

同济大学学生宿舍管理系统
体系结构设计文档



小组成员： 涂远鹏-1652262

刘铸煌-1652313

黎盛烜-1652310

指导老师： 王继成

目录

1. 引言	1
1.1. 编制目的	1
1.2. 词汇表	1
1.3. 参考资料	1
2. 产品概述	2
3. 体系结构模型	2
3.1. 整体架构描述	2
3.2. 用户层面模块分解	7
3.3. 用业务逻辑层模块分解	9
3.4. 数据层模块分解	11
3.5. 界面模块设计	13
3.6. 运行时组件	14
3.7. 物理部署	14
4. 详细需求描述	15
4.1. 对外接口需求	15
4.2. 功能需求	15
5. 系统体系结构设计思路	18

1 引言

1.1 编制目的

本报告详细完成对宿舍管理系统的概要设计，达到指导详细设计和开发的目的，同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

1.2 词汇表

词汇名称	词汇含义	备注
DM	宿舍管理系统	Dormitory manager
Bomb	云端服务器	在线云端服务器

1.3 参考资料

- 1) 郭霖. 第一行代码[M]. 人民邮电出版社:郭霖, 2014. 69-307
- 2) IEEE[1471-2000]标准
- 3) 骆斌. 软件工程与计算（卷二）——软件开发的技术基础[M]. 机械工业出版社:骆斌, 2016.82-118, 403-410
- 4) 校内运动场地管理系统软件体系结构设计文档

2 产品概述

同济大学宿舍的外来人员登记、强开宿舍电闸申请、个人信息查询等服务都只能在一楼的宿管处进行查询，而住在高层的学生经常需要从高层到一楼进行信息登记等操作，不仅浪费时间也给学生造成较大不便，另外有时遇上宿管休息的时候，而学生进行报修申请时，必定会叫醒宿管，从而打扰宿管休息，造成学生、宿管双方的不悦。另一方面人为记录的管理方式具有易出错，较繁琐，不易更改等诸多缺点。

另外由于近年来学生数量的增加，导致了每个学年开始以及结束之际，入宿退宿的学生人数增加，每个学年入宿退宿查询工作的手工工作不仅操作缓慢耗费大量时间，排队现象严重，浪费大量时间并且造成诸多不便，其次就是出错的概率较大，登记的信息优势较为繁琐，很容易出现错误的情况，并且宿舍管理员在查询退宿入宿填写信息时也难以将如此大量的信息中的错误信息全部纠正。其次就是学校希望通过在线管理的服务削减人员方面的开支，降低人力成本，统一化宿舍管理，为全校学生提供高效无误的服务。

同济大学宿舍管理系统就是为了满足同济大学宿舍管理发展要求而开发的。它包含一个数据集中服务器和若干个移动客户端，数据服务器将所有用户的数据集中存储便于管理与维护，用户则通过客户端完成报修申请、电费查询等便捷服务，客户端与数据集中服务器采用 TCP/IP 协议的实时通信的方式完成数据的交换与维护。

宿舍管理系统是为同济大学开发的校内宿舍管理运营系统, 开发的目的是帮助学校进行日常的宿舍管理以及信息发布。通过该系统, 用户可以查看学生的基本信息、宿舍信息等各方面的资料, 能够方便的了解学生和宿舍的总体情况。该管理系统分别为学生 (基本信息查询、报修申请、电费查询、入宿退宿状态查询)、宿舍管理员 (学生信息查询、学生住宿管理、离返校情况统计、卫生评比、报修管理、外来人员登记、楼层公告管理)、学校人事管理部门 (学生信息管理、宿管人员管理) 三类对象提供对应的在线服务。

通过在线宿舍管理系统的应用, 期望提高学校宿舍管理效率, 降低人力管理成本, 并减少人工登记所可能产生的错误, 不仅为学校管理层面提供便利的在线服务, 同时也为学生用户提供电费缴纳查询等便捷服务, 实现学校管理运营学生宿舍利润最大化, 并做到一些公示信息通知到每一个学生等在线便捷服务。

3 体系结构模型

3.1 整体架构描述

3.1.1 逻辑视角

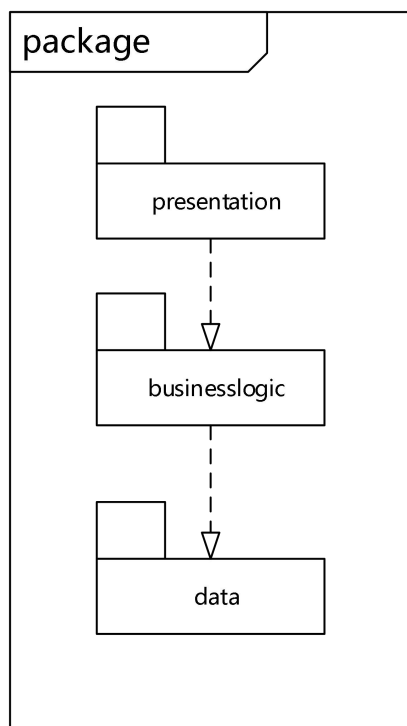
在同济大学学生宿舍管理系统中, 我们依旧选择了我们更为熟悉的分层风格, 将系统分为 3 层:

展示层 (presentation)

