[HTMLPreview](http://markdown.pioul.fr/)

822 words

# 公司培训选课系统架构设计文档

## 一.概述

### 需求背景

* **目前公司选课所遇到的问题**

1. 行政人员发布课程，统计课程报名信息，提醒学员讲师上课，发送评价连接，管理课程信息等方面需要耗费 较多时间和精力
2. 行政人员在统计课程，提醒学员或讲师上课等流程中可能出现漏掉信息甚至出错
3. 学员选课难以获取课程完整信息，选择的课程可能与自己想象的不太一样
4. 学员选错课，修改选课信息不是很方便

### 需求规格描述

* **管理员**

1. 新增课程，即新增课程及其详细信息
2. 发布课程，即将课程及其信息发布出来供员工选课。
3. 查看课程，即查看课程的详细信息
4. 修改课程，即对课程信息进行修改
5. 下架课程，即学员可选课程已经取消了
6. 踢出课程已选人员，即有些课程可能只限于一定岗位的人选择，可以踢出不符规则的人
7. 查看报名和签到情况，即查看讲师所讲授课程报名和签到情况
8. 获取签到二维码，即获取学员签到二维码
9. 帮助学员报名功能
10. 讲师姓名提示和验证功能，会在新建课程和提交修改课程时提示讲师姓名。
11. 指定课程性质，必修课不计入一年选课限制中，选修课计入一年选课限制次数中。

* **学员**

1. 查看可选课程列表，可以对课程按分类，名称进行搜索，展示可选课程列表。
2. 课程报名，即课程进行报名
3. 课程报名取消，即对已报名的课程取消报名
4. 查看已选课程，即查看完成报名的课程
5. 课程评价，即对已上完的课程进行评价
6. 查看课程详情，即查看课程详细信息
7. 申请开课，即学员申请开课
8. 选课限制，学员一年选选修课不超过6门
9. 查看该课程其他报名人员

* **讲师**

1. 查看课程列表，即查看所讲授课程列表，分页列表
2. 查看课程详情，即查看所讲授单个课程的详细信息
3. 查看报名和签到情况，即查看所讲授课程报名和签到情况
4. 获取签到二维码，即获取学员签到二维码

