青岛理工大学

软件课程设计

题 目：宿舍信息管理系统(DMIS)

院　　别：信息与控制工程学院

专业班级：计算 183 班

学生姓名：陈杨瑞禧 田显阳 杨金树

指导老师：张楠

学　　号：201811050094 201811050095 201811050096

2020 年 7 月 23 日

前言

随着科学技术的进步与生产方式的快速发展，人类知识的总量也在不断的增长与加快，面对生产社会化逐渐扩大的趋势，以及社会对产品多样性的需求，使组织在运营活动中涉及的内外部信息迅速膨胀，人们对信息的采集、传输和处理过程中的时间性与准确性要求日益增高。面对庞大的信息处理要求，传统手工作业方式的信息处理系统在效率上已无法满足，并逐渐陷入信息逐渐庞大但效率日益降低的困境。以计算机为载体的信息处理系统成为摆脱这种困境的一个良好的出路。基于计算机对数据的快速、准确、可靠处理信息的能力，以及能存储庞大数据的特点，管理信息系统应势而生。

管理信息系统（Management Information System，简称 MIS）是一个以人为主导，利用计算机硬件、软件、网络通信设备以及其他办公设备，进行信息的收集、传输、加工、储存、更新、拓展和维护的系统。管理信息系统利用计算机的硬件和软件，手工规程、分析、计划、控制和决策用的模型，以及数据库对信息进行收集、传输、加工、保存和使用。因此，管理信息系统是一个标准的信息处理系统。

基于青岛理工大学本校学生在校生活状况，以及当前面对新冠疫情的特殊需求，现需设计这样一个为在校学生以及宿舍管理人员的校园宿舍生活所服务的管理信息系统——宿舍管理系统(Domitory Management Information System,DMIS)，本系统的最终用户为宿舍楼管理人员和在宿舍楼居住的学生，依据校内宿舍学生的日常生活需求，以及校方宿舍管理人员的管理需求所设计，实现解决宿舍人员管理问题，公寓设施使用以及报修管理问题，疫情期间外来访客登记管理问题，假期或特殊情况学生离返校管理等问题。

为满足上述问题定义中所提出系统需要解决的问题，我们了解到教务系统登记的学生信息可供提取，以使我们达到学生入住的信息登记录入，有了基础的学生信息，加上简单的数据库搭建以及系统处理，即可解决宿舍人员管理的问题，加上校内设施的联网使用，学生公寓的管理员门口登记等已知条件，若有该系统，即可解决公寓设施使用以及报修、疫情期间外来访客登记管理问题、假期或特殊情况学生离返校管理的问题，且大大方便了学生的公寓生活以及校职工的公寓管理难度，我们可知宿舍管理系统值得我们去制作与设计。

目录

第 1 章 系统需求分析.....4
1.1 宿舍楼的基本情况.....4
1.2 用户对系统的需求.....5
1.2.1 宿舍楼管理员.....5
1.2.2 本宿舍楼的学生.....5
1.2.3 系统功能概览.....5
1.3 性能需求.....6
1.4 可靠性和可用性需求.....6
1.5 出错处理需求.....6
1.6 接口需求.....7
1.7 约束.....7
1.8 逆向需求.....7
1.9 安全性与完整性要求.....7
1.10 完整性要求.....7
1.11 将来可能提出的要求.....8
1.12 数据字典.....8
1.12.1 数据结构.....8
1.12.2 数据流.....9
1.12.3 数据存储.....9
1.12.4 处理过程.....10
第 2 章 系统设计.....11
2.1 总体设计.....11
2.1.1 系统目标.....11
2.1.2 主要功能描述.....11
2.1.2 系统模块层次图.....12
2.1.3 数据流细化和复查.....12
2.1.4 关键的数据结构设计.....15
2.2 详细设计.....15

2.2.1 概念设计 15
2.2.2 数据项 16
2.2.3 运行设计 18
2.2.4 程序界面 19
第 3 章 数据库设计与实现 20
3.1 概念结构设计 20
3.1.1 总体 E-R 图 20
3.1.2 各子系统 E-R 图 20
3.2 逻辑结构设计 23
3.2.1 关系模型 23
3.2.2 用户子模式 23
3.3 物理结构设计 24
3.3.1 基本表设计 24
3.4 数据库的实施与维护 27
3.4.1 创建基本表代码 27
3.4.2 数据库关系图 32
第 4 章 系统实现 33
4.1 实现技术及环境 33
4.2 模块与功能 33
4.2.1 学生模块 33
4.2.2 管理员模块 34
第 5 章 结论与展望 38
5.1 结论 38
5.2 进一步工作的方向 38
5.3 展望 39

第 1 章 系统需求分析

1.1 宿舍楼的基本情况

学生住在宿舍楼中，每栋宿舍楼都会有若干名导员老师和宿舍楼管理人员负责本宿舍楼的日常管理。

1. 学生的基本信息：入校时，每名同学都有唯一的学号，并被分配到指定的宿舍楼、指定的宿舍和指定的床位，也会有一个宿舍号，其入校时间就是他的入住时间。另外，为了管理上的方便，同一院系的学生的宿舍会在一起，并相应地会有其所属的院系名称。

2. 宿舍的基本信息：每间宿舍都有唯一的宿舍号、所居住的学生及床位，每个床位有相应的编号。

3. 宿舍财产的基本信息：每个宿舍的财产属于学校，包括不限于电灯，床铺，柜子，桌椅等，为了对不同的财产进行区分，为每种财产分配不同的财产号。这样有利于财产的报修和管理。

4．公共设施信息：宿舍楼中除舍内财产以外，还有如洗衣机，洗鞋机等公共设施，每种财产分配相应的编号。这样有利于设施的报修。

5．报修的基本信息：宿舍楼中经常出现财产的损坏，比如灯泡坏损，宿舍洗衣机不工作等，这时，同学们需要将财产损坏情况在管理系统内提交报修记录并报告给宿舍楼管理员，以便学校派人进行维修。此时，需要记录报修的宿舍号和损坏的财产编号，同时记录报修的时间和损坏的原因。当损坏的财产维修完毕后，应记录解决时间及该报修是否成功解决。

6．楼内人员的信息 ：每个学生进宿舍楼需先刷自己的学生卡以通过入口闸机，学生卡内包含所属学生的学号，系统将记录此学生进入宿舍楼，学生出公寓亦然。有访客进入宿舍楼时，将在楼管处登记姓名，电话，到访时间，来办事项等信息，并在系统中记录。系统将宿舍楼在指定的时间关门（工作日 11:00 分，周末 11:30），若有同学晚于关门时间会宿舍，需通知宿舍楼管理员开门，同时记录晚归学生的学生信息，时间和晚归原因，以利于学校的管理和查证。

7．离校的基本信息：逢假期时，大部分学生不会留在宿舍。这时，为加强学校对同学假期安全的管理，离校的同学应登记离校时间，待返校后记录返校时

间， 以便学校查证和管理。

1.2 用户对系统的需求

1.2.1 宿舍楼管理员

1. 信息要求

宿舍楼管理员能查询上面提到的宿舍楼的所有相关信息，包括某一学号的学生在宿舍楼中住宿的详细信息，报修的所有信息，宿舍内人员的详细信息和学生离返校的信息。以利于对整个宿舍楼的全面管理。

2. 处理要求

当学生基本信息发生变化时，宿舍楼管理员能对其进行修改。比如，某些同学搬到其他的宿舍楼中去，他们在本宿舍楼中相应的记录就应该删去，搬到同宿舍楼其他的宿舍，他们在宿舍楼中相应的记录应该及时更新；学生转换专业，他们记录中院系的信息也要作相应的修改等等。 当宿舍楼的住宿信息发生变更时，宿舍楼管理员能根据有关证明做出修改。当宿舍财产报修及时解决后，管理员应登记处理时间，表明该报修问题是否已成功解决。

1.2.2 本宿舍楼的学生

1. 信息要求

本宿舍楼的学生能查询其所在的宿舍的所有信息，能查询本楼的指定宿舍管理员的电话号码以及舍友信息，方便于学生入住，能查询自己的宿舍内人员记录和离返校记录。

2. 处理要求

本宿舍楼的学生能在报修信息表中插入报修信息，表示本宿舍的财产或者公共设施发生了损毁需要学校派人维修。学生离校时，能在离返校记录表中插入离校时间；学生返校后，能在离返校记录表中插入返校时间，表示已经回校。

