

# 重庆师范大学

## 全日制本科生毕业设计

题    目： 多排队进度查询微信小程序

学    院： 计算机与信息科学学院

专业年级： 计算机科学与技术 2016 级

学生姓名： 王 振 学号： 2016051603105

指导教师： 杨 有 职称： 副教授

2020 年 4 月 13 日

# 多排队进度查询微信小程序

计算机与信息科学学院 计算机科学与技术专业 2016 级 王振

指导教师 杨有

**摘 要:** 为了解决人们排队等待造成时间浪费的问题, 一款基于微信小程序的多排队进度查询程序应运而生。本文首先介绍了排队小程序运用的理论与技术, 比如微信小程序平台、编程框架 Vue.js 以及基于该基础上的 uni-app 开发框架、uniCloud 云开发平台, 还有常用于前端后台通信的 AJAX 技术, 其次, 从经济、技术以及用户操作三个方面说明了排队小程序开发完全可行, 接着从三种小程序使用者充当的角色的角度介绍了排队小程序明确的需求, 在此基础上开始进行功能设计和数据库设计, 然后展示了前端实现效果, 也介绍了后台实现的具体操作, 最后进行部署测试工作。一系列流程完全规范、合理, 论证了排队小程序的工具性、实用性。

**关键词:** 排队; 微信小程序; 云平台

**Abstract:** In order to solve the problem of waiting in line for time waste, a multi-queue scheduling query program based on WeChat small program is created. This paper also introduces the ajax technology used in front of the front-end communication, which illustrates the complete feasibility of this program development, and then introduces the need for the small sequence of the three kinds of small sequence users, which is based on the design and number of functional design and number, and then shows the implementation effect of the front-end, and introduces the implementation effect of the front-end, and introduces the concrete operation of the background implementation, and the final deployment test work. A series of processes are completely standardized and reasonable, and the instrumental and practicability of the program are demonstrated.

**Key words:** queue; WeChat small program; cloud platform

在二十一世纪, 各行各业蓬勃发展, 人们的工作和生活节奏加快。在这样的

背景下，人们更加注重工作的效率和生活的充实度，不希望把时间花在无关紧要的事情上。另一方面，排队等待是生活中常见的现象，而又与上述愿望相背离，因此需要一款能够告知人们队列进展的程序。而网络上已有的排队通知程序中，多是与银行、医院或者餐馆等实体业务相关联的，以下介绍的是一款基于微信小程序开发的通用的多排队进度查询程序（简称排队小程序）。

## 1 相关理论与技术

排队小程序是基于微信小程序平台开发的，前端技术使用了基于 `vue.js` 的 `uni-app` 开发框架，后台技术依托于 `uniCloud` 云开发平台，其中前端后台交互时大量使用了 `AJAX` 技术。接下来将进行详细介绍。

微信小程序依赖于微信，只要用户的手机或电脑上安装了微信，就可以使用微信小程序。而微信本身有着庞大的用户规模。2020 年 1 月，微信官方公布的数据显示，微信月活跃账户数达 11.51 亿，比去年同期增长了 6%，小程序日活跃用户已经超过 3 亿，2019 年全年累计创造的交易额超过 8000 亿元；人均访问小程序的次数上涨 45%，人均使用小程序的个数上涨 98%。基于这样的事实，选择微信小程序平台可以保证排队小程序能够服务大多数人。再者，基于微信小程序开发可以有效减少开发成本。受益于微信底层封装，开发者不必为了兼容 `Android` 端和 `iOS` 端分别开发出两套程序。此外，微信小程序最大的特点就是随扫随用、用完即走，不会给用户的设备带来存储压力，对于使用低端设备的广大群众较为友好。而工具性是排队小程序的明显特征，人们没有必要将排队小程序长久留存。

`Vue.js` 是用来构建用户界面的框架，与其他大型框架相比，由于其核心库只关注视图层，所以更加容易上手，也便于与既有项目或者其他库资源整合。具体而言，`Vue.js` 的核心是能够进行声明式渲染，也就是通过采用简洁的模板语法，将希望展现的数据渲染进 `DOM`（文档对象模型），当数据发生改变的时候，`DOM` 的内容即刻更新，反过来，`DOM` 内容的更新也会使数据发生改变，这被称为

“MVVM”模式，给开发者提供了极大的便利。在以往的编程实践中，大多数前端开发者使用的是 JQuery 库，它并不是响应式的，只是兼容了多种浏览器，统一了操作 DOM 的方式，数据改变的时候，JQuery 并不会自动更新渲染。当然，对于 IE 9 及其以上版本，Vue 同样兼容。排队小程序发布在微信小程序平台，依托于现代浏览器内核，有较为稳定的更新计划，不会受到该因素影响。

uni-app 是一个使用 Vue.js 开发应用的前端框架，开发者只需要编写一次代码，就可以发布到 Android、iOS、H5 网页以及各种软件的小程序（比如微信、支付宝）等多个平台。同微信小程序一样，对开发者而言，便利性、开发成本低是其明显优势。正因如此，大量的开发者使用 uni-app 进行程序开发，目前已有几十万个应用案例、数千款 uni-app 插件。对比其他跨端框架，比如 Taro，社区生态具有极大优势。

uniCloud 是为使用 uni-app 的开发者提供的基于 serverless 模式和 JS 编程技术的云开发平台。小程序云开发当前已相当流行，微信、支付宝、百度等平台的小程序均提供了云开发，但是都局限于各自的平台，不能做到跨平台开发。uniCloud 正好解决了这个问题，让 uni-app 的所有端应用均能使用。而且目前为止，uniCloud 是免费提供的，并且 DCloud 公司承诺：即使以后收费，也会低于传统云主机的租赁费用。uniCloud 提供的功能包括云函数、数据库、存储和 CDN。其中云函数就是在云端运行的 JS 代码，因为是定制的 node 环境，所以性能和功能有保障。而且 serverless 环境无需自行加购服务器处理增容，开发者可以专注于业务逻辑的处理。数据库依托于阿里云或腾讯云，采用的是基于 NOSQL 的 MongoDB。该数据库以 JSON 格式存储，对前端工程师更友好，不需要专门学习 SQL。但是也有不足，比如 SQL 不如关系数据库丰富。由于排队小程序属于轻量级应用，无需过于强大而又完善的数据库。

AJAX 是一种能在不刷新整个页面的前提下更新部分网页内容的技术。使用本技术的好处有二：一是减轻服务器压力，因为请求的数据并非整个网页，服务器不必处理非请求目标的数据；二是给用户良好的体验，因为不需要刷新整个页

