项目设计文档

——微信公众平台“清华紫荆之声”

Version 1.0

编写者

Prepared by

团队：**扬帆启程**

Group Name: Sailon

陈华榕 **Huarong Chen**	**2011013236**	**chenhuarongzp@gmail.com**
文庆福 **Qingfu Wen**	**2011013239**	**thssvince@163.com**
庄晨帆 **Chenfan Zhuang**	**2011013246**	**zhuangchenfan@gmail.com**
杨　磊 **Lei Yang**	**2011013256**	**yl93528@gmail.com**
林维妮 **Wheini Lin**	**2013400888**	**winnie180690@gmx.de**
刘　谦 **Qian Liu**	**2011013242**	**solidsnakekill@gmail.com**

Instructor: 刘强，刘璘

Qiang Liu, Lin Liu

Course: 软件工程（3）

Software Engineering (3)

Teaching Assistant: 龚云飞，王得希

Yunfei Gong, Dexi Wang

Date: **2013/12/28**

目录

文档修订

版本号 Version	主要作者 Primary Author(s)	简述 Description of Version	完成时间 Date Completed
1.0	杨磊、文庆福、陈华榕	对项目整体设计进行分析，本文档撰写完成。	2013/12/28

1 简述

本项目是“清华紫荆之声”公众平台的一部分，是该公众平台的访问入口，集成了活动票务管理的一系列功能与公众平台的公共类服务（如学号绑定、帮助等）以及公众平台其余功能的接入。

2 开发规范

2.1 开发人员

开发人员	负责工作
陈华榕	票务管理系统，后台检票系统，二维码生成系统。
庄晨帆	票务管理系统，页面票据打印。
文庆福	微信交互设计，二维码生成系统。
杨磊	微信交互页面，票务管理系统。

2.2 需求准备

2.2.1 需求与变动

2.2.1.1 功能性需求

功能性需求	说明
用户认证	用户通过“清华紫荆之声”微信公众平台给定的链接，用清华大学信息门户的账号完成用户认证。
活动查询	用户查看当前可抢票的活动，通过给定链接可查看活动详情。
票务查询	用户查看已抢到的电子票，通过给定链接可查看电子票详情。
活动抢票	用户对校团委近期发起的特定活动进行抢票。如果抢票成功，用户获得该活动的电子票；如果抢票失败，用户获得抢票失败的提示信息。
活动检票	票务管理系统可以检录有效电子票 一张电子票只能被检录一次。
二维码生成	平台可以生成有效、不重复的二维码作为电子票。

2.2.1.2 非功能性需求

非功能性需求	说明
使用标准	符合微信公众平台服务号的相关标准。
系统需求	● 平台应该和配套开发的票务管理系统交互。 ● 微信页面能在各种设备上正常显示。
性能需求	● 活动抢票服务应该支持在任何时刻接受较高并发请求。 ● 平台任何一次收到用户消息时应该在 3s 内完成所有数据处理。 ● 平台应该在 5s 内回复用户消息。 ● 活动检票系统可以快速扫描用户的电子票。

2.2.1.3 需求变动

需求变动	说明
活动查询	活动查询更改为查询校团委发布的可抢票活动。
“尹福”知道	相关功能已经被 Lab.mu 学生兴趣团队实现了，避免重复性，该功能在现有的“清华紫荆之声”平台中取消。

2.2.2 涉众和用户说明

2.2.2.1 涉众概要

涉众	描述	职责
Sailon 开发团队	开发“清华紫荆之声”公众平台	对“清华紫荆之声”公众平台进行设计、架构、编码、测试、部署以及后期维护等。
校团委新媒体组	“清华紫荆之声”的主要管理人员	是“清华紫荆之声”平台的需求方，平台完成后，运营平台。
在校师生	平台的用户	关注“清华紫荆之声”，并使用其功能。

2.2.2.2 用户环境

校团委新媒体组的运营人员通过浏览器进入微信公众平台管理页面可以对在线平台进行运营管理，进入后台管理页面对数据库的部分数据进行管理；

师生可以通过 IOS、Android 等手持设备，登录微信，关注“清华紫荆之声”，即可享受平台所提供的服务。

2.3 开发记录

时间节点	项目进度
2013 年 12 月 4 日（第一次迭代完成）	微信 query 处理、微信页面、后台管理页面、检票系统。
2013 年 12 月 25 日（第二次迭代完成）	微信菜单调整、服务器配置、微信页面优化、后台页面完善。

