西北工业大学明德学院
JOURNAL OF NORTHWESTERN POLYTECHNICAL UNIVERSITY MING DE COLLEGE

《数据库实训项目报告》

项目名称 校园一卡通管理系统(DBMS)

专业名称 计算机科学与技术

学生班级 101011901

组长 杨乃宸

组员 孙岩 檀磊 杨天硕

完成日期 2022 年 09 月 13 日 星期二

一.　需求分析(主要完成人：全体成员)2

1.1 系统需求分析 3

1.2 需求规格说明书 5

二.　概要设计(主要完成人：全体成员)6

2.1 系统功能模块 6

2.2 各模块功能设计 6

三.　详细设计(主要完成人：全体成员)7

3.1 系统业务流程分析　7

3.2 各子模块的详细设计　8

四. 数据库设计与实现(主要完成人：全体成员) 10

4.1 概念模型设计(E-R 模型) 10

4.2 数据库逻辑结构设计　11

4.3 数据库物理结构实现(创建数据库过程)　12

4.4 其他对象的创建(视图、存储过程、触发器.....) 15

一.　需求分析(主要完成人：全体成员)

1.1 系统需求分析

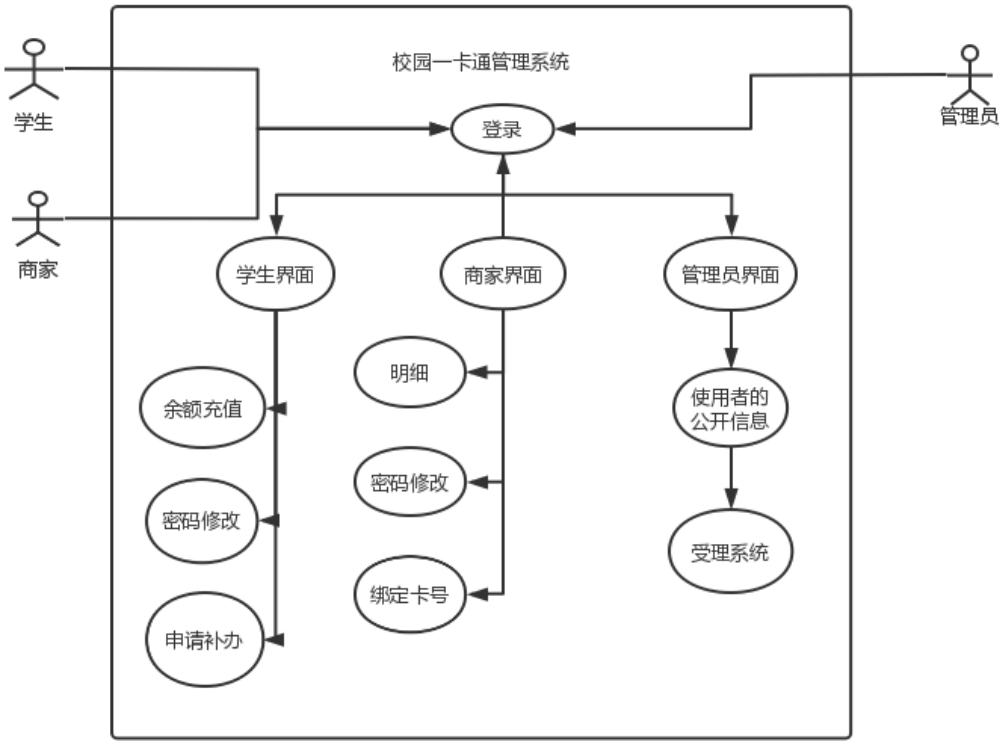
1、校园一卡通管理系统：

为了服务学生校园生活，同学们通过这个系统可以到超市、餐厅、健身房、小吃城、理发店、洗衣房等校园场所以及校园的水电费、学费都可以通过此系统实现。

2、实体：

- 全体学生：学生们统一办理校园卡，填写个人信息注册校园一卡通系统，学生们可以通过校园一卡通系统进行校园卡的余额充值以及个人账户信息修改。
- 全体商家：入驻校园的商家可以进行办理注册一卡通商家系统，通过此系统商家可以注册店铺信息以及可以查看学生是否成功消费(增强服务的稳定)。
- 管理员：对系统进行简单操作设置 和商家以及学生的基础信息进行查询和找回以及挂失处理等等服务。