* **系统登录对接**

1. 支持企业微信三方登录 待补充功能点详细说明

### 系统限制

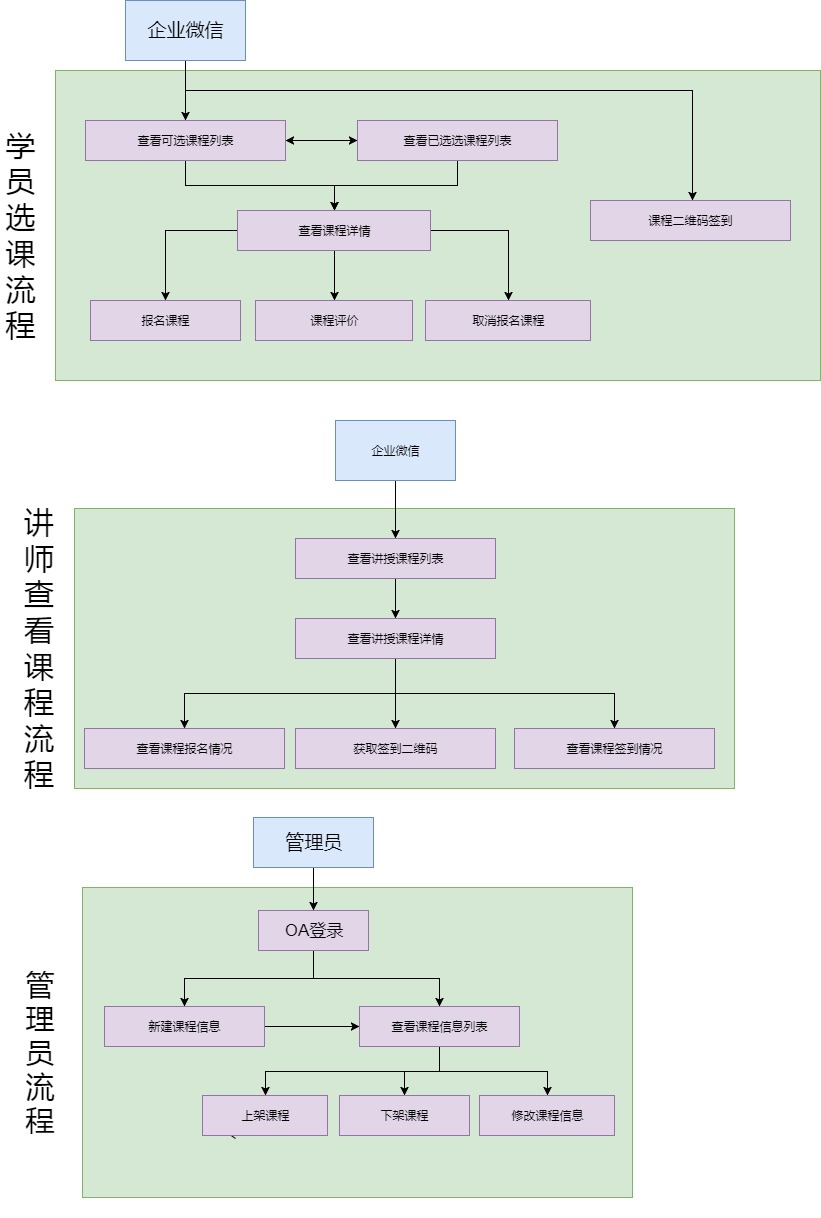
* **本节描述待开发系统的不支持的功能**

1. 暂不支持学员上课提醒功能
2. 暂不支持学员查看历史课程功能
3. 暂不支持讲师查看学员对该课程的评价和留言
4. 暂不支持管理员限制学员选课功能
5. 暂不支持学员申请开课
6. 暂不支持管理员帮助学员报名功能

### 设计思路

* **本节描述待开发系统的基本设计想法**

**总体设计 选课系统的总体结构如下**：

****

* 1.选课系统界面分为三部分设计开发，分别是管理员部分，讲师部分，学员部分，各自实现相应的功能，前端编写界面，由后端提供相应的接口。
* 2.本系统涉及二维码获取，二维码签到，企业微信登录等功能

### 对外接口

* **本节描述待开发系统的对外的接口。即外部系统如何通过这些接口进行功能调用。需要描述清接口的调用形式、格式、各个字段类型、取值范围。一般接口调用都有请求和返回。请分别写清楚。可以考虑直接贴入协议文件。比如说pb格式的.proto文件，http的json格式文件。此处以json文件举例。 批注：建议接口和需求规格分类对齐，按角色去划分接口。接口这里具体功能我先不看了。等看下一稿 restful规范**

#### 一.系统配置

1.1 获取课程一级分类

* URL: course/category-list
* Method: GET
* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : //响应描述，code非0时有错误描述

"data" :[

{

"category\_id": 1 ,//一级分类id号

"title": "一级分类名称",

"son\_list":

{

"category\_id": 2,//二级分类id号

"title": "二级分类名称"

}

}

]

}

1.2 获取讲师或学员身份

* URL: course/identity
* Method: GET
* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : //响应描述，code非0时有错误描述

"data" :[

{

"identity": 1,//1代表学员，2代表讲师

}

]

}

1.3 对接企业微信登录

* URL: course/wechat
* Method: GET
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| code | 包含用户企业微信userid等信息 | 否 |
| state | 包含企业所传信息 | 否 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : //响应描述，code非0时有错误描述

}

1.4 对接oa登录功能

* URL: admin/validate
* Method: GET
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| AUTH\_TICKET | oa验证信息 | 是 |
| AUTH\_DOMAIN | oa验证信息 | 是 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : //响应描述，code非0时有错误描述

}

#### 二.学员

2.1 查看可选课程列表

* URL: student/course-list
* Method: GET
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| parent\_id | 课程一级类别 | 否 |
| category\_id | 课程二级类别 | 否 |
| title | 课程名称，允许课程名称模糊匹配 | 否 |
| page\_size | 每页条数 默认20 | 否 |
| page\_index | 当前页码 默认第一页是1 | 否 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : //响应描述，code非0时有错误描述

"data" :{

"page\_index":1, //页数，表示第几页

"page\_size":20, //表示一页显示的课程数

"page\_total":10,//表示总共有多少页

"list":[

{

"course\_id": 1 ,//课程id号

"title": "课程名称",

"date": "2012-12-12 14:10" //开课日期

}

]

}

}

2.2 我的课程（已选课界面）

* URL: /student/my-courses
* Method: GET
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| page\_size | 每页条数 默认20 | 否 |
| page\_index | 当前页码 默认第一页是1 | 否 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : "",//响应描述，code非0时有错误描述