2.4 开发环境与工具

2.4.1 开发环境

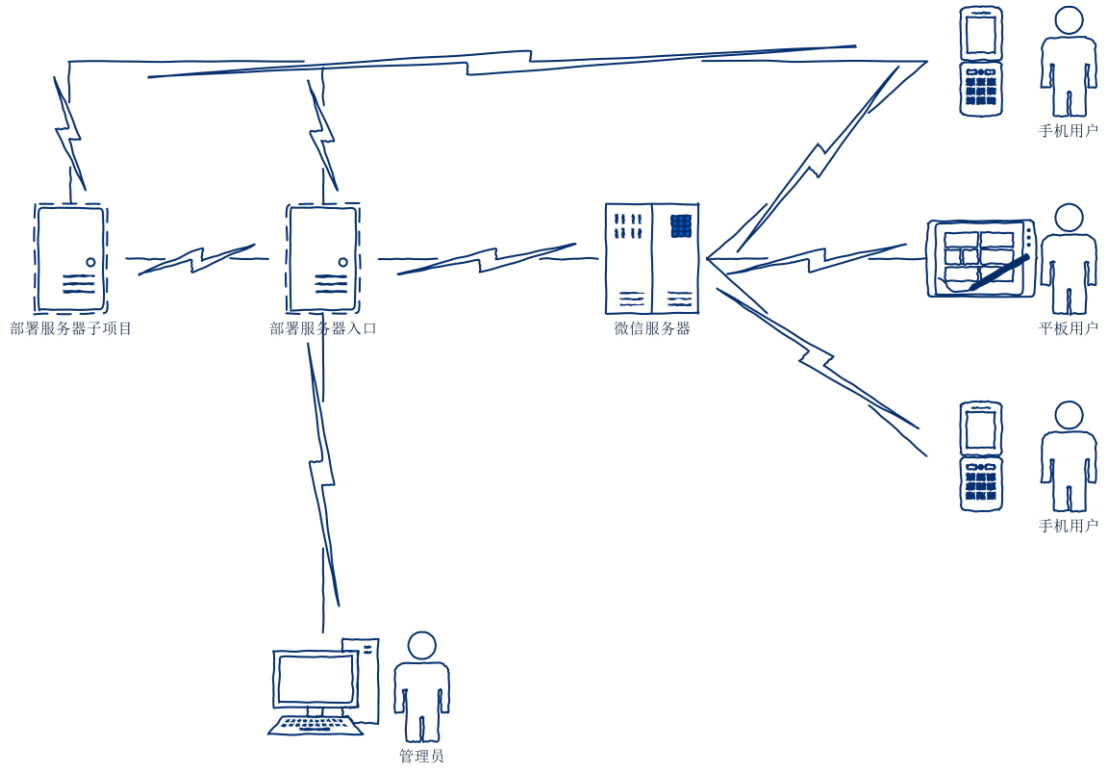
开发语言	Python 2.7 以上（含），3.0 以下（不含）版本。
开发框架	Django 1.6.1 及以上版本。
数据库环境	MySQL 5.6.14 及以上版本。
开发服务器	BAE 3.0。目前只支持 1.4.2 的 django 版本，且不够稳定。
部署服务器	使用了软件学院学生科协的服务器作为部署服务器。其操作系统为 Ubuntu Server 12.04。
微信服务器	微信在 5 秒内收不到响应会断掉连接，access_token 的有效期为 7200 秒，获取凭证接口以及自定义菜单接口每天都有调用次数限制。

2.4.2 开发工具

PyCharm	编译、运行、调试 python 程序。
Git	分布式的代码版本控制软件。
SourceTree	Windows 平台和 Mac 平台上的 Git GUI 工具。
Google Chrome 32.0	移动端（手机、平板）上页面显示调试。

3 系统设计

3.1 系统部署结构



用户通过手机、平板等设备访问微信服务器，微信服务器将消息转发给部署服务器，部署服务器根据需要可能再次转发给其本机的子项目，接着将处理结果返回部署服务器入口，进而返回给微信服务器，微信服务器再帮我们返回给用户设备。

