|  |  |
| --- | --- |
| **文档名称：** | **软件设计说明书** |
| **项目名称：** | **西师宿管** |

|  |  |
| --- | --- |
| **团队编写：** | **202014** |
| **团队组长：** | **温腾虎** |
| **团队成员：** | **常梦娇，料堃焱，司绍斌** |
| **日期：** | **2020年06月04日** |
| **院系：** | **计算机科学与工程学院** |
| **专业：** | **计算机科学与技术** |

目录

**[1. 引言 3](#_Toc17325)**

[1.1编写目的 3](#_Toc11291)

[1.2项目背景 3](#_Toc26318)

**[2.概述 3](#_Toc32498)**

[2.1目标 3](#_Toc12604)

[2.2运行环境 4](#_Toc28654)

[2.3条件与限制 4](#_Toc13152)

**[3.总体设计 4](#_Toc28965)**

[3.1系统结构 4](#_Toc4023)

[3.2系统管理模块 6](#_Toc16077)

[3.3宿管系统模块 7](#_Toc12203)

[3.4学籍信息系统模块 8](#_Toc26937)

[3.5后勤管理系统模块 9](#_Toc5462)

[3.6宿舍信息系统模块 9](#_Toc28150)

[3.6关于系统模块 10](#_Toc26337)

**[4.数据库逻辑结构 10](#_Toc18635)**

[4.1 E-R图： 10](#_Toc15573)

[4.2 E-R 图向关系模型的转换 10](#_Toc5313)

**[5.软件重用方案 12](#_Toc5011)**

[5.1知识重用方案 12](#_Toc23374)

[5.2方法和标准的重用方案 12](#_Toc31168)

[5.3软件成分的重用方案 13](#_Toc28175)

[5.4类构件实现软件重用方案设计 14](#_Toc32076)

**[6.设计关键类的重点服务 15](#_Toc3018)**

**[7.接口设计 15](#_Toc27037)**

[7.1用户接口 15](#_Toc14959)

[7.2外部接口 16](#_Toc13956)

[7.3内部接口 16](#_Toc11864)

**[8.运行设计 16](#_Toc7063)**

[8.1运行模块组合 16](#_Toc19158)

[8.2运行控制 16](#_Toc14482)

[8.3运行时间 16](#_Toc7079)

**[9.系统出错处理设计 17](#_Toc20862)**

[9.1出错信息 17](#_Toc9955)

[9.2补救措施 17](#_Toc17766)

[9.3系统维护设计 17](#_Toc17271)

# 引言

## 1.1编写目的

在编写《西师宿管》软件之前，对同类型产品的市场进行了前期调查，与多位软件设计者和使用者进行了探讨和分析，之后由软件项目小组共同协商提出了这份软件设计说明书。该软件设计说明书对学生宿舍管理系统软件进行了全面细致的用户需求分析，明确所要开发的软件应具有的功能、性能与界面。

## 1.2项目背景

学校有若干公寓，需要一个公寓管理系统实现寝室分配、学生管理、信息查询、财产管理、出入登记、系统管理等功能。经过实际分析调查，一个学生宿舍管理系统可以方便宿舍管理人员的工作，提高其工作效率，不仅可以方便广大学生，而且还可以方便于教师和相关人员。

