

本科生毕业论文(设计)

****

**题 目 会议室预约及签到微信小程序的设计与实现**

学生姓名 李臣斌

学 号 201883290508

学 院 计算机学院、软件学院、网络空间安全学院

专 业 软件工程

指导教师 韩进

**二Ｏ一二 年 四 月 二十 日**

目 录

[1. 绪论 1](#_Toc4377)

[1.1. 研究背景 1](#_Toc13970)

[1.2. 国内外研究现状 1](#_Toc11601)

[1.3. 研究目的和意义 2](#_Toc32644)

[2. 开发工具及开发技术介绍 2](#_Toc25031)

[2.1. 开发工具 2](#_Toc31705)

[2.1.1 IDEA 2](#_Toc18625)

[2.1.2 MYSQL数据库 2](#_Toc14243)

[2.1.3 微信开发者工具 3](#_Toc13665)

[2.2. 相关技术介绍 3](#_Toc15208)

[2.2.1 JAVA语言介绍 3](#_Toc20543)

[2.2.2 Springboot介绍 3](#_Toc28391)

[2.2.3 JPA介绍 3](#_Toc17816)

[2.2.4 WXSS和WXML介绍 3](#_Toc2807)

[2.2.5 Maven介绍 3](#_Toc31282)

[3. 需求分析和可行性分析 4](#_Toc23406)

[3.1 需求分析 4](#_Toc9252)

[3.2 用户非功能性需求定义 4](#_Toc22834)

[3.3 系统可行性分析 5](#_Toc20262)

[3.3.1 技术可行性研究 5](#_Toc9699)

[3.3.2 经济可行性研究 5](#_Toc20499)

[3.3.3 运行可行性研究 5](#_Toc32188)

[3.4 业务流程分析 5](#_Toc2497)

[4. 系统设计 7](#_Toc6562)

[4.1 系统的功能结构图 7](#_Toc2676)

[4.2 系统数据库设计 7](#_Toc13049)

[4.2.1 数据库E-R图 7](#_Toc8220)

[4.2.2 数据库表字段设计 8](#_Toc19144)

[5. 系统功能实现 9](#_Toc16887)

[5.1 登录模块实现 9](#_Toc20101)

[5.2 发布会议模块 10](#_Toc2941)

[5.2.1会议信息 10](#_Toc16857)

[5.2.2全体会议列表 11](#_Toc22374)

[5.2.3二维码网页 12](#_Toc11153)

[5.2.4关闭签到 14](#_Toc29837)

[5.3 扫码签到模块 15](#_Toc32076)

[5.4 预约模块 15](#_Toc23401)

[5.4.1会议室预约情况图 15](#_Toc11702)

[5.4.2预约 16](#_Toc12244)

[5.5 信息管理模块 18](#_Toc16789)

[5.5.1查看预约情况 18](#_Toc31347)

[5.5.2查看签到情况 19](#_Toc8745)

[6. 结论 21](#_Toc7682)

[参考文献 22](#_Toc21432)

[致谢 23](#_Toc28744)

会议室预约及签到微信小程序的设计与实现

李臣斌

南京信息工程大学计算机学院、软件学院、网络空间安全学院，江苏 南京 210044

摘要: 近年来，企业公司数量越来越多，伴随而来企业公司资源管理所需的人力也会增多，这使得对企业的资源管理十分具有挑战性。为了解决资源管理中的会议室管理部分的问题，减少在此方面的人力资源分配，因此开发了会议室预约及签到的微信小程序。本系统总体上采用springboot框架和数据库的JPA框架，数据库采用MySQL数据库。微信小程序由于它简单方便的特性，在这个时代非常流行，它不需要下载安装，只需要有微信app即可使用，用户通过扫二维码或搜索关键词就可以找到并打开应用。所以选择以微信小程序来呈现此系统，使得该系统更加便利，这对减少管理会议室的人力具有重大意义。

关键词: springboot；JPA；微信小程序

**Design and implementation of wechat applet for meeting room reservation and check-in**

Chenbin Li

School of Computer Science, NUIST, Nanjing 210044, China

**Abstract:** In recent years, the number of enterprises is increasing, and the manpower required for enterprise resource management will also increase, which makes the resource management of enterprises very challenging. In order to solve the problem of meeting room management in resource management and reduce the allocation of human resources in this regard, a Wechat Applet for meeting room reservation and check-in is developed. The system generally adopts springboot framework and JPA framework of database, and the database adopts MySQL database. Due to its simple and convenient characteristics, Wechat Applet is very popular in this era. It does not need to download and install. It only needs wechat app. Users can find and open the application by scanning QR code or searching keywords. Therefore, This is of great significance to reduce the manpower for managing the conference room.

**Key words:** springboot, JPA, Wechat Applet

# 绪论

本章介绍系统研究背景，研究现状，以及研究的目的和意义。

## 研究背景

在当前社会中，伴随着大学生创业人数在不断地扩大，公司企业的也在不断增多。因此对于内部资源的管理所需要的人力也会增多，但效率却不高，随着时间的推移，人力管理资源的弊端会越来越明显，这会给企业运营带来不小的压力。在我国现在的环境下，企业内的会议越来越频繁，因此更加需要对会议室进行管理的系统，这是现代化发展的必要条件。

会议管理是一个动态的合成词。顾名思义，“会”基本上是指会议、聚会、集会等，“议”基本上是指讨论和协商。管理包括执行，控制和改善。现代意义上的会议是组织和引导人们讨论问题的活动。它体现了会议的四个基本条件:有组织、有领导、商议事情和集会。一个会议的构成，包括以下要素:会议时间、会议地点、会议出席人员等。会议室管理是为了保证会议的正常进行并提高会议的效率，避免发生不同的会议的时间发生冲突的情况，而对会议室的预约、签到、查询等工作的一种有效的协调。

## 国内外研究现状

会议室管理本质上是管理信息系统的一部分，以下简单介绍一下管理信息系统的发展概况。

二十世纪，全世界的经济都得到了极大的发展，非常多专注于经济领域的学者都提出了关于管理的新颖思想。二十世纪五十年代，西蒙是第一个想到关于管理依赖于信息和决策的理论的学者。并且他在相同时期的维纳发表了控制论，表达管理是一个过程的思想。1958年，盖尔提出：“要对管理做到良好的控制，就要以较低的成本得到及时准确的信息。”从此，计算机在会计工作中发挥越来越大的作用，数据处理一词也是在那时候诞生的。