面，就不会导致网页瞬间空白，同时正是因为传输数据量较少，Web 应用后台能更快地返回给用户。虽然该技术同样有不足之处，主要在于部分老旧浏览器的支持程度不够，但是排队小程序依托于微信小程序，不会受到该问题的困扰。

## 2 系统分析

可行性分析和需求分析是软件设计的基础。排队小程序希望解决的问题是能够通知排队成员在某个时刻的队列人数或自身排名，而且希望人人都能创建队列、加入或者仅关注其他人创建的队列。下面进行详细介绍。

### 2.1 可行性分析

可行性分析是需求分析的重要环节，人们可以随心所欲地想，行动却要脚踏实地。下面将从经济可行性、技术可行性以及操作可行性方面进行介绍。

在经济可行性方面，排队小程序建立在上述平台与技术基础之上。根据以上介绍，微信小程序平台可以兼容 Android 和 iOS 两大移动操作系统，uniapp 开发框架使用了易上手的 Vue.js 做基础，uniCloud 则充当服务器功能，开发者不必单独购买云服务器，这一切都有助于降低开发成本，对排队小程序而言，无需过高的经济成本即可投入开发。

在技术可行性方面，基于 Vue.js 建立的 uni-app 框架确保程序设计不至于太过繁琐，而且其广泛的生态环境可以确保开发者遇到问题后能够及时得到有效解决；uniCloud 云开发平台确保了后台系统能够持久而又稳定地运行，AJAX 技术是前端与后台沟通的桥梁，正是借助于成熟的 AJAX 技术，才能实现对用户友好的系统交互。

在操作可行性方面，排队小程序的前端界面与市面上常见的移动应用程序较为相似，没有过高的学习成本。而且借助于微信小程序平台，用户只需要打开安装了微信的手机或者电脑即可使用排队小程序，无需去应用商店下载，另外排队小程序体积小不会占用太多存储空间，对用户而言没有过高的门槛，用户操作完

全可行。

## 2.2 需求分析

排队小程序涉及到三种角色，一种是自身创建了队列，充当管理员角色，第二种是加入了其他队列，成为其他队列的成员，第三种是并不加入其他人的队列，而只是关注目标队列总人数的变化。接下来将分这三种角色进行详细介绍。

**普通用户：**任何使用排队小程序的用户起初都算是普通用户。但是必须登录了才有创建、关注以及加入队列的操作权限。如果用户点击关注队列，他并不会成为该队列的一员，只是会在队列人数达到设定值时发送通知。用户在“我关注的队列”界面可以看到自己关注的所有队列以及相应队列当前的人数，也可以随时取消对队列的关注。如果用户发现队列描述或用途等违反法律法规，也可以进行举报。普通用户用例图如图 2-1 所示。

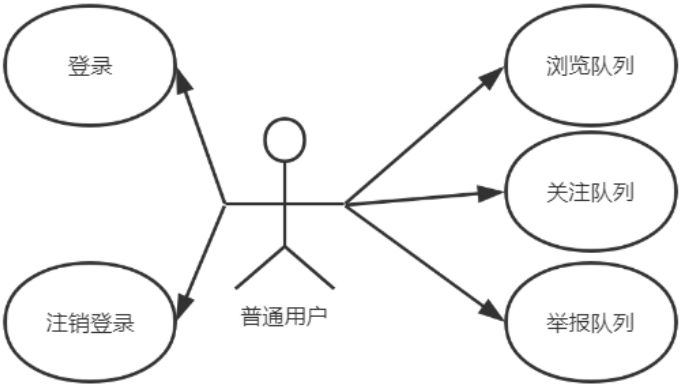


图 2-1 普通用户用例图

**队列管理员：**当用户发布队列以后，就成为了该队列的管理员。管理员用例图如图 2-2 所示。在“我创建的队列”页面可以查看到自己创建的所有队列以及相应队列的总人数。之后对该队列有以下操作：（1）决定是否停止排队。可能一些队列不适合太多人进入，或者管理员的精力有限，无法将队列中的成员一一叫号，这样就要视情况决定是否拒绝新成员加入。（2）修改队列信息。可能管理员在发布队列之后觉得信息不够完善，可以重新编辑，此时创建队列的时间也会更新。（3）成员管理，也就是叫号功能，这是整个程序的核心，当管理员叫号并处

理完成之后，队列人数就会发生变动，队列中每个成员的名次相应发生改变。(4)  
删除队列。管理员随时有权删除自己发布的队列，删除之后任何人都将无法检索到该队列。

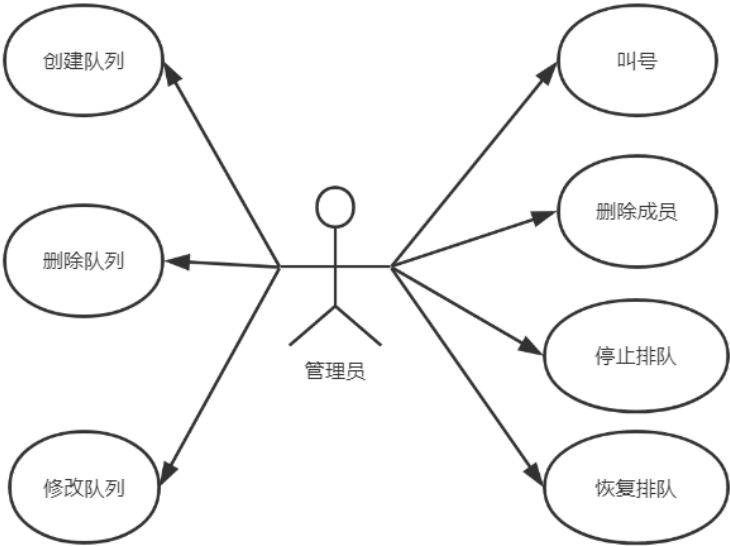


图 2-2 管理员用例图

队列成员：当用户加入队列的时候会提示“当排在第几位时发出通知”，用户输入一个数字点击提交之后，就成为了该队列的成员。这时，用户可以在“我加入的队列”页面查看到自己加入的所有队列，以及在相应队列自身的排名。此时用户可以退出队列。队列成员用例图如图 2-3 所示。

