



系统设计&数据库设计

待就业六人组

主讲人：王铭君



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



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

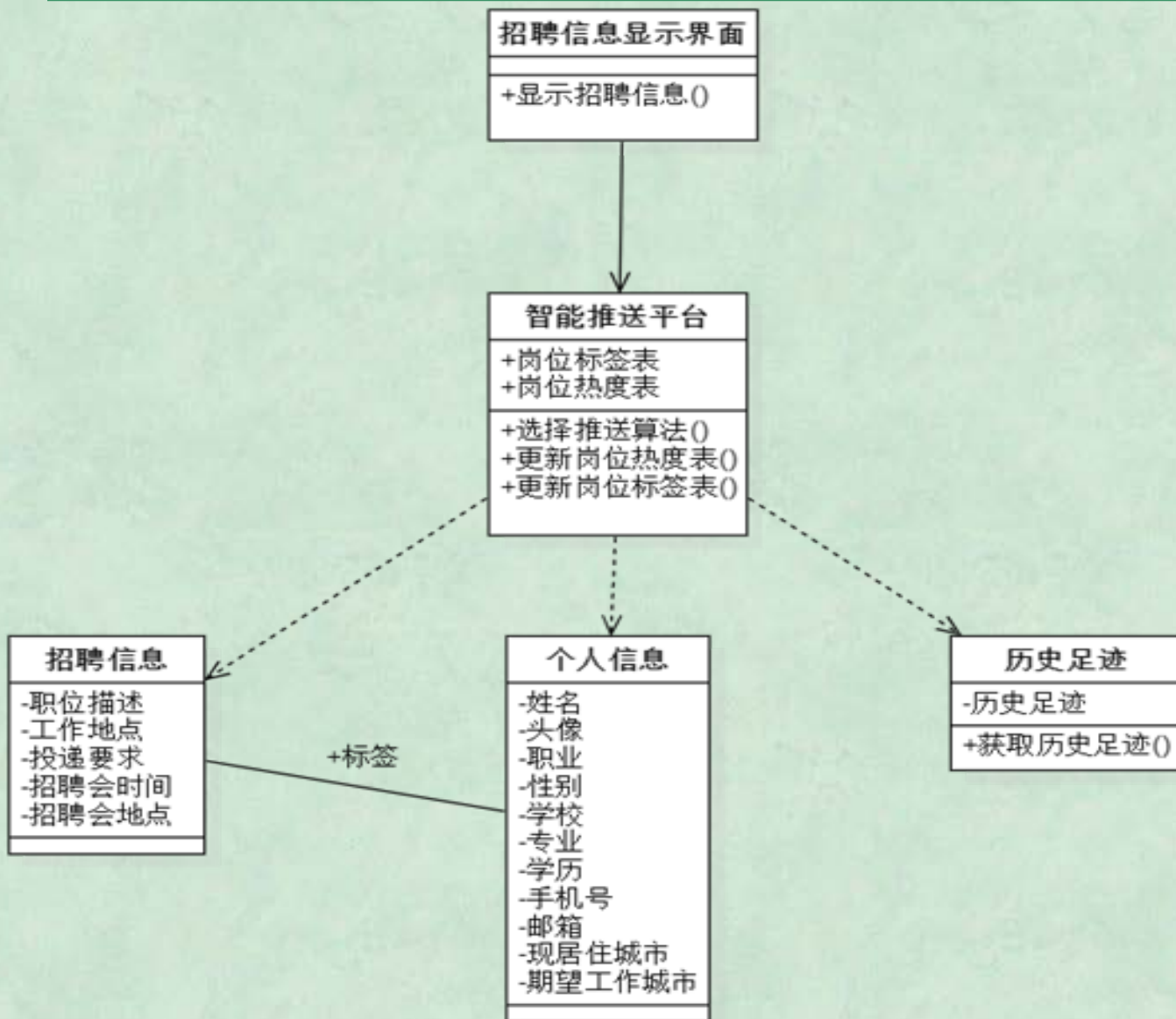
