概要设计

**深圳市三满科技有限公司**

**2018/2/28**

摩尔衣橱APP项目

[设计目标 2](#_Toc507610374)

[技术选型和规范 2](#_Toc507610375)

[技术优势 3](#_Toc507610376)

[风险控制： 3](#_Toc507610377)

[美工，架构设计： 4](#_Toc507610378)

[数据库设计： 4](#_Toc507610379)

[Web前台设计： 5](#_Toc507610380)

[PHP工程师： 5](#_Toc507610381)

[前后端集成及数据交互设计： 6](#_Toc507610382)

# 设计目标

实现共享衣橱项目相关业务的Web端首页，后台管理和APP（IOS、Android）应用。

APP名称：摩尔衣橱，More Closet

详细需求可参考《摩尔衣橱项目需求清单》

# 技术选型和规范

* 服务器：阿里云 LAMP+万网域名；
* 使用三满科技的轻量级Bible框架简化设计思路和代码编写：
  1. 所有业务操作（数据操作）统一用存储过程实现； - 数据分析维护更清晰；
  2. 服务器PHP层不对数据进行过多的处理直接透传给前端；- 减轻服务器负荷；
  3. 前端通过Js将数据格式化展现在UI中； - 数据处理前移缩短了交互时间；
  4. 数据的更新、提交全部通过Ajax方式实现； - 减少页面数量、获得更好的感知；
  5. UI页面设计模块化，模版化，极大提高开发速度和协调性；
* 聚焦于业务实质，注重数据的整洁，交互逻辑的实现大部分由js完成；
* 使用流行的前台Bootstrap框架和APICloud快速实现界面呈现；
* 项目进度跟踪和分解分配，文档管理，使用<https://app.geeseteam.com/> 在线平台
* 开发工具：
  1. WAMP：本地数据库，网站调试；【数据库、后端工程师】
  2. PHPMyadmin：数据库创建，存储过程编写；【数据库】
  3. Hbuilder：CSS，HTML5，JS代码编辑器；【前台css，js】
  4. APICloud Studio2 （带SVN）：APP界面开发；【APP】
  5. Netbeans：PHP代码编辑，quanwutong【PHP】

# 技术优势

* 引入APICloud平台，一次开发直接生成IOS和安卓两个APP，提高开发效率；
* 引入APICloud平台，大量成熟功能模块，可快速集成，免去原生Java代码的开发过程；ajax
* Bible框架实现了数据，后台PHP，前台UI的完全隔离，各模块可独立开发和调试，提高效率；
* Bible框架实现了界面开发的模版化和模块化，不管有多少网页，只要网页中有雷同的内容，这段雷同的内容在项目中只需要编辑一次

# 风险控制：

* 业务逻辑代码完全独立开发，似懂非懂，不了解原理的代码一律不使用，避免了问题定位，源代码读取的时间；
* 项目开发使用的是最流行的语言和技术，技术支撑人员非常多，交接简单；
* 申请APICloud VIP服务，阿里云VIP服务，紧急问题出现，会有官方专家一对一协助解决；

