



学习 VS 弃疗

软工微信项目

本次大作业唯一具有可玩性的微信号【学习 vs 弃疗】上线了！！

期末考试到了，您是否还因尚未复习而担忧？通过本测试号，您可以在线练习我们精心挑选的考题。

如果您想弃疗，可以进入“开脑洞”“猜数字”“猜歌曲”“点歌台”愉快地弃疗^^

“少爷吉祥”小组

种璐瑶 邱泓钧 张佳瑜

目录

壹、产品目标.....3

 系统功能目标.....3

 “我要复习” 功能.....3

 “我要弃疗”功能：3

 “排行榜” 功能：4

 系统性能目标.....4

贰、开发管理.....4

 过程管理.....4

 人员分工.....5

 开发日程.....5

 开发环境.....5

 配置管理.....6

叁、系统设计.....6

 前端交互.....6

 后端模块.....6

 微信消息处理.....6

 答题模块.....6

 猜数字模块.....7

 点歌台模块.....7

 排行榜模块.....7

 管理员模块.....8

 接口规范.....8

数据库设计.....	9
用户 Player.....	9
课程 Course	10
题目 Question.....	10
排行榜记录 Course_record	10
肆、重点难点.....	11
逻辑框架.....	11
交互体验.....	11
菜单排版.....	11
切换状态.....	11
昵称设置.....	12
点歌格式.....	12
添加题库.....	12
获取歌曲.....	13
猜歌名.....	13
点歌台.....	13
伍、测试总结.....	14
臭虫大战.....	14
排行榜.....	14
点歌台.....	14
交互体验.....	14
功能测试.....	14

性能测试.....	15
陆、系统部署.....	15
绑定——WeChat.....	15
框架——Django.....	15
服务器——SAE.....	16
版本管理——SVN.....	16
数据库——MySQL.....	16

壹、产品目标

系统功能目标

“学习 vs 弃疗”微信公众平台计划提供以下功能：

“我要复习”功能

“我要复习”功能目前针对清华大学软件学院二字班在 2014~2015 秋季学期期末所设的四门考试科目（【操统】、【软基】、【软工】、【计网】）设计。用户选择某门考试科目后，公众号将不断提供给用户与这门课程相关的题目供用户回答，直到用户回答错误为止；