信息查询类图

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

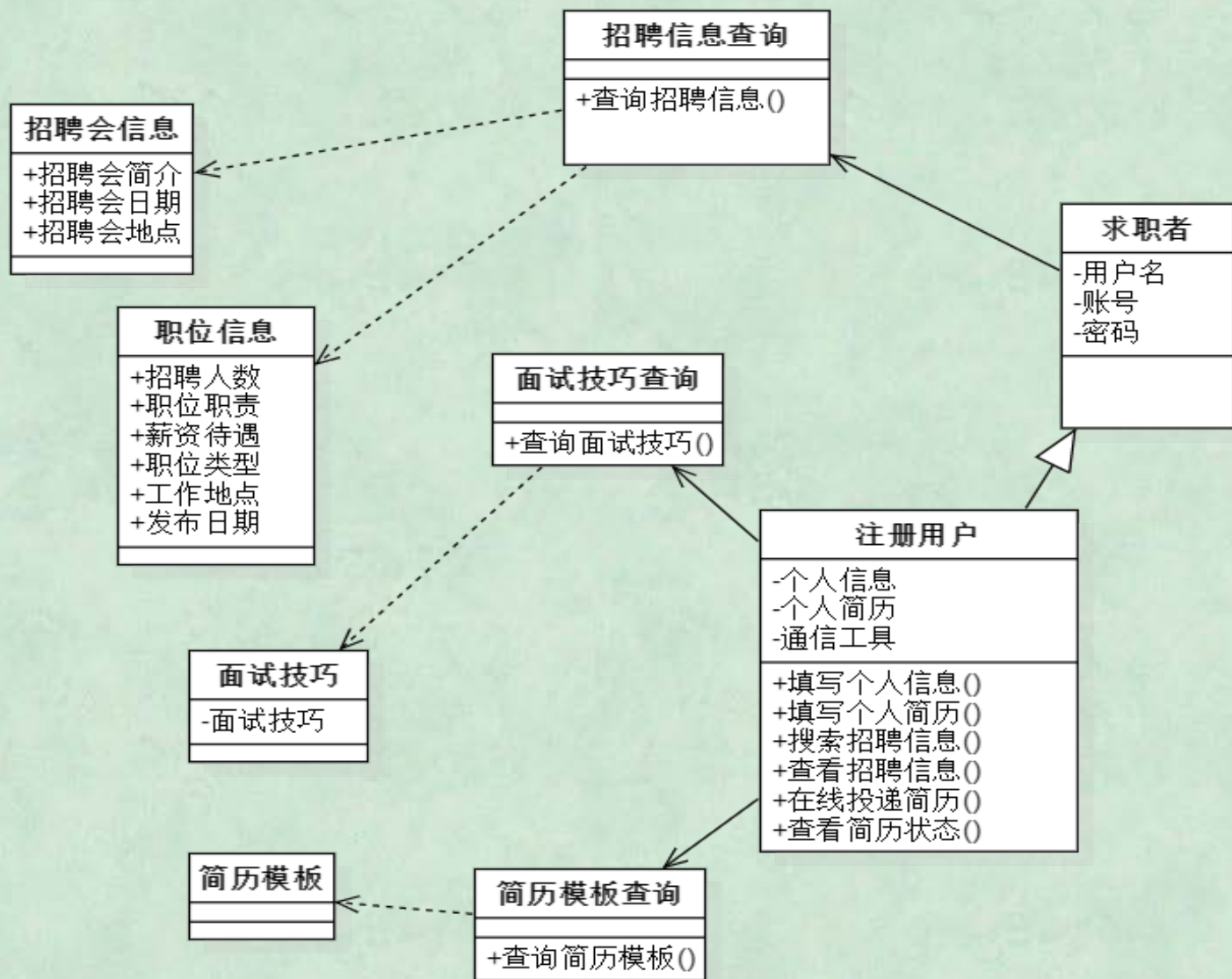


智能推送类图





信息查询类图





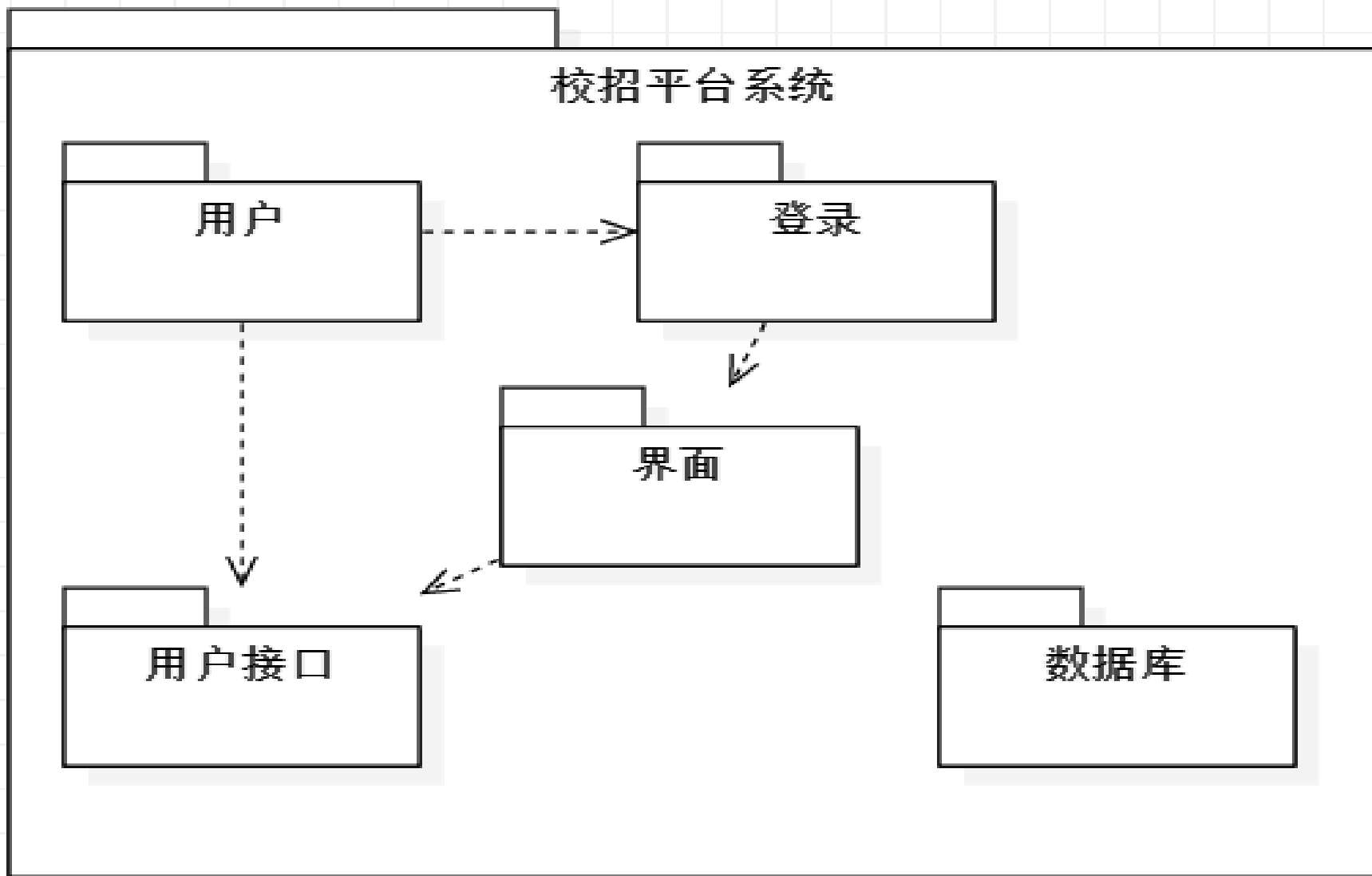
