

摘要	I
Abstract	II
引言	III
第一章 HOHO`系统概述	1
第二章 HOHO`系统需求分析	2
第三章 数据库设计	3
3.1 学生专业 (Nov30th_ZhuanYe)	3
3.2 社团成员职务 (Nov30th_ZhiWu)	3
3.3 学生旷课 (Nov30th_Truant)	3
3.4 系统加载信息 (Nov30th_SysOpt)	4
3.5 学生科目 (Nov30th_Subject)	4
3.6 学生服务列表 (Nov30th_StudentServiceList)	5
3.7 学生服务 (Nov30th_Service)	5
3.8 学生奖惩 (Nov30th_Rewards)	5
3.9 学生关系 (家长) 详细 (Nov30th_Relation)	6
3.10 专业公告 (Nov30th_PMessage_ZhuanYe)	7
3.11 班级公告 (Nov30th_PMessage_Class)	7
3.12 社团私人信息 (Nov30th_PersonalMessages)	8
3.13 社团公告消息 (Nov30th_OrganMessages)	8
3.14 社团信息 (Nov30th_Organ)	9
3.15 教师与学生消息 (Nov30th_Messages)	9
3.16 社团成员 (Nov30th_Members)	10
3.17 学生成绩 (Nov30th_Mark)	10
3.18 学生登陆记录 (Nov30th_Login)	11
3.19 科目教师关系 (Nov30th_Lessions)	11
3.20 社团活动消息 (Nov30th_HuoDongMessages)	11
3.21 社团活动 (Nov30th_HuoDong)	12
3.22 综合评测数据 (Nov30th_Evaluating_JYXY)	12
3.23 学生详细信息 (Nov30th_Detail)	13
3.24 智能课程表 (Nov30th_Curriculum)	14
3.25 学生班级 (Nov30th_Class)	14
3.26 社团分类 (Nov30th_Category)	15

3.27 社团部门 (Nov30th_BuMen)	15
3.28 系统管理员 (教师) (Nov30th_Admin)	15
3.29 数据库视图	16
第四章 系统功能模块设计	22
4.1 主登陆模块	22
4.2 管理员模块	22
4.2.1 管理员设置	23
4.2.2 发送消息	23
4.2.3 专业管理	24
4.2.4 修改班级	25
4.2.5 课程管理	26
4.2.6 成绩管理	26
4.2.7 家长管理	27
4.2.8 学生资料	27
4.2.9 请假管理	28
4.2.10 奖惩管理	28
4.2.11 发送通知	29
4.2.12 综合评测	30
4.2.13 密码管理	30
4.2.14 为您服务	31
4.3 管理员插件部分	31
4.3.1 社团管理	31
4.3.2 社团消息	32
4.4 统计模块	32
4.5 学生模块	33
4.5.1 个人消息	33
4.5.2 综合评定	34
4.6 手机访问模块	35
4.7 社团模块	36
4.7.1 我的社团	36
4.7.2 我的消息	37
4.7.3 退出社团	37

4.7.4 加入社团	37
4.7.5 系统消息	38
4.7.6 社团管理	38
第五章 系统功能实现	41
5.1 检查是否学生在一专业中的类.....	41
5.2 数据库连接基类	42
5.3 手机 WAP 访问类	45
5.4 社团控制类	48
5.5 社团成员操作类	56
5.6 社团消息操作类	59
5.7 公共权限检查类	63
5.8 安全检查类	66
5.9 学生信息操作类	70
5.10 系统插件类.....	75
5.11 Asp.Net 全局操作文件	76
5.12 根目录 web.config 文件内容	78
第六章 系统测试	82
论文小结	83
致谢	84
文献综述	85

摘要

HOHO``系统采用 ASP.NET 与 SQLEXPRESS 数据库设计。工作在 .Net Framework 2.0 上的 WEB 应用程序。

系统作为一个校园 WEB 系统，向学生，家长，教师等提供方便快捷的互动与管理。该系统以安全最先，效率其次，可维护性，可扩展性的原则设计。采用双层验证。所有涉及数据的界面均采用 AJAX 技术。提供手机访问支持以及插件功能支持。基于 Middleware 基准测试规范系统中没有使用存储过程。

硬件需求：IIS 服务器支持 .Net Framework 2.0，SQL Express 数据库或者更高版本。

关键词：校园，HOHO``，互动，社团，管理，学生。

Abstract

The system adopts ASP.NET and SQLEXPRESS database design. The work is in the .Net Framework 2.0. The last WEB applies procedure.

The system is a campus WEB system, to the student, parent, teacher's etc. provides convenient fast interaction and the management. That system the safety is earliest, the efficiency can support sex secondly, can expand a sexual principle design. The adoption double layer verifies. All interfaces which involve a data all adopt an AJAX technique. The interview of providing the cellular phone supports and the plug-in function support. Has no usage in the norm system according to the Middleware basis test saving process.

Hardware need: The support .Net Framework of the IIS server 2.0, the database of SQL Express or higher edition

Keywords: Campus, HOHO``, Interaction, consortium, manage, student.

引言

随着互联网的发展和普及，信息时代已经完全降临。作为一个学校，必须有她自己独立的信息处理系统和互动系统。HOHO``系统是按照求是应用技术学院当前的管理体制和分配制度开发的。为了可以让更多没有独立信息系统的学校来使用该系统，预留了多种扩展方式。以满足不同需求。

该系统工作于.Net 平台上。所有 HTML 代码完全符合 W3C XHTML 1.1 过渡规范。支持 FF, IE7 等符合 W3C 所有浏览器。安全性是当前互联网中系统最重要的考虑因素，HOHO``系统使用 Form 和 Session 双重验证。最大限度保证用户的用户登录的合法性及系统的安全性。接着往下看，你将会了解该系统如何工作的。

第一章 HOHO``系统概述

HOHO``系统能实现学校，家长，学生的互动。系统中有学生成绩管理，社团成员管理，综合评测统计，通知发布，学生互动交流等许多实用的功能。

HOHO``系统能节省大量，复杂的人力物力资源。一切成绩统计只需在网络上进行。该系统支持内网与外网安全设置，可以让学校内网和外网同时存在的情况下，控制外网访问的权限，保证系统的安全。

HOHO``系统采用 ASP.NET，具有很好的扩展性，一个学校使用网络进行所有的管理，即可标志该学校的整个体制的改变。今后的几年，预计所有的学校都会使用信息化的管理。通过互联网进行管理和操作是大势所趋。

系统运行需要支持 .net Framework 2.0 的 IIS 以及 SQLEXPRESS 数据库。

第二章 HOHO``系统需求分析

考虑到现在学生群体庞大，学校以学生为中心的各项管理增多，管理难度加大，HOHO``系统以学生管理系统为原型，针对现有的管理问题，开辟学生平台，教师平台，社团平台和学生会平台。主要满足学生能自主查询各项成绩等相关信息的功能，本系统考虑到查询信息的灵活性，还加入了手机查询模块，方便了学生及时快速的了解到自己的成绩。系统将动态的显示学生的各项评比情况，例如每学年一次的综合评定。在社团平台由学生直接参与讨论及管理。教师方面将赋予老师一定的权限，进行成绩管理和信息发布。教师和学生间平时缺少沟通也是普遍存在的问题，本系统中，教师和学生之间可以通过 message 形式进行沟通，促进了师生间的互动。

第三章 数据库设计

数据库名：C:\INETPUB\WWWROOT\APP_DATA\STUDENT.MDF

3.1 学生专业 (Nov30th_ZhuanYe)

表 3.1 学生专业表 (Nov30th_ZhuanYe)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	ID	int	4	0	√	√			
2	Name	varchar	50	0					
3	Memo	varchar	255	0					

3.2 社团成员职务 (Nov30th_ZhiWu)

表 3.2 社团成员职务表 (Nov30th_ZhiWu)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	ZhiWuID	int	4	0	√	√			
2	ZhiWuName	varchar	20	0					
3	ZhiWuMemo	varchar	50	0					
4	OrganID	int	4	0					

3.3 学生旷课 (Nov30th_Truant)

表 3.3 学生旷课表 (Nov30th_Truant)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	ID	int	4	0	√	√			
2	StudentID	int	4	0					
3	Date	datetime	8	3					
4	SubjectName	nvarchar	40	0			√		

5	Reason	varchar	50	0		√	
6	Times	smallint	2	0			((0))
7	Memo	varchar	50	0		√	
8	IsAFK	bit	1	0			((0))

3.4 系统加载信息 (Nov30th_SysOpt)

表 3.4 系统加载信息 (Nov30th_SysOpt)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	ID	int	4	0	√	√			
2	OptName	nvarchar	20	0					
3	OptValue	int	4	0			√		
4	OptChar	nvarchar	100	0			√		
5	CorV	bit	1	0					
6	Memo	nvarchar	100	0					

3.5 学生科目 (Nov30th_Subject)

表 3.5 学生科目 (Nov30th_Subject)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	SubjectID	int	4	0	√	√			
2	Name	varchar	50	0					
3	ZhuanYe	int	4	0					
4	Credit	float	8	0					
5	RuXue	smallint	2	0				((2000))	
6	Term	tinyint	1	0				((1))	
7	Memo	varchar	255	0			√		
8	Category	smallint	2	0				((0))	

3.6 学生服务列表 (Nov30th_StudentServiceList)

表 3.6 学生服务列表 (Nov30th_StudentServiceList)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	ServiceID	int	4	0	√	√			
2	ServiceName	varchar	20	0					
3	ServiceCat	int	4	0					
4	Context	varchar	6000	0					
5	ZhuanYeID	int	4	0					
6	Memo	varchar	6000	0					
7	Year	smallint	2	0					

3.7 学生服务 (Nov30th_Service)

表 3.7 学生服务 (Nov30th_Service)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	StartTime	datetime	8	3		√			
2	SourceAdd	varchar	50	0					
3	Status	bit	1	0					
4	FromIP	char	15	0					

3.8 学生奖惩 (Nov30th_Rewards)

表 3.8 学生奖惩 (Nov30th_Rewards)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	ID	int	4	0	√	√			
2	Date	datetime	8	3					
3	StudentID	int	4	0					

4	Subject	varchar	100	0	
5	Rank	varchar	50	0	
6	Degree	varchar	50	0	(' ?')
7	Memo	char	1000	0	
8	Disposal	bit	1	0	((0))
9	Term	tinyint	1	0	((1))
10	Score	float	8	0	((0))
11	Category	smallint	2	0	((0))

3.9 学生关系（家长）详细 (Nov30th_Relation)

表 3.9 学生关系（家长）详细 (Nov30th_Relation)

序号	列名	数据类型	长度	小数位数	主键	允许空	默认值	字段说明
1	ID	int	4	0	√	√		
2	Student ID	int	4	0				
3	Name	varchar	50	0				
4	Sex	bit	1	0				
5	Birthdate	datetime	8	3			(((2000) - (1)) - (1))	
6	Relation	varchar	50	0				
7	Address	varchar	50	0		√		
8	Cellphone	varchar	50	0		√		
9	Telephone	varchar	50	0		√		

1	Fax	varcha	50	0	√	
0		r				
1	Zipcode	varcha	50	0	√	
1		r				
1	Company	varcha	50	0	√	
2		r				
1	Memo	varcha	25	0	√	
3		r	5			
1	Passwor	char	32	0		('81dc9bdb52d04dc20036dbd8313
4	d					ed055')

3.10 专业公告 (Nov30th_PMessage_ZhuanYe)

表 3.10 专业公告 (Novth30th_Pmessage_ZhuanYe)

序 号	列名	数据类型	长 度	小 数 位	标 识	主 键	允 许 空	默 认 值	字 段 说 明
1	ID	int	4	0	√	√			
2	AdminName	varchar	20	0					
3	ZhuanYeID	int	4	0					
4	ExpDate	datetime	8	3					
5	PubDate	datetime	8	3					
6	Context	text	16	0					
7	Tip	varchar	50	0					

3.11 班级公告 (Nov30th_PMessage_Class)

表 3.11 班级公告 (Nov30th_Pmessage_Class)

序 号	列名	数据类型	长 度	小 数 位	标 识	主 键	允 许 空	默 认 值	字 段 说 明
1	ID	int	4	0	√	√			
2	AdminName	varchar	20	0					
3	ClassID	int	4	0					

4	ExpDate	datetime	8	3
5	PubDate	datetime	8	3
6	Context	text	16	0
7	Tip	varchar	50	0

3.12 社团私人信息 (Nov30th_PersonalMessages)

表 3.12 社团私人信息 (Nov30th_PersonalMessages)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	MessageID	int	4	0	√	√			
2	SendStudentID	int	4	0					
3	ReceiveStudentID	int	4	0					
4	IsRead	bit	1	0					
5	MessageContext	varchar	255	0					
6	SendTime	smalldatetime	4	0					
7	MessageSign	tinyint	1	0					

3.13 社团公告消息 (Nov30th_OrganMessages)

表 3.13 社团公告信息 (Nov30th_OrganMessages)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	MessageID	int	4	0	√	√			
2	SendStudentID	int	4	0			√		
3	OrganID	int	4	0					
4	AIMTpye	bit	1	0					
5	BuMenID	int	4	0					
6	ZhiWuID	int	4	0					
7	MessageContext	varchar	255	0					

