软件(结构)设计说明(SAD)

说明：

1.《软件(结构)设计说明》(SDD)描述了计算机软件配置项(CSCI的设计。它描述了CSCI级设计决策、CSCI体系结构设计(概要设计)和实现该软件所需的详细设计。SDD可用接口设计说明IDD和数据库(顶层)设计说明DBDD加以补充。

2.SDD连同相关的IDD和DBDD是实现该软件的基础。向需方提供了设计的可视性，为软件支持提供了所需要的信息。

3.IDD和DBDD是否单独成册抑或与SDD合为一份资料视情况繁简而定。

目录

[软件(结构)设计说明(SAD) 1](#_Toc134814596)

[1引言 3](#_Toc134814597)

[1.1标识 3](#_Toc134814598)

[1.2系统概述 3](#_Toc134814599)

[1.3文档概述 3](#_Toc134814600)

[1.4基线 4](#_Toc134814601)

[2引用文件 4](#_Toc134814602)

[3 CSCI级设计决策 4](#_Toc134814603)

[4 CSCI体系结构设计 8](#_Toc134814604)

[4.1体系结构 8](#_Toc134814605)

[4.1.1程序(模块)划分 8](#_Toc134814606)

[4.1.2程序(模块)层次结构关系 9](#_Toc134814607)

[4.2全局数据结构说明 11](#_Toc134814608)

[4.2.1常量 11](#_Toc134814609)

[4.2.2变量 12](#_Toc134814610)

[4.2.3数据结构 12](#_Toc134814611)

[4.3 CSCI部件 13](#_Toc134814612)

[4.4执行概念 16](#_Toc134814613)

[4.5接口设计 17](#_Toc134814614)

[4.5.1外部接口 17](#_Toc134814615)

[4.5.2内部接口 20](#_Toc134814616)

[5 CSCI详细设计 23](#_Toc134814617)

[6需求的可追踪性 25](#_Toc134814618)

[7注解 25](#_Toc134814619)

[附录 25](#_Toc134814620)

# 1引言

## 1.1标识

本文档适用的系统：校园卡管理系统

软件的完整标识：校园卡管理系统V1.0-CN( FULL VERSION -α)

1）标题：校园卡管理系统

2）缩略词语：校园卡管理

3）版本号：1.0.0

4）发行号：1.0.0a

## 1.2系统概述

（1）本文档适用的项目：校园卡管理系统

（2）软件的用途：通过校园卡管理系统，对全校师生的校园卡进行统一规范的管理，既方便校园卡的日常功能使用，同时对产生的各类数据进行集合，及时获取变更信息，提高校园管理工作的效率。

（3）项目开发、运行和维护的历史：到目前为止，项目开发进度为注册及登陆后用户显示的功能页面，相关功能实现仍在持续开发中。

（4）项目开发方：@该吃饺子了 5人软件开发小组

（5）预计项目用户：校园全体师生

（6）当前运行现场：开发小组个人笔记本电脑

（7）计划运行现场：有专门的服务器，让所有用户均能使用私人计算机访问该系统并使用

（8）其他相关文档：可行性分析报告、软件测试说明、软件测试报告、项目开发总结报告等。

## 1.3文档概述

（1）文档用途及内容：本文档是校园卡管理系统的设计说明，基于先前完成的系统需求分析，描述系统的CSCI设计决策、体系结构设计以及详细设计等。

（2）使用保密性和私密性要求：该软件结构设计说明文档SAD仅限于阅读评价，不可随意将报告内容泄露与他方，否则报告所属团队有权追究其责任。

## 1.4基线

该系统以主页面为主，对各个功能模块进行定义、开发与延伸。

# 2引用文件

[1]BENSHNEIDERMAN, CATHERINEPLAISANT. 用户界面设计:有效的人机交互策略[M]. 电子工业出版社, 2006.

[2]Garlan,David.“Software architecture:a roadmap.”International Conference on Software Engineering (2000).

[3]David Garlan and Mary Shaw,School of Computer Science Carnegie Mellon Universit ."An Introduction to Software Architectur"(January 1994)

[4]百度文库.梅澎《关于软件体系结构的作用和意义的理解》.

[5]陈澎,李江.软件体系结构的分析[J].计算机工程,2005(20):82-84.

[6]毛斐巧,齐德昱.软件体系结构风格研究现状及存在的问题[J].计算机应用研究,2008(08):2270-2273.

# 3 CSCI级设计决策

（1）根据系统功能的不同，目前可以将系统划分为以下功能模块：

用户：

①基本信息模块：查询/修改用户个人基本信息；

②账单信息模块：用户可选择查询个人账户的消费记录/充值记录；支持查询末次充值记录。

③银行卡绑定模块：用户进入该模块，如未绑定银行卡，可输入卡号进行绑定，以便校园卡充值；如已绑定银行卡，可以选择解绑；

④银行卡充值：如用户账号绑定了银行卡，则可以通过绑定的银行卡给校园卡充值；

⑤线上消费模块：又细分为电费缴纳、水费缴纳、线上订餐三个子模块，可使用校园卡余额进行消费；

⑥挂失模块：首先确认校园卡状态处于正常状态，然后用户输入正确的校园卡密码即可进行挂失，将校园卡状态设置为挂失态；用户校园卡状态若为非正常态（挂失状态等）则不能进行校园卡挂失。

⑦解挂模块：首先确认校园卡状态处于挂失状态，然后用户输入正确的校园卡密码即可进行解挂，将校园卡状态设置为正常态；用户校园卡状态若为非挂失状态则不能进行校园卡解挂。

⑧帮助中心：以文本形式为用户提供一些校园卡系统使用帮助；

⑨账号注册模块：用户通过输入学号与密码等基本信息进行注册，如用户不存在且基本信息合法则注册成功。

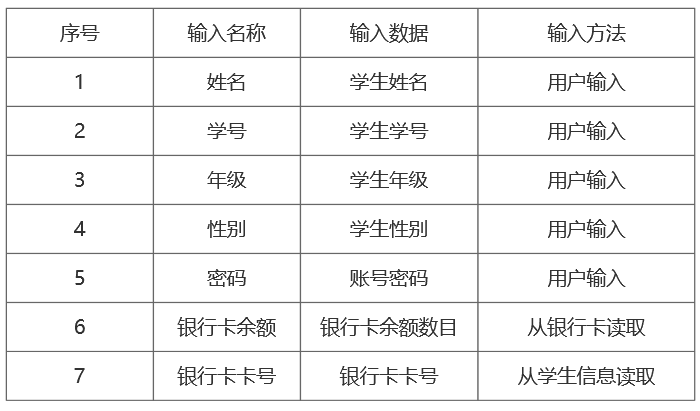
⑩账号登陆模块：用户通过注册得到的账号和密码进行登录，如若账号和密码正确则登陆成功进入用户系统主页面，可选择功能进行使用。

管理员：

①用户信息管理模块：管理员可查询/增加/删除/修改用户信息；

②帐单信息查询模块：管理员可查看所有用户的消费记录/充值记录；可根据用户校园卡卡号等查询相关用户的消费记录/充值记录；

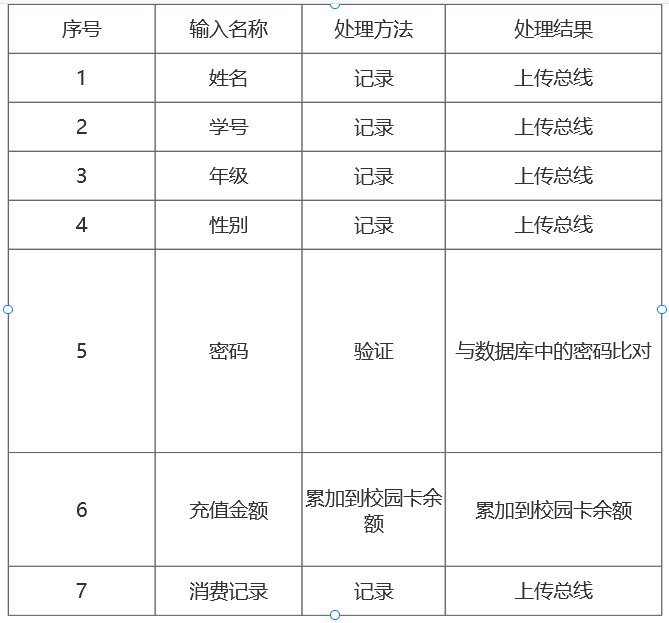
（2）软件的输入设计决策见下表



软件的输出设计决策见下表：



（3）CSCI对每个输入的处理行为设计决策：



（4）响应时间：该系统的操作响应时间控制在1s之内。

（5）数据库设计（目前来说）：

①用户信息表，包含学号、姓名、性别、学院、年级、密码、校园卡卡号、重置密令。



②充值记录，包括充值编号、校园卡卡号、充值时间、银行卡卡号、充值金额、校园卡余额。



③校园卡信息表，包含校园卡卡号、学号、校园卡余额、状态（是否挂失）、密码。



④消费信息表，包含消费信息编号、校园卡卡号、消费时间、消费产品类型、消费价格、消费商家、校园卡余额。



⑤银行卡信息表，包含学号、银行卡卡号、银行卡密码。



⑥餐食信息表，包含餐食编号、餐食名称、价格、商家



而对于其余数据库和数据文件需求，需要随着软件设计开发过程决定。

（6）安全性：系统会按时对数据库的数据进行人工备份，从而保证系统的安全性，防止数据丢失；设置了挂失操作作为用户校园卡丢失时的一种暂时性冻结账户的安全保护机制等。

（7）保密性与私密性：

①根据本系统的保密性和私密性需求，用户和管理员拥有的权限是不相同的。对于管理员，可对用户部分信息进行管理，包括增加、删除、修改等，同时管理员会保管好账号密码，不透露给他人使用，谨守道德规范，规范操作系统，不侵犯用户信息安全。而用户只能登录自身账号，对自己的信息进行管理以及使用该系统的功能。同时，用户可以进行修改登陆密码、对校园卡进行挂失、根据密保找回密码等操作，保障系统使用的安全性。