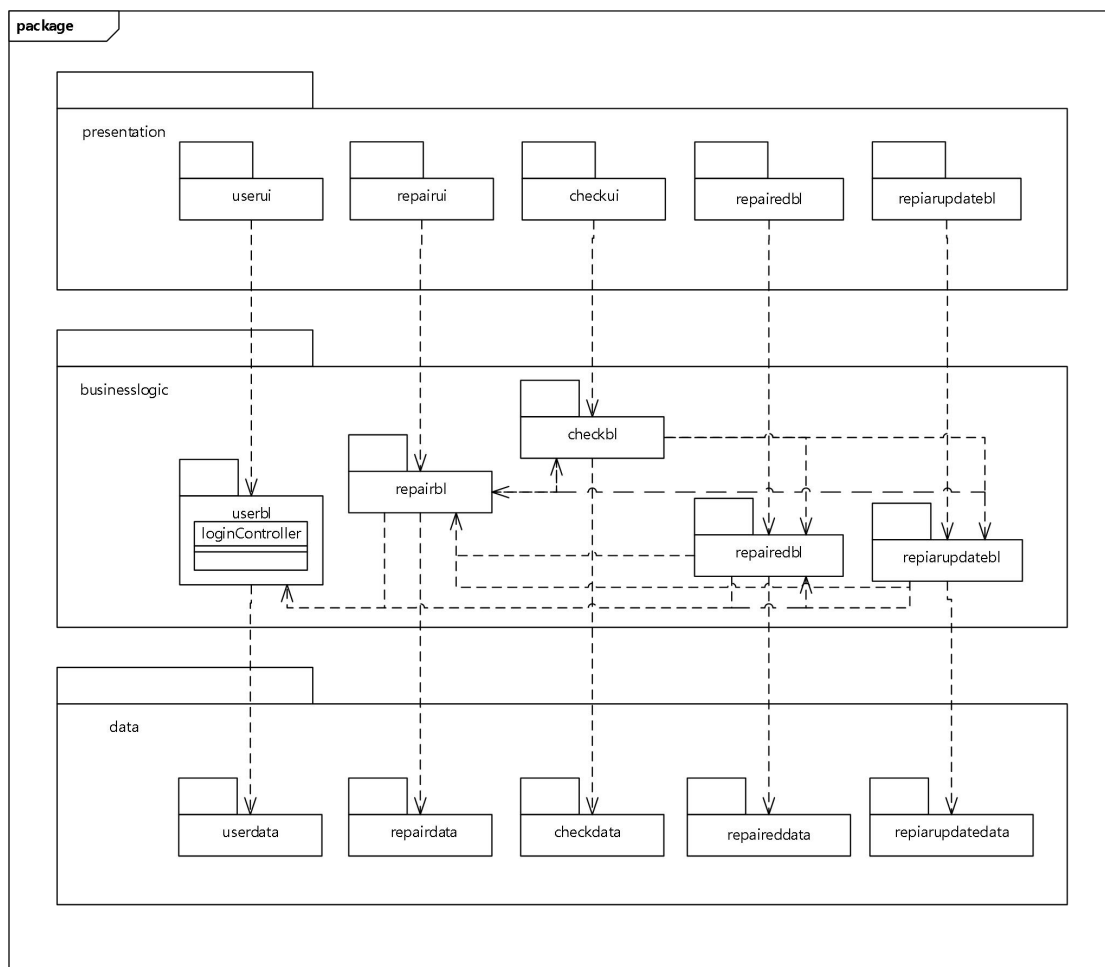
业务逻辑层 (businesslogic)

数据层 (data)

其中, 展示层包含 GUI 页面的实现, 业务逻辑层包含业务逻辑处理的实现, 数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如下:



参照体系结构风格的包图表达逻辑视角



软件体系结构逻辑设计方案

3.1.2 组合视角

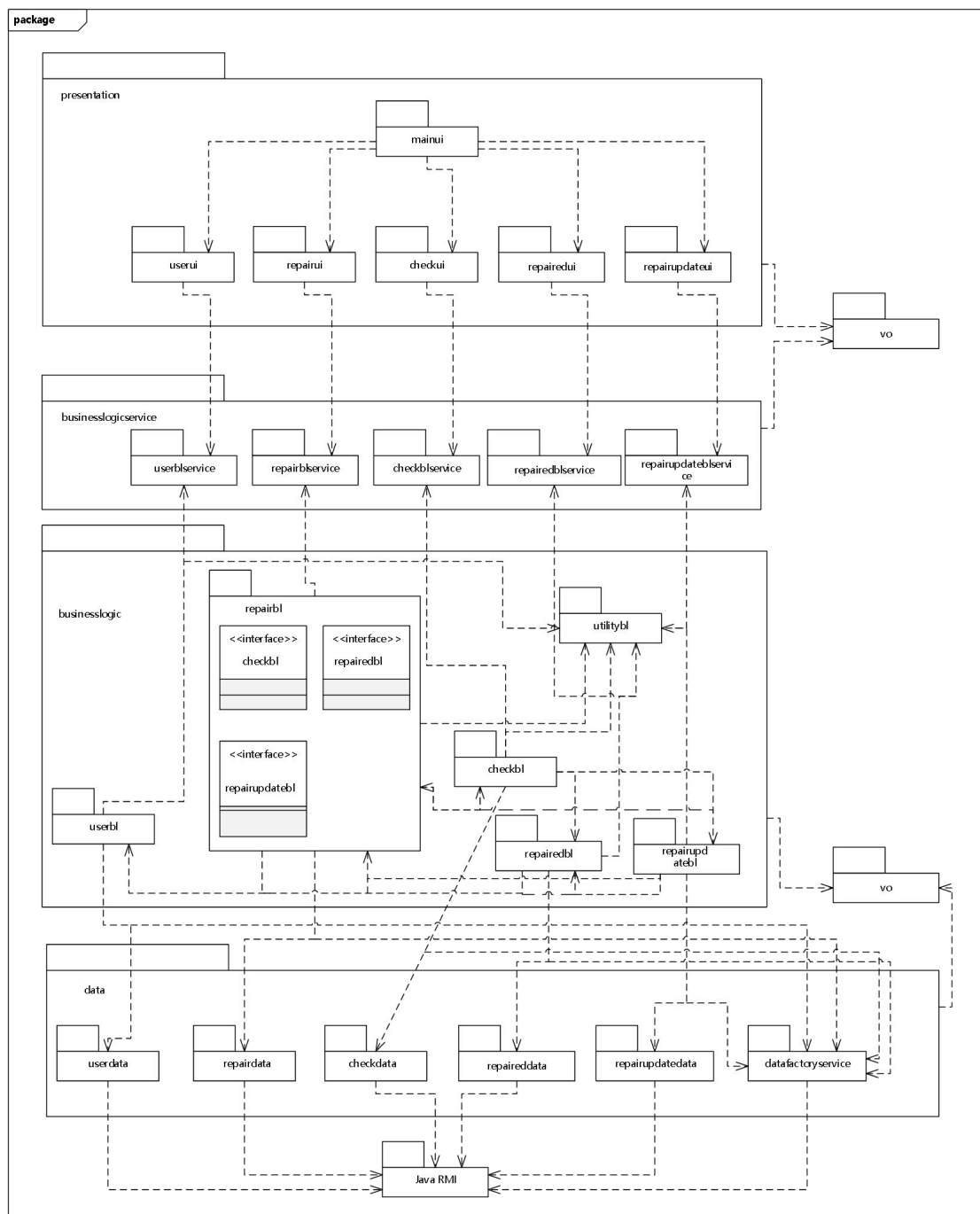
3.1.2.1 开发包图

同济大学学生宿舍管理系统的最终开发包设计表如下：

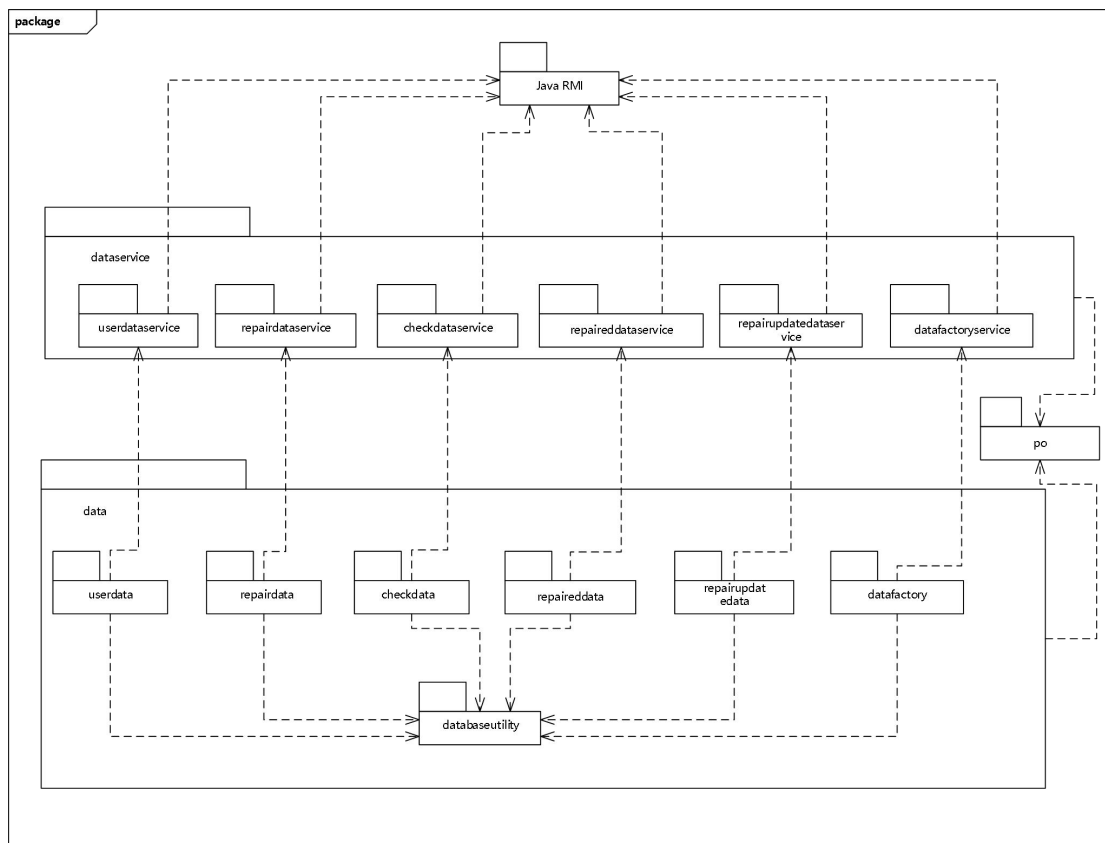
开发（物理）包	依赖的其他开发包
mainui	userui, reserveui, checkui, unsubscribeui, partnerui, vo
userui	userblservice, 界面类库包
userblservice	
userbl	UserInterface, UserDataClient, UserPO
userdataservice	Java RMI, po
userdata	RMI, po, databaseutility
repairui	repairblservice, 界面类库包, vo
repairblservice	
repairbl	repairblservice, repairdataservice, po, checkbl, userbl
repairdataservice	Java RMI, po
repairdata	databaseutility, po, repairblservice
checkui	checkblservice, 界面类库包
checkblservice	
checkbl	checkblservice, checkdataservice, po,

	reservebl,unsubscribebl,partnerbl
checkdataservice	Java RMI, po
checkdata	Java RMI, po, databaseutility
repairedui	repairedblservice, 界面类库包
repairedblservice	
repairedbl	repairedblservice, repaireddataservice, vo, userbl, reservebl
repaireddataservice	Java RMI, po
repaireddata	Java RMI, po, databaseutility
repairupdateui	repairupdateblservice, 界面类库包
repairupdateblservice	
repairupdatebl	repairupdateblservice, repairupdatedataservice, po, reservebl, unsubscribebl
repairupdatedataservice	Java RMI, po
repairupdatedata	RMI, po, databaseutility
vo	
po	
utilitybl	
界面类库包	
Java RMI	
databaseutility	JDBC