8 SendTime smalldatetime 4 0

3.14 社团信息 (Nov30th_Organ)

表 3.14 社团信息 (Nov30th_Organ)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	OrganID	int	4	0	√	√			
2	OrganName	varchar	50	0					
3	OrganIntro	varchar	255	0					
4	People	int	4	0					
5	OrganCharge	int	4	0					
6	OrganCreateTime	smalldatetime	4	0					
7	OrganTeacher	varchar	10	0					
8	CatID	int	4	0					
9	IsActive	bit	1	0					
10	OrganMemo	varchar	7000	0					

3.15 教师与学生消息 (Nov30th_Messages)

表 3.15 教师与学生消息 (Nov30th_Messages)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	ID	int	4	0	√	√			
2	AdminName	varchar	20	0					
3	StudentID	int	4	0					
4	SendDate	datetime	8	3					
5	RecvDate	datetime	8	3			√		
6	StuRead	bit	1	0			√	((0))	
7	TeaRead	bit	1	0			√	((0))	
8	ExpDate	datetime	8	3					

9	TeaContext	varchar	8000	0			√
10	StuContext	varchar	8000	0			√

3.16 社团成员 (Nov30th_Members)

表 3.16 社团成员表 (Nov30th_Members)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	MemberID	int	4	0	√	√			
2	MemberStudentID	int	4	0					
3	NickName	varchar	12	0					
4	Contact	varchar	50	0					
5	OrganID	int	4	0					
6	ZhiWuID	int	4	0					
7	BuMenID	int	4	0					
8	JoinTime	smalldatetime	4	0					
9	IsReg	bit	1	0					
10	RegMemo	varchar	50	0					

3.17 学生成绩 (Nov30th_Mark)

表 3.17 学生成绩表 (Nov30th_Mark)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	ID	int	4	0	√	√			
2	Date	datetime	8	3					
3	StudentID	int	4	0					
4	SubjectID	int	4	0					
5	Normal	float	8	0			√	((0))	
6	Repair	float	8	0			√	((0))	
7	MakeUp	float	8	0			√	((0))	

8	Term	smallint	2	0		√	((1))
9	Overall	float	8	0		√	((0))
10	GetCredit	float	8	0		√	((0))

3.18 学生登陆记录 (Nov30th_Login)

表 3.18 学生登陆记录 (Nov30th_Login)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	StudentID	int	4	0		√			
2	Password	nvarchar	64	0					
3	NickName	nvarchar	40	0					
4	Count	int	4	0					
5	LastLogin	datetime	8	3					
6	LastIP	nvarchar	30	0					

3.19 科目教师关系 (Nov30th_Lessions)

表 3.19 科目教师关系表 (Nov30th_lessons)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	ID	int	4	0	√	√			
2	Name	varchar	50	0					
3	Teacher	varchar	10	0					
4	SubjectID	int	4	0					
5	ClassID	int	4	0					

3.20 社团活动消息 (Nov30th_HuoDongMessages)

表 3.20 社团活动消息 (Nov30th_HuoDongMessages)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
----	----	------	----	-----	----	----	-----	-----	------

				位		空	值
1	MessageID	int	4	0	√	√	
2	HuoDongID	int	4	0			
3	SendStudentID	int	4	0			
4	SendTime	smalldatetime	4	0			
5	MessageContext	varchar	255	0			
6	MessageSign	tinyint	1	0			

3.21 社团活动 (Nov30th_HuoDong)

表 3.21 社团活动表 (Nov30th_HuoDong)

序 号	列名	数据类型	长 度	小 数 位	标 识	主 键	允 许 空	默 认 值	字 段 说 明
1	HuoDongID	int	4	0	√	√			
2	SignUpTime	smalldatetime	4	0					
3	BeginTime	smalldatetime	4	0					
4	EndTime	smalldatetime	4	0					
5	Place	varchar	50	0					
6	People	int	4	0					
7	OrganID	int	4	0					
8	Context	varchar	255	0					
9	Title	varchar	20	0					
10	Intro	varchar	50	0					
11	Money	int	4	0					
12	Category	int	4	0					
13	InReg	bit	1	0					
14	RegMemo	varchar	50	0					

3.22 综合评测数据 (Nov30th_Evaluating_JYXY)

表 3.22 综合评测数据 (Nov30th_Evaluating_JYXY)

序	列名	数据类	长	小	数	标	主	允	许	默	认	字	段	说
---	----	-----	---	---	---	---	---	---	---	---	---	---	---	---

号		型	度	位	识	键	空	值	明
1	ID	int	4	0	√	√			
2	StudentID	int	4	0					
3	D1	float	8	0					
4	D2	float	8	0					
5	D3	float	8	0					
6	D4	float	8	0					
7	Z1	float	8	0					
8	Z2	float	8	0					
9	T1	float	8	0					
10	T2	float	8	0					
11	T3	float	8	0					
12	A	float	8	0					
13	F	float	8	0					
14	Term	tinyint	1	0					
15	Publish	bit	1	0					

3.23 学生详细信息 (Nov30th_Detail)

表 3.23 学生详细信息 (Nov30th_Detail)

序 号	列名	数据类型	长 度	小 数 位	标 识	主 键	允 许 空	默认值	字段 说明
1	StudentID	int	4	0		√			
2	Name	varchar	50	0					
3	FamilyAdd	varchar	80	0					
4	Contact	varchar	100	0					
5	Dormitory	varchar	50	0					
6	IsPM	tinyint	1	0				((0))	
7	Birthday	smalldatetime	4	0					
8	Address	varchar	50	0					
9	CellPhone	varchar	50	0					

10	Password	char	32	0				
11	ZhuanYe	int	4	0			((0))	
12	RuXue	smallint	2	0				
13	Sex	bit	1	0			((0))	
14	LoginLevel	tinyint	1	0			((100))	
15	ClassID	int	4	0			((0))	

3.24 智能课程表 (Nov30th_Curriculum)

表 3.24 智能课程表 (Nov30th_Curriculum)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	ID	int	4	0	√	√			
2	Week	tinyint	1	0				((0))	
3	Lweek	bit	1	0				((0))	
4	ClassID	int	4	0				((0))	
5	Name	varchar	20	0					
6	Address	varchar	50	0					
7	BeginTime	datetime	8	3					
8	EndTime	datetime	8	3					
9	Memo	varchar	255	0				(' ? 6 ?')	
10	LessonID	int	4	0			√	((0))	

3.25 学生班级 (Nov30th_Class)

表 3.25 学生班级表 (Nov30th_Class)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	ID	int	4	0	√	√			
2	ClassName	varchar	50	0					
3	ZhuanYe	int	4	0					
4	Master	varchar	10	0			√		

5	Contact	varchar	20	0					
6	Term	tinyint	1	0					((0))
7	RuXue	smallint	2	0					((0))

3.26 社团分类 (Nov30th_Category)

表 3.26 社团分类 (Nov30th_Category)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	CatID	int	4	0	√	√			
2	CatName	varchar	20	0					
3	CatMemo	varchar	50	0					

3.27 社团部门 (Nov30th_BuMen)

表 3.27 社团部门 (Nov30th_BuMen)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	BuMenID	int	4	0	√	√			
2	BuMenName	varchar	20	0					
3	BuMenMemo	varchar	50	0					
4	OrganID	int	4	0					

3.28 系统管理员 (教师) (Nov30th_Admin)

表 3.28 系统管理员 (教师) (Nov30th_Admin)

序号	列名	数据类型	长度	小数位	标识	主键	允许空	默认值	字段说明
1	ID	int	4	0	√	√			
2	Name	varchar	50	0					
3	Password	char	32	0					
4	level	tinyint	1	0					

5	Memo	varchar	255	0	
6	Question	varchar	50	0	✓
7	Answer	varchar	50	0	✓
8	Email	varchar	100	0	✓
9	CellPhone	varchar	50	0	
10	LastLogin	smalldatetime	4	0	
11	LoginTimes	int	4	0	((0))
12	Activated	bit	1	0	
13	ZhuanYeID	int	4	0	((0))