二十世纪七十年代，沃尔特·T·肯尼文用这样一个定义形容刚诞生的管理信息系统：“通过口头或笔头的形式，在适当的时间向相关人员提供过去、现在以及未来关于企业内部以及环境的信息，来援助他们决策。”固然其中强调了信息支持决策，但事实上应用模型却没有被重点突出，忽视了应用计算机的可能性。

1985年，作为管理学教授的戈登.B.戴维斯，创建了管理信息系统，对管理信息系统下了一个比较完善的定义，即“管理信息系统是一个人机交互系统，使用了计算机资源、手工作业、分析、计划、控制和决策模型以及数据库。它能提供支持企业运行所需的信息，并且还支持管理和决策。”此定义从各个方面表述了管理信息系统的目的、作用以及组成，并且反映了当时管理信息系统的水平。[4]

我国管理信息系统起步的时间比较晚，对于它比较权威的定义是这么说的:“此系统由人和计算机等组成，它的功能有收集信息、传送信息、存储信息、加工信息、维护信息和使用信息。管理信息系统能实测企业的各种运行情况,利用过去的数据预测未来,从企业全局出发辅助企业进行决策,利用信息控制企业的行为,帮助企业实现其规划的目标”。定义的前一半解释了物理功能，后一半解释了应用功能，并重点强调计算机是管理信息系统的一种工具。

管理信息系统经历了人工阶段、机械系统阶段、计算机系统阶段。

人工阶段：企业的信息管理全部依靠人力来完成。

机械系统阶段：1896年，统计学家赫尔曼·霍勒雷斯（Herman Hollerith）发明了穿孔卡片进行统计的方法。

计算机系统阶段：1946年，第一台电子计算机问世，信息处理技术发生了重大革命。1954年，美国通用电气第一次在Unicac 1上计算工资，使计算机的应用首次进入管理业务的领域。

## 研究目的和意义

根据本文的研究内容和当前信息管理系统的现状，本文将实现一个便利的、综合的基于微信小程序的会议室预约及签到系统。此系统将包含签到、预约、查询会议室预约情况、查询签到状态，以及管理员独有的发布会议、关闭会议、查看签到人员。

本课题研究的主要目的是开发一个会议室预约及签到的微信小程序，提高对会议室资源管理的效率，通过此系统可以更为方便直观的进行会议室预约，避免不同会议的时间及地点发生冲突。此系统需要实现对人员信息、微信相关的信息以及会议信息的存储，增删改查。

本课题研究的意义在于可以很方便的让人了解各个会议室当前的预约状态，已约或者空闲。本系统既实现了对会议室的基本操作，包括预约、取消预约，又能够进行签到、查询会议室的预约情况和签到状态，相比于传统的会议室管理系统，基于微信小程序的该系统更加便利高效，只要拥有手机能够随时随地进行相关操作，对于高效化会议室管理具有重大意义。

# 开发工具及开发技术介绍

本章介绍系统开发的相关工具，相关环境，相关技术等开发手段。

## 开发工具

2.1.1 IDEA

IDEA的全名是 IntelliJ IDEA，它提供了整合 java程序设计语言的开发环境。IntelliJ是业内十分优秀的 java工具，特别是智能代码助手，自动代码提示，重构， JavaEE支持， git, svn, JUnit, CVS整合，代码分析，创新的 GUI。其主要的功能是： HTML、 CSS、 PHP、 MySQL。Python等。自由版本仅对少量的 Java、 Kotlin提供了支持。

2.1.2 MYSQL数据库

MYSQL为轻量级关系型数据库，原本的研发公司是MySQL AB，如今该软件归属于ORCALE公司。MYSQL数据库体积小，速度快，成本低，性能卓越，是一个很好的开发工具[8]。该数据库用C/C++高级语言开发，有很好的可移植性，它提供了一系列的API接口，可使用这些接口的语言有JAVA，Python等，接口方便了用户使用这些编程语言进行项目的开发，实现数据的存储，这些接口可以使用TCP/IP，ODBC等方式连接数据库。MYSQL使用了多线程进行算法的优化，极大的提升了CPU的使用效率，也提高了数据库查询效率和准确率，降低了系统的整体时间复杂度。本系统采用MYSQL数据库来存储数据，提升了系统的开发效率，降低了系统的开发成本，并让系统有着很好的兼容性和可靠性，易于后期的维护和管理。

2.1.3 微信开发者工具

微信开发者工具是用来开发微信小程序的工具，它拥有开发，调试，预览，上传等许多功能。开发者可以通过专门的开发文档和设计指南解决开发问题，而开发者工具则有助于帮助开发者简单高效地开发微信小程序。

启动工具时，开发者需要使用已在后台绑定成功的微信号扫描二维码登录，后续所有的操作都会基于这个微信的帐号。

## 相关技术介绍

2.2.1 JAVA语言介绍

一方面，JAVA语言与C++程序设计语言语法形式类似，这使得计算机编程人员很容易上手使用。另一方面，JAVA语言去掉了C++中的许多晦涩难懂的特征，如指针操作运算，多继承机制，隐式类型转换。JAVA通过引用取代了指针，并提供了垃圾回收机制，便于使用，也消除了内存溢出的隐患。JAVA在继承方面更多使用的是接口实现机制，并支持接口的多继承，接口与C++中的抽象函数很相似，综上所述，JAVA是一个纯粹的面向对象的编程语言。

2.2.2 Springboot介绍

Spring Boot是一个比较新的框架，它能简化Spring应用搭建以及开发的过程。这个框架的配置以特定的方式进行，因此开发人员不再需要定义样板化的配置。boot采用了config的方式进行配置，开发中使用大量注解，效率更高。并且内置tomcat。现在的主流是分布式、微服务架构系统，而spring boot框架能够很好地支持微服务开发，它是基础，构建一个系统也可以使用它。

2.2.3 JPA介绍

JPA是JDK 5.0的注解，也可以用于在XML中描述对象之间的映射关系，并将运行期的实体[对象持久化](https://baike.baidu.com/item/%E5%AF%B9%E8%B1%A1%E6%8C%81%E4%B9%85%E5%8C%96/7316192" \t "https://baike.baidu.com/item/JPA/_blank)到数据库中。JPA的主要目的之一是简化编程模型：在JPA框架下，可以像创建Java 类一样简单地创建实体，几乎没有约束和限制，并且框架和接口简单易懂，不拘泥于规则和设计模式，便于开发者掌握。此外，JPA基于非侵入式原则设计，所以便于与其它容器或者框架集成。