②校园卡管理系统的核心是一个数据库，存储了用户的基本信息和交易记录，以及各种权限和规则。校园卡管理系统通过加密算法和数字签名来保证数据的安全性，防止数据被篡改或泄露。校园卡管理系统通过身份验证和权限控制来保证数据的保密性和私密性，只允许合法的用户访问和操作数据，防止数据被滥用或盗用。

（8）灵活性：系统开发维护流程便捷，可较好地适应用户需求变化。

（9）可用性：本系统界面简洁、操作简单，同时用户可通过系统中的帮助中心查看一些系统使用相关帮助提示。

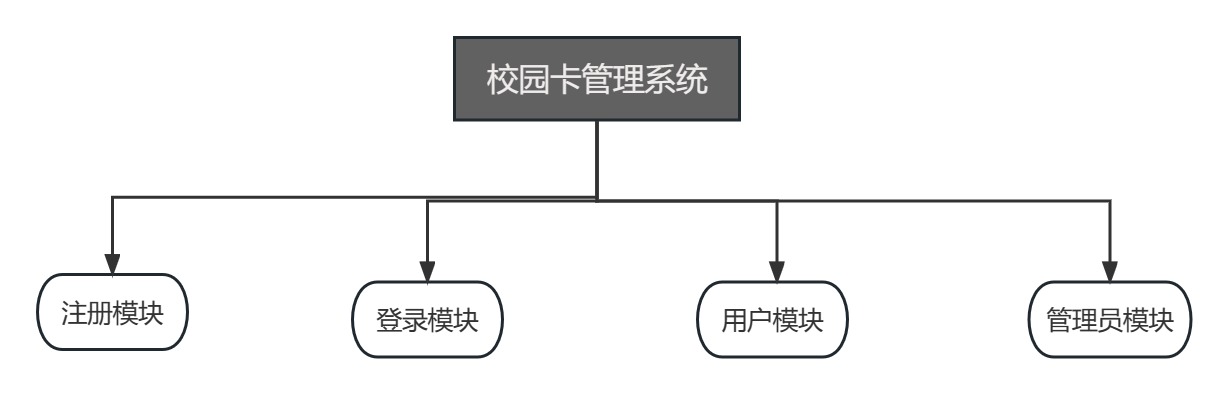
（10）可维护性：维护管理员会密切关注用户使用过程，及时修复漏洞；日常保持系统正常运行；安排专门的时间检查功能使用情况，排查异常问题；根据用户使用需求开发新的功能等。

# 4 CSCI体系结构设计

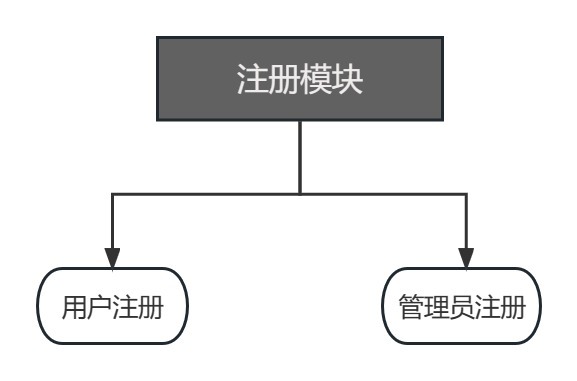
## 4.1体系结构

### 4.1.1程序(模块)划分

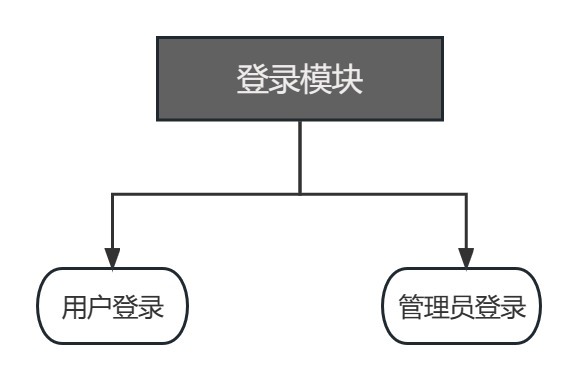
校园卡管理系统一共分为四大模块，分别为登录模块、注册模块、用户模块、系统管理员模块：如下图所示：