3.29 数据库视图

各功能视图如下

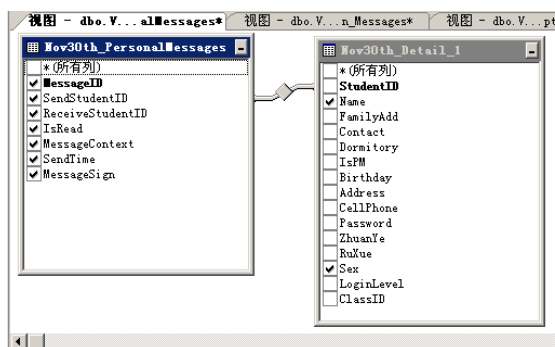


图 3.1 私人消息视图

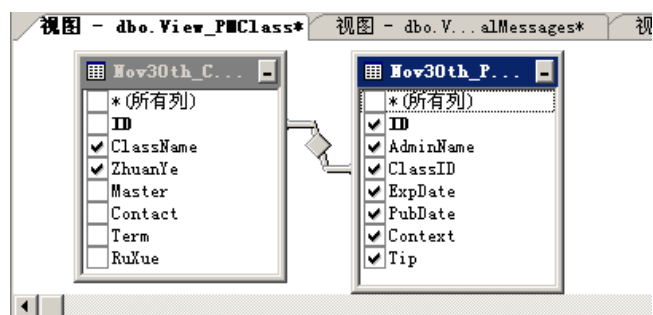


图 3.2 班级公告视图

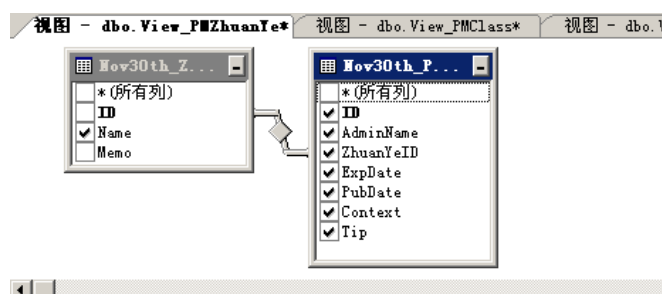


图 3.3 专业公告视图

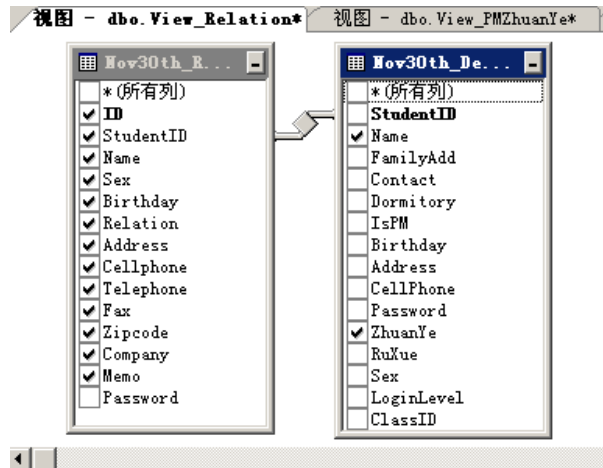


图 3.4 学生家长信息视图

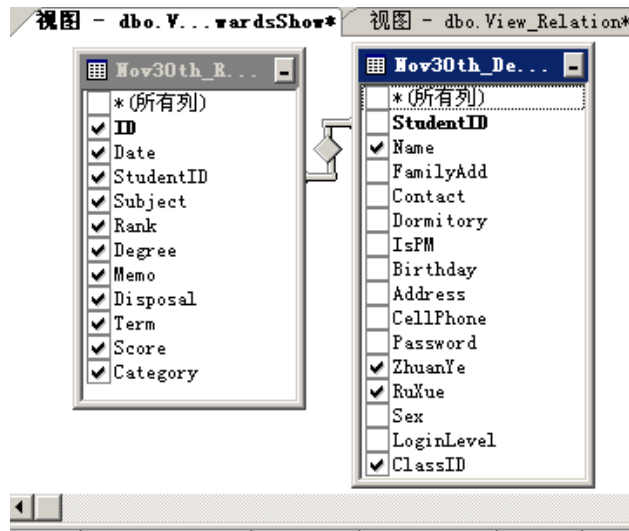


图 3.5 奖惩视图

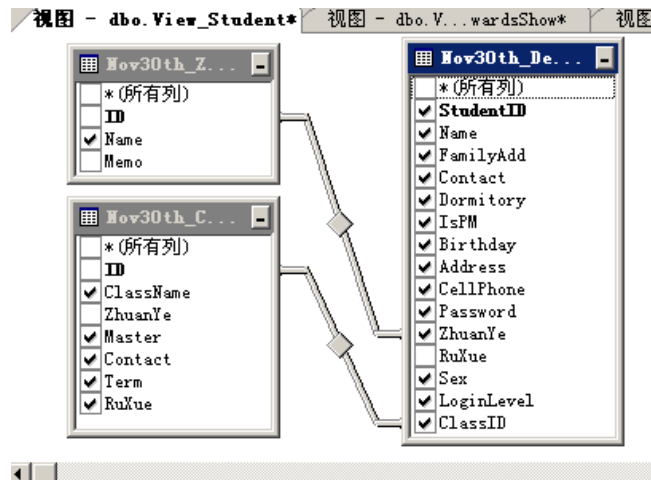


图 3.6 学生详细信息视图



图 3.7 科目视图

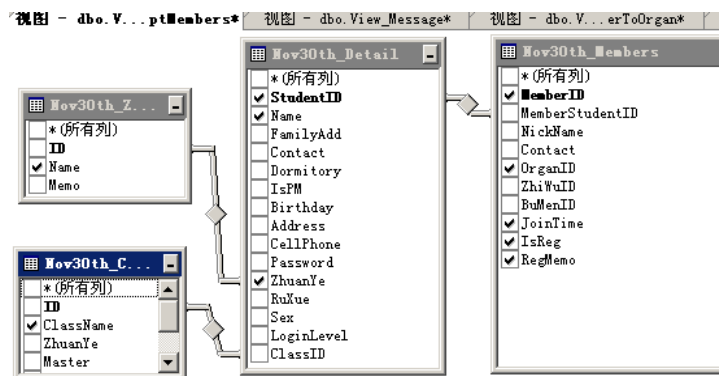


图 3.8 学生基本信息视图

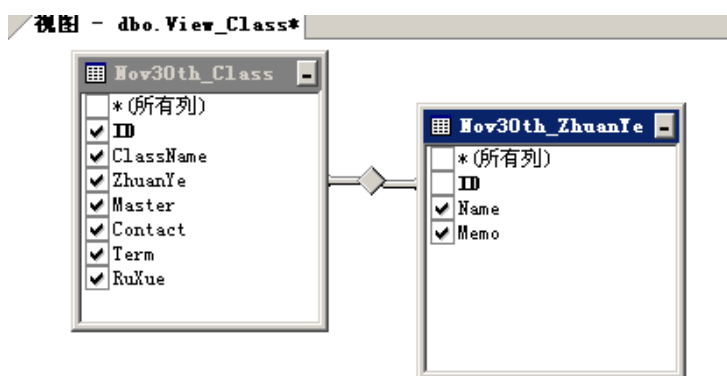


图 3.9 班级信息视图

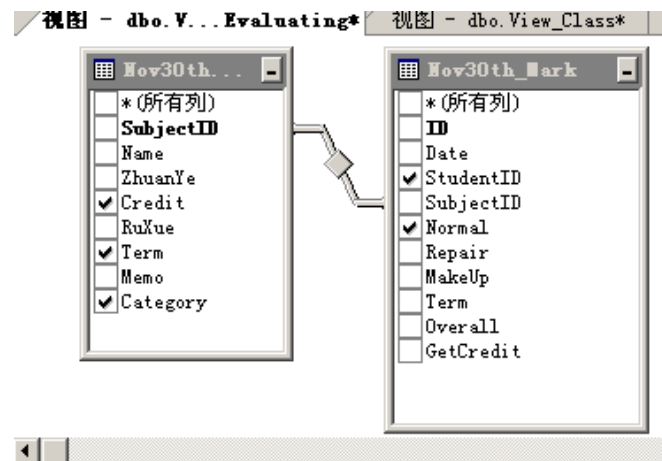


图 3.10 成绩视图

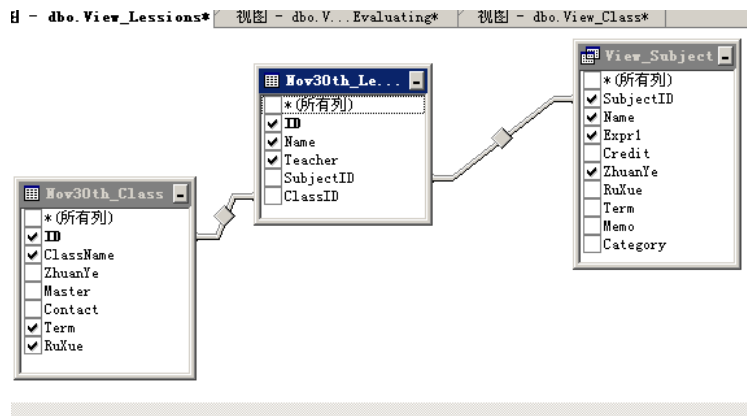


图 3.11 课程视图

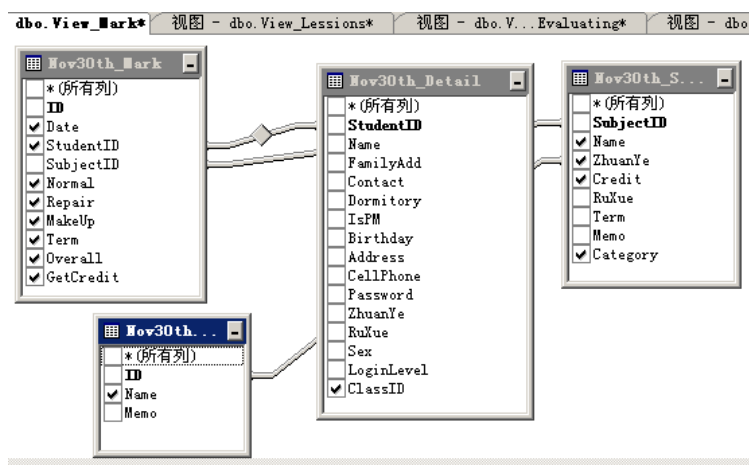


图 3.12 成绩视图

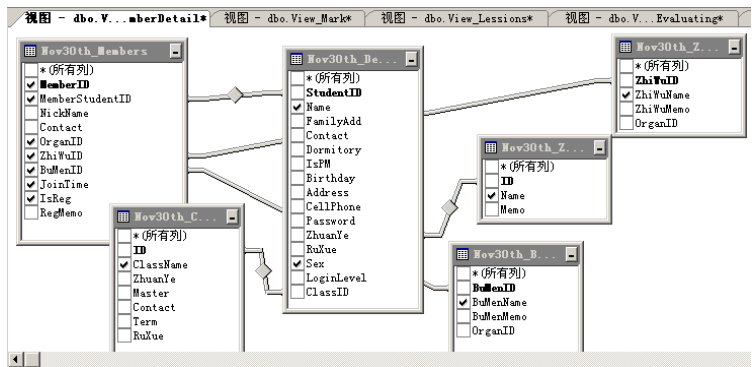


图 3.13 社团成员信息视图

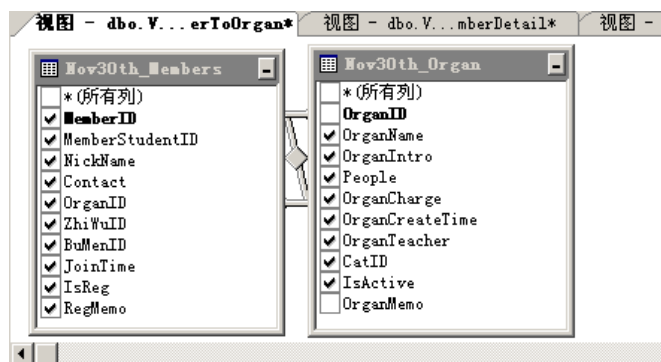


图 3.14 学生所在社团视图

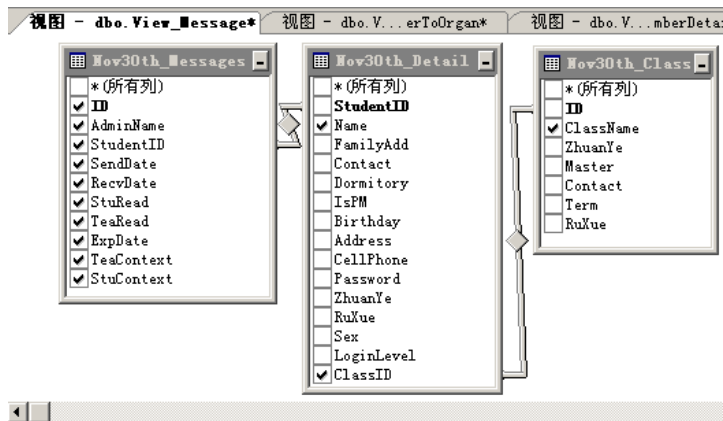


图 3.15 学生收发消息视图

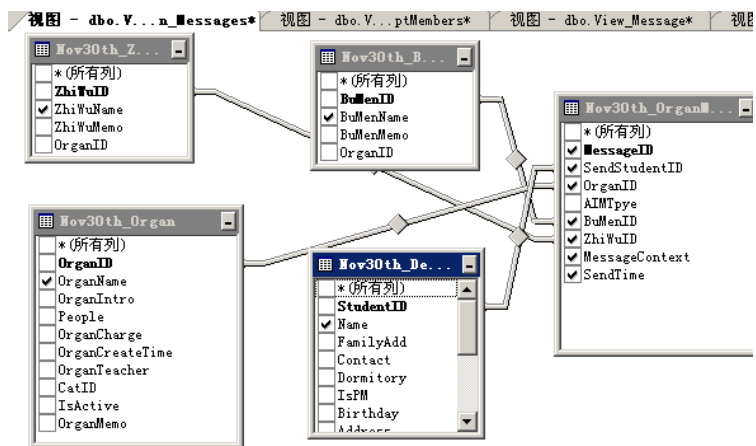


图 3.16 社团消息视图

第四章 系统功能模块设计

4.1 主登陆模块

登陆界面，在这里可以输入管理员用户名或者学号进行登陆，由系统自动判断用户类型。输入错误，均会出现**密码错误!**或者**没有这个用户名**的提示。



图 4.1 系统登陆界面

使用超级管理员的身份登陆之后，出现后台操作界面。其中系统设置是针对整个网站的安全性进行配置。可以选择是否允许学校或者家长登陆，是否只在学校内网使用等。安全模式用于系统处理批量数据时，防止学生或者教师操作冲突，可以暂停数据服务。停止服务则是关闭整个网站服务系统。

4.2 管理员模块



图 4.2 管理员模块

4.2.1 管理员设置

管理员设置中，超级管理员（等级 255）可以添加或者删除所有管理员（除本身）。该页面只有超级管理员能够访问。管理员必须对应一个学校的专业。有且只有一个。

现有管理员

	ID	用户名	专业	等级	备注	最后登录	登录次数	帐户状态	联系电话
编辑 删除	2	阿奔	[计算机信息技术部门]	254	123456	2007-2-7 8:45:00	2	<input type="radio"/> 激活 <input type="radio"/> 停用	123456
编辑 删除	1	superadmin	[计算机信息技术部门]	255	?	2007-2-12 18:07:00	94	<input type="radio"/> 激活 <input type="radio"/> 停用	13588864866

如果您更改了某个管理员的信息，也必须重新登录才能生效。

管理员添加修改

用户名(20字符以内)	<input type="text"/>
专业	[计算机信息技术部门]
密码	<input type="text"/>
级别(1-255的数字)	<input type="text"/>
备注	<input type="text"/>
手机号码	<input type="text"/>
<input type="button" value="新增管理员"/>	

新建管理员时，所有内容都必须填写。
另外，SuperAdmin的资料是无法修改的。

图 4.3 管理员设置

4.2.2 发送消息

教师可以与学生通过类似留言的方式交流。主要面向与发送个人通知，事物之类的消息。学生可以回复教师的消息，但只能回复一次。该功能只有超级管理员能够使用。

与学生之间的消息

姓名	班级	学号	回复已读	发送日期	回复日期	过期日期	编号
选择	阿邦	软件042	504011151	False	2007-2-12 17:38:13		2006-1-1 0:00:00 1

发送人	<input type="text"/>
学号	<input type="text"/>
发送日期	<input type="text"/>
回复日期	<input type="text"/>
是否已回复	<input type="checkbox"/>
回复已读	<input type="checkbox"/>
过期日期	<input type="text"/>
发送内容	<input type="text"/>
回复内容	<input type="text"/>
<input type="button" value="插入"/> <input type="button" value="取消"/>	

图 4.4 发送消息

4.2.3 专业管理

该设置是使用这个系统的前提，所有数据分配都按照专业级别划分，其中的专业中应该包含入学时间，比如 04 届计算机专业。每个管理员都对应一个专业。在这里可以编辑，添加专业。

课程专业设置

现有专业

	专业编号	专业名称	专业简介
编辑	1	计算机信息技术部门	计算机

添加专业

专业名称	<input type="text"/>
备注 (最多255个字)	<input type="text"/>
<input type="button" value="新增专业"/>	

图 4.5 专业管理

如同其他页面一样，都采用了 AJAX 技术。页面无刷新编辑。

	专业编号	专业名称	专业简介
更新 取消	1	计算机信息技术部门	计算机

图 4.6 专业信息

(a) 科目管理。这里查看已经有的科目及对应的教师。也可以编辑科目信息。

科目管理

科目编号	科目名字	入学年份	学期	科目学分	所属专业	备注	成绩类别
选择 1	语文	2004	1	4	计算机信息技术部门	语文	0

现有科目的课程教师

当前没有该课程的教师或者您没有选择课程。
 请选择上面的课程，这里将会显示该课程对应的教师信息。

科目详细信息 (科目类别说明)

