# 软件设计文档

版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 作者 | 参与者 | 日期 | 备注 |
| 1.0 | 蔡凡 |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 修改版本 | 修改章节 | 修改描述 | 作者 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

修改记录

目录

**[1. 引言](#_Toc5888_WPSOffice_Level1)** **[3](#_Toc5888_WPSOffice_Level1)**

[1.1编写目的和范围](#_Toc13386_WPSOffice_Level2) [3](#_Toc13386_WPSOffice_Level2)

[1.2术语表](#_Toc10787_WPSOffice_Level2) [3](#_Toc10787_WPSOffice_Level2)

**[2. 全局数据结构说明](#_Toc13386_WPSOffice_Level1)** **[3](#_Toc13386_WPSOffice_Level1)**

[2.1 常量](#_Toc24634_WPSOffice_Level2) [3](#_Toc24634_WPSOffice_Level2)

[2.2 变量](#_Toc3776_WPSOffice_Level2) [3](#_Toc3776_WPSOffice_Level2)

[2.3 数据结构](#_Toc2496_WPSOffice_Level2) [4](#_Toc2496_WPSOffice_Level2)

**[3. 架构设计](#_Toc10787_WPSOffice_Level1)** **[4](#_Toc10787_WPSOffice_Level1)**

[3.1 架构描述](#_Toc23094_WPSOffice_Level2) [4](#_Toc23094_WPSOffice_Level2)

[3.2 架构图](#_Toc25828_WPSOffice_Level2) [4](#_Toc25828_WPSOffice_Level2)

**[4. 模块划分](#_Toc24634_WPSOffice_Level1)** **[5](#_Toc24634_WPSOffice_Level1)**

[4.1 视图层](#_Toc20315_WPSOffice_Level2) [5](#_Toc20315_WPSOffice_Level2)

[4.2 控制器](#_Toc13413_WPSOffice_Level2) [5](#_Toc13413_WPSOffice_Level2)

[4.3 模型](#_Toc3111_WPSOffice_Level2) [6](#_Toc3111_WPSOffice_Level2)

## 引言

### 1.1编写目的和范围

本设计说明文档编写的目的是说明程序模块的设计考虑，包括程序描述，输入/输出，算法和逻辑流程等，为软件编程和系统维护提供基础。本文档的预期读者为系统设计人员，软件开发人员，软件测试人员和项目评审人员。

### 1.2术语表

|  |  |
| --- | --- |
| 术语 | 含义 |
| 学生 | 普通登陆用户，中山大学学生，有查看课程目录，加入课程，课程签到等功能。 |
| 教师 | 普通登陆用户，中山大学教师，有查看课程目录，创建课程，修改课程信息，发起签到等功能。 |
| 管理员 | 维护小程序的信息，包括对网站内容的更新、发布、删除。 |
| 课程目录 | 用户的课程列表 |
| 课程信息 | 一门课程的基本信息，包括课程名称、上课地点、上课时间、教室、课程人数、课程ID等 |

## 全局数据结构说明

### 常量

### 变量

wxId: 微信的标识Id

name: 用户输入的姓名

IdNum：用户输入的学号

### 数据结构

globalData: {

wxId: ' ',

name: ' ',

IdNum: ' ',

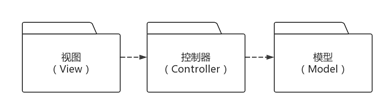
},

## 架构设计

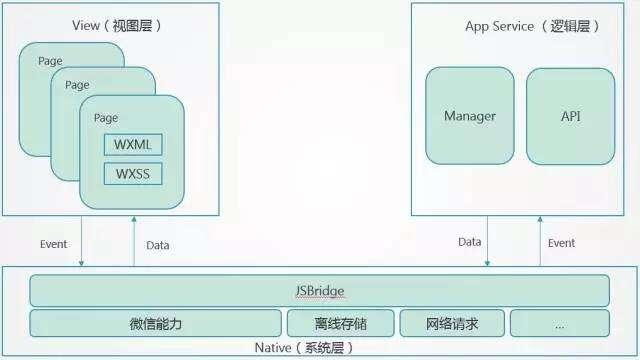
### 架构描述

应用的架构为MVC三层架构，分别为表现层，业务层和数据层。MVC架构将三层分开，使得可以在改变表现层时不用再重新编写业务逻辑，从而实现了开发的模块化，使得开发清晰明了，大大提升了开发效率。表现层为图形界面，主要负责与用户的交互。业务层负责处理业务逻辑，对请求进行处理。模型层封装了与应用程序业务逻辑相关的数据。

### 架构图



本项目采用微信小程序开发工具作为开发工具，其官方架构图如下：



## 模块划分

### 视图层

小程序视图层共划分为四个模块：

1. 注册登录界面 （index）
2. 学生界面 （student）
3. 教师界面 （teacherPage）
4. 课程信息修改页面 （Form）

小程序的视图层每个page对应有自己的逻辑处理函数和配置文件。通过系统层的微信能力和网络请求，和控制器进行数据交互。

### 控制器

控制器接受用户输入信息，与后端数据库进行数据交互并调用模型和视图完成用户的需求。

本系统的控制层由以下几个模块组成：

1. 登录注册逻辑（register）
2. 发起签到逻辑（teacherControl）
3. 参与签到逻辑（signup）
4. 课程管理逻辑（ManagerControl）

### 4.3 模型

模型封装与应用程序相关的企业数据和业务规则。模型可为多个视图提供数据，重用性高。

本系统的模型层由以下两个数据模块构成：

1. user（存储用户信息的模型，主要包括学号姓名与唯一绑定的微信号以及身份标签）
2. Courses（存储课程信息的模型，包括了课程名称，发起/结束时间，地点，签到学生信息，发起人姓名）
3. SignInTable（存储签到信息的模型，包括签到者的学号，签到状态）

## 5.软件设计技术

Structure Programming：是按照模块划分原则以提高程序可读性和易维护性、可调性和可扩充性为目标的一种程序设计方法。在结构化的程序设计中，只允许三种基本的程序结构形式，它们是顺序结构、分支结构、(包括多分支结构) 和循环结构，这三种基本结构的共同特点是只允许有一个流动入口和一个出口，仅有这三种基本结构组成的程序称为结构化程序。

在登录注册模块中，按照用户输入信息—点击注册—数据保存至后台数据库这一顺序结构。

Object-oriented Programming: 一种计算机编程架构。其一条基本原则是计算机程序是由单个能够起到子程序作用的单元或对象组合而成。

在各个页面中都有Data这个数据结构，主要负责对页面组件的数据渲染，在js逻辑文件内能够通过this对象访问；在app.js内设置的globaldata数据结构，能在其余各个js文件内访问。

### Design Patterns: 代表了最佳的实践，通常被有经验的面向对象的软件开发人员所采用。设计模式是软件开发人员在软件开发过程中面临的一般问题的解决方案。在本项目中采用了MVC设计模式。

共有三个模型user，SignInTable，courses。通过服务端的controllers里的js接口对三个模型进行数据操作，并将获取的数据渲染到客户端的页面pages中