"data" :{

"page\_index":1, //页数，表示第几页

"page\_size":20, //表示一页显示的课程数

"page\_total":10,//表示总共有多少页

"list":[

{

"course\_id": 1 ,//课程id号

"title": "课程名称",

"date": "2012-12-12 14:10" //开课日期

}

]

}

}

2.3 查看课程详情

* URL: /student/detail-course
* Method: GET
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| course\_id | 课程id | 是 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : "",//响应描述，code非0时有错误描述

"data" :

{

"title": "课程名称",

"teacher\_name": "" ,//课程讲师姓名

"start\_date": "" //课程开课日期Y-m-d H:i

"end\_date": "" //课程下课日期Y-m-d H:i

"address": "" , //课程开课地点

"total\_num": 1 ,//课程可选总人数

"selected\_num": 1 ,//课程已选人数

"description":"" ,//课程内容介绍

"nature":"" ,//课程性质，0表示选修，1表示必修

"comment\_url":"" ,//课程评价url

"enroll\_status":0 ,//报名状态，0表示未报名，1表示已报名

"is\_full": 0 ,//报名人数是否已满,1表示已满，0表示未满

"is\_close": 0 //报名日期是否截止，1表示已经截止，0表示未截止

}

}

2.4 学员报名

* URL: /student/enroll-course
* Method: POST
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| course\_id | 课程id | 是 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : "",//响应描述，code非0时有错误描述

}

2.5 学员取消报名

* URL: /student/drop-out-course
* Method: POST
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| course\_id | 课程id | 是 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : "",//响应描述，code非0时有错误描述

}

2.6 学员签到

* URL: /student/signin-course
* Method: POST
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| course\_id | 课程id | 是 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : "",//响应描述，code非0时有错误描述

}

#### 三.讲师

3.1 我开设的课程列表

* URL: /teacher/my-courses
* Method: GET
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| course\_id | 课程id | 是 |
| page\_size | 每页条数 默认20 | 否 |
| page\_index | 当前页码 默认第一页是1 | 否 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : //响应描述，code非0时有错误描述

"data" :{

"page\_index":1, //页数，表示第几页

"page\_size":20, //表示一页显示的课程数

"page\_total":10,//表示总共有多少页

"list":[

{

"course\_id": 1 ,//课程id号

"title": "课程名称",

"date": "2012-12-12 14:10" //开课日期

}

]

}

}

3.2 查看签到/报名情况

* URL: /teacher/list-students-enroll
* Method: GET
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| course\_id | 课程id | 是 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : "",//响应描述，code非0时有错误描述

"data" :{

"total\_num": 1 ,//报名人数上限

"selected\_num": 1 ,//报名人数

"signin\_num": 1 ,//课程已签到人数

"list" :[

{

"chinese\_name": "" ,//学员中文姓名

"english\_name": "" ,//学员中文姓名

"avator": "" ,//学员头像

"signin\_status": 0 //签到状态,0表示未签到，1表示已签到

}]

}

}

3.3 获取签到二维码

* URL: /teacher/get-qrcode
* Method: GET
* response 输出是一个图片

{

"code" : 0, //状态码，0为成功，1为失败

"message" : "",//响应描述，code非0时有错误描述

"data" :

{

`course\_QR\_code`: //课程二维码

}

}

#### 四.管理员界面

4.1 管理员新建课程

* action: /admin/create-course
* Method: POST
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| title | 课程名称 | 是 |
| staff\_id | 非0表示讲师企业员工id,0表示非企业员工 | 是 |
| teacher\_name | 课程讲师姓名，企业员工必须是英文名 | 是 |
| start\_date | 课程开课日期 | 是 |
| end\_date | 课程开课日期 | 是 |
| nature | 课程性质，0为选修，1为必修 | 是 |
| address | 课程开课地点 | 是 |
| description | 课程简介 | 否 |
| total\_num | 课程可选人数 | 否 |
| category1\_title | 课程一级分类 | 是 |
| category2\_title | 课程二级分类 | 是 |
| comment\_url | 课程评价连接 | 否 |

4.2 查看课程列表

* action: /admin/list-courses
* Method: GET
* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : //响应描述，code非0时有错误描述

"data" :{

"page\_index":1, //页数，表示第几页

"page\_size":20, //表示一页显示的课程数

"page\_total":10,//表示总共有多少页

"list":[

{

"course\_id": 1 ,//课程id号

"title": "课程名称",

"status": 0 //课程上架状态，1表示上架，0表示下架

}

]

}

}

4.3 上架课程

* action: /admin/publish-courses
* Method: POST
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| course\_id | 课程名称 | 是 |

