

# 软件设计文档

## 一． 技术选型

### 1. 操作系统——Linux 和 Windows

Linux：主要是用于部署的环境和开发环境，实际部署时使用的是 CentOS5.6，开发使用的是 Ubuntu14.04

Windows：主要用于开发，由于主要使用的语言 Python 是双平台的，因此系统可以在 Windows 和 Linux 进行开发

### 2. 数据库——MySQL

版本：MySQL5.1

类型：关系型数据库

引擎：MyISAM，因为主要的数据库操作都是查询和插入，且因是校内使用，对并发时一致性的要求不高

### 3. 后台框架——Django

版本：Django1.5.1

介绍：

Django 是一个开放源代码的 Web 应用框架，由 Python 写成。采用了 MVC 的软件设计模式，即模型 M，视图 V 和控制器 C。它最初是被开发来用于管理劳伦斯出版集团旗下的一些以新闻内容为主的网站的。并于 2005 年 7 月在 BSD 许可证下发布。这套框架是以比利时的吉普赛爵士吉他

手 Django Reinhardt 来命名的。

优点：

- 完善的官方文档，可以让新手迅速了解 Django 的各种特性和机制，开发一些逻辑较为简单的应用。

- 提供了大部分 web 需要的解决方案，如用户模块，后台管理模块，session 模块，cache 模块，表单验证模块等

- 配合 python 的正则匹配，可以设计出优雅的，可扩展性高的 URL 路由配置

- 自身提供的 ORM 可以让开发者方便地进行数据库操作，而无需开发者自身去考虑 SQL 的语法和一些防注入的细节

- 新版本的 Django 提供了方便的数据迁移操作，对于开发中对 model 层的修改，都可以方便地映射到数据库中

缺点：

- 系统紧耦合，很多模块之间关联性较强，比如使用用户模块的话就必须使用 Session 模块。这样就导致了系统的可扩展性很差，用户难以在使用 Django 部分模块的情况下开发自己定制模块

- Django 的 ORM 效率较差，且实现的功能较少，也不能很好地与 JSON 结合

- Template 功能太少,需要一些高级功能的话需要开发者自己编写

- 对非关系型数据库支持不好,需要开发者使用第三方的中间件,比如: PyMongo, 然后这样会导致自身一些其他模块无法使用(Admin 模块)