1.2.3 系统功能概览

1. 学生用户管理部分

① 处理学生用户登录；

② 学生用户可以查询学生信息；

③ 学生用户可以提交与查询报修信息；

④ 学生用户可以查询与修改用户自己的离返校信息；

⑤ 学生用户可以修改用户自己密码。

2. 管理员用户管理部分

① 处理管理员登录；

② 管理员可以查询学生信息；

③ 管理员可以查询与修改报修信息；

④ 管理员可以查询与修改楼内人员信息；

⑤ 管理员可以查询离返校学生信息；

⑥ 管理员可以查询与修改晚归学生信息。

1.3 性能需求

1. 系统响应时间：对远程查询响应的平均时间是 3 秒，最长是 10 秒。

2. 吞吐量：每秒钟处理事务数达到 500 条。

3. 容量：系统可以容纳的同时操作人员是 500 人同时使用。

1.4 可靠性和可用性需求

1. 本系统在一天 24 小时内是随时可以使用的。

2. 本系统一周内不能出现两次以上的故障。

3. 平均故障间隔时间是每月 1 次。

4. 平均故障修复时间是 3 小时。

5. 双服务器备份保证数据完整。

1.5 出错处理需求

1. 在用户注册时，用户名不应为乱码与特殊字符，用户信息应提前录入学院，专业，班级等信息，防止乱入。

2. 用户除了用户名，学生信息/管理员信息等账号绑定信息外，应自动生成一个用户 ID 方便进行用户统计与删改。

1.6 接口需求

1. 用户界面：本系统的用户界面为网络程序。

2. 硬件接口：无。

3. 软件接口：访问的数据是 SQL sever,软件输出的数据格式是 xlsx。

1.7 约束

1. 设计语言：JAVA。

2. 软件使用硬件平台：运行 Windows 的计算机。

1.8 逆向需求

1. 要保证不出现数据的重复，例如学生的学号、同一公寓楼内的宿舍号、宿舍内物品编号、公寓楼内公共物品编号。

2. 用户与用户之间的无关联性，如用户名不可重复。

3. 还有就是用户与设施信息的更新，设备的报修与已修都要第一时间更新已避免出现用户与用户需求的冲突，离校返校的状态需要第一时间更新有利于校园管理，入住情况需要有明确的统计。

1.9 安全性与完整性要求

1. 系统应设置访问用户的标识以鉴别是否是合法用户，并要求合法用户设置其密码，保证用户身份不被盗用。

2. 系统应对不同的数据设置不同的访问级别，限制访问用户可查询和处理数据的类别和内容。

3. 系统应对不同用户设置不同的权限，区分不同的用户，如区分普通用户（学生），宿舍管理员。

1.10 完整性要求

1. 各种信息记录的完整性，部分信息记录内容不能为空。

2. 各种数据间相互的联系的正确性。

3. 相同的数据在不同记录中的一致性。

1.11 将来可能提出的要求

校园生活日新月异，新的功能与需求，淘汰的规则与设备也应随之增删，软件也应拥有更新的能力，将来可能提出的要求有：

1. 新的宿舍内设备录入与旧设备删除。
2. 新的公寓功能的添加，如公寓打印店与公寓咖啡厅。
3. 快递的与信件可以自由出入校内后的公寓邮件收发。

1.12 数据字典

1.12.1 数据结构

表 1.1 数据结构

数据结构名	组成
住宿学生信息	学号，姓名，性别，专业，楼号，宿舍号，电话，入住时间，楼内状态
宿舍信息	楼号，宿舍号，宿舍长学号
公共设施信息	楼号，设施号，设施名
报修信息	订单号，楼号，宿舍号，维修物品（设施）号，提交日期，解决日期，报修状态，报修原因
离返校信息	学号，离校时间，返校时间
访客人员信息	姓名，到访时间，电话，到访原因
用户信息	用户 ID，用户密码，用户类型
楼内人员信息	住宿学生信息，访客人员信息
宿舍楼信息	楼号，类型，管理员工号

1.12.2 数据流

表 1.2 数据流

数据流名	数据流来源	数据流去向	组成
登记报修信息	学生	报修登记表	报修信息
查询报修信息	报修登记表	管理员	报修信息
登记已修信息	管理员	报修登记表	报修信息
登记用户信息	学生	用户信息表	用户信息
查询用户信息	用户信息表	学生，管理员	用户信息
登记住宿学生信息	学生	学生信息表	住宿学生信息
查询住宿学生信息	学生信息表	学生，管理员	住宿学生信息
登记楼内状态信息	管理员	学生信息表	住宿学生信息
登记访客人员信息	管理员	访客人员登记表	访客人员信息
登记离返校信息	学生	离返校登记表	离返校信息
查询离返校信息	离校外出信息表	学生，管理员	离返校信息
更新楼内人员信息	学生信息表,访客人员登记表	楼内人员表	楼内人员信息
登记宿舍楼信息	管理员	宿舍楼信息表	宿舍楼信息
查询宿舍楼信息	宿舍楼信息表	管理员，学生	宿舍楼信息

1.12.3 数据存储

表 1.3 数据存储

数据存储名	输入的数据流	输出的数据流	组成
报修登记表	报修信息 已修信息	报修信息	报修信息
用户信息表	用户信息	用户信息	用户信息
学生信息表	住宿学生信息	住宿学生信息	住宿学生信息
访客人员登记表	访客人员信息	访客人员信息	访客人员信息
离返校登记表	离返校信息	离返校信息	离返校信息
楼内人员表	住宿学生信息 访客人员信息	楼内人员信息	楼内人员信息
宿舍楼信息表	宿舍楼信息	宿舍楼信息	宿舍楼信息

1.12.4 处理过程

表 1.4 处理过程

处理过程名	输入数据流	输出数据流
登记报修	报修信息	报修信息
查询报修	报修信息	报修信息
登记已修	报修信息	报修信息
登记用户信息	用户信息	用户信息
查询用户信息	用户信息	用户信息
登记住宿学生信息	住宿学生信息	住宿学生信息
查询住宿学生信息	住宿学生信息	住宿学生信息
登记访客人员信息	访客人员信息	访客人员信息
查询访客人员信息	访客人员信息	访客人员信息
登记离返校信息	离返校信息	离返校信息
查询离返校信息	离返校信息	离返校信息
查询楼内人员信息	楼内人员信息	楼内人员信息
更新楼内人员信息	住宿学生信息 访客人员信息	楼内人员信息
更新宿舍楼信息	宿舍楼信息	宿舍楼信息

第 2 章 系统设计

2.1 总体设计

2.1.1 系统目标

设计一个在 web 端运行的，以宿舍楼管理员和学生为用户的宿舍管理系统（DMIS）。系统能具备第 1 章系统需求分析所包含的功能，且满足性能、可靠性和可用性等所列出的各类需求。