4.4 下架课程

* action: /admin/drop-courses
* Method: POST
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| course\_id | 课程名称 | 是 |

4.5 获取课程信息

* action: /admin/get-course
* Method: GET
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| course\_id | 课程id | 是 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : "" , //响应描述，code非0时有错误描述

"data" :{

"course\_id": 1 ,//课程ID

"title": "" , //课程名称

"staff\_id": "" ,//非0表示讲师企业员工id,0表示非企业员工

"teacher\_name": "" ,//课程讲师姓名，企业员工必须是英文名

"start\_date": "" //课程开课日期Y-m-d H:i

"end\_date": "" //课程下课日期Y-m-d H:i

"nature": "" , //课程性质，0为选修，1为必修

"address": "" , //课程开课地点

"description": "" , //课程简介

"total\_num": "" , //课程可选总人数

"category1\_title": "" , //课程类别一级名称

"category2\_title": "" , //课程类别二级名称

"comment\_url": "" , //课程评价链接

}

}

4.6 修改课程

* action: /admin/change-course
* Method: POST
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| course\_id | 课程id | 是 |
| title | 课程名称 | 是 |
| staff\_id | 讲师是企业员工id,0为非企业员工 | 是 |
| teacher\_name | 讲师姓名，企业员工填写英文名称，非企业员工任意 | 是 |
| start\_date | 课程开课日期 | 是 |
| end\_date | 课程结束日期 | 是 |
| nature | 课程性质，0为选修，1为必修 | 是 |
| adress | 开课地点 | 是 |
| description | 课程简介 | 否 |
| total\_num | 课程可选总人数 | 否 |
| category1\_title | 课程一级分类 | 是 |
| category2\_title | 课程二级分类 | 是 |

4.7 查看签到/报名情况

* URL: /teacher/list-students-enroll
* Method: GET
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| course\_id | 课程id | 是 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : "",//响应描述，code非0时有错误描述

"data" :{

"total\_num": 1 ,//报名人数上限

"selected\_num": 1 ,//报名人数

"signin\_num": 1 ,//课程已签到人数

"list" :[

{

"chinese\_name": "" ,//学员中文姓名

"english\_name": "" ,//学员中文姓名

"avator": "" ,//学员头像

"signin\_status": 0 //签到状态,0表示未签到，1表示已签到

}]

}

}

4.8 获取员工id

* URL: /admin/get-staff-id
* Method: GET
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| teacher\_name | 讲师姓名 | 是 |

* response

{

"code" : 0, //状态码，0为成功，1为失败

"message" : "",//响应描述，code非0时有错误描述

"data" :

{

"staff\_id": 0 //非0为讲师员工id，0表示该员工不存在

}

}

4.9 获取签到二维码

* URL: /teacher/get-qrcode
* Method: GET
* response 输出是一个图片

{

"code" : 0, //状态码，0为成功，1为失败

"message" : "",//响应描述，code非0时有错误描述

"data" :

{

"course\_QR\_code": //课程二维码

}

}

4.10 获取讲师姓名

* URL: /admin/get-teacher-name
* Method: GET
* Query Parameters:

| **字段** | **说明** | **是否必填** |
| --- | --- | --- |
| teacher\_name | 讲师姓名 | 是 |

* response {

"code" : 0, //状态码，0为成功，1为失败

"message" : "",//响应描述，code非0时有错误描述

"data" :[

{

"teacher\_name": "teemo" //匹配的讲师姓名列表 }

]

