

# Student Information System

## Technical Design

## **1. Introduction:**

### **A. Purpose:**

This document describes the architecture and functionality of our project about student information management system. It includes some software or technology used to develop our system, these processes would be followed the knowledge got from web courses, trying to make the system more convenient and comfortable to be used.

### **B. Definitions:**

Here provides some basic information may covered by our project, including some explanations in our project and some software used to develop the system.

System	Student information management system- the system can store the information of students, easily query information for students and easily modify information for professors.
JSP	JavaServer Pages – A server-side scripting language used to produce dynamic, platform-independent web applications
XML	Extensible Markup Language – A powerful markup language used to transmit, serialize, and format data
Tomcat	Apache Tomcat - An open-source web server that implements JSP
MySQL	MySQL – Using this software to design our database to carry on the data storage and extraction.

### **C. Software**

MyEclipse 8.5 Version

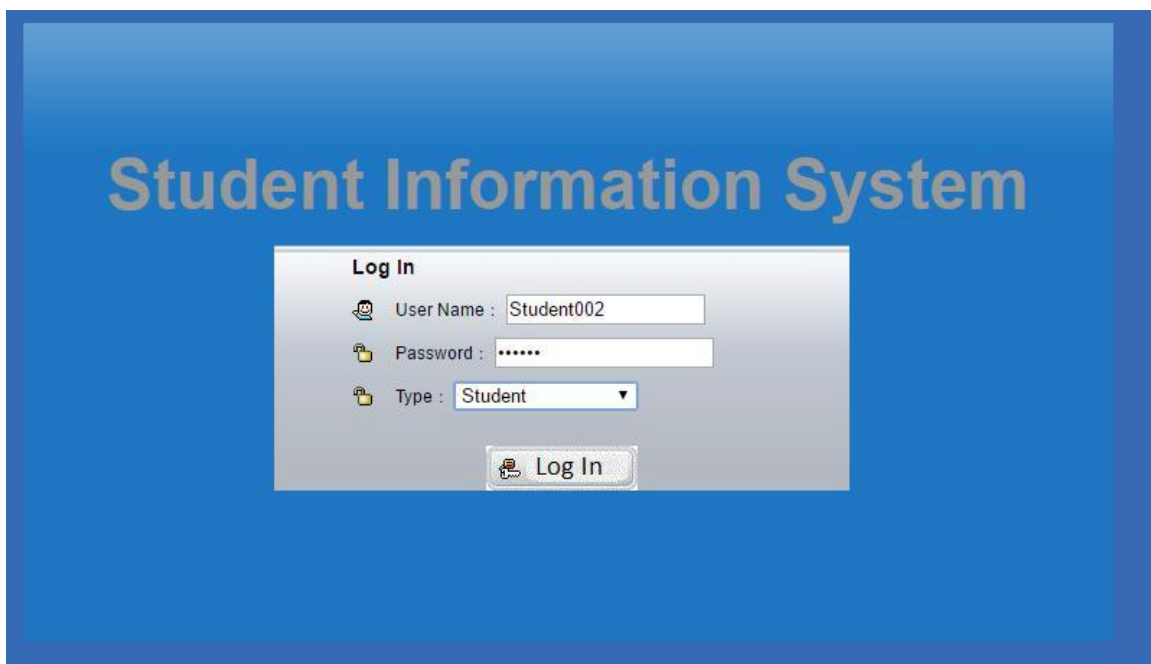
Mysql 5.0 Version

PhpMyAdmin Database Manager Version 2.10

#### **D. System Functions Overview**

The student information system is mainly used to manage the information of student and faculty. First thing, for both of them, they don't have the authority to add or change exist user. All of the user account are given by the administrator account. The most functions of these three types account are same.

