DOI: 10.21276/sjet

**Scholars Journal of Engineering and Technology (SJET)** **ISSN 2321-435X (Online)**
**Sch. J. Eng. Tech., 2017; 5(8):440-446** **ISSN 2347-9523 (Print)**

(An International Publisher for Academic and Scientific Resources)
www.saspublisher.com

Research Article

# Efficient Mobile Student Management System Based on Cloud Computing

**Can Zhou, Haibo Yi***
School of Computer Engineering, Shenzhen Polytechnic, Shenzhen, China

***Corresponding author**
Haibo Yi
Email: haiboyi@126.com

**Abstract:** Student management have been playing more and more important roles at institutions of higher education. In order to enhance student growth and development, an efficient mobile system for college student management is proposed. The student management system includes two parts, i.e. client and server. Client is designed based on Android system and server is designed based on OpenStack cloud computing system. The student management system can bring information of students and campus together from diverse sources on campus in a uniform way. The student management system provides a consistent look and feel with access control and procedures for multiple applications and databases of student management. We divide the management system into five functional modules, i.e. login module, message module, management module, storage module, administrator module. Based on cloud computing, we use OpenStack, Android and MySQL to implement the system.
**Keywords:** student management; cloud computing; Android; mobile system; OpenStack

## INTRODUCTION

Student management system is a management information system for education establishments to manage student data. Student management systems provide capabilities for registering students in courses, documenting grading, transcripts, results of student tests and other assessment scores, building student schedules, tracking student attendance, and managing many other student-related data needs at institutions of higher education [1-5].

Student management systems vary in size, scope and capability, from packages that are implemented in relatively small organizations to cover student records alone, to enterprise-wide solutions that aim to cover most aspects of running large multi-campus organizations and their online schools with significant local responsibility. Many systems can be scaled to different levels of functionality by purchasing add-on "modules" and can typically be configured by their home institutions to meet local needs.

These management systems are used via computer and Internet. Thus, few of them can provide efficient and mobile services for students.

In this paper, we design an efficient mobile system for student management at institutions of higher education, which can advances student learning and development; fosters community engagement; promotes diversity, inclusion and respect; and empowers students to thrive.

The student management system includes two parts, i.e. client and server. Client is designed based on Android system and server is designed based on OpenStack cloud computing system. The student management system can bring information of students and campus together from diverse sources on campus in a uniform way. The student management system provides a consistent look and feel with access control and procedures for multiple applications and databases of student management.

We divide the management system into five functional modules, i.e. login module, message module, management module, storage module, administrator module. Based on cloud computing, we use OpenStack, Android and MySQL to implement the system.

(1) Login module: the management system needs a login module to identify roles, e.g. students, teachers and system administrators. The login module is

used as an access between the cloud system on the server and the system website on user's terminals. It is designed based on keystone, which provides a central directory of users mapped to the OpenStack services they can access.

(2) Message module: the message module is the core of the management system. It enables displaying messages of system events and other users. The message module comprises three main parts, i.e. message, message notify and message subscribe. Message provides the core of the message stack. It provides a message entity type that can be subtyped into one or more message types, a special multi-value message text field, a message view and some permissions and configuration settings. The message notify provides a system for forwarding messages. It uses a plugin system to define message notifiers. It comes with a default email notifier plugin and an SMS notifier plugin that is available when the SMS Framework module is installed. The message subscribe provides a mechanism for users to tell the system to send them notification messages when events occur.

(3) Management module: Teachers need to use the management system to process student affairs. They use their usernames and passwords to login the system. They can change their passwords to protect student information. They can use the system to add, delete, update and query student information. Students are the most important users of the management system. Students can login the system by using their usernames and passwords. They can change their passwords to protect their information. They can use the system to update their information.

(4) Storage module: We use MySQL to design entities for the student management system. MySQL is an open-source relational database management system. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is implemented on cloud computing systems.

(5) Administrator module: system administrators are responsible for the users and group administration, server configuration, and server maintenance. They are the "root" users of the systems.