“我要弃疗”功能：

【开脑洞】：用户点击开脑洞栏目后，公众号将不断提供给用户一些奇怪且富有趣味性的题目供用户回答，直到用户回答错误为止；

【猜数字】：用户点击猜数字栏目后，公众号将随机生成 5 位不重复的数字；用户向公

众号发送一个自己猜测的 5 位不重复的数字后，公众号将给出和正确答案相关的提示信息，用户可以不断猜测直到猜对为止；

【猜歌名】：用户点击猜歌名栏目后，公众号将不断提供给用户一小段音乐，用户输入自己所猜测的这一段音乐的歌名，直到用户猜测错误为止；

【点歌台】：用户点击点歌台栏目后，可以向公众号发送歌曲名或歌曲名加歌手名，公众号将根据此关键字为用户搜索该歌曲，若找到该歌曲则将其发送给用户；

“排行榜” 功能：

我们将记录所有参与答题的用户的答题情况，生成排行榜供用户查看，以鼓励用户多答题、勤复习。我们将针对“我要复习” 功能以及【开脑洞】、【猜歌名】栏目设置排行榜。

系统性能目标

“学习 vs 弃疗” 微信公众平台计划能够支持 100 人同时使用，另外，由于我们并没有类似抢票的高并发功能，因此对系统性能的要求并不高。

贰、开发管理

过程管理

每周例会	每周四下课后一起吃饭、讨论当前进度和下一步计划
集体开发	每周四、周六下午进行集体开发

人员分工

邱泓钧 前期调研、配置环境、管理员模块、数据库模块、猜数字模块、点歌台模块

种璐瑶 前期调研、配置环境、框架搭建、答题模块

张佳瑜 排行榜模块逻辑及前端页面

开发日程

时间	内容	具体内容
第九周	选题构思	经过一番激烈的讨论,最终决定做兼具学习和娱乐功能的公众平台
第十周	初步设计	对公众号的菜单栏、具体功能、开发环境做出了初步的构想
第十一周	配置环境	将最基础的 Django 代码部署到 sae 服务器上,并完成微信的接入
第十二周	框架搭建	接收用户发送的微信消息并进行解析和分发,设计各模块间的接口
第十三周	完善功能	填充各个接口所应该完成的功能,进行拼接、整合
第十四周	测试改进	系统基本搭建完成,进行测试,修复一些 bug,完善细节

开发环境

开发语言	Python2.7.8
开发框架	Django1.7
数据库	Mysql Server5.6
服务器	SAE
版本管理	SVN

配置管理

我们采用 sae 服务器+svn 的方式进行代码管理。具体来说，针对此项目，我们在 sae 上创建了多个版本；开发过程中，每个人在自己开发时分别使用独属于自己的 2,3,4 号版本，而整合代码时则使用 1 号版本。这样既方便了每个人开发自己负责的模块，又很好地使用了 svn 的代码合并功能，同时保证即使合并有问题也有各自的备份。

叁、系统设计

前端交互

前端交互主要体现在排行榜上。用户可以通过左右滑动查看各门课程的排行榜。排行榜显示前六名。用户可以由此知道自己的成绩在排行榜中的位置。

后端模块

微信消息处理

接收到微信服务器转发的消息后，提取用户 uid、消息类型、消息内容等信息，根据这些信息调用相应的处理函数，得到应该回复给用户的内容。将内容封装为 xml 数据后发回跟用户；

答题模块

若用户通过菜单选择某门课程，则将该用户的状态更改为相应课程的答题状态。同时，从数据库中随机查找一道对应课程的题目回复给用户；

若用户发送文本消息给公众号,且该用户此时处于某门课程的答题状态,则判断该用户的回答是否正确;若回答正确,则在数据库中随机查找一道用户在这一轮答题中未答过的题发回给用户,若题库中的问题都已答完,则发回题目全部答完的提示信息;若回答错误,则提醒用户本轮答题结束,同时更新排行榜并提示用户当前的排名;

猜数字模块

若用户通过菜单选择猜数字游戏,则将该用户的状态更改为猜数字状态。同时,将有关猜数字游戏的提示信息回复给用户;

若用户发送文本消息给公众号,且该用户此时处于猜数字状态,则对比该用户的猜测与正确答案,以 xAyB 的形式回复用户对比的结果;若用户猜测正确,则返回祝贺信息;

点歌台模块

若用户通过菜单选择点歌台栏目,则将该用户的状态更改为点歌状态。同时,将有关点歌台的提示信息回复给用户;

若用户发送文本消息给公众号,且该用户此时处于点歌状态,则根据用户输入的信息搜索歌曲,若搜到了歌曲,则将该音频回复给用户;若没有搜到,则回复没有搜到歌曲的提示信息;