2.2.4 WXSS和WXML介绍

WXSS是微信开发者工具中使用组件样式语言。 WXML 组件的属性由WXSS 决定。为了让前端开发者快速理解，WXSS 拥有 CSS 大部分特性。与此同时为了更方便微信小程序的开发，WXSS 对 CSS 进行了一些扩充以及修改。CSS是指层叠样式表，是一种用来结构化文档，添加样式的计算机语言。该语言可以用来优化前端页面，如进行美观处理，结构处理。WXML是微信官方设计的一套标签语言，通过结合基础组件、事件系统，就能够构造出页面结构,主要功能有数据绑定、条件渲染、列表渲染、引用以及模板。

2.2.5 Maven介绍

Maven根本上来说是一个项目管理工具，可以通过简单的信息来管理项目建设和报告以及文档。Maven不仅程序构建能力出色，而且提供高级项目管理工具。因为 Maven 的缺省构建规则是高度可重用的，因此构建简单的项目通常只需用两三行 Maven 构建脚本。

# 需求分析和可行性分析

本章从实际的需求角度来讨论需要实现的功能，有计划有目的的进行系统的开发工作，以及论述系统的可行性。

## 3.1 需求分析

会议室预约及签到的微信小程序的主要的服务对象包括：普通用户和系统管理员，其中普通用户可以进行预约、取消预约、签到、查询会议室预约情况以及签到状态；管理员在系统中拥有最高的权限，除了拥有普通用户所有的功能，还可以发布全体会议、关闭会议签到以及查看签到人员列表。

综上所述，该系统应该具有以下的功能

1. 会议室预约：对预约表的信息进行修改、增加、删除
2. 签到：对签到表中进行增加、统计人数
3. 会议发布：管理员发布全体参加的会议，并且管理员可以随时关闭签到
4. 信息查询：可以查询自己的签到状态（已签到/未签到）以及各个会议室的在各个时间段的预约情况，管理员可以额外下载签到列表

系统功能如下图3.1所示

下载签到人员名单

系统功能

会议室预约

信息管理

发布会议(管理员)

签到

预约

取消预约

查询某天会议室预约情况

查询

自己所有会议的签到状态

发布全体会议

关闭签到

加入签到名单中

图3.1系统功能图

## 3.2 用户非功能性需求定义

非功能性需求包括系统的安全性，健壮性，界面的美观程度，系统的响应速度，这些功能都要符合规范，系统整体要做到风格一致，界面美观，响应迅速，安全性高。系统的设计从用户的角度来进行考量，应使系统的设计符合人的基本使用习惯、基本思维方式，能够更加方便用户的使用。

(1)安全性：不同权限下的用户操作不尽相同。

(2)界面美观：采用WXSS和JavaScript进行优化。

(3)系统的稳定性：系统在运行过程中不会突然崩溃，影响用户体验。

## 3.3 系统可行性分析

该系统开发的总体要求是完成会议室的预约管理以及签到，会议预约信息的录入和查询，签到信息的查看。使得会议室的管理更加便利化，高效化。该系统用户通过微信小程序前端即可进行相关操作并访问后端的数据库，实现对预约会议室信息、发布会议、签到信息等的一系列操作。

3.3.1 技术可行性研究

技术可行性主要关注现有的技术条件和软硬件条件是否能够满足系统的开发需求，以及开发人员的技术水平是否满足开发的需求。当前的微信小程序的要求配置条件完全不高，几乎微信用户都可以使用。该系统基于的编程语言、框架和数据库技术基础等技术在如今已经非常成熟，大部分的程序员都会使用这种技术，因此该系统在技术上可以很方便的实现。

3.3.2 经济可行性研究

该项研究主要从项目的经济层面进行评估，一方面是开发该项目的支出，其中这方面支出包括工作量成本、软硬件费用等，另一方面是项目的收益。由于本系统是自己的毕业设计，花费的都是自己的时间，所以在工作量成本的方面不需要进行太多的考虑；在软硬件设施费用上，因为开发环境用的是自己的电脑，项目也没有收益和损失，所以在经济方面该项目是可行的。仅仅需要的是部署云服务器以及域名的部分成本。

3.3.3 运行可行性研究

该项研究主要从系统的运行环境，运行条件两方面进行分析。该项目的运行环境为

移动端。由于该系统的容量适宜，运行时候不会消耗过多的内存，所以只需要拥有微信客户端，就可以实现该项目的运行。此外，项目在运行过程中即便出错误也不会造成系统的崩溃。因此，该项目在运行方面具有可行性。

## 3.4 业务流程分析

在整个系统流程分析中，调查分析是一比较重要的工作。考虑到系统中的业务流程都存在相关性这一客观前提，所以在会议室预约及签到微信小程序的整体设计基础上，需要考虑和确保信息正确性，数据库存储的完整性。系统开发首先要进行需求分析，划分系统模块，设计数据库模型，然后开始具体模块的代码编写工作，最后对代码模块进行整合，调试优化。基于以上所述的实际操作，可以得出开发流程图如下图3.2所示：

图3.2开发流程图

需求分析

系统模块划分，系统功能设计，数据库设计

分解系统模块后，进行代码编写

独立编写模块，整合模块

对系统进行调试优化

开始

结束

登录流程：使用者首先要进行登录操作，而由于用户的微信账号是唯一的，所以仅需要在登录时输入自己的工号，之后系统会将工号与用户的微信账户绑定，下一次进入时会跳过登录直接进入主界面。用户的登录流程如下图3.4所示：

N

Y

开始

用户在登陆界面输入工号

点击绑定

进入系统

结束

是否通过验证

图3.3系统登陆模块

# 系统设计

本章主要讲述系统的功能结构，E-R图的绘制，数据库表段设计，数据库中的表段在项目中起到的具体作用。

## 4.1 系统的功能结构图

以系统需求分析和可行性分析为依据，本会议室预约及签到小程序实现的主要功能包括：普通用户可以在签到、预约、查看自己的签到情况和会议室预约情况。管理员用户拥有普通用户的所有功能，并且额外加上查看签到名单、发布全体会议、关闭签到功能。其功能结构图如下图4.1所示：

