软件设计文档

技术选型及理由

前端：小程序开发框架。
理由：小程序是基于微信的 web 开发，必须使用微信小程序开发框架。框架提供了自己的视图层描述语言 WXML 和 WXSS，以及基于 JavaScript 的逻辑层框架，并在视图层与逻辑层间提供了数据传输和事件系统，可以让开发者可以方便的聚焦于数据与逻辑上。
框架的核心是一个响应的数据绑定系统。整个系统分为两块视图层（View）和逻辑层（App Service）框架可以让数据与视图非常简单地保持同步。当做数据修改的时候，只需要在逻辑层修改数据，视图层就会做相应的更新。框架提供了小程序的页面管理、基础组件、微信原生 API。

后端：后端框架为 Apache+MySQL+PHP。其中 Apache 是 web 服务器软件，提供 HTTP 服务，MySQL 是数据库，提供数据处理服务，PHP 为后端开发语言。
理由：Apache 是排名第一的 web 服务器软件，快速、可靠、跨平台。MySQL 是最流行的关系型数据库管理系统之一，体积小、成本低、开源，一般中小型网站都选择 MySQL。PHP 是最流行的 web 后台脚本语言，上手快、开源、跨平台。考虑到学习成本以及稳定可靠性，选择了 Apache+MySQL+PHP 的组合，这个组合也是非常流行的组合并且都是开源的，有很多的学习资料。

软件设计技术

软件设计技术：结构化程序设计，面向对象程序设计，面向切面编程，面向服务的架构设计，设计模式。

结构化程序设计：
采用自顶向下、逐步求精及模块化的程序设计方法；使用三种基本控制结构构造程序，任何程序都可由顺序、选择、循环三种基本控制结构构造。结构化程序设计主要强调的是程序的易读性。
该部分设计在整个程序许多地方都有涉及，例如：Page/register/resgister.js
文件中，第 69 行到 79 行的 formSubmit 的函数中的一部分代码块，它负责用户信息的提交过程，它首先包含了信息判断的语句，用于判断信息是否为空。仅仅只有顺序和选择两种结构，并且只有一个出入口

面向对象的程序设计(OOP)：
该技术针对业务处理过程中的实体及其属性和行为进行抽象封装，以获得高效清晰的逻辑单

元划分。
主要是项目中的 wx.request()语句，例如：Page/register/register.js 中，第 82 行到 115 行的函数，主要涉及的功能是用户数据的提交，并且可以在控制台返回一个签到的函数。对象即为用户信息的数据 data，包含了 name,number,school 三个信息。并且封装了将用户信息存入手机缓存的方法，更改全局变量，提示跳转的动态窗口，以及跳转的方法。并且将函数通过数据库的 url 指定接口进行联系。

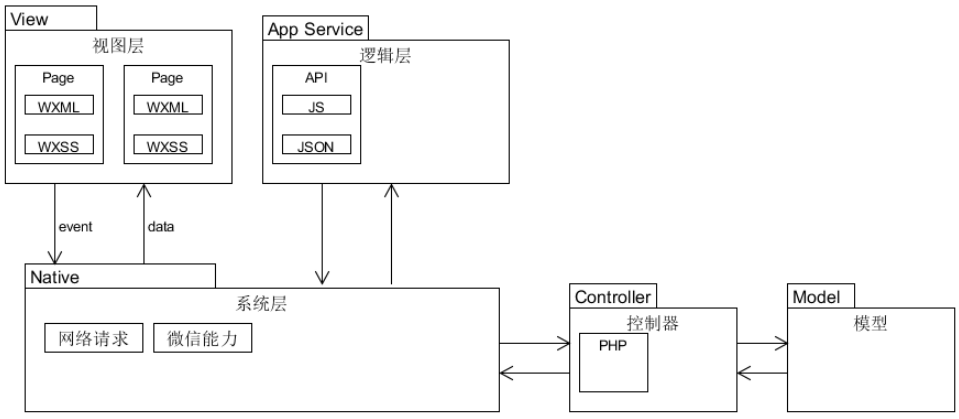
面向切面编程(AOP):
对约为逻辑的各个部分进行隔离，从而使得业务逻辑各部分之间的耦合度降低，提高程序的可重用性和开发效率。与 OOP 的区别是，OOP 强调封装，AOP 强调隔离，是 OOP 的延续，适合为分散对象引入公共行为。动态 AOP 通过预编译方式和运行期动态代理实现程序功能的统一维护的一种技术。
这种技术应用普遍存在在项目的 JSON 文件中，比较典型的例子就是根目录下的，app.json 文件，它包含了整个主界面的 windows 和 tabbar 的风格。还有所有的 wxss 文件，包含了一个界面统一的风格，如果未定义则根据 app.wxss 的风格来定义。

面向服务的架构设计(SOA):
一种组件模型，通过接口将不同的功能单元（服务）联系起来。 接口的定义应该独立于提供服务的硬件、操作系统、编程语言。 以达到各个功能单元能以统一、通用的接口进行交互。
具体的实现部分对应了各文件夹中的 wxml 文件，以 register 为例，它通过界面的方式将注册这个大功能下的诸多小功能联系起来，通过姓名，学号，学校的输入框，来完成用户信息的输入，然后通过 注册 的按钮来实现用户信息的提交和基本的验证工作。并且完成下一界面的跳转。