要求系统有效、快速、安全、可靠和无错误的完成需要的操作，并要求客户机的界面要简单明了，易于操作。服务器程序易于维护。

# 2.概述

## 2.1目标

本设计要实现的是高校学生宿舍管理系统，在设计该系统时，应尽可能的贴近学生,便于用户操作。系统在实现上应该具有如下功能：

1.系统要求用户必须输入正确的用户名和密码才能进入系统。

2.系统应该提供学生住宿情况的基本登记。

3.系统应提供学生打卡功能，疫情期间提供提问上报功能。

4.系统应提供学生在线提交报修信息以及充值电卡水卡的功能。

5.系统应提供查询功能，以方便用户对学生基本信息的查询(要实现按多种条件的查询)及楼房信息的查询。

6.系统应提供增加、删除、修改用户帐户的功能。

## 2.2运行环境

|  |  |
| --- | --- |
| CPU | Pentium Ⅳ |
| 内存 | 512MB |
| 硬盘空间 | 以上硬盘剩余空间 |
| 输入设备 键盘/鼠标 |  |
| 操作系统 | Windows XP Server以上 |
| 数据库 | MySql |
| 开发环境 | Eclipse，IDEA |
| 数据库工具 | Navicat |

## 2.3条件与限制

由于时间比较短，使用计算机不方便以及对于网络编程不熟悉，本学生宿舍管理系统并没有提供数据的远程访问功能。对信息的保护手段仅限于设置用户级别，以及提供数据文件的备份，比较简单，安全性能有待进一步完善。由于硬件有局限性，电卡以及水卡充值功能可能具体实现有问题，但是可以留下一个借口，等以后硬件完善之后再进行添加。由于系统开发人员较少，后期的维护工作比较难以进行，希望进一步完善。

# 3.总体设计

## 3.1系统结构

MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

Model（模型）是应用程序中用于处理应用程序数据逻辑的部分。

　　通常模型对象负责在数据库中存取数据。

View（视图）是应用程序中处理数据显示的部分。

　　通常视图是依据模型数据创建的。

Controller（控制器）是应用程序中处理用户交互的部分。

　　通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

MVC 分层有助于管理复杂的应用程序，因为您可以在一个时间内专门关注一个方面。例如，您可以在不依赖业务逻辑的情况下专注于视图设计。同时也让应用程序的测试更加容易。

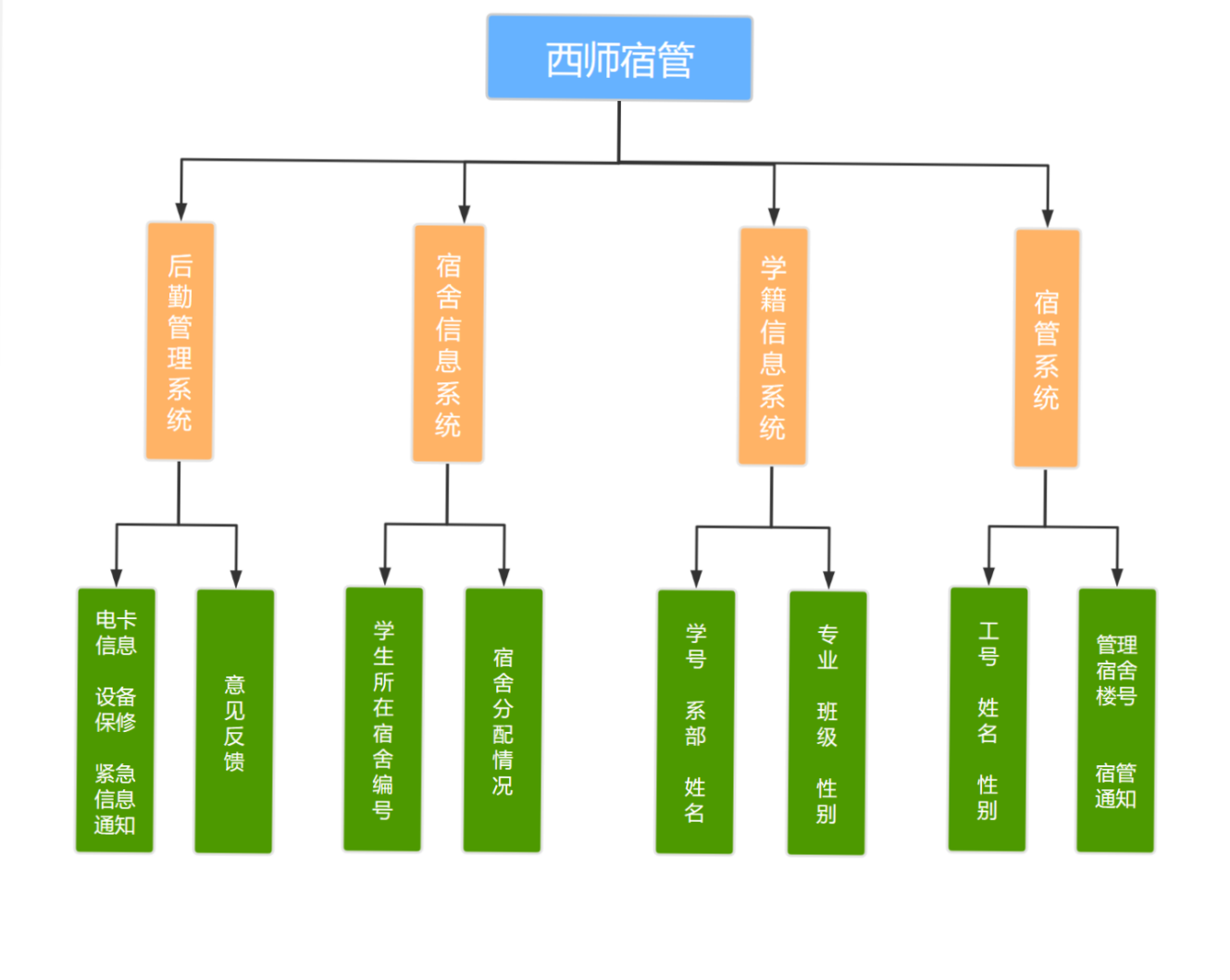
B/S结构（Browser/Server，浏览器/服务器模式），是WEB兴起后的一种网络结构模式，WEB浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器，如Netscape Navigator或Internet Explorer，服务器安装SQL Server、Oracle、MYSQL等数据库。浏览器通过Web Server 同数据库进行数据交互。