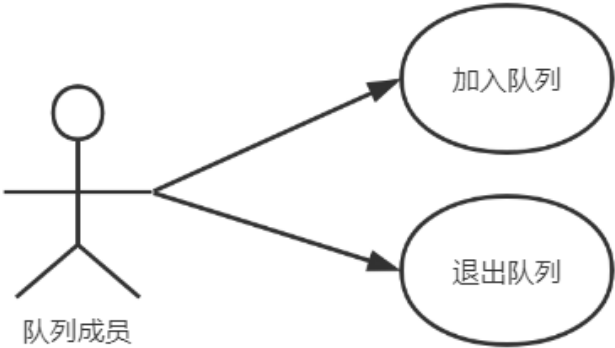


图 2-3 队列成员用例图

### 3 系统设计

一般来说，一款网络应用包括前端和后台两部分，而微信小程序也不例外。在前端方面，包括逻辑处理和视图显示两部分，在后台方面，包括逻辑处理和数据库两部分。前端后台又是通过网络层这个“桥梁”进行连接的，用户首先发起触摸、输入等事件，前端逻辑层进行判断和处理，满足条件即会将数据传给后台，后台又进行数据处理，比如骚扰拦截、数据库的增删查改等，最后将结果反馈给前端，前端经过处理之后展示给用户。系统总体流程如图 3-1 所示。接下来将分功能设计、数据库设计进行阐述。



图 3-1 系统交互流程图

#### 3.1 功能设计

排队小程序主要功能有：登录注册、查看队列信息、创建队列、加入队列、关注队列、叫号、退出加入队列、取消关注队列、发送队列变动通知。接下来将重点介绍登录注册、创建队列、查看队列信息、加入和关注队列和叫号这几大功能。

（1）登录注册。微信官方要求，用户进入小程序时不能强制登录。排队小程序在用户要进行创建队列、加入或关注队列等操作时，就会提示“请登录”。登录应当采用微信授权方式，微信后台天然鉴权，不必担心信息错误。如果是第一次登录，相当于自动注册，对用户而言十分便捷。

（2）创建队列。在首页应当设置一个加号形状的按钮，用户点击以后，就可以跳转到创建队列的页面，提交队列的基本信息，包括队列名称、队列详情（可上传图片）、取号数上限、开始排队时间、是否仅支持完整 ID 搜索、以及可选择设置加入队列提交的验证问题。



(3) 查看队列信息。用户进入小程序首页以后，就能看到一些队列的基本信息，比如队列 ID 编号、开始排队时间、队列名称、队列详情、创建者和创建时间等，当用户点击队列卡片进入队列详情页面时，就应当展现队列所有信息，比如图片、验证问题和队列历史（比如某人加入或退出队列、管理员变更队列状态等）。

(4) 加入和关注队列。用户可加入其他人创建的队列，如果没有设置验证信息，当用户点击“加入队列”按钮以后，出现一个提示框，提示输入“当您排在第几位时通知您”，用户输入并点击确定以后就自动加入，否则将验证信息提交给管理员审核，审核通过方能加入。关注队列也是同理，但是无需验证，关注成功时队列人数也不增加。

(5) 叫号。当有人加入队列以后，队列管理员就可以开始叫号，队列第一个人的状态发生改变，由“在队列中”变更为“正在处理”，当处理完成后，管理员点击“结束”按钮，该用户就出队，队列总人数发生改变，这时应当通知设定了在此时通知的加入或关注队列的用户。

## 3.2 数据库概念设计

排队小程序需要数据库支持，用数据库来保存用户、队列等数据信息。数据库概念设计包括设计各个实体以及实体之间的联系。一个合理且规范的数据库至少要满足第三范式，本数据库的概念设计充分考虑到了函数依赖，每张表中的所有属性都不依赖于其它非主属性。

### 3.2.1 数据库实体

根据上述功能需求分析一节可知，排队小程序以用户为中心，以队列管理为重点。总共有用户、队列、队列历史、通知消息、举报信息等几大实体。

(1) 用户实体。用户实体应当包括用户名、昵称、密码、头像、token 及其过期时间这几个属性，其中用户名必须唯一，是用户表的主键。用户实体及其属性如图 3-2 所示。

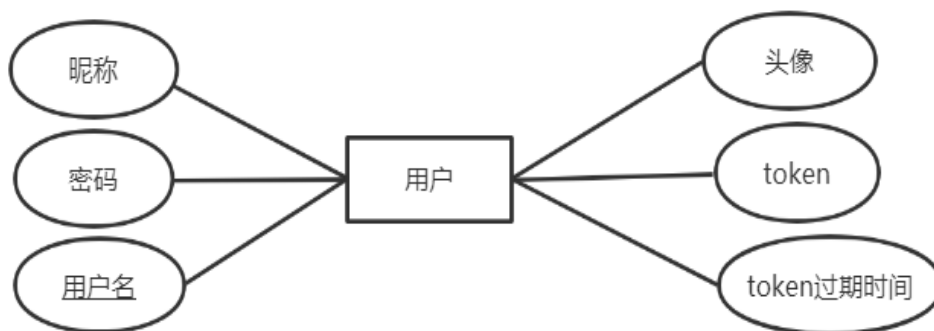


图 3-2 用户实体属性图

（2）队列实体。队列实体包括队列编号、创建时间、队列名、队列描述、队列图片的 URL 地址、加入队列的条件、队列最大取号数、队列当前取号数、允许开始排队的时间、队列当前状态（是否正常排队，或者已被管理员停止，或者满员）、是否仅允许被完整 ID 搜索找到以及创建者 ID 等属性，其中队列编号必须唯一，是队列表的主键。队列实体及其属性如图 3-3 所示。

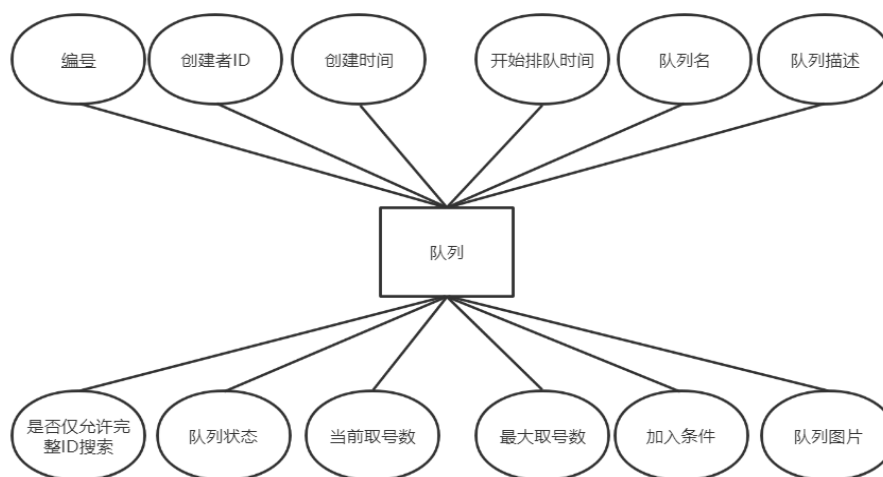


图 3-3 队列实体属性图

（3）队列历史实体。队列历史实体包括编号、队列 ID 号、事件发生时间、发起事件的用户 ID 以及事件类型（创建队列、停止排队、继续排队、加入队列、退出队列等），其中编号必须唯一，是队列历史数据表的主键。队列历史及其属性如图 3-4 所示。