}

### 5. 数据结构描述

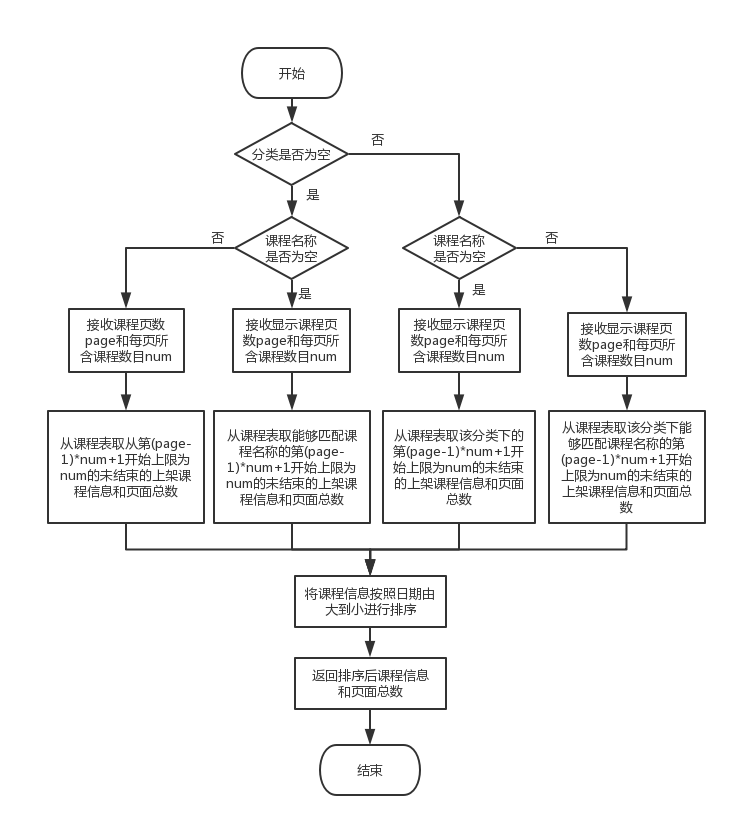
* **本节描述待开发系统的主要数据结构。如果系统比较复杂，那么该节必须具备。特别是存储相关系统，需要描述用什么数据结构来描述存储的内容**

### 6. 主要流程描述

* **本节描述待开发系统内部的主要功能的处理流程。一般来说都采用流程图的方式描述。需要描述出整个流程中什么时候会使用各个模块，最好在流程图中展现（比如入库，入gearman），或者添加额外描述**