第一层是浏览器，即客户端，只有简单的输入输出功能，处理极少部分的事务逻辑。由于客户不需要安装客户端，只要有浏览器就能上网浏览，所以它面向的是大范围的用户，所以界面设计得比较简单，通用。

第二层是WEB服务器，扮演着信息传送的角色。当用户想要访问数据库时，就会首先向WEB服务器发送请求，WEB服务器统一请求后会向数据库服务器发送访问数据库的请求，这个请求是以SQL语句实现的。

第三层是数据库服务器，他扮演着重要的角色，因为它存放着大量的数据。当数据库服务器收到了WEB服务器的请求后，会对SQL语句进行处理，并将返回的结果发送给WEB服务器，接下来，WEB服务器将收到的数据结果转换为HTML文本形式发送给浏览器，也就是我们打开浏览器看到的界面。

在 B/S 结构下的该寝室管理系统大致可以划分为以下模块：

****

在整体设计中，我们将宿舍管理系统分为六个大的模块：系统管理模块、宿管系统、学籍信息系统、后勤管理系统、宿舍信息系统、关于模块。每个模块将实现不同的功能。下面将具体进行介绍。

## 3.2系统管理模块

系统管理模块包括：超级用户登录、普通宿管用户登录、 学生用户登录、用户管理、退出系统五部分。

1. 超级用户管理：实现超级管理员的登录，可以进行各级用户权限管理，创建新用户等功能。
2. 普通宿管用户登录：实现普通宿管用户登录，可以进行学生基本信息管理，包括修改，更新，删除等操作。
3. 学生用户登录：实现学生用户登录，可以进行打卡，报修，体温上报等功能。
4. 用户管理：实现各级用户的自我管理，可以进行查看基本信息等功能，管理人员授予或取消一般用户登录该系统的用户名和密码。
5. 退出系统：实现正常退出宿舍管理系统

