

河海大学物联网工程学院

课程设计报告

题 目	活动考勤签到小程序
专 业	物联网工程
学 号	142062910127
授课班号	6242774-01
学生姓名	曾翔

目录

1	概述	3
1.1	设计背景	3
1.2	设计目的	3
1.3	设计内容与意义	3
2	业务与功能分析	3
2.1	系统的业务流程	3
2.2	数据需求	4
2.3	功能需求	5
3	数据结构设计	5
3.1	逻辑结构设计	5
3.1.1	E-R 模型图转化为关系模式	6
3.1.2	关系模式规范化	6
3.1.3	关系描述	6
4	系统的实现与测试	8
4.1	系统详细设计与实现	8
5	总结与展望	12
6	参考文献	12

1 概述

1.1 设计背景

在生活日益丰富，活动多种多样的今天，人们在参加各项活动的同时不可避免的遇到活动的考勤与签到。传统的签到采用点名或自行签名的方式有着耗时、代签、难以统计等弊端。而如今，随着互联网的普及以及有着人工智能加持的服务，我们的签到系统也可以跟上时代发展的步伐。智能手机的普及，微信已成为大家装机必备的软件之一，深深地影响着人们的生活习惯。而近年来火爆的微信小程序，借助微信这个平台，凭借自身免下载、功能多样、体积小等特点日益流行。因此，借助微信小程序开发的考勤签到系统能够满足活动的发起者、活动的参与者的需求。

1.2 设计目的

通过本签到系统，活动的发起者可以创建活动，并设置活动的标题、内容、时间、活动的地点以及签到打卡的方式，其中包括：图片签到、地点签到、人脸签到。为了用户使用方便，采用免登录的方式，小程序直接获取用户的微信的个人信息。用户可以上传活动的照片，并保存在腾讯云的对象存储中。并且，对于频繁出现的代签、随意签到的情况，采用地点签到要求在活动地点 500m 的范围内才可以签到，而人脸签到要求活动的参与者现场拍一张自己的照片，并利用 Face++ 提供的人脸对比的接口，要求置信度大于 50% 才可以签到成功。因此，采用微信小程序打卡的方式能够方便参与者打卡，方便活动的发起者查看有效的打卡情况，杜绝作弊的发生。

1.3 设计内容与意义

通过本套系统，用户很容易可以管理自己的活动。首先，用户可以搜索当前的所有活动，并支持关键词搜索，用户无需输入活动的全称，对活动进行报名。系统会自动按照已报名的活动是否在规定的时间内进行分类，防止出现遗漏打卡的情况。而在创建活动的时候，系统会根据输入的字词进行联想，方便用户更快地找到活动的地点。勾选不同的打卡方式能更好地适应不同的活动，更加灵活。用户可以查看所有打卡的情况，在督促自己的同时提醒其他活动参与者签到。在智能手机如此普及的今天，我们只需在日常使用的微信上就可以进行活动的签到而无需额外的下载一个 app，增加手机的负担。

