Efficient and Automated Management for Student Affairs at Institutions of Higher Education

Haibo Yi[1,a,*] and Zhe Nie[1,b]

[1] School of Computer Engineering, Shenzhen Polytechnic, Xili Lake, Nanshan District, Shenzhen, Guangdong Province, P.R.China
[a] haiboyi@126.com, [b] niezhe@szpt.edu.cn
*corresponding author

Keywords: Student affairs, Efficient and automated management, Management systems.

Abstract. Student affairs, which enhance student growth and development, have been playing more and more important roles at institutions of higher education. An efficient and automated management system for student affairs can advance student learning and development; foster community engagement; promote diversity, inclusion and respect; and empower students to thrive. With the rapid development of the Internet technology, more and more management systems for student affairs have been designed and implemented. However, few of them can provide efficient and automated services of student affairs. In this paper, we design an efficient and automated management system for student affairs at institutions of higher education. We divide the management system into four functional modules, i.e. login module, student module, practitioner (teacher) module and administrator module. Based on three-tier B/S architecture, we use Microsoft ASP.Net and SQL Server to implement the system.

1. Introduction

Student affairs are the division of services for students at institutions of higher education [1-5]. The division which processes student affairs is known as student affairs office. The organization of a student affair office may vary based on the size and type of an institution. In most of the institutions, student affairs offices are led by a secretary or a vice secretary. People who work in this field are known as practitioners (teachers) of student affairs. These practitioners, who work in student affairs offices at institutions of higher education, are charged with the tasks of working in specific services for students, e.g. academic tutoring, assessment, research, admissions, enrollment, financial aids, scholarships, student associations, internships, orientation, alumni, campus life, counseling, health, residence life, sports and recreation.

Although institutions of higher education have been dealing with student affairs for as long as they have existed, student affairs emerged first in the late 19th century as a distinct field. Since then, there have been more and more professional organizations for student affairs at the international,

national, regional, and institutional levels, e.g. international association of student affairs and services was established in 2010. Management systems are one of the most important tools in engineering area. Most of organizations have been using management systems to manage student affairs with the aid of the rapid development of Internet technology. However, few of them can provide efficient and automated services of student affairs.

In this paper, we design an efficient and automated management system for student affairs 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. We divide the management system into four functional modules, i.e. login module, student module, practitioner module and administrator module. Based on three-tier B/S architecture, we use Microsoft ASP.Net (version 4.0) and SQL Server (version 2008) to implement the system.

The rest of this paper is organized as follows: in Section 2, we analyze the system requirement of student affairs at institutions of higher education; in Section 3, we design an efficient and automated management system for student affairs; in Section 4, we use Microsoft ASP.Net and SQL Server to implement the system; in Section 5, conclusions are summarized.

2. Requirement Analysis

In system engineering and software engineering, requirement analysis encompasses those activities that goes into determining the needs to meet for a new system. Requirement analysis is critical to the success or failure of a new system.

Student affairs at institutions of higher education have various departments, e.g. academic tutoring, assessment, research, admissions, enrolment, financial aids, scholarships, student associations, internships, orientation, alumni, campus life, counselling, health, residence life, sports and recreation. An efficient and automated management system for student affairs can advance student learning and development; foster community engagement; promote diversity, inclusion and respect; and empower students to thrive. We divide the management system into four functional modules, i.e. login module, student module, practitioner module and administrator module, which are illustrated as follows.

(1) Login module: the management system needs a login module to identify roles, e.g. students, practitioners and system administrators. The login module is used as an access between the back-end system on the server and the system website on user's terminals.

(2) Student module: students at institutions of higher education 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. In addition, they can use the management system to apply for a scholarship or a financial aid; they can look for internships by using the system; they can present applications to join a student association, etc.

(3) Practitioner module: practitioners of student affairs 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. In addition, they can review the applications for scholarships, financial aids, internships, joining student associations, etc.

(4) Administrator module: system administrators are responsible for the users and group administration, server configuration, and server maintenance. They can manage information of students, scholarships, financial aids, internships, student associations, etc.