科目编号	1
科目名字	语文
专业编号	1
科目学分	4
入学年份	2004
学期	<input type="text" value="第一学期"/>
备注	语文
科目类别	<input type="text" value="普通课程"/>
编辑	

查询专业:

图 4.7 科目管理

点“编辑”按钮之后可以直接进行科目编辑。(AJAX)

科目详细信息 (科目类别说明)

科目编号	1
科目名字	语文
专业编号	1
科目学分	4
入学年份	2004
学期	第一学期
备注	语文
科目类别	普通课程
<input type="button" value="更新信息"/> <input type="button" value="取消操作"/>	

图 4.8 科目详细

(b) 添加科目。在这里页面可以用来添加学校的科目，课程。

添加新的科目**添加科目 (科目类别说明)**

科目名字	
所属专业	计算机信息技术部门
入学年份	
学期	第一学期
科目学分	
科目类别	普通课程
备注	

图 4.9 添加科目

点“科目类别说明”可以看到有关与综合评测的项目已经必须遵循的规定。

说明文档

科目类别说明(数字): 0. 普通课程 1. 体育类 2. 选修课程

奖励类别说明(数字): 0. 不计分 1. 国际等竞赛获奖及等级考试 2. 体育类获奖 3. 优秀班干部及校级处分

图 4.10 科目管理说明

4.2.4 修改班级

这里将显示该管理员相对应的专业中所有的班级，该页面提供编辑班级，添加班级信息的功能。

修改班级

现有班级信息

选择	编号	班级名称	专业	班主任	联系方式	学期	入学年份	备注
<input checked="" type="checkbox"/>	1	软件042	计算机信息技术部门	刘德华	香港	1	2004	计算机

班级详细信息

编号	1
班级名称	软件042
专业	计算机信息技术部门
班主任	刘德华
联系方式	香港
学期	1
入学年份	2004
编辑 新建 删除	

图 4.11 班级管理

4.2.5 课程管理

一个学校不可能一个教师上所有同样的课程，于是必须要多个教师对应一个课程。

课程管理

现有课程教师

选择	编号	课程名	教师	班级	科目名	专业	入学年份	学期
<input checked="" type="checkbox"/>	1	04班语文	杜德伟	软件042	语文	计算机信息技术部门	2004	1

课程教师编辑

编号	1
课程名	04班语文
教师	杜德伟
科目名称	语文
班级	软件042
编辑 新建 删除	

图 4.12 课程管理

4.2.6 成绩管理

这里添加学生的成绩。作为辅助。主要途径是用插件或者其他工具进行批量数据库录入。

成绩管理	
添加学生成绩	
学生姓名	<input type="text"/>
学生学号	<input type="text"/>
考试日期	<input type="text"/>
学期	第一学期
所属专业	计算机信息技术部门
所属科目	语文
考试成绩	<input type="text"/>
总评成绩	<input type="text"/>
重修成绩	<input type="text"/>
<input type="button" value="添加学生成绩"/>	

图 4.13 成绩管理

4.2.7 家长管理

查询，修改和录入学生家长的信息。方便教师和学生家长沟通。可以根据多个条件搜索学生的家长信息。

家长管理									
输入条件按回车查询									
学生学号:	<input type="text" value="504011151"/>								
学生姓名:	<input type="text"/>								
家长姓名:	<input type="text"/>								
<input type="button" value="插入新关系"/>									
查询结果									
选择	阿邦	1	504011151	测试	<input checked="" type="checkbox"/>	父亲	123456	13588864866	某某公司
家长详细信息									
序号	1								
学号	504011151								
关系姓名	测试								
女性?	<input checked="" type="checkbox"/>								
出生日期	1985-10-3 0:00:00								
关系	父亲								
联系地址	浙江杭州								
手机号码	13588864866								
联系电话	123456								
传真	123456								
邮政编码	310000								
公司名称	某某公司								
备注	备注								
编辑 新建 删除									

图 4.14 家长管理

4.2.8 学生资料

学生信息的加添和修改页面。作为辅助。主要用插件或者其他工具将学生信息批量录入数据库。

学生资料

学号	姓名	性别	寝室	面貌	专业	入学
504011151	阿邦	True	嘿嘿	2	1	2004 选择
504011152	阿曼	False	哈哈	1	1	2004 选择
504011153	哈哈	False	不知道	1	1	2004 选择

插入新学生

学生详细信息

编辑 新建

学号	504011153
姓名	哈哈
班级	软件042
性别	<input checked="" type="radio"/> 男 <input type="radio"/> 女
家庭地址	拉拉
联系方法	不知道
寝室地址	不知道
政治面貌	团员
生日	2000-1-1 0:00:00
地址	不知道
手机号码	123456789
登陆密码	
入学年份	2004

图 4.15 学生资料

4.2.9 请假管理

录入和查询学生的请假旷课信息。

请假管理

查询请假旷课信息

学号:

科目:

当前没有任何的旷课或者请假记录, 请单击下面的按钮新增记录。

新的请假旷课信息

选择上面的请假或者旷课记录, 将在这里显示更详细的信息。

图 4.16 请假管理

4.2.10 奖惩管理

添加修改删除奖惩信息。会作用到最后的综合评定。

奖惩管理

现有奖惩

姓名	学号	学期	奖励/处分项	日期	程度	荣誉	分数	分数类别	处分?
选择	阿邦	504011151	1	全国美男子	2007-1-1 0:00:00	一等奖	美男	100	1

添加奖惩信息

奖惩详细信息 (奖惩类别说明)

编号	1
日期	2007-1-1 0:00:00
学号	504011151
学期	1
获奖/处分项	全国美男子
荣誉/处分	美男
获奖/处分程度	一等奖
荣誉称号	美男
扣分/加分	100
分数类别	1
备注	美男加分
类别	<input checked="" type="radio"/> 处分 <input type="radio"/> 奖励
编辑 新建 删除	

图 4.17 奖惩管理

4.2.11 发送通知

该页面用于发送班级通知和专业通知，所属的学生登陆之后都能看到该通知的内容。

发送通知给学生

当前公告 (专业)

编号	发布人	专业名	过期日期	发布日期	标题	
选择	1	superadmin	计算机信息技术部门	2007-3-3 0:00:00	2007-2-12 17:46:32	测试, 专业

添加专业通知

选择上面的公告, 将在这里列出详细信息。

当前公告 (班级)

编号	发布人	班级	过期日期	发布日期	标题	
选择	2	superadmin	软件042	2007-4-4 0:00:00	2007-2-12 17:47:12	测试

添加班级通知

选择上面的公告, 将在这里列出详细信息。

图 4.18 发送通知界面

4.2.12 综合评测

综合评测的统计和查看，输入学生学号后，系统将根据学生的所有评测联系项目进行统计评分。页面中的评测统计按照 2002 年求是应用技术学院学生手册公式计算。

综合评测发布

综合评测学生学号: 504011151

请在下面添加或者更改评定分数

编号	学号	D1	D2	D3	D4	Z1	Z2	T1	T2	T3	附加分	综合评定	学期	已评定
1	504011151	23	23	43	0	50	0	0	2	0	0	51.28	2	<input checked="" type="checkbox"/>

学号: 504011151
 学生姓名: 阿邦
 当前操作: 修改评定

品德评分		
D1-班级评分数	<input type="text" value="23"/>	←在这里录入评定分数
D2-班主任评分数	<input type="text" value="23"/>	
D3-寝室卫生分数	<input type="text" value="43"/>	
D4-R3-班干部, 处分分数	<input type="text" value="0"/>	通过公式计算, 不可更改.
智力评分		
Z1-M0-所有成绩分数	<input type="text" value="50"/>	通过公式计算, 不可更改.
Z2-R1-证书等得奖加分	<input type="text" value="0"/>	通过公式计算, 不可更改.
体育评分		
T2-出勤分数	<input type="text" value="2"/>	←在这里录入评定分数
T1-M1-体育成绩分数	<input type="text" value="0"/>	通过公式计算, 不可更改.
T3-R2-体育类获奖加分	<input type="text" value="0"/>	通过公式计算, 不可更改.
操作		
综合评定附加分	<input type="text" value="0"/>	←根据需要扩展扣加分
在哪个学期	<input type="text" value="第二学期"/>	←选择该评定的学期
是否最后评定	<input type="checkbox"/> 最后评定	←选择后将发布该评定到学生。
<input type="button" value="修改评定"/> <input type="button" value="重新计算"/> <input type="button" value="读取评分"/>		

请注意! 改生评定已经发布! 请考虑后进行编辑!

图 4.19 综合评测发布

4.2.13 密码管理

管理员用于重新设置自己的账户密码。

密码管理

修改自己的密码

旧的密码:

新的密码:

确认新的密码:

图 4.20 管理密码

4.2.14 为您服务

学生界面的菜单修改。当学生点击该菜单中的一个项目时，将用整个网页显示该菜单所包含的内容。即通知。菜单可以为不同专业的学生显示不同的内容。

为您服务-菜单修改

为您服务

服务名	服务类别	备注	入学年份
选择	专升本	0	专升本通知
			2004

添加新的为您服务

服务ID	1
服务名称	专升本
服务类别 (NUM)	0
内容	专升本通知。..... 这里可以包含HTML内容。一些服务或者通知。
专业 (NUM)	0
简明备注	专升本通知
入学年份	2004
编辑 新建 删除	

注意:专业项填写 0 为该入学年度所有专业都可见该服务项。(只有超级管理员有效)
内容最多只能6000字符,约等于6KB.

图 4.21 为您服务界面

管理员界面顶部菜单，有退出系统和插件部分的功能。默认情况下，系统中有 2 个插件，分别为社团管理和系统消息，是针对社团模块的控制。

[退出系统](#) 插件：[社团管理](#) [系统消息](#)

图 4.22 顶部菜单

4.3 管理员插件部分

4.3.1 社团管理

创建，批准，修改社团信息。

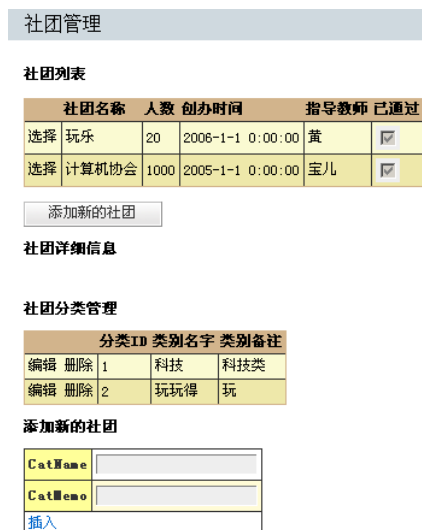


图 4.23 社团管理

4.3.2 社团消息

修改社团模块的系统消息。



图 4.24 社团系统消息

4.4 统计模块

用于比较和统计学生的成绩。管理员登陆之后，首先要进行统计的专业及班级设置。

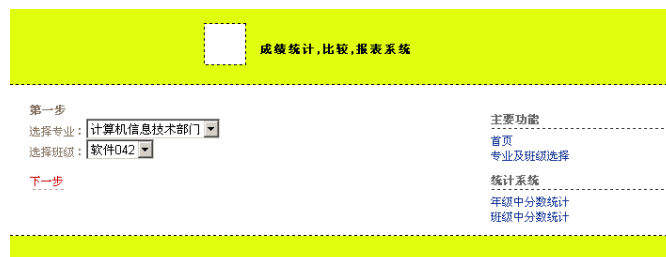


图 4.25 统计模块之第一步



图 4.26 统计模块之第二步

OK!设定完成,您可以随时点击右边的导航菜单返回这里设置。现在您可以进行统计操作。

图 4.27 统计完成

班级中分数统计，每一项科目中有该班级中最高，最低，和平均分统计。

科目	最高分	最低分	平均分
语文	70	30	53.33333333333333

图 4.28 成绩统计

4.5 学生模块

以学生身份登陆之后，出现学生的操作界面。默认界面是专业和班级的通知。



图 4.29 学生模块

4.5.1 个人消息

用于存放教师发送给学生的信息，可以回复一次。



图 4.30 个人消息

4.5.2 综合评定

当给学生综合评定完毕后,选择了发布,该生就能从学生界面看到每个学期的综合评定。

学期	第2学期
D1-小组评分	23
D2-班主任评分	23
D3-寝室卫生评分	43
D4-处分及班干部评分	0
Z1-各科成绩公式计算	50
Z2-证书及校级获奖	0
T1-体育成绩	0
T2-出勤率分数	2
T3-体育类获奖分	0
A-其他附加分数	0
F-最终评定分	51.28

计算公式:
 $D_{总分} = D1 \times 0.5 + D2 \times 0.3 + D3 \times 0.2 + D4$
 $Z_{总分} = Z1 + Z2$
 $T_{总分} = T1 \times 0.6 + T2 \times 0.4 + T3$
 $F_{最终评定} = D_{总分} \times 0.2 + Z_{总分} \times 0.7 + T_{总分} \times 0.1$

