



系统设计&数据库设计

待就业六人组

主讲人：王铭君、康小榕、黎焕明



Rethink & Improve



根据分析类图模型，系统能够根据用户简历给予职位推荐吗？



简历模板、职位分类等这些信息在类图中如何体现？



管理员端有没实现？



职位推荐如何实现，采用什么算法？



涉及优化匹配算法的参数在哪个类中体现？





类图改进

主讲人：王铭君



参与者类图

信息管理类图

投递简历类图

私信类图

登录子系统类图

智能推送类图

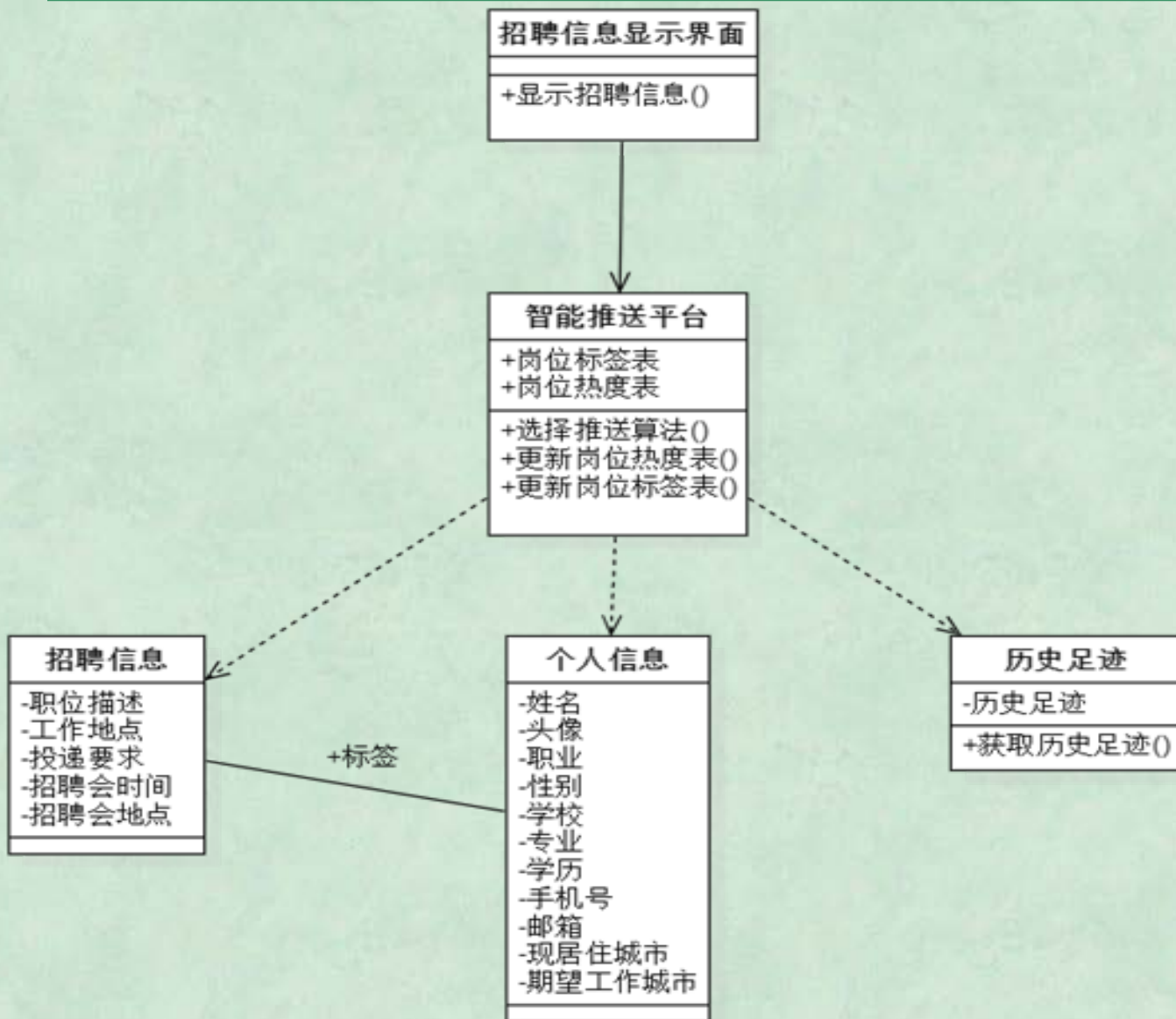
信息查询类图

审核简历类图

我们将OOA的大类图按照模块细分以上八个功能类图

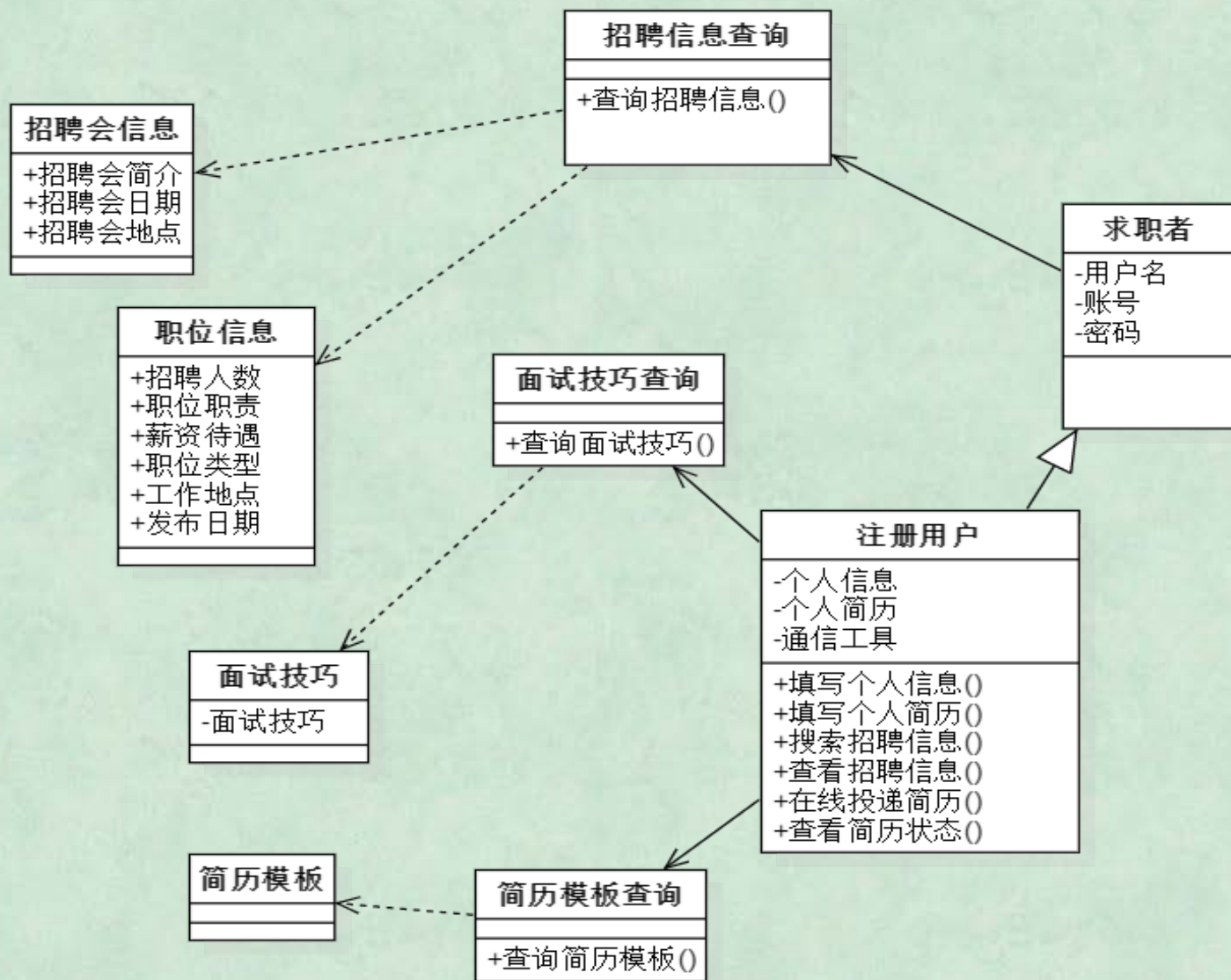


智能推送类图





信息查询类图





| 体系结构设计 |

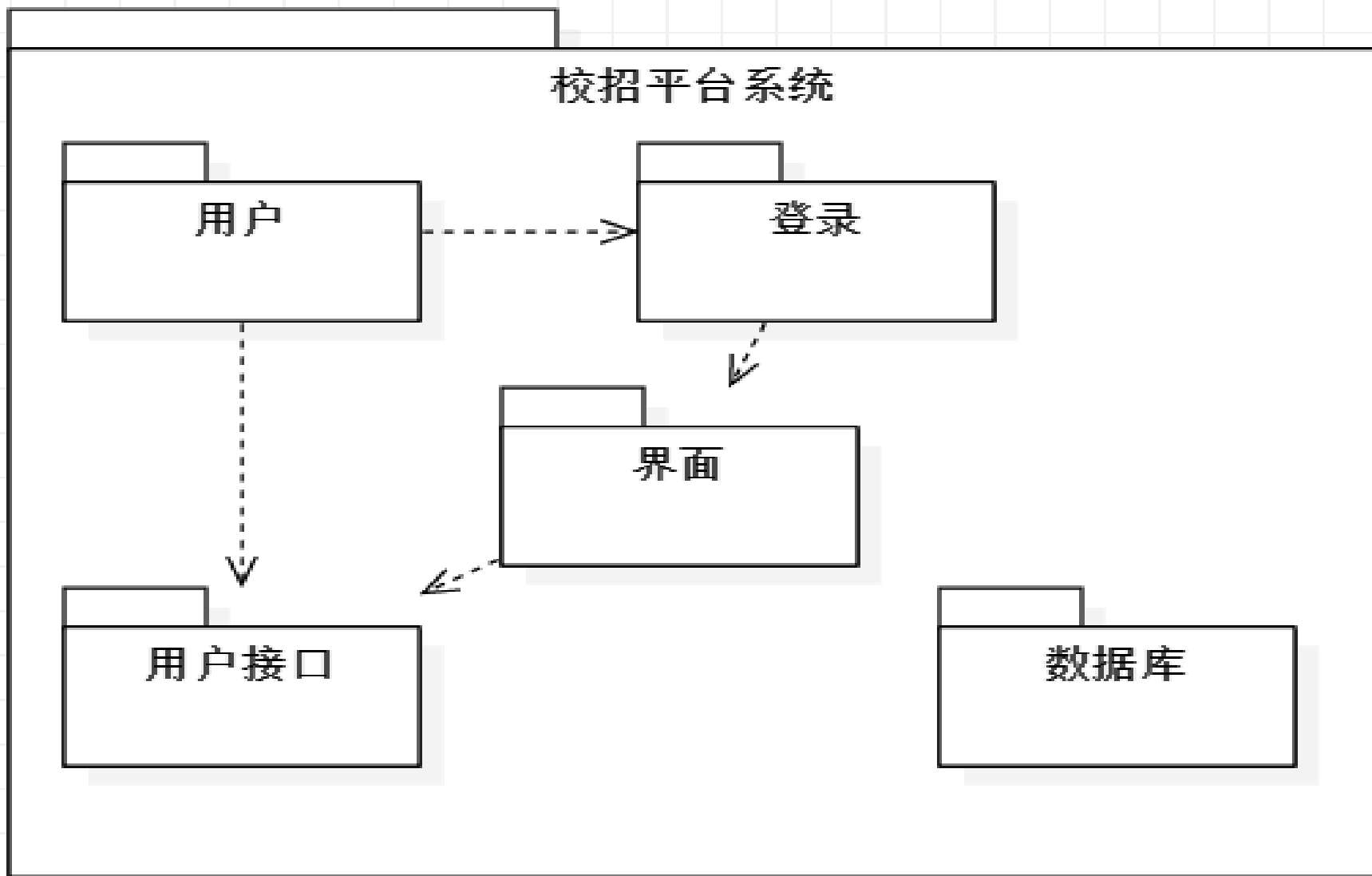