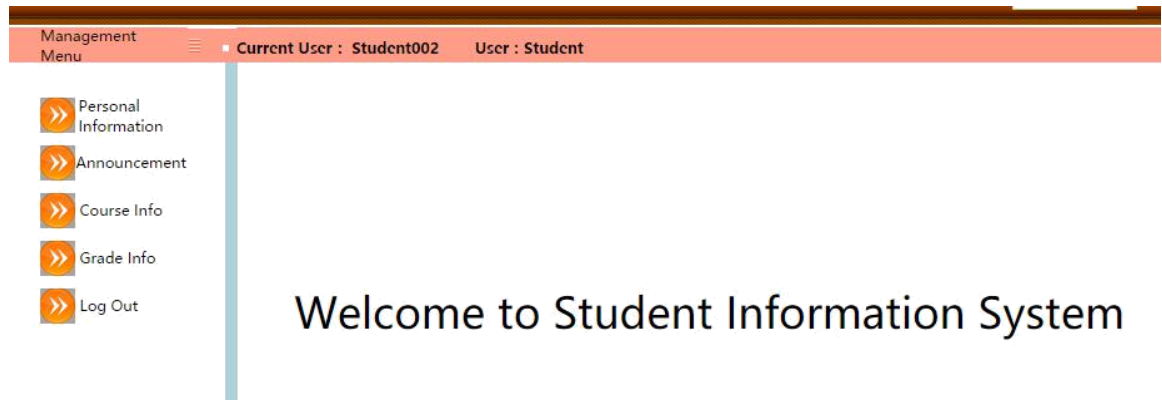
##### **i. Login Page**



There are three types of account: Student, Faculty and Administrator.

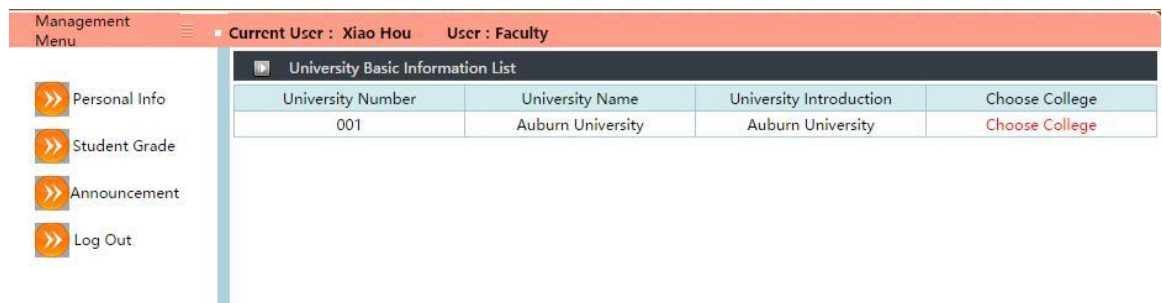
##### **ii. Student**

For student, it is the simplest version, they can view and change the personal information, view the public announcement and student grade. (They can also view the course information which faculty don't have.)



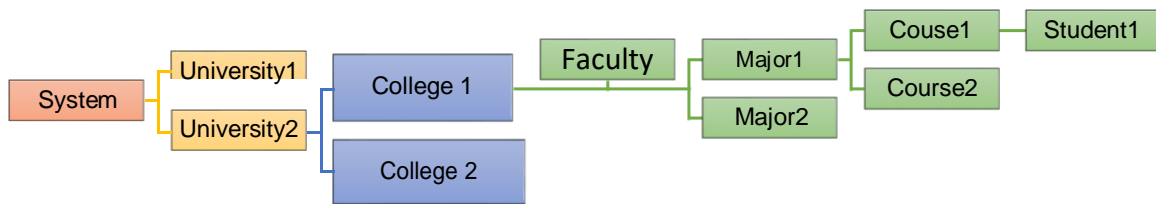
### iii. Faculty

For faculty, except the functions above, they can grade the assignment and test for students.



### iv. Administrator

Then, for administrator, the version is the complicated one. Except the functions above, it can manage the personal information for both student and faculty. It also can add and manage university, college, course and student to it. The relationship is like the picture below.



It can also add and manage courses in the system which is different from what our team wanted to do in the proposal. In our proposal, faculty can add and manage courses by themselves which may cause mess as they may misact online. So, we set the administer account which can manage almost everything.

Management Menu

Current User : System Administrator

» Change Password
» Basic Info Management
» Course Info Management
» Student Grades
» Announcement Management
» Log Out

Course Basic Information List

add

Course No	Course Name	No. of Credits	Type	Basic Operation
001	Basic Writing	3	1000	<a>Edit</a> <a>Delete</a>
002	Basic Reading	3	2000	<a>Edit</a> <a>Delete</a>

Back

And, it has the function to search for student grades, which is searching the data in the database by choosing keywords.

Choose Details :

Auburn University

Samuel Ginn College of Engineering

Mechanical Engineering

ME Class 001

Choose Course :

Basic Writing

Submit

Student	Course	Grade	Test Time	Test Type
Ben	Basic Writing	10	2017-05-01	Quiz
BigBen	Basic Writing	100	2017-05-04	Final Exam

Finally, it also has the function to publish announcement to other users.