评定的分数是该学期与上学期的所有分数。即一个学年。

图 4.31 个人综合评定

以下三项分别可以查询该生登陆对应的学生成绩,旷课请假,奖励惩罚信息。



图 4.32 菜单

奖励

获奖日期	奖励项目	名次	荣誉	评测分	学期
2007-1-1 0:00:00	全国美男子	一等奖	美男	100	1

处罚

没有任何处罚或者处分记录,非常不错。

图 4.33 奖惩情况

4.6 手机访问模块

手机访问该系统时，会自动定到手机页面。页面编码全采用手机浏览器规范。

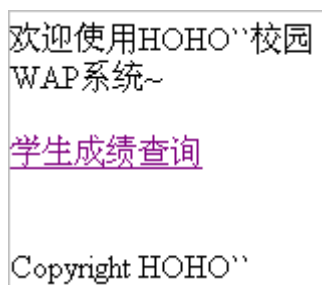


图 4.34 手机访问界面

选中成绩查询可以进入学生登陆界面。

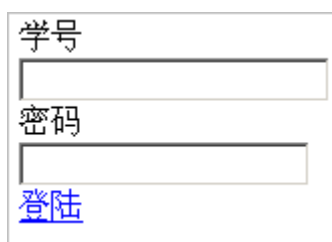


图 4.35 手机登陆系统界面

填写学号和密码后登陆。显示查询成绩界面。

可以选择更换学生登陆（方便用一个手机查询 N 个学生成绩）或者返回主界面（使用其他功能）。选择要查询的学期成绩和科目的关键字（词）。

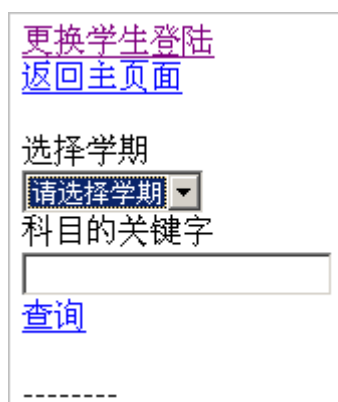


图 4.36 手机查询界面

选择学期和输入科目关键字查询，关键字为空查询该学期所有科目成绩。选择第一学期后，关键字留空，出现以下界面。有多个结果将会分页显示。

考试日期:2006-5-4 0:00:00
 考试科目:数学
 考试成绩:45
 总评成绩:78
 获得学分0
 科目学分:4

图 4.37 查询结果

4.7 社团模块

登陆社团操作界面，显示如下界面。默认主画面显示所有存在的社团。



图 4.38 社团界面

4.7.1 我的社团

显示已经加入的社团，点击后可以在右边显示该社团的详细信息。



图 4.39 我的社团

4.7.2 我的消息

有个人消息项，点击后可以发送和接收个人消息。类似短消息。

消息ID	姓名	女性?	学号	已读?	发送时间	级别
选择 10	阿邦	<input checked="" type="checkbox"/>	504011151	<input type="checkbox"/>	2007-2-6 19:53:00	1
10						
123456789						
删除						
发送消息到			504011151			
消息						
<input type="button" value="发送消息"/>						

图 4.40 社团里个人消息

4.7.3 退出社团

退出社团功能用于当学生不再想参加一个社团时，可以自主退出该社团。该操作不能撤销。

您可以选择退出的社团			
	社团ID	社团名	学号
退出该社团	2	玩乐	504011151
退出该社团	1	计算机协会	504011151
你确定要退出选择的社团?			
确定			
p.s: 社团管理人退出社团不会有任何影响。			

图 4.41 推出社团

4.7.4 加入社团

申请想加入的社团，经社团的管理员批准后并设定相应的职位部门后，即可成为社团成员。

选择	社团名	介绍	人数	
选择	玩乐	没有什么介绍	20	2006-1-
选择	计算机协会	没有什么介绍好介绍的。	1000	2005-1-
没有什么介绍好介绍的。				
计算机的协会Y 计算机的协会Y 计算机的协会Y 计算机的协会Y 计算机的协会Y 计算机				
这里填写加入社团的申请消息				
<input type="button" value="加入该社团"/>				

图 4.42 加入社团

右上有一行快捷操作。快捷定位到常用的操作界面。

成绩查询 | 系统消息 | 我的消息 | 社团管理 | 帮助 | 退出

图 4.42 快捷定位

1. 成绩查询：转到学生成绩查询界面。
2. 系统消息：显示管理员用插件设定的系统消息。
3. 我的消息：显示该学生接收的所有消息。
4. 社团管理：显示可以管理的社团，选择相应的社团可以进行详细的操作。
5. 帮 助：社团的操作帮助。
6. 退 出：退出学生登陆。

4.7.5 系统消息

显示一些重要的通知。

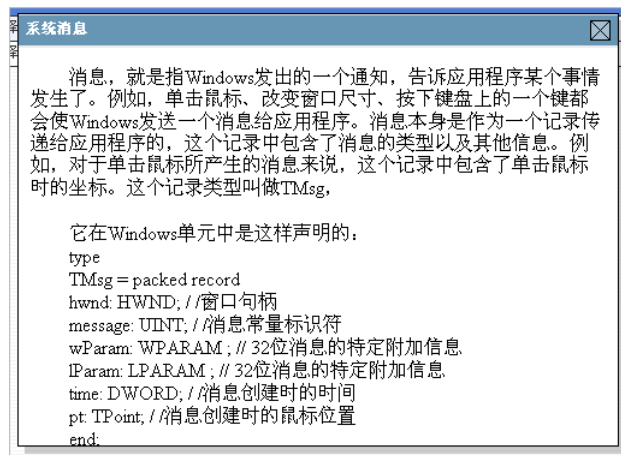


图 4.43 显示系统重要通知的界面

4.7.6 社团管理

选择后出现详细的每一项可管理项。



图 4.44 社团管理

(a) 验证加入成员

社团管理人员在这里通过或者拒绝提出加入社团的成员请求。



图 4.45 验证新进成员

(b) 社团信息维护

上传社团标志。

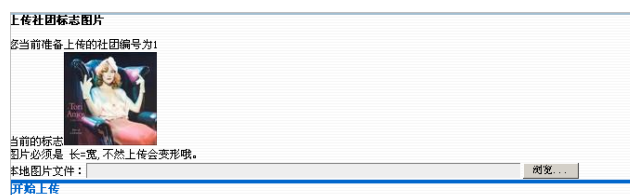


图 4.46 上传社团标志

(c) 成员管理

重新调整社团成员职务和部分或者踢出社团成员。



图 4.47 社团成员管理

(d) 结构管理

调整社团的整体结构。删除，添加。

The screenshot shows a web interface for managing club structure. It is divided into two main sections: '删除项目' (Delete Item) and '添加' (Add).

删除项目 (Delete Item):

部门ID	部门名称	部门说明
选择 1	会长	会长助理

职务ID	职务名称	职务说明
选择 3	会长助理	会长助理

添加 (Add):

名称:

说明:

添加职务 添加部门

[添加新的项目](#)

图 4.48 社团结构调整

第五章 系统功能实现

5.1 检查是否学生在一专业中的类

```
ChkStuInZY.cs
public class ChkStuInZY : IDisposable
{
    private int ZYid;
    private SqlConnection conn = new SqlConnection();
    private SqlCommand cmd = new SqlCommand();
    public ChkStuInZY(int zhuanYeID)
    {
        conn.ConnectionString =
ConfigurationManager.ConnectionStrings["StudentConnectionString"].Conne
ctionString;
        conn.Open();
        cmd.Connection = conn;
        ZYid = zhuanYeID;
    }
    public int IsStuidINzy(int studentId)
    {
        //判断StudentID是否在指定的专业ID里
        cmd.Parameters.Clear();
        cmd.CommandText = "SELECT TOP 1 [StudentID] FROM
[Nov30th_Detail] WHERE [ZhuanYe]=@zhuanye AND [StudentID]=@studentid";
        cmd.Parameters.Add("@zhuanye", SqlDbType.Int).Value = ZYid;
        cmd.Parameters.Add("@studentid", SqlDbType.Int).Value =
studentId;
        return Convert.ToInt32(cmd.ExecuteScalar());
    }
}
```

5.2 数据库连接基类

ConnectDB.cs

```
public class ConnectDB : IDisposable
{
    private SqlConnection conn = new SqlConnection();
    private SqlCommand cmd = new SqlCommand();
    public SqlCommand Cmd
    {
        get { return cmd; }
        set { cmd = value; }
    }
    private SqlDataReader reader;
    public SqlDataReader Reader
    {
        get { return reader; }
        set { reader = value; }
    }
    private bool isConn;
    public bool IsConn
    {
        get { return isConn; }
    }
    private bool doNotReplaceCmd;
    public bool DoNotReplaceCmd
    {
        get { return doNotReplaceCmd; }
        set { doNotReplaceCmd = value; }
    }

    public ConnectDB()
```

```
{
    conn.ConnectionString =
ConfigurationManager.ConnectionStrings["StudentConnectionString"].Conn
ectionString;
    {
        conn.Open();
        cmd.Connection = conn;
        isConn = true;
    }
    /// <summary>
    /// Create Reader
    /// </summary>
    /// <param name="strcmd">SQL Command</param>
    /// <returns>Is success</returns>
    public bool ExeReader(string strcmd)
    {
        bool isSuccess = false;
        if (!isConn)
            return false;
        try
        {
            cmd.CommandText = strcmd;
            reader = cmd.ExecuteReader();
            isSuccess = true;
        }
        catch
        {
            //do something
        }
        return isSuccess;
    }
    /// <summary>
```

```
/// 检查管理员的用户名和 MD5 密码，并更新登陆次数
/// </summary>
/// <param name="strusername">用户名</param>
/// <param name="strpassword">MD5 加密后的密码</param>
/// <returns>
/// -1 处理错误
/// -2 输入的数据不符合要求
/// 0 用户名或者密码错误
/// 1 用户名和密码正确
/// </returns>
public int CheckAP(string strusername, string strpassword)
{
    int retval = -1;
    if (strusername.Length < 4 || strpassword.Length != 32)
        return -2;
    cmd.CommandText = "update [Admin] set [LoginTimes] =
[LoginTimes] + 1 where [Name] = @username and [Password] = @password";
    cmd.Parameters.Clear();
    cmd.Parameters.Add("@username", SqlDbType.VarChar,
50).Value = strusername;
    cmd.Parameters.Add("@password", SqlDbType.VarChar,
32).Value = strpassword;
    retval = cmd.ExecuteNonQuery();
    return retval;
}
/// <summary>
/// Create Reader Without Command
/// </summary>
/// <returns>是否创建 Reader 成功</returns>
public bool ExeReader()
{
    bool isSuccess = false;
```

```
if (!isConn)
    return false;
try
{
    reader = cmd.ExecuteReader();
    isSuccess = true;
}
catch
{
    //do something
}
return isSuccess;
}
```

5.3 手机 WAP 访问类

```
MobileStudentClass.cs
/// <summary>
/// Can Login With StudentId And Password?
/// </summary>
/// <param name="passWord">Password Without MD5 HASH</param>
/// <returns>LoginLevel</returns>
public int StudentLogin(string passWord)
{
    object retSql;
    Cmd.Parameters.Clear();
    Cmd.CommandText = "SELECT TOP 1 [LoginLevel] FROM
[Nov30th_Detail] WHERE [StudentID]=@stuid AND [Password]=@pwd";
    Cmd.Parameters.Add("@stuid", SqlDbType.Int).Value =
studentId;
    Cmd.Parameters.Add("@pwd", SqlDbType.VarChar, 32).Value =
System.Web.Security.FormsAuthentication.HashPasswordForStoringInConfig
```

```

File(passWord, "MD5").ToLower();

    retSql = Cmd.ExecuteScalar();
    if (DBNull.Value == retSql)
        return -1;//Not Pass
    else
    {
        return Convert.ToInt32(retSql);
    }
}

/// <summary>
/// Return String including Student's Mark
/// </summary>
/// <param name="term">Term</param>
/// <param name="keyWords">KeyWords of Subject</param>
/// <returns>String including Student's Mark(WAP)</returns>
public string ShowWapMark(byte term, string keyWords)
{
    StringBuilder strapp = new StringBuilder("SELECT [Date],
[Expr1], [Normal], [Repair], [MakeUp], [Overall], [GetCredit], [Credit]
FROM [View_Mark] WHERE [StudentID]=@stuid AND [Term]=@term");
    StringBuilder retstr = new StringBuilder("-以下是该学期成绩
-<br />");
    Cmd.Parameters.Clear();
    if (!string.IsNullOrEmpty(keyWords))
        strapp.Append(" AND [Expr1] LIKE @keywords");
    strapp.Append(" ORDER BY [Date] DESC");
    Cmd.CommandText = strapp.ToString();
    Cmd.Parameters.Add("@stuid", SqlDbType.Int).Value =
studentId;
    Cmd.Parameters.Add("@term", SqlDbType.TinyInt).Value =
term;
    if (!string.IsNullOrEmpty(keyWords))