同济大学学生宿舍管理系统客户端开发包和服务端开发包如下图所示：



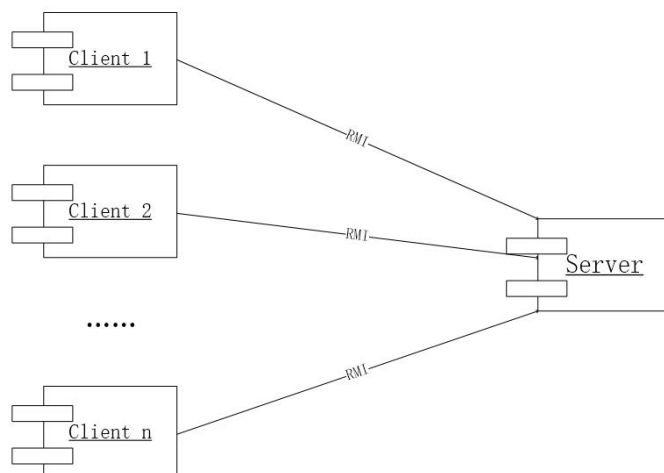
同济大学学生宿舍管理系统客户端开发包图



同济大学学生宿舍管理系统服务器端开发包图

3.1.2.2 运行时进程

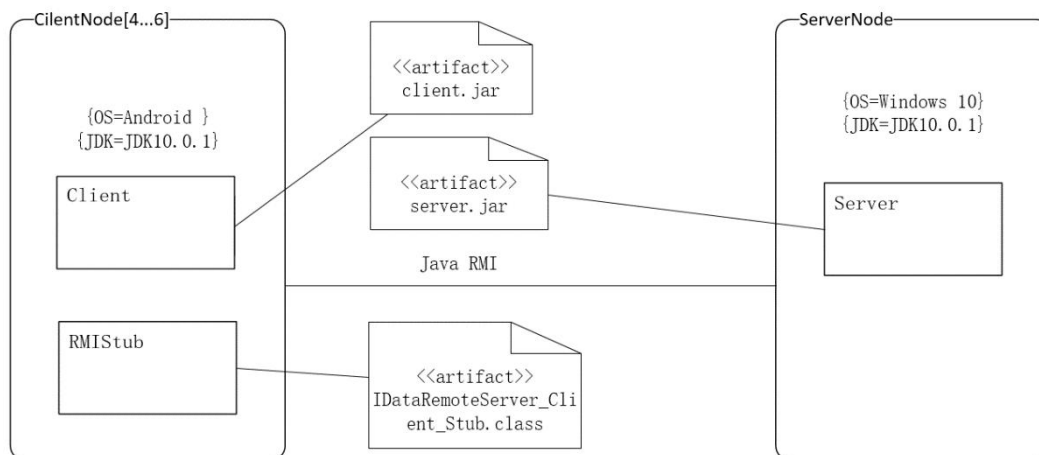
在同济大学学生宿舍管理系统中，会有多个客户端进程和一个服务器端进程，其进程图如下图所示，结合部署图，客户端进程是在客户端机器上运行，服务器端进程在服务器端机器上运行。



进程图

3.1.2.3 物理部署

同济大学学生宿舍管理系统中客户端构件是放在客户端机器上, 服务器端构件是放在服务器端机器上。在客户端节点上, 还要部署 RMISub 部件, 由于 Java RMI 构件属于 JDK10.0.1 的一部分。所以, 在系统 JDK 环境已经设置好了的情况下, 不需要再独立部署。部署图如下图所示。



部署图

3.2 用户层面模块分解

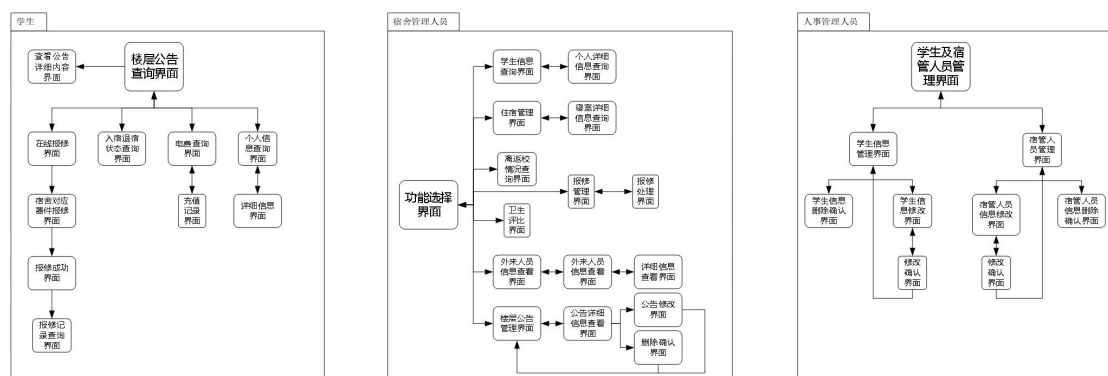
根据需求, 本系统主要存在以下用户界面, 同样面向三个不同的对象:

学生: 楼层公告查询界面、在线报修界面、电费查询界面、个人信息查询界面、入宿退宿状态查询界面、查看公告详细内容界面、宿舍对应器件报修界面、报修成功界面、报修记录查询界面、充值记录界面、详细信息界面;

宿舍管理员: 功能选择界面、学生信息查询界面、住宿管理界面、离返校情况查询界面、卫生评比界面、报修信息处理界面、外来人员信息查看界面、楼层公告管理界面、外来人员信息登记界面、详细信息查看界面、公告详细信息查询界面、公告修改界面、删除确认界面、报修处理界面、个人详细信息查询界面、寝室详细信息查询界面;