# 美工，架构设计：

【Scope】

美工负责整个站点的美学：风格，色调，前景色，背景色，字体，字体颜色，Logo，图片，链接样式，强调，警告，图标，文字描述等；

负责制定图片规范，比如图片格式，命名规则，图片大小等；

开发过程分两个阶段：

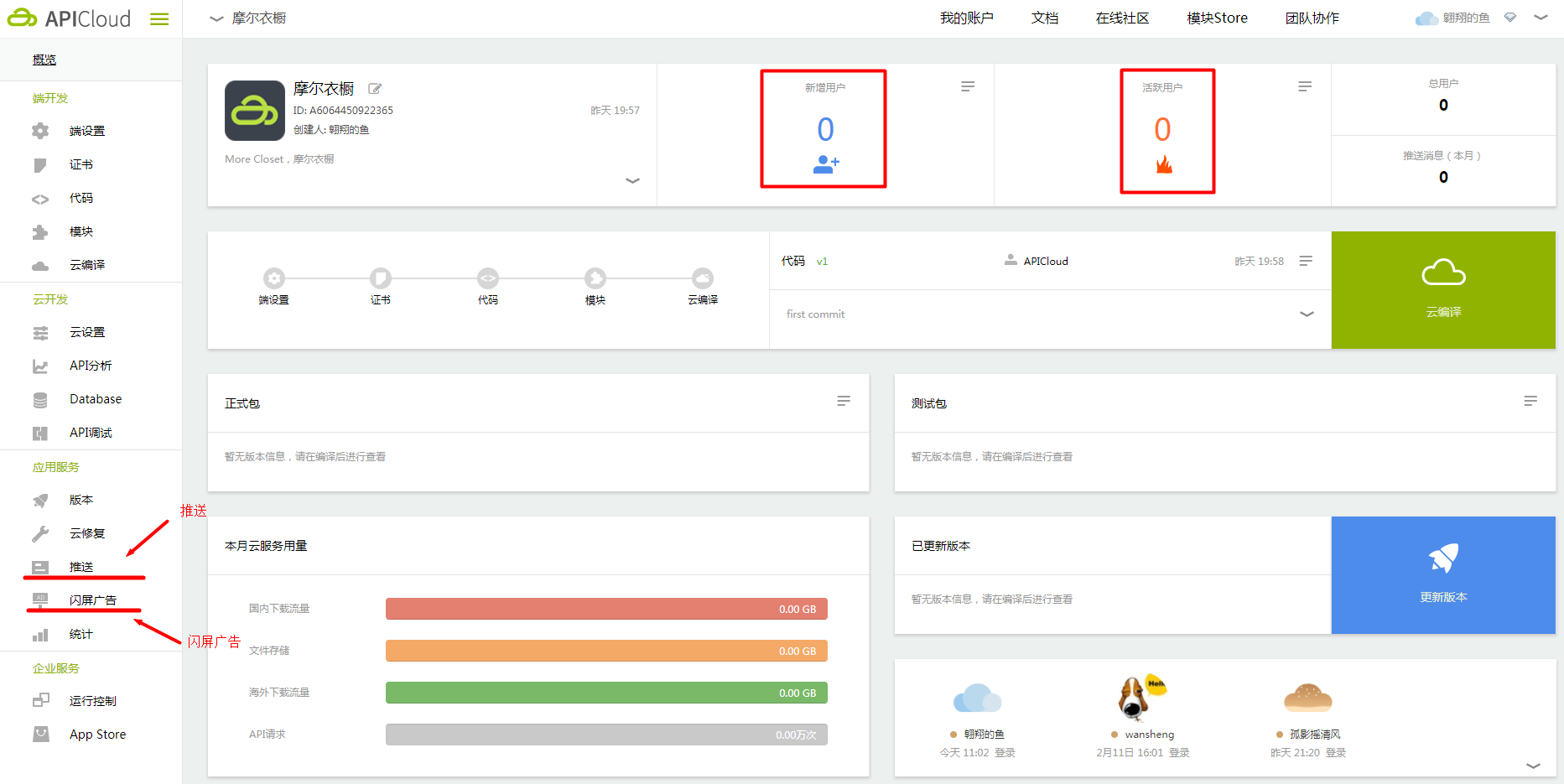
第一个阶段需要快速的给出一套临时性图片，用于内部前台开发，调试用；并给出相应的规范供开发者参考；

第二阶段是根据渐进的设计，开发者的反馈，客户的反馈，或者审美的提升，而给出的最终版图片，最终版图片只要一比一替换内部版的图片即可实现图片的更换；

【思路】

* Web首页样式 参考：<http://www.byton.com/>
* Web后台样式，参考：<http://wrapbootstrap.com/preview/WB0573SK0>
* App 样式，参考：女神派 + 流行元素：
  + Logo：简约大气，大牌，背景单一，不花里胡哨
  + 启动页：单色简约Logo（给出三份草图）
  + 广告引导页：倒计时（内容可动态更改）
  + 推送功能：
  + 首页：