2 业务与功能分析

2.1 系统的业务流程

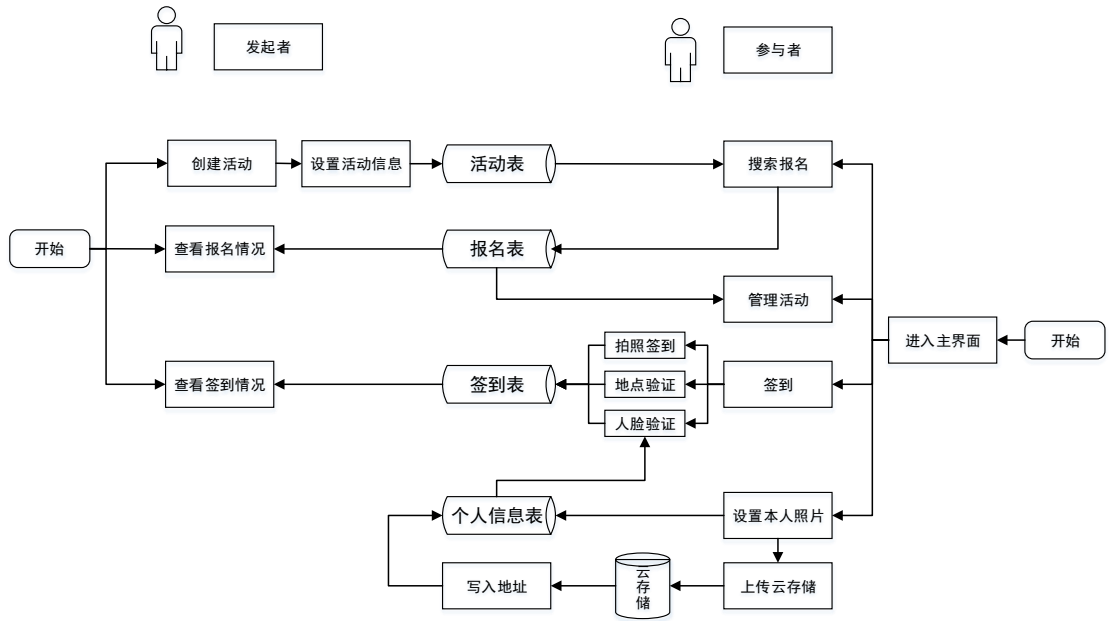
该活动签到小程序主要分为三个部分，创建活动、活动报名、活动签到。活动的创建者

在创建活动之后自动报名参加该活动。用户同时也可以管理自己参加的所有活动。活动的创建包括填写活动主题、填写活动简介，选择活动的起止日期并在地图中搜索并确定活动的地点，系统会记录下具体的经纬度。同时，创建者可以根据自己的需求选择拍照打卡、地点打卡、人脸打卡这几种打卡方式的组合。

其他用户可以通过活动的关键字搜索当前所有的活动，并且报名加入。同时对自己所参加的活动进行管理，根据活动的起止日期分为正在进行中和已经结束。

对于已报名的活动，可以查看活动的详情，活动的报名情况以及大家的打卡的状态。而参与者若要打卡则必须按照活动所要求的打卡方式进行签到包括上传打卡照片，照片会上传并保存在腾讯云存储上，地点的验证以及是否是本人的人脸的判断，只有完成了全部的要求，才能完成签到。

该签到考勤系统的业务流程如下图所示：



考勤系统业务流程图

其中需要注意的是当用户面临需要人脸验证才能签到时，必须先要补全本人的头像，只有当前的照片与个人信息中的照片为同一个人时，才能完成签到。

2.2 数据需求

用户在进入本签到小程序后无需登录，系统会获取用户微信的注册信息，可以直接根据要求直接创建活动，并根据自身活动的需要设置活动的不同签到方式。活动地点的经纬度是根据腾讯地图的 api 获取的，存储在数据库中。

与此同时，参加活动的用户可以搜索并报名参加自己或其他用户所创建的活动，只有报名过的活动才会显示在活动管理界面。活动签到的图片会存储在云存储上，而签到的信息会存放在服务器的数据库中。用户还可以更改个人的头像，同样，为了图片上传和下载速度，将会存放在腾讯云存储上，同时返回的图片地址存放在数据库中。