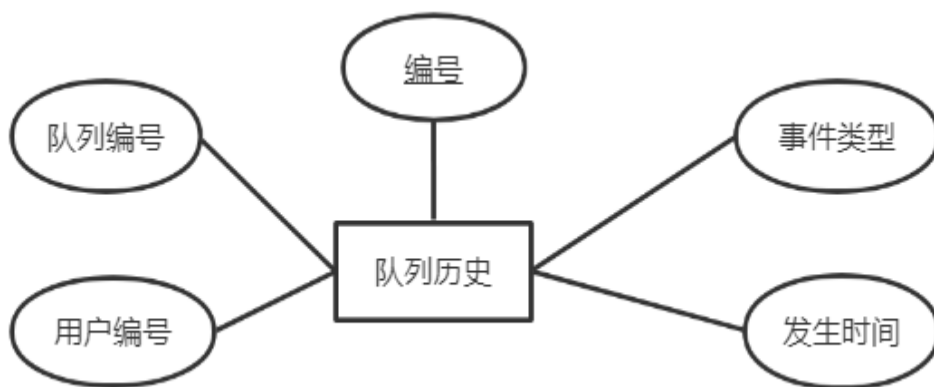


图 3-4 队列历史实体属性图

(4) 通知消息实体。通知消息实体包括消息编号、发送者 ID、接收者 ID、消息内容、发送时间，以及查看详情的链接，其中消息编号必须唯一，是消息表的主键。通知消息实体及其属性如图 3-5 所示。

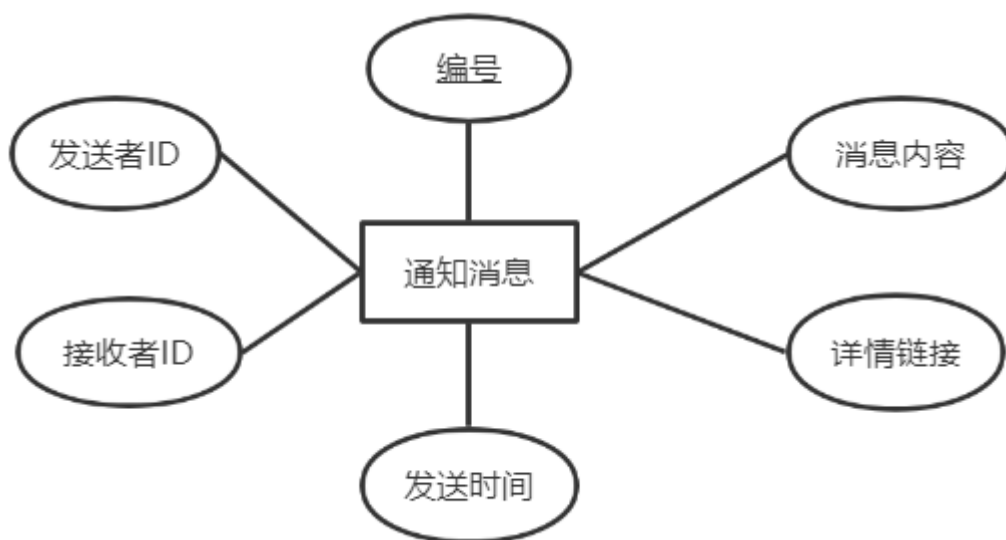


图 3-5 通知消息实体

(5) 举报信息实体。举报信息实体包括编号、举报人 ID、被举报队列 ID、举报内容以及举报时间，其中编号必须唯一，是举报信息表的主键。举报信息实体及其属性如图 3-6 所示。

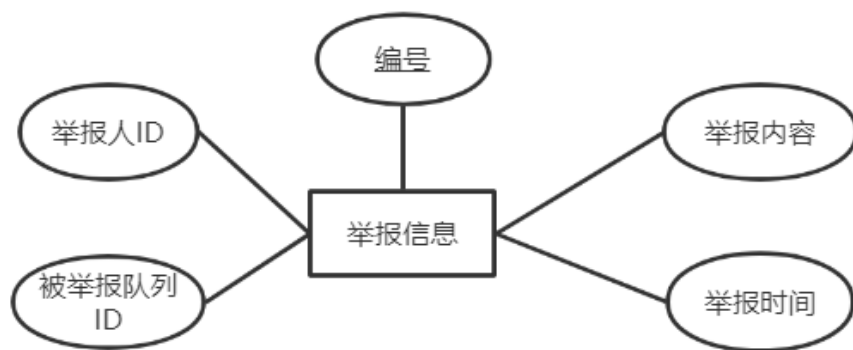


图 3-6 举报信息实体属性图

根据上述需求分析以及数据库实体的设计，排队小程序主要是用户对队列的操作，由此形成的实体-联系图如图 3-7 所示。其中包括用户实体和队列实体以及关注、发布、举报三种多对多联系。用户实体的属性主要有用户名、昵称等属性，队列实体的属性主要有队列编号、创建时间、队列名、队列描述等属性，用户可以关注队列，这时会衍生出关注时间和发通知时的排队人数两个联系属性；用户也可以加入队列，这时会衍生出加入队列的时间、取号的编号、发通知时的排名三个联系属性；用户还可以举报队列，这时会衍生出举报时间和举报内容两个联系属性。