【详细输出】

<Anex1\_美工设计.xls>

# 数据库设计：

【Scope】

负责数据库的创建，设计，以及网站所用到的所有交互逻辑，直接从存储过程中实现；

设计的目标是网站中不能有任何一个地方需要通过PHP代码直接访问数据库；

数据库使用Mysql， 设计数据库工具使用PHPmyAdmin

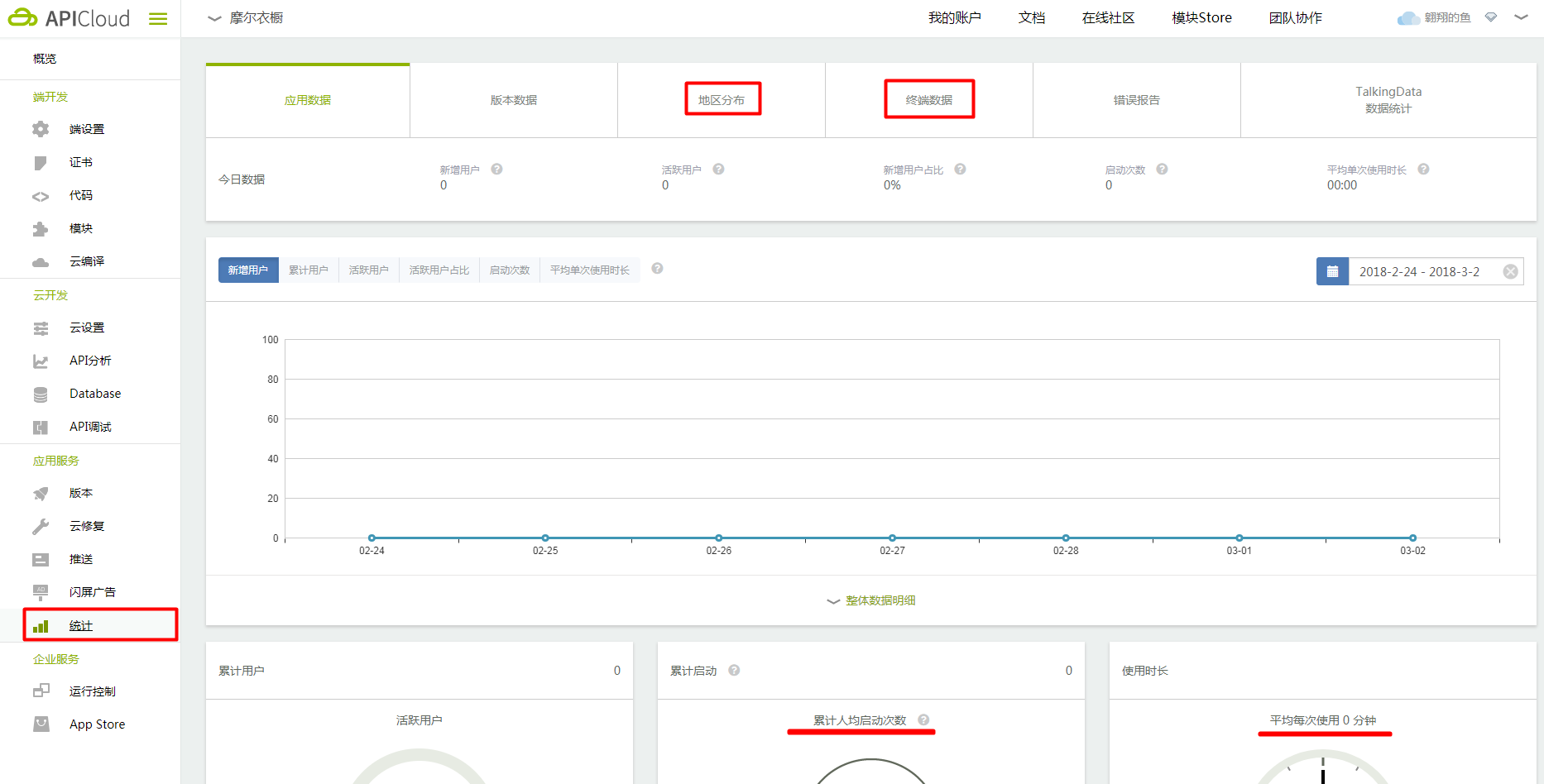
【思路】

User：用户登录，注册，角色，权限等作为独立的数据库进行设计 – 优点：安全，易维护和拓展；

Morecloset数据库：模块化封装业务需要的数据，支撑后续未知的新业务思路出现后快速上线；

后台数据需要支撑各种各样灵活分析以供管理层决策谋划新的营销或商业活动：

APP层面：



数据库层面：

【详细输出件】

《Anex2\_数据库设计.xls》

# Web前台设计：

【scope】

负责网站中所有带有客户可见输出的页面的创建和编辑；

本次项目页面使用bootstrap前端框架（主要目的是响应式，PC和手机同步）；

开发过程分三个阶段：

第一个阶段：

按照美工的设计，选择合适的框架模版，和插件，快速堆叠出所有用到的静态网页；

网页的位置，对资源文件的引用（图片的引用等）需要统一考虑,要能支撑后续对界面更改后不会有返工现象；

第二个阶段：

对每个页面进行精细化设计，要能支撑各项业务操作过程中可能出现和使用的所有样式并预先准备好；比如：

设计结果要包括该页面所有可能用到的样式（CSS），

以及与后台数据交互无关的JS函数（如js插件的初始化，组件内部的显示逻辑等）的实现；

所有设计内容只能在HTML，js，css文件实现，不要依赖任何PHP代码；

对于所有需要后台提供的数据，都以JSON格式的js变量为起点进行初始化，并支持更新操作；

对于所有需要向后台提交的数据，也都以JSON格式的方式暂存在js变量中；

【详细输出件】

《Anex3\_前台设计线框图+文件地图》

# PHP工程师：

【scope】

负责用户访问控制，用户不可见的服务器端必要的逻辑实现（如登录验证，权限认证，跳转，文件操作，数据库读取，浏览器数据输入输出等），全部以函数的形式实现，不会与css，js，html等掺杂在一起；

PHP工程师负责为数据库和前台设计的人编写测试代码并进行测试；

PHP工程师也会参与到前台设计和数据库设计工作中；

【详细输出件】

服务器上的一堆php文件，一堆测试网址，一堆测试数据；

# 前后端集成及数据交互设计：

【scope】

通过Bible框架等机制，有机的将静态的前台页面和数据库联通起来，实现端到端的业务联通（PHP，JS，Ajax）

【详细输出件】

<前后台交互接口文档.xls>

整站代码