主讲人：康小榕



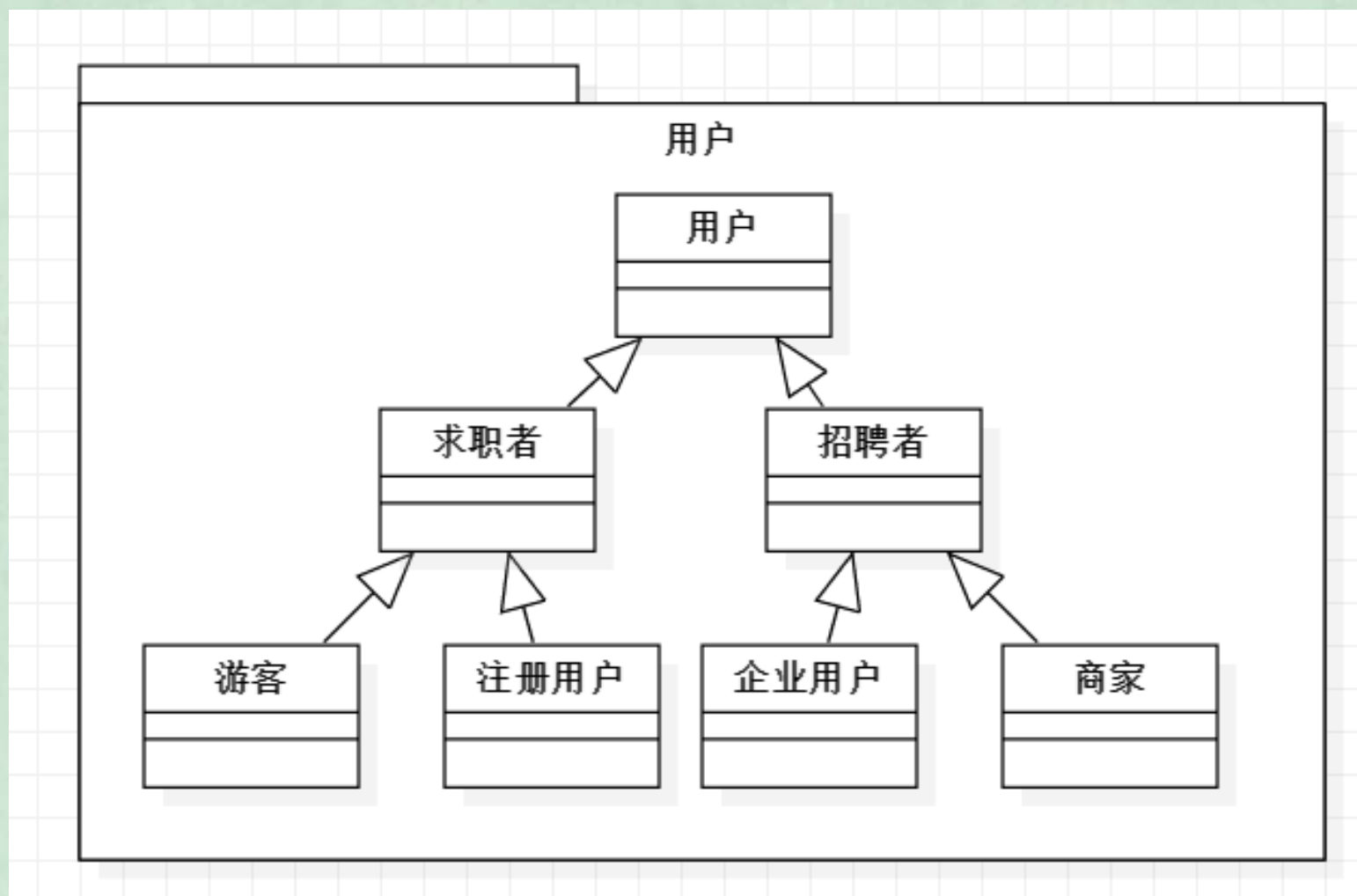
系统分析

系统包图

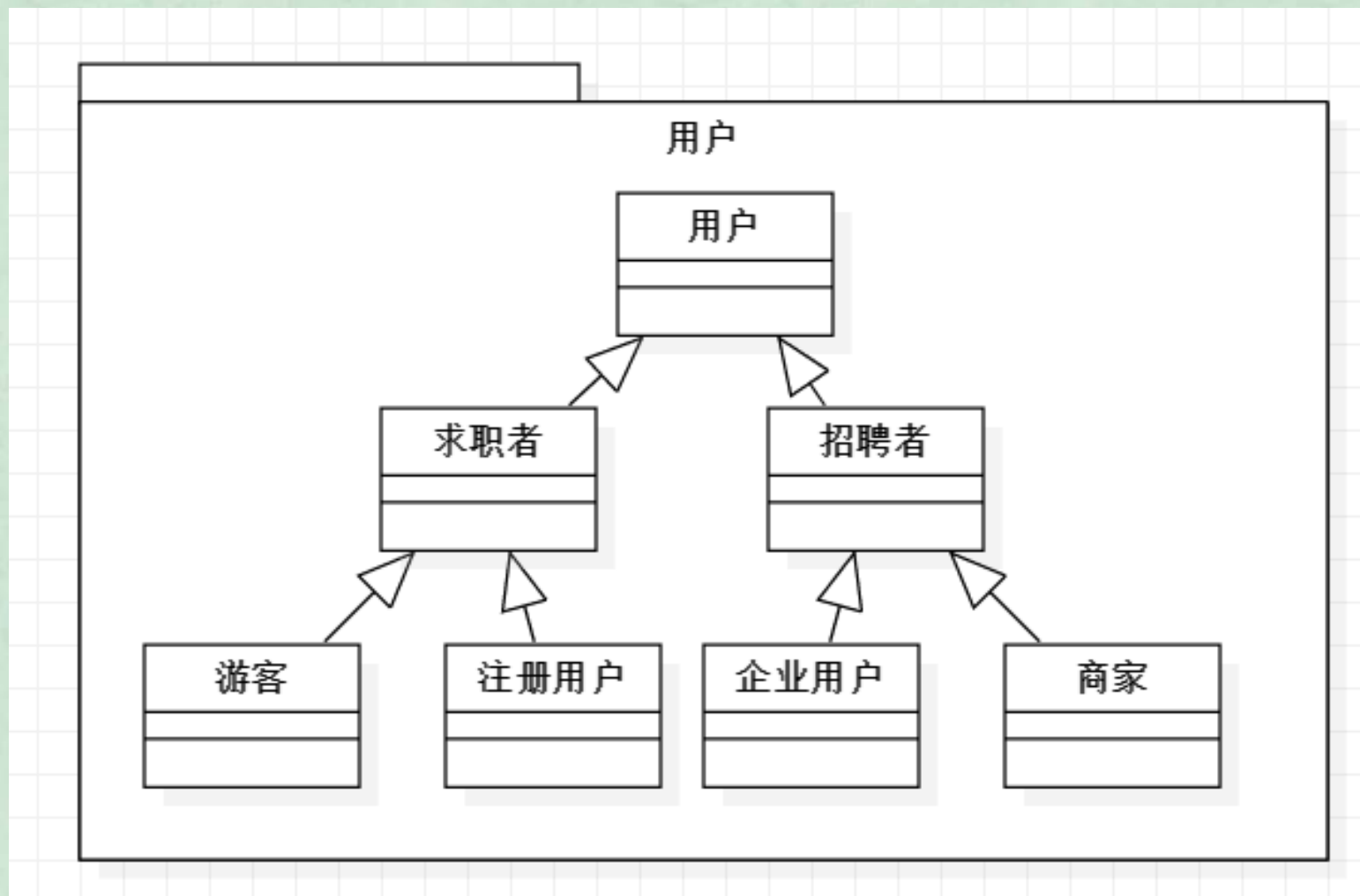
类图体现了校招平台在微观上的静态结构，但由于整个系统内容繁多，较为庞大，使用类图分析将大大增多工作量，因此我们从宏观上对整个系统进行分析，将之划分为互有联系又相对独立的几部分，如右图所示。