用户设备也可能直接访问部署服务器：用户通过回应的消息点击相应链接，将打开相应页面，对应正是直接访问部署服务器。

管理员通过计算机直接访问部署服务器，进行信息查询与维护。

3.2 交互设计

3.2.1 后台管理

活动列表查看

活动订票、入场情况查看，座位分布选择

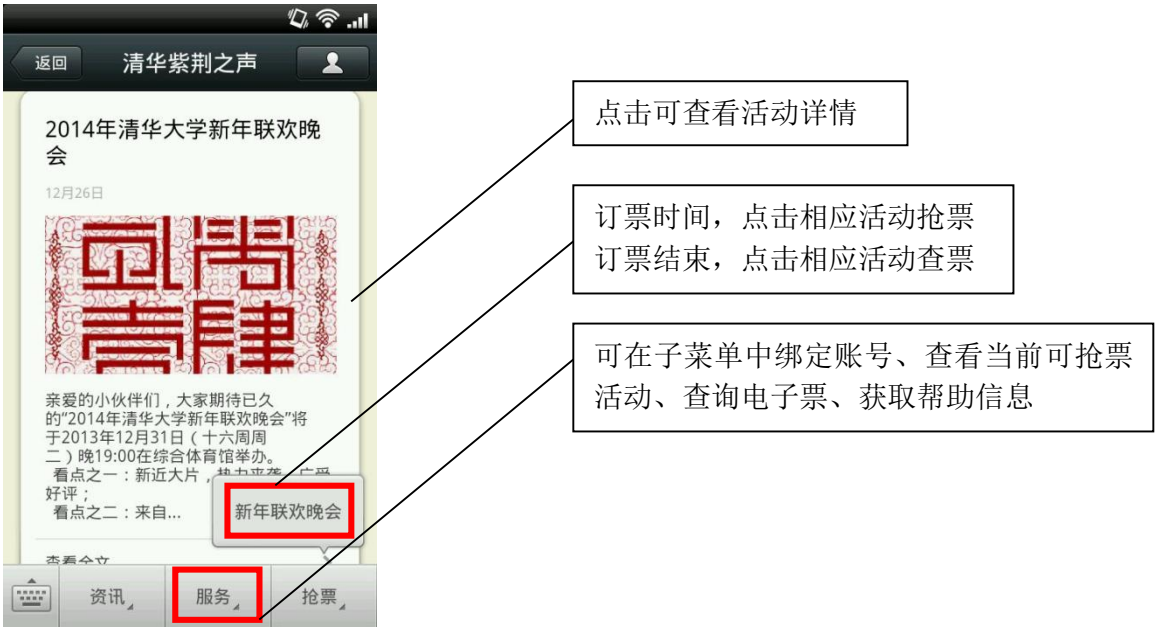
设置微信抢票菜单

活动暂存、发布、修改

3.2.2 活动检票

利用扫描枪扫描二维码读出电子票，也可以直接输入绑定的学号进行验证。

3.2.3 微信交互



3.3 模块设计

3.3.1 项目入口

为减小微信处理模块对 Django 的 URL 托管的依赖，设计了项目入口。如此，微信处理模块仅依赖 Django 的 Model 模块。

项目入口要做的正是判断请求的 URL，若是 LUCKY_URL（项目中设定为/weixin），交由微信处理模块处理，否则托管给 Django 处理。

3.3.2 通用微信模块

该模块对应为项目中的 Python 包 weixinlib。

目前“清华紫荆之声”仅获得了自定义菜单的微信接口，因此在 weixinlib 中只实现了该接口的调用。

此外，还封装了一些函数，比如检查某个抢票活动是否适合加入菜单（通过判断其时间）的 custom_menu.check_if_activity_out 函数等。

微信的一些 URL 设置在 weixin_urls.py 和 url_generators.py 中。向指定 URL 发送 GET 或 POST 请求的函数封装在__init__.py 中。默认的菜单与一些菜单设置在 settings.py 中。

3.3.3 微信处理模块

该模块对应为项目中的 Python 包 queryhandler。

3.3.3.1 基础封装

对微信消息结构、消息模板等进行一系列封装，以此提高代码质量，降低维护成本。细致到文件的封装情况如下：

文件	封装内容
weixin_msg.py	方便获取微信传输数据中的相应信息，减小出错可能。
weixin_reply_templates.py	微信回复内容的模板。
weixin_text_templates.py	本项目回复文本内容的模板。
handler_check_templates.py	消息检验函数（详见后文）的判断模板。

3.3.3.2 票务处理

集中于 tickethandler.py 中，是__init__.py 中定义的一系列 check-response 函数的实现。

3.3.3.3 消息转发

在 query_transfer.py 中，实现了将消息通过 HTTP 请求转发给处理资讯的项目。若在其他项目中实现了本公众平台的新功能，也可通过类似的方式进行请求转发。