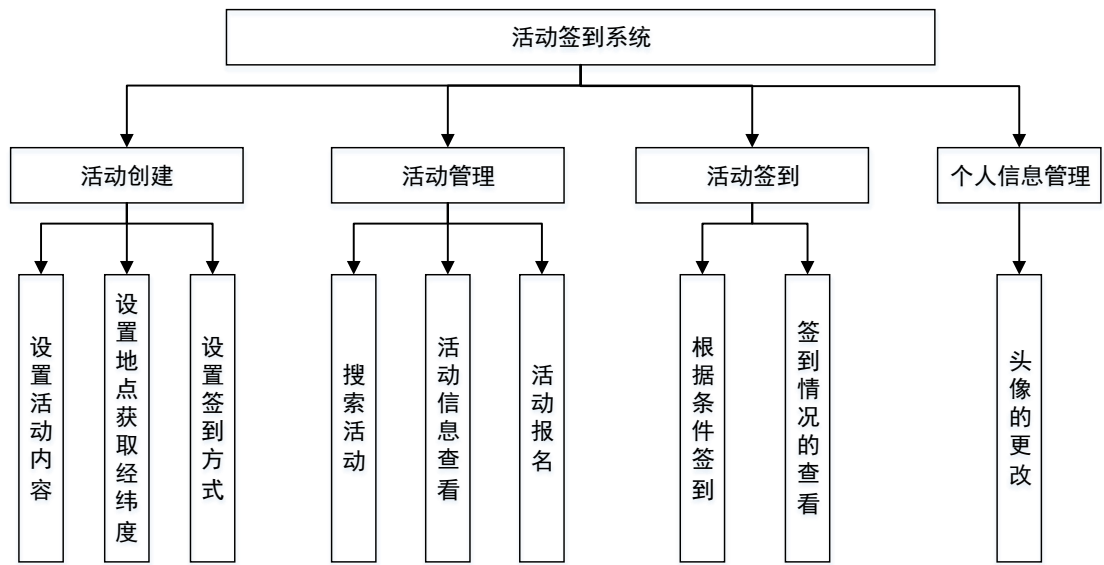
2.3 功能需求

活动签到小程序方便了用户创建活动，并提供了多样化的签到方案，而用户也可以通过关键词的搜索找到自己所要参加的活动，并且根据要求进行签到。基于微信小程序平台，方便了用户获取签到程序，防止了下载多余的客户端造成的麻烦。同时，相较于传统的签到方式的繁琐与耗时，活动的参与者们可以并行得签到，同时设置了地点签到与人脸验证在一定程度上杜绝了作弊情况的发生，确保只有参与活动的本人才能签到成功。同时，签到照片的上传可以帮助活动发起人对于大家参与活动情况的评估，判断参与者的专注程度。

当用户在进行签到遇到需要人脸验证的情况时，必须在个人信息界面补全个人近期照片，系统会根据个人信息照片与当前的照片通过人脸对比服务，判断是否是同一个人。而个人信息的照片可以更改，防止因为面容改变的过大而导致签到失败的情况发生。