| 体系结构设计 |



系统分析

系统包图

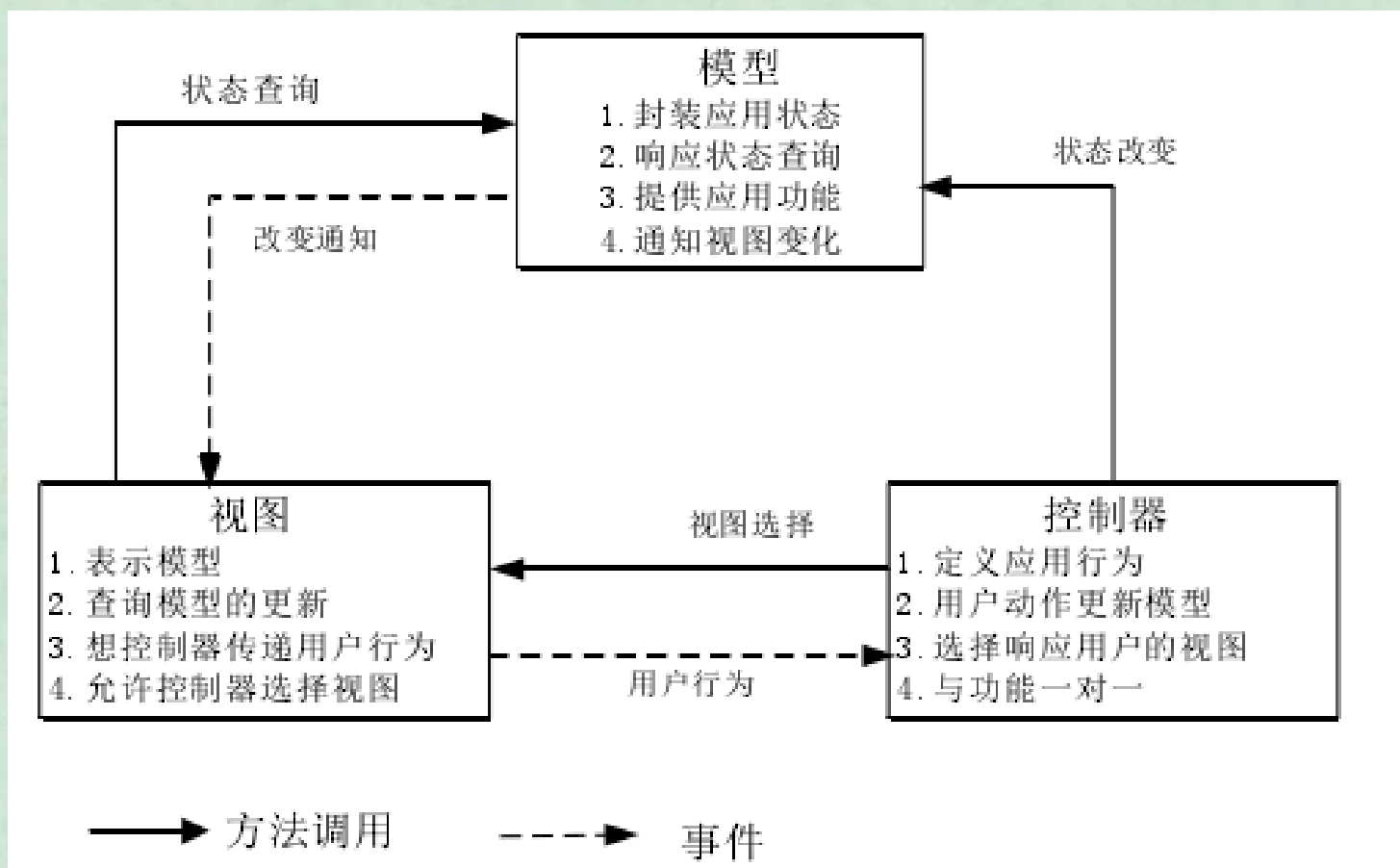
思想：
将一个大的复杂系统
划分成一个个小的子
系统，有助于理解和
组织。





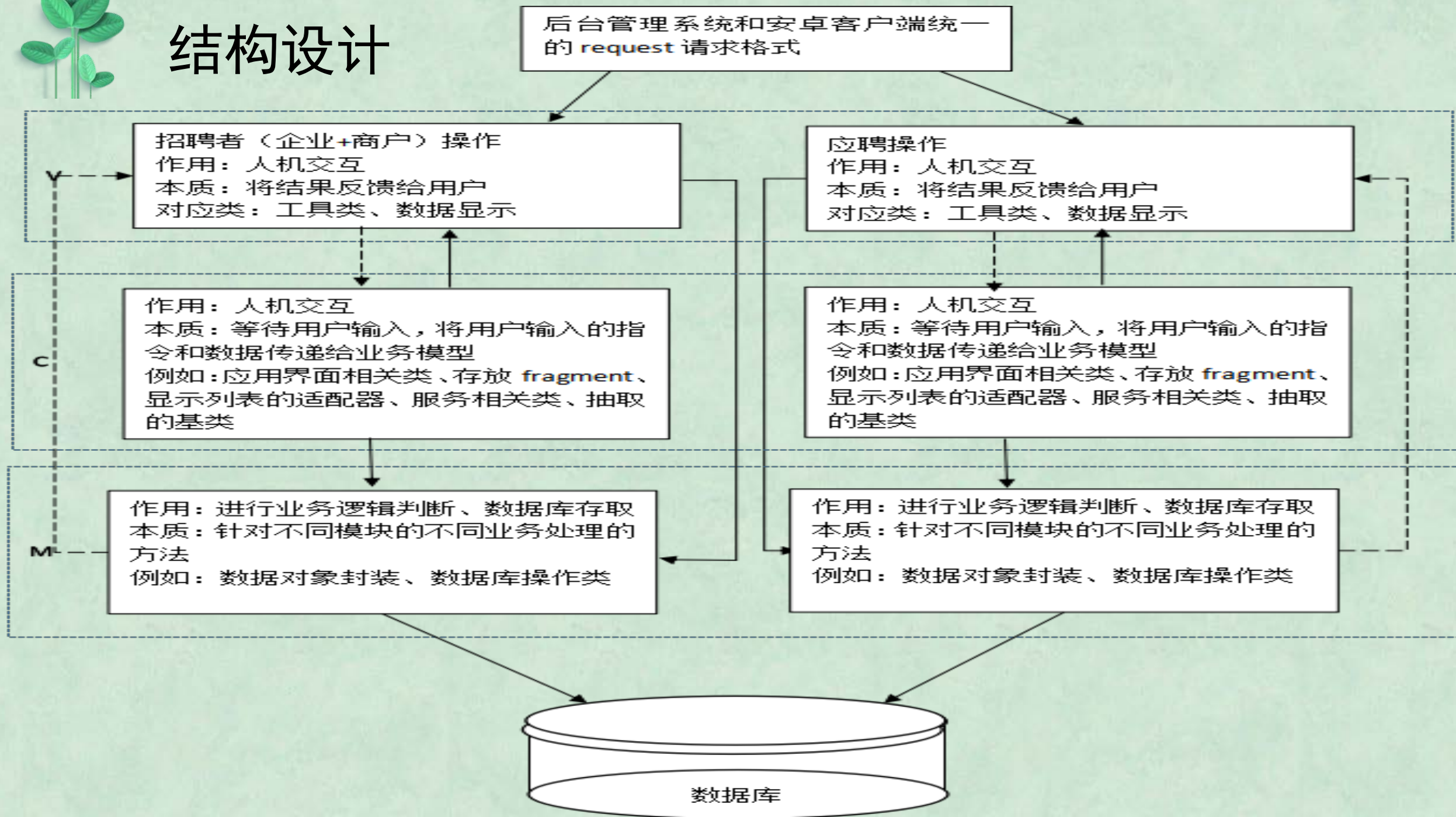
M V C 模式