会议室预约及签到微信小程序

普通用户

管理员用户

签到

预约

会议室

查看签到状态

签到

预约

会议室

查看签到状态

查询预约情况

发布会议

查询预约情况

下载签到名单

关闭签到

图4.1 系统功能结构

## 4.2 系统数据库设计

4.2.1 数据库E-R图

E-R图即Entity-Relation是一种实体-关系图。它从逻辑上来描述数据库的结构关系，数据的属性和联系。其可以有效地使用概念模型来描述各个数据之间的关系。本系统中的E-R图如下所示：

图4.2 系统E-R图

m

管理员

Id

工号

权限

名字

Openid

普通用户

Id

工号

权限

名字

Openid

会议室

会议室名

时间段

空闲状态

使用时间

会议

Id

会议室名

时间

Token

1

1

开会

发布

1

n

预约

1

n

签到

n

4.2.2 数据库表字段设计

系统开始运行后使用的数据库信息来自于不同的表段，如在登陆界面输入工号，前端返回到后台进行验证，这时候就需要匹配用户表中的用户信息进行验证工作。在会议室预约及签到小程序中，多个数据表被包含其中，其设计和划分依据是根据系统的不同模块的不同功能。本数据库系统的表段设计如下表所示：

表4.1 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 中文名称 | 类型 | 长度 | 允许空 | 主/外键 |
| id | 编号 | bigint | 19 | 否 | 主键 |
| user\_jobnumber | 工号 | varchar | 255 | 是 | 否 |
| user\_authority | 用户权限 | int | 10 | 是 | 否 |
| user\_name | 名字 | varchar | 255 | 是 | 否 |
| user\_openid | 微信 | varchar | 255 | 是 | 否 |

表4.2 会议室预约表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 中文名称 | 类型 | 长度 | 允许空 | 主/外键 |
| id | 编号 | bigint | 19 | 否 | 主键 |
| meeting\_room\_id | 会议室名 | varchar | 255 | 是 | 否 |
| time\_index | 时间序列 | int | 10 | 否 | 否 |
| user\_date | 日期 | varchar | 255 | 是 | 否 |
| useroom\_user\_id | 用户微信 | varchar | 255 | 是 | 否 |

会议室预约表是将会议室的信息与用户的信息以及时间关联起来。

表4.3 签到信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 中文名称 | 类型 | 长度 | 允许空 | 主/外键 |
| id | 编号 | bigint | 19 | 否 | 主键 |
| meeting\_id | 会议编号 | bigint | 19 | 是 | 否 |
| sign\_in\_name | 签到人员名字 | varchar | 255 | 是 | 否 |
| timestamp | 签到时间 | varchar | 255 | 是 | 否 |
| user\_code | 工号 | varchar | 255 | 是 | 否 |
| user\_openid | 用户微信 | varchar | 255 | 是 | 否 |

签到信息表用来存储签到人员的名字和对应微信账号的openid以及对应的会议编号，在系统中可以用于管理员下载签到名单查看。

表4.4 全体会议信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 中文名称 | 类型 | 长度 | 允许空 | 主/外键 |
| id | 编号 | bigint | 19 | 否 | 主键 |
| is\_open | 会议状态 | bit | 1 | 否 | 否 |
| meeting\_place | 地点 | varchar | 255 | 是 | 否 |
| meeting\_time | 名字 | varchar | 255 | 是 | 否 |
| token | 随机码 | varchar | 255 | 是 | 否 |

全体信息表用来存储管理员发布的全体会议的基本信息，其中的token是用来组成二维码网页url的一部分。

# 系统功能实现

## 5.1 登录模块实现

模块功能介绍：

该页面是进入小程序后初始页面后展现的页面。在此界面中，在LOGO下方居中显示文本框以输入工号，下方有绑定按钮。用户可以在文本框中输入自己的工号后，点击文本框下方的绑定按钮进行用户的登录。由于已将用户微信与该小程序绑定，当下次在进入小程序时不再需要输入工号，直接进入主页面。

代码流程：

通过@RequestParam注解将微信前端的openid和工号数据映射到功能方法参数上，然后判断openid是否为空，若为空，返回null。通过自己编写的removeZero方法将工号中开头的数字0全部消除。通过Jpa框架中的JpaRepository接口中的findByxx方法从数据库user表中查询该工号对应的记录，返回给用户对象，代码如下：

User u = userService.findByCode(job\_number);

并通过判断该对象的内容执行不同操作，如果没有对应工号的记录,返回对象u本身，否则

并且微信openid也不为空的话，则返回一个新的用户对象，并将该记录的openid赋值给用户对象对应的属性,然后通过JPA框架JpaRepository接口中save方法将u的各项属性保存到user表中。

其流程图如下图5.1所示：

开始

输入工号

失败

Openid是否合法？

工号是否正确？

登陆成功

Y

N

Y

图5.1 用户登录流程图

运行后数据库user表如下图5.2所示

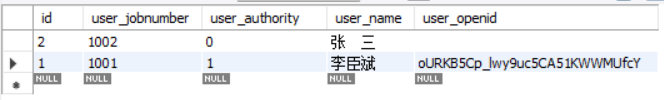


图5.2 运行后数据库user表

## 5.2 发布会议模块

5.2.1会议信息

功能介绍：

此模块将为管理员发布总体会议生成随机的token码，并将会议信息存储在数据库中。

代码流程：

使用@GetMapping注解处理请求方法的GET类型，@RequestParam注解将前端的date，time，openId数据映射到功能方法参数上，此系统中把全体会议的会议地点设为固定的“总会议室”，再通过jpa框架中的findBy方法查询数据库user表中用户openid对应的记录，若没有记录或这没有管理员权限，则返回错误信息“error”,否则创建一个新的总体会议对象，并通过lombok自动生成的set方法设置该对象的时间、地点、状态属性，代码如下：