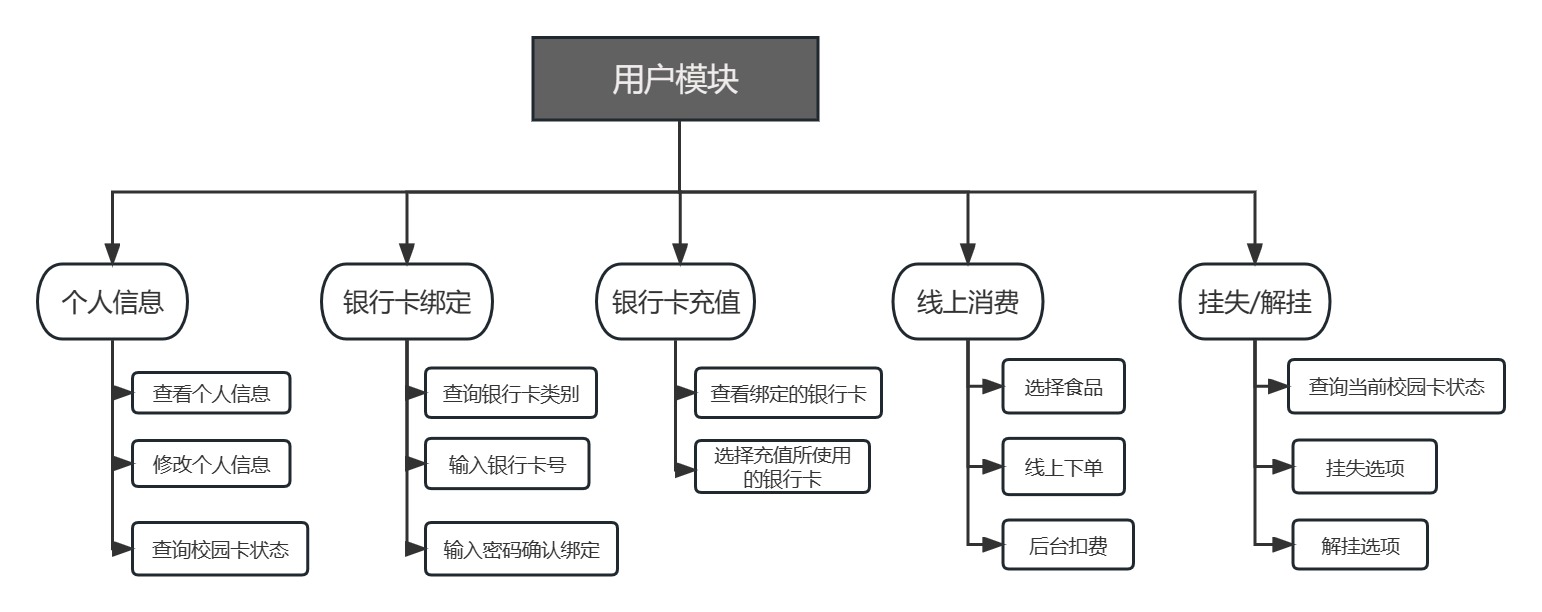
注册模块：



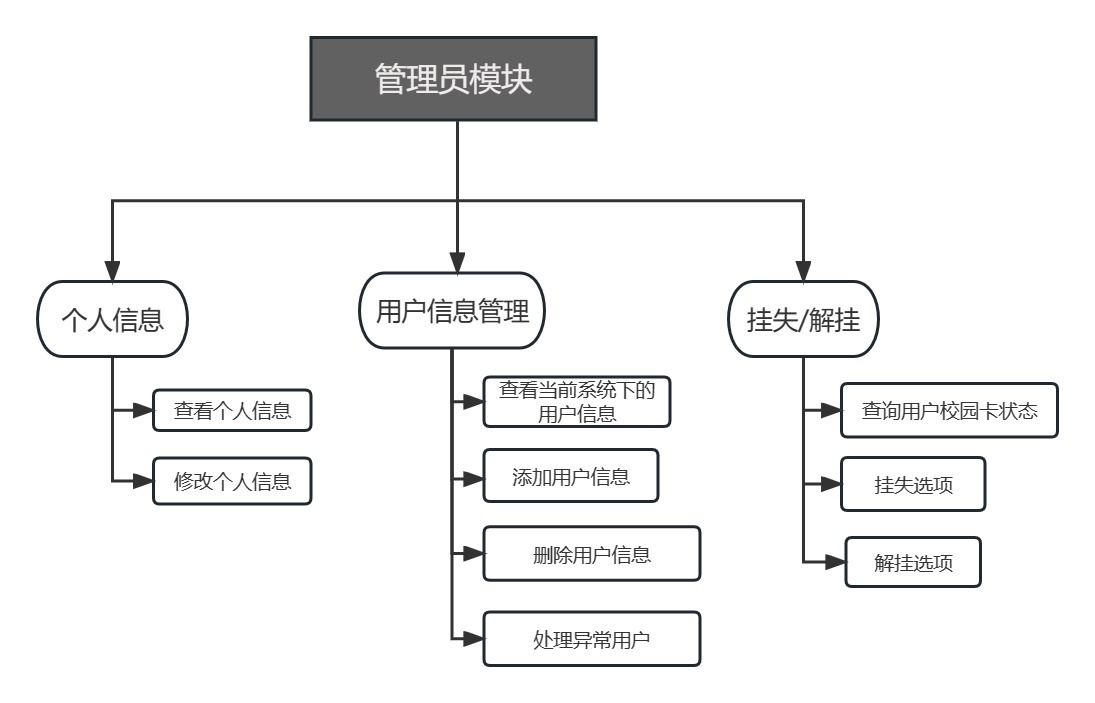
登录模块：



用户模块：



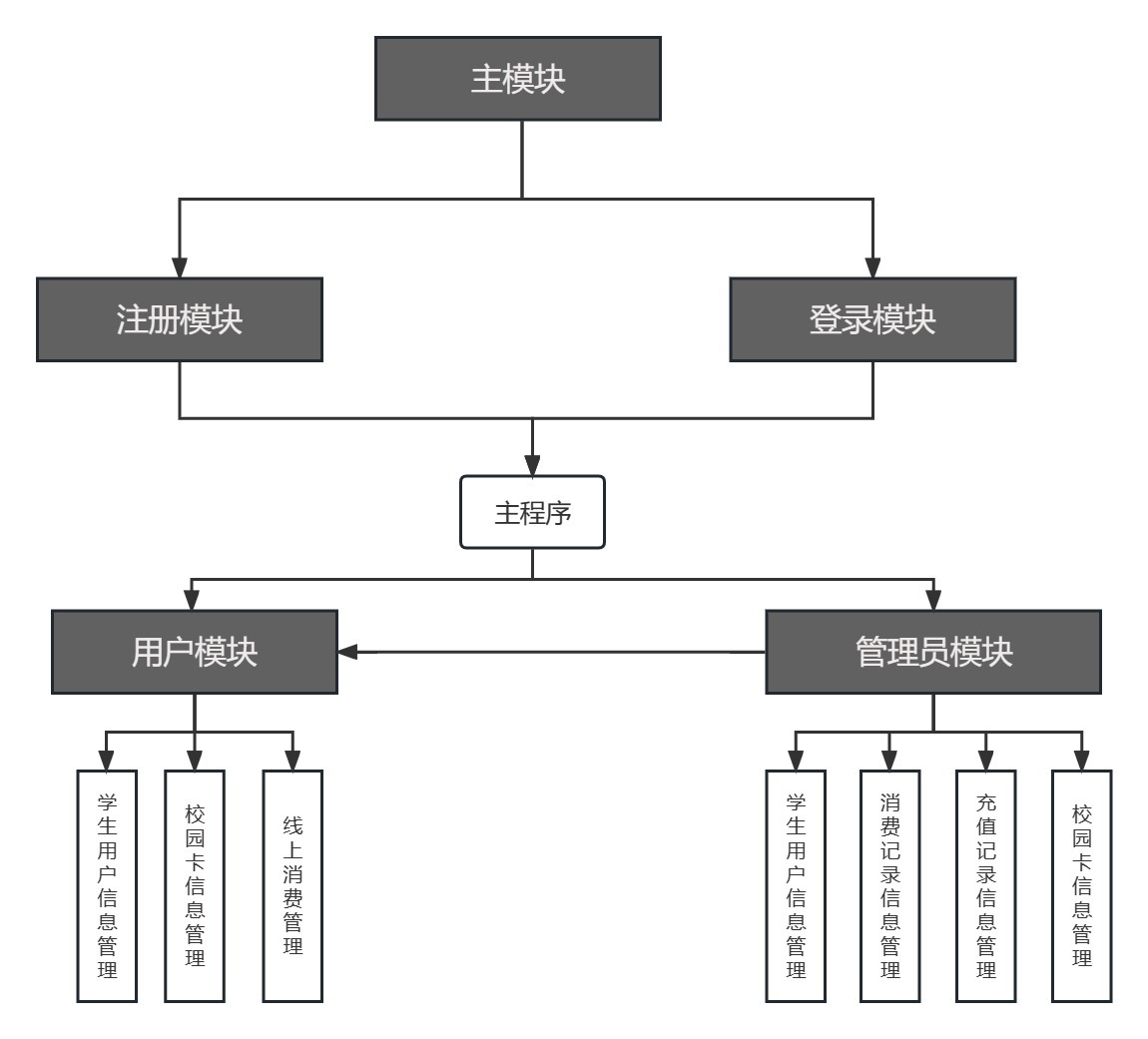
管理员模块：



### 4.1.2程序(模块)层次结构关系

关于下面关系图的说明：

校园卡管理系统总结构可以分为四个层次。最上层是整个系统的总体控制模块，也是整个实现过程中的主模块，第二层是进入系统所必须的模块操作，如果已经在系统中存在账号，那么进入登录模块，否则不存在账号信息，需要先进入注册模块注册账号信息，接下来进入登录模块，这是整个系统的安全性验证，判断是正确账号信息登录系统。在完成前两层结构的操作后，进入第三层，系统会自动根据登录时输入的账号信息判断是该层结构中的哪一个模块，若账号类别为用户，那么系统会自动进入用户模块，在该模块下，用户可以进行本校园卡管理系统所允许的权限操作，在账号操作不处于异常情况下，系统会根据用户的实时操作进行客户端与服务器数据库间的通信从而及时更新数据信息；若账号类别为管理员，那么系统会自动进入管理员模块，在该模块下，管理员拥有对管理系统较高的操作权限，管理员可以进行多种后台操作，在确保不侵犯涉及所操作用户的隐私信息的条件下，管理员拥有对系统内用户的移除修改等多种管理操作，同时管理员也可以查看到系统内是否存在行为异常的用户，若发现存在该类用户，那么管理员有权限对该用户进行处理，该用户后续可以联系后台服务工作人员进行账号的异常消除，管理员可以查看到用户的消费和充值信息，但是在该操作下，管理员不具备修改删除添加消费和充值记录信息的权限，只是可以查看用户的记录，由此体现系统的安全性大大提高，管理员还可以查看系统内校园卡状态信息，并对校园卡进行适当权限内的管理操作。以上为校园卡管理系统的层次结构分布情况与调用关系介绍。



以下以表格形式分别表示用户和管理员中模块的层次结构关系：

用户：

|  |  |  |
| --- | --- | --- |
| 模块 | 调用模块 | 调用的数据库表格 |
| 基本信息 | 修改用户信息 | 用户信息表students |
| 修改密码 | 用户信息表students、校园卡信息表card |
| 账单信息 | 消费记录 | 消费信息表bill |
| 充值记录 | 充值记录表chargerecord |
| 银行卡关联 | / | 银行卡记录表bank\_card |
| 银行卡充值 | / | 银行卡记录表bank\_card、充值记录表 |
| 线上消费 | 线上订餐 | 餐食表food、消费信息表bill、校园卡信息表card |
| 电费缴纳 | 消费信息表bill、校园卡信息表card |
| 水费缴纳 | 消费信息表bill、校园卡信息表card |
| 挂失 | / | 校园卡信息表card |
| 解挂 | / | 校园卡信息表card |
| 帮助中心 | / | / |

管理员：

|  |  |  |
| --- | --- | --- |
| 模块 | 调用模块 | 调用的数据库表格 |
| 用户账户管理 | 查询用户信息 | 用户信息表students |
| 添加用户信息 | 用户信息表students |
| 修改用户信息 | 用户信息表students |
| 删除用户信息 | 用户信息表students |
| 用户账单信息管理 | 查看用户消费信息 | 消费信息表bill |
| 查看用户充值信息 | 充值记录表chargerecord |

## 4.2全局数据结构说明

### 4.2.1常量

在校园卡管理系统中，重点服务对象是在校经过认证的持有校园卡的用户人群，学号和姓名是区别用户之间信息的主要数据常量，此外由学号和姓名等常量信息还涉及到其他数据文件中的常量数据，下表为系统中存在的常量数据说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 常量名称 | 所在目录 | 初值设置 | 功能说明 | 具体常量说明 |
| 学号 | students | 在注册时输入设置 | 系统中的身份认证 | 区别不同用户信息 |
| 姓名 | students | 在注册时输入设置 | 与学号绑定 | 用户具体信息录入 |
| 性别 | students | 在注册时输入设置 | 身份信息 | 用户具体信息录入 |
| 学院 | students | 在注册时输入设置 | 身份信息 | 用户具体信息录入 |
| 年级 | students | 在注册时输入设置 | 身份信息 | 用户具体信息录入 |
| 银行卡号 | bank\_card | 用户绑定时设置 | 使用银行卡充值 | 用户充值校园卡 |
| 商家 | food | 数据库数据输入 | 消费物品所属商家 | 用于线上消费 |
| 食品名称 | food | 数据库数据输入 | 消费物品名称 | 用于线上消费 |

注：此表格中常量所在目录的位置为该常量实际属性所在的目录，不包括以该常量为主键的所在目录。

### 4.2.2变量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量名称 | 所在目录 | 初值设置 | 功能说明 | 具体变量说明 |
| 校园卡号 | students | 系统随机生成 | 用户使用校园卡消费 | 挂失可重新申请校园卡，卡号可变 |
| 账号密码 | students | 初始均为123456 | 用于登录系统密码验证 | 用户可登录系统自行修改登录密码 |
| 卡内余额 | card | 初始以校园卡内具体余额为主 | 用户消费受余额限制 | 消费时卡内余额减少，充值时卡内余额增加 |
| 食品ID | food | / | 消费物品ID | 用于线上消费 |
| 食品价格 | food | / | 消费物品价格 | 用于线上消费 |
| 消费时间 | bill | 由系统生成 | 消费时间记录 | 导出账单时显示消费时间 |
| 充值时间 | chargerecord | 由系统生成 | 充值时间记录 | 显示(末次)充值记录时显示充值时间 |