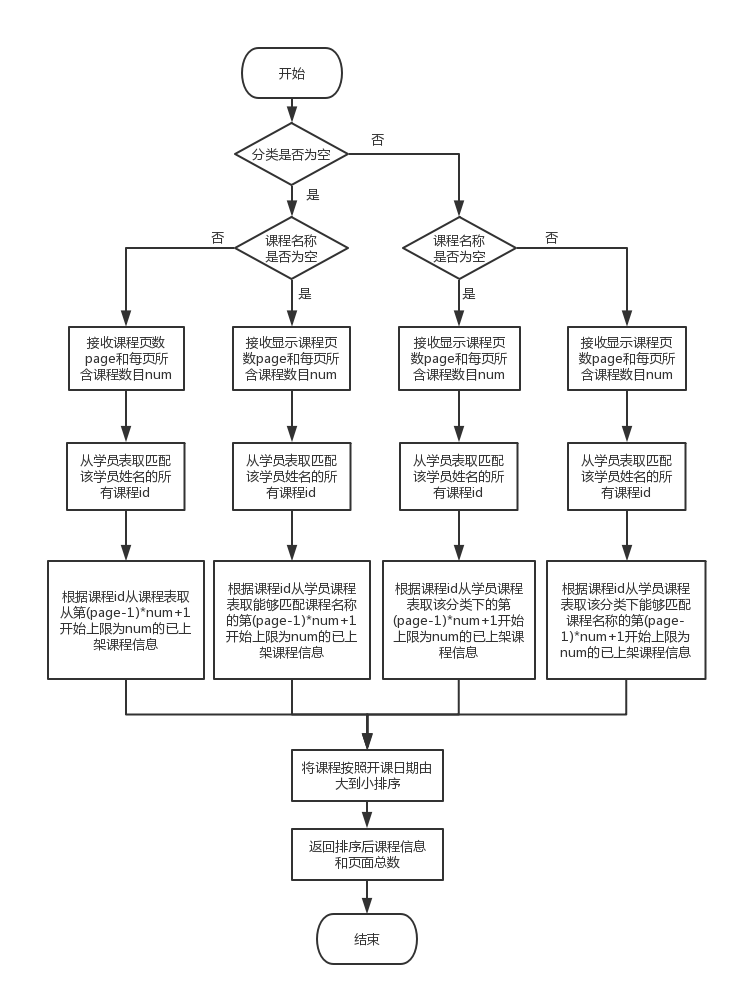
1. **学员**

**1.1查看可选课程列表**



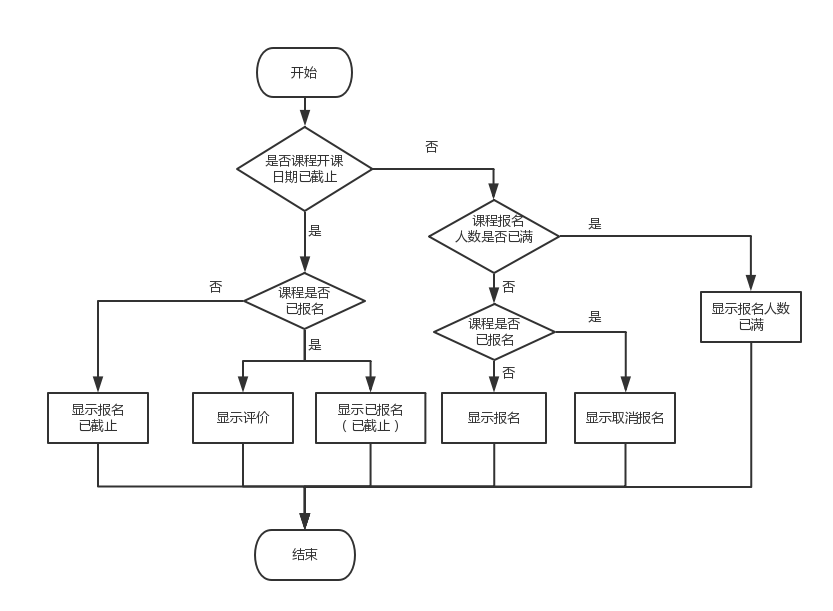
页面总数：课程列表有多少页

**1.2查看已选课程列表**

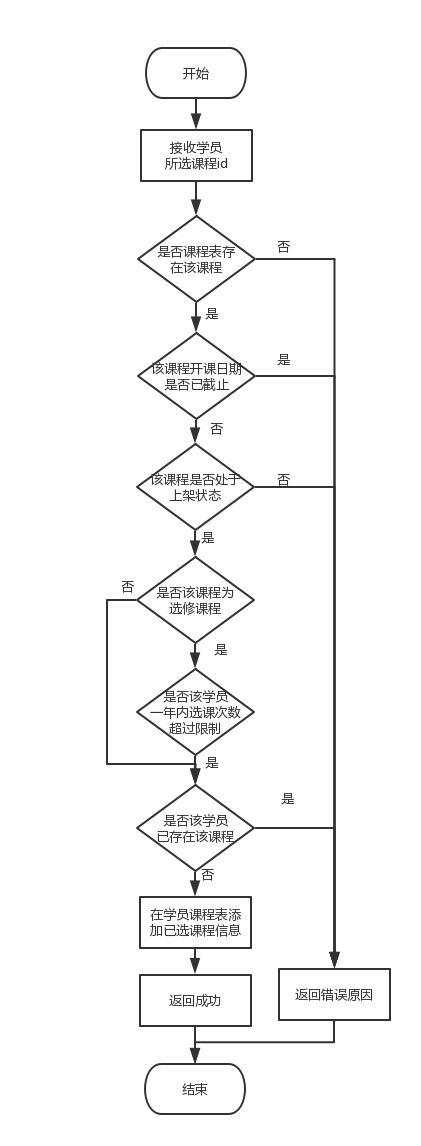


页面总数：课程列表有多少页

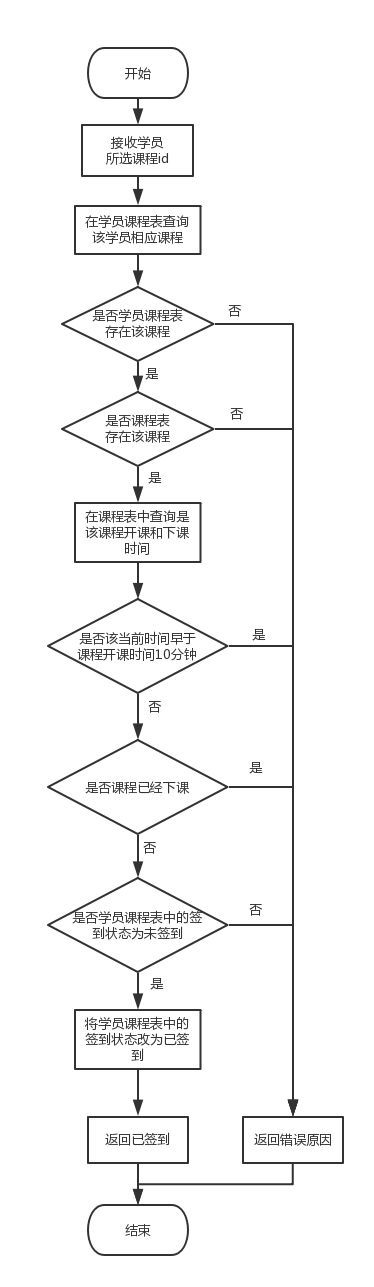
**1.3显示课程详情**

****

**1.4 报名课程**

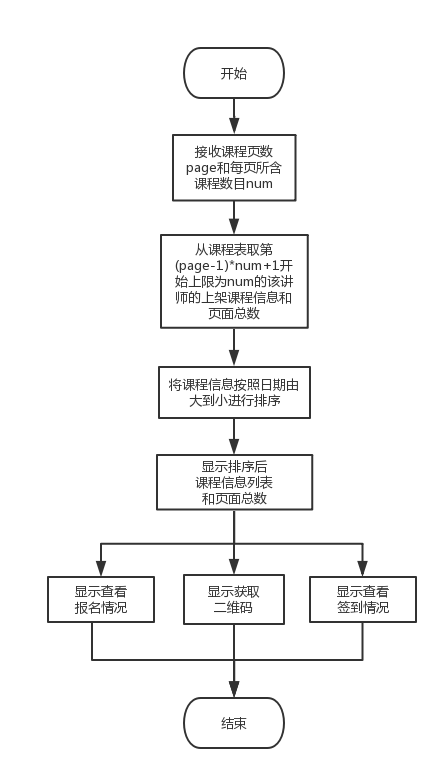
****

**1.5 签到**

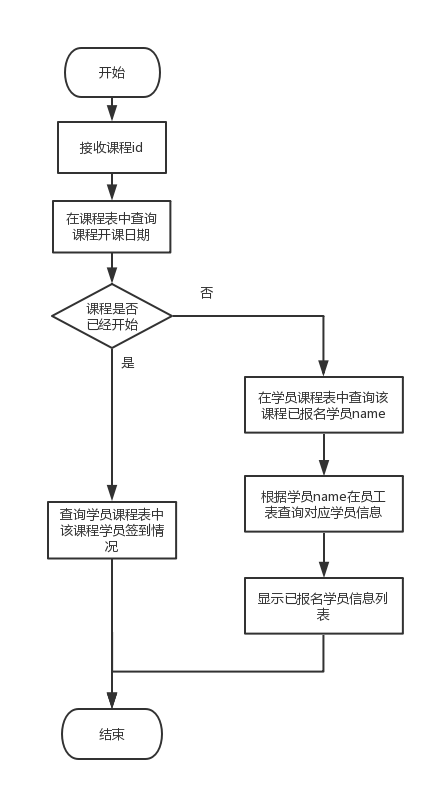