3、登录界面：



A. 可实现账号密码的输入，可跳转到学生、商家、管理员等界面。

B. 可实现用户的注册、密码修改等功能。

i. 管理员界面：

- 可以看到所有使用该系统所有用户的基本信息
- 对受理系统有权限，可以受理学生和商户申请的相关服务

ii. 学生界面：

- 可以在该系统内进行账户密码的修改
- 可以办理一卡通挂失以及补办服务
- 可以通过该系统给一卡通进行充值服务

iii. 商家界面：

- 可以在校园一卡通系统中修改账户密码以及个人信息
- b.可以在系统中查看每月的流水明细
- c.商家可以在系统中绑定自己的银行卡，可实现提现金额以及上交校方租金

4、余额充值：

- 自定义充值金额，对剩余金额继续补充

A. 学生密码修改：输入现有旧密码，可对密码进行修改。密码没有特殊样式规则要求，根据喜好进行新密码的自定义。

B. 申请补办：发送请求，通过已有信息对以往信息进行找回处理。通过账户绑定内容发出找回密码等等请求，以待管理员处理详至业务。

C. 明细：查询消费情况。

D. 商家密码修改：对密码进行修改。

E. 绑定卡号：选定自己的卡号，通过信息的填写补充卡号信息。以卡号信息为核心，管理员可以检索相关其他用户(全部商家以及全部学生)的公开基本信息。

F. 使用者公开信息：公开自己的信息，防止不必要的情况。

G. 受理系统：处理客户(主要为学生)发送过来的请求，进行需求满足。

1.2 需求规格说明书

1、目的要求：

为校园学生生活提供便利。

2、项目背景：

未响应校园号召，增强学生编写代码的能力，丰富项目设计能力。

3、定义：

以学生信息办理基础账户消费业务。

4、任务概述：

A. 目标：设计一个简单的账户管理的系统，管理人员能够通过系统对学生账户信息进行管理，包括数据的添加、修改删除和浏览受理。

B. 一卡通系统提供 UI 界面，可以方便学生进行功能选择，实现信息的管理和查询，并可以清晰地显示相关信息。

C. 运行环境：MySQL、IDEA

二.　概要设计(主要完成人：全体成员)

2.1 系统功能模块

1、账户功能：

可实现登录功能、密码修改功能、注册功能以及密码找回功能。

2、用户功能：

A.　学生功能：可查询自身信息、密码的修改以及校园卡的补办找回申请

B.　商家功能：可查看自身信息、可查看自身账户流水、密码修改、绑定卡号等功能。

C.　管理员功能：可查看所有学生和商家的公开信息、可受理用户的需求。

2.2 各模块功能设计

1、登录界面：

可输入账号密码、可跳转下一页(1、密码相关界面：密码修改、密码找回、个人信息界面：学生界面、商家界面、管理员界面)。

2、学生界面：

查询个人信息、修改个人信息(账户密码)、余额充值。

3、商家界面：

查询个人信息、明细查询(可删记录)、绑定卡号、密码修改。

4、管理员界面：

查询个人信息、查询全体学生及所有商家信息、解决申请问题。

三.　详细设计(主要完成人：全体成员)