注：此表格中变量所在目录的位置为该变量实际属性所在的目录，不包括以该变量为主键的所在目录。

### 4.2.3数据结构

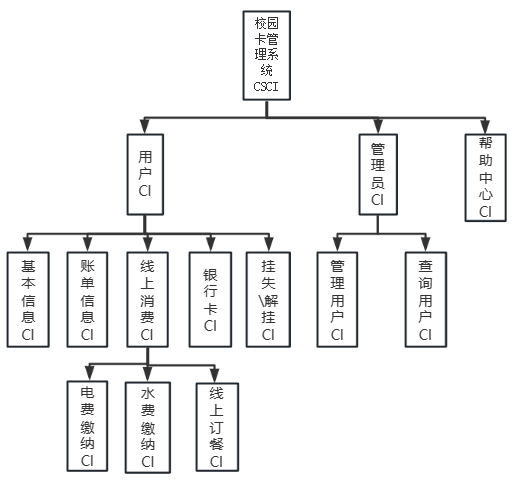
|  |  |  |  |
| --- | --- | --- | --- |
| 数据结构名称 | 数据变量 | 功能说明 | 具体数据结构说明 |
| 学生/students | s\_id | 用户学号信息 | 定义学生类别数据结构体，取值设置均为登录系统注册时设置，用于记录系统内用户的个人信息。 |
| s\_name | 用户姓名信息 |
| sex | 用户性别信息 |
| academy | 用户学院信息 |
| grade | 用户年级信息 |
| 校园卡/card | s\_id | 用户学号信息 | 定义校园卡类别数据结构，取值设置为系统根据学号分配校园卡号，标记余额和卡状态信息，区分挂失和解挂状态，用于记录系统内校园卡的具体信息，便于管理系统的管理。 |
| card\_id | 用户校园卡号信息 |
| remain\_sum | 校园卡余额信息 |
| state | 校园卡状态信息 |
| 银行卡/bank\_card | credit\_card\_num | 银行卡卡号 | 定义银行卡类别数据结构，取值通过读取数据库数据信息设置，该类别与校园卡充值功能相联系，通过验证密码判别本人操作进行绑定，充值时使用银行卡充值。 |
| b\_password | 银行卡密码 |
| 消费物品/food | product\_id | 消费物品ID | 定义消费物品类别数据结构，取值通过读取数据库数据信息设置，该类被与线上消费功能模块相关联，消费信息会写入到账单部分，用于线上消费模块信息展示。 |
| product\_name | 消费物品名称 |
| price | 消费物品价格 |
| shangjia | 消费物品所属商家 |

## 4.3 CSCI部件

**a.校园卡管理系统软件配置项以其标识符：**



**b.各软件配置项之间的静态关系：**



**c.各软件配置项的用途：**



**d.标识每个软件配置项的开发状态/类型**

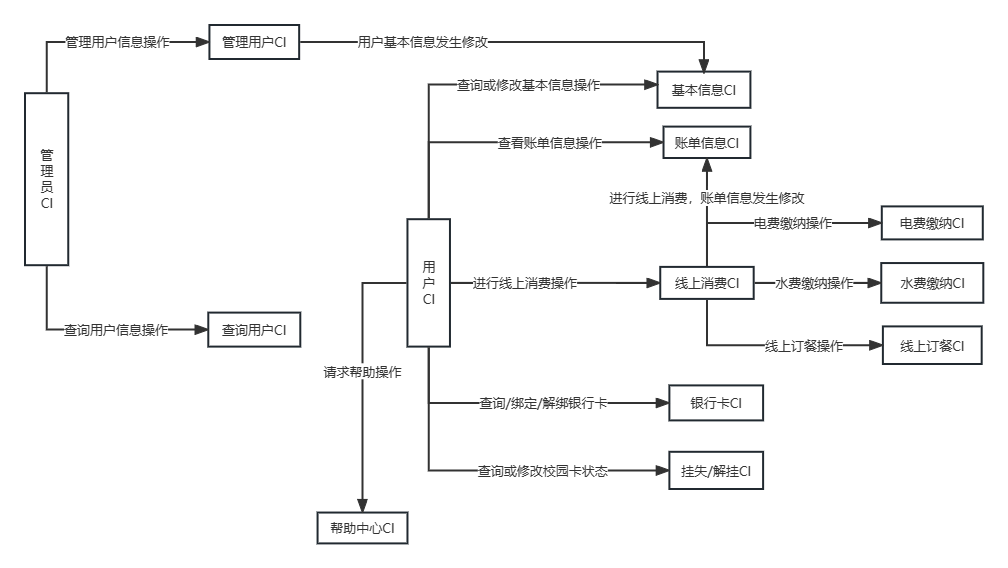
截至目前，大部分软件配置项，如用户CI、管理员CI、基本信息CI、账单信息CI等已经开发完毕，但仍有部分软件配置项，如电费缴纳CI、水费缴纳CI仍在开发状态。

**e.描述CSCI(若适用，每个软件配置项)计划使用的计算机硬件资源**

该CSCI计划使用的计算机硬件资源有：小型服务器一台，为用户实际使用提供网络服务；管理员PC若干，用于管理员进行系统管理。

## 4.4执行概念

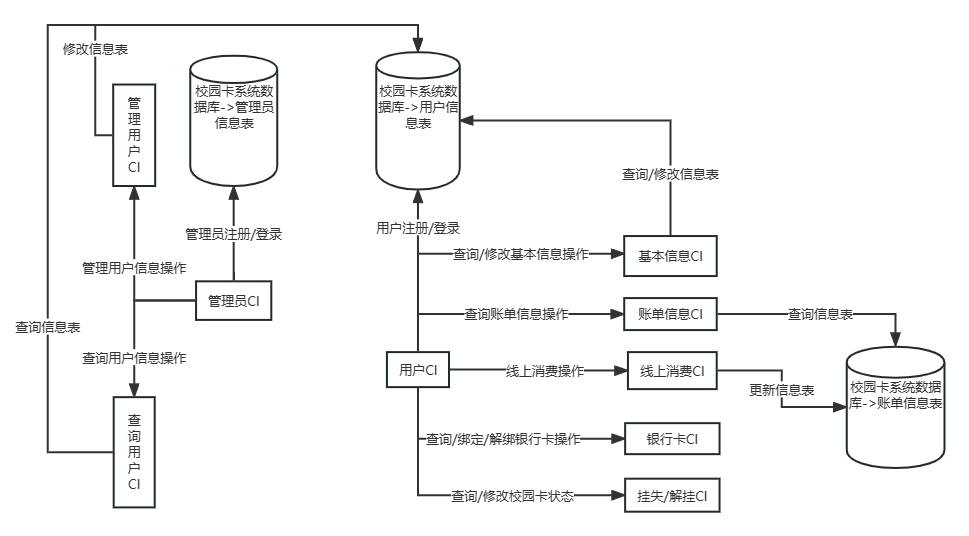
动态行为表：



在上述动态表中，管理员CI通过管理用户操作、查询用户操作分别与管理用户CI、查询用户CI相连接。用户CI则通过下述操作分别与下述CI连接。

1. 用户CI通过查询或修改基本信息操作与基本信息CI连接；
2. 用户CI通过查看账单信息操作与账单信息CI连接；
3. 用户CI通过进行线上消费操作与线上消费CI连接；
4. 用户CI通过查询/绑定/解绑银行卡操作与银行卡CI连接；
5. 用户CI通过查询或修改校园卡状态操作与挂失/解绑CI连接。

数据流：



在上述数据流中，管理员CI通过管理用户操作、查询用户操作分别与管理用户CI、查询用户CI相连接。在管理用户时，管理员CI对用户信息表进行修改；在查询用户时，管理员CI打开用户信息表。用户CI则通过下述操作分别与下述CI连接，且打开或修改一些信息表。

1. 用户CI通过查询或修改基本信息操作与基本信息CI连接，打开或修改用户信息表；
2. 用户CI通过查看账单信息操作与账单信息CI连接，打开账单信息表进行查询；
3. 用户CI通过进行线上消费操作与线上消费CI连接，更新账单信息表；
4. 用户CI通过查询/绑定/解绑银行卡操作与银行卡CI连接；
5. 用户CI通过查询或修改校园卡状态操作与挂失/解绑CI连接。

