



快乐学堂交付文档

—— 致力于提供便利的学堂助手

曹智林 刘立 江昊琛 孙安临

快乐学堂交付文档

—— 致力于提供便利的学堂助手

产品交付情况

产品二维码：



服务器 IP：

123.47.59.249

GITHUB 版本库地址：

<https://github.com/jianghc724/HappyXueTang>

文档提要

1. 产品交付
2. 产品目标
3. 开发组织管理
4. 系统设计
5. 重点难点及解决
6. 测试总结
7. 系统部署

产品目标：

系统功能：

产品专注于学生用户所需求的功能，该系统支持功能的详细情况如下：

1.课程表展示：通过课程表展示用户所选课程，可选择周次、星期来查看当天课程。点击课程可以进入课程详情。

2.课程详情展示：展示课程评分、课程公告、作业的课程信息。可由课程详情页面进入课程评价页面。

3.课程 DDL 展示：页面展示未交的未过期作业，并用进度条显示剩余时间。

4.课程评价：用户可对已选课程进行评价，包括评分与文字评价。这部分数据将展示在搜课之后的课程结果中，为搜课的用户提供参考信息。

5.课上弹幕：用户向公众号发送弹幕消息，弹幕内容可显示在相应的弹幕展示页面。

6.课程搜索：用户可通过关键字搜索目标课程查看评分。

7.自习室使用与预约：提供自习室实时使用信息，并与学校的文图自习室、校内研讨间的预约平台练级，用户可在公众号内进行预约。

8.新动态提示：按时发消息提醒用户当前未读公告数目与待交作业数目。

9.用户绑定/解绑：用户可绑定校园账号，公众号其他功能需要绑定校园账号之后才能使用。

性能及用户体验要求：

主观要求：

用户在使用“快乐学堂”公众号的时候，应当不能感觉到明显的卡顿和较长等待时间，要求相应速度不能慢于旧版网络学堂，同时远远快于新版网络学堂。

用户在选择公众号提供的功能时，应该得到足够的新手指导而不至于困惑，同时菜单分类清晰，便于用户查找自己想要使用的功能。

服务器应当保护用户的校园网账号密码的安全性，避免重要信息泄露。

界面应保持一定的美观性，使用起来令人赏心悦目。

客观要求：

项目应该能够支持 300 个及以上的并发请求不出现差错（由于学堂助手的特性，并没有支持超高并发率的需要，因此标准较低）；

对于弹幕功能等要求并发程度较高的功能来说，我们要求服务器能够同时处理 1000 并发的弹幕请求，以达到实时而准确的弹幕显示效果。

50%及以上的用户在体验过产品之后会考虑使用该公众号。

该公众号能吸引到到 100 个以上的常驻用户。

过程管理：

我们在开发过程中使用 github 作为我们的版本管理以及团队协作工具。同时也使用了 tower.im 网站记录了我们的部分开发记录和任务分配情况。同时我们使用了庄天翼组提供的 api 接口网站及 gitbook 文档帮助我们的开发。

人员分工：

由于小组内有两个男生和两个女生，我们将前端部分和后端部分的任务均分到两个小组。

曹智林和刘立同学主要负责大部分前端界面的设计和功能实现，以及功能测试部分。

江昊琛和孙安临同学主要负责后端 api 以及数据库维护的工作以及它们对应的单元测试工作，其中前者负责较多 api 工作，因此后者也兼负弹幕功能的前端开发工作。

开发环境：

操作系统： Mac OSX && Windows 10

开发语言： Python3.4 HTML5 JavaScript

后端框架： Django

前端工具： Vue.js

配置管理：

将 config.json 中的 SITE_DOMAIN 设为自己服务器的地址；

进入项目目录，输入 uwsgi -ini uwsgi.ini 即可运行我们的项目；

在开发过程中，DEBUG 选项置为 True，在实际使用中，该选项置为 False。

前端交互：

前端页面主要分为两个大类：user 类型的前端页面和 student 类型的前端页面。user 页面包括所有人通用的界面，不具有特异性：如用户的绑定页面，以及课程弹幕功能的界面都包含在 user 页面中。而 student 类型的页面更多地服务于每个同学所要求的更具体更细节的功能，包括除了以上两个页面之外的其他全部页面。

