

一、概念设计

父实体集 User(userID, loginID, password, {role})

loginID 为用户登录时所用的账号，也是本人对应身份的工号/学号，是字符串

role 是表示用户角色的多值属性（是多值的，最终会被转化成一个关系模式）

该父实体集被以下子实体集继承：

Administrator(name)

Technician(name, title)

Instructor(name, title)

Student(name, major, class)

枚举集 Role

ROLE_ADMIN, ROLE_TECHNICIAN, ROLE_INSTRUCTOR, ROLE_STUDENT

实体集 Semester(semesterID, firstHalfYear, secondHalfYear, season, startDate, weekCount)

枚举集 Season

SPRING, FALL

实体集 Laboratory(labID, labNumber, name, labType, deviceCount)

labID 是一间实验室在该管理系统中的唯一标识符（无实际意义，是主键），labNumber 是现实中的房间编号（有实际意义，不能作为主键）

关系集 ResponsibleFor，实验员和实验室之间的关系，多对多，实验室全参与

枚举集 LabType

SOFTWARE, HARDWARE, NETWORK, SYSTEM, IOT

软件，硬件，网络，系统，物联网

以下所有实体集和关系集中，section 是复合属性，包括 weekday（星期几）和 slot（每天中的节次）两个部分。在转化而成的关系表格中，section 直接用这两个域代替。

枚举集 Weekday

MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY

枚举集 Slot

ONE_TO_TWO, THREE_TO_FIVE, SIX_TO_SEVEN, EIGHT_TO_NINE, TEN_TO_TWELVE, THIRTEEN_TO_FIFTEEN

实体集 LongArrangement(longArrangementID, startWeek, endWeek, section, studentClass, studentCount)

(学期、实验室、排课申请之间的关系，也可以是单独的实体集(避免三元关系)，也可以是与 instructor request 之间的关系(有一些重叠属性))

startWeek, endWeek 和 section 表示“实际”安排的时间，studentClass 和 studentCount 表示“实际”安排的班级和人数

与 InstructorRequest 之间的关系，多对一，arrangement 全参与

与 Laboratory 的关系，多对一，arrangement 全参与，这里的实验室表示“实际”安排的实验室

实体集 ShortArrangement(shortArrangementID, week, section)

(是学期、实验室、借用申请之间的关系，也可以是实体集)

与 StudentRequest 之间的关系，一对一，arrangement 全参与

与 Laboratory 之间的关系，多对一，arrangement 全参与

实体集 InstructorRequest(instructorRequestID, requestTime, labType, startWeek, endWeek, section, course, studentClass, studentCount, status, adminProcessTime, adminMessage)

注意要有申请状态

与学期的关系，多对一，request 全参与

与教师的关系，多对一，request 全参与

枚举集 InstructorRequestStatus

ARRANGED, NOT_ARRANGED

实体集 RepairRequest(repairRequestID, requestTime, requestDescription, status, repairStartTime, repairEndTime, technicianMessage)

与实验室的关系，多对一，request 全参与

与教师的关系，多对一，request 全参与

枚举集 RepairRequestStatus

REPAIRED, NOT_REPAIRED, REPAIRING

实体集 StudentRequest(studentRequestID, week, section, reason, requestTime, status, adminProcessTime, adminMessage, useCompleteTime)

与学生的关系，多对一，request 全参与

与实验室的关系，多对一，request 全参与

与学期的关系，多对一，request 全参与

枚举集 StudentRequestStatus

NOT_VIEWED, APPROVED, DENIED, USE_COMPLETE

二、函数设计

1. 平台角色(使用该平台的用户)说明

平台各类角色的信息，已经通过带继承的用户系统（User 实体集以及其子实体集所生成的那些关系模式）存储。可以通过多值的角色属性 role（最终其实是一张表），对用户权限进行 CRUD。

2. 平台中主要公共数据说明

学期和实验室的数据，已经通过各自的实体集体现，可以进行 CRUD。上课节次也可以在用到的地方通过枚举集的值体现。

3. 平台要求的功能

3.1 系统管理员的功能

(1) 学期管理

列出平台已经有的学期、添加新学期：通过学期实体集的表实现

设置平台的当前学期，设置后，平台其他的所有功能都是指在“当前学期”的工作：在服务端实现（但需要将这个数据持久化，可以以配置文件参数的方式持久化）

(2) 平台用户管理

实验员、教师、学生等人员的增加、删除、修改、重置密码：通过用户系统的表格的 CRUD 实现

每一类用户的管理需要列出全部该类用户，提供按姓名查询的功能：同上

用户的批量导入：同上，只不过技术上需要进行表格文件的传输和读取

(3) 实验排课

根据“教师”填报的“实验课申请登记”，将实验课安排到实验室：向 LongArrangement 的关系表插入表项

根据“实验课申请登记”的“实验室类型”确定可以选择的实验室范围：查询 Laboratory 的关系表

根据“实验课申请登记”的“学生人数”和实验室设备数决定分配哪几间实验室：查询 Laboratory 的关系表

根据“实验课申请登记”的“起始周”、“结束周”和已经排课的情况避免实验室冲突：查询 LongArrangement 和 ShortArrangement 的关系表，确定冲突情况