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包括三个部分:模型、视图、控制器。





结构设计



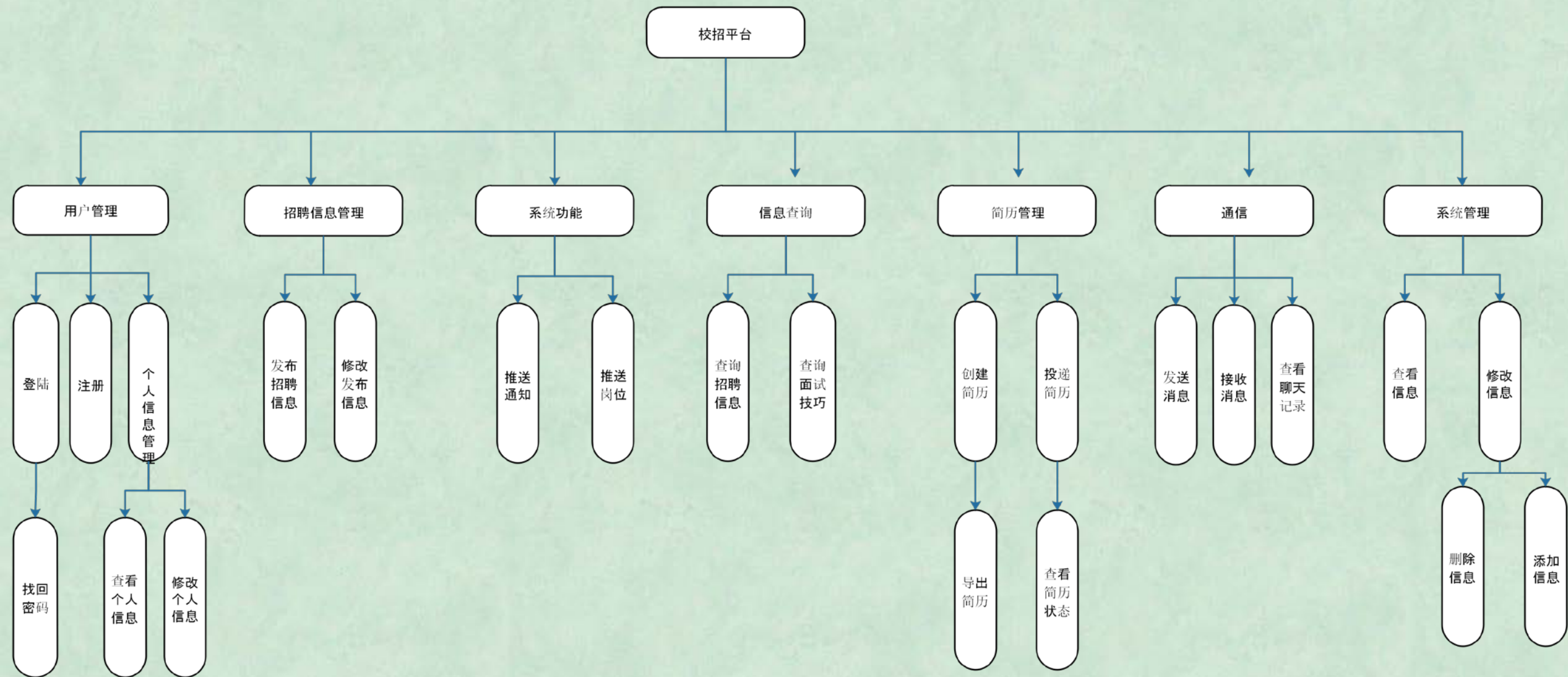


| 功能模块层次 |

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