****

二．讲师

2.1 讲师查看课程页面展示

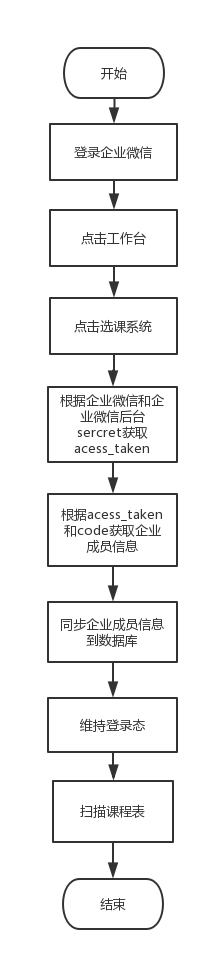


2.2签到/报名页面课程展示

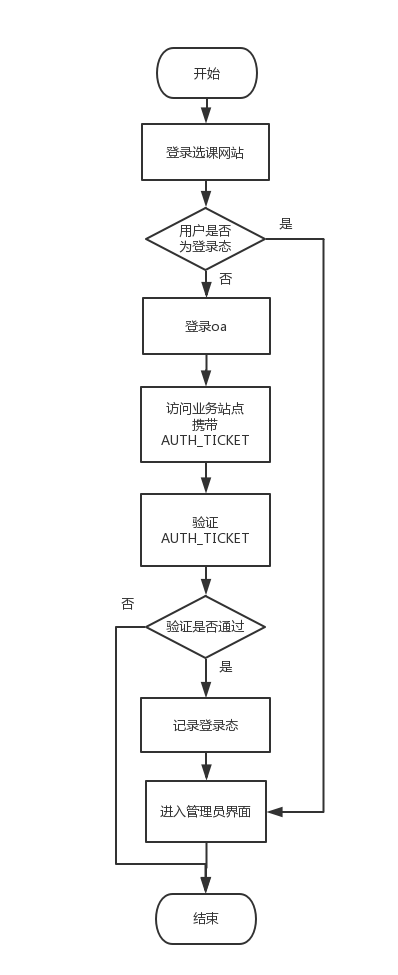


三．企业微信登录

1. 企业微信登录流程展示



1. oa登录



### 7. 数据库表设计

* **本节描述系统需要用到的数据库表结构。请描述是否需要分库分表，分库分表用的key是什么；各个数据库表字段、字段的取值、备注、表的索引、字符集（默认都需要是utf-8）。请直接用建表语句描述，便于后继修改，修改该语句后需要定期同步到本文档。建议所有的字段名用小写和下滑线表示**
* 本系统设计的数据库名称为course
* 本系统涉及到数据库表如下几个