Management Menu

Current User : System Administrator

» Change Password
» Basic Info Management
» Course Info Management
» Student Grades
» Announcement Management
» Log Out

Announcement List

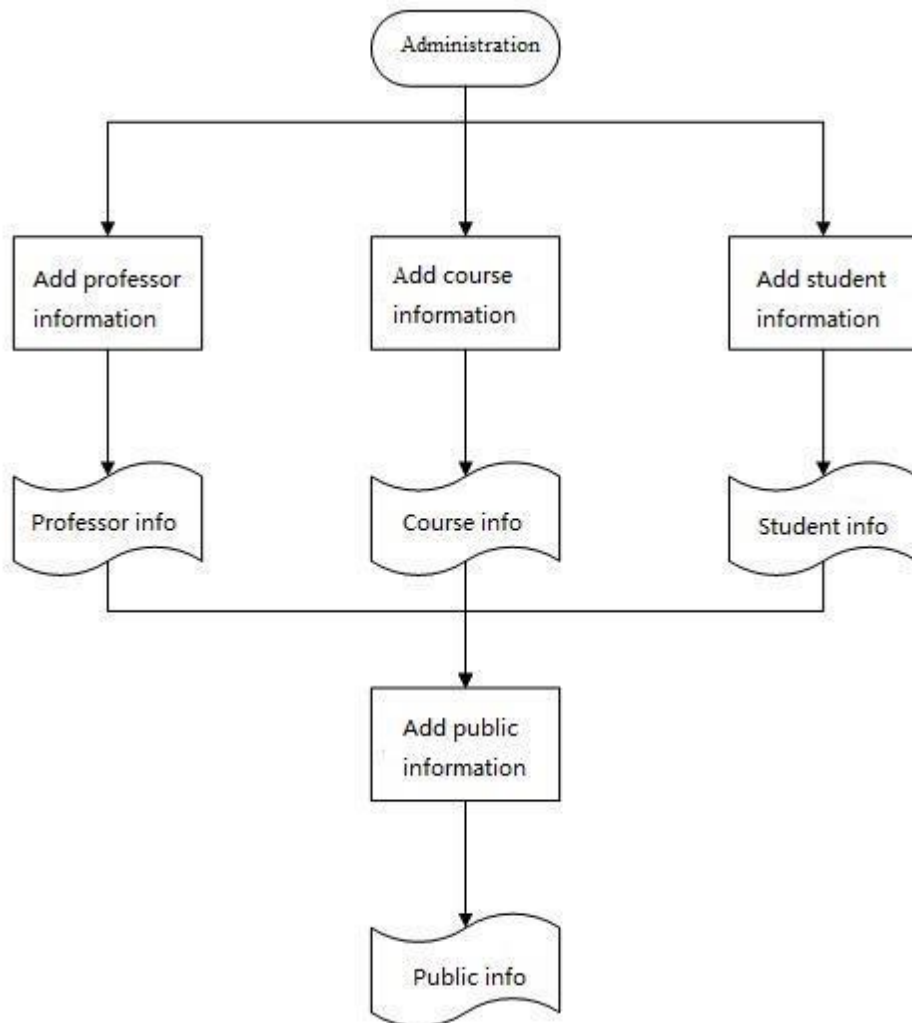
add

Title	Publish Time	Publisher	Content	Basic Operation
Test	2017-5-1 2:05:12	admin	Check Details	<a>Delete</a>
Spring Break	2017-3-1 1:09:50	admin	Check Details	<a>Delete</a>