MeetingRelease meetingRelease = new MeetingRelease()；

meetingRelease.setMeetingTime(date + " " + time)；//会议具体时间

meetingRelease.setMeetingPlace(place)；//会议地点

meetingRelease.setOpen(true)；//会议状态

然后通过UUID生成唯一的字符串标识，赋值给String对象uuid，代码如下：

String uuid = UUID.randomUUID().toString(),

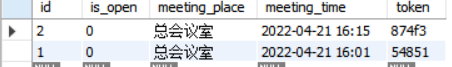
再将0到uuid长度之间的随机值赋值给index，随后将uuid同一个新的字符串标识连接并将其中所有的“\\-”替换为“”空字符串,然后取从index位置开始往后5个字符作为会议的token属性，并通过save方法将发布会议的信息保存在meeting\_release表中，代码如下：

meetingRelease.setToken( (uuid+UUID.randomUUID().toString()).replaceAll("\\-", "").substring(index, index+5))；//产生随机token

meetingReleaseService.save(meetingRelease);//保存到meeting\_release表中

之后返回发布成功的提示。

运行截图如下：

图5.3 运行后meeting\_release表

5.2.2全体会议列表

功能介绍：

将所有全体会议按时间降序排列，时间越早的越往下。

代码流程：

定义一个JSONArray类型的方法，使用@GetMapping注解处理请求方法的GET类型，@RequestParam注解获得前端的openid，通过jpa框架的findby方法查找与openid对应的user表中的记录，若没有记录，返回一个新的JSONArray对象。否则，通过findAll方法无条件查询meeting\_release表中的所有记录，存放于对应类型的集合中，代码如下：

List<MeetingRelease> meets = meetingReleaseService.findAll();

然后用Collections的sort方法将meets集合按照指定比较器引发的顺序进行排序，代码如下：

Collections.sort(meets, new Comparator<MeetingRelease>() {

public int compare(MeetingRelease arg0, MeetingRelease arg1) {

return -1 \* arg0.getMeetingTime().compareTo(arg1.getMeetingTime() );

}

});

//该比较器为比较会议的开始时间，并将返回的结果置为相反数，1为升序，-1为降序

//将所有发布的会议按时间降序排序

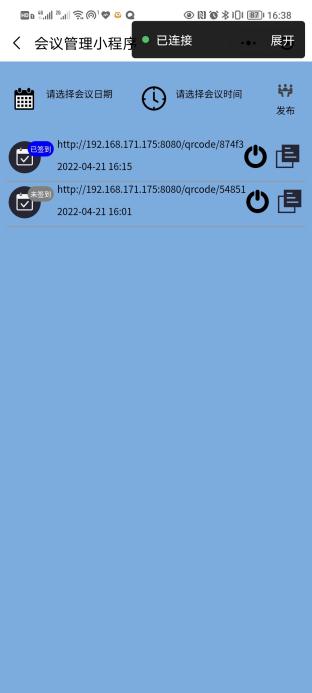
结果如下：

图5.4 发布成功

5.2.3二维码网页

功能介绍：生成供扫码签到的二维码，以网页形式呈现，会自动统计签到人数，并呈现在网页上。

代码流程：

首先先要生成二维码，参考Zxing库中的二维码生成工具类，创建一个新的Hashtable对象，然后设置字符编码类型为UTF-8，代码如下：

hints.put(EncodeHintType.CHARACTER\_SET, "utf-8");

BitMatrix bitMatrix = new MultiFormatWriter().

encode(url, BarcodeFormat.QR\_CODE, 200, 200, hints);

//生成二维码的核心代码，二维矩阵转为一维像素数组

开发过程中考虑了两种获取二维码的方法，一种是输出图片到指定目录，另一种是生成二维码图片并将图片转为二进制传递给前台，通过网页呈现，考虑到微信小程序大部分时候基于手机，并且为了便于人员直接扫码，选择了第二种方法。

先创建一个缓冲区，然后将bitMatrix类型转化为BufferedImage类型赋值为image对象，再通过ImageIO的write方法输出图片，代码如下：

ByteArrayOutputStream out = new ByteArrayOutputStream();

//创建一个32字节（默认大小）的缓冲区

BufferedImage image = toBufferedImage(bitMatrix);

//将Image转成BufferedImage的方法

ImageIO.write(image, "png", out);

//图片输出格式为png、存储在缓冲区中

创建新分配的字节数组。它的大小是此输出流的当前大小，并且缓冲区的有效内容已复制到其中，接着把字节数组转为二进制，将字节数组中的数据编码为BASE64字符串赋值给binary，返回binary。

然后实现显示带参二维码网页的接口。使用@GetMapping注解处理请求方法的GET类型，@PathVariable注解将URL中占位符参数绑定到控制器处理方法的入参中，@PathVariable的required参数设为false，如果没有传这个参数，那么它会去找这个参数去掉之后的替代url ，如果发现有替代的url，就可以处理这个请求，如果没有找到，就抛出异常不去处理。如果token内容为空，则往前台传数据，代码如下：

if( "".equals(token)){

model.addAttribute("notoken", true);//往前台传数据

return "index"；}

设置时间显示格式代码如下：

SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss")；

然后将规定格式的时间赋值给String类型的datetime，并设置一个随机数，随即范围为[0,1000)，代码如下：

Random random = new Random();

random1= random.nextInt(1000);//随机生成的随机数

通过findBy方法在meeting\_release表中查询token对应的记录，赋值给MeetingRelease类型的mr对象,如果表中没有记录，则往前台传表示出错的数据，网页显示找不到对应的会议号！

若对应记录的会议状态已关闭或是会议开始前15分钟以及开始后30分钟，则无法签到，往前台传表示出错的数据，网页显示会议已关闭签到。 判断的代码如下：

if( !mr.isOpen() || System.currentTimeMillis() < mrDate - spanStart || System.currentTimeMillis() > mrDate + spanEnd ){

model.addAttribute("error", "会议已经关闭签到！"),

return "index"；}

其中，mrdate为/将记录中的会议时间转化为日期后，获取自1970年1月1日00：00：00GMT以来的毫秒数，spanStart和spanEnd分别为900000和1800000。

将token、datetime、random1以"/"间隔组成字符串codeParam。接着用上述的生成二维码的接口处理codeParam，并往前台传数据，代码如下：

String qrCode = "data:image/png,base64,"+qrCodeGenerate.generalQRCode(codeParam)；

//base64编码的png图片数据

model.addAttribute("notoken", false);

model.addAttribute("qrcode",qrCode)；

model.addAttribute("datetime",datetime)；//往前台传数据

统计签到人数代码流程为，用findby方法查找meeting\_sign\_in签到表中所有meetingid与此会议相同记录，存入集合中，然后返回集合的大小即为签到人数。