By integrating the designs of five modules, we use Android and OpenStack to implement the student management system. Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. OpenStack is a free and open-source software platform for cloud computing. The software platform consists of interrelated components that control diverse, multi-vendor hardware pools of processing, storage, and networking resources throughout a data center. Users either manage it through a web-based dashboard, through command-line tools, or through an API.

The rest of this paper is organized as follows: in Section Ⅱ, we design a mobile management system for students at institutions of higher education; in Section Ⅲ, we implement the mobile student management system based on cloud computing; in Section Ⅳ, conclusions are summarized.

## MOBILE STUDENT MANAGEMENT SYSTEM

### A. Architecture Design

We divide the management system into five functional modules, i.e. login module, message module, management module, storage module, administrator module, where the hierarchical modules of the management system is shown in Figure 1.

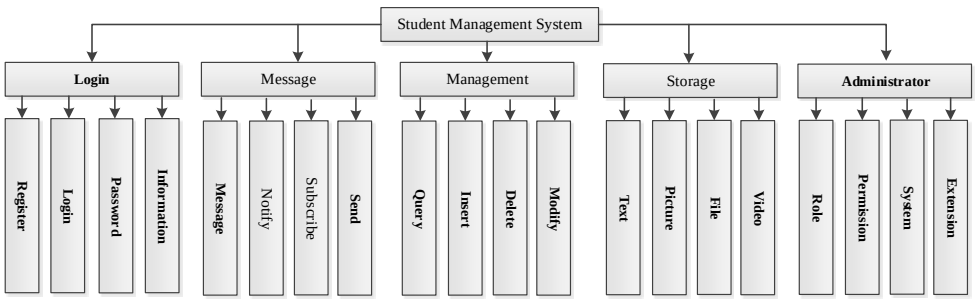


**Fig-1: Hierarchical Modules of the Student Management System**

(1) Login module: login module is used to identify user's roles, which includes register, login, password and information. The main pages include register page, login page, password page and information page. New users can register the management system with their usernames and passwords via the register page. Users use their usernames and passwords to login the management system via the login page. They can change their passwords to protect their information via password page. In addition, users can maintain their information via information page. The login module is used as an access between the cloud system on the server and the system website on user's terminals.

It is designed based on keystone, which provides a central directory of users mapped to the OpenStack services they can access. It acts as a common authentication system across the cloud operating system and can integrate with existing backend directory services. It supports multiple forms of authentication including standard username and password credentials, token-based systems. In addition, the catalog provides a queryable list of all of the services deployed in a cloud computing system in a single registry. Users and third-party tools can programmatically determine which resources they can access.

(2) Message module: the message module is the core of the management system, which includes message, notify, subscribe and send management. It enables displaying messages of system events and other users. Users can subscribe messages from specific users. When new message from the user occurs, it is delivered to subscribe users.

The message module comprises three main parts, i.e. message, message notify and message subscribe. Message provides the core of the message stack. It provides a message entity type that can be subtyped into one or more message types, a special multi-value message text field, a message view and some permissions and configuration settings. The message notify provides a system for forwarding messages. It uses a plugin system to define message notifiers. It comes with a default email notifier plugin and an SMS notifier plugin that is available when the SMS Framework module is installed. The message subscribe provides a mechanism for users to tell the system to send them notification messages when events occur.

(3) Management module: the management module is used to manage the student information, which includes query, insert, delete and modify management.

Teachers need to use the management system to process student affairs. They use their usernames and passwords to login the system. They can change their passwords to protect student information. They can use the system to add, delete, update and query student information. Students are the most important users of the management system. Students can login the system by using their usernames and passwords. They can change their passwords to protect their information. They can use the system to update their information.

(4) Storage module: the storage module is used to store data, which includes text, picture, file and video management. We use MySQL, cinder and swift to store the data.

Block Storage: Cinder provides persistent block-level storage devices for use with OpenStack compute instances. The block storage system manages the creation, attaching and detaching of the block devices to servers. Block storage volumes are fully integrated into OpenStack Compute and the Dashboard allowing for cloud users to manage their own storage needs.