2.1.2 主要功能描述

1. 学生用户管理部分

① 处理学生用户登录；

② 学生用户可以查询与修改自己的学生信息；

③ 学生用户可以提交与查询报修信息；

④ 学生用户可以查询与修改用户自己的离返校信息；

⑤ 学生用户可以修改用户自己密码。

2. 管理员用户管理部分

① 处理管理员登录；

② 管理员可以查询学生信息；

③ 管理员可以查询与修改报修信息；

④ 管理员可以查询与修改楼内人员信息；

⑤ 管理员可以查询离返校学生信息；

2.1.2 系统模块层次图

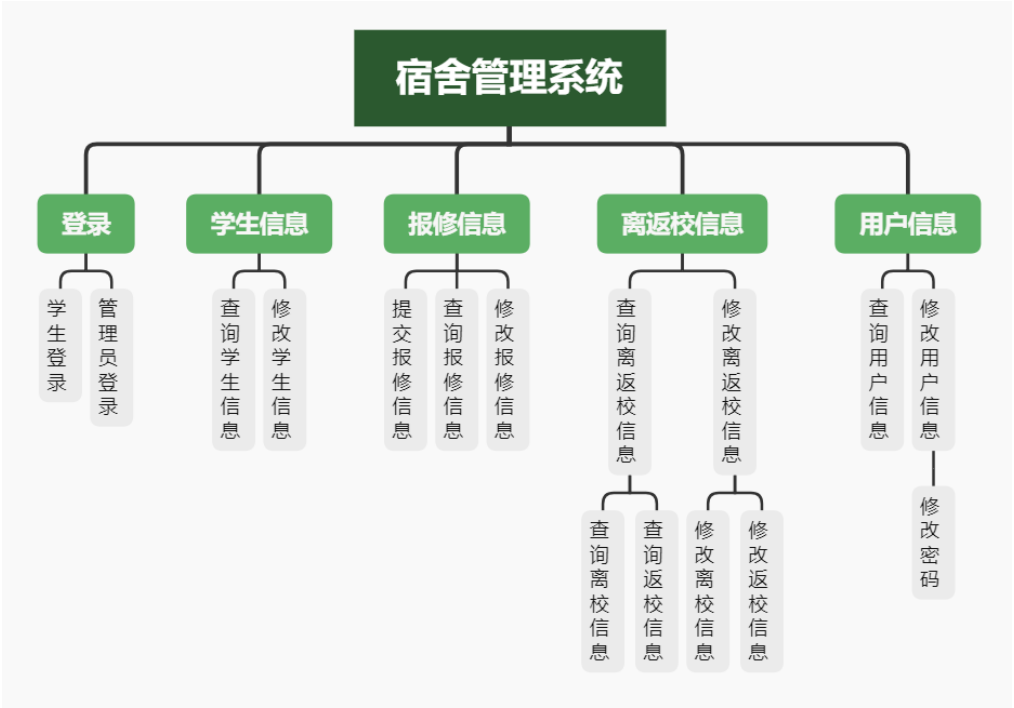


图 2.1 系统模块层次图

2.1.3 数据流细化和复查

1. 报修数据流图

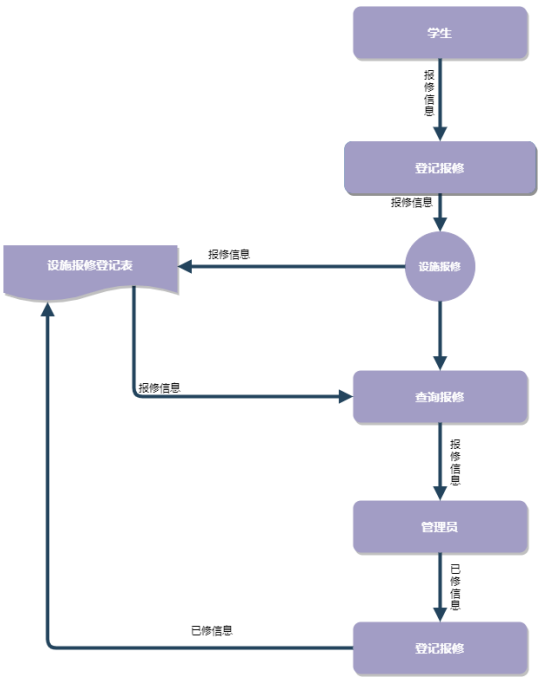


图 2.2 报修数据流图

2. 公共设施使用部分数据流图

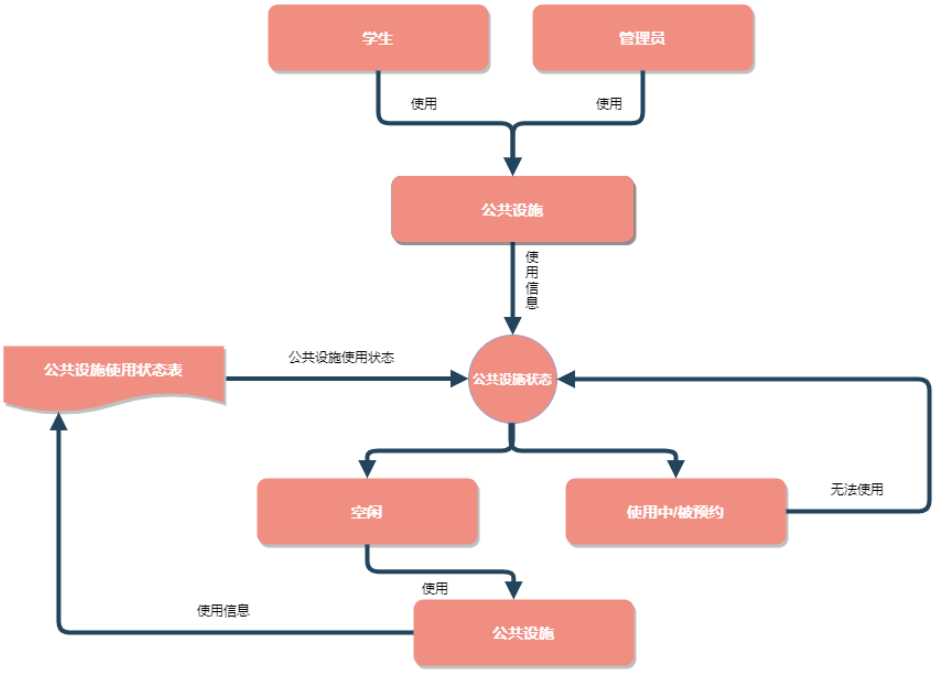


图 2.3 公共设施使用部分数据流图

3. 离返校部分数据流图

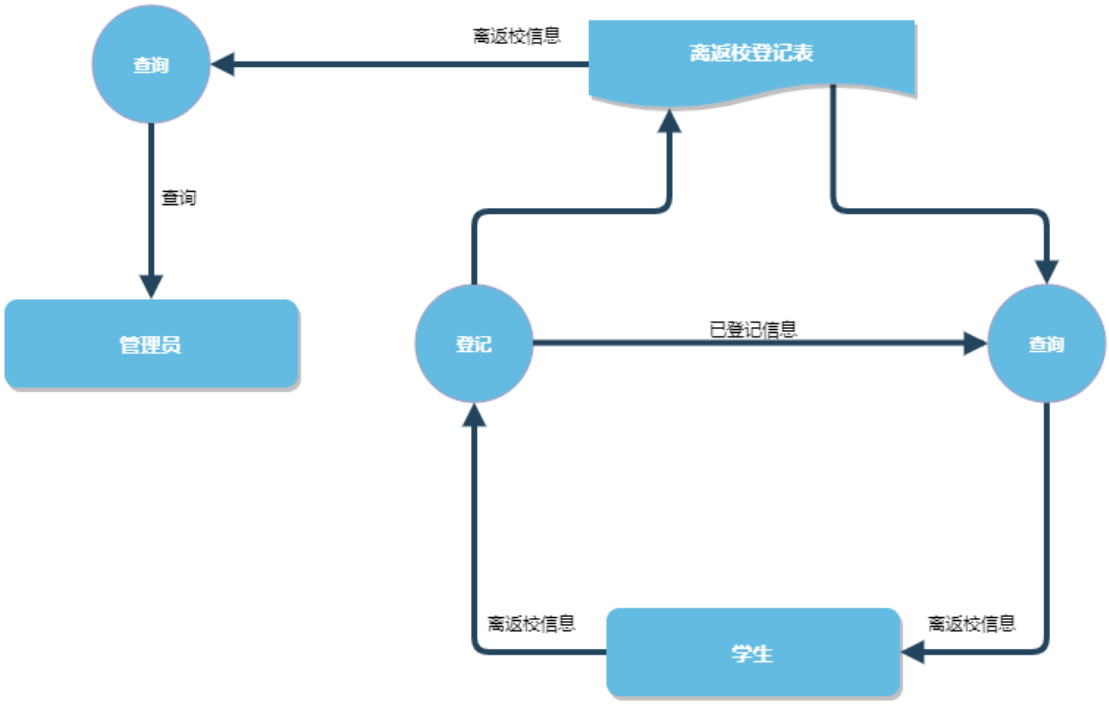


图 2.4 离返校部分数据流图

5. 楼内人员部分数据流图

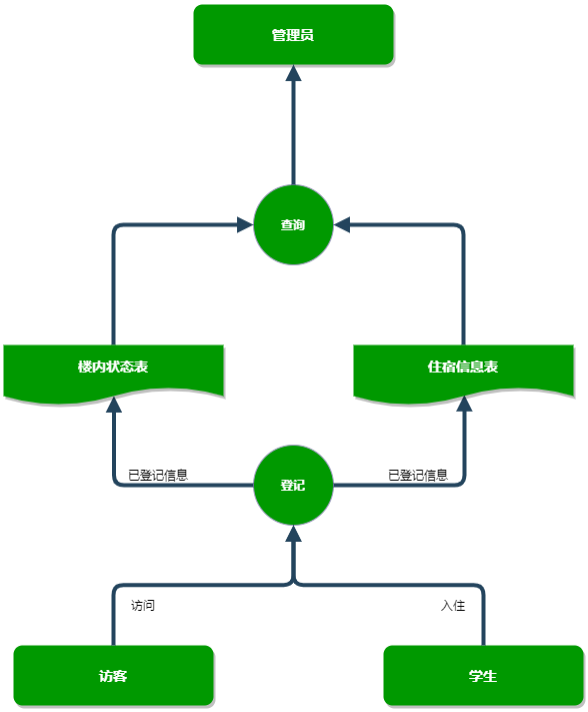


图 2.5 楼内人员部分数据流图

6. 总流程图部分数据流图

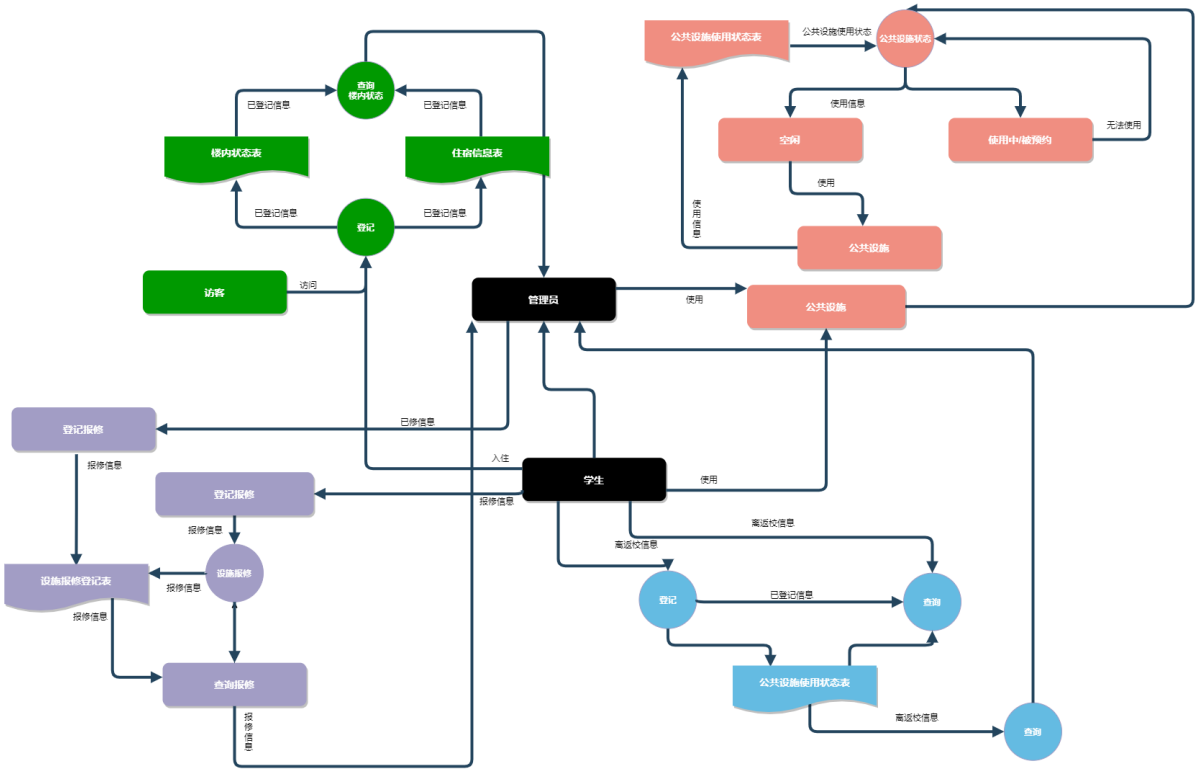


图 2.6 总流程图部分数据流图

2.1.4 关键的数据结构设计

表 2.1 数据结构

数据结构名	组成
住宿学生信息	学号，姓名，性别，专业，楼号，宿舍号，电话，入住时间，楼内状态
宿舍信息	楼号，宿舍号，宿舍长学号
公共设施信息	楼号，设施号，设施名
报修信息	订单号，楼号，宿舍号，维修物品（设施）号，提交日期，解决日期，报修状态，报修原因
离返校信息	学号，离校时间，返校时间
访客人员信息	姓名，到访时间，电话，到访原因
用户信息	用户 ID，用户密码，用户类型
楼内人员信息	住宿学生信息，访客人员信息
宿舍楼信息	楼号，类型，管理员工号

2.2 详细设计

2.2.1 概念设计

1. 学生用户管理部分

① 处理学生用户登录；

② 学生用户可以查询学生信息；

③ 学生用户可以提交与查询报修信息；

④ 学生用户可以查询与修改用户自己的离返校信息；

⑤ 学生用户可以修改用户自己密码。

2. 管理员用户管理部分

① 处理管理员登录；

② 管理员可以查询学生信息；

③ 管理员可以查询与修改报修信息；

④ 管理员可以查询与修改楼内人员信息；

⑤ 管理员可以查询离返校学生信息；

⑥ 管理员可以查询与修改晚归学生信息。

2.2.2 数据项

1. 住宿学生信息数据项

表 2.2 住宿学生信息数据项

属性名	存储代码	类型	长度	备注
学号	Sno	char	12	学生学号
姓名	Snam	char	20	学生姓名
性别	Ssex	char	4	学生性别
专业	Sdep	char	40	学生专业
楼号	Bno	char	6	学生所在的宿舍楼楼号
宿舍号	Dno	char	6	学生所在宿舍的宿舍号
电话	Stel	char	11	学生电话