3.1 系统业务流程分析

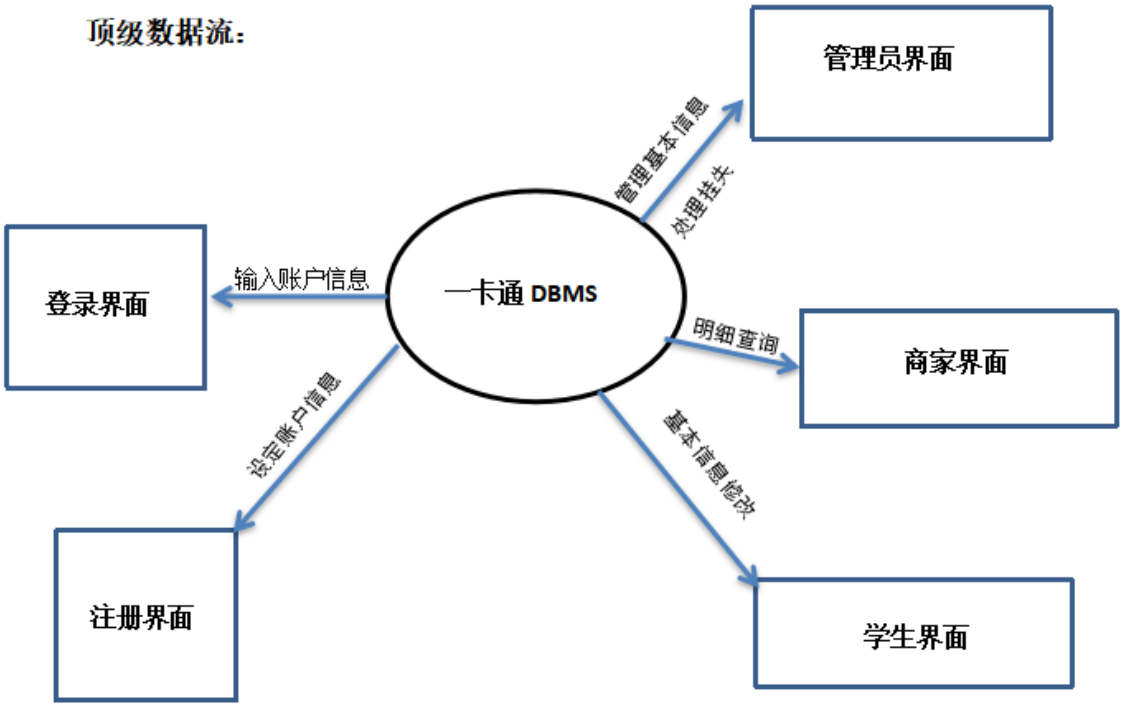


图 3.1 DBMS · DFD LV.TOP

- 一卡通充值系统 *DBMS*

A. 登录：使用账号密码进行登录以及信息的记录

B. 注册：注册账号密码以便进行下次登录以及上次登陆信息的二次查询

C. 商家：支持一卡通收费系统的商家人员和以及相关店铺

D. 学生：在校办理一卡通业务学习用户人员

E. 管理员：学校相关全员资本信息管理

3.2 各子模块的详细设计

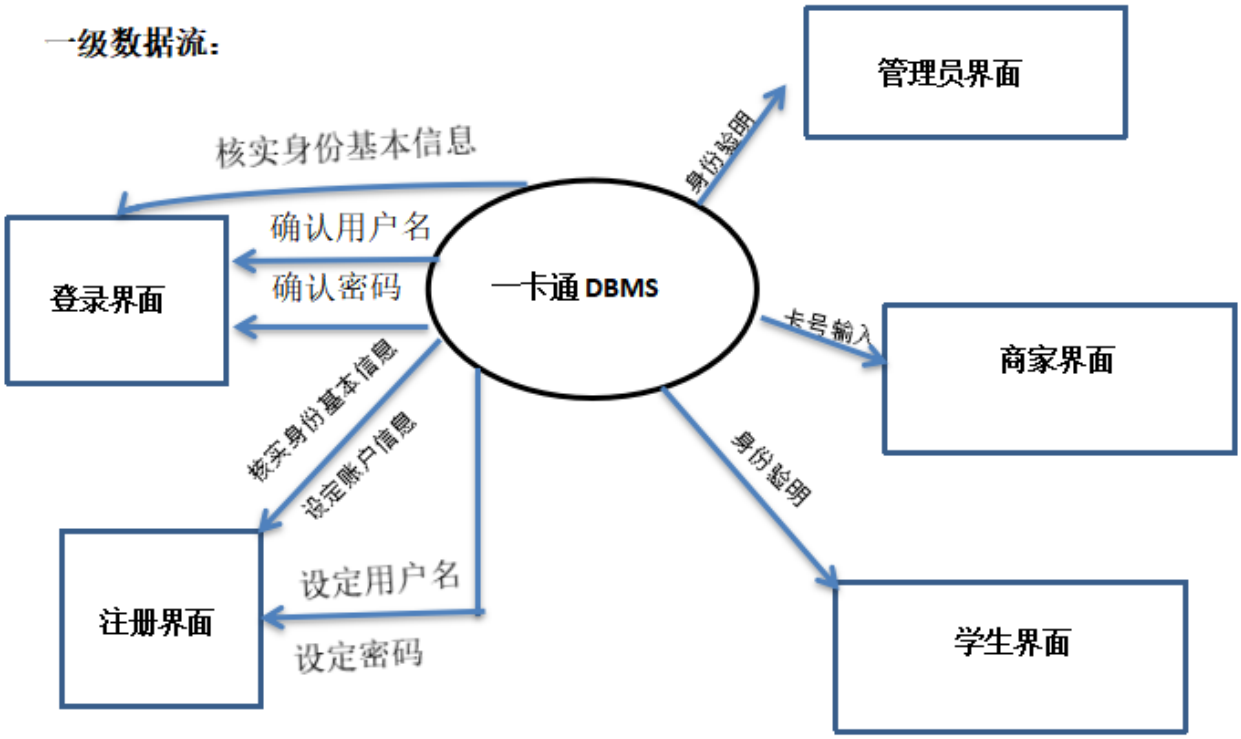


图 3.2 DBMS · DFD 一级数据流

➢ 一级数据流

A. 一卡通系统 DBMS 可通过登录界面核实身份基本信息，确认用户名，确认密码。