3. System Design

3.1. Architecture Design

We design the system architecture for student affairs at institutions of higher education based on the requirement analysis mentioned in Section 2. We adopt three-tier B/S architecture of Microsoft ASP.Net (Version 4.0) in our system, which is shown Fig. 1.
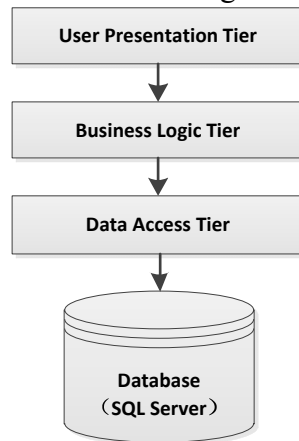

Fig. 1 Three-tier B/S Architecture of the System

Three-tier architecture is a software architecture pattern in which the user presentation, business logic and data access are developed and maintained as independent modules. It was developed by John J. Donovan in his tool company - Open Environment Corporation (OEC) in Cambridge, Massachusetts. Typically, three-tier B/S architecture includes user presentation tier, business logic tier and data access tier. The user presentation tier is the topmost level of the architecture, which displays information via web browsers (e.g. IE, Chrome). It receives requests of users (students, practitioners and administrators) and returns processing results to them via web applications (e.g. ASP. net). The business logic tier is pulled out from the user presentation tier and, as its own layer, it controls a system functionality by performing detailed processing. The data access tier provides an Application Programming Interface (API) to the business logic tier that exposes methods of managing the stored data without exposing or creating dependencies on the data storage mechanisms (e.g. SQL Server).

We choose the three-tier system architecture and divide the system into four functional modules, i.e. login module, student module, practitioner module and administrator module, where the hierarchical modules of the system is shown in Fig. 2. Login module includes register management, login management and password management; student module includes inquire, application and reception; practitioner module includes review, process application and data management; administrator module includes role management, permission management and system management.
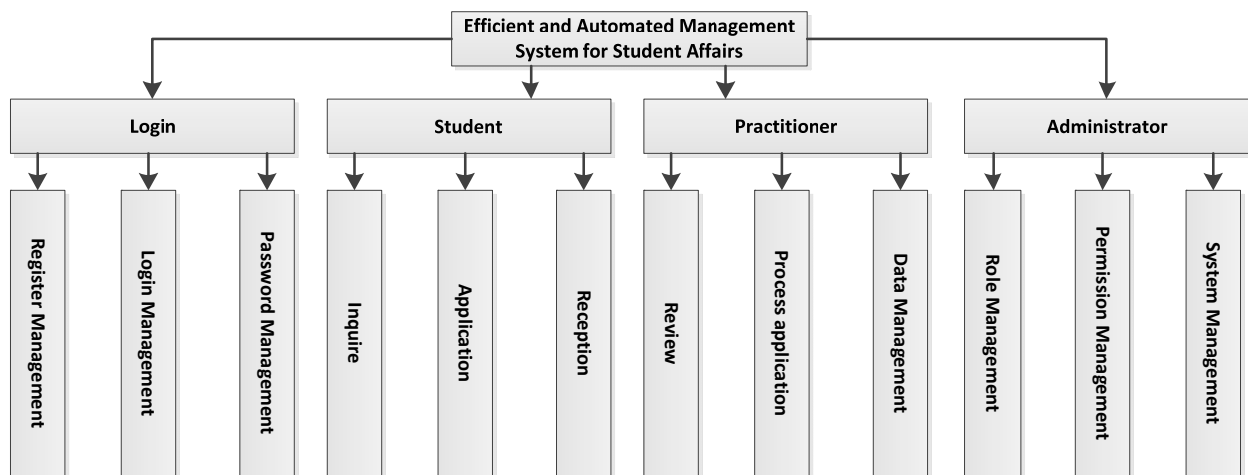
Fig. 2 Hierarchical Modules of the System

Correspondingly, Fig. 3 depicts the function of the system. Students and practitioners can use the system to process student affairs, which are illustrated as follows.

(1) Student information management: students can login the system and update their information; practitioners can use the system to add, delete, update and query student information.

(2) Scholarship management: students can use the system to apply for scholarships; practitioners can review and process the applications; administrators can add, delete, update and query scholarship information.

(3) Financial aid management: students can use the system to apply for financial aids; practitioners can review and process the applications; administrators can add, delete, update and query financial aid information.

(4) Internship management: students can use the system to apply for internships; practitioners can review and process the applications; administrators can add, delete, update and query internship information.