运行截图如下：

图5.5 二维码网页

5.2.4关闭签到

功能：关闭对应会议的签到，之后任何人无法再进行对应会议的签到，签到后会显示会议已关闭签到。

代码流程如下：首先根据openid判断是否为管理员，若不是返回错误信息error，

否则将会议状态改为关闭后，返回成功信息success。其中将会议状态改为关闭的代码如下：

@Query(value = "update meeting\_release set is\_open=false where id=:id ", nativeQuery = true)

public void close(@Param("id")long id);

运行后签到网页截图如下：

 图5.6 签到失败

## 5.3 扫码签到模块

模块功能介绍：

当管理员发布了全体会议时，将会用到签到功能，所有参会人员都需要进行扫码签到。签到后，人员的相应信息都会储存到签到表中，以便管理员后续查看人员签到情况。

代码流程：

首先，在微信开发前端调用扫码微信接口进行扫码，后端通过@RequestParam将对应数据映射到功能方法参数上，通过微信openid在user表中查找对应用户的记录，有记录则先规定时间格式为“年-月-日 时：分：秒”，然后将二维码的字符串以"/"作分割分别存入String数组sourceStrArray中，由之前二维码网页部分可知，codeParam的从头开始依次往后的三个部分为token、datetime、random1，因此sourceStrArray[0]的内容即为token，再根据这个token从meeting\_release表中查找对应的记录，若没有记录，返回签到失败信息；反之，从meeting\_sign\_in表中查询签到人员工号和会议编号对应的记录，已有记录则表明已签到，返回已签到成功的信息，否则创建一个新的MeetingSignIn类型对象，将各项数据赋值给它的各项属性，然后通过jpa框架的sava方法保存到数据库中，返回信息表示签到成功，代码如下：

meetingSignIn.setMeetingId(ms.getId());//会议编号

meetingSignIn.setUserOpenId(openid);

meetingSignIn.setUserCode(u.getCode());//工号

meetingSignIn.setTimestamp(timestamp);//时间

meetingSignIn.setSignInName(userService.findByOpenId(openid).getName());

//姓名

meetingSignInService.save(meetingSignIn);//保存

签到后数据库meeting\_sign\_in表如下图所示：

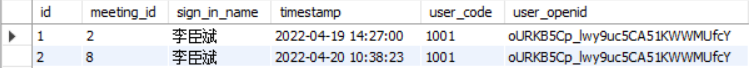


图5.7 meeting\_sign\_in表

## 5.4 预约模块

模块功能介绍：

用户可以在选择会议室和日期后查看对应的预约情况，该界面会显示从8:30开始的以一小时为间隔的所有时间段，若空闲对应时间段则会显示“空闲”，若已被预约，则会显示“已约”及预约人的姓名，并且该部分颜色会发生变化，选择空闲的时间段（可以选择多个时间段），并点击预约按钮，进行预约。此外，预约之后可以通过再次点击该时间段来取消预约，若如此做，该时间段显示会恢复原来的颜色。

5.4.1会议室预约情况图

功能介绍：

展示选中会议室在选中日期中各个时间段的预约情况。

代码流程：

用@RequestMapping注解将请求和处理请求的控制器方法关联起来，建立映射关系。后端通过@RequestParam将对应数据映射到功能方法参数上，从meeting\_room\_usage表中查找所有与该会议室与日期对应的记录放入集合中，创建一个新的JSONArray对象arr，用foreach循环遍历集合中所有记录，在循环中先创建一个JSONObject类型的对象obj，从user表中查找与本次循环中记录对应openid的记录赋值给用户对象u，如果user表没有该记录，则使用lombok的builder方法将u的工号、openid、姓名等各项属性设为空字符串，然后将，u的姓名、openid，以及本次循环中记录中的时间序列、id分别在obj中组成键值对，然后把obj添加到arr中，代码如下：

obj.put("user", u.getName());//预约人姓名

obj.put("openId", u.getOpenId());//预约人openid

obj.put("index", mru.getTimeIndex());//预约会议室时间序列

obj.put("id", mru.getId());//预约会议室编号

arr.add(obj);

循环结束。在创建新的JSONObject对象，将times和arr在其中分别组成键值对。其中times是在前面定义的一个String数组，其中存放从8:30开始的以一小时为间隔的所有时间段，例如times[0]的内容为"8:30-9:30"，代码如下：

obj.put("timeIndex", times);

obj.put("booked", arr);

在前端wxml中根据“booked”的内容显示已约或空闲。

界面如下：

图5.8 预约界面

5.4.2预约

功能介绍：

选择空闲的时间段（可以选择多个时间段），并点击预约按钮，进行预约，再次点击时间段可以取消该时间段的预约。

代码流程：

先将前端数据与后端功能方法的参数建立映射关系，从user表中查询与参数中openid对应的记录，若没有记录，返回错误信息fail。将参数中的indexs按","分割存入String类型数组tmp中，用foreach循环遍历tmp数组中的每个位置的字符串，在循环中从meeting\_room\_usage表中查找会议室名、日期和时间序列与roomid、date、本次循环中的t（转化成十进制整型）都对应的记录，如果没有记录，则进行下一次循环，否则，用lombok自带的builder方法为

MeetingRoomUsage对象mru各项属性赋值，代码如下：

MeetingRoomUsage mru = MeetingRoomUsage.builder()

.meetingRoomId(roomid)//会议室名

.timeIndex(Integer.valueOf(t))//预约时间序列

.useDate(date)//预约日期

.userOpenId(openId)//预约人openid

.build()；

将mru用jpa框架的save方法保存到meeting\_room\_usage表中，并返回成功信息“success”。

运行界面：

图5.9 预约完成

取消预约功能主要通过@RequestParam注解获取前端的预约编号，使用del方法根据预约编号从meeting\_room\_usage表中删除对应的记录，代码如下：

@Query(value="delete from meeting\_room\_usage where id=:id ", nativeQuery = true)//nativeQuery = true 代表的本地查询

public void del(@Param("id")long id);//@Param的作用是给参数命名

## 5.5 信息管理模块

5.5.1查看预约情况

功能：用户选择日期后，界面会显示预约情况列表

代码流程：

编写queryUsages()方法用于从meeting\_room\_usage表中查询所有大于等于当前日期的记录，并按使用日期和时间序列分组排序，存入集合mrus中，queryUsages代码如下：

@Query(value = "select \* from meeting\_room\_usage where use\_date >=CURDATE() order by use\_date, time\_index", nativeQuery = true)

public List<MeetingRoomUsage> queryUsages();