活动管理根据所报名活动的起止时间分成正在进行中与已结束，方便用户对活动进行签到，在一定程度上防止用户错过了活动。同时，活动管理能记录用户所有参加的活动。

下图为系统的功能需求图：



功能需求图

3 数据结构设计

3.1 逻辑结构设计

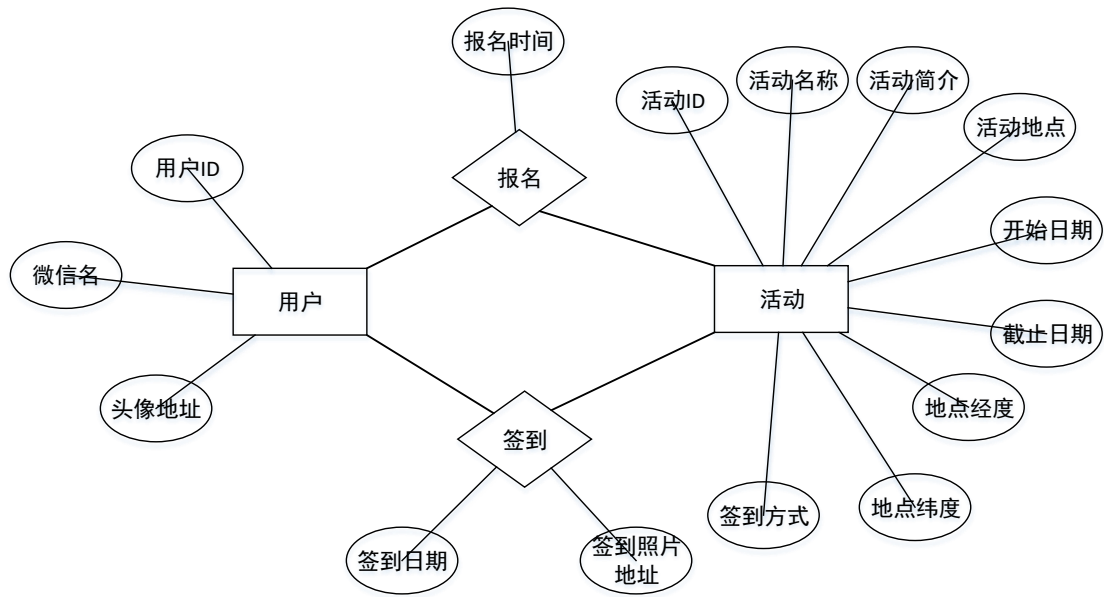
由于该活动签到小程序的功能并不复杂，简单的设置了两个实体，分别是用户和活动。

首先在用户实体中有如下几个属性：微信名、真实照片在云存储上的地址。活动实体包括创建者 ID、活动名称、活动描述、活动地点信息、签到打卡的方式。

一个用户可以报名参加多个活动，于是用户和活动多对多的关系，在活动报名表中设置两个外键分别关联了用户表中的用户的 ID 与活动表中的活动 ID，同时附加产生了报名日期这个属性。

而签到表中同样也是用户与活动多对多的关系，因此也设置了外键关联用户的 ID 和活

动的 ID 同时增加签到图片地址以及签到时间这个两个属性。



E-R 图

3.1.1 E-R 模型图转化为关系模式

用户（用户 ID，微信名，照片地址）

主键为用户 ID

活动（活动 ID，创建者 ID 活动名称，活动简介，活动地点，开始日期，截止日期，活动经度，活动纬度，签到方式）

主键为活动 ID，外键为创建者 ID

报名（报名 ID，活动 ID，用户 ID，报名日期）

主键为报名 ID，外键为用户 ID、活动 ID

签到（签到 ID，活动 ID，用户 ID，签到日期，签到图片地址）

主键为签到 ID，外键为用户 ID、活动 ID

3.1.2 关系模式规范化

首先，在设计数据库关系模式时候，每张表都满足了第一范式的要求即每个属都是不可分割的简单项。在满足了第一范式的要求后，所设计的表中的每一个属性完全依赖于关系 R 的任意一个候选码，在这些表中，自增的 ID 可以唯一的标识非主属性。不存在非主属性对候选码的部分依赖，因此满足了第二范式。同时，我们可以看到我们的表中每一个非主属性都不传递依赖于某一个候选码，满足了第三范式。

3.1.3 关系描述

表 1 用户表

主键	字段名	数据类型	长度	非空	备注
√	F_ID	int	20	√	ID
	F_Name	nvarchar	20		用户微信名
	F_PhotoUrl	nvarchar	200		用户照片地址

表 2 活动表

主键	字段名	数据类型	长度	非空	备注
√	F_ID	int	20	√	活动 ID
	F_CreaterId	int	20	√	创建者用户 ID
	F_Caption	nvarchar	20		活动名称
	F_Desc	nvarchar	200		活动介绍
	F_Location	nvarchar	50		活动地点名称
	F_Lat	float			地点经度
	F_Lng	float			地点纬度
	F_StartDate	date			开始日期
	F_EndDate	date			结束日期
	F_IfFace	int	1		是否人脸验证
	F_IfLocation	int	1		是否地点验证
	F_IfPhoto	int	1		是否图片签到

表 3 报名表

主键	字段名	数据类型	长度	非空	备注
√	F_ID	int	20	√	报名 ID
	F_ApplyerId	int	10	√	报名用户 ID
	F_ActivityId	int	20	√	活动 ID
	F_ApplyDate	date			报名日期