|  |  |
| --- | --- |
| 算法逻辑 |  |

## 3.3宿管系统模块

宿管系统模块包括：通知更改功能，学生基本信息查看及修改功能，宿舍报修及充电卡功能三部分。

1. 通知更改功能：更改首页中的轮播图信息通告，保证实时有效。
2. 学生基本信息查看及修改功能：查看学生基本信息，如果需要修改则进行修改，删除等操作，还可以进行高级查询。
3. 宿舍报修及充电卡功能：实时查看那些宿舍需要报修及充电卡的需求，和及时和后期部门联系，进行维修或者充电。

|  |  |
| --- | --- |
| 算法逻辑 |  |

## 3.4学籍信息系统模块

学籍信息功能包括查看学生的学号，姓名，学院，专业等基本信息，查看宿舍号，楼号等公寓信息，宿管进行修改，查询，删除等操作。

1. 查看学生基本信息：宿管登录后台可以查看学生基本信息，学生登录后可以查看自己的基本信息。
2. 修改基本信息：宿管可以进行信息的查询，修改，删除学生的基本信息操作。

## 3.5后勤管理系统模块

后勤管理系统包括报修信息填报功能，电卡充值功能，紧急通知，意见反馈四部分。

1. 报修信息填报：实现学生登录后可以进行报修信息填报的功能，宿管后台实时查看。
2. 电卡充值功能：实现学生登录后可以进行电卡充值的功能，宿管后台实时查看。
3. 紧急通知：宿管从后台发布紧急通知的功能，并且学生在登录首页就可以查看。
4. 意见反馈：学生可以进行意见反馈，通过留言的形式给宿管后台留言提意见。

## 3.6宿舍信息系统模块

宿舍信息系统包括学生宿舍分配情况，空宿舍查询，调整宿舍管理三部分。

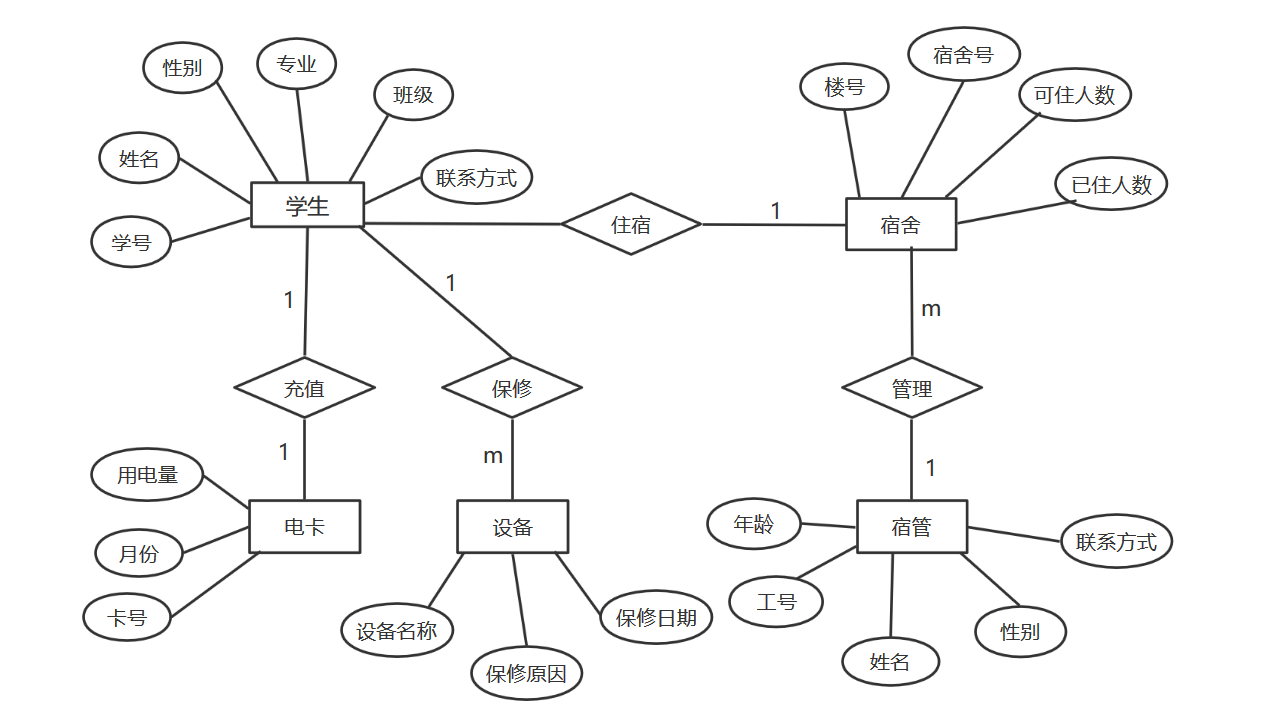
1. 学生宿舍分配情况：宿管可以查看每个学生的宿舍分配情况，尤其是大一新生的宿舍分配。
2. 空宿舍查询：对每一间宿舍都进行管理，查询空宿舍情况。
3. 调整宿舍管理：学生可以进行宿舍调整申请，宿管根据申请情况进行查看，选择同意与否。