创建一个名为params的HashMap，设置房间编号roomId为空，创建一个存放整型的双向列表timeSlides，然后用foreach循环遍历mrus集合中的每条记录，在循环中，先设置一个空的名为tmp的HashMap，然后将本次循环中的记录的日期赋值给字符串类型的date，tmp通过date“键”从params中获得对应的“值”，若没有，则将tmp初始化，并再次赋值roomId为空字符串，再将timeSlides也初始化。若tmp中本次记录的会议室名对应的值为空，且timeSlides的大小大于0，则在tmp中将此roomId与timeSlides关联;但若timeSlides大小为0，则将本记录的会议室名赋值给roomId，初始化timeSlides,将本次记录的timeindex追加到timeSlides末尾，然后在tmp中将此roomId与timeSlides关联。若tmp中本次记录的会议室名对应的值不为空，那么从tmp中获取本次记录会议室名对应的整型双向列表赋值给timeSlides，再将本次记录的timeindex追加到timeSlides末尾，然后在tmp中将此roomId与这时的timeSlides关联。结束后，在params中将此date与tmp关联。

接下来是会议室预约情况。创建一个名为useSilds的HashMap，其中把日期与使用预约状态相关联。再定义一个HashMap集合convert，设置String类型的集合infos，用以表示该时间段对应的使用情况（已约/空闲），代码如下：

List<String> infos = merge(times, t.getValue())；

//merge方法显示该时间段对应的使用情况（已约/空闲）

再用foreach循环遍历infos集合，在循环中，将本次的元素按","分割存入tmp数组中，将数组中第一个位置和第二个位置字符串中隐含的数字转成10进制整数，在之前的功能中提到的times数组中找到对应位置的时间段，分别赋值给start和end，代码如下：

String[] tmps = info.split(",");

String start = times[Integer.valueOf(tmps[0])];

String end = times[Integer.valueOf(tmps[1])];

会议时间跨度span则用substring方法实现：

String span = start.substring(0, start.indexOf("-")) +"-"+ end.substring(end.indexOf("-")+1);

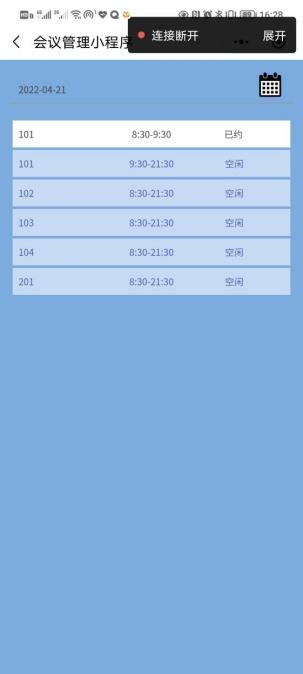
然后将convert获取day映射到的值赋值给roomInfo，若为空，初始化roomInfo。再把roomInfo中会议室名映射的值放入finalInfo集合中。若集合为空，初始化该集合，否则将span和flag追加到集合末尾（flag即为预约或空闲），在roomInfo中把会议室名与finalInfo集合关联，在convert中把日期和roomInfo关联，最后返回convert。代码如下：

finalInfo.add(new String[]{span, flag});

roomInfo.put(rId, finalInfo);

convert.put(day, roomInfo);

运行界面：

图5.10 预约列表

5.5.2查看签到情况

模块功能介绍：

用户可以查看自己各个会议的签到情况，管理员可以额外查看签到名单，签到名单的内容是工号、姓名和签到状态。

代码流程：

创建一个StringBuffer对象sb，使用 StringBuffer 类时，每次都会StringBuffer 对象本身进行操作，而不是生成新的对象。从meeting\_sign\_in表中查找对应全体会议的标号，把所有符合条件的记录存入s集合中。定义两个集合signed和unsigned用以存放签到人员和未签到人员，foreach循环遍历人员总名单，设flag初始值为false，将每次获得的工号与meeting\_sign\_in表中的工号依次比较，若有相同的，令flag为true，将对应的用户追加到signed已签到列表的末尾，否则，追加到unsigned未签到列表末尾。代码如下：

for(User u : uList.getList()){

String code = u.getCode();//lombok

boolean flag = false;

for(MeetingSignIn si : s){

if( si.getUserCode().equals(u.getCode())){

flag = true;

signed.add(u);

}

}

if( !flag ){

unSigned.add(u);

}

}

再用两个foreach循环分别规定名单格式。用自己编写的DownloadFromString类的exec方法实现名单的显示。exec方法中，先设置response对象的头类型，代码如下：

response.setHeader("content-type", "text/plain");

//setHeader方法总是用新值去替换旧值

response.setContentType("text/plain");

//让浏览器以普通文本TXT解析

创建空的BufferedInputStream对象bis，使用 Content-Disposition告诉浏览器这个文件的名字和类型，代码如下：

response.setHeader("Content-Disposition", "attachment;filename=" +java.net.URLEncoder.encode("会议签到表.txt", "UTF-8"));

//java.net.URLEncoder.encode将内容进行编码

然后使用平台的默认字符集将其编码为字节序列，并将结果存储到新的字节数组中，将读取的字节数赋值给i，只要未到末尾，就继续写入输出流中，再将i本次读取的字节数再赋值给i。

截图如下：

图5.11 签到名单

# 结论

本小程序作品使用了springboot+jpa作为系统的总体架构。使用的主要编程语言是Java语言，后端进行数据存储的数据库是使用灵活方便的Mysql，前端展示页面通过微信开发者工具编写，包括js、wxml、wxss，优化了页面的整体布局和实现。整体后端的开发平台为IDEA，这个平台拥有很多功能强大的插件，界面简洁美观，十分适合项目的开发。

通过查阅资料以及运用这些开发工具，我终于完成了会议室预约及签到的微信小程序。在该小程序中，普通用户绑定工号登录后，可以实现扫码签到，会议室预约、查询预约情况以及签到情况等功能，而管理员拥有额外的功能，例如：发布会议、关闭签到、查看签到名单等操作。这些操作的逻辑基础是通过在前端页面输入信息或者点击按钮，调用后台的对应操作方法，后端的Controller进行事件处理，几乎所有功能都要调用数据库工具来实现对数据库中信息的增删改查。

本系统的最大特色是将不便庞大的会议预约系统以轻便小程序的方式实现，可以随时通过手机进行操作，降低了硬件要求，更加简单方便，适合现在的节奏，并且本系统中部分功能使用了许多Hashmap，大大提高了搜索查询效率。