## 4.5接口设计

### 4.5.1外部接口

1. **用户模块**

（1）主页面：由11个功能按钮构成

基本信息、账单信息（消费记录/充值记录）、银行卡关联、银行卡充值、线上消费（电费缴纳/水费缴纳/线上订餐）、挂失、解挂、帮助中心

（2）基本信息页面：

学号：显示学号；

姓名：显示姓名；

性别：显示性别；

学院：显示学院；

年级：显示年级；

密码：显示密码；

校园卡卡号：显示校园卡卡号；

修改按钮：修改用户密码；

返回按钮：返回主页面；

（3）消费记录信息页面：

消费信息编号：显示消费信息编号；

校园卡卡号：显示校园卡卡号；

消费时间：显示消费时间；

消费产品类型：显示消费产品类型；

消费价格：显示消费价格；

消费商家：显示消费商家；

校园卡余额：显示校园卡余额；

查询编辑框：输入需要查询的相关信息；

查询按钮：显示对应的查询结果；

返回按钮：返回主页面；

（4）充值记录信息页面：

充值编号：显示充值编号；

校园卡卡号：显示校园卡卡号；

充值时间：显示充值时间；

银行卡卡号：显示银行卡卡号；

充值金额：显示充值金额；

校园卡余额：显示校园卡余额；

查询编辑框：输入需要查询的相关信息；

查询按钮：显示对应的查询结果；

返回按钮：返回主页面；

（5）银行卡关联页面：

绑定按钮：进行银行卡绑定

解绑按钮：进行银行卡解绑

（6）银行卡充值页面：

校园卡卡号：显示当前登录用户的校园卡卡号

学号：显示学号

校园卡余额：显示校园卡余额

银行卡卡号：显示已绑定的银行卡的卡号

充值金额编辑框：输入要充值的金额

确认按钮：提交请求

重置按钮：清空选择/输入项，可重新选择/输入

返回按钮：返回主页面；

（7）电费缴纳页面：

校园卡卡号：显示当前登录用户的校园卡卡号

校园卡余额：显示校园卡余额

楼号编辑框：输入要缴纳电费的楼号

宿舍号编辑框：输入要缴纳电费的宿舍号

缴纳金额编辑框：输入需要缴纳的电费的金额

返回按钮：返回主页面

（8）水费缴纳页面：

校园卡卡号：显示当前登录用户的校园卡卡号

校园卡余额：显示校园卡余额

楼号编辑框：输入要缴纳水费的楼号

宿舍号编辑框：输入要缴纳水费的宿舍号

缴纳金额编辑框：输入需要缴纳的水费的金额

返回按钮：返回主页面

（9）线上订餐页面：

餐食序号：显示餐食序号

食品名称：显示食品名称

商家：显示商家名称

价格：显示餐食价格

购买按钮：进行购买对应餐食

返回按钮：返回主页面

（10）挂失页面：

校园卡密码编辑框：输入校园卡密码

确认按钮：提交挂失申请

返回按钮：返回主页面

（11）解挂页面：

校园卡密码编辑框：输入校园卡密码

确认按钮：提交解挂申请

返回按钮：返回主页面

（12）帮助中心页面：

返回按钮：返回主页面

1. **管理员模块**

（1）主页面：由3个功能按钮构成

用户基本信息管理、用户账单信息查询（消费记录查询/充值记录查询）

（2）用户基本信息管理页面：

学号：显示学号；

姓名：显示姓名；

性别：显示性别；

学院：显示学院；

年级：显示年级；

密码：显示密码；

校园卡卡号：显示校园卡卡号；

查询输入框：输入查询条件查询用户

修改按钮：修改用户信息

添加按钮：添加新用户

删除按钮：删除用户信息

返回按钮：返回主页面；

（3）消费记录查询页面：

消费信息编号：显示消费信息编号；

校园卡卡号：显示校园卡卡号；

消费时间：显示消费时间；

消费产品类型：显示消费产品类型；

消费价格：显示消费价格；

消费商家：显示消费商家；

校园卡余额：显示消费完后的校园卡余额；

查询编辑框：输入需要查询的相关信息；

查询按钮：显示对应的查询结果；

返回按钮：返回主页面；

（4）充值记录查询页面：

充值编号：显示充值编号；

校园卡卡号：显示校园卡卡号；

充值时间：显示充值时间；

银行卡卡号：显示银行卡卡号；

充值金额：显示充值金额；

校园卡余额：显示充值完后的校园卡余额；

查询编辑框：输入需要查询的相关信息；

查询按钮：显示对应的查询结果；

返回按钮：返回主页面；

1. 软件接口

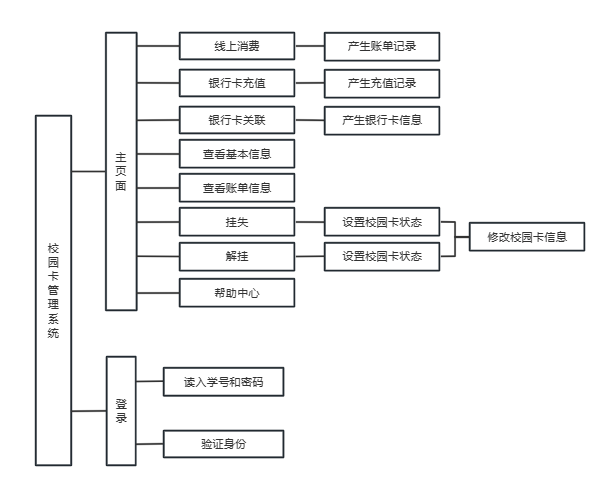
使用MySQL数据库的驱动程序，通过浏览器访问系统。

1. 硬件接口

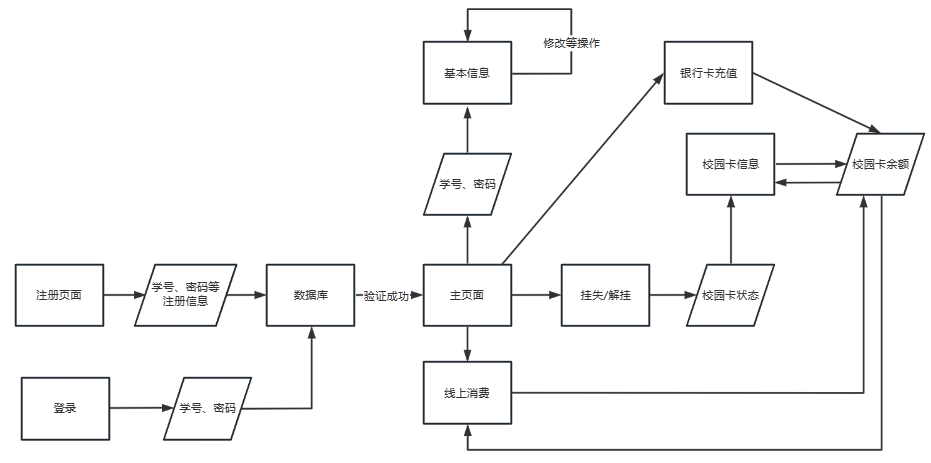
支持windows10及以上PC机

### 4.5.2内部接口

内部接口调用关系：



内部接口数据交互：



**1.用户**

（1）主模块

输入：用户登录传入的账号密码等以及用户的其余输入数据

输出：用户界面显示

上层模块：无

下层模块：基本信息模块、账单信息模块、银行卡关联模块、银行卡充值模块、线上消费模块、挂失模块、解挂模块、帮助中心模块

（2）基本信息模块：

输入：无

输出：用户基本信息显示

上层模块：主模块

下层模块：数据库查询模块，数据显示模块

（3）消费记录信息模块：

输入：无

输出：用户消费信息显示

上层模块：账单信息模块

下层模块：数据库查询模块，数据显示模块

（4）充值记录信息模块：

输入：无

输出：用户充值记录信息显示

上层模块：账单信息模块

下层模块：数据库查询模块，数据显示模块

（5）银行卡关联模块：

输入：无

输出：界面显示

上层模块：主模块

下层模块：数据显示模块

（6）银行卡充值模块：

输入：充值金额

输出：校园卡卡号、学号、校园卡余额、银行卡卡号

上层模块：主模块

下层模块：数据库查询模块、数据显示模块

（7）电费缴纳模块：

输入：宿舍号、楼号、缴纳金额

输出：校园卡卡号、校园卡余额

上层模块：线上消费模块

下层模块：数据库查询模块、数据显示模块

（8）水费缴纳模块：

输入：宿舍号、楼号、缴纳金额

输出：校园卡卡号、校园卡余额

上层模块：线上消费模块

下层模块：数据库查询模块、数据显示模块

（9）线上订餐模块：

输入：无

输出：餐食序号、食品名称、商家、价格

上层模块：线上消费模块

下层模块：数据库查询模块、数据显示模块

（10）挂失模块：

输入：校园卡密码

输出：挂失是否成功的信息

上层模块：主模块

下层模块：数据库查询模块、数据显示模块

（11）解挂模块：

输入：校园卡密码

输出：解挂是否成功的信息

上层模块：主模块

下层模块：数据库查询模块、数据显示模块

（12）帮助中心模块：

输入：无

输出：帮助信息页面

上层模块：主模块

下层模块：无

**2.管理员**

（1）主模块：

输入：管理员登录传入的账号密码等以及管理员的其余输入数据

输出：管理员界面显示

上层模块：无

下层模块：用户基本信息管理模块、用户账单信息查询模块

（2）用户基本信息管理模块：

输入：用户信息关键字

输出：相关用户信息条目

上层模块：主模块

下层模块：数据库查询模块、数据显示模块

（3）消费记录查询：

输入：用户消费信息关键字

输出：相关消费记录显示

上层模块：用户账单信息查询模块

下层模块：数据库查询模块、数据显示模块

（4）充值记录查询：

输入：用户充值记录信息关键字

输出：相关充值记录显示

上层模块：用户账单信息查询模块

下层模块：数据库查询模块、数据显示模块

# 5 CSCI详细设计

**a.配置项设计决策，诸如(如果以前未选)要使用的算法**

①聚类算法：通过对校园卡消费数据进行分析，可以将学生分为不同的消费群体，从而了解学生的消费习惯和需求，为学校提供有针对性的服务和优化建议。例如，利用K-Means算法，可以将学生按照消费金额、消费时间、消费地点等特征进行聚类，发现不同类型的消费者。

②关联规则算法：通过挖掘校园卡消费数据中的频繁项集和强规则，可以发现学生的消费行为和学习成绩之间的关系，为学生提供个性化的辅导和指导。例如，利用Apriori算法，可以找出学生在食堂、图书馆、体育馆等场所的消费模式，以及这些模式与学生的成绩水平的相关性。

③人脸识别算法：通过利用移动端微应用和人脸识别技术，可以实现校园卡的虚拟化平台，为用户提供更安全、更便捷、更可扩展、更高集成度的虚拟校园卡服务。例如，利用深度神经网络和卷积神经网络等算法，可以实现高精度、高速度、高鲁棒性的人脸检测和识别功能，从而实现用户的身份验证和支付授权。

以上算法功能尚未实现，计划会在管理系统上线后，在实际应用中逐步实现和完善。

**b.软件配置项设计中的约束、限制或非常规特征**

①技术条件。校园卡管理系统的设计应该符合相关的技术标准和规范，例如安全性、可靠性、可维护性、可扩展性等。技术条件会影响到系统的架构、模块、接口、数据结构等方面，因此需要进行详细的技术设计和评估，以确保系统的稳定性和性能。

②成本预算。校园卡管理系统的设计应该考虑到项目的成本预算，包括硬件设备、软件开发、人力资源、运维支持等方面。成本预算会限制系统的功能范围、质量水平、交付时间等方面，因此需要进行合理的成本控制和优化，以实现系统的高效率和高效益。

③法律法规。校园卡管理系统的设计应该遵守相关的法律法规，例如个人信息保护、知识产权保护、税务管理等。法律法规会规范系统的行为和责任，因此需要进行充分的法律咨询和审查，以避免系统的合法性和合规性问题。