(5) Student association management: students can use the system to apply for student associations; practitioners can review and process the applications; administrators can add, delete, update and query student association information.
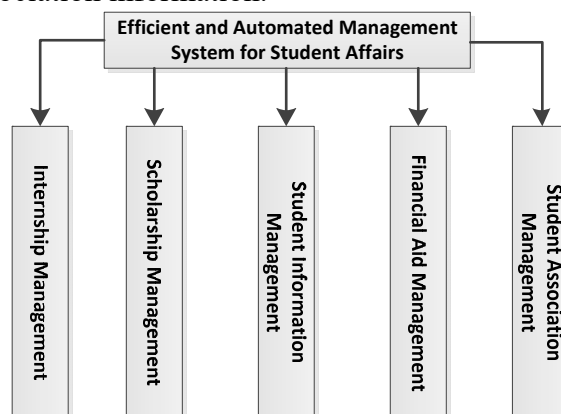

Fig. 3 Function of the System

3.2. Database Design

We use Microsoft SQL Server 2008 to design entities for the management system, which are shown in Fig. 4. Login information table stores usernames, passwords and types for students,

practitioners and administrators. Student information table stores student's ID, name and class. Internship information table stores internship's ID, name, type and student's ID. Scholarship information table stores scholarship's ID, name, type and student's ID.

4.   Implementation

According to our system design, we implement the efficient and automated management system for student affairs from four functional modules, which are the login module, the student module, the practitioner module and the administrator module. We adopt Microsoft SQL Server 2008 for database design, ASP. net 4.0 for web design and Microsoft Visual Studio 2010 for development environment.

4.1.   Implementation of Login Module

The management system needs a login module to identify roles, e.g. students, practitioners and system administrators. The login module is used as an access between the back-end system on servers and the system website on user's terminals.

The login module is the entrance of the management system. Users input their user names and passwords; once the system verifies their information, it identifies their roles and navigates them to pages of their own based on login information table. After that users can use the system. Besides, users can register an account of the system by using this module. Fig. 5 shows the flowchart of the login module.

4.2.   Implementation of Student Module

Students at institutions of higher education are the most important users of the management system. Students can login the system by using their usernames and passwords from login information table. They can change their passwords to protect their information. They can update their information based on student information table. In addition, they can use this system to apply for a scholarship (based on scholarship information table) or a financial aid (based on financial aid information table); they can look for internships by using this system (based on internship information table); they can present applications to join a student association (based on student association information table), etc. Fig. 6 shows the flowchart of the student module.

4.3.   Implementation of Practitioner module

Practitioners of student affairs need to use this system to process student affairs. They use their usernames and passwords (from login information table) to login the system. They can change their passwords to protect teacher information. They can use the system to add, delete, update and query student information (based on student information table). In addition, they can review the applications for scholarships (based on scholarship information table), financial aids (based on financial aid information table), internships (based on internship information table), joining student associations (based on student association information table), etc. Fig. 7 shows the flowchart of the practitioner module.
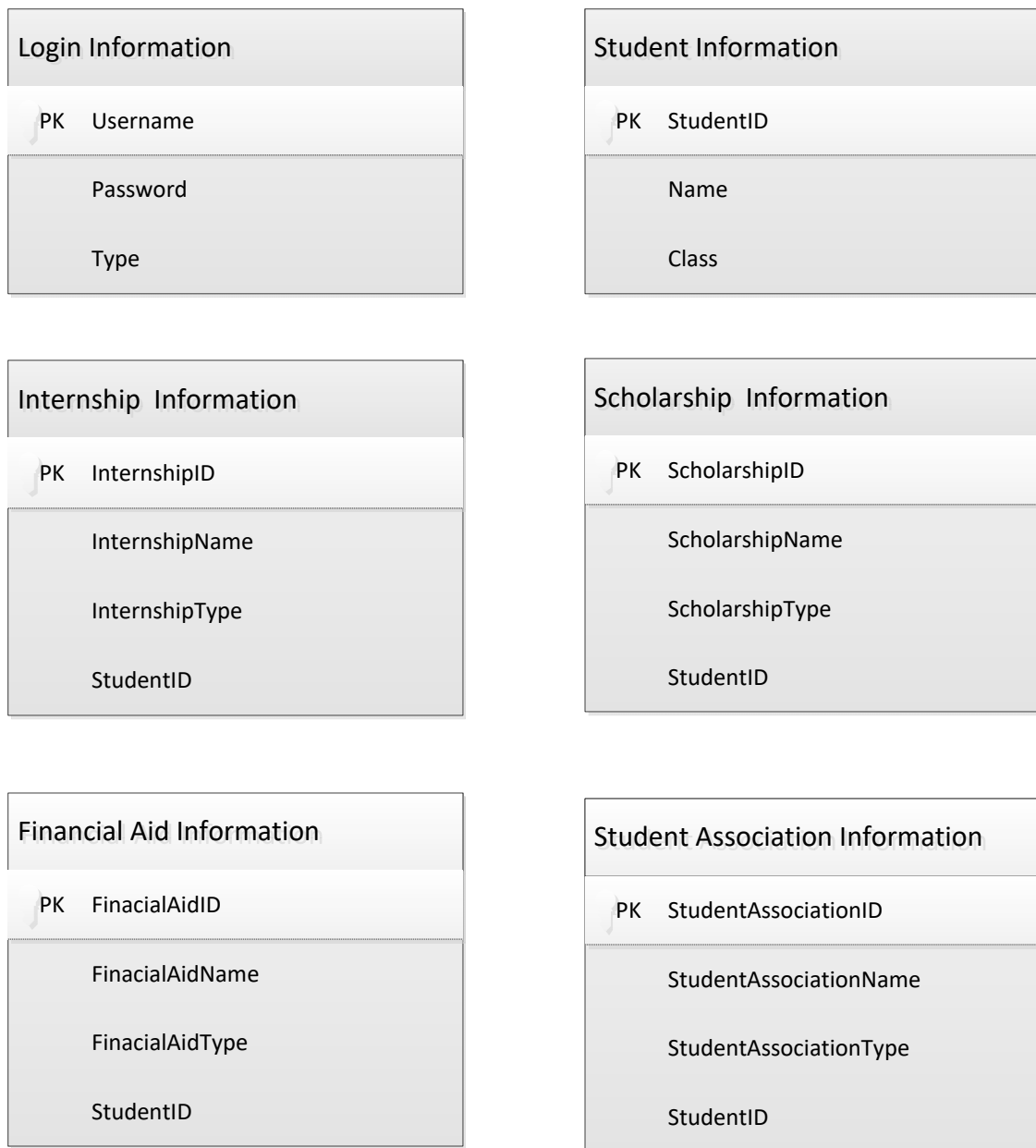
Login Information