## 3.6关于系统模块

此模块是对学生宿舍管理系统开发者相关信息的一个简单介绍。

# 4.数据库逻辑结构

## 4.1 E-R图：



## 4.2 E-R 图向关系模型的转换

**学生表：学号、姓名、性别、专业、班级、联系方式 主键：学号**

**宿舍表：楼号、宿舍号、可住人数、已住人数 主键：楼号、宿舍号**

**报修表：物品名称、报修原因、报修日期、报修人 主键：报修日期 外键：报修人**

**电卡表：月份、用电量、卡号 主键：月份**

**宿管表：姓名、性别、年龄、管理宿舍楼号、工号、联系方式 主键：工号 外键：管理宿舍楼号**

**学生报修表：宿舍号、楼号 主键：宿舍号、楼号**

**管理表：楼号、宿舍号、职工号 主键：宿舍号、楼号**

# 5.软件重用方案

## 5.1知识重用方案

知识如此珍贵，最好能重复使用：不仅自己使用，也让别人使用。让自己人不用花代价，就可以获得知识。于是，知识传播也变得重要起来。人类的祖先“智人”之所以能够战胜“尼人”，一个重要的原因就是“智人”可以用语言传播知识。设备远程维护最好让设备开发商参与，就是因为他们便于知识复用。我们国家制造业的优势在于市场大——市场大的好处就是便于知识的重用。在有些行业中，企业越来越大。其中重要的“粘合剂”就是知识。

一般来说，知识需要在使用的过程中才能不断丰富、完善。这就叫“从实践中来，到实践中去”。实践多了，才能走向成熟。知识成熟了，使用的风险就小。所以，阿波罗计划拒绝采用不成熟的技术。进入现代工业社会以后，企业都要求按照标准化进行生产。而“标准”就是一种成熟以后固化起来的知识。在智能化的时代，知识可以固化在计算机里，自动地使用知识；可以在互联网上传播，极大地促进知识的重用。知识被重用的次数多了，获得知识的成本就可以被摊平，从而进一步促进知识的产生。而大数据的时代，能够帮助人类方便地获得更多的知识——甚至包括图像识别这样不容易描述的“感性知识”。所以，认识智能化时代的一个角度，是知识经济。

对于本系统，学习过的知识可以得到充分利用，比如数据库，JAVAEE，WEB前端技术等。对于我们软件需求分析以及后期软件开发都有很好的作用。

## 5.2方法和标准的重用方案

对于本系统，采用面向对象的方法，对于标准的重用，我们采用国家规定的软件开发规范。整体遵守代码规范，对每个人的编码都进行规范化，使得后期维护方便快捷，代码整体结构清晰。定义通用的接口以及方法，使用过程中直接调用即可。