排行榜模块

若用户点击菜单中的排行榜,则跳转至排行榜页面。排行榜列出操统,计网等六个项目的排名,根据用户连续答对题目的数量占题库题目总量的比例换算成分值进行排名。

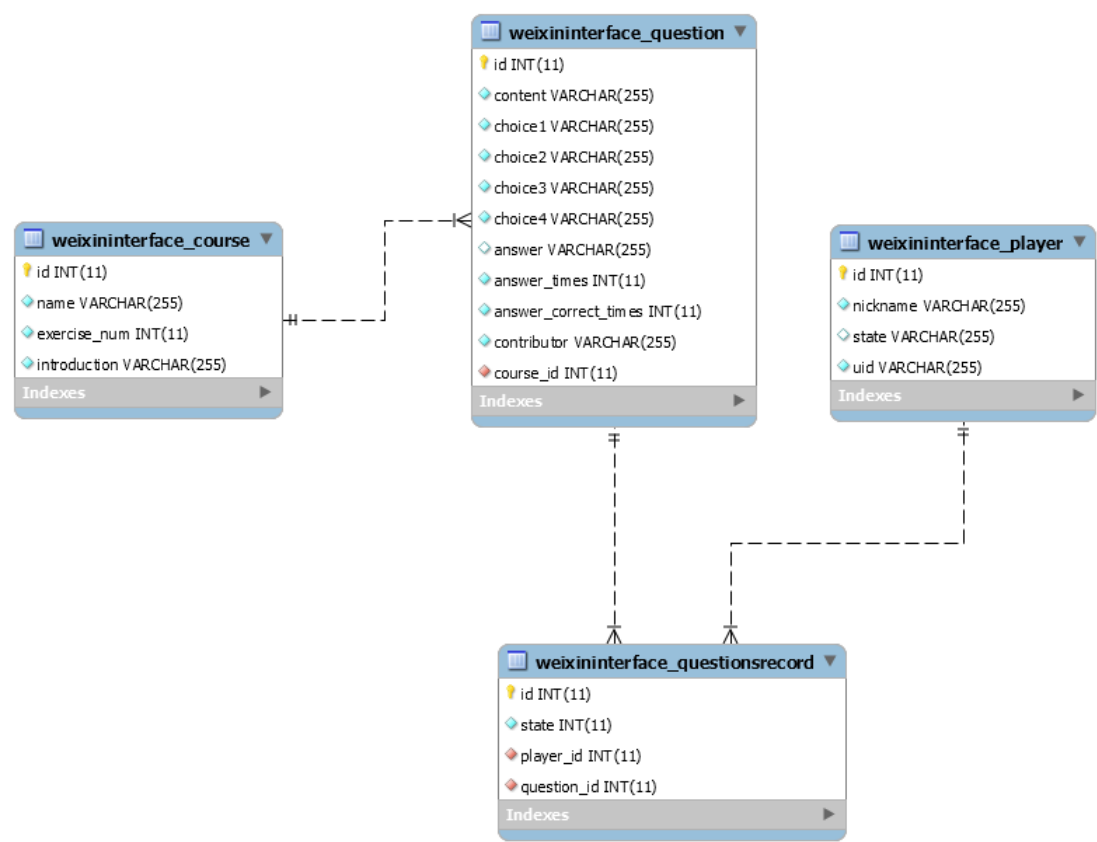
管理员模块

使用 Django 自带的管理员模块；

接口规范

接口	功能
msg_handler	根据输入的 xml 数据，调用相应的消息处理函数，得到要回复的消息及消息类型
reply_handler	根据要回复给用户的消息及消息类型生成 xml 数据包，用来发送给用户
event_questionhandler	根据用户 uid 和菜单事件 key 值所指示的课程名，从数据库中随机生成一道对应课程的题目回复给用户
text_questionhandler	根据用户 uid 和对前一道题的回答，判断回答的正误，并将对应的回应回复给用户
event_gamehandler	根据用户 uid，更改该用户当前的状态为“猜数字中”，并将相应的消息回复给用户
text_gamehandler	根据用户 uid 及其回复，判断其猜数字的情况，将相应的提示信息回复给用户
event_ordersonghandler	根据用户 uid，更改该用户当前的状态为“点歌中”，并将相应的消息回复给用户
test_ordersonghandler	根据用户 uid 及其输入的信息，为用户搜索歌曲，将搜到的歌曲或提示信息回复给用户

数据库设计



用户 Player

字段	类型	备注
uid	varchar(255)	根据 fromusername 生成的用户对此公众号的唯一标识
nickname	varchar(255)	用户关注公众号时为自己取的昵称，用于排行榜上显示用户名称
state	varchar(255)	记录用户当前的答题或娱乐状态

课程 Course

字段	类型	备注
name	varchar(255)	课程名称，包括【开脑洞】【猜歌名】两个项目
Introduction	varchar(255)	课程介绍，供管理员编辑、查看