由于本学期的很多时间都用在研究生复试上了，所以没有太多的精力去开发毕业设计，再加上对微信小程序开发的了解比较浅薄，导致了最后完成的功能相对简单。在开发过程中我也遇到了很多难题，如微信开发者工具前端与后端数据交互不同问题，app.js中的baseurl的设置问题以及微信开发工具中的真题调试问题等等。通过解决这些难题提高了自己的动手实践能力。在编写代码的过程中，一定要养成经常写注释的好习惯，以免在下一次工作时候可能会忘记之前做过了哪里，后面通过勤写代码注释让我能够很好的回忆自己上一次的进度以及工作内容，对我完成系统的开发起了很大的帮助。

通过本次毕业设计让我更加了解了整个系统从前端到后端再到数据库的开发流程。我编写代码的能力和应对困难的能力也被很好的锻炼了。在开发过程中难免会遇到许多技术难题，通过请教我的朋友和指导老师以及自己查阅资料逐一解决这些难题的过程让我获得了很大的成就感。因为自己的时间比较紧迫，再加上能力不足，导致很多想法还没有能够得到实现，如智能推荐合适的会议室等功能。在今后我会更加努力地学习技术，提高自身水平，将开发工作做得更好。

参考文献

[1]李志忠.基于ASP框架的会议室预约系统研究[J].科技资讯,2010(24):162.

[2]葛霁霆.基于ASP.NET技术的会务管理系统的设计与实现[D].天津:天津大学,2015.

[3]陈瑶.基于Node.js高并发web系统的研究与应用[D].成都:电子科技大学,2014.

[4]王飞. XX公司管理信息系统应用研究[D].北京邮电大学,2013.

[5]刘泉.基于生成式对抗网络的图像压缩方法研究[D].西安:西安电子科技大学,2019.

[6]刘送杰,郭德福,彭伟.基于微信小程序的CTS信息系统开发与应用[J.涟钢科技与管理，2019(6):27-29.

[7]王政嘉,孙志敏,周志军.开放实验室预约系统的应用研究[J].河北农业大学学报:社会科学版，2020,22(1):119-123.

[8]李双远,谈国胜."互联网+"高校实验室信息管理平台的构建与研究[J].现代经济信息，2019(9):424.

[9]明艳,李强,王敏琦.等.实验室开放预约信息管理系统的开发与实践[J].数字通信，2014,41(5):91-95.

[10]王玉洋.基于微信小程序的移动学习平台环境构建与系统设计开发[D].南京:南京大学，2018.

[11]王西芳,王西平,杜慧芳.等.实验教学示范中心实验预约系统的设计与应用[J].实验室研究与探索，2016,35(5):158-161.

[12]袁浩铭,廖昌军,王以伍.等.高校实验教学中心开放预约管理系统的开发与实践[J.教育现代化，2017,5(18):144-146.

[13]丁金婵.企业会议室预约管理平台的设计与实现[J].软件,2020,41(11):91-93.

[14][逯连静](https://kns.cnki.net/kns8/Detail?sdb=CJFD&sfield=%e4%bd%9c%e8%80%85&skey=%e9%80%af%e8%bf%9e%e9%9d%99&scode=000024465258&acode=000024465258" \t "https://kns.cnki.net/kns8/defaultresult/knet),[陈国荣](https://kns.cnki.net/kns8/Detail?sdb=CJFD&sfield=%e4%bd%9c%e8%80%85&skey=%e9%99%88%e5%9b%bd%e8%8d%a3&scode=000028382211&acode=000028382211" \t "https://kns.cnki.net/kns8/defaultresult/knet), [徐琳君](https://kns.cnki.net/kns8/Detail?sdb=CJFD&sfield=%e4%bd%9c%e8%80%85&skey=%e5%be%90%e7%90%b3%e5%90%9b&scode=000024294122&acode=000024294122" \t "https://kns.cnki.net/kns8/defaultresult/knet).新媒体环境下科技期刊微信小程序应用探究[J]传播与版权. 2022,(04)

[15]蒋佳龙.基于微信小程序的实验室服务平台设计与实现[J]信息技术与信息化. 2022,(03)

[16]杜雨荃,王晓菊,田立勤.基于微信小程序的网上购物系统的设计与实现[J].网络安全技术与应用,2022(04):62-64.

[17]Kassahun Ayalew,Bloo Robbin,Catal Cagatay,Mishra Alok. Dairy Farm Management Information Systems[J]. Electronics,2022,11(2).

[18]Shengli Chen. Application of Management Information System for College Apartments[J]. Journal of International Education and Development,2021,5(9).

[19]New Meeting Room Solution for Microsoft UC Deployments[J]M2 Presswire2019.

[20]Miao Peng. University Canteen Supervision and Management System Based on Wechat Applet[J]. International Journal of Frontiers in Sociology,2021,3.0(11.0).

致谢

时光荏苒，日月如梭。一眨眼的功夫，我的四年的大学生活逐渐来到了尾声。遥想这四年大学时光，我对教导我的老师们，我的同学们以及我遇见的善良的人们表达我最深的感激与敬意，经历了许多事情让我成长了很多，也成熟了很多。由衷地感谢陪伴我走过这几年的人们，他们是我在这段青葱岁月遇到的最可爱和可敬的一群人。

首先，我发自内心地感谢悉心教导我的恩师们，你们的谆谆教诲让我深深地明白“明德格物，立己达人”的校训的深刻内涵。特别要感谢我的毕设指导教师—韩进老师，他不仅非常体贴学生，不让毕业设计占用研究生复试的准备时间，而且对于我请教的问题耐心的回答，讲解通俗易懂，帮助我解决了许多难题。也因为韩进老师的宽容，才让我能够投入全部精力准备研究生复试，最后成功上岸。

其次，我要感谢和我一起生活的舍友们，和你们在一起生活让我觉得自在舒适，你们对我也很包容体谅。不仅包容我的缺点，而且经常帮助我；；其次我要感谢考研期间陪我一起奋斗的朋友，李恒泰、吴爽、吴辰旭、吴王雨辰，是你们陪我熬过了最困难的一段时光，才造就了现在的我，没了你们的陪伴，我可能会半途而废。

最后，感谢一路过来的自己没有放弃，努力奋斗。我也希望自己在今后的生活中能够保持乐观向上的心态，不怕困难，勇敢拼搏！