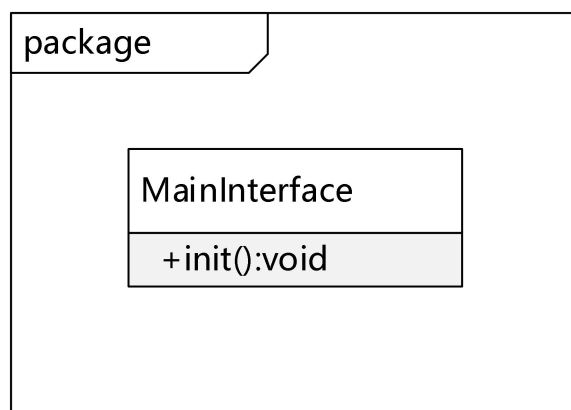
人事管理人员: 学生信息管理界面以及宿管人员管理界面、学生信息删除确认界面、学生信息管理界面、学生信息修改界面、宿管人员信息删除确认界面、宿管人员信息管理界面、宿管人员信息修改界面、修改确认界面;

界面跳转如下图所示 (看不清请对页面进行放大):



用户界面跳转

服务器端和客户端的用户界面设计接口保持一致，只是具体页面不一样。用户界面类如下图所示：



用户界面类

3.2.1 用户界面层模块的职责

模块	职责
MainInterface	界面 Interface，负责界面的显示和界面的跳转

3.2.2 用户界面层模块的接口规范

MainInterface	语法	init(args:String[])
	前置条件	无
	后置条件	显示 Interface 以及 LoginPanel
需要的服务（需接口）		
服务名		服务
businesslogicservice.LoginBLservice		登录界面的业务逻辑接口
businesslogicservice.*BLService		每个界面都有一个相应的业务逻辑接口

3.2.3 用户界面模块设计原理

用户界面利用 Java 的 Swing 和 SWT 库来实现，在 xml 文件进行编程和修改。

3.3 业务逻辑层模块分解

业务逻辑层包括多个针对界面的业务逻辑处理对象。例如，User 对象负责处理登录界面的业务逻辑；Reserve 对象负责处理预定界面的业务逻辑。



3.3.1. 业务逻辑层模块的职责

业务逻辑层模块的职责如下表所示。

模块	职责
Userbl	负责实现身份认证和登录界面需要的服务
Repairbl	负责实现报修界面需要的服务
Checkbl	负责实现报修确认界面需要的服务
Repairedbl	负责实现身份认证和登录界面需要的服务
Repiarupdatebl	负责实现报修界面需要的服务

3.3.2. 业务逻辑层的接口规范

Userbl 模块接口规范

提供的服务（供接口）		
User.login	语法	Public ResultMessage login(long id,String password)
	前置条件	Password 符合输入规则
	后置条件	查找是否存在相应的 user，根据输入的 password 返回登录验证的结果
需要的服务（需接口）		
服务名	服务	
DataBaseFacory.getUserDataBase	得到 User 数据库的服务的引用	
UserDataService.insert(UserPo po)	在数据库中插入 UserPo 对象	
UserDataService.delete(UserPo po)	在数据库中删除 UserPo 对象	

Repairbl 模块接口规范

提供的服务（供接口）		
Repair.addMember	语法	Public ResultMessage addMember(long id)
	前置条件	启动一个报修回合
	后置条件	在一个报修回合中，增加报修的用户信息
Repair.addCommodity	语法	Public ResultMessage addCommodity(long id,long quantity)
	前置条件	启动一个报修回合
	后置条件	在一个报修回合中，增加报修的宿舍信息
	语法	Public ResultMessage addMessage(long

Repair.addMessage		id,long quantity)
	前置条件	已添加报修的详细信息
	后置条件	返回报修是否成功
Repair.endRepair	语法	Public ResultMessage endRepair (long id,long quantity)
	前置条件	已报修
	后置条件	结束此次报修回合，持久化更新涉及的领域对象的数据
需要的服务（需接口）		
服务名		服务
RepairDataservice.find(long id)		根据 id 进行查找单一持久化对象
RepairDataservice.insert(RepairPo po)		插入单一持久化对象
RepairDataservice.delete(RepairPo po)		删除单一持久化对象
RepairDataservice.update(RepairPo po)		更新单一持久化对象
RepairDataservice.init(RepairPo po)		初始化单一持久化对象
RepairDataservice.finish(RepairPo po)		结束持久化数据库的使用

Checkbl 模块接口规范

提供的服务（供接口）		
Repair.check	语法	Public ResultMessage check(long id)
	前置条件	当前 id 存在
	后置条件	查找当前 id 对应的报修信息
需要的服务（需接口）		
服务名		服务
DataBaseFacory.getRepairDataBase		得到 RepairData 数据库的服务的引用

Repairedbl 模块接口规范

提供的服务（供接口）		
Repaired.addMember	语法	Public ResultMessage addMember(long id)
	前置条件	启动一个报修完成回合
	后置条件	在一个报修完成回合中，增加报修完成的用户信息
Repaired.addCommodity	语法	Public ResultMessage addCommedity(long id,long quantity)
	前置条件	启动一个报修完成回合
	后置条件	在一个报修完成回合中，增加报修完成的宿舍信息
Repaired.end	语法	Public ResultMessage end Unsubscribe (long id,long quantity)
	前置条件	报修已完成
	后置条件	结束此次报修完成回合，持久化更新涉及的领域对象的数据
需要的服务（需接口）		

服务名	服务
RepairDataservice.find(long id)	根据 id 进行查找单一持久化对象
RepairDataservice.insert(ReservePo po)	插入单一持久化对象
RepairDataservice.delete(ReservePo po)	删除单一持久化对象
RepairDataservice.update(ReservePo po)	更新单一持久化对象
RepairDataservice.init(ReservePo po)	初始化单一持久化对象
RepairDataservice.finish(ReservePo po)	结束持久化数据库的使用