入住时间	Sche	date	8	学生搬入宿舍的时间
楼内状态	Scon	tinyint	1	是否在宿舍楼内

2. 宿舍信息数据项

表 2.3 宿舍信息数据项

属性名	存储代码	类型	长度	备注
楼号	Bno	char	6	所在的宿舍楼楼号
宿舍号	Dno	char	6	宿舍的宿舍号
宿舍长学号	Sno	char	12	宿舍长的学号

3. 公共设施信息数据项

表 2.4 公共设施信息数据项

属性名	存储代码	类型	长度	备注
楼号	Bno	char	6	所在的宿舍楼楼号
设施号	Fno	char	4	楼内所属设施的设施号
设施名	Fnam	char	20	设施的名称
设施状态	Fcon	tinyint	1	是否在使用中

4. 报修信息数据项

表 2.5 报修信息数据项

属性名	存储代码	类型	长度	备注
订单号	Ono	char	10	订单号
楼号	Bno	char	6	所在的宿舍楼楼号

宿舍号	Dno	char	6	报修物品所在宿舍
维修设施号	Rno	char	2	报修物品的物品号
提交日期	Rsub	date		报修提交的日期
解决日期	Rsol	date		报修解决的日期
报修原因	Rrea	char	50	报修的原因陈述

5. 离返校信息数据项

表 2.6 离返校信息数据项

属性名	存储代码	类型	长度	备注
学号	Sno	char	12	离校学生的学号
离校时间	Llea	date	8	离校的时间
返校时间	Lret	date	8	返校的时间

6. 访客人员信息数据项

表 2.7 访客人员信息数据项

属性名	存储代码	类型	长度	备注
姓名	Vnam	char	20	访客姓名
楼号	Bno	char	6	所在的宿舍楼楼号
到访时间	Vtim	timestamp	14	访客到访时间
电话	Vtel	char	11	访客的联系电话
到访原因	Vrea	char	50	访客的到访原因

7. 管理人员信息数据项

表 2.8 管理人员信息数据项

属性名	存储代码	类型	长度	备注
工号	Ano	char	12	管理员的工号
姓名	Anam	char	20	管理人员姓名
电话	Atel	char	11	管理人员的联系电话
楼号	Bno	char	6	管理人员所属楼号

8. 用户信息数据项

表 2.9 用户信息数据项

属性名	存储代码	类型	长度	备注
用户 ID	Uid	char	20	
用户密码	Upw	char	20	
用户类型	Utyp	boolean	1	是否为管理员

9. 宿舍楼信息数据项

表 2.10 宿舍楼信息项

属性名	存储代码	类型	长度	备注
楼号	Bno	char	6	宿舍楼的楼号
类型	Btyp	tinyint	1	判断男女宿舍
管理员工号	Dno	char	12	确定管理员

2.2.3 运行设计

1. 运行模块的组合

由上面的模块结构图可以看出，在本系统中有 “学生登录”和“管理员登录”两个模块，点击两个模块处对再进行使用会引起不同的模块进行反映：

登录模块顶部设置两个连接 “学生登录” 和“管理员登录”,用户单击 “学生登录”之后就进入学生系统登录模块；单击“管理员登录”之后将进入管理员系统登录模块。用户输入密码和口令执行登录操作之后，登录模块以用户名和密码作为参数调用验证用户登录信息。