Object Storage: Swift is a scalable redundant storage system. Objects and files are written to multiple disk drives spread throughout servers in the data center, with the OpenStack software responsible for ensuring data replication and integrity across the cluster. Storage clusters scale horizontally simply by adding new servers.

MySQL: MySQL is an open-source relational database management system. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is implemented on cloud computing systems.

(5) Administrator module: administrator module includes role, permission, system and extension management, which is responsible for the users and group administration, server configuration, and server maintenance. The main pages include administrator page, system page and extension page. System administrators can manage information of users via administrator page. They can manage the system via system page. In addition, they can extend the system via extension page.

**B. Cloud Computing System Design**

We implement a cloud computing system on Ubuntu 14.04 for the student management system by using OpenStack. OpenStack is a free and open-source software platform for cloud computing. The software

platform consists of interrelated components that control diverse, multi-vendor hardware pools of processing, storage, and networking resources throughout a data center. Users either manage it through a web-based dashboard, through command-line tools, or through an API.

Open Stack has a modular architecture with various code names for its components.

(1) Compute: Nova is a cloud computing fabric controller, which is the main part of a cloud computing system. It is designed to manage and automate pools of computer resources and can work with widely available virtualization technologies.

(2) Network: Neutron is a system for managing networks and IP addresses. Neutron ensures the network is not a limiting factor in a cloud deployment and gives users self-service ability. Neutron provides networking models for different applications. Standard models include flat networks or VLANs that separate servers and traffic. Neutron manages IP addresses, allowing for dedicated static IP addresses or DHCP. Floating IP addresses let traffic be dynamically rerouted to any resources in the IT infrastructure, thus users can redirect traffic during maintenance or in case of a failure.

(3) Dashboard: Horizon provides administrators and users with a graphical interface to access, provision, and automation of cloud-based resources. The design accommodates third party products and services, such as billing, monitoring, and additional management tools. The dashboard is also brand-able for service providers and other commercial vendors who want to make use of it. The dashboard is one of several ways users can interact with OpenStack resources.

(4) Image: Glance provides discovery, registration, and delivery services for disk and server images. Stored images can be used as a template. It can also be used to store and catalog an unlimited number of backups. Glance can store disk and server images in a variety of back-ends. Glance provides a standard REST interface for querying information about disk images and lets clients stream the images to new servers.

### C. Database Design

We use MySQL to design entities for student management system, which are shown in Figure 2. MySQL is an open-source relational database management system. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is used in many high-profile, large-scale websites, including Google, Facebook, Twitter and YouTube.

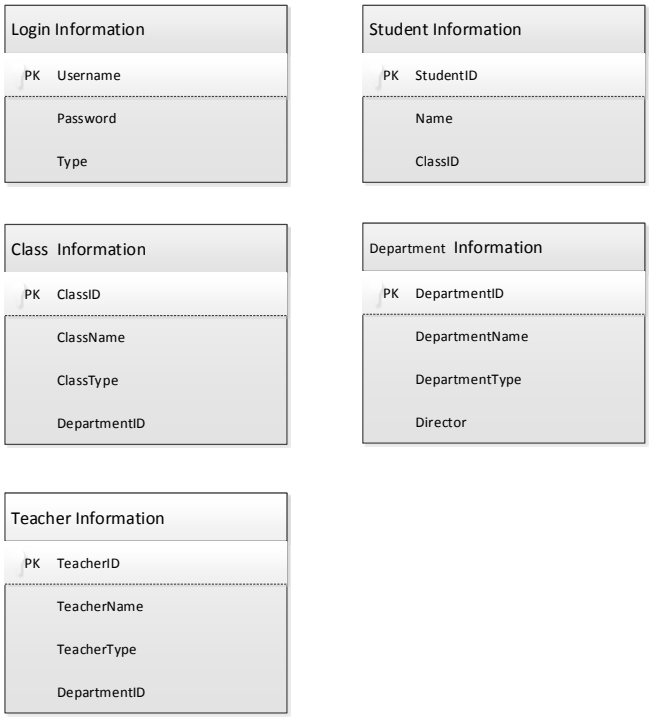


**Fig-2: Entities of the System**

The database of the student management system are designed as follows.