#### 4. 前端框架

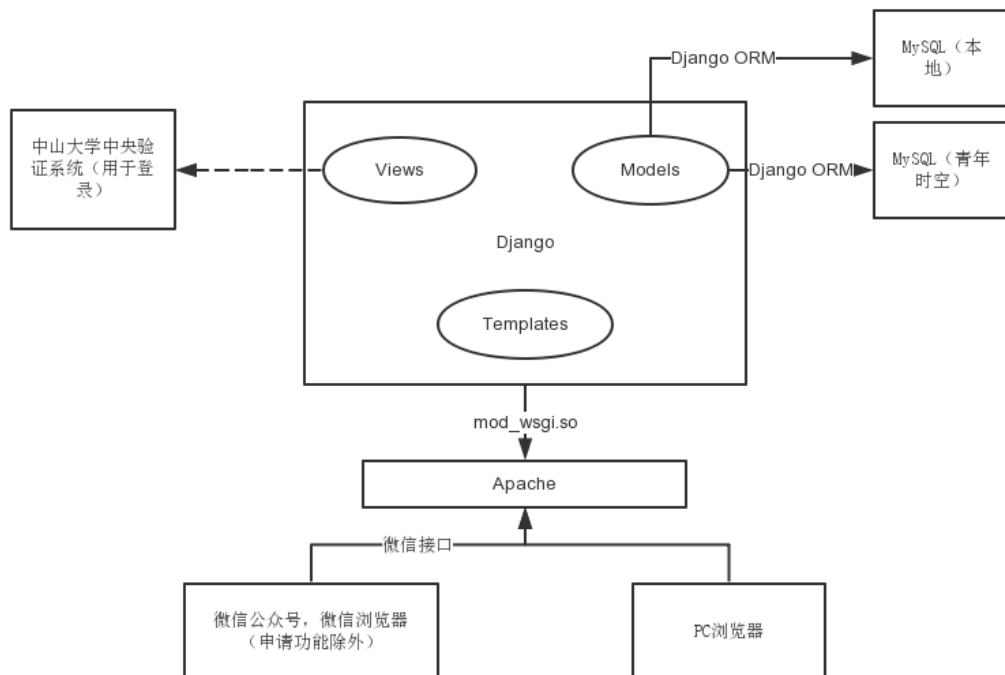
结构: 多页面结构

模板框架: Django 自带的 Template

UI 框架: Bootstrap, 比较常用的 UI 框架, 且拥有丰富的文档, 即便是新手也能快速上手

DOM 操作框架: jQuery, 同样拥有丰富的文档, 且调试方便

## 二 . 架构设计



### 三． 模块划分

活动模块(activity): 主要是处理有关活动的业务逻辑, 比如申请活动, 报名活动, 修改活动等

活动新闻模块(activityNews): 主要是通过对接青年时空的数据库来向用户的微信推送活动消息和供用户在三微站点查看青年时空的活动信息

基础模块(base): 主要是向其他模块的 template 提供基础模板

团委模块(committee): 主要是处理有关团委操作的各种业务逻辑

反馈模块(feedback): 主要是处理用户在微信中提交的维权反馈消息

负责人模块(principal): 主要是处理各个校区团委负责人的业务逻辑的模块

学生模块(student): 主要是处理有关学生的各种业务逻辑的操作的模块

教师模块(teacher): 主要是处理有关团委老师各种业务逻辑的模块

### 四． 使用的软件设计技术

#### 1. Object Relational Mapping

对象关系映射, 用于实现面向对象编程语言里不同类型系统和数据之间的转换

使用位置: 所有模块底下的 models.py 文件, 例子:

```

class Submission(models.Model):
    description = models.CharField(max_length = 32, null = True, blank = True)
    file = models.FileField(upload_to = 'files', null = True, blank = True)
    type = models.CharField(max_length = 16, choices = (
        ('classroom', u'课室申请'),
        ('chair', u'桌椅申请'),
        ('poster', u'张贴申请'),
        ('location', u'校区场地申请'),
        ('actionCenter', u'学生活动中心申请'),
        ('functionRoom', u'校团委功能室申请'),
        ('car', u'大型车辆进出申请'),
    ))
    startTime = models.DateTimeField(null = True, blank = True)
    endTime = models.DateTimeField(null = True, blank = True)
    location = models.CharField(max_length = 64,
        null = True, blank = True)
    locations = models.ManyToManyField("Location", related_name = "submission")
    number = models.IntegerField(default = 0)
    chairNumber = models.IntegerField(default = 0)
    tentNumber = models.IntegerField(default = 0)
    otherNumber = models.CharField(max_length = 256, null = True, blank = True)
    activity = models.ForeignKey(Activity)
    submissionMark = models.BooleanField(default = True)
    startTimeMark = models.BooleanField(default = True)
    endTimeMark = models.BooleanField(default = True)

```

## 2. Web Crawler

通过网络爬虫技术向本校区的中央验证系统发出请求来验证学生的身份信息

使用位置：sysuer/sysunetid.py

```

# -*- coding:UTF-8 -*-
import urllib
import urllib2
import cookielib
import re

def sysuNetid(username, password):
    cj = cookielib.MozillaCookieJar()
    cookieSupport = urllib2.HTTPCookieProcessor(cj)
    opener = urllib2.build_opener(cookieSupport, urllib2.HTTPHandler)
    urllib2.install_opener(opener)
    url = 'https://sso.sysu.edu.cn/cas/login'
    text = urllib2.urlopen(url) # the first open, I got cookie, It and execution from this
    cookie = ""
    # print text.read()
    try:
        cookie = repr(cj)
        patOfCk = re.compile("value=(.*?)",)
        cookie = 'JSESSIONID=' + patOfCk.search(cookie).group(1)
        # cookie has been got by re
    except:
        pass
    dataDic = getData(text)
    headers = {
        'Host': 'sso.sysu.edu.cn',
        'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:30.0) Gecko/20100101 Firefox/30.0',
        'Referer': 'https://sso.sysu.edu.cn/cas/login',
        'Content-Type': 'application/x-www-form-urlencoded',
        'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
        'Accept-Language': 'zh-cn,zh;q=0.8,en-us;q=0.5,en;q=0.3',
        'Connection': 'keep-alive',
        'Cookie': cookie,
        'Cache-Control': 'max-age=0'
    }
    data = {
        'username': username, # use you netid
        'password': password, # use you password
        'lt': dataDic['lt'],
        'execution': dataDic['execution'],
        '_eventId': dataDic['eventId'],
        'code': "",
        'submit': '登录'
    }