## 5.3软件成分的重用方案

软件重用分类比较困难，因为软件重用技术众多，一种重用技术可以包括多种重用形式。比如说：框架即可以包括代码级重用，也可以包括设计级重用。有一种分类方法是按照软件重用所应用的领域范围，把重用划分为两种：横向重用和纵向重用。

1. 横向重用是指重用不同应用领域中的软件元素，例如数据结构、分类算法、人机界面构件等。标准函数库是一种典型的原始的横向重用机制。
2. 纵向重用是指在一类具有较多公共性的应用领域之间进行软部品重用。因为在两个截然不同的应用领域之间实施软件重用非常困难，潜力不大，所以纵向重用才广受瞩目，并成为软件重用技术的真正所在。纵向重用活动的主要包括以下几个步骤：

1) 首先进行域分析。根据应用领域的特征及相似性预测软部件的可重用性。

2) 然后进行软部品的开发。一旦确认了软部件的重用价值，即可进行软部品的开发并对具有重用价值的软部品进行一般化，以便它们能够适应新的类似的应用领域。

3) 最后，软部件及其文档即可进入软部品库，成为可供后续项目使用的可重用资源。

在本系统中，在后期编码的过程中，使用调用库函数，实现代码重用，可以大大提高代码的效率。对于编写的源代码、用户界面的设计、数据等都可以重用，比如源代码的编写中，所要用到的一些前端框架，我们就可以结合自身本系统的需求进行分析，然后进行框架整合，这样就减少了我们的编码工作量，有助于加快项目进度。在UI界面中，可以重用之前的原型设计中设计好的界面，对应进行设计，因为已经对原型进行了调研和试用，所以重用原型设计来设计界面，可以保证我们最后研发出来的产品是满足用户审美的，是符合用户需求的。对于数据重用，比如数据库中表的设计，我们可以对逻辑分析中的E-R图进行加工，根据E-R图进行建表等工作，根据逻辑分析设计数据库。

## 5.4类构件实现软件重用方案设计

利用面向对象技术,可以更方便更有效地实现软件重用。面向对象技术中的“类”,是比较理想的可重用软构件,不妨称之为类构件。类构件有3种重用方式,分别是实例重用、继承重用和多态重用。下面进一步讲述与类构件有关的内容。

类构件的重用方式

1. 实例重用

由于类的封装性,使用者无须了解实现细节就可以使用适当的构造函数,按照需要创建类的实例,然后向所创建的实例发送适当的消息,启动相应的服务,完成需要完成的工作,这是最基本的重用方式。此外,还可以用几个简单的对象作为类的成员创建出一个更复杂的类,这是实例重用的另一种形式。

虽然实例重用是最基本的重用方式,但是,设计出一个理想的类构件并不是一件容易的事情。例如,决定一个类对外提供多少服务就是一件相当困难的事,提供的服务过多会增加接口复杂度,也会使类构件变得难于理解;提供的服务过少,则会因为过分一般化失去重用价值。每个类构件的合理服务数都与具体应用环境密切相关,因此找到一个合理的折衷值是相当困难的。

(2)继承重用

面向对象方法特有的继承性提供了一种对已有的类构件进行裁剪的机制,当已有的类构件不能通过实例重用完全满足当前系统需求时,继承重用提供了一种安全地修改已有类构件,以便在当前系统中重用的手段。

为提高承重用的效果,关键是设计一个合理的、具有一定深度的类构件继承层次结构。这样做有下述两个好处:

1）每个子类在继承父类的属性和服务的基础上,只加入少量新属性和新服务,这不仅降低了每个类构件的接口复杂度,表现出一个清晰的进化过程,提高了每个子类的可理解性,面且为软件开发人员提供了更多可重用的类构件。因此,在软件开发过程中,应该时刻注意提取这种潜在的可重用构件,必要时应在领域专家帮助下,建立符合领域知识的继承层次。