1) Login information table: it stores usernames, passwords and types for users, e.g. students, teachers and administrators.
2) Student information table: it stores student's ID, name, classID.
3) Class information table: it stores class's ID, name, type and departmentID.
4) Department information table: it stores department's ID, name, type and director.
5) Teacher information table: it stores teacher's ID, name, type and departmentID.

## IMPLEMENTATION

According to our system design, we implement the mobile management system for students at institutions of higher education based on Android, which includes login module, message module, management module, storage module, administrator module. We adopt MySQL for database design, Java and Android for system design, and OpenStack for cloud storage.

### D. Implementation of Login Module

Figure 3 shows the flowchart of the login module. The login module is used as an access between the cloud system on the server and the system website on user's terminals.

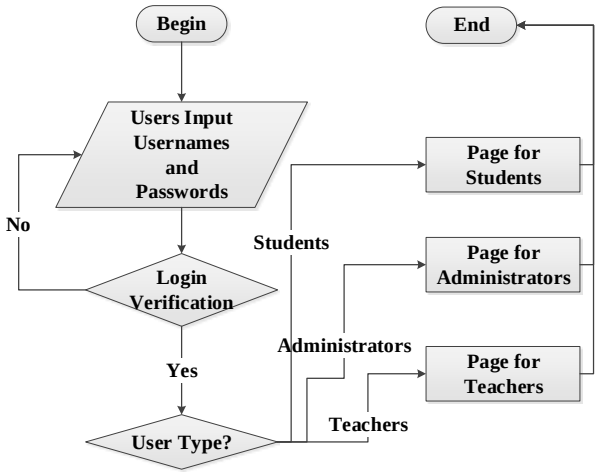


**Fig-3: The Flowchart of Login Module**

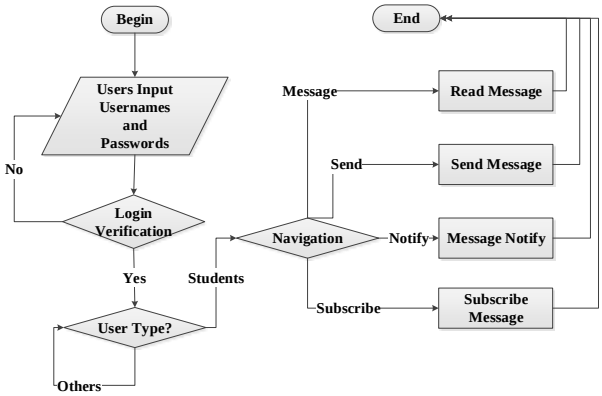


**Fig-4: The Flowchart of Message Module**

### E. Implementation of Message Module

Figure 4 shows the flowchart of the message module. The message module comprises three main parts, i.e. message, message notify and message subscribe.

### F. Implementation of Management Module

Figure 5 shows the flowchart of the management module. The management module is used to manage the student information, which includes query, insert, delete and modify management.