B. 一卡通系统 DBMS 内包含注册功能，用户通过注册界面可以设定用户名，设定密码。一卡通系统可以通过注册界面来收集核实用户身份信息，建立账户

信息。

C. 一卡通系统内还包含管理员、学生、商家等界面，可以实现管理员的身份验明，个人银行卡号输入，学生的身份验明等功能。

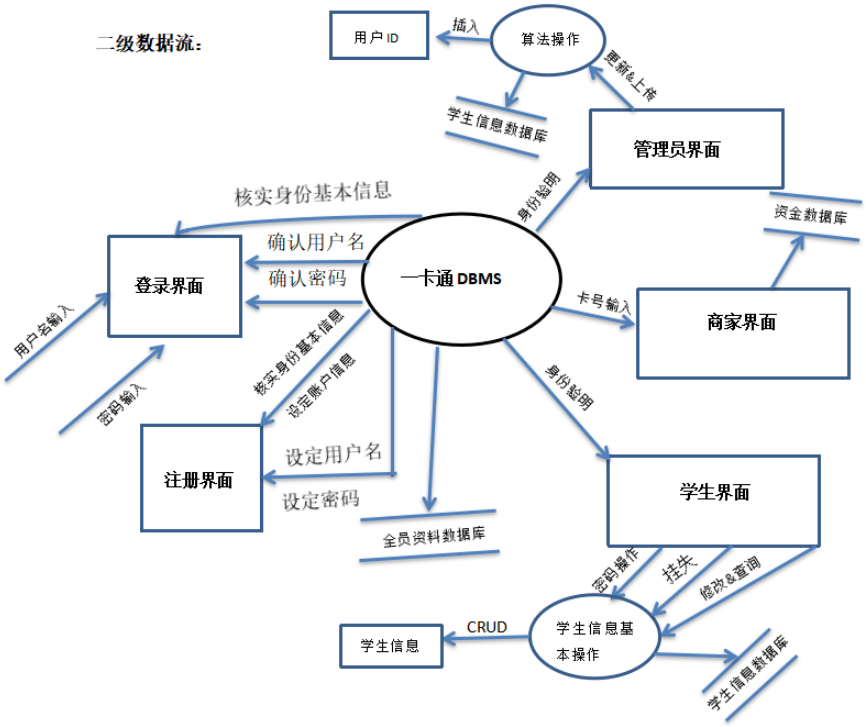


图 3.3 DBMS · DFD 二级数据流

➢ 二级数据流

A. 一卡通系统 DBMS 可以查询全员资料数据库。

B. 登录界面使用用户名登录并输入密码。

C. 通过算法操作实现插入用户 ID，以及连接学生信息数据库。

D. 商家界面连接资金数据库。

E. 学生界面有学生信息基本操作，如密码操作、挂失、修改和查询，并连接学生信息数据库。

四. 数据库设计与实现(主要完成人：全体成员)

4.1 概念模型设计(E-R 模型)