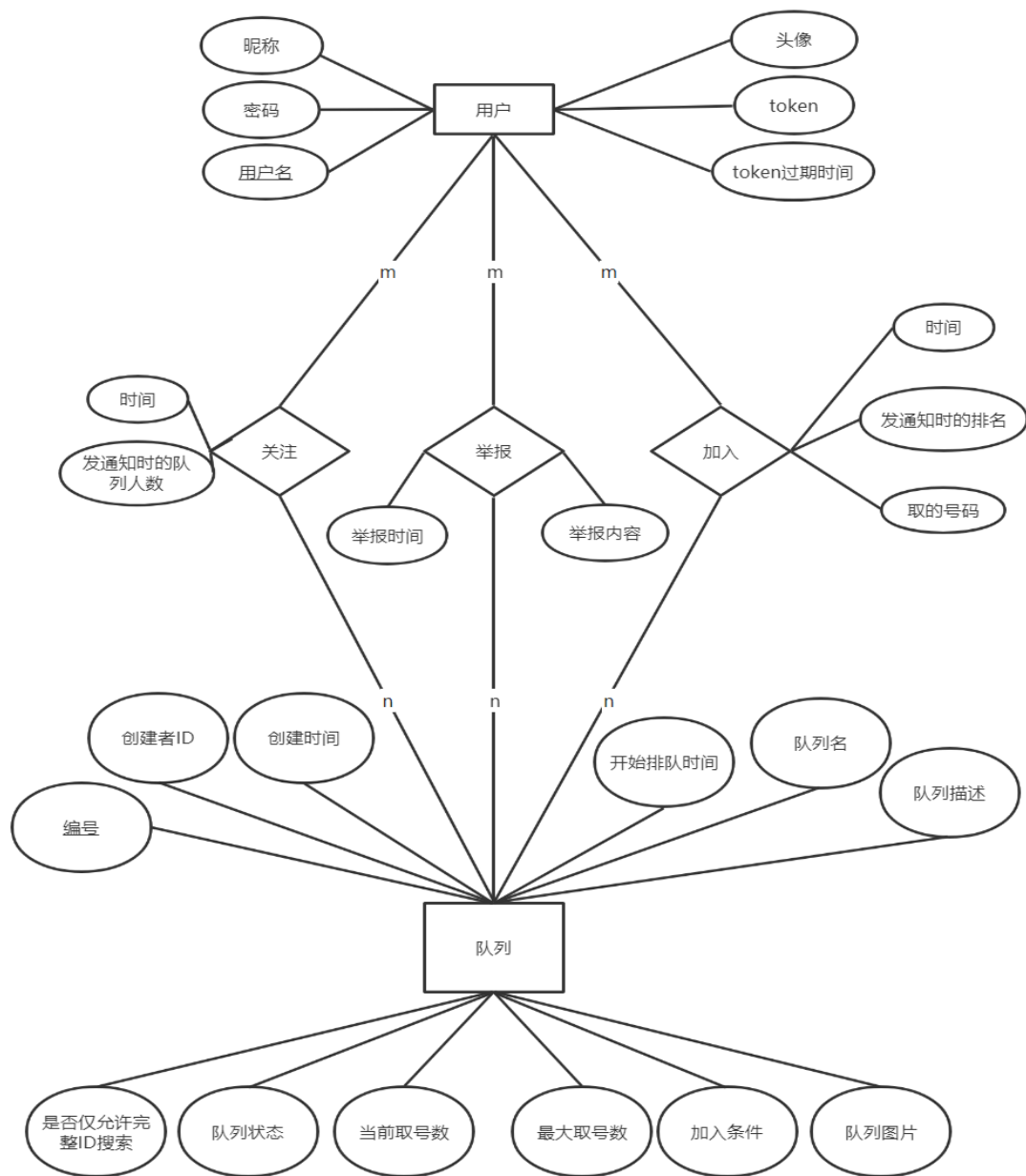


图 3-7 实体-联系图

### 3.2.2 数据字典

要设计一个合理又高效的数据库，数据类型的定义、主键和外键的设置必不可少。排队小程序共有 9 张数据表，接下来将对每张表的字段进行详细说明。

(1) 用户信息表：用来存储用户信息，其中包括 `userId`、`avatar`、`name`、`password`、`token`、`tokenExp` 几个字段，分别表示用户唯一标识、头像、昵称、密

码、会话标识以及会话过期时间，其中 `userId` 是该表的主键。详见表 3.1:

表 3.1 用户信息表

字段名	类型	长度	允许为空	外键	主键	备注
userId	varchar	255	否	否	是	用户唯一标识
avatar	varchar	255	否	否	否	用户头像
name	char	20	否	否	否	用户昵称
password	char	20	否	否	否	密码
token	varchar	255	否	否	否	会话标识
tokenExp	int	13	否	否	否	会话过期时间

(2) 队列信息表: 用来存储队列信息, 包括 `queueId`、`creatorId`、`createTime`、`startTime`、`title`、`description`、`imgList`、`joinCondition`、`maxPeopleCount`、`number`、`state`、`onlySearch` 这些字段, 分别表示队列编号、创建者 ID、创建时间、开始排队时间、队列标题、队列描述、队列图片、加入条件、最大取号数、当前取号数、队列状态和是否仅支持完整 ID 搜索, 其中 `queueId` 是该表的主键。详见表 3.2:

表 3.2 队列信息表

字段名	类型	长度	可空	外键	主键	备注
queueId	int	13	否	否	是	队列编号
creatorId	varchar	255	否	否	否	创建者 ID
createTime	int	13	否	否	否	创建时间
startTime	Int	13	否	否	否	开始排队时间
title	char	20	否	否	否	队列标题
description	varchar	255	是	否	否	队列描述
imgList	varchar	255	是	否	否	队列图片
joinCondition	varchar	255	是	否	否	加入条件
maxPeopleCount	int	3	否	否	否	最大取号数

number	int	3	否	否	否	当前取号数
state	enum	255	否	否	否	正常排队、满员、已停止、已删除
onlySearch	bool	1	否	否	否	是否仅支持完整 ID 搜索

(3) 队列问题回答记录表：存放所有加入需要验证信息的人对队列问题的回答，包括 id、queueId、userId、time、answer、state 等字段，分别表示记录的编号、队列 ID、用户 ID、提交回答时间、答案、表明队列管理员是否已处理的标志，其中 id 是该表的主键，queueId 和 userId 是该表的外键，详见表 3.3：

表 3.3 队列问题回答记录表

字段名	类型	长度	可空	外键	主键	备注
id	int	13	否	否	是	唯一编号
queueId	int	13	否	是	否	队列 ID
userId	varchar	255	否	是	否	用户 ID
time	int	13	否	否	否	提交回答时间
answer	varchar	255	否	否	否	答案
state	enum	255	否	否	否	表明队列管理员是否已处理

(4) 加入队列记录：存放所有用户加入队列的记录，包括 queueId、userId、time、state、sendable、number、joinNotifyNumber 等字段，分别表示队列 ID、用户 ID、加入队列时间、成员状态、是否允许接收推送通知、取号的号码、排在哪个名次时推送通知，其中 queueId 和 userId 既是主键又是外键。详见表 3.4：

表 3.4 加入队列记录表

字段名	类型	长度	可空	外键	主键	备注
queueId	int	13	否	是	是	队列 ID
userId	varchar	255	否	是	是	用户 ID
time	int	13	否	否	否	加入队列时间
state	enum	255	否	否	否	在排队、在处理、已出队

sendable	bool	1	否	否	否	允许接收推送通知
number	int	3	否	否	否	取号的号码
joinNotifyNumber	int	3	否	否	否	排在哪个名次时推送通知

(5) 关注队列记录表: 存放所有用户关注队列的记录, 包括 `queueId`、`userId`、`time`、`state`、`sendable`、`watchNotifyNumber` 几个字段, 分别表示队列 ID、用户 ID、关注队列时间、是否已经发过通知、是否允许接收推送通知以及队列总人数为多少时推送通知, 其中 `queueId` 和 `userId` 既是主键又是外键。详见表 3.5:

表 3.5 关注队列记录表

字段名	类型	长度	可空	外键	主键	备注
queueId	int	13	否	是	是	队列 ID
userId	varchar	255	否	是	是	用户 ID
time	int	13	否	否	否	关注队列时间
state	enum	255	否	否	否	是否已经发过通知
sendable	bool	1	否	否	否	是否允许接收推送通知
watchNotifyNumber	int	3	否	否	否	队列总人数为多少时推送通知

(6) 消息表: 存放用户加入队列时给管理员发送的消息和退出队列时给后面队列成员以及关注队列的用户发送的消息, 包括 `fromId`、`toId`、`content`、`href`、`read`、`time` 等字段, 分别表示记录编号、发送者 ID、接收者 ID、消息内容、查看详情的链接、是否已读和发送时间, 其中 `id` 是该表的主键, `fromId` 和 `userId` 是该表的外键。详见表 3.6:

表 3.6 消息表

字段名	类型	长度	可空	外键	主键	备注
id	Int	13	否	否	是	编号
fromId	varchar	255	否	是	否	发送者 ID



toId	varchar	255	否	是	否	接收者 ID
content	varchar	255	否	否	否	消息内容
href	varchar	255	否	否	否	查看详情的链接
read	bool	1	否	否	否	是否已读
time	int	13	否	否	否	发送时间

(7) 队列历史表：存放队列变动数据，比如有人加入或退出，或者管理员设定停止排队、继续排队等，包括 id、userId、queueId、type、time 几个字段，分别表示记录编号、用户 ID、队列 ID、队列变动的行为和发生变动的时间，其中 id 是该表的主键，userId 和 queueId 是该表的外键。详见表 3.7：

表 3.7 队列历史表

字段名	类型	长度	可空	外键	主键	备注
id	int	13	否	否	是	编号
userId	varchar	255	否	是	否	用户 ID
queueId	int	13	否	是	否	队列 ID
type	enum	255	否	否	否	加入、退出、停止、继续等行为
time	int	13	否	否	否	发生变动的时间

(8) 举报表：存放用户对队列的举报信息，包括 id、queueId、userId、reportReason、time 等字段，分别表示记录编号、队列 ID、用户 ID、举报理由和举报时间，其中 id 是该表的主键，queueId 和 userId 是该表的外键。详见表 3.8：

表 3.8 举报表

字段名	类型	长度	可空	外键	主键	备注
id	int	13	否	否	是	编号
queueId	int	13	否	是	否	队列 ID
userId	varchar	255	否	是	否	用户 ID
reportReason	varchar	255	否	否	否	举报理由
time	Int	13	否	否	否	举报时间

(9) 反馈表: 存放用户对本小程序的意见或建议, 包括 `id`、`creator`、`content`、`contactWay` 和 `time` 几个字段, 分别表示记录编号、反馈人、反馈内容、联系方式和反馈时间, 其中 `id` 是该表的主键, `creator` 是该表的外键。详见表 3.9。其中 `creator` 之所以不设为外键, 是因为可能用户并没有登录, 所以只存放用户名称, 如果用户没有登录, 会默认显示“匿名”。

表 3.9 反馈表

字段名	类型	长度	可空	外键	主键	备注
id	int	13	否	否	是	编号
creator	varchar	255	否	否	否	反馈人
content	varchar	255	否	否	否	反馈内容
contactWay	varchar	255	否	否	否	联系方式
time	int	13	否	否	否	反馈时间

## 4 系统实现

排队小程序借助于微信小程序平台, 使用 Hbuilder 开发工具进行开发。首先去微信公众平台注册账号, 申请个人类小程序, 获取 `appID` 和 `AppSecret`, 然后下载 Hbuilder 开发工具进行开发。Hbuilder 创建项目时选择“uniapp”项目, 之后就开始编写代码, 并随时预览、调试。

### 4.1 前端实现

前端主要包括界面设计和逻辑设计两部分。界面设计即用户见到的图形界面, 逻辑设计是用户进行触摸、输入等操作以后, 前端逻辑层的响应。以下对排队小程序中的几个主要界面从这两方面进行说明。

(1) 首页。用户进入小程序以后, 首先打开的是首页, 为了表明排队小程序的主要功能, 首页以卡片的形式展示用户创建的队列, 如图 4-1 所示。其中包括队列 ID 编号、队列当前人数、开始排队时间、队列标题、队列主要说明、创建者、关注和加入按钮, 其中关注和加入按钮只有当用户不是创建者或者未登录

时才会显示，如果是创建者，则显示修改和管理按钮，点击以后分别跳转到“创建队列”页面和“队列成员管理”页面进行下一步操作。

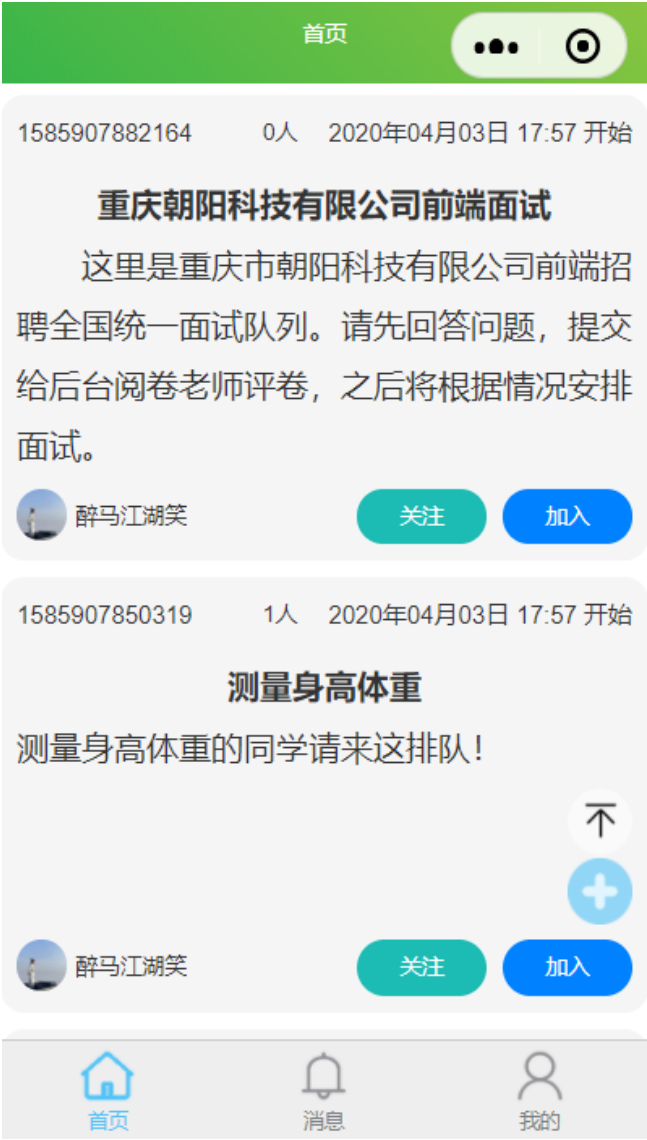


图 4-1 小程序首页图

（2）“队列详情”页面。当用户触摸或点击卡片以后，会自动跳转到队列详情页，详情页既包括卡片，又包括队列历史，比如某人加入了队列、退出队列，或者是管理员停止排队、恢复排队等，如果加入队列需要回答问题，也会在详情页显示。用户输入验证信息方可提交申请，队列管理员会收到消息，并决定是否允许其加入。如图 4-2 所示。