设计模式:
本次项目设计时，主要采用的 MVC 的设计模式，将项目分为 model,view,control 三大类，model 的部分主要涉及 PHP 代码部分对于用户模型的建立。View 的部分最主要是 wxml 部分文件，负责建立起前台的界面视图部分。control 主要实现在 js 文件中，实现各个功能的控制语句编译工作。

架构设计

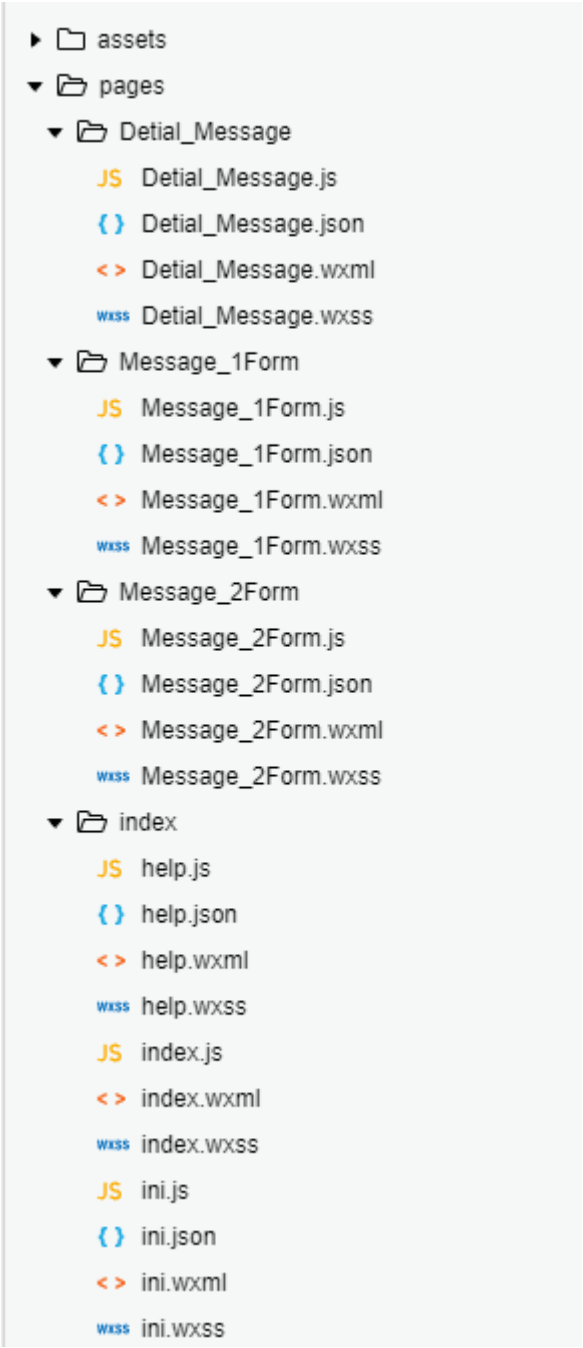
本系统结合微信小程序系统，采用 MVC（Model-View-Controller）架构模式，将微信小程序系统封装成 View 层。因此系统分成微信小程序端部分（View）、控制器（Controller）和模型(Model) 三层。

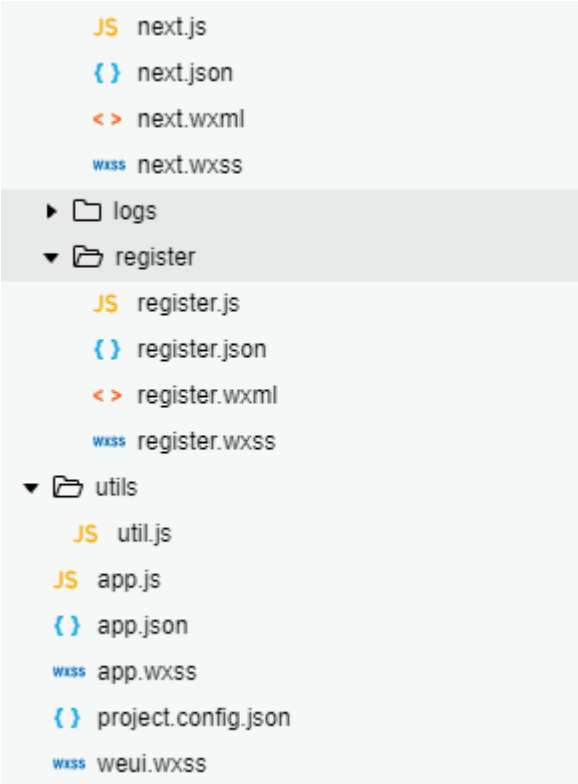


模块划分

前端:
以功能为基本单位，分为以下模块
注册/登录
发起签到
查看已发起的签到以及签到详情
参与签到
查看已参与的签到
删除已发起的签到
使用帮助信息

代码结构


assets
pages
Detial_Message
Detial_Message.js
Detial_Message.json
Detial_Message.wxml
Detial_Message.wxss
Message_1Form
Message_1Form.js
Message_1Form.json
Message_1Form.wxml
Message_1Form.wxss
Message_2Form
Message_2Form.js
Message_2Form.json
Message_2Form.wxml
Message_2Form.wxss
index
help.js
help.json
help.wxml
help.wxss
index.js
index.wxml
index.wxss
ini.js
ini.json
ini.wxml
ini.wxss



后端:

每个模块对应前端相应的模块，与前端各个模块进行数据交互。

代码结构