1. 表staff，记录员工信息(员工表)
2. 表student\_course，记录员工作为学员所选的课程（学员课程表）
3. 表course，记录课程详细信息（课程表）
4. 表course\_category，记录课程分类名称（课程分类表）

CREATE TABLE `staff` (

`id` int(10) NOT NULL AUTO\_INCREMENT COMMENT '自增id',

`staff\_id` varchar(30) DEFAULT 0 COMMENT '员工id',

`staff\_name` varchar(30) NOT NULL COMMENT '员工英文姓名',

`chinese\_name` varchar(20) COMMENT '员工中文姓名',

`avatar` varchar(20) COMMENT '员工企业微信头像',

`sex` tinyint COMMENT '员工性别，0代表女，1代表男',

`created\_at` int(10) COMMENT '创建时间',

`updated\_at` int(10) COMMENT '修改时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE= utf8\_general\_ci COMMENT='员工信息';

CREATE TABLE `student\_course` (

`id` int(10) NOT NULL COMMENT ',

`staff\_id` varchar(30) NOT NULL COMMENT '员工id',

`course\_id` int(10) NOT NULL COMMENT '课程id，对应于course表的id值',

`signin\_status` tinyint NULL COMMENT '学员对该课程的签到状态',

`signin\_time` int(10) NULL COMMENT '学员对该课程的签到时间',

`created\_at` int(10) COMMENT '创建时间',

`updated\_at` int(10) COMMENT '修改时间',

PRIMARY KEY (`id`),

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE= utf8\_general\_ci COMMENT='学员已选课程';

CREATE TABLE `course` (

`id` int(10) NOT NULL AUTO\_INCREMENT COMMENT '课程id',

`title` varchar(40) NOT NULL COMMENT '课程名称',

`staff\_id ` tinyint DEFAULT 0 NOT NULL COMMENT '讲师身份，非0为企业员工id，0表示不是企业员工',

`teacher\_name` int(10) NOT NULL COMMENT '课程讲师姓名，企业员工必须使用英文名',

`start\_date` int(10) NOT NULL COMMENT '开课日期Y-m-d H:i',

`end\_date` int(10) NOT NULL COMMENT '下课日期Y-m-d H:i',

`created\_at` int(10) COMMENT '创建时间',

`updated\_at` int(10) COMMENT '修改时间',

`nature` tinyint COMMENT '课程性质，0表示选修，1表示必修',

`description` text NULL COMMENT '课程描述',

`total\_num` int(6) NULL COMMENT '课程可选总人数',

`address` int(6) NULL COMMENT '课程开课地点',

`selected\_num` DEFAULT 0 int(6) NULL COMMENT '课程已选人数',

`parent\_id` int(3) NOT NULL COMMENT '课程类别一级id',

`category\_id` int(3) NOT NULL COMMENT '课程类别二级id',

`comment\_url` varchar(255) NULL COMMENT '课程评价链接',

`status` tinyint DEFAULT 0 NOT NULL COMMENT '课程上下架状态，0为下架，1为上架',

`signin\_num` DEFAULT 0 int(10) NOT NULL COMMENT '课程签到人数',

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE= utf8\_general\_ci COMMENT='课程详细信息';

CREATE TABLE `course\_category` (

`id` int(10) NOT NULL AUTO\_INCREMENT COMMENT '课程类别id',

`title` varchar(20) DEFAULT NULL COMMENT '课程类别名称',

`parent\_id` int(10) DEFAULT 0 COMMENT '课程类别父级id，0表示一级',

`created\_at` int(10) COMMENT '创建时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE= utf8\_general\_ci COMMENT='课程分类名称';

CREATE INDEX idx\_staff\_staff\_id ON staff (staff\_id);

CREATE INDEX idx\_course\_staff\_id ON course (staff\_id);

CREATE INDEX idx\_student\_course\_staff\_id ON student\_course (staff\_id);

### 8. 性能分析

* **本节描述系统性能的各种瓶颈**

### 9. 运维手册

### 10. 性能测试

### 11. 附录