# 关于微信机器人实例个数的选择

## 方案一

购买云服务器, 让微信机器人在上面跑.

使用itchat作为调用微信网页接口的库,

使用配置文件配置我们写的机器人,

配置文件可以被使用微信小程序的班级管理员修改,

然后机器人可以检测到配置文件的修改, 然后更改内存中的匹配规则.

所有的用户(同学 被管理者) 都需要先加这个微信机器人作为好友.

于是, 服务上就运行一个机器人实例, 管理方便, 用户体验良好, 没有复杂的注册要求.

## 方案二

每个班级都有自己的微信机器人账号.

当班长在使用本产品的时候, 需要提交一个公共使用的账号,

这个账号就就会跑在服务上了.

所以在同一时间, 服务器上会运行多个机器人实例.

对此, 需要对数据库进行微调, 让数据流向明确.

这样的话, 就不用给服务器准备账号, 减少运营成本,

并且, 可以减少因为只有一个账号的时候, 回复消息过于密集造成的封号风险.

# 关于管理员使用的后台的平台的选择

## 使用微信小程序作为开发平台.

使用微信小程序, 有利于多个手机平台的使用, 做到一次开发多次使用.

同时, 微信小程序有审核制度, 我们的项目作为一种消息分发系统, 有可能会不通过审核,

所以有一定的风险.

小组成员对web开发有一定的经验, 所以可以较快上手开发微信小程序, 但是组员对js并不是很熟悉, 所以还是有一定的不确定性.

所以说, 微信小程序作为多个手机平台都可以使用的开发环境, 是有很强的吸引力的, 对用户比较友好.

小程序的测试环境还在开发维护阶段, 社区也还在建立, 这会对我们的开发造成一点影响, 小程序的测试环境可以使用腾讯的XTest, 虽然只是支持安卓平台上的测试, 但是作为新手的开发, 应该是够了的.

## 使用android作为开发平台

Java语言是组员的必修课, 所以能较快的上手android的开发.

使用android开发会比较自由, 不需要经过注册什么的.

而且组员有一定的基础, 能写出容易维护的java代码.

android的开发环境非常成熟, 有许多测试工具.

但是, 因为手机的多平台性, 所以, 不能顾及ios用户,

所以就需要让班长和其他班级管理人员都要使用android手机.

我们不考虑ios的开发.