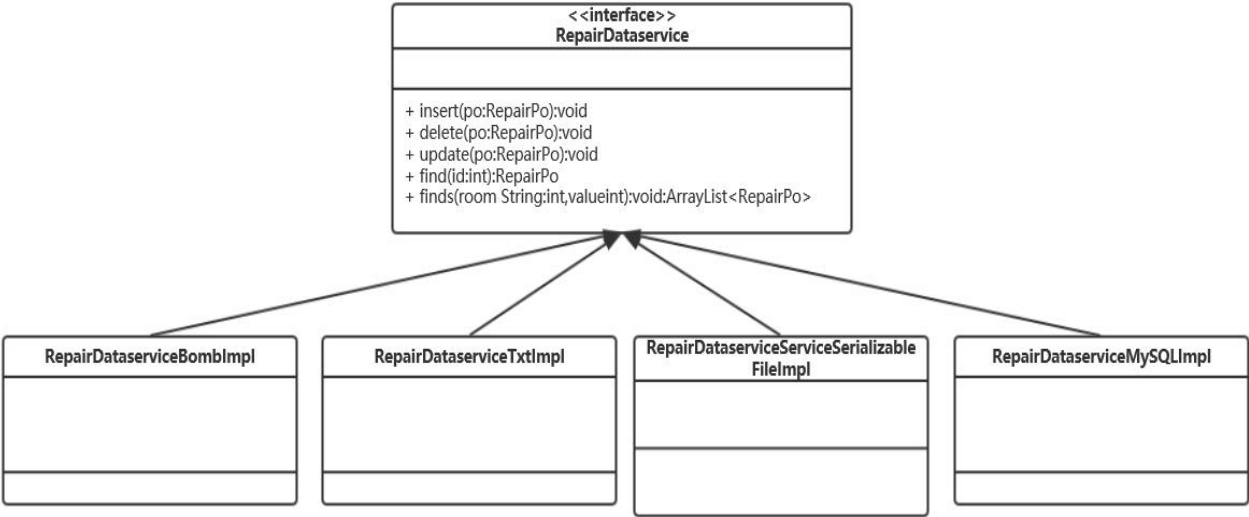
Repiarupdatebl 模块接口规范

提供的服务（供接口）		
Repiarupdate.addMember	语法	Public ResultMessage addMember(long id)
	前置条件	启动一个报修信息更新回合
	后置条件	在一个报修信息回合中，增加报修信息的用户信息
Repiarupdate.addCommodity	语法	Public ResultMessage addCommodity(long id,long quantity)
	前置条件	启动一个报修信息回合
	后置条件	在一个报修信息回合中，增加报修信息的场地信息
Repiarupdate.addMessage	语法	Public ResultMessage addMessage(long id,long quantity)
	前置条件	已添加用户信息和场地信息
	后置条件	添加一个报修更新信息
Repiarupdate.end Repiarupdate	语法	Public ResultMessage endRepairupdate(long id,long quantity)
	前置条件	信息更新完成
	后置条件	结束此次报修信息更新回合，持久化更新涉及的领域对象的数据
需要的服务（需接口）		
服务名	服务	
RepairDataservice.find(long id)	根据 id 进行查找单一持久化对象	
RepairDataservice.insert(ReservePo po)	插入单一持久化对象	
RepairDataservice.delete(ReservePo po)	删除单一持久化对象	
RepairDataservice.update(ReservePo po)	更新单一持久化对象	
RepairDataservice.init(ReservePo po)	初始化单一持久化对象	
RepairDataservice.finish(ReservePo po)	结束持久化数据库的使用	

3.4 数据层模块分解

数据层主要给业务逻辑层提供数据访问，包含对持久化与阶段性数据的增、删、改、查。Announce 业务逻辑需要的服务由 AnnounceDataservice 接口提

供。StudentSearch 业务逻辑需要的服务由 StudentSearchDataservice 接口提供。Manager 业务逻辑需要的服务由 ManagerDataservice 接口提供。Repair 业务逻辑需要的服务由 RepairDataservice 接口提供。由于四类业务均以数据库的增删改查为主，所以以 Repair 业务为例阐述，其余三类服务不再赘述。由于持久化数据和阶段性数据保存可能存在多种形式：TXT 文件，序列化文件，数据库等类型，所示抽象化了数据服务。Repair 业务数据层模块的描述具体如下所示：



3.4.1 数据层模块的职责

模块	职责
RepairDataservice	持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
RepairDataserviceTxtImpl	基于 TXT 文件的持久化数据库的接入，提供集体载入、集体保存、增、删、改、查服务
RepairDataserviceMySQLImpl	基于 MySQL 本地数据库的持久化数据库的接入，提供集体载入、集体保存、增、删、改、查服务
RepairDataserviceBomImpl	基于在线云服务器数据库的持久化数据库的接入，提供集体载入、集体保存、增、删、改、查服务
RepairDataserviceServiceSerializableFileImpl	基于序列化文件的持久化数据库的接入，提供集体载入、集体保存、增、删、改、查服务

3.4.2 数据层模块的接口规范

提供的服务（供接口）		
RepairDataservice.find	语法	Public void RepairPO find(int id) throws RemoteException
	前置条件	无
	后置条件	以用户的用户编码 ID/用户学工号

		StudentID 作为搜索依据进行查找并返回查找结果并返回结果 RepairPo
RepairDataservice.insert	语法	Public void RepairPO insert(int id) throws RemoteException
	前置条件	相同的 ID 的 po 在 Mapper 中不存在
	后置条件	在数据库中增加一个 po 记录
RepairDataservice.delete	语法	Public void RepairPO delete(int id) throws RemoteException
	前置条件	在当前数据库中不存在与删除 ID 相同的 ID 号的 po
	后置条件	删除当前 ID 的一个 po
RepairDataservice.update	语法	Public void RepairPO update(int id) throws RemoteException
	前置条件	在当前数据库中不存在与更新 ID 相同的 ID 号的 po
	后置条件	按照用户 ID 在数据库中进行更新当前 ID 的一个 po
RepairDataservice.init	语法	Public void RepairPO init() throws RemoteException
	前置条件	无
	后置条件	初始化持久化数据库
RepairDataservice.finish	语法	Public void RepairPO finish() throws RemoteException
	前置条件	无
	后置条件	结束对持久化数据库的使用以及访问申请