(4) 实验室借用审批

查看“学生”填报的“实验室借用申请”：查询 StudentRequest 的表

审批“实验室借用申请”，通过或不通过：向 ShortArrangement 插入表项，并修改

StudentRequest 中的 status 和 adminMessage，检查冲突的情况同上，仍然是查

ShortArrangement 表

3.2 教师的功能

(1) 实验课申请登记

列出当前学期的自己的所有申请：查询 InstructorRequest 的表

新增申请，填写新的申请，新申请的状态为“未排课”：向 InstructorRequest 的表插入表项

修改申请，对处于“未排课”状态的申请可以修改内容：修改 InstructorRequest 表项，检查状态用的是 status 属性

申请内容包括：上课学期(默认当前学期)、申请教师(默认登录教师)、课程名称、需要的实验室类型、学生班级、学生人数、起始周、结束周、节次：InstructorRequest 的关系表（由相关实体集和关系集生成的关系表）已经体现

(2) 实验室设备报修

列出教师的所有设备报修：查询 RepairRequest 的表

新增报修，填写新的报修，新报修的状态为“未维修”：向 RepairRequest 的表插入表项
报修的内容包括，报修教师、实验室编号、故障描述(文本)、报修日期：已经通过相关实体集和关系集体现

3.3 实验员的功能

设备报修处理

列出属于自己管理的实验室的报修：通过查询 ResponsibleFor 和 RepairRequest 的表格实现

开始维修时，把报修申请的状态设置为“维修中”：修改 RepairRequest 中的 status 和 repairStartTime 时间戳

维修结束后，把报修申请的状态设置为“已维修”，并填写维修情况说明：修改 RepairRequest 的 repairEndTime 时间戳和 technicianMessage

3.4 学生功能

实验室借用申请

列出自己所有填报过的实验室借用申请(申请状态有：未审核、通过、驳回、使用完毕)：查询 StudentRequest 的表

新增申请，填写新的申请，新申请的状态为“未审核”：向 StudentRequest 的表插入表项

修改申请，对处于“未审核”状态的申请可以修改内容：修改 StudentRequest 的表的表项
申请的内容包括，申请学期、申请学生、申请周次、申请节次、申请实验室编号(应根据学期、周次、节次检查是否可用)、申请原因、填报日期：已经体现

对于通过的申请，在使用完成后进行确认(将申请状态设置为“使用完毕”)：修改 StudentRequest 的表的 status，并更新相应时间戳

3.5 通用功能(全部角色和未登录均可使用)

实验室课程表查询，建议在首页显示，表格如下图所示：

2022-2023-2 实验室排课表								
	实验室	机房	1—2	3—5	6—7	8—9	10—12	13—15
星期一	软件实验室	732		课程: 汇编语言 教师: 张三 班级: 21级软件工程 周次: 5-12		课程: 汇编语言 教师: 张三 班级: 21级软件工程 周次: 5-12		
		734						
		736						
		738						
		735						
		532						
		536						
		537						
	535							
	计算机系统实验室	704- 708						
	计算机硬件实验室	703- 707						
	计算机系统实验室	801、803						
物联网实验室	806、808							
计算机网络实验室	832、834							
星期五	软件实验室	732						
		734						
		736						
		738						
		735						
		532						
		536						
		537						
	535							
	计算机系统实验室	704- 708						
	计算机硬件实验室	703- 707						
	物联网实验室	806、808						
计算机网络实验室	836、838							

表格的任意一大行的内容是，整个学期的所有周里，某个 weekday（星期几）的实验室排课情况（group by weekday）。接着 group by 实验室类别、具体房间号，将大行划分为很多小行，每一小行的前 3 列可以看作分别是 weekday、labType、labNumber。每一小行的后几个域的名字为各个 Slot 集的值，域的值表示这个 slot 有没有排课，排的是什么课。

实体集 LongArrangement (longArrangementID, startWeek, endWeek, section, studentClass, studentCount)