```

```

        'submit': '登录'
    }
    urlData = urllib.urlencode(data) # the second open, I post the data(include cookie) which were got from the first time
    try:
        req = urllib2.Request(url = url, data = urlData, headers = headers)
        res = urllib2.urlopen(req)
        text = res.read()
    except:
        return ""
    if judge(text):
        try:
            req = urllib2.Request('http://my.sysu.edu.cn/mail2')
            req.add_header('User-Agent', 'Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:30.0) Gecko/20100101 Firefox/30.0')
            res = urllib2.urlopen(req)
            t = res.read()
            if len(t) < 2:
                return username
            else:
                return getName(t)
        except:
            return 'outdate'
    else:
        return ""

def getData(text):
    flag = 0
    lt = ""
    execution = ""
    code = ""
    submit = ""
    patOfLt = re.compile("name=lt. value=(.*?)...>")
    patOfExecution = re.compile("name=execution. value=(.*?)...>")
    patOfEventId = re.compile("name=_eventId. value=(.*?)...>")
    for line in text:
        # get lt
        checkLt = patOfLt.search(line)
        if checkLt:
            flag += 1
            lt = checkLt.group(1)
        # get execution

```

```

# get execution
checkExecution = patOfExecution.search(line)
if checkExecution:
    flag += 1
    execution = checkExecution.group(1)
# get _eventId
checkEventId = patOfEventId.search(line)
if checkEventId:
    flag += 1
    eventId = checkEventId.group(1)
# jump judge
if flag == 3:
    break
dic = {}
dic['It'] = It # It has been got by re
dic['execution'] = execution # execution has been got by re
dic['eventId'] = eventId # eventId has been got by re
return dic

def getName(text):
    pat = re.compile('data-controlPanelCategory="my" > 
    Alias /static/ "/www/web/ThreeMicro/Code/static/"
    Alias /media/ "/www/web/ThreeMicro/Code/media/"
    DocumentRoot "/www/web/ThreeMicro/Code/"
    ServerName swgc.sysu.edu.cn

    WSGIScriptAlias / "/www/web/ThreeMicro/Code/django.wsgi"

    <Directory "/www/web/ThreeMicro/Code/">
        Allow from all
    </Directory>
</VirtualHost>

```

使用 mod\_wsgi 来链接 Django 和 Apache，配置文件如下：

```
import os, sys
apache_configuration = os.path.dirname(__file__)
project = os.path.dirname(apache_configuration)
workspace = os.path.dirname(project)

sys.stdout = sys.stderr
sys.path.append(workspace + '/ThreeMicro/Code/')

os.environ['DJANGO_SETTINGS_MODULE'] = 'settings'
import django.core.handlers.wsgi
application = django.core.handlers.wsgi.WSGIHandler()
```