3.3.4 后台管理模块

该模块对应为项目中的 Python 包 urlhandler/adminpage。

该模块遵循 Django 库的规范，按 MVT 架构方式，使用 urlhandler/urlhandler.models 中定义的 Model，在 urlhandler/adminpage.views 中定义 View（起到 MVC 中的 Controller 的作用），在 urlhandler/adminpage/templates 中定义 Template（可直接渲染的模板，起到 MVC 中的 View 的作用）。

该模块的页面 URL 配置定义在 urlhandler/adminpage.urls 中。

该模块包含的页面、功能等包括：

功能	View 函数	URL 配置
登录页面与相关处理	home, login, logout	/, /login/, /logout/
活动列表	activity_list	/list/
活动的增删改查	activity_add, activity_delete, activity_post, activity_detail	/add/, /delete/, /modify/, /detail/<actid>/
微信菜单获取、查改	custom_menu_get, adjust_menu_view, custom_menu_modify_post	/menu/get/, /menu/adjust/, /menu/submit/
检票	activity_checkin, activity_checkin_post	/checkin/<actid>/, /checkin/<actid>/check/

电子票打印的登录页面、登录功能、登出、列表、打印	order_index, order_login, order_logout, order_list, print_ticket	/order/, /order_login/, /order_logout/, /order_list/, /print/<unique_id>/

此外还通过 urlhandler/adminpage.safe_reverse 定义了一系列 URL 反向翻译，实际是 Django 反向翻译函数的封装，以此降低维护成本。

3.3.5 微信页面模块

该模块对应为项目中的 Python 包 urlhandler/userpage。

该模块同样遵循 Django 库的规范。

该模块的页面 URL 配置定义在 urlhandler/userpage.urls 中。

该模块包含的页面、功能等包括：

功能	View 函数	URL 配置（均有前缀/u）
学号绑定页面与功能	validate_view, validate_post	/validate/<openid>/, /validate/try/
活动详情	details_view	/activity/<activityid>/
电子票	ticket_view	/ticket/<uid>/
用户指南	help_view, helpact_view, helpclub_view, helplecture_view	/help/, /helpact/, /helpclub/, /helplecture/
电子节目单	activity_menu_view	/activity/<actid>/menu/

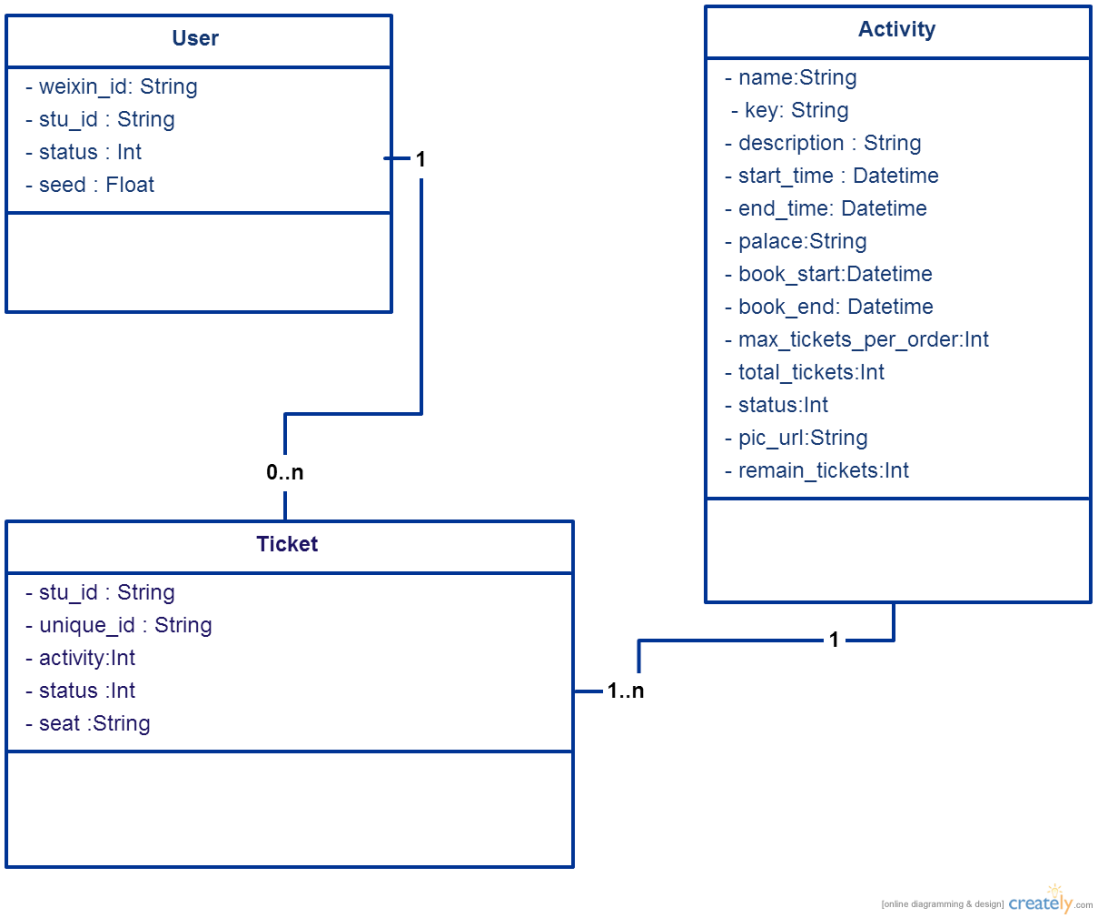
同样定义了 URL 反向翻译：urlhandler/userpage.safe_reverse。