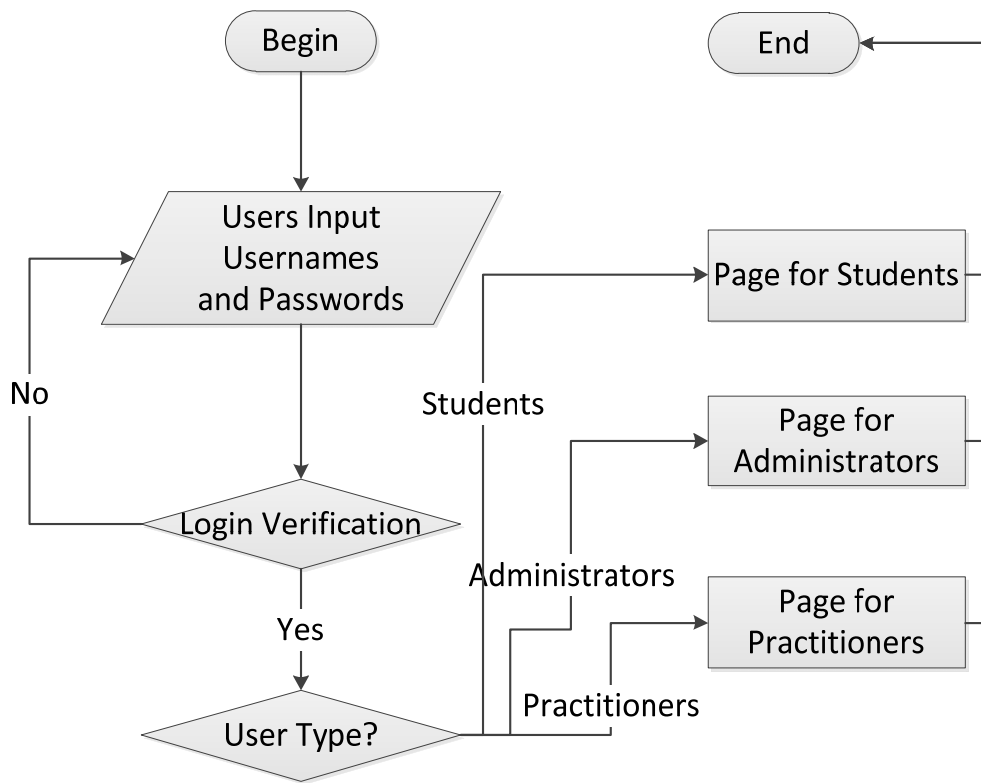|	
PK	Username
	Password
	Type

Student Information

PK	StudentID
	Name
	Class

Internship Information

PK	InternshipID
	InternshipName
	InternshipType
	StudentID

Scholarship Information

PK	ScholarshipID
	ScholarshipName
	ScholarshipType
	StudentID

Financial Aid Information

PK	FinacialAidID
	FinacialAidName
	FinacialAidType
	StudentID

Student Association Information

PK	StudentAssociationID
	StudentAssociationName
	StudentAssociationType
	StudentID