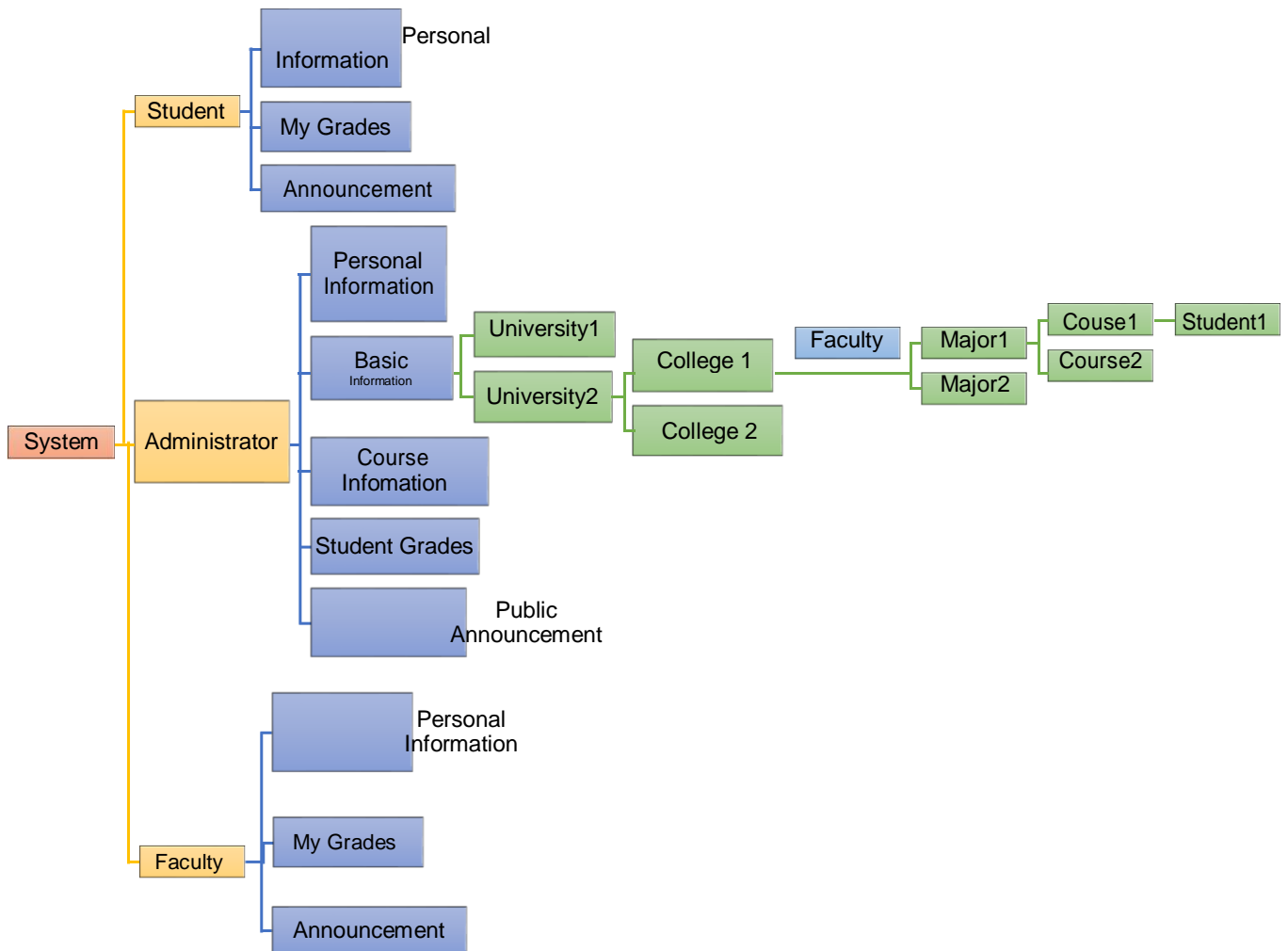
Above are functions of our project.

## 2. System analysis

### I. The process flow of the system



### II. System function modular structure chart



### III. The Database Structure

Table III.1 Administration information

Column name	Type	Description
id	Int(4)	ID, primary key



name	Varchar(30)	Account name
pwd	Varchar(20)	Users password

Used to store the information of account name and matched password.

Table III.2 Grades information

Column name	Type	Description
id	Int(4)	Grades ID, Primary ID
student_id	Int(4)	Student ID
course_id	Int(4)	Course ID
Grade	Int(4)	Value of grades
Time	Datetime(8)	Exam time
Comment	Varchar(50)	Additional information

Grades information can be stored by the database which can be easily read and modified by different users.

Table III.3 Public information

Column name	Type	Description
id	Int(4)	Information ID, Primary key

Title	Varchar(20)	Title of information
Content	Varchar(100)	Contents
Time	Datetime(8)	Published time

Used to store the public information.

Table III.4 Student information

Column name	Type	Description
id	Int(4)	No., Primary key
name	Varchar(20)	Name
Age	Int(4)	Age
truename	Varchar(20)	True name
Gender	Varchar(6)	Gender
Address	Varchar(100)	Address
College	Int(100)	College name
Student_id	Int(4)	Student_id
Tel	Varchar(20)	Telephone number
Major	Varchar (100)	Major information

Used to store the student information.

Table III.5 Professor information

Column name	Type	Description
id	Int(4)	No, Primary key
name	Varchar(20)	User Name
pwd	Varchar(20)	Password
truename	Varchar(20)	True name
Gender	Varchar(6)	Gender
Address	Varchar(100)	Address

Age	Int(4)	Age
Tel	Varchar (20)	Telephone number

Used to store the professor information.