与 InstructorRequest 之间的关系，多对一，arrangement 全参与

与 Laboratory 的关系，多对一，arrangement 全参与，这里的实验室表示“实际”安排的实验室

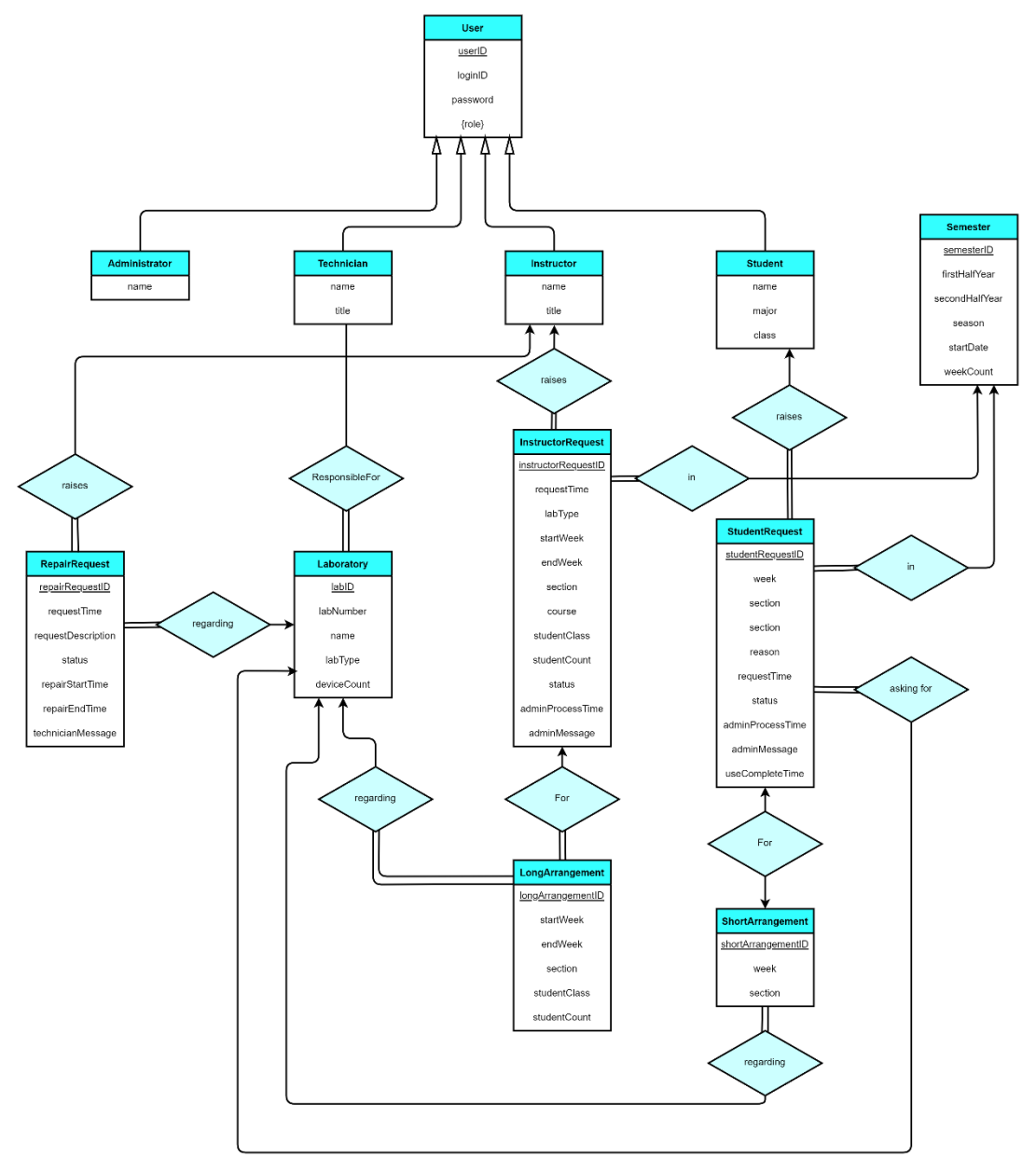
LongArrangement 实体集及其相关的关系集可以帮助找到所有信息：

通过 LongArrangement 中的 section 中的 weekday，可以筛查到一个 weekday 里的所有排课。通过其与 Laboratory 的关系，可以查到 labType 和其他 lab 的信息。课程名称和教师姓名可以通过其与 InstructorRequest 的关系查到。实际排课的班级和节次可以通过其属性查到（注意不要通过 request 去查，因为请求的班级和排课时间可能和实际的班级和排课时间不同，比如有可能是这样，张老师申请在周二 8-9 节为计算机 1-4 班 120 位同学上实验课，结果由于排课冲突、容量不足等种种原因，她的请求被处理成了 2 个排课 LongArrangements，一个是在某个时间段为 1-2 班的 60 位同学上课，另一个是在另一个时间段为 3-4 班的 60 位同学上课）