功能模块层次



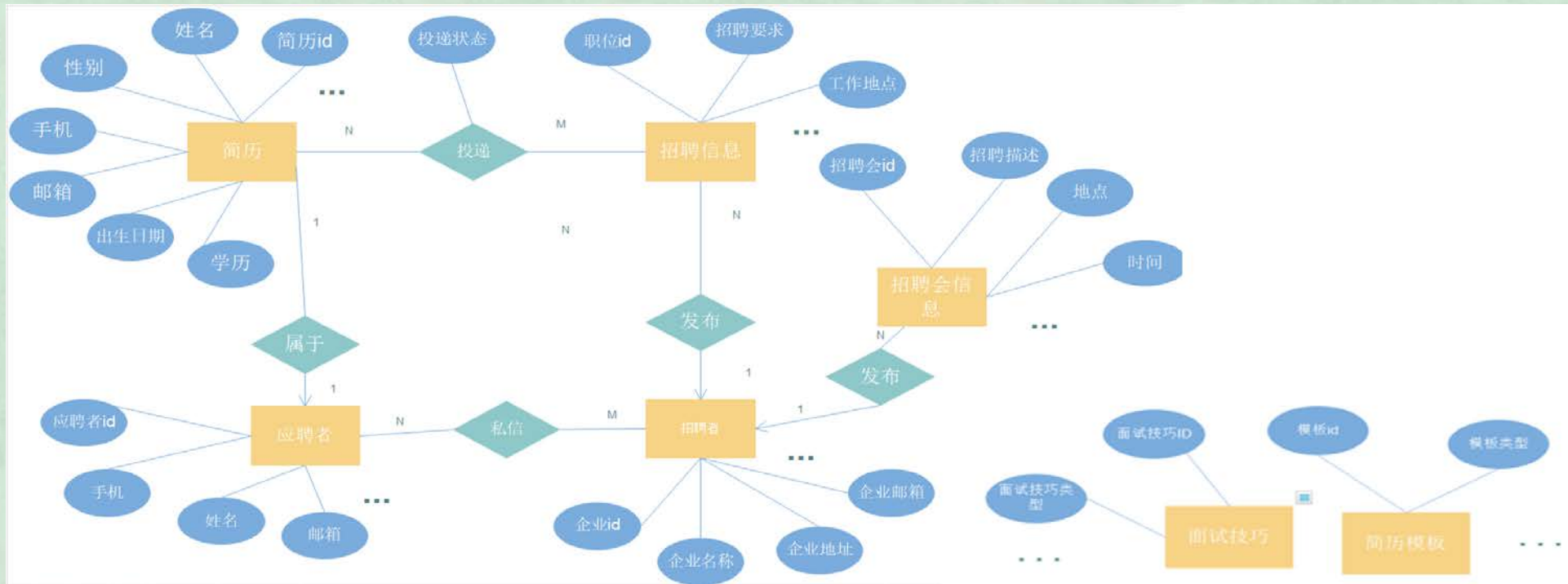


数据库设计

概念结构设计、逻辑结构设计、表结构设计、规范化



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关系联系在一起。



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、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。



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
- 2.数据库设计需满足第三范式
- 3.系统对不同用户要有明确的权限划分