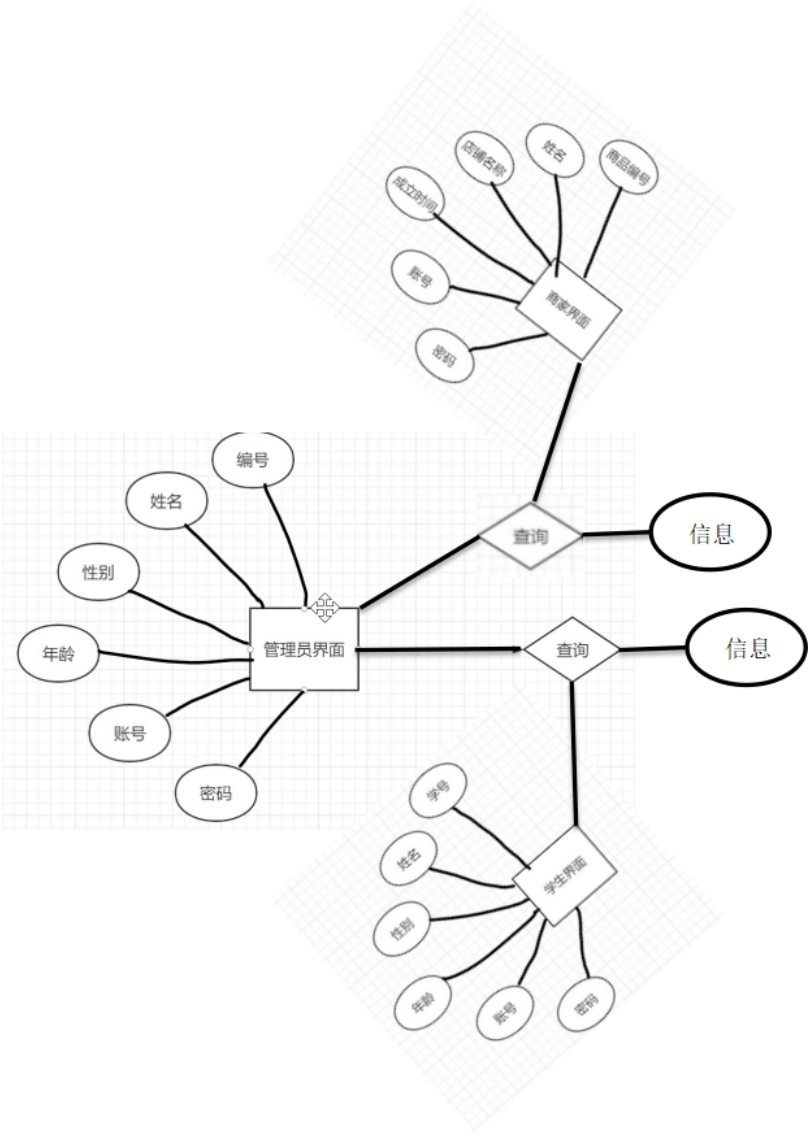


图 4.1 DBMS_ E-R MODEL

4.2 数据库逻辑结构设计

逻辑结构是独立于任何一种数据模型的信息结构。逻辑结构的任务是把概念结构设计阶段设计好的基本 E-R 图转化为宜选用的 DBMS 所支持的数据模型相符合的逻辑结构，并对其进行优化.

E-R 图向关系模型转化要解决的问题是如何将实体型和实体间的联系转化为关系模式，如何确定这些关系模式的属性

设计校园一卡通管理数据库，包括学生、一卡通、银行卡、消费账单四个关系，其关系模式中对每个实体定义的属性如下：

- 一卡通信息表

Card：(卡号，密码)

- 账单表

Bill：(账单号，时间，转入金额，支出金额，余额)

- 学生信息表

Student：(

学号、姓名、性别、年龄

所属班级、系别、年龄、账号

密码、手机、邮箱、余额

)

- 商家绑定信息表

Business：(

商品编号、姓名、商家主体、商家名

办理时间、开店时间、商家 ID 号、密码账号

手机、邮箱、余额

)

● 管理员查询处理信息表

Root：(编号,姓名,性别,年龄,账号,密码)

设计出 E-R 图后，可将 E-R 图转换为数据库模式

4.3 数据库物理结构实现(创建数据库过程)