图 4-2 队列详情页面

(3) “创建队列”页面。在首页的右下方，有个加号形状的按钮，触摸或点击它会进入创建队列页面。在该页面填写队列名、队列介绍、开始排队时间等信息，然后提交，就可以创建队列。其中，只有队列名是必填项，其他可为空或有默认值，比如取号上限默认为 999 号，达到这么多以后，任何人都无法加入该队列。点击提交按钮的时候，前端逻辑层会进行验证，如果队列名称为空，则会提示“请补全信息”。如果没有登录，也会提示“请登录”。页面展示如图 4-3 所示。

创建队列

队列名称\*

请输入队列名称

请填写具体介绍

队列详情

图片上传

0/4

取号数上限

999

开始排队时间

2020/04/07 09:31:00

仅支持完整ID搜索

×

设置加入队列的验证信息

免验证

提交

图 4-3 创建队列页面

（4）“我的”页面。该页面包括展现个人信息、页面跳转（分别到创建队列列表页、关注队列列表页、加入队列列表页等）以及一个夜间模式的开关，打开则启动黑色主题。如果用户未登录则显示“登录”按钮，否则出现“退出登录”按钮。还可以点击“意见反馈”提交意见或建议，还有一个“关于”选项，可以查看排队小程序的相关信息。如图 4-4 所示。



图 4-4 “我的” 页面

(5) “我创建的队列” 页面。该页面显示的是所有自己创建且未被删除的队列列表，以条块形式呈现。管理员拖住条块向左滑动，会出现停止、修改、成员管理、删除几种操作。如图 4-5 所示。当用户点击停止之后，首先弹出确认提示框，用户再次点击确定以后，别人就不能加入队列，同时该条块左方“正常”二字会变为“停止”，而该操作条块上的“停止”二字，会变成“继续”。当用户点击“修改”以后，会跳转到“创建队列”页面，但是初始内容不为空，会把已保存的信息展示出来。当用户点击“成员管理”后，会跳转到“队列成员管理”页面。如果队列较多，不能一次性显示完全，会提示“点此加载更多”，用户点击以后会获取更多队列，显示在下方，否则会显示“加载完毕”。



图 4-5 创建队列列表页

(6) “队列成员管理”页面。该页面会显示所有正在等待及正在处理的用户，也是以条块方式呈现。如图 4-6 所示。管理员可以点击“叫号”按钮，队列首位成员就会变更状态为“正在处理”，同时“叫号”按钮变为“结束”，之后管理员再次点击，该成员就会出队，队列人数减少。当然，也可以直接滑动条块，选择“移除”操作，也就是强制出队。



图 4-6 队列成员管理

(7) “我加入的队列”页面。该页面显示所有用户自己加入的队列，向左滑动条块，可显示“退出”按钮，点击之后弹出确认对话框，用户再次点击“确定”以后就会退出该队列。如图 4-7 所示。“我关注的队列”页面同理，不过右方显示的不是前方人数，而是队列总人数。



图 4-7 我加入的队列页面

(8) 系统消息页面。该页面显示的是系统消息，比如有人加入了自已创建的队列，或者自己加入了别人的队列，或者加入的队列前方有人出队，或者关注的队列的人数有变动等等，都会发送系统通知。当点击查看详情时，会跳转到“队列详情”页面，如果是有人申请加入队列，就会跳转到“审核”页面。如图 4-8 所示。



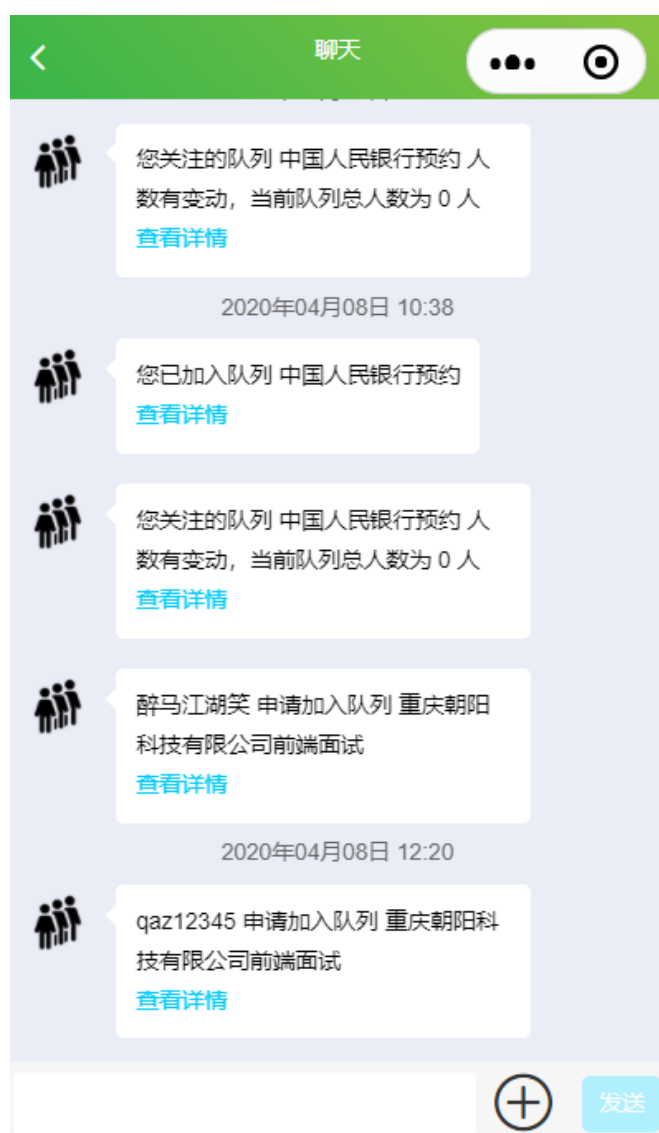


图 4-8 系统消息

(9) “审核”页面。该页面只对管理员开放，如果不是管理员，进入时会提示“权限有误”，并强制返回。该页面显示其他用户申请加入队列的验证信息，并有“通过”和“拒绝”两种操作。管理员阅览完毕所有回答之后可以决定是否允许其加入队列，这个决定无法反悔，不能重复操作。如图 4-9 所示。