7. 运行控制方式采用超级连接触发的方式。

2.2.4 程序界面



图 3.8 程序主界面图



图 3.9 学生系统界面图



图 3.10 管理员系统界面图

第 3 章 数据库设计与实现

3.1 概念结构设计

3.1.1 总体 E-R 图

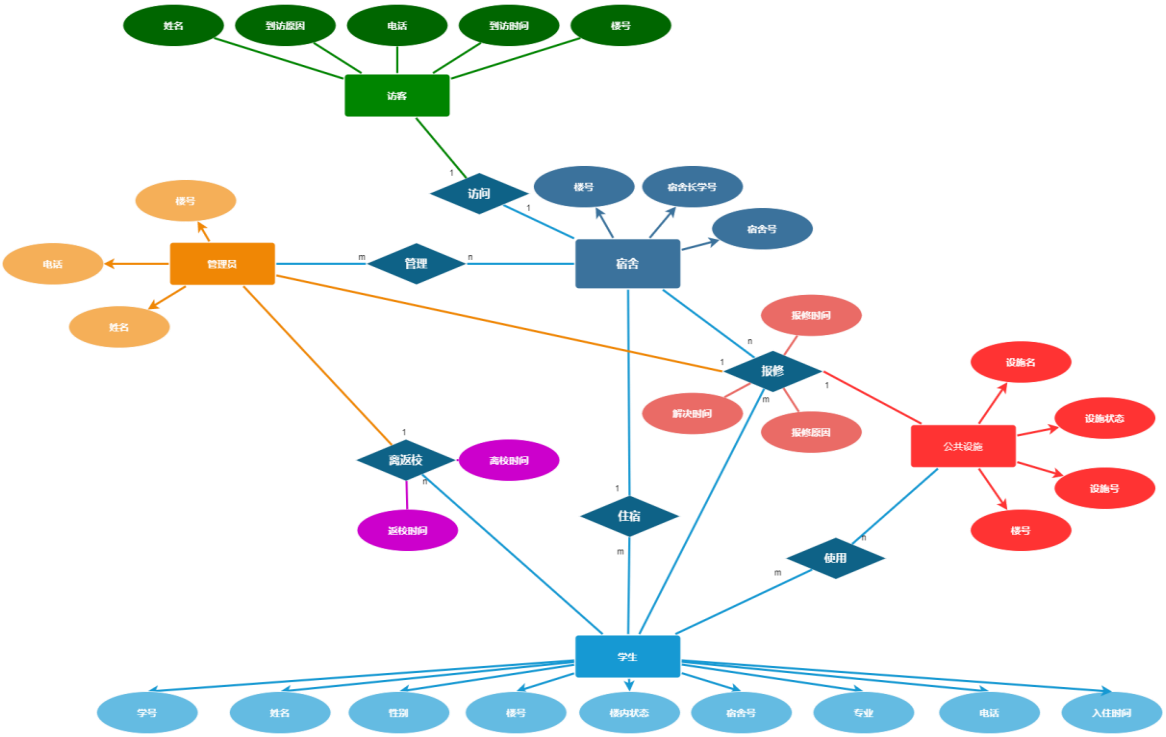


图 3.1 总系统 E-R 图

3.1.2 各子系统 E-R 图

1. 学生入住系统

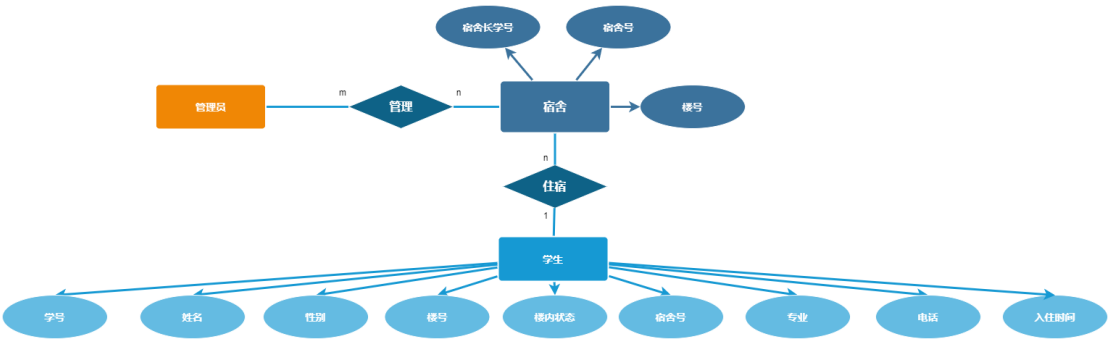


图 3.2 学生入住系统 E-R 图

2. 访客系统

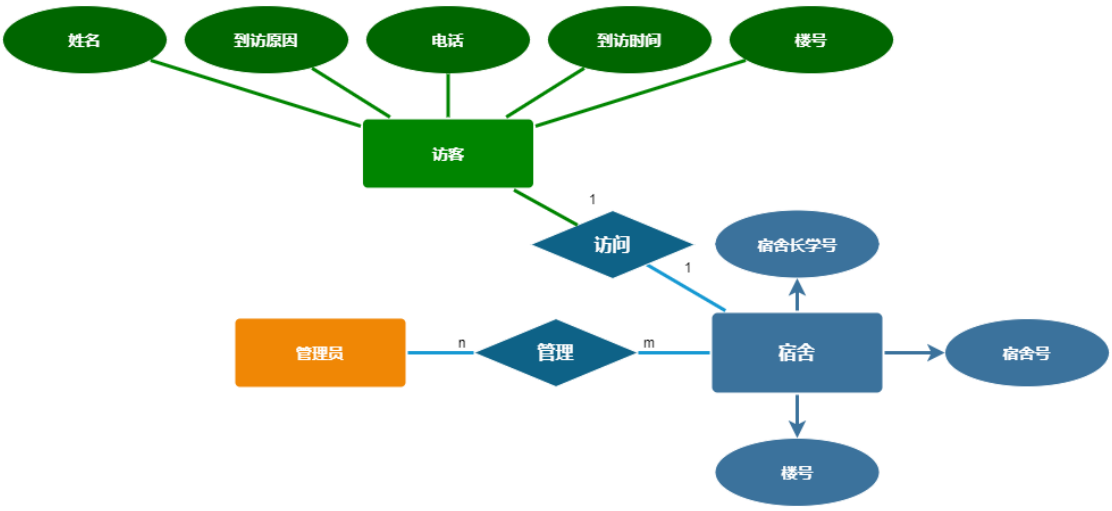


图 3.3 访客系统 E-R 图

3. 管理人员系统

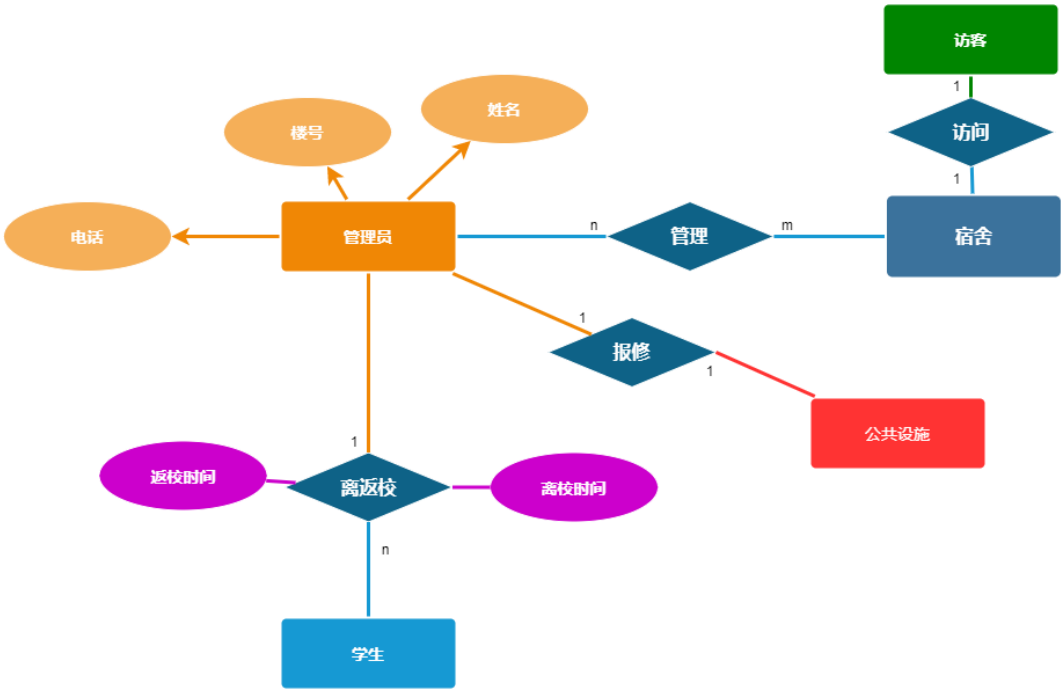


图 3.4 管理人员系统 E-R 图

4. 公共设施系统

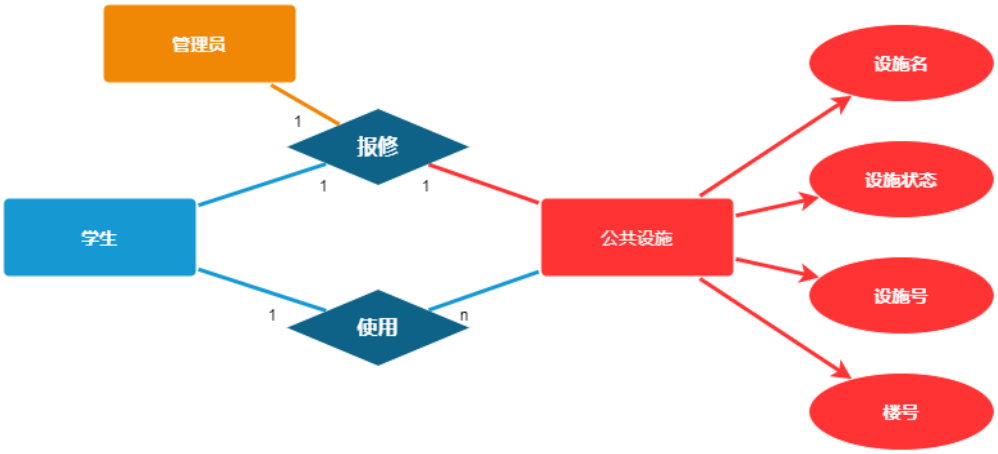


图 3.5 公共设施系统 E-R 图

6. 报修系统

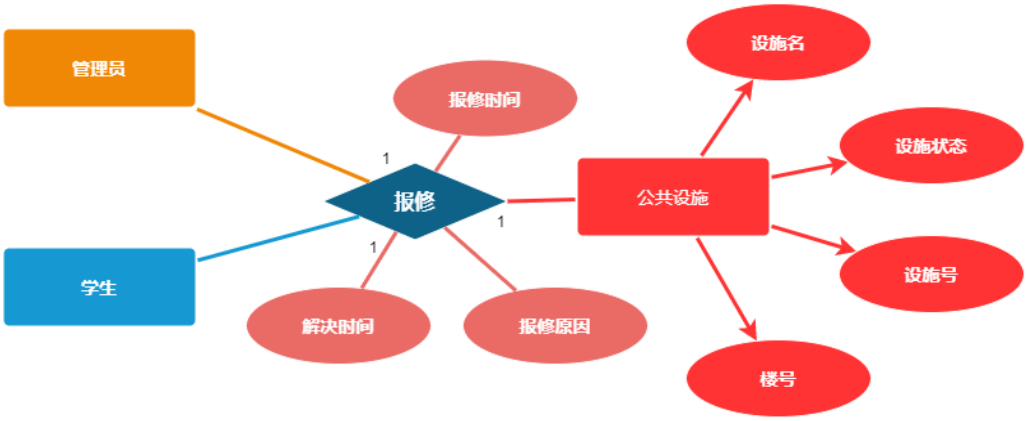


图 3.7 报修系统 E-R 图

7. 离返校系统

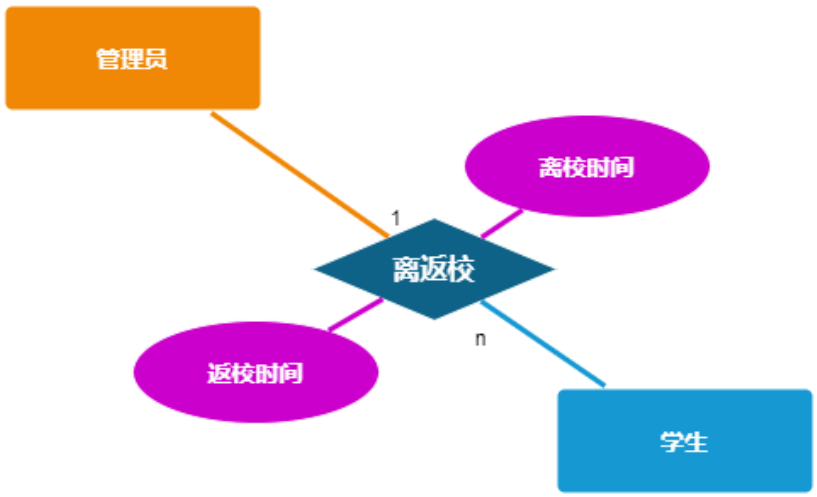


图 3.8 离返校系统 E-R 图

3.2 逻辑结构设计

3.2.1 关系模型

由 3.1，将所属 E-R 图转换为关系模型：

用户（用户 ID，用户密码，用户类型）

学生（学号，姓名，性别，专业，宿舍号，楼号，电话，入住时间，楼内状态）

宿舍（宿舍号，宿舍长学号，楼号）

住宿（学号，宿舍号）

管理人员（工号，电话，姓名，楼号）

管理（楼号，宿舍号）

宿舍财产（物品号，物品名，楼号，宿舍号）

拥有（物品号，宿舍号）

公共设施（设施号，设施名）

使用（学号，设施号）

物品（设施）报修（宿舍号，物品号，楼号，提交日期，解决日期，报修原因）

报修（楼号，设施号，物品号，电话）

晚归（学号，晚归时间，晚归原因）

离校（学号，离校时间，返回时间）

访客（电话，姓名，到访时间，到访原因）

访问（电话，宿舍号）

3.2.2 用户子模式

1.用户视图

学生信息视图（学号，姓名，性别，专业，宿舍号，楼号，电话，入住时间，楼内状态）

管理人员信息视图（工号，姓名，楼号，电话，住宿学生信息，宿舍财产报修，公共设施报修，学生晚归记录，学生离校记录，访客来访记录）

报修视图（订单号，楼号，宿舍号，维修物品（设施）号，报修时间，解决时间，报修原因）

学生晚归视图（楼号，学号，姓名，性别，专业，宿舍号，时间，晚归原因）

学生离校视图（学号，姓名，性别，专业，宿舍号，离校时间，返校时间）

访客来访视图（楼号，姓名，电话，到访原因，到访时间）

3.3 物理结构设计

3.3.1 基本表设计

1. 住宿学生信息（students）基本表

表 3.1 students 基本表

属性名	存储代码	类型	长度	空值	备注
学号	Sno	char	12	×	主码
姓名	Snam	char	20	×	
性别	Ssex	char	4	×	
专业	Sdep	char	40	×	
楼号	Bno	char	6	×	外码
宿舍号	Dno	char	6	×	外码
电话	Stel	char	11	×	主码
入住时间	Sche	date		×	
楼内状态	Scon	tinyint	1	×	

2. 宿舍信息（domitor）基本表

表 3.2 domitory 基本表

属性名	存储代码	类型	长度	空值	备注
楼号	Bno	char	6	×	外码
宿舍号	Dno	char	6	×	主码
宿舍长学号	Sno	char	12	×	

3. 宿舍财产信息（d_property）基本表

表 3.3 d_property 基本表

属性名	存储代码	类型	长度	空值	备注
宿舍号	Dno	char	6	×	外码
物品号	Pno	char	7	×	主码
物品名	Pnam	char	20	×	
楼号	Bno	char	6	×	外码

4. 公共设施信息（facility）基本表

表 3.4 facility 基本表

属性名	存储代码	类型	长度	空值	备注
楼号	Bno	char	6	×	外码
设施号	Fno	char	4	×	主码
设施名	Fnam	char	20	×	

5. 报修信息（repair）基本表

表 3.5 repair 基本表

属性名	存储代码	类型	长度	空值	备注
订单号	Ono	char	10	×	主码
楼号	Bno	char	6	×	外码
宿舍号	Dno	char	6	×	
维修物品（设施）号	Rno	char	7	×	
提交日期	Rsub	date		×	