3.5 界面模块设计

界面跳转设计：

学生用户：主界面为楼层公告查询界面，公告界面可通过点击对应按钮切换到在线报修、电费查询、个人信息查询界面或是入宿退宿状态查询界面。其中楼层公告查询界面可点击对应公告跳转到查看公告详细内容界面，在线报修界面点击对应物品进入宿舍对应器件报修界面，填写完成并点击提交按钮跳转到报修成功界面，点击返回键返回报修记录查询界面。跳转到电费查询界面后，点击详情按钮跳转到充值记录界面查看电费充值记录，点击返回键返回电费查询界面，所有由公告界面进入的界面均可通过返回键返回到公告查询界面，公告界面跳转到个人信息界面后，可点击对应信息项跳转到详细信息界面查看对应项信息的详细情况。

宿舍管理员用户：主界面为功能选择界面，通过选择对应按钮分别跳转到学生信息查询界面、住宿管理界面、离返校情况查询界面、卫生评比界面、报修信息处理界面、外来人员信息登记界面以及楼层公告管理界面。外来人员登记界面为外来人员信息查看界面，点击“新建”跳转到外来人员信息登记界面，点击对应人员项跳转到该人员详细信息查看界面。公告管理界

面点击对应公告跳转到公告详细信息查看界面，点击删除跳转到删除确认界面，确认后跳转回公告管理界面，点击修改跳转到公告修改界面，点击确认修改跳转回公告管理界面。报修管理界面点击对应报修申请项跳转到报修处理界面。学生信息查询界面点击对应学生跳转到该用户详细信息查询界面，住宿管理界面与学生信息查询界面跳转类似。

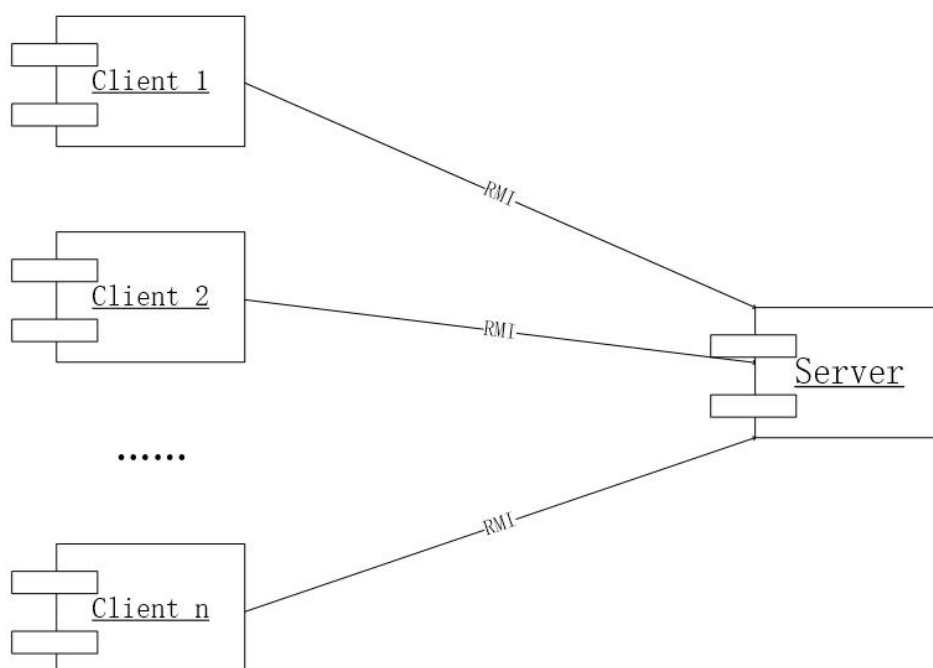
人事管理人员用户：主界面为学生信息管理界面以及宿管人员管理界面，学生信息管理界面通过点击删除按钮跳转到学生信息删除确认界面，点击确认按钮跳转回学生信息管理界面，在学生信息管理界面点击修改按钮跳转到学生信息修改界面，点击确认跳转到修改确认界面，确认后跳转回学生信息管理界面。宿管人员管理界面通过点击删除按钮跳转到宿管人员信息删除确认界面，点击确认按钮跳转回宿管人员信息管理界面，在宿管人员管理界面点击修改按钮跳转到宿管人员信息修改界面，点击确认跳转到修改确认界面，确认后跳转回宿管人员管理界面。

界面风格：

界面采用的风格为简约风格，以单一色调为主，背景设置为动画背景。界面直观对用户透明，用户接触该软件之后对界面上的功能一目了然，无需培训便可使用该软件。界面设计采用以线性布局 LinearLayout 为主，相对布局 RelativeLayout 为辅的设计方法。字体为灰底黑字，部分为灰底红字，字体大小为宋体，大小为 14sp，部分显示带边框，14 对齐方式为居中对齐。