```

```
Cmd.Parameters.Add("@keywords", SqlDbType.VarChar,
20).Value = "%" + keyWords + "%";
DoNotReplaceCmd = true;
ExeReader();
while (Reader.Read())
{
    retstr.Append("<br />考试日期:");
    retstr.Append(Reader.GetDateTime(0));
    retstr.Append("<br />考试科目:");
    retstr.Append(Reader.GetString(1));
    retstr.Append("<br />考试成绩:");
    retstr.Append(Reader.GetDouble(2));
    if (Reader.GetDouble(3) >= 1)
    {
        retstr.Append("<br />补考成绩:");
        retstr.Append(Reader.GetDouble(3));
    }
    if (Reader.GetDouble(4) >= 1)
    {
        retstr.Append("<br />重修成绩:");
        retstr.Append(Reader.GetDouble(4));
    }
    retstr.Append("<br />总评成绩:");
    retstr.Append(Reader.GetDouble(5));
    retstr.Append("<br />获得学分");
    retstr.Append(Reader.GetDouble(6));
    retstr.Append("<br />科目学分:");
    retstr.Append(Reader.GetDouble(7));
    retstr.Append("<br />=====");
}
Reader.Close();
retstr.Append("<br />-成绩显示结束-");
```

```
        return retstr.ToString();
    }
}
```

5.4 社团控制类

OrganControl.cs

```
    /// <summary>
    /// Check if is The Organ's Leader
    /// </summary>
    /// <param name="istudentId">Studnet ID</param>
    /// <returns>Result</returns>
    public bool GetCanControl(int studentId)
    {
        Cmd.Parameters.Clear();
        Cmd.CommandText = "SELECT COUNT(1) FROM [Nov30th_Organ] WHERE
[OrganCharge]=@stuid AND [OrganID]=@organid";
        Cmd.Parameters.Add("@stuid", SqlDbType.Int).Value =
studentId;
        Cmd.Parameters.Add("@organid", SqlDbType.Int).Value =
organId;
        object obj = Cmd.ExecuteScalar();
        if (obj == null)
        {
            return false;
        }
        else
        {
            return obj.ToString() == "0" ? false : true;
        }
    }
    /// <summary>
```



```
    /// Read Organ Information
    /// </summary>
    /// <returns>
    /// -1:Create Reader Error
    /// 0:No Such Information
    /// 1:Success
    /// </returns>
    public int GetOrganInfo()
    {
        Cmd.Parameters.Clear();
        Cmd.CommandText = "SELECT TOP 1 [OrganName], [OrganIntro],
[People], [OrganCharge], [OrganCreateTime], [OrganTeacher], [CatID],
[IsActive], [OrganMemo] FROM [Nov30th_Organ] WHERE [OrganID]=@organId";
        Cmd.Parameters.Add("@organId", SqlDbType.Int).Value =
organId;

        DoNotReplaceCmd = true;
        if (!ExeReader())
            return -1;
        if (!Reader.Read())
        {
            Reader.Close();
            return 0;
        }
        organName = Reader.GetString(0);
        organIntro = Reader.GetString(1);
        organPeople = Reader.GetInt32(2);
        organCharge = Reader.GetInt32(3);
        organCreateTime = Reader.GetDateTime(4);
        organTeacher = Reader.GetString(5);
        catID = Reader.GetInt32(6);
        isActive = Reader.GetBoolean(7);
        organMemo = Reader.GetString(8);
```

```

        Reader.Close();
        return 1;
    }
    /// <summary>
    /// Accept Member To Organ
    /// </summary>
    /// <param name="memberId">Member ID</param>
    /// <param name="buMenId">Bu Men ID</param>
    /// <param name="zhiWuId">Zhi Wu ID</param>
    /// <returns>Is Success?</returns>
    public bool SetAcceptMember(int memberId, int buMenId, int
zhiWuId)
    {
        Cmd.Parameters.Clear();
        Cmd.CommandText = "UPDATE [Nov30th_Members] SET [IsReg]=0,
[BuMenID] = @bumenid, [ZhiWuId] = @zhiwuid WHERE [MemberID]=@memberid AND
[OrganId]=@organid";
        Cmd.Parameters.Add("@bumenid", SqlDbType.Int).Value =
buMenId;
        Cmd.Parameters.Add("@zhiwuid", SqlDbType.Int).Value =
zhiWuId;
        Cmd.Parameters.Add("memberid", SqlDbType.Int).Value =
memberId;
        Cmd.Parameters.Add("organid", SqlDbType.Int).Value =
organId;
        return Cmd.ExecuteNonQuery() > 0 ? true : false;
    }
    /// <summary>
    /// Create Bu Men
    /// </summary>
    /// <param name="buMenName">Bu Men Name</param>
    /// <param name="buMenMemo">Bu Men Memo</param>

```

```

    /// <returns>Is Success?</returns>
    public bool InsertBuMen(string buMenName, string buMenMemo)
    {
        if (CountBuMen() > BUMENMAXCOUNT)
            return false;
        Cmd.Parameters.Clear();
        Cmd.CommandText = "INSERT INTO [Nov30th_BuMen] ([BuMenName],
[BuMenMemo], [OrganID]) VALUES (@BuMenName, @BuMenMemo, @OrganID)";
        Cmd.Parameters.Add("@BuMenName", SqlDbType.VarChar,
20).Value = buMenName.Substring(0, buMenName.Length > 20 ? 20 :
buMenName.Length);
        Cmd.Parameters.Add("@BuMenMemo", SqlDbType.VarChar,
50).Value = buMenMemo.Substring(0, buMenMemo.Length > 50 ? 50 :
buMenMemo.Length);
        Cmd.Parameters.Add("@OrganID", SqlDbType.Int).Value =
organId;
        return Cmd.ExecuteNonQuery() > 0 ? true : false;
    }
    /// <summary>
    /// Count The BuMen
    /// </summary>
    /// <returns>BuMen Amount</returns>
    public int CountBuMen()
    {
        Cmd.Parameters.Clear();
        Cmd.CommandText = "SELECT COUNT(1) FROM [Nov30th_BuMen] WHERE
[OrganID] = @OrganID";
        Cmd.Parameters.Add("OrganID", SqlDbType.Int).Value =
organId;
        return Convert.ToInt32(Cmd.ExecuteScalar());
    }
    /// <summary>

```

```

    /// Create Zhi Wu
    /// </summary>
    /// <param name="zhiWuName">Zhi Wu Name</param>
    /// <param name="zhiWuMemo">Zhi Wu Memo</param>
    /// <returns>Is Success?</returns>
    public bool InsertZhiWu(string zhiWuName, string zhiWuMemo)
    {
        if (CountZhiWu() >= ZHIWUMAXCOUNT)
            return false;

        Cmd.Parameters.Clear();

        Cmd.CommandText = "INSERT INTO [Nov30th_ZhiWu] ([ZhiWuName],
[ZhiWuMemo], [OrganID]) VALUES (@ZhiWuName,@ZhiWuMemo,@OrganID)";

        Cmd.Parameters.Add("@ZhiWuName",      SqlDbType.VarChar,
20).Value = zhiWuName.Substring(0, zhiWuName.Length > 20 ? 20 :
zhiWuName.Length);

        Cmd.Parameters.Add("@ZhiWuMemo",      SqlDbType.VarChar,
50).Value = zhiWuMemo.Substring(0, zhiWuMemo.Length > 50 ? 50 :
zhiWuMemo.Length);

        Cmd.Parameters.Add("@OrganID",      SqlDbType.Int).Value =
organId;

        return Cmd.ExecuteNonQuery() == 0 ? false : true;
    }
    /// <summary>
    /// Count The ZhiWu
    /// </summary>
    /// <returns>ZhiWu Amount</returns>
    public int CountZhiWu()
    {
        Cmd.Parameters.Clear();

        Cmd.CommandText = "SELECT COUNT(1) FROM [Nov30th_ZhiWu] WHERE
[OrganID] = @OrganID";

        Cmd.Parameters.Add("OrganID",      SqlDbType.Int).Value =

```

```

organId;

        return Convert.ToInt32(Cmd.ExecuteScalar());
    }
    /// <summary>
    /// Count People in ZhiWu ID
    /// </summary>
    /// <param name="zhiWuId">Zhi Wu Id</param>
    /// <returns>People</returns>
    public int CountPeopleInZhiWu(int zhiWuId)
    {
        Cmd.Parameters.Clear();
        Cmd.CommandText = "SELECT COUNT(1) FROM [Nov30th_Members]
WHERE [ZhiWuID]=@ZhiWuId";
        Cmd.Parameters.Add("@ZhiWuId", SqlDbType.Int).Value =
zhiWuId;

        return Convert.ToInt32(Cmd.ExecuteScalar());
    }
    /// <summary>
    /// Count People in BuMen ID
    /// </summary>
    /// <param name="buMenId">Bu Men Id</param>
    /// <returns>People</returns>
    public int CountPeopleInBuMen(int buMenId)
    {
        Cmd.Parameters.Clear();
        Cmd.CommandText = "SELECT COUNT(1) FROM [Nov30th_Members]
WHERE [BuMenID]=@BuMenId";
        Cmd.Parameters.Add("@BuMenId", SqlDbType.Int).Value =
buMenId;

        return Convert.ToInt32(Cmd.ExecuteScalar());
    }
    /// <summary>

```

```
/// Delete bumen
/// </summary>
/// <param name="buMenId">bumen id</param>
/// <returns>is success</returns>
public bool DeleteBuMen(int buMenId)
{
    if (CountPeopleInBuMen(buMenId) != 0)
        return false;
    Cmd.Parameters.Clear();
    Cmd.CommandText = "DELETE FROM [Nov30th_BuMen] WHERE [BuMenID]
= @BuMenID";
    Cmd.Parameters.Add("@BuMenID", SqlDbType.Int).Value =
buMenId;
    return Cmd.ExecuteNonQuery() > 0 ? true : false;
}
/// <summary>
/// Delete zhiwu
/// </summary>
/// <param name="zhiWuId">zhiwu id</param>
/// <returns>is success</returns>
public bool DeleteZhiWu(int zhiWuId)
{
    if (CountPeopleInZhiWu(zhiWuId) != 0)
        return false;
    Cmd.Parameters.Clear();
    Cmd.CommandText = "DELETE FROM [Nov30th_ZhiWu] WHERE [ZhiWuID]
= @ZhiWuID";
    Cmd.Parameters.Add("@ZhiWuID", SqlDbType.Int).Value =
zhiWuId;
    return Cmd.ExecuteNonQuery() > 0 ? true : false;
}
/// <summary>
```

```

    /// Delete Member From organ
    /// </summary>
    /// <param name="memberId">Member ID</param>
    /// <returns>is success?</returns>
    public bool DeleteMember(int memberId)
    {
        Cmd.Parameters.Clear();
        Cmd.CommandText = "DELETE FROM [Nov30th_Members] WHERE
[MemberID]=@MemberID AND [OrganID]=@OrganID";
        Cmd.Parameters.Add("@MemberID", SqlDbType.Int).Value =
memberId;
        Cmd.Parameters.Add("@OrganID", SqlDbType.Int).Value =
organId;

        return Cmd.ExecuteNonQuery() > 0 ? true : false;
    }
    /// <summary>
    /// Get My Information In This Organ
    /// </summary>
    /// <param name="istudentId">Student ID</param>
    /// <returns>is success</returns>
    public bool GetMyInfo(int istudentId)
    {
        int studentId = 0;
        if (istudentId == 0)
        {
            if (HttpContext.Current.Session["Nov30th_StudentID"] !=
null)
                studentId =
Convert.ToInt32(HttpContext.Current.Session["Nov30th_StudentID"].ToStr
ing());
        }
        else

```

```
        studentId = istudentId;
        Cmd.Parameters.Clear();
        Cmd.CommandText = "SELECT [BuMenID], [ZhiWuID], [MemberID],
[JoinTime] FROM [Nov30th_Members] WHERE [OrganID]=@OrganID AND
[MemberStudentID]=@MemberStudentID";
        Cmd.Parameters.Add("@OrganID", SqlDbType.Int).Value =
organId;
        Cmd.Parameters.Add("MemberStudentID", SqlDbType.Int).Value
= studentId;
        DoNotReplaceCmd = true;
        if (!ExeReader())
            return false;
        if (!Reader.Read())
        {
            Reader.Close();
            return false;
        }
        myBuMen = Reader.GetInt32(0);
        myZhiWu = Reader.GetInt32(1);
        myMemberId = Reader.GetInt32(2);
        myJoinedTime = Reader.GetDateTime(3);
        Reader.Close();
        return true;
    }
}
```

5.5 社团成员操作类

OrganMember.cs

```
public string ShowJoinedOrgan()
{
    StringBuilder retStr = new StringBuilder();
```



```

        Cmd.Parameters.Clear();

        Cmd.CommandText = "SELECT [OrganName], [OrganID] FROM
[View_MemberToOrgan] WHERE [MemberStudentID]=@StudentId AND [IsReg]=0
AND [IsActive]=1";

        Cmd.Parameters.Add("@StudentId", SqlDbType.Int).Value =
studentId;

        DoNotReplaceCmd = true;
        if (!ExeReader())
            return string.Empty;
        while (Reader.Read())
        {
            retStr.Append(Reader.GetString(0));
            retStr.Append("|");
            retStr.Append(Reader.GetInt32(1));
            retStr.Append("|");
        }
        Reader.Close();
        return retStr.ToString();
    }

    /// <summary>
    /// Return Is Joined The Organ
    /// </summary>
    /// <param name="organId">Organ ID</param>
    /// <param name="isReged">
    /// 1:Reged
    /// -1:Reging
    /// 0:All Records
    /// </param>
    /// <returns>Joined?</returns>
    public bool IsJoined(int organId, int isReged)
    {
        StringBuilder sql = new StringBuilder("SELECT TOP 1 [NickName]