解决日期	Rsol	date		×	
报修原因	Rrea	char	50	×	

6. 离返校信息（l&a-school）基本表

表 3.6 l&a-school 基本表

属性名	存储代码	类型	长度	空值	备注
学号	Sno	char	12	×	主/外码
离校时间	Llea	date		×	
返校时间	Lret	date		×	

7. 晚归信息（arrive-late）基本表

表 3.7 arrive-late 基本表

属性名	存储代码	类型	长度	空值	备注
学号	Sno	char	12	×	主/外码
晚归时间	Btim	timestamp	6	×	
晚归原因	Brea	char	20	×	

8. 访客人员信息（visitors）基本表

表 3.8 visitors 基本表

属性名	存储代码	类型	长度	空值	备注

姓名	Vnam	char	20	×	主码
楼号	Bno	char	6	×	外码
到访时间	Vtim	timestamp	6	×	
电话	Vtel	char	11	×	主码
到访原因	Vrea	char	50	×	

9. 管理人员信息（roots）基本表

表 3.9 roots 基本表

属性名	存储代码	类型	长度	空值	备注
工号	Ano	char	12	×	主码
姓名	Anam	char	20	×	
电话	Atel	char	11	×	主码
楼号	Bno	char	6	√	外码

10. 用户信息（users）基本表

表 3.10 users 基本表

属性名	存储代码	类型	长度	空值	备注
用户 ID	Uid	char	12	×	主码
用户密码	Upw	char	20	×	
用户类型	Utyp	tinyint	1	×	

11. 宿舍楼信息（buildings）基本表

表 3.11 buildings 基本表

属性名	存储代码	类型	长度	空值	备注
楼号	Bno	char	6	×	主码
类型	Bty	tinyint	1	×	
管理员工号	Ano	char	12	√	外码

3.4 数据库的实施与维护

3.4.1 创建基本表代码

1.住宿学生信息（students）

```
CREATE TABLE `students` (
  `Sno` char(12) NOT NULL COMMENT '学号',
  `Snam` char(20) NOT NULL COMMENT '姓名',
  `Ssex` char(4) NOT NULL COMMENT '性别',
  `Sdep` char(40) NOT NULL COMMENT '专业',
  `Bno` char(6) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '楼号',
  `Dno` char(6) NOT NULL COMMENT '宿舍号',
  `Stel` char(11) NOT NULL COMMENT '电话',
  `Sche` date NOT NULL COMMENT '入住时间',
  `Scon` tinyint(1) NOT NULL COMMENT '楼内状态',
  PRIMARY KEY (`Stel`,`Sno`),
  KEY `Dno` (`Dno`),
  KEY `Sno` (`Sno`),
  KEY `Bno8` (`Bno`),
  CONSTRAINT `Bno8` FOREIGN KEY (`Bno`) REFERENCES `buildings` (`Bno`),
  CONSTRAINT `Dno` FOREIGN KEY (`Dno`) REFERENCES `domitory` (`Dno`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

2.宿舍信息（domitory）

```
CREATE TABLE `domitory` (
  `Bno` char(6) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '楼号',
  `Dno` char(6) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '宿舍号',
```

```
`Sno` char(12) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '宿舍长学号',
PRIMARY KEY (`Dno`),
KEY `Sno` (`Sno`),
KEY `Bno3` (`Bno`),
CONSTRAINT `Bno3` FOREIGN KEY (`Bno`) REFERENCES `buildings` (`Bno`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