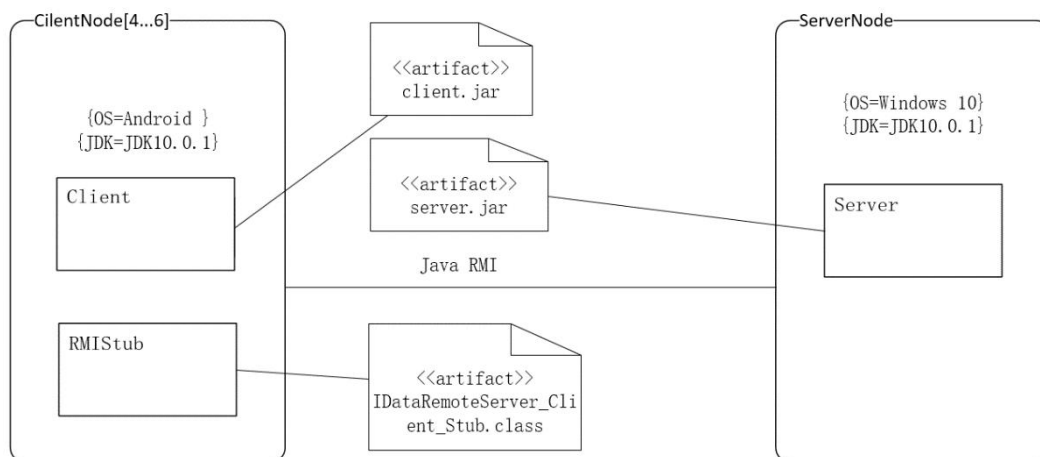
3.6 运行时进程

在同济大学宿舍管理系统中，会有多个客户端进程和一个服务器端进程，其进程图如下图所示，结合部署图，客户端进程是在客户端机器上运行，服务器端进程在服务器端机器上运行。



3.7 物理部署

校内运动场地管理系统中客户端构件是放在客户端机器上,服务器端构件是放在服务器端机器上。在客户端节点上,还要部署 RMISub 部件,由于 Java RMI 构件属于 JDK10.0.1 的一部分。所以,在系统 JDK 环境已经设置好了的情况下,不需要再独立部署。部署图如下图所示。



4 模块之间的映射

4.1 调用关系映射

学生用户:

- (1) 主模块调用基础信息查询模块、在线报修模块、电费查询模块、入宿退宿状态查询模块、楼层公告查看模块;
- (2) 报修模块调用报修信息填写模块;

宿舍管理员用户:

- (1) 主模块调用学生信息查询模块、住宿管理模块、离返校情况查询模块、卫生评比模块、报修处理模块、外来人员信息登记模块、楼层公告管理模块;
- (2) 卫生评比模块调用卫生评比信息录入模块
- (3) 外来人员信息登记模块调用信息登记模块

人事管理人员用户:

- (1) 主模块调用学生信息管理模块、宿管人员管理模块;
- (2) 学生信息管理模块调用学生信息查询模块、学生信息修改模块、学生信息删除模块;

4.2 数据模型信息视角

4.2.1 数据持久化对象

系统的 po 类就是对应的相关的实体类,在此只作简单介绍

*StudentPo 类包含学生用户编号、学工号、登录密码、姓名,性别、手机号码、邮箱、身份类型、所属部门、楼字号、宿舍号。

*ManagerPo 类包含宿管人员编号、登录密码、性别、手机号码、身份类型、所属部门、楼字号
*TopmanagerPo 包含人事管理人员的编号、登录密码
*RepairPo 包含报修时间、报修物品、报修宿舍楼宇、报修宿舍号、处理状态
*AnnouncePo 包含公告发布时间、公告发布人员编号、公告详情
其中以持久化对象 StudentPo 的定义为例，其定义如下、其余 Po 类定义与 StudentPo 类定义类似：

```
public class StudentPo implements Serializable{
    int id;
    String stuID;
    String password;
    String Username;
    String gender;
    String phone;
    String mail;
    String identity;
    String department;
    String buildingID;
    String roomID;
    StudentRole role;
    public StudentPo(int i,String a1,String a2,String a3,String a4,String a5,String a6,String
a7,String a8,String a9,String a10,StudentRole role1) {
        id=i;
        stuID=a1;
        password=a2;
        Username=a3;
        gender=a4;
        phone=a5;
        mail=a6;
        identity=a7;
        department=a8;
        buildingID=a9;
        roomID=a10;
        role=role1;
    }

    public int getId() {
        return id;
    }

    public String getBuildingID() {
        return buildingID;
    }
}
```

```
public String getDepartment() {  
    return department;  
}  
  
public String getGender() {  
    return gender;  
}  
  
public String getIdentity() {  
    return identity;  
}  
  
public String getMail() {  
    return mail;  
}  
  
public String getPassword() {  
    return password;  
}  
  
public String getPhone() {  
    return phone;  
}  
  
public String getRoomID() {  
    return roomID;  
}  
  
public String getStuID() {  
    return stuID;  
}  
public String getUsername() {  
    return Username;  
}  
public StudentRole getRole() {  
    return role;  
}  
}
```

4.2.2Txt 持久化格式

Txt 数据保持格式以 RepairPo.txt 为例，每行对应报修编号，报修时间，报修楼宇号，报修宿舍号、报修状态，中间以冒号隔开如下所示：

001:2018-08-07: 嘉定 7 号楼: 402: 已处理
002:2017-11-07: 嘉定 8 号楼: 312: 已处理
003:2018-06-13: 四平西南一: 302: 未处理
004:2018-09-07: 嘉定 7 号楼: 502: 未处理
005:2018-08-11: 嘉定 7 号楼: 422: 未处理
.....

4.2.3 云服务器数据库表

数据库中包含 Student 表, Manager 表, Topmanager 表, Repair 表, Announce 表, Repairhistory 表, Cleanresult 表, Outpersoninfo 表, Inoroutschool 表

5 系统体系结构设计

5.1. 用户界面的特殊细节

- 1) 用户登录界面时, 若 login 返回的结果为 NoUser, 发出错误提示, 并询问是否需要注册当前账号。
- 2) 用户登录界面时, 若 login 返回的结果为 GetUserDataFail, 输出提示“当前与服务器连接失败, 请稍后再试”。
- 3) 学生用户添加报修信息时, 若 repair.addMember 返回为 EXISTED, 表明同一场地已存在报修信息, 将两次的报修信息合并处理。
- 4) 宿管管理报修信息时, 将信息更新结果反馈到报修的学生端。