Fig. 4 Entities of the System
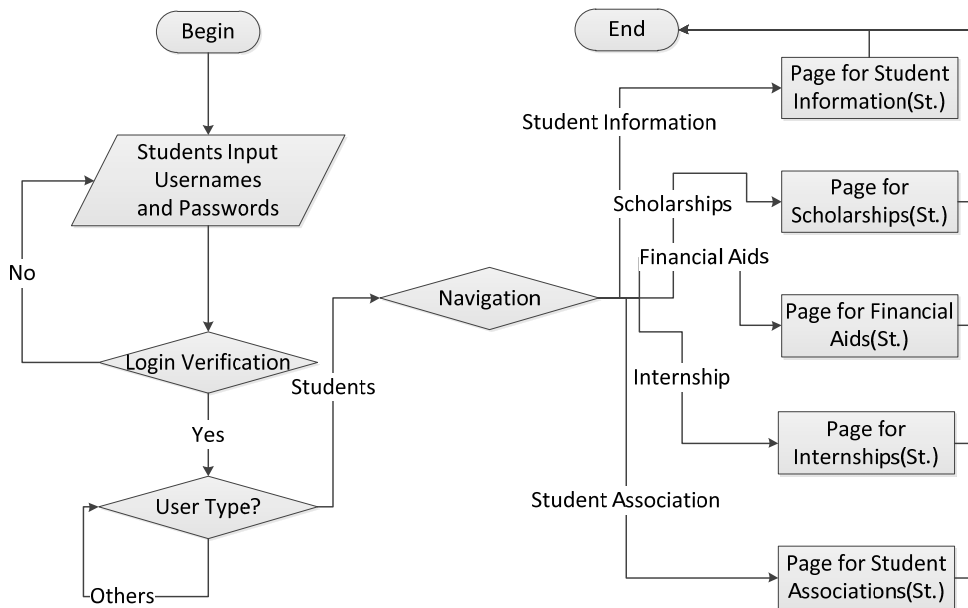
Fig. 5 The Flowchart of the Login Module

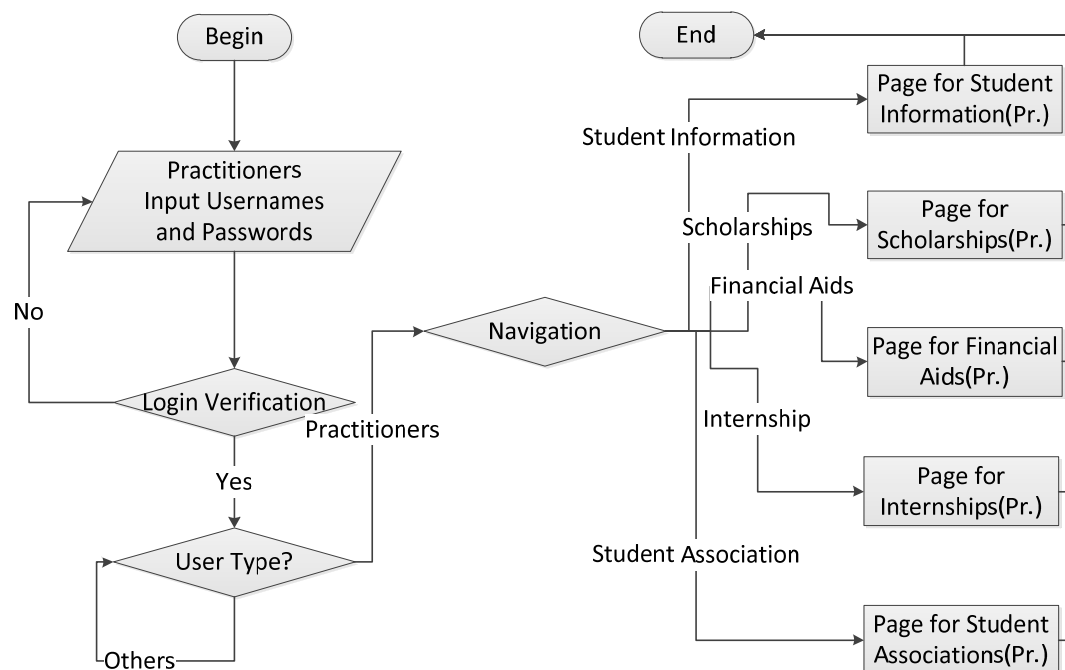Fig. 6 The Flowchart of the Student Module