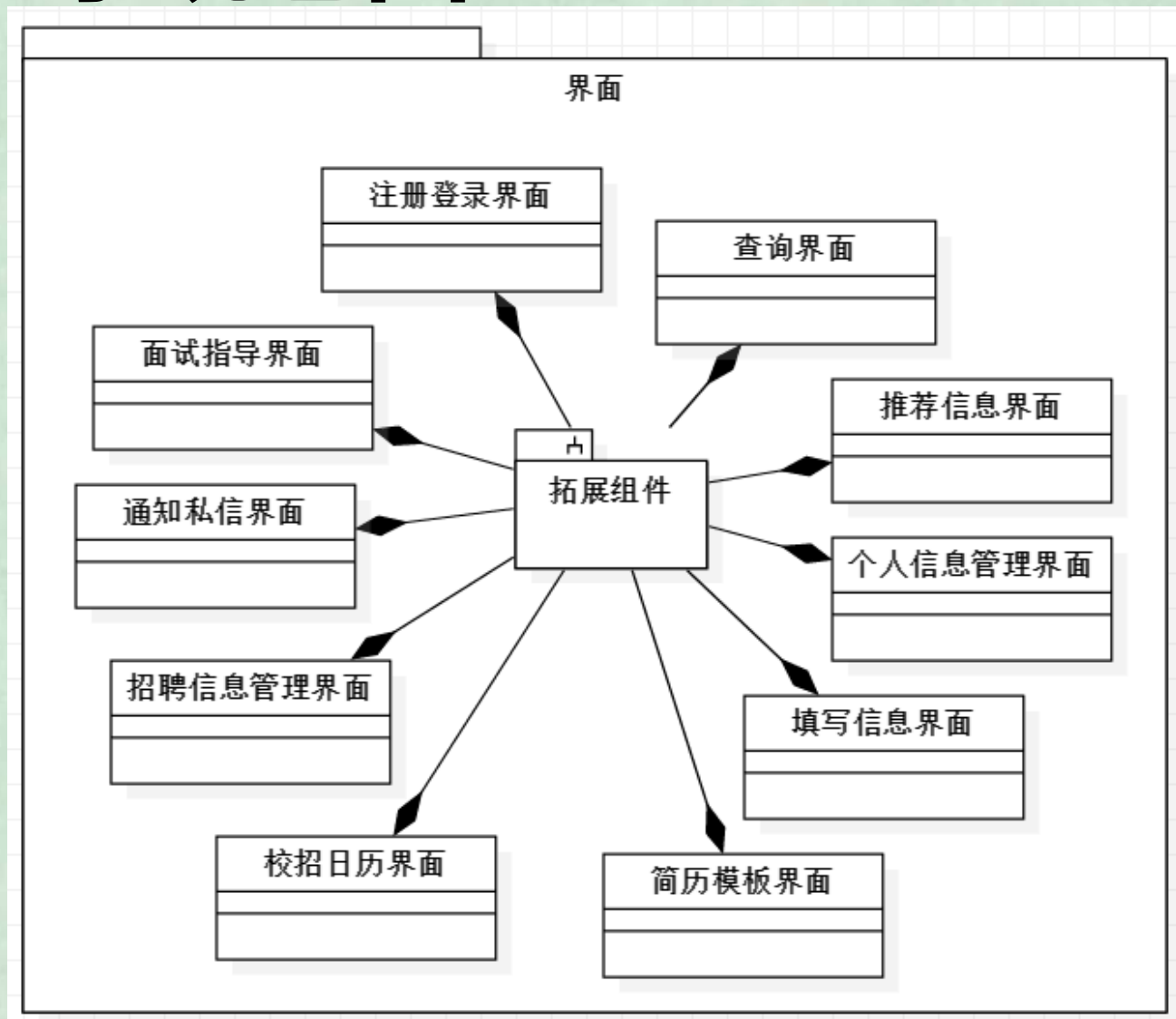
登录—子系统包图



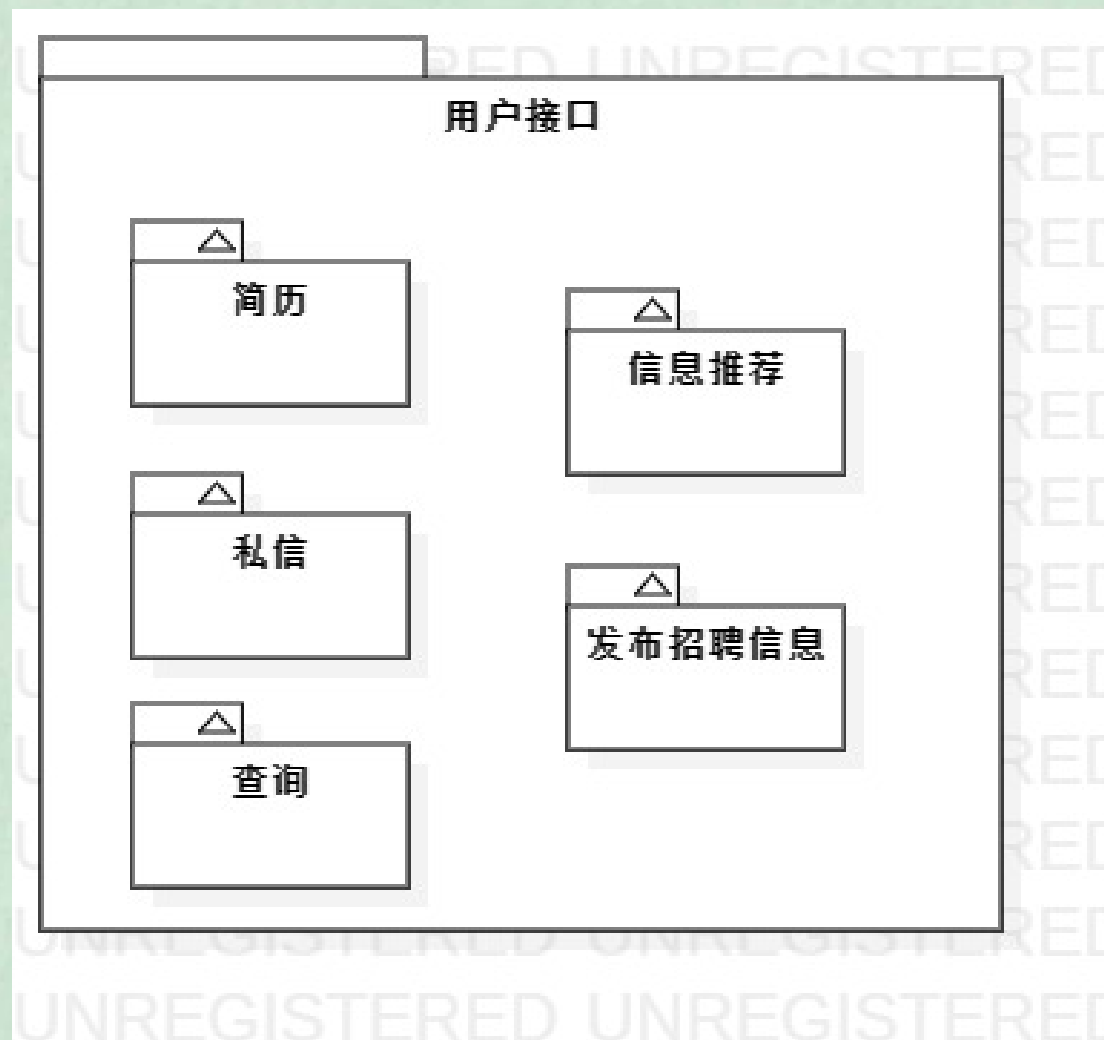
用户—子系统包图



界面—子系统包图



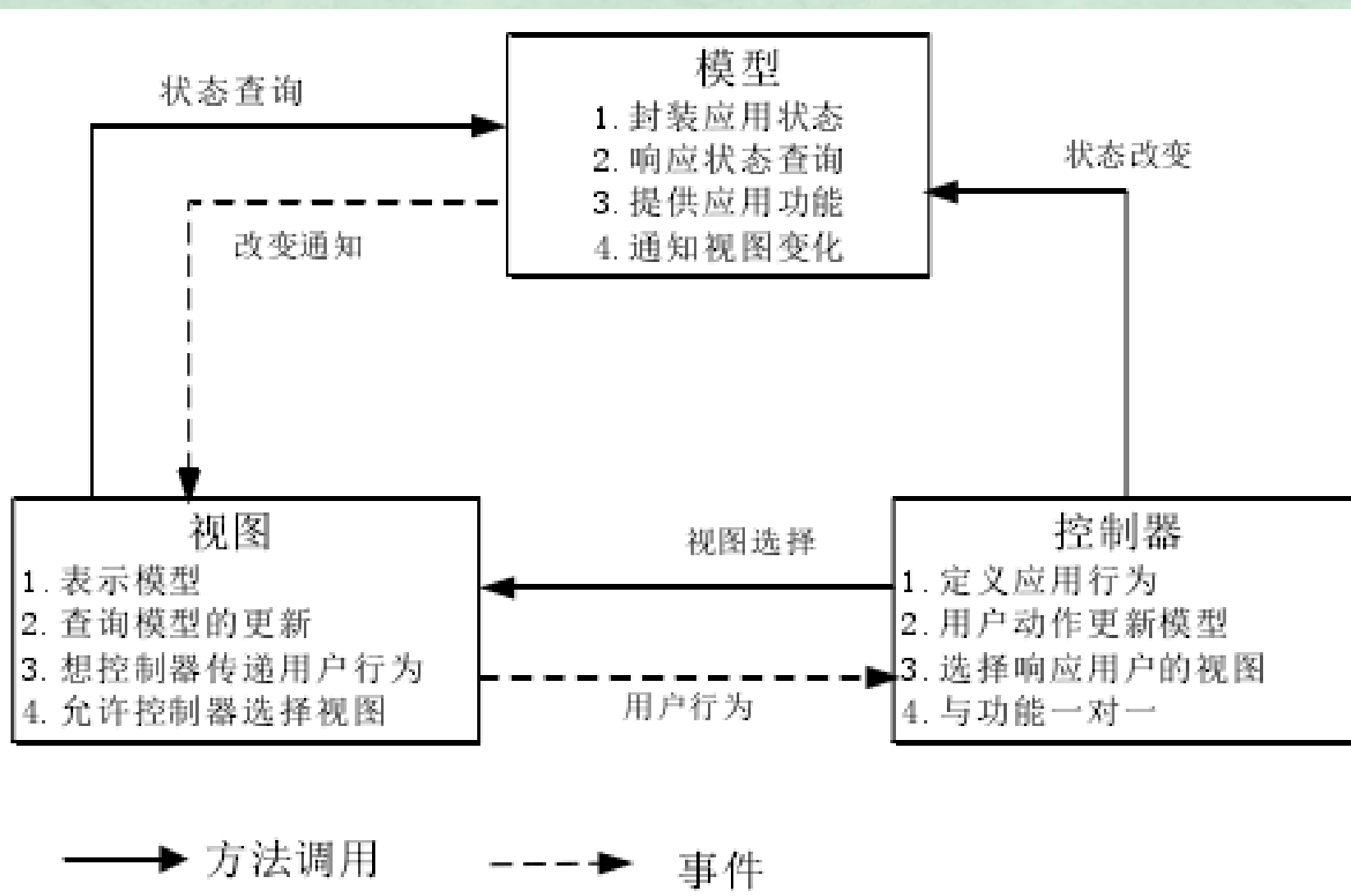
接口—子系统包图



M V C 模式

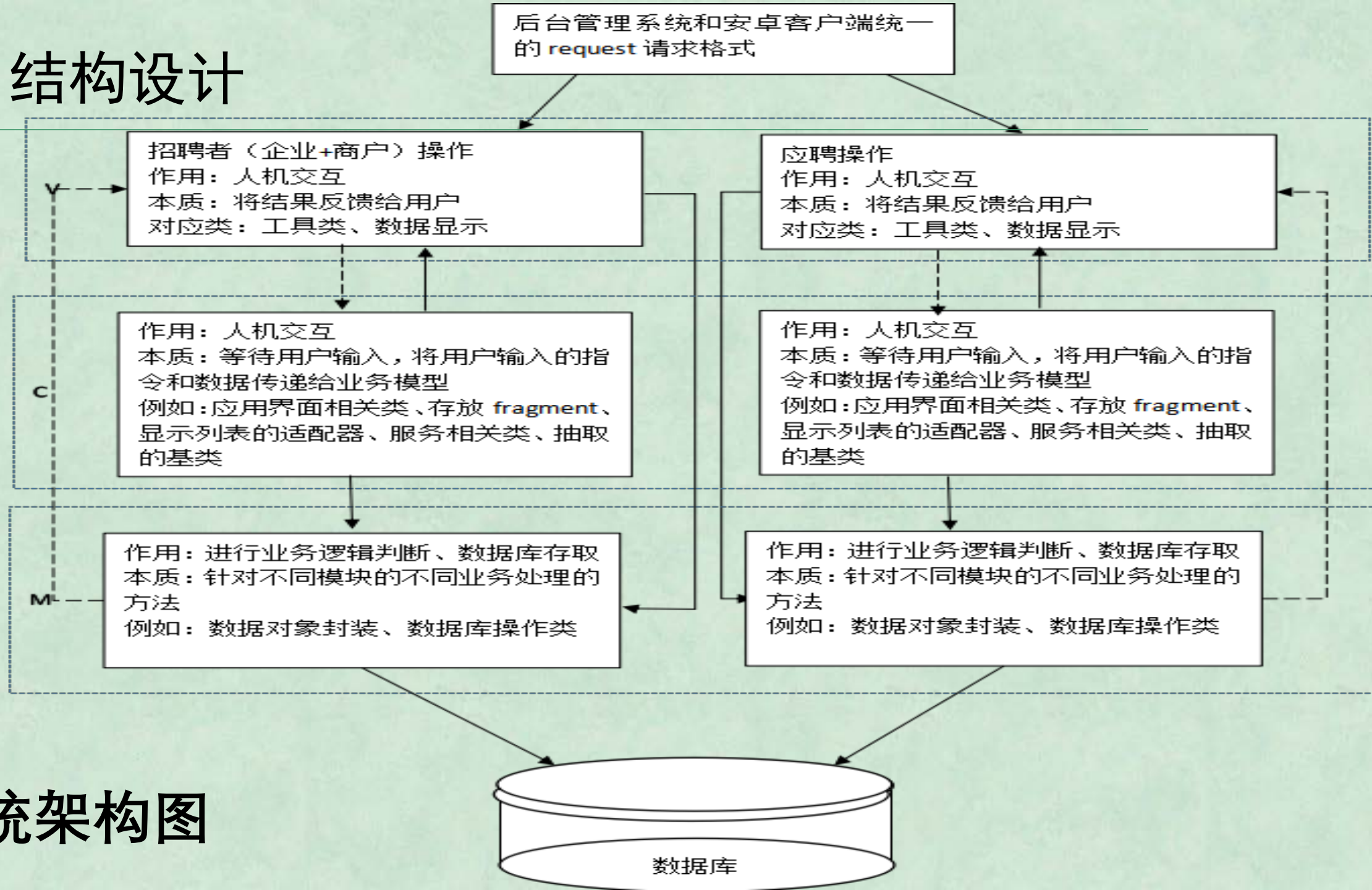
- 进行了以上分析之后，本平台根据小组成员过去的开发经验，决定采用MVC框架模式。MVC采用单一入口模式进行项目部署和访问，准确处理好模块与模块之间的联系。
- MVC包括三个部分:模型、视图、控制器。
- **模型**：定义相应的控制器编写算法等等实现程序功能、实现具体的数据管理和数据库设计
- **视图**：实现静态的图形界面设计
- **控制器**：定义后使用视图和模型，负责通信、转发请求、响应请求