3.宿舍财产信息（d_property）

```
CREATE TABLE `d_property` (
`Dno` char(6) NOT NULL COMMENT '宿舍号',
`Pno` char(7) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '物品号',
`Pnam` char(20) NOT NULL COMMENT '物品名',
`Bno` char(6) NOT NULL COMMENT '楼号',
PRIMARY KEY (`Pno`),
KEY `Dno2` (`Dno`),
KEY `Bno2` (`Bno`),
CONSTRAINT `Bno2` FOREIGN KEY (`Bno`) REFERENCES `buildings` (`Bno`),
CONSTRAINT `Dno2` FOREIGN KEY (`Dno`) REFERENCES `domitory` (`Dno`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

4.公共设施信息（facility）

```
CREATE TABLE `facility` (
`Bno` char(6) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '楼号',
`Fno` int NOT NULL COMMENT '设施号',
`Fnam` char(20) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '设施名',
```

```
    PRIMARY KEY (`Fno`),
    KEY `Bno5` (`Bno`),
    CONSTRAINT `Bno5` FOREIGN KEY (`Bno`) REFERENCES `buildings` (`Bno`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

5.报修信息（repair）

```
CREATE TABLE `repair` (
    `Ono` char(10) NOT NULL COMMENT '订单号',
    `Bno` char(6) NOT NULL COMMENT '楼号',
    `Dno` char(6) NOT NULL COMMENT '宿舍号',
    `Rno` char(7) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '维修物品（设施）号',
    `Rsub` date NOT NULL COMMENT '提交日期',
    `Rsol` date NOT NULL COMMENT '解决日期',
    `Rrea` char(50) NOT NULL COMMENT '报修原因',
    PRIMARY KEY (`Ono`) USING BTREE,
    UNIQUE KEY `Ono` (`Ono`) USING BTREE,
    KEY `Pno` (`Rno`),
    KEY `Bno` (`Bno`),
    CONSTRAINT `Bno` FOREIGN KEY (`Bno`) REFERENCES `buildings` (`Bno`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

6.离返校信息（l&a-school）

```
CREATE TABLE `l&a-school` (
    `Sno` char(12) NOT NULL COMMENT '学号',
    `Llea` date NOT NULL COMMENT '离校时间',
    `Lret` date NOT NULL COMMENT '返校时间',
    PRIMARY KEY (`Sno`),
    CONSTRAINT `Sno2` FOREIGN KEY (`Sno`) REFERENCES `students` (`Sno`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

7.晚归信息（arrive-late）

```
CREATE TABLE `arrive-late` (
  `Sno` char(12) NOT NULL COMMENT '学号',
  `Btim` timestamp(6) NOT NULL COMMENT '晚归时间',
  `Brea` char(20) NOT NULL COMMENT '晚归原因',
  PRIMARY KEY (`Sno`),
  CONSTRAINT `Sno` FOREIGN KEY (`Sno`) REFERENCES `students` (`Sno`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

8.访客人员信息（visitors）

```
CREATE TABLE `visitors` (
  `Vnam` char(20) NOT NULL COMMENT '姓名',
  `Bno` char(6) NOT NULL COMMENT '楼号',
  `Vtim` timestamp(6) NOT NULL COMMENT '到访时间',
  `Vtel` char(11) NOT NULL COMMENT '电话',
  `Vrea` char(50) NOT NULL COMMENT '到访原因',
  PRIMARY KEY (`Vnam`,`Vtel`),
  KEY `Bno9` (`Bno`),
  CONSTRAINT `Bno9` FOREIGN KEY (`Bno`) REFERENCES `buildings` (`Bno`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

9.管理人员信息（roots）

```
CREATE TABLE `roots` (
  `Ano` char(12) NOT NULL COMMENT '工号',
  `Anam` char(20) NOT NULL COMMENT '姓名',
  `Atel` char(11) NOT NULL COMMENT '电话',
  `Bno` char(6) CHARACTER SET utf8 COLLATE utf8_general_ci DEFAULT NULL COMMENT '楼号',
  PRIMARY KEY (`Ano`,`Atel`) USING BTREE,
```

```
    KEY `Bno7` (`Bno`),
    CONSTRAINT `Bno7` FOREIGN KEY (`Bno`) REFERENCES `buildings` (`Bno`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

10.用户信息（users）

```
CREATE TABLE `users` (
    `Uid` char(12) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '用户 ID',
    `Upw` char(20) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '用户密码',
    `Utyp` tinyint(1) NOT NULL COMMENT '用户类型',
    PRIMARY KEY (`Uid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

11.宿舍楼信息（buildings）

```
CREATE TABLE `buildings` (
    `Bno` char(6) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '楼号',
    `Bty` tinyint(1) NOT NULL COMMENT '类型',
    `Ano` char(12) CHARACTER SET utf8 COLLATE utf8_general_ci DEFAULT NULL COMMENT '管理员工号',
    PRIMARY KEY (`Bno`),
    KEY `Ano` (`Ano`),
    CONSTRAINT `Ano` FOREIGN KEY (`Ano`) REFERENCES `roots` (`Ano`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

3.4.2 数据库关系图

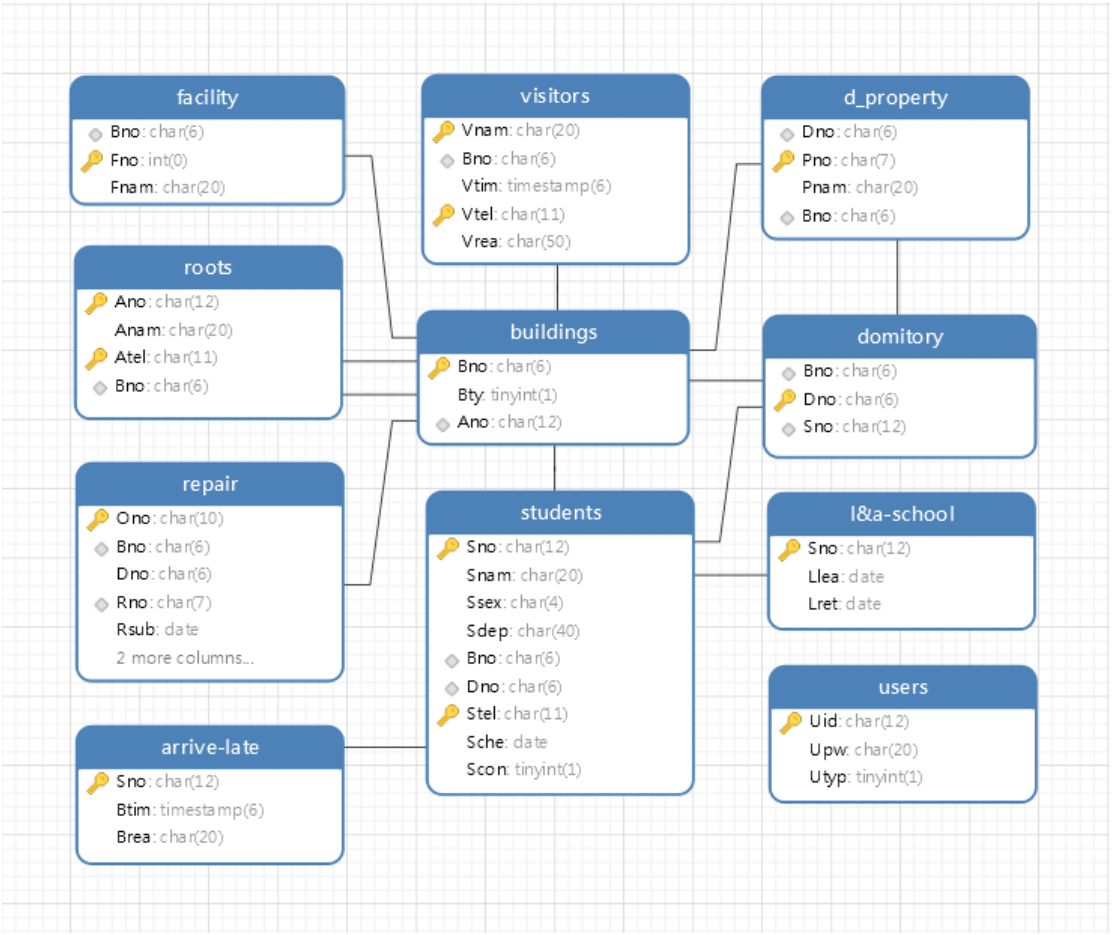


图 3.9 数据库关系图

第 4 章 系统实现

4.1 实现技术及环境

宿舍管理系统运用的技术：有 java、javaweb、数据库和 HTML5

数据库环境 ： MYSQL

数据库可视化界面 ： Navicat Premium

Java 环境：Eclipse

HTML5 运行环境 ： 谷歌浏览器

4.2 模块与功能

4.2.1 学生模块

1. 登录

登录需要输入学生的账号与密码，输入账号密码后交由数据库进行查询判断如果账号与密码相匹配则进入管理员界面，否则提示账号或密码错误。

```
// 实现查询用户信息
public LoginUser queryByUid(String uid) {

    LoginUser loginUser = null;
    PreparedStatement pstmt = null;
    Connection conn = null;
    ResultSet rs = null;
    try {