运行本系统时在库中所建立的表代码分别如下：

```
CREATE TABLE stuM( # 学生表:学号 姓名 性别 年龄 账号 手机 邮箱 余额
    ID INT PRIMARY KEY AUTO_INCREMENT,
    password VARCHAR(100),
    name VARCHAR(100),
     num INT,
    sex char(1),
    age INT,
    tel INT,
    email VARCHAR(50),
    balance DOUBLE(10,2) default 0.00
```

```
    )
  COMMENT='stuM Table structure may be changed later.'
```

```
 CREATE TABLE busM( #  商家表:商品号  姓名  商家名  开店时间
账号  手机  邮箱  余额
    ID INT PRIMARY KEY AUTO_INCREMENT,
    password VARCHAR(100),
    name VARCHAR(100),
    num INT,
    brand VARCHAR(100),
    time INT,
    tel INT,
    email VARCHAR(50),
    balance DOUBLE(10,2) default 0.00
  )
  COMMENT='busM Table structure may be changed later.'
```

```
  CREATE TABLE rootM( #  管理表:账号名  密码
    ID INT PRIMARY KEY AUTO_INCREMENT,
    password VARCHAR(100),
    name VARCHAR(100)
            )
```

```
   CREATE TABLE stuLog(
     ID INT PRIMARY KEY AUTO_INCREMENT,
     Time DATETIME,
     CreateTime TIMESTAMP NOT NULL
DEFAULT CURRENT_TIMESTAMP,
     UpdateTime TIMESTAMP NOT NULL
DEFAULT CURRENT_TIMESTAMP
ON UPDATE CURRENT_TIMESTAMP,
     OpType VARCHAR
     )
   CREATE TABLE busLog(
     ID INT PRIMARY KEY AUTO_INCREMENT,
     Time DATETIME,
     CreateTime TIMESTAMP NOT NULL
DEFAULT CURRENT_TIMESTAMP,
     UpdateTime TIMESTAMP NOT NULL
DEFAULT CURRENT_TIMESTAMP
ON UPDATE CURRENT_TIMESTAMP
     )
   CREATE TABLE rootLog(
```

```
    ID INT PRIMARY KEY AUTO_INCREMENT,
    Time DATETIME,
    CreateTime TIMESTAMP NOT NULL
DEFAULT CURRENT_TIMESTAMP,
    UpdateTime TIMESTAMP NOT NULL
DEFAULT CURRENT_TIMESTAMP
ON UPDATE CURRENT_TIMESTAMP
    )
```

4.4 其他对象的创建(视图、存储过程、触发器...)

> 1. 视图

```
CREATE VIEW Sl_S
AS SELECT * FROM stuM; # 请用DESC查询
```

```
CREATE VIEW Sl_SP
AS SELECT password FROM busM
```

```
CREATE VIEW Sl_SB
AS SELECT balance FROM busM
```

```
CREATE VIEW Sl_B
AS SELECT * FROM busM
```

```
CREATE VIEW Sl_B_T
AS SELECT * FROM busM ORDER BY time DESC
```

```
CREATE VIEW Sl_BP
AS SELECT password FROM busM
```

```
CREATE VIEW Sl_BB
AS SELECT balance FROM busM
```

```
CREATE VIEW SI_R
AS SELECT DISTINCT * FROM rootM
```

```
CREATE VIEW SI_RN
AS SELECT name FROM rootM
```

2. 存储过程

CREATE PROC `rootM_insert`
AS `INSERT INTO rootM (password,name) VALUES ("root","UNDEFINED")`

3. 触发器

CREATE TRIGGER `rootM_after_insert`
AFTER INSERT ON rootM
FOR EACH ROW AS `INSERT INTO rootLog (time,OpType) VALUES (NOW(),"Insert")`

CREATE TRIGGER `stuM_before_update`
BEFORE UPDATE ON stuM
FOR EACH ROW AS `INSERT INTO stuLog (time,OpType) VALUES (NOW(),"Update")`

CREATE TRIGGER `stuM_before_delete`
BEFORE DELETE ON stuM
FOR EACH ROW AS `INSERT INTO stuLog (time,OpType) VALUES (NOW(),"Delete")`

CREATE TRIGGER `busM_before_update`
BEFORE UPDATE ON busM
FOR EACH ROW AS `INSERT INTO busLog (time,OpType) VALUES (NOW(),"Update")`

CREATE TRIGGER `busM_before_delete`
BEFORE DELETE ON busM
FOR EACH ROW AS `INSERT INTO busLog (time,OpType) VALUES (NOW(),"Delete")`