# 关于微信机器人实例个数的选择

## 方案一

整个服务器里面运行一个微信机器人实例, 只使用一个账号, 所有的消息都由这个机器人分发, 这样可以让这个分发服务相对于用户看上去, 更加的透明化, 这样之后, 作为每个班级的管理员用户可以更加方便的注册和使用, 同时减少数据库的复杂度, 减少工作量, 可以更加容易的被实现.

但是, 有风险, 会造成微信机器人的回复频率很容易超过微信官方的阈值, 如果同时处理太多的请求, 然后在不减少用户体验的情况下, 迅速的回复那些请求, 就会触发微信的保护机制, 使得账号被封号一段时间, 这个是灾难性的事故.

但是如果人为的控制消息回复频率, 就会造成用户体验不够好, 会导致用户重复的发消息, 然后造成更久的延迟.

## 方案二

服务内运行多个机器人的实例. 每个班级对应一个实例.

该实例的账号需要让使用的班级提供 ,这个账号就专门变成了微信机器人了. 这个账号的实际拥有者就不能再使用这个账号了.

这个方案的话, 因为有多个实例, 所以可以减少被微信官方封号的可能性. 但是会增加用户的操作,和前期工作, 会造成一定的门槛.同时, 我们需要增加数据库的表项来支持不同的微信机器人实例. 会增加编码难度.

## 方案三

本方案认为需要运营者来提供多个账号, 一个微信机器人对应多个班级, 同时一个班级里面的同学会加到不同的微信机器人.

让同一个班级的成员加不同的微信机器人账号,可以减少一个微信机器人发消息的频率, 因为请求的高发期经常是由于班长群发消息造成的, 所以如果一个班级分成多个机器人分发, 就会降低这个班级的消息对同一个机器人造成的冲击.

同时有利于管理流量的分布, 我们可以按照一段时间内的机器人的工作量, 给同学们更换机器人. 使得所有的机器人的工作量不会太高而引起封号.

那么这个需要更加多的代码量, 和更加复杂的数据库和程序逻辑的设计.

# 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.