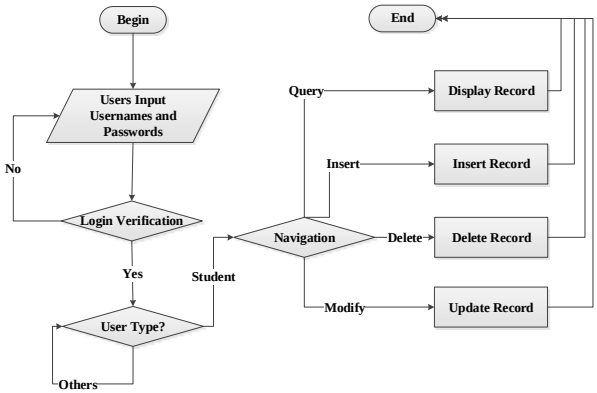


**Fig-5: The Flowchart of Management Module**

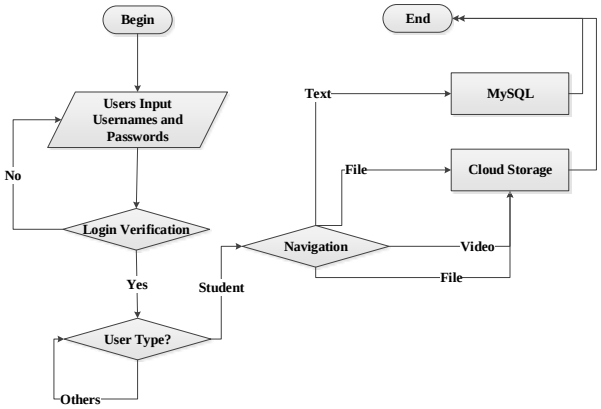


**Fig-6: The Flowchart of Storage Module**

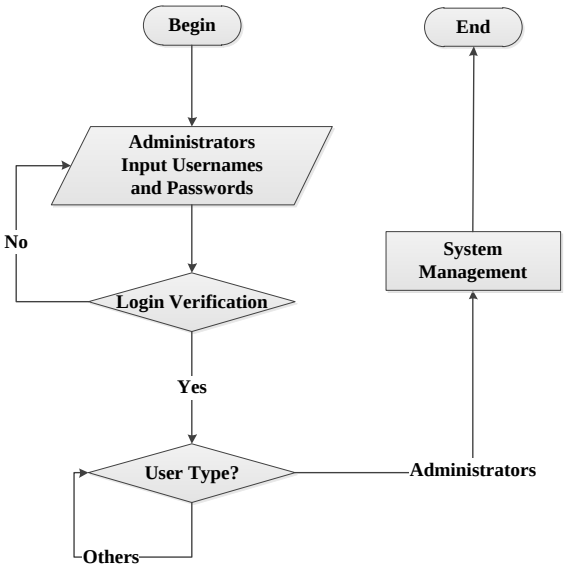


**Fig-7:The Flowchart of Administrator Module**

### G. Implementation of Storage Module

Figure 6 shows the flowchart of the storage module. The storage module is used to store data, which includes text, picture, file and video management.

### H. Implementation of Administrator Module

Figure 7 shows the flowchart of the administrator module. Administrator module includes role, permission, system and extension management, which is responsible for the users and group

administration, server configuration, and server maintenance.

## CONCLUSION

In this paper, we design an efficient mobile system for student management at institutions of higher education, which includes two parts, i.e. client and server. Client is designed based on Android system and server is designed based on OpenStack cloud computing system. We divide the management system into five functional modules, i.e. login module, message module, management module, storage module, administrator module.

Based on cloud computing, we use OpenStack, Android and MySQL to implement the system. By integrating the designs of five modules, we use Android and OpenStack to implement the student management system.

## ACKNOWLEDGMENT

This work is supported by Shenzhen Science and Technology Program under Grant (No. JCYJ20170306144219159), Science and Technology Program of Shenzhen Polytechnic (No. 601722K20018).

## REFERENCES

1. Stuppy JJ. Automated testing and electronic instructional delivery and student management system: US, US6146148, 2000.
2. Weaver D, Spratt C, Nair CS. Academic and student use of a learning management system: Implications for quality. Australasian journal of educational technology. 2008 Jan 16;24(1).
3. Smith B, Felmeister C, Galeeki L. Student interaction management system: US, US20070218446, 2007.
4. Natek S, Zwilling M. Student data mining solution–knowledge management system related to higher education institutions. Expert Systems with Applications. 2014; 41(14):6400-6407.
5. Yue ZG, Jin YW. The Development and Design of the Student Management System Based on the Network Environment. International Conference on Multimedia Communications. IEEE Computer Society. 2010;5-8.