminals should be Android 2.1 or above.

4.1. Functional Testing

Functional testing cover full functionalities of the Android client and the Web client. The Android client is facing staff and students, while the Web client is facing departments in school. Test items and results are shown in Table 2.

Table 2. Functional Testing

Functional Items	Android Client	Web Client
User registration	Succeed if user information is correct; fail otherwise	Succeed if user information is correct; fail otherwise
User Login	Succeed if user information is correct; fail otherwise	Succeed if user information is correct; fail otherwise
Dynamics publishing	Pass	Pass
Dynamics receiving	Pass	Pass
Dynamics replying	Pass	Pass
Announcement publishing	Pass	Pass
Activities launching	Pass	Pass
Activities participating	Pass	Pass
User searching	Pass	Pass
Maps display	Pass	Pass
Campus navigation	Pass	Pass

4.2. Performance Testing

The well-known stress test tool Webbench is used to carry on performance testing, it can simulate instantaneous concurrent accesses of large numbers of users to the server to test its stability and compressive strength, at the same time packet capture tool WireShark is used to grab packets and verify the validity of the testing.

In the testing, after the server was enabled, Webbench submitted a large number of concurrent requests by simulating 100 clients within 30 seconds, shown in Figure 5. There were a total of 15,834 packets requested, including 15,834 successful requests, 0 failure, and the success rate was 100%; grabbing packets by WireShark showed the average data flow was more than 10,000 packets/s within the duration of 30 seconds; all status codes of HTTP packets were 200OK, that meant the server had responded to all requests successfully.

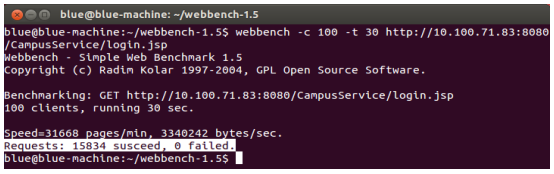


Figure 5. Testing Results of Webbench

5. Conclusions

In The Twelfth Five-Year Guideline planed early in 2010, Zhejiang University proposed a program to build exciting "wisdom campus", one of its features is to integrate information services provided by computer networks to various services of school, to achieve interconnection and collaboration. Campus information service system CampusService designed in the paper based on dual-mode of Android and Web, is to transfer service information among students, teachers and the management departments representing the school authority by the use of computer networks and mobile communication networks. The system completes design, development and deployment of functionality and framework including Android client, Web client and server. The testing showed that the system had achieved functions of dynamic information sharing, participation in school activities, user searching, rich campus, and so on, it could work well and its performance is stable.

The system has a great potential for development. In the future, through improving and expanding, it can rapidly deploy many other services such as enrollment, payment, etc. to provide strong support for teach, research, management and decision-making of the school, and help to truly realize rich campus life in mobile Internet era.

Acknowledgements

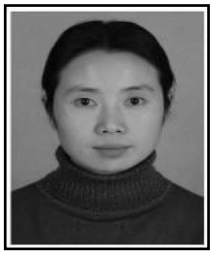
This work was supported in part by Shaanxi Provincial Natural Science Foundation Project (No. 2012JQ8030) and National Training Program of Innovation and Entrepreneurship for Undergraduates (No. 201310704021).

References

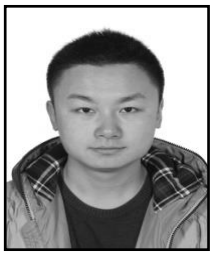
[1] K. Zhang, Y. Mi, H. Yan, W. Xiang, B. X. Liu and Q. J. Wu, "iFudan - Mobile Application Development and Practice of Fudan University", Journal of Wuhan University (Natural Science Edition), vol. 58, no. S1, **(2012)**, pp. 265-270.
[2] G. Q. Li and X. G. Li, "Research on Mobile Platform-Oriented Web Information Interaction Model", Computer Applications and Software, vol. 28, no. 1, **(2011)**, pp. 176-180.
[3] Z. H. Wang, Y. L. Lu and J. H. Liu, "On AJAX-Based Web Service Framework and Security", Computer Applications and Software, vol. 28, no. 3, **(2011)**, pp. 296-299.

[4] T. Palpanas and A. Mendelzon, "Web prefetching using partial match prediction", Proceedings of the 4th Web Caching Workshop, **(1999)** March 31-April 1, San Diego, USA.
[5] M. Deshpande and G. Karypis, "Selective Markov models for predicting Web page accesses", ACM Trans on Internet Technology, vol. 4, no. 2, **(2004)**, pp. 163-184.
[6] L. Shi, Y. Zhang, Y. X. Pei and Z. M. Gu, "PPM Prediction Model Based on Web Object's Popularity", Journal of Chinese Computer Systems, vol. 27, no. 7, **(2006)**, pp. 1378-1382.

Authors



Feng Jian, August, 1973. She received her doctoral degree of Computer Software and Theory from Northwest University, Xi'an, China, in 2008. And research interests on computer network and communication, network security, distributed computing. She is currently an Associate Professor with College of Computer Science & Technology, Xi'an University of Science and Technology, Xi'an, China.



Ren Jian, August, 1993. He received his bachelor degree of Network Engineering from Xi'an University of Science and Technology, Xi'an, China, in 2015. And research interests on Server-side development, Android software development, computer network and communication.

He is currently worked as an Android software development engineer in Alibaba Group.

Fan Ren-yi, August, 1991. He received his bachelor degree of Network Engineering from Xi'an University of Science and Technology, Xi'an, China, in 2015. And research interests on computer network and communication, network security, distributed computing. He currently works in Kingsoft Corporation Limited. And he is interested in Android Development.

Lei Jing, February, 1992. She received her bachelor degree of Network Engineering from Xi'an University of Science and Technology, Xi'an, China, in 2015. And research interests on computer network and communication, Web development.

She is currently worked as a Web application development engineer in YY Inc.