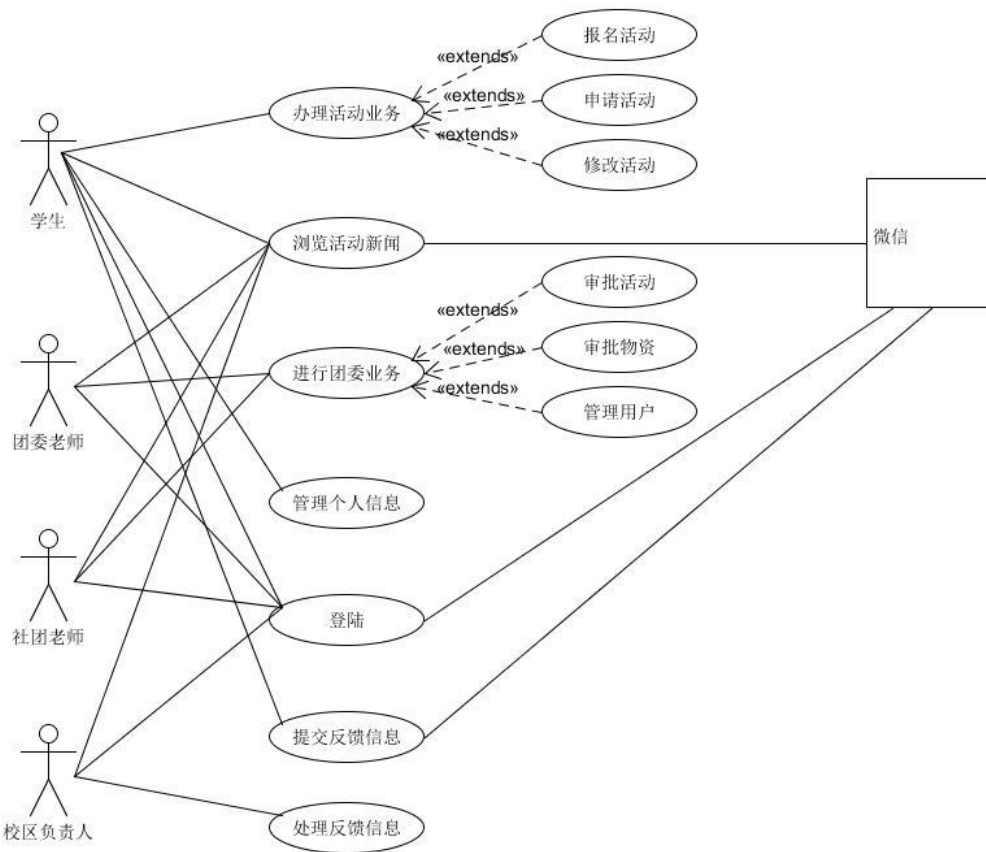
#### 4. 响应式 web 开发

通过检测客户端的类型来用不同的前端页面去相应请求

```
template = 'login.html'
USER_AGENT = request.META['HTTP_USER_AGENT']
phoneList = ['MicroMessenger', 'iPhone', 'Android']
for phone in phoneList:
    if phone in USER_AGENT:
        template = 'microLogin.html'
```

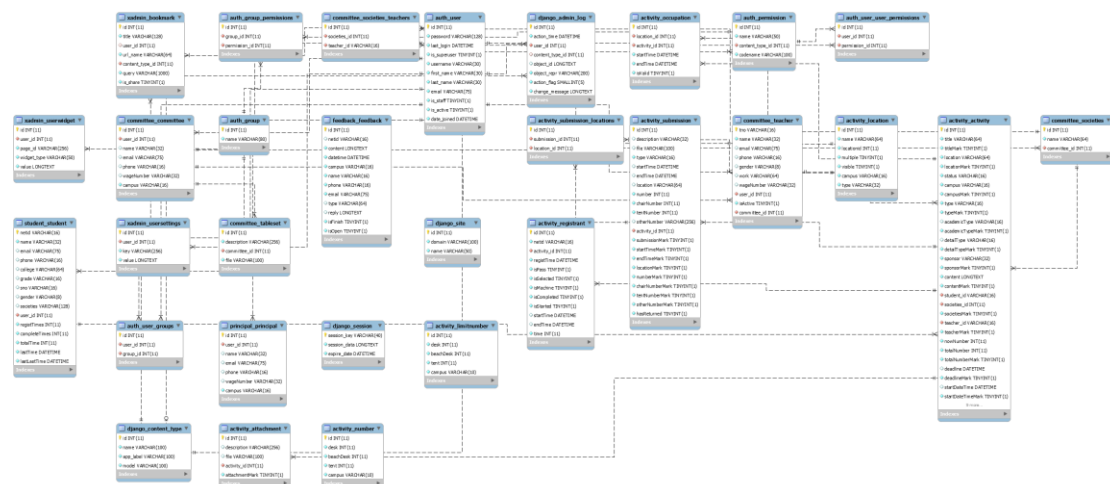


## 五． 系统用例图



## 六. 数据模型

ER 图:



详细请见相同目录下的 pdf