#### IV. The database used

Server: localhost Database: db\_xxxx

Table	Action	Records	Type	Collation	Size	Overhead
t_admin		1	InnoDB	utf8_general_ci	16.0 KiB	-
t_banji		1	InnoDB	utf8_general_ci	16.0 KiB	-
t_chengji		4	InnoDB	utf8_general_ci	16.0 KiB	-
t_gonggao		2	InnoDB	utf8_general_ci	16.0 KiB	-
t_kecheng		2	InnoDB	utf8_general_ci	16.0 KiB	-
t_stu		2	InnoDB	utf8_general_ci	16.0 KiB	-
t_tea		2	InnoDB	utf8_general_ci	16.0 KiB	-
t_xibie		2	InnoDB	utf8_general_ci	16.0 KiB	-
t_xuyuan		1	InnoDB	utf8_general_ci	16.0 KiB	-
t_zhuanye		3	InnoDB	utf8_general_ci	16.0 KiB	-
10 table(s)	Sum	20	MyISAM	utf8_general_ci	160.0 KiB	0 B

Check All / Uncheck All With selected: ▾

Print view Data Dictionary

Create new table on database db\_xxxx

Name:  Number of fields:

Go

Open new phpMyAdmin window

Every data has been stored in separate database. The relationship is as below.

Table	Usage
t_admin	admin
t_banji	class
t_chengji	grade
t_gonggao	announcement
t_kecheng	course
t_stu	student
t_tea	teacher
t_xibie	college
t_xuyuan	university
t_zhuanye	major

We can easily change value by using these table. For example.

phpMyAdmin

Database

db\_xxxx (10)

db\_xxxx (10)

- t\_admin
- t\_banji
- t\_chengji
- t\_gonggao
- t\_kecheng
- t\_stu**
- t\_tea
- t\_xibie
- t\_xuyuan
- t\_zhuanye

Server: localhost Database: db\_xxxx Table: t\_stu "InnoDB free: 4096 kB"

Browse Structure SQL Search Insert Export Import Operations Empty Drop

Showing rows 0 - 1 (2 total, Query took 0.0004 sec)

SQL query:

```
SELECT *
FROM 't_stu'
LIMIT 0,30
```

[ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

Query results operations

Print view Print view (with full texts) Export

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None Go

	stu_id	stu_xuehao	stu_realname	stu_sex	stu_age	stu_card	banji_id	zhuanye_id	xibie_id	xuyuan
<input type="checkbox"/>	3	Stu001	Ben	Male	23	123456789	1	1	1	
<input type="checkbox"/>	4	Stu002	BigBen	Male	23	123456798	1	1	1	

Check All / Uncheck All With selected: ☐ ☒

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Open new phpMyAdmin window

phpMyAdmin

Database

db\_xxxx (10)

db\_xxxx (10)

- t\_admin
- t\_banji
- t\_chengji
- t\_gonggao
- t\_kecheng
- t\_stu**
- t\_tea
- t\_xibie
- t\_xuyuan
- t\_zhuanye

Server: localhost Database: db\_xxxx Table: t\_stu "InnoDB free: 4096 kB"

Browse Structure SQL Search Insert Export Import Operations Empty Drop

Field	Type	Function	Null	Value
stu_id	int(11)			3
stu_xuehao	varchar(50)			Stu001
stu_realname	varchar(50)			Ben
stu_sex	varchar(50)			Male
stu_age	varchar(50)			23
stu_card	varchar(50)			123456789
banji_id	int(11)			1
zhuanye_id	int(11)			1
xibie_id	int(11)			1
xuyuan_id	int(11)			1
login_name	varchar(50)			T001
login_pw	varchar(50)			123456
stu_del	varchar(50)			no
stu_one1	varchar(50)		<input checked="" type="checkbox"/>	
stu_one2	varchar(50)		<input checked="" type="checkbox"/>	
Go				
stu_one3	varchar(50)		<input checked="" type="checkbox"/>	
stu_one4	varchar(50)		<input checked="" type="checkbox"/>	
stu_one5	datetime		<input checked="" type="checkbox"/>	
stu_one6	datetime		<input checked="" type="checkbox"/>	
stu_one7	int(11)		<input checked="" type="checkbox"/>	
stu_one8	int(11)		<input checked="" type="checkbox"/>	

Save and then Go back to previous page

Go Reset

### **3. Conclusion**

Our team have used JSP, xml and SQL database to design and finish the project: Student Information System. During the work time, we found that a good proposal is a good instruction for the whole work. However, during the work, we have to change and add some functions to make it better. To be honest, some function in proposal is above our team ability, we replaced it with another one, like set a admin account to manage most things.

All in all, this is different from the proposal in details but it can reach the requirements we set in proposal.