为了保持前端设计的美观性，在开发前期我们使用墨刀工具，花费了大量精力来设计我们的前端页面，为后续的开发带来了极大的鼓励效应和引导作用，目标也更加明确。前端设计总体保持配色简洁明快的风格，操作也较为简便，达到“懒人”公众号的目的。

后端设计：

后端组根据前端组开发页面的需求，提供她们所需要的 api 和 url 跳转代码，以及相应的数据库存储和外组 API 信息转化成 json 格式的功能。

接口规范：

我们在项目前期撰写了一个简单的前后端联系的文档。其中要求前端页面可以有 GET 请求和 POST 请求，分别对应一个 API 类中的不同成员函数，实现前端组和后端组规定好这两个请求需要的接口（输入参数和输出参数）。如果运行过程中出现了异常，则抛出 error，并把它显示在服务器中，能被前后端获取到并显示给用户。

数据库设计：

包含 student, course, comment 等和学习相关的字段，由于学生、课程等字段存在多对多的关系，因此对于每一组关系，都维护一个关系字段来方便信息查找和存储（如 studentcourse）。

在我们公众号的开发过程中遇到了各种各样不同的问题，我们精选了一些让人印象深刻的问题，并把解决方法列举在下面。

根据期中和期末项目的用户体验过程来看，有一个巨大的问题是在跳转网页的时候，总会出现一个微信的中间页面待确认，造成了比较大的困扰，所以我们尝试点击按钮直接跳转网页。但是由于微信提供接口的限制，将按钮属性设置成“view”的时候，生成的公众号按钮的 URL 参数固定，不能传递 open_id 或者学号等参数，并不能满足用户的需求，所以最终只能作罢。

在制作弹幕界面的时候，我们需要使用 Qt 工具制作一个透明的浏览器部件。但由于 Qt 以 C++ 语言为基础，很难与 Django 项目本身进行通信交流。最后我们想到了一个折中的办法，首先在 Django 内部编写我们的网页，之后使用 Qt 自带的文字输入功能输入该页面的地址间接显示它，就避免了跨语言问题的出现。

另外，由于需要定时向用户推送提醒，因此需要使用 celery 框架进行定时任务的处理。在运用这一框架时，因为不能写在原来的 handlers.py 文件，而必须写在 tasks.py 中，导致我们无法使用原来的 handler 发消息，只能向微信服务器 post 一个 json 向用户发送消息，起初由于试图绕开这一限制，浪费了不少时间在研究如何解决上，发现最终这一问题只能向微信服务器发送请求来解决，因此导致 celery 的运用没有想象中那么的好，不过依然实现了功能

最后，在获取公告内容与标题时，由于 API 组直接将网页消息爬取下来，没有对一些转义字符进行处理，导致在显示时有许多奇怪的字符产生，此处调用 html 中的 unescape 函数在后端进行处理，是的前端能够正常的显示空格等字符。

测试总结

为了保证公众号各功能的正确运行，我们采取了较为详尽的测试方法。

首先对于后端 API 函数来说，我们使用了测试驱动开发的方法，一边进行每个函数的开发，一边对它们进行单元测试，所以在前后端交流的时候，出现的问题较少，测试覆盖率基本达到了 100%。

其次功能测试也非常重要，因为这关系到用户直接的使用体验。对于每个前端页面和 handler 逻辑，我们设计了各种各样“奇怪”的测试用例，来保证用户使用的过程中不会出现奇怪的错误，造成不必要的困惑。比如未绑定的用户在点击我们的功能菜单时，会弹出模板消息，提示用户绑定，保证使用流程的流畅度和清晰度。总体测试覆盖率达到 70% 以上。

服务器端采用 Python3 + Django 1.9 + uWSGI + nginx 进行配置，修改了 /etc/nginx/nginx.conf 与 /etc/sysctl.conf 的参数。

/etc/sysctl.conf 文件：新增 net.core.somaxconn = 4096 这一选项。

/etc/nginx/nginx.conf 文件：新增一段 uWSGI 和 nginx 桥接的代码。

感谢您的耐心阅读，来自用户的支持是对我们最大的鼓励！