3.3.6 二维码子系统

这是一个与主项目无关的子系统，主项目 queryhandler.settings 中 QRCODE_URL 正是部署的该子系统域名。

该子系统基于 Django 库，配置两条 URL：/<qrmsg>/，/fit/<qrmsg>/。前者是适用于微信图文链接显示的二维码图片（因此增加了左右的空白像素），后者是生成的纯二维码图片（也就是没有左右的空白像素）。

3.4 数据结构



3.5 接口规范

3.5.1 外部接口

3.5.1.1 二维码接口

采用 QRCODE_URL/<qrmsg>/作为返回给用户的图文链接的电子票图片，QRCODE_URL/fit/<qrmsg>/作为用户电子票页面中的电子票图片。

QRCODE_URL 定义在 queryhandler.settings 中。

3.5.1.2 资讯接口

在 queryhandler.query_transfer 中定义，接口函数为 get_information_response，参数是微信服务器发来的 POST 原始数据。在该函数中通过 HTTP 请求向负责资讯的项目提交，再将提交结果返回即可。

提交的 HTTP 请求 URL 为 queryhandler.settings.INFORMATION_SITE_DOMAIN。

3.5.1.3 WSGI 接口

本项目提供 WSGI 接口以进行高效部署。WSGI 模块为 localwsgi.py，WSGI 函数入口为 app.py 中的 app 函数。

3.5.2 内部接口

3.5.2.1 check-response 规范

这是本项目中大大提高可维护性的漂亮实践，受影响的是微信处理的模块。

我们规范本项目中的所有微信处理均实现两个函数：check 与 response，它们的参数均是一个解包微信 xml 数据后得到的 dict 对象。其中 check 函数返回 True 或 False，判断是否接受该消息，若返回 True 说明接受，就交给 response 函数处理。response 函数返回的是经 xml 封装的回复数据。

有了这样的规范，就能独立地去实现各个功能函数，互不干扰，而且因为 check 函数往往只需要进行简单的判断，因此效率也能保证。

现在项目中收到微信处理请求后，将原始数据进行 xml 解包得到 dict 对象，就按顺序枚举所有 check 函数（聚合在一个 list 中），当有 check 函数接受该数据，就返回对应 response 函数的处理结果。如果没有任何 check 函数接受数据，就转发给负责资讯的项目。

如此，未来无论是新增功能还是维护现有功能，都只需要增加 check-response 函数，并调整其在 list 中的位置即可。

3.5.2.2 文本模板

为减小维护成本，系统中的微信处理模块封装了所有的文本模板。主要分为返回消息的模板和回复内容的文本模板两类。

这两类模板分别实现在 queryhandler.weixin_reply_templates.py 和 queryhandler.weixin_text_templates.py 中。