        String sql = "SELECT * FROM USERS WHERE UID=?";
        Object [] params={uid};
        rs=DBUtil.executeQuery(sql, params);


        if (rs.next()) {
            String upw = rs.getString("Upw");
            int utyp = rs.getInt("Utyp");

            return loginUser = new LoginUser (uid,upw,utyp);

        }

        return null;
```

图 4.1 学生登录

2. 修改信息

学生可以修改自己非主属性的信息，如电话和性别。

```
// 更新学生信息
public boolean updateBySno(Student student) {

    String sql="UPDATE students SET ssex=? ,sdep=? ,bno=? ,dno=?,stel=? WHERE sno=?";
    Object []params ={student.getSsex(),student.getSdep(),student.getBno(),student.getDno(),
    return  DBUtil.executeUpdate(sql, params);
}
```

图 4.2 学生信息修改

3. 修改密码

```
// 完成用户的密码更改
public boolean update(LoginUser loginUser) {

    String sql="UPDATE users SET upw=? WHERE uid=?";
    Object []params ={loginUser.getUpw(), loginUser.getUid()};
    return  DBUtil.executeUpdate(sql, params);
}
```

图 4.3 学生密码修改

4. 设施报修

```
// 完成添加报修信息
public boolean addRepair(Frepair frepair) {

    String sql="INSERT INTO  f_repair VALUES(?,?,?,?,?)";
    Object []params ={frepair.getBno(), frepair.getFno(),frepair.getRsub(),null,frepair.getRrea(
    return  DBUtil.executeUpdate(sql, params);
}
```

图 4.4 设施报修

5. 申请离校

```
// 增加 离校申请
public boolean addLASchool(LASchool laSchool) {

    String sql="INSERT INTO  laschool  VALUES(?,?,?)";
    Object []params ={laSchool.getSno(), laSchool.getLlea(),laSchool.getLret()};
    return  DBUtil.executeUpdate(sql, params);
}
```

图 4.5 申请离校

4.2.2 管理员模块

1. 登录

登录需要输入管理员的账号与密码，输入账号密码后交由数据库进行查询判断，如果账号与密码相匹配则进入管理员界面，否则提示账号或密码错误。

```
// 实现查询用户信息
public LoginUser queryByUid(String uid) {

    LoginUser loginUser = null;
    PreparedStatement pstmt = null;
    Connection conn = null;
    ResultSet rs = null;
    try {

        String sql = "SELECT * FROM USERS WHERE UID=?";
        Object [] params={uid};
        rs=DBUtil.executeQuery(sql, params);


        if (rs.next()) {
            String upw = rs.getString("Upw");
            int utyp = rs.getInt("Utyp");

            return loginUser = new LoginUser (uid,upw,utyp);

        }

        return null;
```

图 4.6 管理员登录

2. 查看学生信息

管理员可以查看宿舍楼内所有学生的信息，包括：学号，姓名，性别，专业，楼号等。

```
//  查询所有学生
    public List<Student> queryAllStudents() {
        List<Student> students =new ArrayList<>();
        Student student = null;
        PreparedStatement pstmt = null;
        ResultSet rs=null;
        try {
            String sql ="SELECT * FROM STUDENTs ";
            rs=DBUtil.executeQuery(sql, null);

            while (rs.next()) {
                String sno = rs.getString("Sno");
                String sname = rs.getString("Sname");
                String ssex = rs.getString("Ssex");
                String sdep = rs.getString("Sdep");
                String bno = rs.getString("Bno");
                String dno = rs.getString("Dno");
                String stel = rs.getString("Stel");
                Date sche =rs.getDate("Sche");
                int scon =rs.getInt("Scon");

                student =new  Student(sno, sname, ssex, sdep, bno, dno, stel, sche, scon);
                students.add(student);

            }
```

图 4.7 查看学生信息

3. 查看访客记录

管理员可以查看到访宿舍的访客。

```
// 查询所有访客
    public List<Visitor> queryAllVisitor() {

        List<Visitor> visitors =new ArrayList<>();
        Visitor visitor = null;
        PreparedStatement pstmt = null;
        ResultSet rs=null;
        try {
            String sql ="SELECT * FROM visitors ";
            rs=DBUtil.executeQuery(sql, null);

            while (rs.next()) {
                String vnam = rs.getString("Vnam");
                String bno = rs.getString("Bno");
                Date vtim = rs.getDate("Vtim");
                String vtel = rs.getString("Vtel");
                String vrea = rs.getString("Vrea");

            //System.out.println(vnam+" aaaa");
                visitor =new  Visitor(vnam, bno, vtim, vtel, vrea);
                visitors.add(visitor);

            }
```

图 4.8 查看访客记录

4. 记录访客信息

管理员将访客的信息记录在数据库中，包括：姓名，到访楼号，到访时间，访客电话和到访原因。

```
// 增加访客
    public boolean addVisitor(Visitor visitor) {

        String sql="INSERT INTO  visitors VALUES(?,?,?,?,?)";
        Object []params ={ visitor.getVnam(),  visitor.getBno(), visitor.getVtim(), visitor.getVtel
        return  DBUtil.executeUpdate(sql, params);
    }
```

图 4.9 记录访客记录

5. 查询设施报修

管理员可以查看学生所上报的设备报修

```
// 查询所有 报修记录
public List<Frepair> queryAllFrepair() {

    List<Frepair> frepairs =new ArrayList<>();
    Frepair frepair = null;
    PreparedStatement pstmt = null;
    ResultSet rs=null;
    try {
        String sql ="SELECT * FROM f_repair ";
        rs=DBUtil.executeQuery(sql, null);

        while (rs.next()) {
            String bno = rs.getString("Bno");
            int fno = rs.getInt("Fno");
            Date rsub = rs.getDate("Rsub");
            Date rsol = rs.getDate("Rsol");
            String rrea = rs.getString("Rrea");

            frepair =new  Frepair(bno, fno, rrea, rsub, rsol);
            frepairs.add(frepair);

        }

        return frepairs;
```

图 4.10 查询设施报修

6. 查询学生离校信息

```
// 查询所有 离校记录
public List<LASchool> queryAllLASchool() {

    List<LASchool> laSchools =new ArrayList<>();
    LASchool laSchool = null;
    PreparedStatement pstmt = null;
    ResultSet rs=null;
    try {
        String sql ="SELECT * FROM laschool ";
        rs=DBUtil.executeQuery(sql, null);

        while (rs.next()) {
            String sno = rs.getString("Sno");

            Date llea = rs.getDate("Llea");
            Date lret = rs.getDate("Lret");


            laSchool =new  LASchool(sno, llea, lret);
            laSchools.add(laSchool);

        }

        return laSchools;
```

图 4.11 查询离校信息

第 5 章 结论与展望

5.1 结论

在报告前面四个章节，我们介绍了整个宿舍管理系统的系统核心的内容，设计构想及单元模块的实现方式，证明整个系统与管理方法合理。并在小组组员生活中对本系统进行实验与运用以证明其实用性。在此，对所做工作进行总结,并提出系统的改设想以及展望。

本次软件课程设计中，计算 183 班陈杨瑞禧-田显阳-杨金树组使用了 java、javaweb 与 html 语言，开发了一款面向青岛理工大学以及全体大学生的宿舍管理系统，主要实现了学生入住信息管理，宿舍财产管理及报修，公共财产管理及报修，外来访客信息管理，学生离返校信息管理等功能。

在对系统性能的评估中，展现了对服务器的要求性，在目前本组成员的测试中所搭建的服务器，可满足 50 人左右的适用规格，若对服务器性能进行提升，将满足更多的用户进行使用。

由于课程设计的局限性与特殊性，本系统仅作为小组成员的设计试验品以及课设作业进行提交，故功能上有所限制与削减，最大的不足就在于时间的不足导致的系统功能不全，无法真实完善的满足现实生活中学生的需求与功能。

5.2 进一步工作的方向

本次课程设计的研究虽然取得了初步的成功，但依然任重道远，尚有许多有待进一步深入进行的研究工作，这里择其要者简要讨论如下：

关于系统维护：

完善性维护构想，当前设想可添加的功能有：

学生晚归信息管理功能。

校园邮件寄收功能。

宿舍外卖系统。

宿舍浴室预约功能。

账户注册功能

账户密码找回功能

适应性维护展望：公寓系统应不仅存在于网页，更可移植成微信小程序，或移植到 windows 端与 MacOS 端完善成一电脑程序，方便用户使用。

关于数据读取：

应设计与教务系统的联动进行学生信息的读取。

5.3 展望

通过本次的软件工程课程设计，使小组成员整体受益良多，我们明白了自己原来知识还欠熟练，更多课本上学不到的知识需要自己去亲身实践才能够学会，学以致用，因用致学，我们要学习的东西还很多,学习是一个长期积累的过程,计算机科学与技术更是一个成长中的专业，我们在以后的工作、生活中都应该不断的学习,努力提高自己的知识素养和综合素质，在本次课程设计中更培养了我们独立工作与思考的能力,激发了我们的创新意识。使我充分体会到了在创造过程中的探索的艰难和成功的喜悦，虽然这个项目还不是很完善 但是在设计过程中所学到的东西是这次课程设计的最大收获和财富,它将使我们终身受益。