图 4-9 审核页面

## 4.2 后台实现

后台功能主要是对前端操作进行相应处理。主要分为登录、获取队列列表、查询队列历史、数据库增加或改动队列信息、数据库增加消息、数据库增加或改动关注和加入队列的记录等。一般都是对数据库的操作，大致流程参见图 3-1。

(1) 新增数据表。首先到 uniCloud 云开发平台新增数据表，共有 9 张数据表，分别为用户信息表、队列信息表、队列历史记录表、队列验证信息回答记录表、加入队列记录表、关注队列记录表、系统消息表、举报信息表、反馈表，将其一一添加。

(2) 获取记录。以获取队列列表为例，首先后台获取前端参数，比如 type 参数值可能为“GETLIST”（获取首页列表）、“GETBYID”（根据 ID 搜索）、“GETCREATED”（获取用户创建的队列）、“GETJOINED”（获取用户加入的队列）、“GETWATCHED”（获取用户关注的队列）等，然后在数据库查询语句里进行相应的筛选操作。每种类型返回字段不尽相同，比如获取用户创建的队列，

不需要返回管理员信息，因为用户自己就是管理员，所以返回消息需要进一步处理。

（3）新增和修改记录。以新增队列信息为例，用户在小程序上填写“创建队列”表单以后，点击“提交”按钮，将数据发送到后台，后台首先进行判断数据是否合法，如不合法返回失败状态码和提示消息，否则进行插入数据操作，如果依旧失败，同样返回失败状态码和提示消息，否则返回成功状态码。前端判断状态码，如为失败，则显示后台传输的消息，否则提示“操作成功”。修改记录与新增记录大致相同，但是需要先找到原记录，再进行修改。

（4）删除记录。以删除队列为例，首先根据主键找到记录的位置，再将其state字段改为“DELETED”，从此虽然数据库虽然保留，但是用户并不能获取到该队列的信息。如果是关注队列的删除，则是从数据库里面真正删除，因为该记录没有保留的必要。

（5）消息推送。这是排队小程序的核心内容，当满足用户设定的条件，就通过微信给用户发送“服务通知”，这需要到微信开发者平台申请订阅消息的模板，并在后台调用接口，最后效果如图 4-10 所示。上方消息是队列成员收到的通知，下方是点击“关注”队列的用户收到的通知。



图 4-10 微信通知效果

### 4.3 系统测试

系统测试是开发过程中为保证程序机制完善、用户友好的重要环节。排队小程序主要测试设备的相关信息如表 4.1 所示：

表 4.1 测试机信息

机型	系统版本	微信版本
荣耀 9x	Android 10	7.0.13

使用该设备进行了多次黑盒测试。比如在未登录的情况下，点击“加入”按钮，或者创建新队列，或者在未完善队列信息的情况下提交创建，或者是拒绝授权检测能否收到通知，或者是数据库删除了队列等等。部分黑盒测试项目的具体操作和结果如表 4.2 所示：

表 4.2 测试项目及结果

前提	操作	结果	是否合理
未登录	点击“加入”	提示“请登录”	是
未登录	创建队列	提示“请登录”	是
已登录	点击“加入”并拒绝授权	提示“授权失败，无法通知”	是
已登录	点击“加入”并允许授权	提示“操作成功”	是
队列信息不完整	创建队列	提示“请补全信息后提交”	是
队列不在排队时间内	查看队列详情	没有“加入”按钮	是
数据库删除所有队列	查看队列列表	无结果	是

上表仅列出了部分测试数据，根据结果来看，排队小程序总体运行效果较为合理，在主要功能方面，有较为完善的错误捕获机制以及信息提示，能够带给用户良好的体验。

## 5 总结

一个明确的需求是程序设计的开端。排队小程序有着清晰明确的目标：提高人们的排队效率。排队小程序得益于微信小程序平台以及 uni-app 框架等，相比于传统应用程序，大大缩短了开发周期，借助于云开发平台，省去了服务器配置等事务的时间，同时能够获得良好的服务器支持，节省了大量的开发成本。在功能设计方面，队列前方成员出队时会发送通知给后方成员，能够满足在排队这个特定场景下人们节省时间的需求。在数据库设计方面，能够满足第三范式设计的要求。排队小程序总体上较为成功。

排队小程序也存在一些不足，也是未来工作的开展方向，主要包括：因为涉及到信息发布，微信小程序不能上线，无法让更多的人使用排队小程序；在聊天功能方面，排队小程序应当提供同队列成员进行交流的功能，可以让用户更好地了解队列的用途；在算法方面增加出队时间预测，可以让用户心中有数，当然，只对排队人数达到一定程度的队列有效。

### 参考文献：

- [1] 詹舒波. 医疗挂号系统微信公众号的设计与实现[D]. 北京:北京邮电大学, 2018.
- [2] 徐玲利, 李唯. 在线点餐系统的设计与实现[J]. 电脑知识与技术, 2018.
- [3] 王婷婷. 微信小程序开发[J]. 信息技术与信息化, 2018(12):62-63.
- [4] 汪峰. 在线课堂微信小程序的设计与实现[D]. 兰州:西北民族大学, 2019.
- [5] 舒嘉豪. 基于微信小程序的答题系统的设计与实现[D]. 武汉:华中科技大学, 2019.
- [6] 杨毅, 李兆瑞. 微信小程序在高校图书馆中的创新实践[J]. 信息记录材料, 2019, 20(02):68-70.
- [7] 彭守镇. 微信小程序应用探究[J]. 信息与电脑(理论版), 2018(22):22-23.
- [8] Lei Hao, Fucheng Wan, Ning Ma, Yicheng Wang. Analysis of the Development of WeChat Mini Program[J]. Journal of Physics: Conference Series, 2018, 1087(6).
- [9] Kumar J A, Bervell B. Google Classroom for mobile learning in higher education: Modelling the initial perceptions of students[J]. 2019(1).
- [10] Yue Jia Xin, Wang Zhong, Lai Hong. Smart Gate System Design and Implementation Based on Cloud Platform[J]. Procedia Computer Science, 2019, 154.
- [11] Tu Pan, Zhan Ying. Thoughts on the Construction of Thesecond-hand Trading Platform of WeChat Small Program Campus[J]. Academic Journal of Engineering and Technology Science, 2019, 2(1).