3.5.2.3 微信接口

接口函数	调用说明	调用微信官方接口
weixinlib.base_support check_weixin_signature	提供微信接入的参数验证，返回验证结果。	接入指南 验证 URL 有效性
weixinlib.base_support get_access_token	无参数调用即可，返回 access_token。	基础支持 获取 access_token
weixinlib.custom_menu get_custom_menu	无参数调用，以 dict 返回当前的微信自定义菜单。	自定义菜单 自定义菜单查询接口
weixinlib.custom_menu modify_custom_menu	提供新抢票菜单的 dict，返回微信处理结果。	自定义菜单 自定义菜单创建接口
weixinlib.custom_menu add_new_custom_menu	提供希望增加菜单的 name 与 key，添加至抢票菜单中，返回微信处理结果。	自定义菜单 自定义菜单创建接口

4 实现细节

4.1 目录结构

Tsinghuatuan	configurations	init	系统Job配置文件
		nginx	nginx配置文件
	docs		项目相关文档
	queryhandler		微信处理模块
	urlhandler	adminpage	后台管理模块
		urlhandler	Django入口与配置、数据库
		userpage	微信页面模块
	weixinlib		通用微信模块
QRservice	qrhandler	qrhandler	二维码Django入口与配置

4.2 后台管理

后台管理页面主要包括登陆、活动列表、活动详情、调整微信菜单和检票页面。每个页面都对应一个 View 函数。各页面在被访问时都会检查用户是否登录，如用户未登录则跳回登录界面。访问活动列表、活动详情时处理页面请求；修改活动信息时处理 POST 请求；检票页面和微信菜单调整页面分别在 4.5 和 4.7 中阐述。

在 safe_reverse.py 中，用 reverse 函数反解析 url 以直接访问其它视图，当访问路径需要更改时，只需要更改此文件。

活动列表页面根据后台传入的数据，动态生成文件列表，根据活动的不同状态，判断当前活动的状态、活动是否能够检票，是否能够被删除等。

活动详情页面根据当前时间判断哪些选项不可修改，新增活动时若缺少所需字段，会在发布时给出提示；新增活动后，在详情页点击发布可以修改活动信息；通过重置可以将活动信息还原回修改前的状态。

4.3 微信交互

微信交互主要由 tickethandler.py 实现，处理从微信服务器 POST 来的请求，接收到这个请求后，需要解析微信服务器发送来得 XML 包，对于其不同的请求类型和不同的请求内容做相应的处理。具体而言，在_init_.py 中定义了处理函数和响应函数的关系表，每个请求用对应的方式处理。

目前只支持文本消息和点击事件，不支持如图片等其他形式的消息请求。处理完毕之后，用 HttpResponse 将信息包装成 XML 格式发送给微信服务器，可以发送文字消息，图片消息和图文消息等。

对于“绑定”的消息，通过表单提交来模拟网络学堂登陆，以此验证校内用户的合法性，验证合法性后用户即可抢票，数据库中不保存用户的账号密码。

对于“取票”、“查票”、“抢啥”的消息只读取数据库；对于 “退票”、“解绑”的消息需要更新数据库；对于抢票开始前的“抢票”消息，根据微信服务器的时间，返回当前距抢票开始的时间；对于抢票期间的“抢票”消息，需要保证数据库操作的原子性，具体在 4.6 中说明。

微信前端页面采用比例布局以适应不同设备的显示，活动详情页面只在访问时向服务器请求一次数据，倒计时用 js 的计数器实现。

4.4 二维码系统

二维码系统只实现了一个 View 函数：get_qr_code，附带参数为 qrmsg 和 qrtype，通过 URL 配置指定 qrtype 为 fit 或 wide，函数中根据该参数生成期望的二维码数据并返回。

生成二维码通过第三方库，实现起来不难。使用的二维码配置是版本 1，低容错。

4.5 检票系统

后台实现比较容易，对应两个 View 函数，一个处理页面请求，一个处理检票的 POST 请求。

前端实现设计为全屏抢票模式，当浏览器处于全屏状态则可进行检票，而处于非全屏状态则进行相关提示。票据信息输入框为一个始终拥有焦点的 input，可通过录入票据信息或学号两种方式进行检票，结果包括检票成功、票据已使用、无效票等多种情况，在检票界面都有相关提示。

现场录入电子票通过二维码扫描枪实现。由于扫描枪可视为键盘录入设备，因此将检票系统通过网页进行实现，使其直接具备跨平台能力。

4.6 并发原子锁

并发原子锁主要在抢票中使用，用 django 中的事务（transaction）和 F 函数来保证并发时更新数据库的原子性。

F 函数可以保证从数据库最新的数据开始更新，而不是之前读出的数据，且该数据库操作为原子操作。若访问数据库时发现当前更新数据库条件不满足（如：余票数为 0），则完全回滚至事务开始前的状态。