MVC通用的模型设计





结构设计



系统架构图



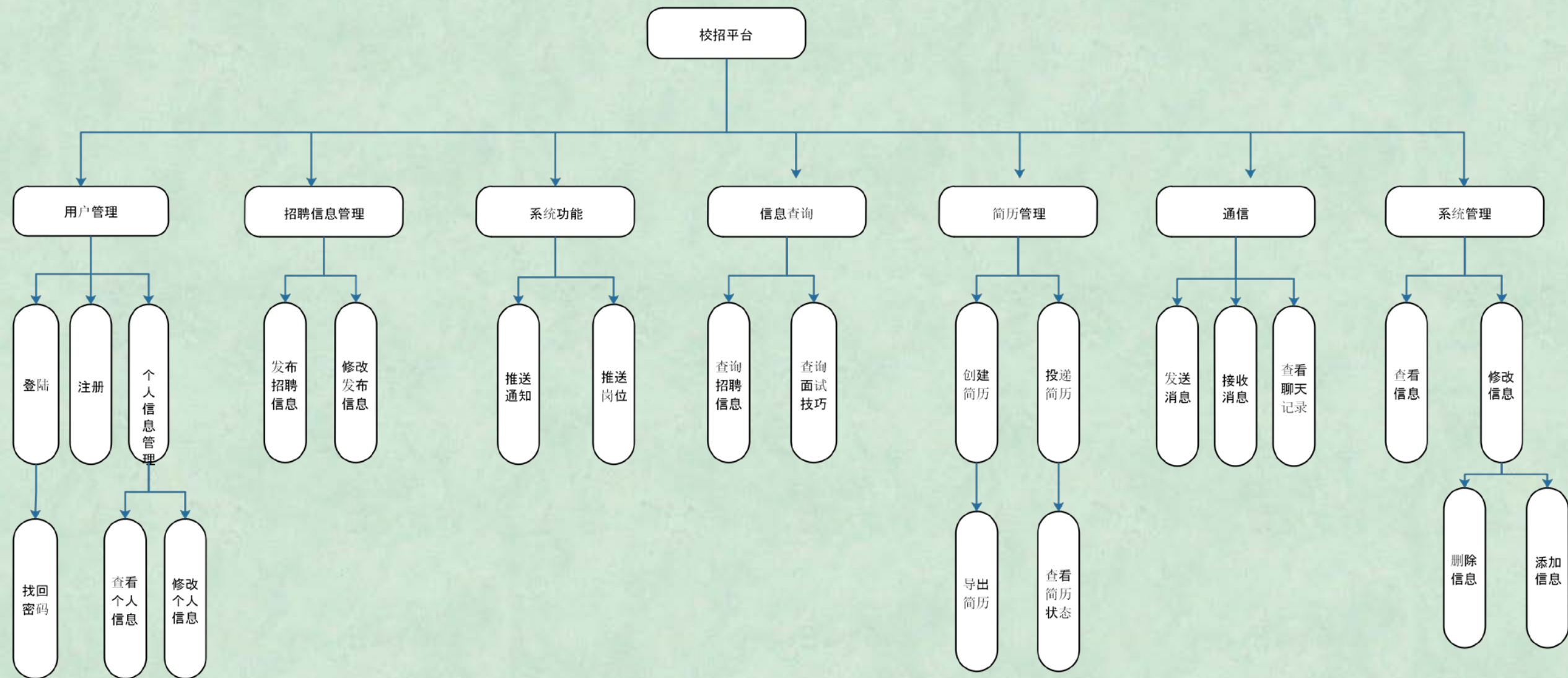
| 功能模块层次 |

我们根据以上以上分析，将系统进行了功能模块层次的划分

主讲人：康小榕



功能模块层次





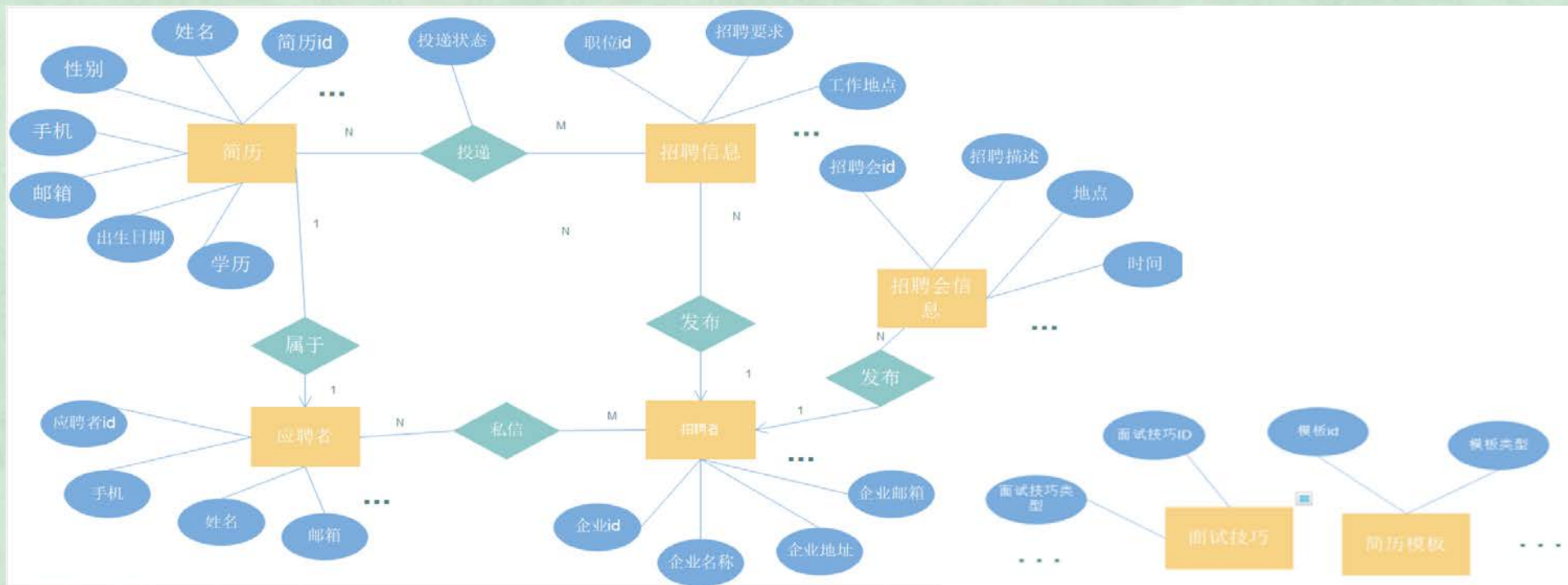
数据库设计

概念结构设计、逻辑结构设计、表结构设计、规范化

主讲人：黎焕明



概念结构设计



E-R图

根据前面需求分析的结果可以得到上图，简历和招聘信息使用投递关系联系在一起，简历和应聘者使用属于关系联系在一起，招聘信息和招聘会使用发布关系联系在一起，招聘者和应聘者使用私信联系在一起。



概念结构设计



根据《数据库系统概论》7.4.1 E-R图向关系模型的转换（p231）可以得到以下模式：

公司（公司id，电话号码，密码，公司名，头像链接，邮箱，企业描述，是否通过审核）

学生（学生id，密码，电话号码，用户名，头像链接，邮箱，性别，学校，专业，职业，当前城市，期望城市）

招聘会（招聘会id，公司id，时间，地点，面向人群，描述）

招聘信息（招聘信息id，公司id，时间戳，岗位描述，联系人及联系方式，任职资格，工作地点，投递要求，工作职责，薪酬福利，招聘或者兼职，有效）

简历（简历id，student_id，电话号码，用户名，简历头像链接，邮箱，性别，最高学历，职业，当前城市，期望城市，教育背景，证书，项目经历，实践经历，自我评价，简历状态）

简历投递（简历投递id，公司发布的职位信息的id，用户id，简历id，投递状态）

聊天室（学生id，hr_id，聊天室id，发送方）

聊天记录（记录id，聊天室id，内容，时间戳）



实体



学生

学生可以创建简历，投递简历，浏览信息等



企业

企业可以发布招聘信息，修改招聘信息，审核



简历

简历被学生创建，被投递到招聘者中



招聘信息

招聘信息由公司创建，并联系招聘公司与招聘记录



简历投递记录

投递记录由学生产生，并联系简历与招聘信息



聊天室

联系企业与学生



聊天记录

在聊天室内聊天记录，间接联系学生与企业



实体：学生、企业



(1) student 表

字段名	数据类型	空	默认	注释
student_id	char(128)	否	无	随机生成的主键