```

```

FROM [Nov30th_Members] WHERE [MemberStudentID]=@StudentId AND
[OrganID]=@OrganId");
        if (isReged == 1)
            sql.Append(" AND [IsReg]=0");
        else if (isReged == 0)
            sql.Append(" AND [IsReg]=1");
        Cmd.Parameters.Clear();
        Cmd.CommandText = sql.ToString(); ;
        Cmd.Parameters.Add("@StudentId", SqlDbType.Int).Value =
studentId;
        Cmd.Parameters.Add("@OrganId", SqlDbType.Int).Value =
organId;

return !string.IsNullOrEmpty(Convert.ToString(Cmd.ExecuteScalar()));
    }
    /// <summary>
    /// Join Organ
    /// </summary>
    /// <param name="organId">Organ ID</param>
    /// <param name="regMemo">Reg Text</param>
    /// <returns>Success?</returns>
    public bool JoinOrgan(int organId, string regMemo)
    {
        if (IsJoined(organId, 0) || string.IsNullOrEmpty(regMemo))
            return false;
        Cmd.Parameters.Clear();
        Cmd.CommandText = "INSERT INTO [Nov30th_Members]
(MemberStudentID, NickName, Contact, OrganID, ZhiWuID, BuMenID, JoinTime,
IsReg, RegMemo) VALUES (@MemberStudentID, '-', '-', @OrganID, 0, 0,
getdate(), 1, @RegMemo)";
        Cmd.Parameters.Add("@MemberStudentID",
SqlDbType.Int).Value = studentId;

```

```

        Cmd.Parameters.Add("@OrganID", SqlDbType.Int).Value =
organId;

        Cmd.Parameters.Add("@RegMemo", SqlDbType.VarChar, 50).Value
= regMemo.Substring(0, regMemo.Length < 50 ? regMemo.Length : 50);
        return (Cmd.ExecuteNonQuery() == 0) ? false : true;
    }
}

```

5.6 社团消息操作类

```

OrganMessage.cs

    /// <summary>
    /// Send Private Message To Student
    /// </summary>
    /// <param name="recSutdentId">DestStudentId</param>
    /// <param name="messageContext">MessageContext</param>
    /// <param name="messageSign">MessageSign</param>
    /// <returns>ExecuteNonQuery Value</returns>
    public bool SendPrivateMessage(int recSutdentId, string
messageContext, byte messageSign)
    {
        Cmd.Parameters.Clear();
        Cmd.CommandText = "INSERT INTO [Nov30th_PersonalMessages]
(SendStudentID, ReceiveStudentID, IsRead, MessageContext, SendTime,
MessageSign) VALUES (@sstuid, @rstuid, 0, @context, getdate(),
@msgsign)";
        Cmd.Parameters.Add("@sstuid", SqlDbType.Int).Value =
studentId;
        Cmd.Parameters.Add("@rstuid", SqlDbType.Int).Value =
recSutdentId;
        Cmd.Parameters.Add("@context", SqlDbType.VarChar,
255).Value = messageContext;

```

```

        Cmd.Parameters.Add("@msgsign", SqlDbType.TinyInt).Value =
messageSign;

        return Cmd.ExecuteNonQuery() > 0 ? true : false;
    }

    /// <summary>
    /// Count The Message Sent
    /// </summary>
    /// <param name="messageType">
    /// 1. Personal
    /// 2. Organ
    /// 3. HuoDong
    /// </param>
    /// <returns>Count</returns>
    public int CountSentMessages(int messageType)
    {
        Cmd.Parameters.Clear();

        Cmd.CommandText = "SELECT COUNT(1) FROM [Nov30th_" +
(messageType == 1 ? "PersonalMessages" : (messageType == 2 ?
"OrganMessages" : "HuoDongMessages")) + "]" WHERE [StudentID]=@stuid";

        Cmd.Parameters.Add("@stuid", SqlDbType.Int).Value =
studentId;

        return Convert.ToInt32(Cmd.ExecuteScalar());
    }

    /// <summary>
    /// Count The Private Message Received
    /// </summary>
    /// <returns>Count</returns>
    public int CountReceivedMessages()
    {
        Cmd.Parameters.Clear();

        Cmd.CommandText = "SELECT COUNT(1) FROM
[Nov30th_PersonalMessages] WHERE [ReceiveStudentID]=@stuid";

```

```

        Cmd.Parameters.Add("@stuid", SqlDbType.Int).Value =
studentId;

        return Convert.ToInt32(Cmd.ExecuteScalar());
    }

    /// <summary>
    /// The New Private Message Amount
    /// </summary>
    /// <returns></returns>
    public int HasNewMessages()
    {
        Cmd.Parameters.Clear();

        Cmd.CommandText = "SELECT COUNT(1) FROM
[Nov30th_PersonalMessages] WHERE [ReceiveStudentID]=@stuid AND IsRead =
0";

        Cmd.Parameters.Add("@stuid", SqlDbType.Int).Value =
studentId;

        return Convert.ToInt32(Cmd.ExecuteScalar());
    }

    /// <summary>
    /// Send Message To Organ
    /// </summary>
    /// <param name="OrganId">OrganID</param>
    /// <param name="buMenId">BuMenID</param>
    /// <param name="zhiWuId">ZhiWuID</param>
    /// <param name="messageContext">Message Context</param>
    /// <param name="messageType">NO USE, Type 0 Instand</param>
    /// <returns></returns>
    public int SendOrganMessage(int organId, int buMenId, int zhiWuId,
string messageContext, int messageType)
    {
        Cmd.Parameters.Clear();

        Cmd.CommandText = "INSERT INTO [Nov30th_OrganMessages]

```

```

(SendStudentID, OrganID, AIMTpye, BuMenID, ZhiWuID, MessageContext,
SendTime) VALUES (@sstuid, @organid, @aimtype, @bumenid, @zhiwuid,
@context, getdate())";

        Cmd.Parameters.Add("@sstuid", SqlDbType.Int).Value =
studentId;

        Cmd.Parameters.Add("@organid", SqlDbType.Int).Value =
organId;

        Cmd.Parameters.Add("@aimtype", SqlDbType.Int).Value =
messageType;

        Cmd.Parameters.Add("@bumenid", SqlDbType.Int).Value =
buMenId;

        Cmd.Parameters.Add("@zhiwuid", SqlDbType.Int).Value =
zhiWuId;

        Cmd.Parameters.Add("@context", SqlDbType.VarChar,
255).Value = messageContext;

        return Cmd.ExecuteNonQuery();
    }

    /// <summary>
    /// Send Huo Dong Message
    /// </summary>
    /// <param name="huoDongId">HuoDongID</param>
    /// <param name="messageContext">Message Context</param>
    /// <param name="messageSign">Message Sign</param>
    /// <returns>ExecuteNonQuery Value</returns>
    public int SendHuoDongMessage(int huoDongId, string
messageContext, byte messageSign)
    {
        Cmd.Parameters.Clear();

        Cmd.CommandText = "INSERT INTO [Nov30th_PersonalMessages]
(HuoDongID, SendStudentID, SendTime, MessageContext, MessageSign) VALUES
(@huodongid, @sstuid, getdate(), @context, @msgsign)";

        Cmd.Parameters.Add("@huodongid", SqlDbType.Int).Value =

```

```

huoDongId;
        Cmd.Parameters.Add("@sstuid", SqlDbType.Int).Value =
studentId;
        Cmd.Parameters.Add("@context", SqlDbType.VarChar,
255).Value = messageContext;
        Cmd.Parameters.Add("@msgsign", SqlDbType.TinyInt).Value =
messageSign;
        return Cmd.ExecuteNonQuery();
    }
    /// <summary>
    /// is exists student?
    /// </summary>
    /// <param name="studentId">Student ID</param>
    /// <returns>exists?</returns>
    public bool IsExistStudent(int studentId)
    {
        Cmd.Parameters.Clear();
        Cmd.CommandText = "SELECT COUNT(1) FROM [Nov30th_Detail]
WHERE [StudentID]=@StudentID";
        Cmd.Parameters.Add("@StudentID", SqlDbType.Int).Value =
studentId;
        return Convert.ToInt32(Cmd.ExecuteScalar()) == 0 ? false :
true;
    }
}

```

5.7 公共权限检查类

```

PublicCheck.cs
    /// <summary>
    /// 检查允许登陆的情况
    /// Version 0.01

```

```
/// Code By HOHO`~  
/// </summary>  
public class PublicCheck  
{  
    private bool isSafeMode, isStuAllow, isNormalAllow, isParAllow;  
    public bool IsParAllow  
    {  
        get { return isParAllow; }  
        set { isParAllow = value; }  
    }  
    public bool IsNormalAllow  
    {  
        get { return isNormalAllow; }  
        set { isNormalAllow = value; }  
    }  
    public bool IsStuAllow  
    {  
        get { return isStuAllow; }  
        set { isStuAllow = value; }  
    }  
    public bool IsSafeMode  
    {  
        get { return isSafeMode; }  
    }  
    /// <summary>  
    /// 构造函数  
    /// </summary>  
    /// <param name="strIP_Address">客户机的 IP 地址</param>  
    /// <param name="SafeMode">SafeMode 的全局网站变量</param>  
    /// <param name="StuLogin">StuLogin 的全局网站变量</param>  
    /// <param name="ParLogin">ParLogin 的全局网站变量</param>  
    /// <param name="AllowPub">AllowPub 的全局网站变量</param>
```



```
public PublicCheck()
{
    bool isPriIP = false;
    string          strIP_Address          =
HttpContext.Current.Request.ServerVariables["REMOTE_HOST"].ToString();
    int             safeMode              =
Convert.ToInt32(HttpContext.Current.Application["SafeMode"]);
    int             stuLogin              =
Convert.ToInt32(HttpContext.Current.Application["StuLogin"]);
    int             parLogin              =
Convert.ToInt32(HttpContext.Current.Application["ParLogin"]);
    int             allowPub              =
Convert.ToInt32(HttpContext.Current.Application["AllowPub"]);
    //如果不允许外网
    if (allowPub != 1)
    {
        if (strIP_Address.ToString() == "127.0.0.1")
            isPriIP = true;
        if (strIP_Address.Substring(0, 7).ToString() ==
"192.168")
            isPriIP = true;
        if (strIP_Address.Substring(0, 4).ToString() == "172."
&& Convert.ToInt16(strIP_Address.Substring(4, 3)) >= 16 &&
Convert.ToInt16(strIP_Address.Substring(4, 3)) <= 31)
            isPriIP = true;
        if (strIP_Address.Substring(0, 3).ToString() == "10.")
            isPriIP = true;
        if (!isPriIP)
            return;
    }
    isNormalAllow = true;
    //安全模式
```

```
        if (safeMode == 1)
        {
            isSafeMode = true;
            return;
        }
        isStuAllow = (stuLogin == 1) ? true : false;
        isParAllow = (parLogin == 1) ? true : false;
    }
}
```

5.8 安全检查类

SafeCheck.cs

```
public class SafeCheck
{
    private int bytLevel = -1;
    private bool bisOnly = true;
    const byte CStudent = 100;
    const byte CParent = 101;
    const byte CAdmin = 254;
    const byte CSuperAdmin = 255;
    const byte CMaster = 250;
    const byte CStudentWap = 100;
    public SafeCheck(byte level, bool only)
    {
        bytLevel = level;
        bisOnly = only;
    }
    ~SafeCheck()
    {
    }
}
/// <summary>
```

```
/// 验证登陆是否正确
/// 学生 WAP 登陆，需要级别 100
/// </summary>
/// <param name="bytLevel">登陆的级别</param>
/// <param name="bisOnly">是否允许高级别登陆</param>
/// <returns>Boolean</returns>
public bool ChkWAPStudent()
{
    if (bytLevel >= CStudent && bisOnly == false)
        return true;
    else if (bytLevel == CStudent && bisOnly == true)
        return true;
    else
        return false;
}
/// <summary>
/// 验证登陆是否正确
/// 学生登陆，需要级别 100
/// </summary>
/// <param name="bytLevel">登陆的级别</param>
/// <param name="bisOnly">是否允许高级别登陆</param>
/// <returns>Boolean</returns>
public bool ChkStudent()
{
    if (bytLevel >= CStudent && bisOnly == false)
        return true;
    else if (bytLevel == CStudent && bisOnly == true)
        return true;
    else
        return false;
}
/// <summary>
```

```
/// 验证登陆是否正确
/// 班主任登陆，需要级别 250
/// </summary>
/// <param name="bytLevel">登陆的级别</param>
/// <param name="bisOnly">是否允许高级别登陆</param>
/// <returns>Boolean</returns>
public bool ChkMaster()
{
    if (bytLevel >= CMaster && bisOnly == false)
        return true;
    else if (bytLevel == CMaster && bisOnly == true)
        return true;
    else
        return false;
}
/// <summary>
/// 验证登陆是否正确
/// 管理员登陆，需要级别 200
/// </summary>
/// <param name="bytLevel">登陆的级别</param>
/// <param name="bisOnly">是否允许高级别登陆</param>
/// <returns>Boolean</returns>
public bool ChkAdmin()
{
    if (bytLevel >= CAdmin && bisOnly == false)
        return true;
    else if (bytLevel == CAdmin && bisOnly == true)
        return true;
    else
        return false;
}
/// <summary>
```

```
    /// 验证登陆是否正确
    /// 超级管理员登陆，需要级别 255
    /// </summary>
    /// <param name="bytLevel">登陆的级别</param>
    /// <param name="bisOnly">无所谓的参数</param>
    /// <returns>Boolean</returns>
    public bool ChkSuperAdmin()
    {
        if (bytLevel == CSuperAdmin)
            return true;
        else
            return false;
    }
    /// <summary>
    /// 验证登陆是否正确
    /// 家长登陆，需要级别 150
    /// </summary>
    /// <param name="bytLevel">登陆的级别</param>
    /// <param name="bisOnly">是否允许高级别登陆</param>
    /// <returns>Boolean</returns>
    public bool ChkParent()
    {
        if (bytLevel >= CParent && bisOnly == false)
            return true;
        else if (bytLevel == CParent && bisOnly == true)
            return true;
        else
            return false;
    }
}
```