**c.如果软件配置项包含、接收或输出数据，(若适用)应有对其输入、输出和其他数据元素以及数据元素集合体的说明。**

校园卡管理系统数据库是一个关系型数据库，采用PHP myadmin作为数据库管理系统，实现数据的存储、查询、更新和备份。数据库主要包含以下几个方面的数据：

- 用户数据：包括用户的基本信息、身份认证信息、账户信息和消费记录等。

- 业务数据：包括校园卡的消费类型、消费流程和消费状态等。

- 系统数据：包括校园卡的挂失设置、日志记录、权限管理和数据字典等。

校园一卡通管理系统数据库按照模块划分为以下几个子系统：

- **用户子系统**：负责管理用户的基本信息、身份认证信息、账户信息和消费记录等，主要包括以下几个表：

- USER\_INFO：用户基本信息表，存储用户的姓名、性别、学号/工号、所属院系/部门等。

- USER\_AUTH：用户身份认证信息表，存储用户的密码等。

- USER\_ACCOUNT：用户账户信息表，存储用户的账户余额、冻结金额、充值记录等。

- USER\_CONSUME：用户消费记录表，存储用户的消费时间、地点、金额、业务类型等。

- **业务子系统**：负责管理一卡通的业务类型、业务规则、业务流程和业务状态等，主要包括以下几个表：

- BUSINESS\_TYPE：业务类型表，存储一卡通支持的各种业务类型，如餐饮消费、水电缴费等。

- BUSINESS\_PROCESS：业务流程表，存储一卡通各种业务类型的流程，如消费流程、充值流程、挂失流程等。

- BUSINESS\_STATUS：业务状态表，存储一卡通各种业务类型的状态，如正常状态、挂失状态等。

- **系统子系统**：负责管理系统的参数设置、日志记录、权限管理和数据字典等，主要包括以下几个表：

- SYSTEM\_PARAM：系统参数表，存储系统的各种参数设置，如数据库连接参数等。

- SYSTEM\_LOG：系统日志表，存储系统的各种日志记录，如操作日志、异常日志等。

- SYSTEM\_ROLE：系统角色表，存储系统的各种角色定义，如管理员角色、操作员角色、用户角色等。

- SYSTEM\_PRIVILEGE：系统权限表，存储系统的各种权限定义，如查询权限、修改权限、删除权限等。

- SYSTEM\_DICT：系统数据字典表，存储系统的各种数据

# 6需求的可追踪性

|  |  |
| --- | --- |
| 系统功能 | 涉及的子系统 |
| 用户通过本系统查看基本信息 | 用户信息管理子系统 |
| 用户通过本系统修改基本信息 | 用户信息管理子系统 |
| 用户通过本系统查看消费记录 | 账单信息管理子系统 |
| 用户通过本系统查看充值记录 | 账单信息管理子系统 |
| 用户通过本系统进行银行卡关联 | / |
| 用户通过本系统进行银行卡充值 | 账单信息管理子系统、校园卡信息管理子系统 |
| 用户通过本系统进行线上消费 | 账单信息管理子系统、校园卡信息管理子系统 |
| 用户通过本系统对校园卡进行挂失/解挂 | 校园卡信息管理子系统 |
| 用户查看系统使用帮助 | / |
| 管理员通过本系统管理用户账户信息 | 用户信息管理子系统 |
| 管理员通过本系统查看用户消费记录与充值记录 | 账单信息管理子系统 |

# 7注解

（1）MySQL：MySQL数据库服务是一个完全托管的数据库服务,是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。

（2）WampServer64：WampServer（64位）是一个Windows Web开发环境。它允许使用Apache，PHP和MySQL数据库创建Web应用程序。它还带有PHPMyAdmin，可以轻松管理数据库。

（3）phpMyAdmin：phpMyAdmin是一个用PHP编写的免费软件工具， 旨在通过Web处理MySQL的管理。

（4）PHP：PHP 是一种创建动态交互性站点的强有力的服务器端脚本语言。

# 附录

**附录A：学习笔记**

**①202000130221-连欣**

如何设计出卓越、高效且流畅的用户界面，这是进行产品开发的一个热门问题。

用户使用一款产品，首先体验是用户界面。一个美观出色的用户界面可以迅速抓住用户的第一新鲜感，吸引用户继续使用产品。同时，除了美观，好的用户界面应该实用、易用，如果产品的功能不实用，那么再美观的界面也不能达到用户的使用需求，而对于实用且使用简单的产品界面，美观就是锦上添花。总的来说，用户界面设计就是以使用者为中心，使产品达到简单使用和愉悦使用的设计，是软件设计开发的重要一环。

Ben Shneiderman在《用户界面设计：有效的人机交互策略》一书中提出了界面设计的八个黄金法则[1]：

①**一致性**。在设计类似的情况和操作顺序时，通过利用相似的图标、颜色、菜单层次结构、新手引导、用户流程来争取一致性。统一信息传递的方式，确保用户能够将知识从一次点击迁移到另一次；而无需为同样的行为学习新的操作。通过帮助用户熟悉产品的数字环境，一致性扮演着重要的角色，从而可以更轻松地实现目标。

②**让经常使用你产品的用户可以使用快捷键**。随着使用频率的增加，用户需要更快的方法以完成任务。例如，Windows和Mac都为用户提供了用于复制和粘贴的键盘快捷方式，因此随着用户变得更有经验，他们可以更快速和轻松地导航和操作用户界面。

③**提供有意义的反馈**。用户应该始终知道他们当前位于哪里以及发生了什么。对于每一个行动，在合理的时间内应该有适当的、人类可读的反馈。一个很好的例子就是用户在回答多页问卷时，向用户表明他们在哪里。我们经常看到的一个不好的例子是，一个错误信息的提示，显示着错误代码而不是一个可读的、有意义的信息。

④**告知状态**。不要让你的用户猜测。告诉他们他们的行为把他们导向了哪里。例如，用户在完成在线购买后，会很喜欢“感谢购买”这样的提示以及购买收据的证明。

⑤**提供简单的错误处理**。没有人喜欢被告知他们错了，尤其是你的用户。系统的设计应该尽可能地防止错误，但是当发生不可避免的错误时，应确保向用户提供简单、直观的分步说明，以尽快、尽可能没有痛苦得解决问题。例如，标记出用户忘记填写的表单字段。

⑥**允许轻松地进行撤销操作**。设计师应该为用户提供明显的方式来扭转他们的行为。不管是单一行动、数据输入还是整个行动序列，都应允许这些逆转。正如Shneiderman在他的书中所述：“这个功能减轻了焦虑，因为用户知道错误可以被撤销; 因此这鼓励了用户探索不熟悉的功能。”

⑦**让用户有掌控感。**让你的用户成为行动的发起者。让用户感觉到他们完全控制了数字空间中发生的事件。当你设计的系统的行为像他们所期望的那样，你就赢得了他们的信任。

⑧**减少短期记忆负载。**人的注意力是有限的，我们一次只能在短期记忆中记住五项内容左右。因此，界面应尽可能简单，信息层次要适当，让用户去识别信息而不是去回忆。识别事物总是比回忆容易，因为识别涉及找到线索，并进入我们广阔的记忆，允许相关的信息浮出水面。例如，我们经常发现多选题比问答题容易，因为它只需要我们识别答案，而不是从我们的记忆中回想起来。

用户界面是人与机之间交流、沟通的层面。按照八个黄金法则设计出来的界面，充分考虑了用户的使用体验，极大促进了人机交互的和谐融洽，同时也是有效的人机交互。而设计效果差的用户界面，比如界面操作步骤繁琐，缺乏逻辑，盲目堆放界面元素；对重要操作不提供结果反馈；界面言语不专业，存在语病等，是无法在当今时代五花八门的产品中脱颖而出的。

所以我们在进行界面设计时，尽管可能很难去兼顾这八个黄金法则，但也需要充分考虑一个出色卓越的页面带给用户的体验感和接受度。试想一个产品让用户在使用上有负担，无法高效率使用，用户满意度必然是低的，又如何推广开呢？

**②202000130039-高雅**

随着软件系统的规模增加，计算的算法和数据结构不再构成主要的设计问题。当系统由许多组件构成时，整个系统的组织——软件架构提出了一组新的设计问题。软件架构是指软件系统的整体结构，包括系统的组成部分、各部分之间的关系以及与环境的接口。

常见的软件架构风格有很多，其中包括：[3]

·管道和过滤器（Pipes and Filters）

·数据抽象和面向对象组织（Data Abstraction and Object-Oriented Organization）

·基于事件的隐式调用（Event-based, Implicit Invocation）

·分层系统（Layered Systems）

·存储库（Repositories）

·表驱动解释器（Table Driven Interpreters）

·其他常见的架构风格，如客户端-服务器、主从、发布-订阅等

系统的SAD是该系统的一个或多个结构，包括软件元素、这些元素的外部可见属性以及它们之间的关系。“外部可见”属性是指其他元素可以对一个元素做出的假设，如其提供的服务、性能特征、故障处理、共享资源使用等。该定义为SAD中包含哪些信息以及哪些信息被降级到下游文档提供了基本的试金石。

软件体系结构首先体现了关于元素如何相互关联的信息。这意味着体系结构专门省略了与元素交互无关的某些元素信息。因此，软件体系结构是系统的抽象，它抑制了元素的细节，这些细节不会影响它们如何使用、被其他元素使用、与其他元素相关或与其他元素交互。元素之间通过接口进行交互，这些接口将元素的详细信息划分为公共部分和私有部分。软件体系结构与该部门的公共方面有关，这将相应地记录在本SAD中。另一方面，元素的私有细节——仅与内部实现有关的细节不是体系结构的，也不会记录在SAD中。