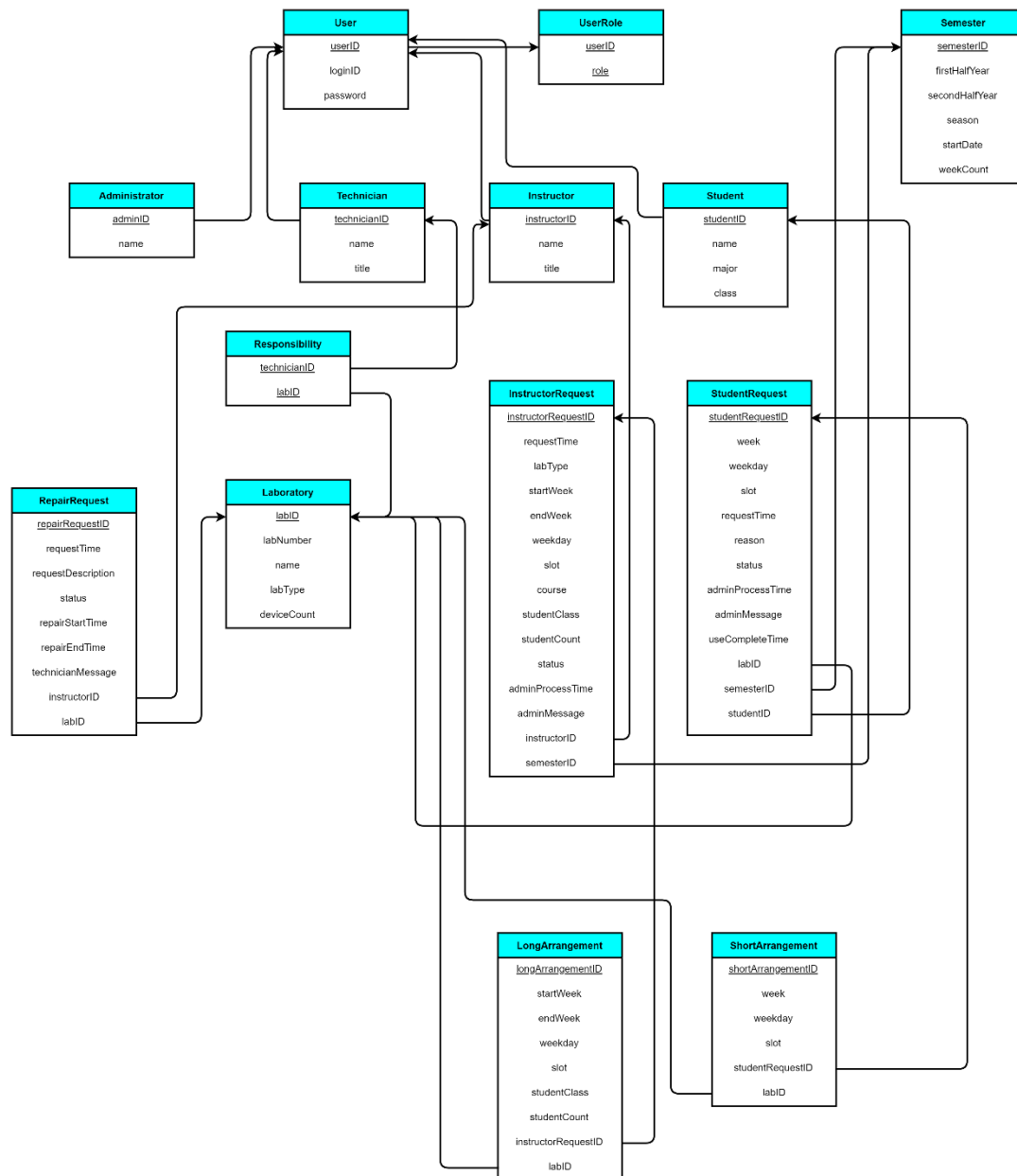
LongArrangement 提供了有排课的时间块，其他没有排课的实验室、时间块，需要通过枚

举类型查到。比如说某学期的周三，系统实验室没有被排课，但是界面上还是需要要有系统实验室这一类的小行，这就可以通过列举 LabType 的所有值，来确定有这一行的存在。

三、ER 图



四、关系模式



对关系模式的说明：

1、Administrator, Technician, Instructor, Student 表里的 ID，既是表的主键，也是指向 User 表的外键，也就是下面的 3 点需要一直成立：

- 1) 这个 ID 在各个表里是唯一且非空的，是主键
- 2) 对于每一个 ID，一定存在 1 个 userID，ID = userID
- 3) ID = userID 在 User 表里也是唯一且非空的，是主键

2、关于 UserRole 表：

假如一个人既是教师也是技术员（实验员），那么 UserRole 里存在 2 行，(x, ROLE_INSTRUCTOR) 和 (x, ROLE_TECHNICIAN)，其中 x 为这个人的 userID

3、创建一个用户至少要有以下操作：

- 1) 插入 User 表项
- 2) 插入相关子表项

3) 更新 UserRole 表项

删除一个用户也要相应 3 步

4、更新用户权限，就是对 UserRole 表项进行 CRUD