Fig. 7 The Flowchart of the Practitioner Module

4.4. Implementation of Administrator module

System administrators are responsible for the users and group administration (based on login information table), server configuration, and server maintenance. They can manage the information of students, practitioners, scholarships, financial aids, internships, student associations (based on student information table, scholarship information table, financial aid information table, internship information table, and student association information table), etc. Fig.8 shows the flowchart of the administrator module.

5. Conclusions

Student affairs have been playing more and more important roles at institutions of higher education. We design an efficient and automated management system for student affairs, which can advance student learning and development; foster community engagement; promote diversity, inclusion and respect; and empower students to thrive.

We divide the management system into four functional modules, i.e. login module, student module, practitioner module and administrator module. Based on the well-known three-tier B/S architecture, we use Microsoft ASP.Net (Visual Studio 2010) and SQL Server to implement the system. (1) The Login module can identify roles, e.g. students, practitioners and system administrators, which is used as an access between the back-end system on the server and the system website on user's terminals. (2) The student module can be used to apply for scholarships, financial aids, internships, joining student associations, etc. (3) The practitioner module can be used to process student affairs, e.g. add, delete, update and query student information, review the applications for scholarships, financial aids, internships, joining student associations, etc. (4) The administrator module is responsible for the users and group administration, server configuration, and server maintenance. It can manage information of students, scholarships, financial aids, internships, student associations, etc.
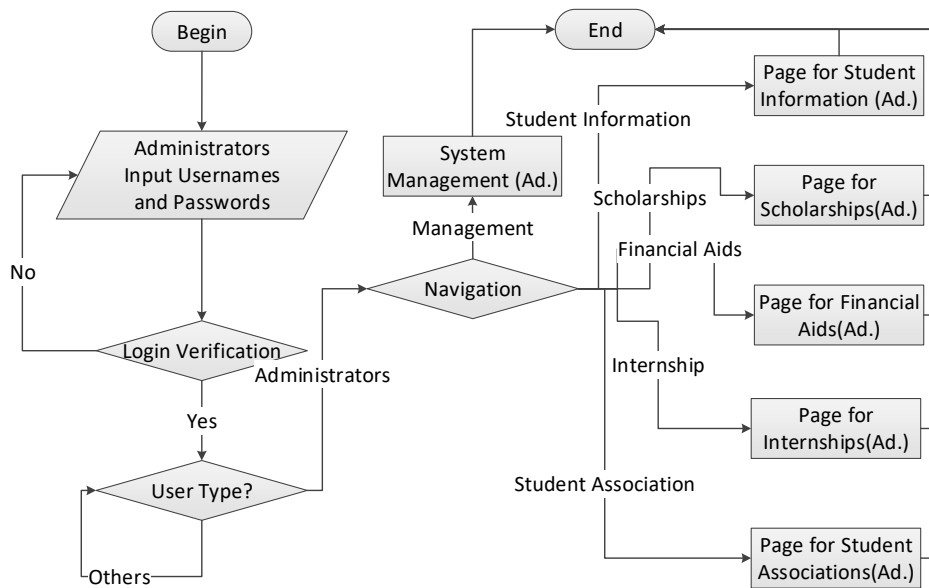
Fig. 8 The Flowchart of the Administrator Module

Acknowledgements

References

[1] Nancy J. Evans and Robert D. Reason. Guiding principles: A review and analysis of student affairs philosophical statements. Journal of College Student Development, 42(4):359, 2001

[2] Gregory S. Blimling. Uniting scholarship and communities of practice in student affairs. Journal of College Student Development, 42(4):81-96, 2001.

[3] Paul A. Bloland et al. Redirecting the role of student affairs to focus on student learning. Journal of College Student Development, 37(2):17-26, 1996.

[4] David C. Turner III. How black students are saving higher education. Journal of Critical Scholarship on Higher Education and Student Affairs, 2(3):2, 2016.

[5] Linda Clement. Making change happen in student affairs: Challenges and strategies for professionals. Journal of College Student Development, 56(4):413-414, 2015.