passwd	char(32)	否	无	密码
telephone	char(14)	否	无	电话号码
user_name	char(20)	否	无	用户名
head_url	varchar(256)	否	无	头像链接
email	char(32)	否	无	邮箱
sex	tinyint(4)	否	无	性别
school	char(64)	否	无	学校
specialty	varchar(64)	否	无	专业
occupation	varchar(32)	否	无	职业
present_city	varchar(64)	否	无	当前城市

(2) 企业信息表

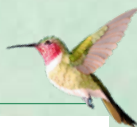
字段名	数据类型	空	默认	注释
company_id	char(128)	否	无	随机生成的主键
telephone	char(14)	否	无	电话号码
passwd	char(32)	否	无	密码
company_name	varchar(50)	否	无	公司名
head_url	varchar(256)	否	无	头像链接
	char(32)	否	无	邮箱
description	text	否	无	企业描述
status	tinyint(4)	否	无	是否审核

企业实体主要有上图所示属性，主键是企业id

学生实体主要有上图所示属性，主键是student_id



实体：招聘信息、简历



(3) 招聘职位信息表

字段名	数据类型	空	默认	注释
<u>publish_time</u>	timestamp	否	当前时间	时间戳
<u>company_id</u>	char(128)	否	无	公司 id
description	text	否	无	岗位描述
<u>recruitment_id</u>	int(11)	否	无	主键
contact	varchar(32)	否	无	联系方式
qualifications	varchar(128)	否	无	任职资格
location	varchar(64)	否	无	工作地点
<u>delivery_request</u>	varchar(64)	否	无	投递要求
duty	varchar(64)	否	无	工作职责
salary	varchar(64)	否	无	薪酬福利
type	int(11)	否	无	招聘类型
validate	int(11)	否	无	有效