2）为多态重用奠定良好基础

(3)多态重用

利用多态性不仅可以使对象的对外接口更加一般化(基类与派生类的许多对外接口是相同的),从而降低了消息连接的复杂程度,而且还提供了一种简便可靠的软构件组合机制,系统运行时,根据接收消息的对象类型,由多态性机制启动正确的方法,去响应一个一般化的清息,从而简化了消息界面和软构件连接过程。

对于本系统我们使用继承重用，对于面向对象语言来说，必须要保证它的可封装性，就要用到继承机制来达到代码复用的目的。对于接口重用，我们考虑用类指针调用派生类的方法来实现接口重用。比如在后期的编码中，学籍信息类可以学生类类。

为了提高重用的效果,关键是设计一个合理的、具有一定深度的类构件继承层次结构，这样每个子类在继承父类的属性和服务的基础上,只需要加入少量新属性和新服务,这不仅降低了每个类构件的接口复杂度,表现出一个清晰的进化过程,提高了每个子类的可理解性,而且为软件开发人员提供了更多可重用的类构件。

# 6.设计关键类的重点服务

本系统根据状态图和数据流图可筛选出本项目的关键类及相应重点服务如下：

宿舍：可以提交报修申请，电卡充值申请等；

宿管人员：根据学生提交的申请进行人员分配，查看，修改以及删除学生基本信息。

学生用户：可以进行报修申请，电卡充值，打卡以及疫情上报。

# 7.接口设计

## 7.1用户接口

使用基于对话框的GUI，用户通过鼠标的点击和键盘的输入完成操作，编辑框用于用户的输入。

用浏览器登录网站，管理员以及学生通过不同的入口进行登录，只有用户名和密码对应正确，才可以登录进入。

## 7.2外部接口

通过JDBC-ODBC桥接口与My SQL数据库连接。在进行与服务器端口的连接，实现数据的交互。

## 7.3内部接口

系统基于MVC模式架构，数据库与系统之间通过JDBC进行连接，各个模块之间相互独立。

# 8.运行设计

## 8.1运行模块组合

1. 报修以及电卡充值模块：实现在线提交申请的功能，后台宿管进行查看的功能。
2. 基本信息查看功能：实现宿管管理以及查看学生基本信息，学生本身只可以查看自身的基本信息。
3. 打卡以及疫情上报模块：学生登录后台后进行每日打卡以及疫情上报功能，后台宿管可以进行实时查看。

## 8.2运行控制

从浏览器进入本系统网站，未登录状态下只可以进行查看公告的功能，登录后进行相应的操作。

## 8.3运行时间

本系统在网络运行顺畅的情况下，反应时间小于1S

在数据过大或者查询条件复杂时，相应时间控制在2S以内

# 9.系统出错处理设计

## 9.1出错信息

1. 输入密码位数不对时，提示密码出错。
2. 用户名为注册时，联系超级管理员新添加新的用户，并且赋予对应的权限。
3. 查询学生基本信息时出错，可以进行清除操作，重新进行查询。
4. 导出学生对应信息时，如果有卡顿，可以重新下载或者在浏览器下载中心查看是否已经下载了。

## 9.2补救措施

1. 提供丰富的出错提示信息；
2. 提供一定的保密手段。
3. 定期对系统进行维护，保证实时修改问题，在意见反馈中让用户积极上报系统出现的问题，有利于用户体验。
4. 定期对数据库进行备份，保证用户的隐私性安全。b.解决数据库使用期间所积累的问题，并对一些冗余数据进行清理。

## 9.3系统维护设计

1. 本系统是模块化设计，各个模块之间独立有相互联系，方便维护检查以及修改。
2. 数据库定期备份以及恢复。