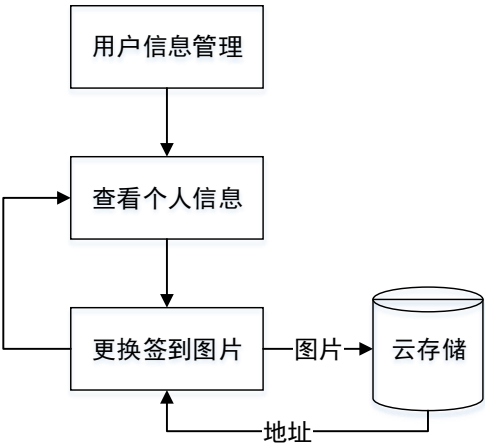
表 4 签到表

主键	字段名	数据类型	长度	非空	备注
√	F_ID	int	20	√	报名 ID
	F_ActivityId	int	10	√	活动用户 ID
	F_UserId	int	20	√	签到用户 ID
	F_CheckDate	date			签到日期
	F_PhotoUrl	nvarchar	200		签到照片地址

4 系统的实现与测试

4.1 系统详细设计与实现

用户信息子系统



用户信息子系统

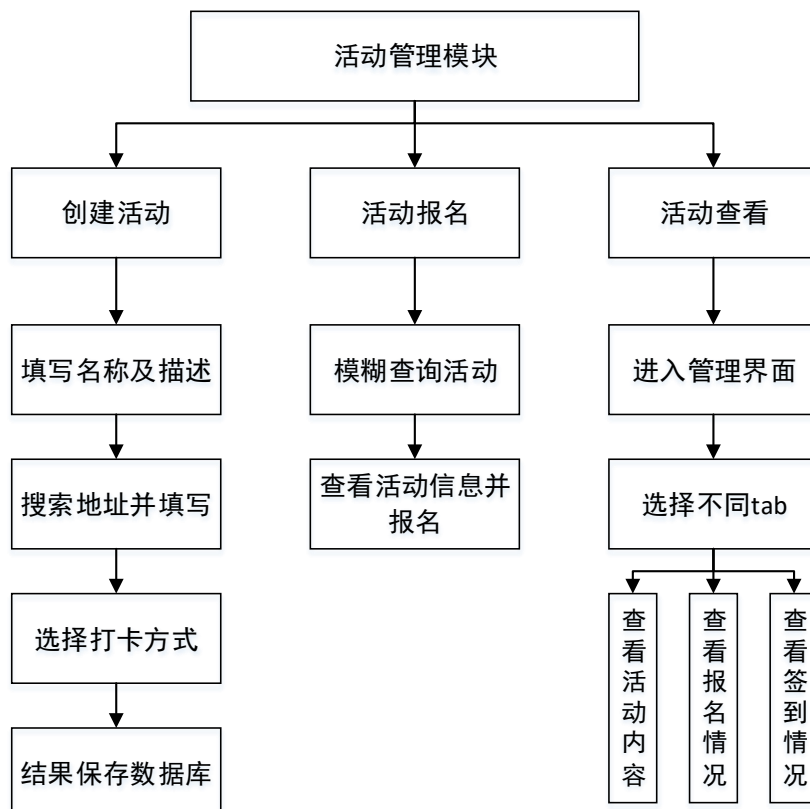
用户初次使用签到小程序时，是没有头像的，需要自己设置一个近期的照片，方便后续人脸的验证的准确性。考虑到图片上传于下载传输速率，我们将图片上传至云存储，将返回的地址存储在数据库中。

界面截图



更改头像界面

活动管理子系统



活动管理子系统

首先，用户可以在创建活动页面填写活动的名称、描述等基本信息，随后点击选择地址按钮，进入地图界面，输入活动地点的关键字。通过地理位置关键字联想推荐的地址，设置活动地点，后台获得活动地点的经纬度，最后选择活动的打卡方式。活动创建完后将