(4) 简历信息表

字段名	数据类型	空	默认	注释
resume_id	int(11)	否	自增	简历 id
user_id	char(128)	否	无	外键
telephone	char(14)	否	无	电话号码
user_name	char(20)	否	无	用户名
head_url	varchar(256)	否	无	头像链接
email	char(32)	否	无	邮箱
sex	tinyint(4)	否	无	性别
highest_education	tinyint(4)	是	NULL	最高学历
occupation	varchar(32)	否	无	职业
present_city	varchar(64)	否	无	当前城市
expected_city	varchar(64)	否	无	期望城市
degree	varchar(1024)	否	无	教育背景
certificate	varchar(1024)	否	无	证书
project_experience	varchar(2048)	否	无	项目经历
practical_experience	varchar(2048)	否	无	实践经历
self_evaluation	varchar(128)	否	无	自我评价
resume_status	int(11)	否	无	0 代表未投递, 1 代表已投递,

招聘信息实体主要有上图所示属性，主键是招聘信息id，外键公司id，即招聘信息所属公司。

简历信息实体主要有上图所示属性，主键是简历id，外键是学生id，即简历所属对象。



实体：简历投递记录



字段名	数据类型	空	默认	注释
resume_delivery_id	int(11)	无	自增	简历投递 id
recruitment_id	int(11)	无	无	招聘信息的 id
resume_id	int(11)	无	无	简历 id
delivery_status	int(11)	无	0	投递状态

简历投递记录主键是简历id，外键包括用户id，简历id，招聘信息id。

因为没什么经验，量力而行，所以α版本暂时没有添加聊天功能和面试技巧等，放在了β版本来实现。



规范化



(5) 简历投递表

字段名	数据类型	空	默认	注释
resume_delivery_id	int(11)	无	自增	简历 id
recruitment_id	int(11)	无	无	招聘信息的 id
user_id	char(128)	无	用户 id	用户 id
resume_id	int(11)	无	无	简历 id
delivery_status	int(11)	无	0	投递状态

字段名	数据类型	空	默认	注释
resume_delivery_id	int(11)	无	自增	简历投递 id
recruitment_id	int(11)	无	无	招聘信息的 id
resume_id	int(11)	无	无	简历 id
delivery_status	int(11)	无	0	投递状态

简历投递表用户id依赖于简历id，简历id依赖于简历投递的id，即简历投递表中包含了简历表中的外键（非关键属性），产生了传递依赖，然后我们把用户id删除之后就消除了传递依赖。

重复以上操作，我们能认为我们的数据库表设计达到了第三范式要求。



| 验收验证标准 |

主讲人：王铭君



验收验证标准



1.系统体系结构需满足MVC设计模式

MVC模式（Model-View-Controller）是[软件工程](#)中的一种[软件架构](#)模式，把软件系统分为三个基本部分：模型（Model）、视图（View）和控制器（Controller）。

优点：

- 1.低耦合性：视图层和业务层分离，这样就允许更改视图层代码而不用重新编译模型和控制器代码，
- 2.高重用性和可适用性
- 3.较低的生命周期成本
- 4.快速的部署
- 5.可维护性
- 6.有利于软件工程化管理





验收验证标准



2.数据库设计需满足第三范式

第一范式（1NF）：强调的是列的原子性，即列不能够再分成其他几列。

第二范式（2NF）：首先是 1NF，另外包含两部分内容，一是表必须有一个主键；二是没有包含在主键中的列必须完全依赖于主键，而不能只依赖于主键的一部分。

第三范式（3NF）：首先是 2NF，另外非主键列必须直接依赖于主键，不能存在传递依赖。即不能存在：非主键列 A 依赖于非主键列 B，非主键列 B 依赖于主键的情况。

优点：

在[数据库设计](#)的时候满足范式要求可以很大限度的合理的处理数据，减少数据的冗余，即多余的数据





验收验证标准



2.数据库对不同用户要有明确的权限划分

游客：只有查看公开数据的权限

求职者：游客的权限+投递简历的权限

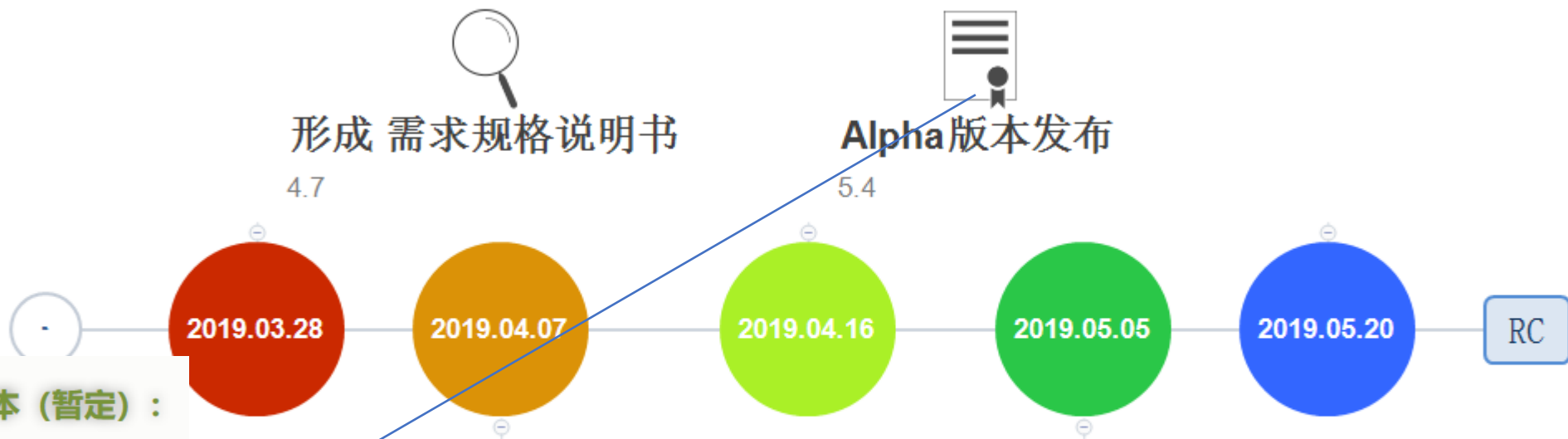
招聘者：游客的权限+发布招聘消息+审核求职信息的权限

系统管理员：具有系统提供的一切权限





项目进度表



Alpha版本 (暂定) :

企业:

- 登录注册
- 信息管理
- 发布信息
- 审核信息

用户

- 登录注册
- 简历管理
- 查看信息
- 搜索、筛选信息
- 报名岗位
- 推送消息

Beta版本 (暂定) :

用户:

- 管理员web端 (管理系统数据)
- 增加个性简历功能 (提供简历模板)
- 增加求职技巧功能 (提供职场干货)
- 增加投递反馈功能 (查询简历投递状态)
- 优化智能推送、智能匹配算法 (更加精准的推送)
- 增加热门岗位排行榜 (随时掌握职场动态)