例如，所有非平凡的软件系统都被划分为实现单元；这些单元被赋予了特定的职责，是编程团队工作分配的基础。这种元素将包括其他实现单元中的软件可以调用或访问的程序和数据，以及私有的程序和资料。

在大型项目中，元素几乎肯定会被细分，分配给子团队。这是一种经常用来描述系统的结构。它是一个非常静态的结构，因为它专注于系统功能的划分和分配给实现团队的方式。尽管软件体系结构倾向于关注结构信息，但每个元素的行为都是软件体系结构的一部分，只要可以从另一个元素的角度观察或辨别该行为。这种行为允许元素相互交互，这显然是软件体系结构的一部分，并将在SAD中记录下来。行为记录在每个视图的元素目录中。

SAD分为以下几个部分：文档路线图和概述提供了有关本文档及其预期受众的信息。它提供了路线图和文档概述。每一位希望找到与本文档中描述的软件体系结构相关的信息的读者都应该从阅读本节开始，本节描述了文档的组织方式以及可以在哪里找到信息。

体系结构背景提供了有关软件体系结构的信息。它描述了软件体系结构的背景和基本原理。它解释了导致当前体系结构的约束和影响，并描述了体系结构中使用的主要体系结构方法。视图和视图之间的映射指定软件体系结构。参考资料、术语表和缩略词提供参考信息。参考材料提供了本SAD其他地方引用的文件的查找信息。术语表和首字母缩略词是一个体系结构元素和关系的索引，给出了它们的定义，以及每个元素在本SAD中的使用位置。

**③202000130045-岑婉琴**

软件体系结构是由软件元素的外在可见性质，以及元素之间的关系组成的一种结构。研究表明，软件体系结构在软件的各个阶段都起到了十分重要的作用。他不但可以控制软件的复杂性和提供高层次的复用，而且作为工程计划的前期，它可以让开发者在早起从全局来考虑关键性设计问题，从而保持整个软件的一致性。[4]

研究软件体系结构的首要问题就是如何表现软件体系结构，即我们要如何对它进行建模。根据建模侧重点的不同，可以将软件体系结构的模型分为5种：结构模型、框架模型、动态模型、过程模型和功能模型。其中，最常用的是结构模型与动态模型。

其次，软件开发模型的发展基于体系结构。软件开发模型是跨越整个软件生存周期的系统开发、运行、维护所实施的所有工作与任务的结构框架。它给出了软件开发活动各阶段之间的关系。目前为止，常见的软件开发模型可大致分为三种：(1)以软件需求完全确定为前提的瀑布模型；(2)在软件开发初始阶段只能提供基本需求时所采用的渐进式开发模型，比如螺旋模型；(3)以形式化开发方法为基础的变换模型。

软件体系结构的开发是大型软件系统开发的关键环节。而体系结构在软件生产线的开发中则起到至关重要的作用。自NATO于1968年提出软件工程概念以来，软件工程界已经提出了一系列的理论、方法及语言等，解决了软件开发过程中的若干问题。然而，软件固有的复杂性、易变性与不可见性，让软件开发周期长、代价高及质量低的问题仍旧存在。为此，软件体系结构概念应运而生，它在软件需求和软件设计间架起一座桥梁，着重解决软件系统的结构和需求向实现平坦地过渡的问题。

鉴于软件体系结构的重要性，D.E.Perry将软件体系结构视为软件开发中第1类重要的设计对象。而Barry Boehm明确指出，在没有设计出体系结构及其规则时，那么整个项目都不能继续下去，而体系结构应该看做是软件开发中可交付的中间产品。由此可见，体系结构在软件开发的过程中为不同的工作人员提供了可以共同交流的语言，体现且尝试了系统早期的设计决策。同时，它作为系统设计的抽象，它为实现框架与构建的共享和复用、基于体系结构的软件开发提供了强有力的支持。

**④202000130156-姜闪**

1.为什么要使用软件体系结构的思想来开发软件产品？在开发软件产品的过程中，软件体系结构的思想又是如何体现的呢？

软件的发展是非常迅速的，从最初的程序发展到如今的程序加文档，所要实现的功能已经是愈加庞大，同时对软件各方面功能和性能的要求也决定了软件系统空前的复杂。软件的开发已经不是简单的写出程序，在软件开发的各个生命周期中，从需求分析到产品设计到编码到测试维护，软件开发的侧重点从编码已经转移到需求分析和软件总体架构设计，而许多项目正是在生成出产时，才发现问题是在结构上，由于结构的局限性，整个开发过程有很多不必要的浪费，在体系结构设计上所体现的重要性就显得特别突出。

1. 软件体系结构在软件开发过程中的作用：

体系结构在某种意义下是与需求密切相关的，明确的需求可以制定明确的软件规格，意味着越明确的规格所设计出来的软件体系架构越清晰，而对于产品的需求变更也是需要考虑的一点，若产品有明确的变更趋势也可以在早期的设计中体现出来。软件的高度复杂也决定了软件开发所需要的充分条件设计，就需要研究开发模式，研究体系结构，从宏观和更抽象的层次把握软件的开发，并把整个过程使用文档的形式记录下来，在软件开发人员与系统设计人员、用户以及其他有关人员之间进行沟通交流，从而达成共同的理解。如果有明确的需求和规格，就可以对软件进行详细的结构设计，从用例到类图到关键部分的序列图活动图等，设计越详细越好，可以尽量让更多的人了解到项目的需求与实现环境，未涉及提出有效建议，结构设计注重体系的灵活性，较多的考虑各种可能出现的变更情况，这是设计过程中较为关键的阶段，但是通常情况下，客户不会给出特别明确的需求。应用软件体系结构的设计思想和方法可以很大程度上的划分范围、确定时间、规划成本、保证质量。而对于整个项目，一般会受到多种限制，特别突出的一点是计划特征，面对这些突出的问题，在设计架构上要注意模块的划分，划分时模块越独立越好，尽量把有明确需求的应用划分为独立的模块，尽量减少模块与模块之间的功能交集，减少出现某个模块出现问题会导致其他模块的风险出现。层次化设计是一层一层分割，将整个系统分成多个明显层次，层次体系结构利用分层的处理方式来处理复杂的功能，层次系统由于是上层子系统使用了下层子系统的功能，而下层子系统不能使用上层子系统的功能，下层每个程序接口执行当前的一个简单的功能，而上层通过调用不同的下层子程序，并按不同的顺序来执行这些下层子程序，有效的减少了不同层次之间不该有的交集，同时也减少了错误的发生，便于软件系统检错过程的执行。[5][6]

**⑤202018130164-张玮玮**

在小组作业中，我负责的模块是软件架构方面，因此，我选择精读了David Garlan的《Software Architecture:a Roadmap》[2]，并查找阅读了其他系列与这一主题相关的文献。通过本次的学习，对软件架构概念、全过程、实现、以及发展有了较为清晰深刻的认识。

随着技术的发展，软件体系结构作为软件工程的一个重要分支受到关注越来越多，不少人提出将软件体系结构设计作为一门独立的工程学科，尽管在技术和方法的基础方面取得了较大的进步，然而，距离让其称为一门独立的工程学科依然任重道远。但通过考察软件体系架构的研究和实践里一些重要趋势，对重要的新兴趋势、挑战和抱负进行推测，可以构架其未来发展路线。

首先，应当清楚软件架构在整个软件设计过程中扮演怎样的角色。尽管关于软件架构的定义有很多，但所有定义的核心都是一个系统的架构描述其总体结构的概念。这个结构说明了 顶层的设计决策，包括系统是如何由相互作用的部分组成的，交互的主要途径在哪里，以及各部分的关键属性是什么。此外，一个体系结构描述包括足够的信息，以允许高 层次的分析和关键评估。通俗而言，软件架构通常作为需求和实现之间的桥梁发挥关键作用。通过提供系统的抽象描述，架构暴露了某些属性，同时隐藏了其他属性。理想情况下，这种表示形式为整个系统提供了一种智力上易于处理的指南,允许设计者推理一个系统满足某些需求的能力,并建议一个系统构建和组成的蓝图。例如，信号处理应用程序的架构可能构为一个数据流网络，其节点读取输入数流， 转换该数据， 并入输出流。设计人员可能使用这种分解，连同输入数据流的估计值 、计算本和缓冲能力，来推理可能的瓶颈、资源需求和计算的可调度性。详细地说，软件架构可以在软件开发的至少六个方面发挥重要作用，分别是理解、重用、构造、演化、分析、管理。