题目 Question

字段	类型	备注
content	varchar(255)	题面内容
choice1	varchar(255)	选项 1
choice2	varchar(255)	选项 2
choice3	varchar(255)	选项 3
choice4	varchar(255)	选项 4
answer	varchar(255)	答案
contributor	varchar(255)	题目贡献者，供管理员查看
course_id	int(11)	所属课程编号，指定该题目属于哪个栏目

排行榜记录 Course_record

字段	类型	备注
player_id	int(11)	外键，指向创造该记录的用户
course_id	int(11)	外键，指向该记录对应的课程
score	int(11)	该条记录的得分

肆、重点难点

逻辑框架

虽然开始开发前已经有了较为详细的讨论规划,但是实际上开发时需求的变动仍然很大,导致代码的框架并不是很明确。尤其项目除了答题选项外,猜歌曲、猜数字、点歌台之间存在一定的差异,很容易造成功能的简单叠加。

经过重构,我们将与“我要学习”模块相近的“开脑洞”和“猜歌曲”部分融入答题逻辑,简化了代码的复杂性,同时提升了项目的可扩展性。

交互体验

在提升交互部分花了很多功夫。

菜单排版

最初的安排使最右的菜单作为 ABCD 的点击按钮,后来发现两级菜单需要点击两次,未必方便。因此最终仍然是设计为回复数字作为凭据。

切换状态

当用户从一个模块跳转到新的模块时,最初的设计是设立一个中间状态,提醒玩家正处在当前模块是否确定跳转,等玩家回复跳转才能跳转。

后来经过实际测试,发现严重影响答题的流畅性,因此改成提醒“您已离开 xx 模块”并立即开始新的模块。

再之后,经过两人讨论,决定将“您已离开 xx 模块”删去,使得整体更为简洁。

昵称设置

答题排行，天经地义。将昵称置入排行榜，才能够充分发挥排名的作用。

最初的想法是直接获取微信昵称。查阅了开发者文档，获取昵称功能会在用户关注时，向用户发送权限请求。我们认为，这样容易损失用户源，不如直接让用户设置自己的昵称。

同时，经过讨论，我们认为不应该让用户随时更改自己的昵称，因此在关注时便直接让用户直接输入。同时，为了保证部分用户确实有修改昵称的需求，因此取消关注后将昵称清空，重新关注后便能重新设置昵称。

点歌格式

最初是设置为“歌曲 歌手”模式。后来考虑到有的歌曲包含空格，无法正确解析。后来改成“歌曲##歌手”，发现有的输入法每次输入#时会自动切换为普通键位，需要重复切换，因此最后精简到只有一个#符号。

添加题库

因为将用户群精确定位到软院二字班，我们添加的题库为了保证精确性和趣味性，并没有从网络上去抓取大量的题库。无论是期末复习题库，还是脑洞大开题库，都能给同学们带来亲切的感觉。

此外，猜歌曲模块则是从酷狗 TOP500 挑选出本人常听的，流量超出很多还是毅然决然下了 125 首歌曲。然后再用 Adobe Audition 精选出 75 首歌曲片段上传到 BAE 上制成外链。其中歌曲片段的选取也是有一定的考究，较生僻的歌主要节选副歌降低难度，较熟悉的歌则故意在高潮戛然而止逼死强迫症.....

获取歌曲

最初的构想是通过语音消息将歌曲片段发送给用户。经了解后,才知道需要将歌曲上传到微信服务器从而获取到一个 MediaId 作为语音回复的参数,而微信服务器由于资源有限每三天就会删除数据。

因此果断投入音乐消息的怀抱。这时遇到一个问题,由于版权问题,音乐链接并不是那么容易获得。

猜歌名

猜歌名部分,起初找不到将音乐上传到网上获取外链的靠谱网站。最后发现百度有个用来提供其自身开发者使用的 BAE 空间可以制作免费外链。经亲测,与其他网站相比下载速率更快,因此统一采用 BAE 制作自身歌曲外链。