5.9 学生信息操作类

```
StudentInfo.cs
public class StudentInfo
{
    private bool isConn;//is DB connected?
    public bool IsConn
    {
        get { return isConn; }
        set { isConn = value; }
    }
    private int intStudentId;//Save The StudentID
    public int IntStudentID
    {
        get { return intStudentId; }
    }
    private byte errorId;//error id
    public byte ErrorId
    {
        get { return errorId; }
    }
    private string errorText;//error text
    public string ErrorText
    {
        get { return errorText; }
    }
    ///Read back information
    private string StudentName;
    public string StudentNameOut
    {
        get { return StudentName; }
    }
}
```

```
}  
private string Dormitory;  
public string DormitoryOut  
{  
    get { return Dormitory; }  
}  
private string Password;  
public string PasswordOut  
{  
    get { return Password; }  
}  
private int ruXue;  
public int RuXueOut  
{  
    get { return ruXue; }  
}  
private bool Sex;  
public bool SexOut  
{  
    get { return Sex; }  
}  
private byte LoginLevel;  
public byte LoginLevelOut  
{  
    get { return LoginLevel; }  
}  
private int ClassId;  
public int ClassIDOut  
{  
    get { return ClassId; }  
}  
private string CellPhone;
```

```
public string CellPhoneOut
{
    get { return CellPhone; }
}
private DateTime Birthday;
public DateTime BirthdayOut
{
    get { return Birthday; }
}
private int zhuanYeId;
public int ZhuanYeIDOut
{
    get { return zhuanYeId; }
}
private string ZhuanYeName;
public string ZhuanYeNameOut
{
    get { return ZhuanYeName; }
}
private string ClassName;
public string ClassNameOut
{
    get { return ClassName; }
}
private string ClassMaster;
public string ClassMasterOut
{
    get { return ClassMaster; }
}
private string MasterName;
public string MasterNameOut
{
```



```
        get { return MasterName; }
    }
    private byte Term;
    public byte TermOut
    {
        get { return Term; }
    }
    /// <summary>
    /// Read Student All Information From DataBase With StuID
    /// </summary>
    /// <param name="Studentid">Student ID</param>
    public StudentInfo(int studentId)
    {
        //Create Conn and Read Info
        SqlConnection conn = new SqlConnection();
        SqlCommand cmd = new SqlCommand();
        conn.ConnectionString =
ConfigurationManager.ConnectionStrings["StudentConnectionString"].Conn
ectionString;
        try
        {
            conn.Open(); //open database connection
            cmd.Connection = conn;
            IsConn = true;
        }
        catch
        {
            //do sth
        }
        if (!IsConn)
        {
            //if db open failed ,return.
```

```
        errorId = 1;
        errorText = "Can not connect Database";
        cmd.Dispose();
        conn.Close();
        conn.Dispose();
        return;
    }
    cmd.CommandText = "SELECT TOP 1 [StudentID], [Name],
[Dormitory], [IsPM], [Birthday], [CellPhone], [Sex], [ClassName],
[ZhuanYe], [ZY_Name], [Master], [Term], [RuXue], [Password], [ClassID],
[LoginLevel], [Master] FROM [View_Student] Where
[StudentID]=@studentid";
    cmd.Parameters.Add("@studentid", SqlDbType.Int).Value =
studentId;

    SqlDataReader dr = cmd.ExecuteReader();
    if (!dr.Read())
    {
        //if no Data With This Student ID, return.
        dr.Close();
        cmd.Dispose();
        conn.Close();
        conn.Dispose();
        errorId = 2;
        errorText = "No Data With This StudentID";
        return;
    }
    //copy student info here
    intStudentId = dr.GetInt32(0);
    StudentName = dr.GetString(1);
    Dormitory = dr.GetString(2);
    Birthday = dr.GetDateTime(4);
    CellPhone = dr.GetString(5);
```

```
Sex = dr.GetBoolean(6);
ClassName = dr.GetString(7);
zhuanYeId = dr.GetInt32(8);
ZhuanYeName = dr.GetString(9);
MasterName = dr.GetString(10);
Term = dr.GetByte(11);
ruXue = dr.GetInt16(12);
Password = dr.GetString(13);
ClassId = dr.GetInt32(14);
LoginLevel = dr.GetByte(15);
ClassMaster = dr.GetString(16);
//copy finished! close dr and conn
dr.Close();
cmd.Dispose();
conn.Close();
conn.Dispose();
}
}
```

5.10 系统插件类

```
SystemPlugin.cs
/// <summary>
/// SystemPlugin 的摘要说明
/// 调用该 CLASS 只能是网站二级目录的文件
/// Version 0.01
/// 插件功能
/// </summary>
public class SystemPlugin
{
    const int MAXPLUGINCOUNT = 10; // The Max Files One Folder Can Read
    public SystemPlugin()
```

```
{
}
/// <summary>
/// Response The Files In Plugin Path (no web.config)
/// </summary>
/// <returns>String[]</returns>
public string[] ReturnAdminPluginNames(string pathName)
{
    string[] retVal = new string[MAXPLUGINCOUNT];
    int fileCount = 0;
    DirectoryInfo mainDir = new
DirectoryInfo(System.Web.HttpContext.Current.Server.MapPath("../Plugin
/") + pathName);
    FileInfo[] fInfo = mainDir.GetFiles("*.aspx");
    for (int i = 0; i < fInfo.Length; i++)
    {
        if (fileCount == MAXPLUGINCOUNT)
            break;//if count Max then break
        retVal[fileCount++] = fInfo[i].Name;
    }
    return retVal;
}
}
```

5.11 Asp. Net 全局操作文件

```
Global.asax
<%@ Application Language="C#" %>
<script RunAt="server">
    protected void Application_Start(object sender, EventArgs e)
    {
```

```

// 在应用程序启动时运行的代码
string AppName = string.Empty;
//检查变量，如果程序已经启动，无须再次启动。
if (Application["nov30th_isrun"] != null &&
string.IsNullOrEmpty(Application["nov30th_isrun"].ToString()))
    return;
ConnectDB Cnt = new ConnectDB();
if (Cnt.IsConn != true)
{
//数据库无法连接
Application["nov30th_isrun"] = "false";
Application["Err_Details"] = "Can Not Connect
DataBase!";
}
else
{
//读入数据到 App 中
if (Cnt.ExeReader("select
[OptName],[OptValue],[OptChar],[CorV] from [Nov30th_SysOpt]"))
{
while (Cnt.Reader.Read())
{
AppName = Cnt.Reader.GetString(0);
if (Cnt.Reader.GetBoolean(3) == false)
{
//数值
Application[AppName] =
Cnt.Reader.GetInt32(1);
//Response.Write("<br/>Key:"
Cnt.reader.GetString(0) + "
Value:"
Cnt.reader.GetInt32(1).ToString());
}
}
}
}

```

```

        else
        {
            Application[AppName] =
Cnt.Reader.GetString(2);
            //Response.Write("<br/>Key:" +
Cnt.reader.GetString(0) + " Value:" + Cnt.reader.GetString(2));
        }
    }
    Application["nov30th_isrun"] =
Convert.ToByte(Application["Running"]) == 1 ? "true" : "false";
    Application["Err_Details"] = "Started From Service";
}
}
Cnt.Dispose();
}
</script>

```

5.12 根目录 web.config 文件内容

```

web.config
<?xml version="1.0"?>
<!--
    注意：除了手动编辑此文件以外，您还可以使用
    Web 管理工具来配置应用程序的设置。可以使用 Visual Stud
    设置和注释的完整列表在
    machine.config.comments 中，该文件通常位于
    \Windows\Microsoft.Net\Framework\v2.x\Config 中
-->
<configuration
xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
    <configSections>
        <section name="magicAjax"

```

```
type="MagicAjax.Configuration.MagicAjaxSectionHandler, MagicAjax"/>
</configSections>
<magicAjax
outputCompareMode="HashCode"
tracing="false">
  <pageStore
    mode="NoStore"
    unloadStoredPage="false"
    cacheTimeout="5"
    maxConcurrentPages="5"
    maxPagesLimitAlert="false"
  />
</magicAjax>
<appSettings>
  <add key="KuangKeTips" value="008:旷课超过 8 节了, 可能会受到警告处分哦|014:旷课超过 14 节了, 再旷课就要受到严重警告处分了!|018:旷课超过 18 节, 奇迹了!"/>
  <add key="Copyright" value="Student"/>
  <add key="Version" value="V0.01"/>
  <add key="InstallKey" value="62e62e62e"/>
</appSettings>
<connectionStrings>
  <add name="StudentSetup" connectionString="Data Source=. \SQLEXPRESS;Integrated Security=True;User Instance=True" providerName="System.Data.SqlClient"/>
  <add name="StudentConnectionString" connectionString="Data Source=. \SQLEXPRESS;AttachDbFilename=|DataDirectory|Student.mdf;Integrated Security=True;User Instance=True" providerName="System.Data.SqlClient"/>
</connectionStrings>
<system.web>
<!--
```

设置 `compilation debug="true"` 将调试符号插入已编译的页面中。但由于这会
影响性能，因此只在开发过程中将此值
设置为 `true`。

-->

```
<roleManager enabled="true" />
```

```
<compilation debug="true"/>
```

<!--

通过 `<authentication>` 节可以配置 ASP.NET 使用的
安全身份验证模式，
以标识传入的用户。

-->

```
<authentication mode="Forms" />
```

<!--

如果在执行请求的过程中出现未处理的错误，
则通过 `<customErrors>` 节可以配置相应的处理步骤。具体说

来，

开发人员通过该节可以配置
要显示的 html 错误页
以代替错误堆栈跟踪。

-->

```
<customErrors mode="RemoteOnly" defaultRedirect="
```

```
Error/Error.html">
```

```
<error statusCode="403" redirect="Error/Error403.html" />
```

```
<error statusCode="404" redirect="Error/Error404.html" />
```

```
</customErrors>
```

```
<pages>
```

```
<controls>
```

```
<add namespace="MagicAjax.UI.Controls" assembly="MagicAjax"
tagPrefix="ajax"/>
```

```
</controls>
```

```
</pages>
```



```
<httpModules>
    <add name="MagicAjaxModule" type="MagicAjax.MagicAjaxModule,
MagicAjax"/>
</httpModules>
</system.web>

<!--
StudentFiles 目录拒绝所有访问
-->
<location path="StudentFiles">
    <system.web>
        <authorization>
            <deny users="*" />
        </authorization>
    </system.web>
</location>

<!--
TEMP 目录拒绝所有访问
-->
<location path="TEMP">
    <system.web>
        <authorization>
            <deny users="*" />
        </authorization>
    </system.web>
</location>
</configuration>
```

第六章 系统测试

本系统已经过测试通过，部分测试内容见第四章。

论文小结

.Net 是一个强大平台，开发 WEB 应用无需涉及 HTML 语言及格式，这与我几年前开发 ASP 已经完全不同。和 JAVA 语言相类似，可以实现类的实现，接口等。代码和 HTML 分离，组织清晰，事件驱动和编程模型，跨浏览器，跨设备。是将来发展的趋势。在开发过程中也学习到很多新的知识。HOHO``系统开发时间不长，在功能上和细节上还有待完善。

致谢

在这次毕业设计过程中，遇到许多问题。Google 及互联网的帮助是最大的，要感谢 Google 系统的开发人员。

文献综述

《 ASP.NET 快速入门教程 》

<http://chs.gotdotnet.com/quickstart/aspplus/doc/quickstart.aspx>

《 Visual C# 开发中心 》

<http://www.microsoft.com/china/msdn/developercenter/VCSsharp/default.aspx>