4.7 更新抢票菜单

在 weixinlib.custom_menu 中实现，前文已描述其中的通用微信接口，这里关注于微信抢票菜单的自动更新。

这部分实现包括：添加/修改活动后自动更新菜单项、手动调整菜单项及其顺序、自动清除无效菜单项。

在 weixinlib.custom_menu 中实现了判断活动是否过时（即不需要显示在菜单中了）、清除所有过时活动等功能的函数。

添加/删除活动后自动更新菜单项的同时也会调用这些函数来移除过时活动。

手动调整菜单项顺序通过前端页面进行菜单增删与顺序调整，提交更改后调用相应后台处理函数。

自动清除无效菜单则是先获取当前菜单，然后清除其中过时活动，再提交给微信服务器，做到自动清除是通过 Linux 服务器的 crontab 计划任务设定每 3 小时运行一次脚本实现的，相应脚本是 tsinghuatuan_crons.py。

5 部署细节

5.1 部署环境

抢票功能对服务器性能有较高要求，因此建议部署在安装了服务器操作系统的硬件设备上。这部分以 Ubuntu server 12.04 为例。

5.2 部署准备

5.2.1 系统依赖

首先需要安装好所需的 Ubuntu server，接着为其安装 python、mysql、nginx 等软件包。接着安装 python-pip，完成后用其为 python 全局安装 virtualenv 以创建虚拟 python 环境来保持不同项目间 python 库的纯洁性。

安装好后通过 virtualenv 创建虚拟环境，在虚拟环境中安装 uwsgi，再装载代码文件，并通过 pip 安装所有第三方依赖。

5.2.2 并发优化

Linux 服务器默认的请求处理队列长度为 128，每个请求都需要一些耗时，因此这样的默认配置容易在超过 100 的并发请求时直接抛弃。

所幸该长度是可以设置的，一般 2048 已经足够满足我们对并发的需求了。通过在终端中运行 sudo sysctl -w net.core.somaxconn=2048 命令可以进行临时设置（重启后失效），也可通过在/etc/sysctl.conf 中增加 net.core.somaxconn=2048 来进行永久设置。

5.3 部署规范

5.3.1 部署数据库

启动 mysql 服务，创建相应数据库，再修改 urlhandler/urlhandler.settings 中的数据库配置，执行 urlhandler/manage.py syncdb 自动创建数据库表，若数据库表成功创建，说明数据库已部署正确。

5.3.2 部署 uwsgi 与 Job

可修改 configurations/init/tsinghuatuan.conf 为部署服务器的路径配置，其中已做好了 uwsgi 的启动命令，可根据需要做一些微调。

修改完毕后，将该文件复制到/etc/init 目录下，就能通过 sudo start tsinghuatuan 来启动本项目的 uwsgi 服务了。

5.3.3 部署 nginx

在 configurations/nginx/tsinghuatuan.conf 中已有本项目的 nginx 的配置，根据需要做一些微调后将其复制到 nginx 配置文件目录下，再执行 nginx -s reload 即可。

本项目的配置文件中设定的 nginx 与 uwsgi 的数据通道为 pipe 文件。

5.3.4 部署 crontab

通过 crontab -e 即可修改当前系统的计划任务，增加执行 tsinghuatuan_crons.py 的命令并设定合适的执行计划即可。

6 附录

6.1 第三方组件

库名称	版本	说明
Django	1.6.1	强大的 Python 开源 Web 应用框架。
MySQL-Python	1.2.4	MySQL 库，方便进行数据库操作。
QRcode	4.0.4	Quick Response Code，矩阵二维码符号。 QRcode 是 Python 中生成二维码的模块。
PIL	1.1.6	Python Imaging Library，Python 下的图像处理模块，支持多种格式，并提供强大的图形与图像处理功能。

6.2 更新思路

1. 进一步改进服务器端，提高并发数量，希望能支持到 1000 个并发请求。
2. 后台添加活动时，增加图片上传功能。
3. 电子票可以打印，完善目前的打印系统。
4. 完善 django-admin，便于查看每次活动记录相应的数据，如：参与抢票人数，抢到某次活动的票人员列表。
5. 添加可定制化座位系统，如新清华学堂，大礼堂，综体各对应一套座位体系。