数据写入后台的数据库中。

在搜索界面，用户可以模糊搜索得到相应的活动的列表，选择后进入活动详情界面，点击报名即可完成报名。

活动管理界面分成两个部分，分别是已结束和正在进行中，点击后进入活动的详情界面，可以查看活动的内容、活动的报名情况以及按照日期排序的活动签到情况。

界面截图



填写活动内容



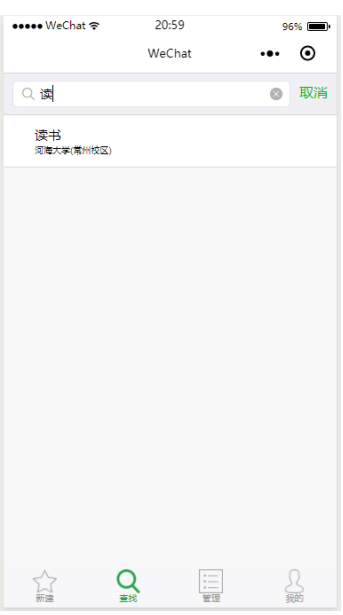
地点搜索



地图标注



选择打卡方式



活动模糊搜索



活动报名



管理-进行中

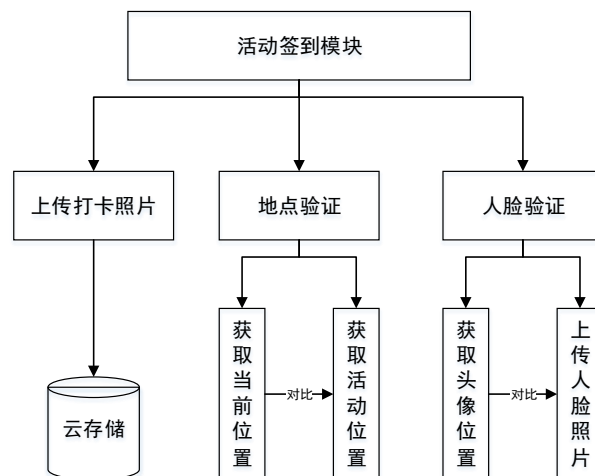


管理-已结束



报名打卡详情

活动签到子系统



活动签到子系统

在活动签到模块中，用户需要根据活动创建者所设定的签到条件进行完成签到工作。只有当所有的条件都得到满足后，签到才能成功，结果将会写入数据库中，并且打卡的照片也会上传至云存储中。

系统截图



活动签到



签到结果显示

5 总结与展望

本次企业综合实践的课程设计要求我们根据微信小程序开发者文档在较短的时间内实现一个完整的小程序。通过接入后台，云存储以及第三方的接口和人脸对比的智能服务，使得我们的小程序在功能上更加的完善。小程序的后台采用的是 Node.js 并采用 Express4.0 框架，响应客户端的请求，图片的上传于下载则是利用了腾讯云的对象存储，在上传之前还搭建了鉴权服务器对用户身份进行了识别。在设置活动地点时，则采用了腾讯地图的 api 的地理信息提示接口，方便用户快速选择活动地点。而在签到时，获取当前经纬度和活动地点经纬度进行对比，计算距离。人脸的验证使用的是 Face++ 的人脸对比接口，当两张图片的人脸对比结果的置信度大于 50% 时可以确定是本人签到。

由于时间紧，签到小程序的界面并不是特别美观，而调用 face++ 的免费的人脸识别接口经常会出现 API 并发控制异常，导致人脸验证失败。在软件编写的过程中，出现了数据库中的表跟不上系统的要求，从而临时更改数据库的现象的发生。这一插曲导致大量的代码需要修改，大大得增加了工作量。

希望在以后的工作中，在软件开始编写之前需要明确的了解系统目标，摸清系统的需求后再着手对系统进行架构以及代码的编写。只有这样才能事半功倍。

6 参考文献

- [1]. 一起学 Nodejs <https://github.com/nswbmw/N-blog>
- [2]. Ceph 中国社区 著, Ceph 分布式存储实战, 机械工业出版社, 2016
- [3]. 朴灵 著, 深入浅出 Node.js, 人民邮电出版社, 2013
- [4]. 巴塞特 (Lindsay Bassett) 著; 魏嘉汛 译, JSON 必知必会, 人民邮电出版社, 2016
- [5]. 小程序开发者文档 <https://mp.weixin.qq.com/debug/wxadoc/dev/api/we-run.html>