点歌台

点歌台部分则较为繁琐。由于版权问题,网上各大歌曲库都不支持外链。最后发现百度有一个非公开的音乐搜索 API,向其发送歌曲请求可以得到一个加密过的 xml 数据。<encode> 包含加密后的音乐链接,<decode>中包含解密后的音乐名称。将两者经过一定的处理后再拼接后便能获取到所需要的外链。

伍、测试总结

臭虫大战

排行榜

在展示的当天由于玩家激增，出现了排行榜崩溃的问题。原因很简单，榜单人数超过原先默认值导致数组越界。虽然这个 bug 很快就修复了，但也让我们体会到惨痛的教训。

点歌台

最初设置空格作为歌手名和歌曲名的区分，无法搜索英文为主包含空格的歌曲。后来改成#号却总是停止服务，最后才发现原来是编码问题。

交互体验

交互体验的具体优化前文已经提及很多，此处简要提一下心得。

有的时候，交互体验并不是纸上谈兵，而是需要实际去体验后才能明白，什么样才是用户真正喜欢的。我们往往谈论了很久终于确定了方案，等到实现出来却又发现还是原先的方案才是更好的。

虽然我们对于交互未必有多深的知识，但是从这一次项目里我们在一遍又一遍的自我肯定和自我否定中，我们对于交互的理解又加深了一步。

功能测试

功能测试方面，由于很多功能是逐步添加上去的，每添加一部分功能之前，我们各自都会在自己的版本号里测试运行通过，再整合到版本 1 里。尤其是猜数字、猜歌曲等相对独立

的模块，在整合到自身代码前还会先进行单元测试。

性能测试

由于我们并没有类似抢票的高并发功能，因此对系统性能的要求并不高，因此没有进行相关的性能测试。目前测试号最多能够提供 100 名用户使用，目前关注者 27 人且运行正常。

陆、系统部署

绑定——WeChat

作为迭代二临时策划的微信小项目，就是抱着让大家玩一玩的心态，因此只申请了测试号。发现测试号除了只能扫码添加以外，并没有什么不好，有效期也高达一年，用户数 100 也足够使用。



只是最开始搭建时，需要绑定 url 到测试号，尽管 token 等是正确的，配置总是失败，整整尝试了一个下午才发现问题所在，原来是少了最后的一个小小的“/”符号因此无法正确解析。

另外，采用了 WeiXinPost 等工具作为调试的主要工具。

框架——Django

由于迭代一看了 Django 的框架，因此迭代二毅然决然采用了 Django。

项目的框架采用 Django1.7，替我们省却了后台搭建的工作，让我们能专心在逻辑设计、前端交互以及题库管理。

项目不大，主要是处理微信响应消息，因此只创建了一个 weixininterface 的 app 处理所

有事件。

服务器——SAE

虽然只是测试号，但总是要开主机做服务器实在太麻烦，且不方便直接使用助教提供的服务器，因此我们微信开发模块决定自己申请服务器。

由于新浪服务器最初不需实名认证，因此申请 SinaAppEngine (SAE)。

我们在 sae 上创建了多个版本。开发过程中，每个人在自己开发时分别使用独属于自己的 2,3,4 号版本，而整合代码时则使用 1 号版本。这样既方便了每个人开发自己负责的模块，同时保证即使合并有问题也有各自的备份。

此外，歌曲片段由于外链需要，存在百度服务器 BAE 上。

版本管理——SVN

相比于 git，SVN 更为简洁，但是功能如分支管理等相对薄弱一些。所幸项目主要是两个人协同开发，使用 SVN 并没有遇到太多的难题，也算是学到了一个新的版本管理工具。

考虑到 SAE 源码不直接公开，为方便老师检查，特将源代码上传至 github 上：

<https://github.com/cocolotte/Wechat>

数据库——MySQL

此次采用 Mysql Server5.6，本地数据库使用正常。但是将本地数据库导入到 SAE 上遇到了重复创建数据库的问题。又是经过长时间的摸索，最后才知道需要删去 sql 文件中创建数据库的语句。

需要添加题库，只需要进入网址 <http://1.thsswelearn.sinaapp.com/admin/> (Username : cly , Password : 123) 修改即可。