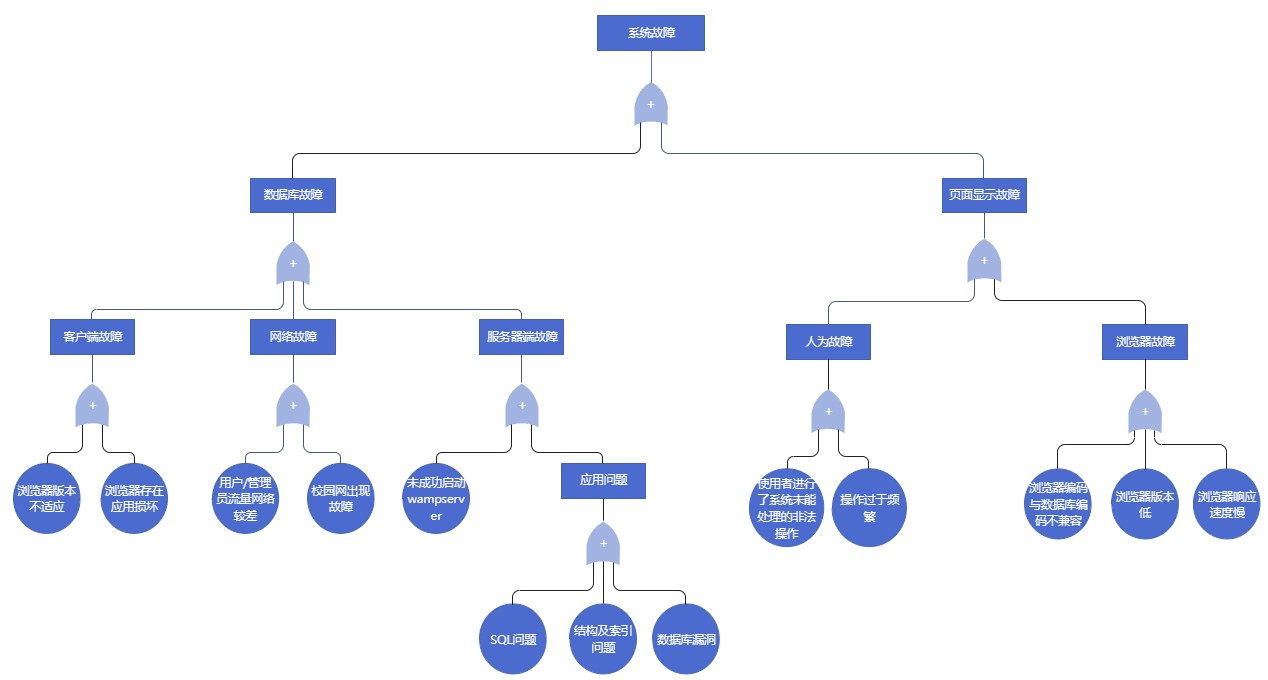
其次，了解现今软件体系架构的技术基础发展状况。如今，体系架构技术基础有了很大的提高，其中重要体现有：体系结构描述语言和工具、产品线和标准、编码和传播。软件体系结构的描述语言前期的发展经历了从模糊的box-and-line描述、到ADLs描述，再到基于集成架构的ADLs工具包的历程，如今有许多提案试图展示如何将在adl中发现概念直接映射到像UML，使用UML这样更通用的 建模语言，其优点是提供了一种实践者更可能熟悉的符号，并提供了更直接的 面向对象实现和开发的链接工具，但是UML这样的通用语言的存在一个明显缺点，就是对象概念词汇表可能不适合表达架构概念，并且自动分析架构属性的机会更少。软件系统架构的一个重要趋势是希望利用多个产品之间的共性，这一趋势的 两个具体表现是我们在组织内创建产品线的能力的提高和跨供应商集成标准的出现。关于产品线，关键的挑战是产品线方法需要不同的开发方法。在单产品方法中，架构必须根据该产品的要求单独进行评估。此外，单个产品可以独立 构建，每个产品都有不同的架构。像产品线方法一样，跨供应商集成标准要求架构框架允许系统开发人员可以通过实例化该框架来配置各种特定的系统。这样的 标准可能是正式的国际标准(例如由 IEEE 或 ISO 赞助的 标准)，或者由行业领导者推动的临时和事实上的标准。

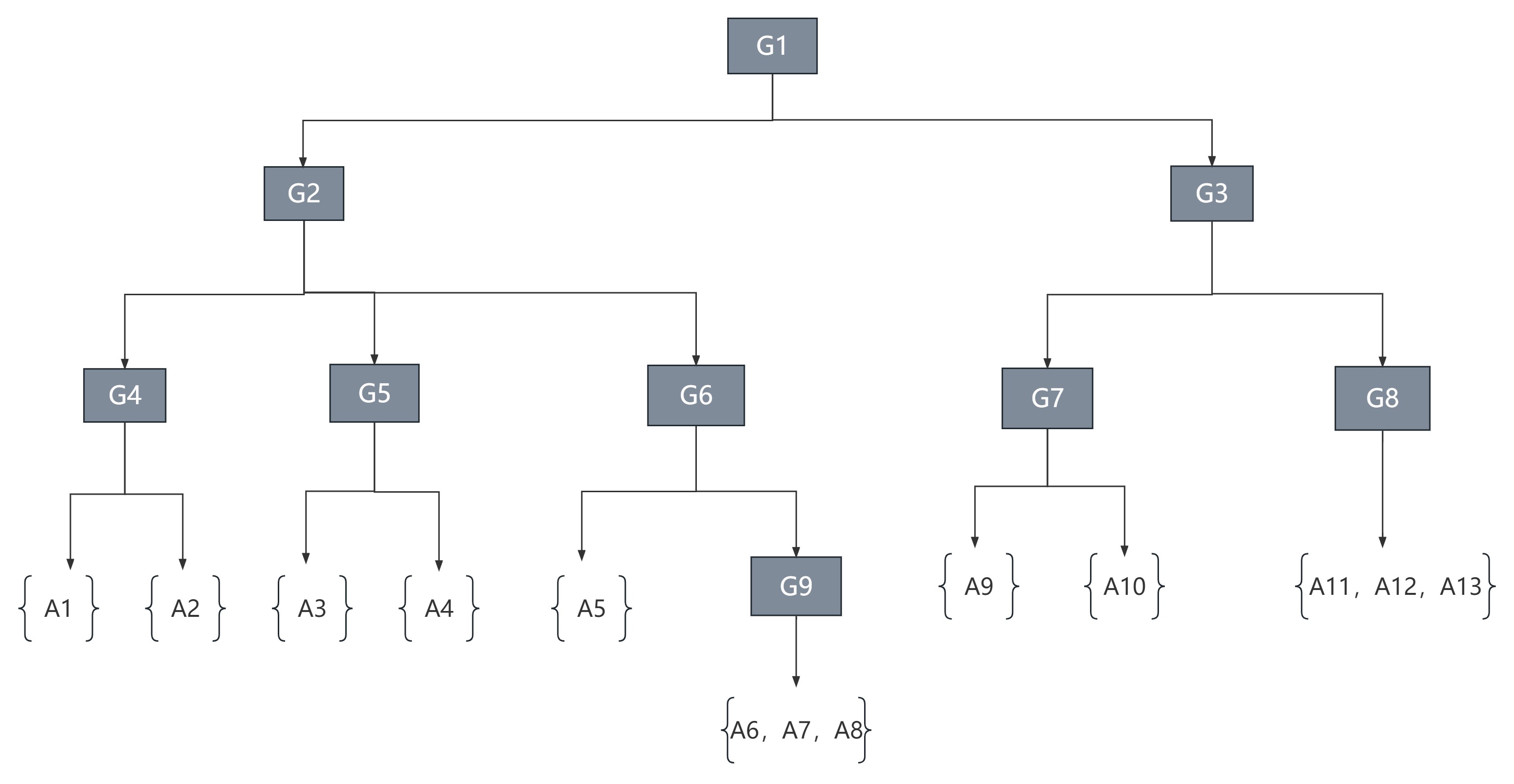
最后，考虑一些突出趋势对软件系统架构未来发展的影响。第一，改变构建与购买之间的平衡。在软件工程的历史上，系统开发的一个关键问题一直是决定系统的哪些部分可以从别处获得，哪些部分可以内部开发。这就衍生出构建与购买之间的矛盾，因为不了解哪些可以通过软件提高，哪些必须经由硬件改善，就会造成购买不必要的硬件设备或者缺少相应的硬件设备，二者之间的平衡即是软件开发性价比之间的权衡要是最优化资源的平衡。第二，以网络位中心的计算模型。从以pc为中心的计算模型向以网络为中心的计算模型是当今计算机领域发展的一个明显的趋势，同样，软件系统架构领域也无法避免，历史上，软件系统一直是作为封闭系统开发的，在个别机 构的控制下开发和运行。组成部分可以从外部来源获得， 但当纳入系统时，它们就处于系统设计者的控制之下。这 种系统的体系结构要么是完全静态的、不允许运行时重构，要么只允许有限形式的运行时变化，显然，这种模式的弊端很多。但是，在通过网络提供普适服务的世界中，系统可能没有这样的集中控制。互联网就是这样一个开放系统的例子: 它进行了最低限度的标准化，主要是在协议、地址和允许个别站点进行交互的表示的层次上。它允许在低于这些标准的硬件和高于这些标准的应用程序方面存在相当大的差异。没有中央权威机构进行控制或验证。各个站点是独立管理的。个体开发者可以随意提供、修改和移除资源；第三，普适计算，在计算世界中充斥着各种各样的异构计算设备，对于这些设备，我们需要在其物理资源和计算能力方面进行特殊考虑，可想而知，处理异构性需要耗费较多冗余的人力和财力的投入，因此构建普适计算统一化标准就显得尤为必要。

通过相关专业知识的学习，我掌握了关于软件系统架构方面的专业知识，在可预见的未来软件系统架构也将有相当大的发展，但是随着它逐渐成熟成为一门被普遍认可和实践的工程学科，也必将面临许多挑战。同时也更加深刻地理解，硬件的发展和技术的进步是相辅相成的，两者相互促进，共同推动软件领域的发展。

**附录B：故障树与割集树**

故障树：





注：上图中所涉及到的节点符号均在下表中给出。

|  |  |
| --- | --- |
| 割集树节点符号 | 符号具体含义 |
| G1 | 系统故障 |
| G2 | 数据库故障 |
| G3 | 页面显示故障 |
| G4 | 客户端故障 |
| G5 | 网络故障 |
| G6 | 服务器端故障 |
| G7 | 人为故障 |
| G8 | 浏览器故障 |
| G9 | 应用问题 |

|  |  |
| --- | --- |
| 割集树叶子结点符号 | 符号具体含义 |
| A1 | 浏览器版本不适应 |
| A2 | 浏览器存在应用损坏 |
| A3 | 用户/管理员流量网络较差 |
| A4 | 校园网出现故障 |
| A5 | 未成功启动wampserver |
| A6 | SQL问题 |
| A7 | 结构及索引问题 |
| A8 | 数据库漏洞 |
| A9 | 使用者进行了系统未能处理的非法操作 |
| A10 | 操作过于频繁 |
| A11 | 浏览器编码与数据库编码不兼容 |
| A12 | 浏览器版本低 |
| A13 | 浏览器响应速度慢 |