

Web高级编程大作业报告

题 目： 校园卡管理系统

专业名称：软件工程

班级学号： **15205236**

学生姓名： **曹金鹏**

2018 年 6 月24日

**目录**

**1.系统分析与设计 1**

1.1系统简介 1

1.2用例分析 1

1.2.1用例与角色 1

1.2.2系统用例图 2

1.2.3用例描述 3

1.3数据库设计 4

1.3.1 E-R图 4

1.3.2 E-R实体关系建模 4

1.3.3数据库设计 7

1.4系统配置以及代码组织结构 9

1.4.1系统配置 9

1.4.21代码组织结构 9

**2.本人工作 9**

2.1本人工作及本人工作说明 9

2.2前端的设计与框架 10

2.2.1 LOGO设计 10

2.2.2 页面风格设计 10

2.2.3 前端结构 11

2.3在线充值(增加余额并添加消费记录)功能的实现 14

2.2.1 界面设计 14

2.2.2 在线充值(增加余额并添加消费记录)组件及其工作过程 15

2.2.3 在线充值(增加余额并添加消费记录)实现 16

2.4查询充值记录功能的实现 21

2.2.1 界面设计 21

2.2.2 查询充值记录组件及其工作过程 21

2.2.3 查询充值记录实现 21

**3.系统测试 24**

3.1在线充值功能 24

3.2查询充值记录功能 24

**4.个人工作总结 25**

# 1.　系统分析与设计

## 1.1　系统简介

校园卡在学校内应用十分广泛、使用方便，它不仅仅可以用于借书、解除门禁，还可以应用于校园内的各个消费场所（如：超市（小卖部）、食堂、澡房、电子阅览室等）。校园卡完成校内各类消费。系统管理员登录时，会为其提供：开户、销户、挂失、学生信息查询、校园卡管理、充值记录查询、消费记录查询等功能；而学生用户登录时，会为其提供：查阅个人信息、消费记录查询、充值记录查询、余额查询等功能。

**1.2　用例分析**

### 1.2.1　用例与角色

系统功能的主要模块：校园卡信息管理，学生信息管理。

校园卡信息管理：

1. 开户销户信息管理：对校园卡的激活及销户操作。
2. 挂失/解挂失：对校园卡挂失及解挂失操作。
3. 消费信息管理：使用校园卡消费，转账，及将消费记录保存至消费记录。
4. 校园卡充值：对校园卡进行充值操作。

学生信息管理：

1. 学生信息添加修改删除管理：对学生信息进行添加修改删除操作。
2. 学生信息查找：对学生进行查找操作。

系统管理员：

* 添加用户
* 查找用户
* 修改用户信息
* 删除用户
* 校园卡注销
* 校园卡挂失/解挂失
* 查询校园卡信息
* 校园卡充值

学生用户：

* 查询个人信息
* 校园内充值
* 查询余额
* 查询消费记录

# 1.2.2 系统用例图

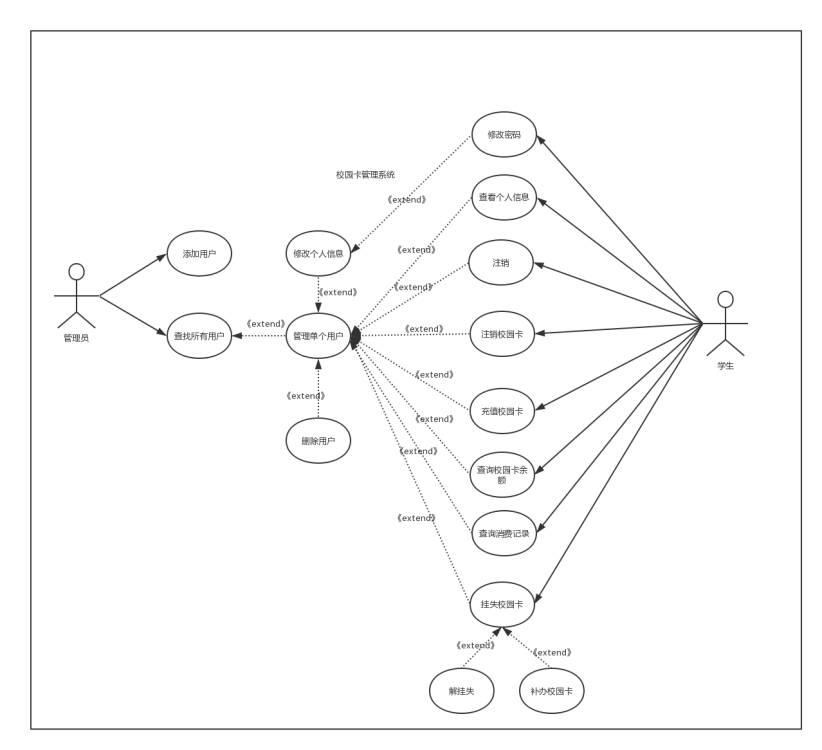


图1.2.3-1用例图

### 1.2.3　用例描述

学生个人信息查询用例

|  |  |
| --- | --- |
| 执行者: | 学生 |
| 前置条件: | 学生登陆成功。 |
| 基本路径: | 用户点击个人信息。  显示个人信息。 |
| 异常路径: | 登陆失败，提示“账号或密码错误！”。 |

学生查询余额用例

|  |  |
| --- | --- |
| 执行者: | 学生 |
| 前置条件: | 学生登陆成功，校园卡处于“正常使用”状态。 |
| 基本路径: | 学生点击查询余额。  判断校园卡状态为“正常使用”。  显示余额信息。 |
| 异常路径: | 登陆失败，提示“账号或密码错误！”。  判断校园卡处于“挂失状态或注销状态”，提示“校园卡异常！”。  校园卡不存在，提示“校园卡不存在！”。 |

学生查询消费记录用例

|  |  |
| --- | --- |
| 执行者: | 学生 |
| 前置条件: | 学生登陆成功，校园卡处于“正常使用”状态。 |
| 基本路径: | 学生点击查询消费记录。  判断校园卡状态为“正常使用”。  显示消费信息。 |
| 异常路径: | 登陆失败，提示“账号或密码错误！”。  判断校园卡处于“挂失状态或注销状态”，提示“校园卡异常！”。  校园卡不存在，提示“校园卡不存在！”。 |

学生挂失校园卡用例

|  |  |
| --- | --- |
| 执行者: | 学生 |
| 前置条件: | 学生登陆成功。 |
| 基本路径: | 学生点击“挂失校园卡”。  系统根据当前学生信息挂失其校园卡。修改校园卡的状态。  系统提示挂失成功。 |
| 异常路径: | 登陆失败，提示“账号或密码错误！”。  校园卡已经处于“挂失”状态，提示“挂失失败”。  校园卡不存在，提示“校园卡不存在！”。 |

学生解除挂失校园卡用例

|  |  |
| --- | --- |
| 执行者: | 学生 |
| 前置条件: | 学生登陆成功。 |
| 基本路径: | 学生点击“解除挂失校园卡”。  系统根据当前学生信息挂失其校园卡。修改校园卡的状态。  系统提示解除挂失成功。 |
| 异常路径: | 登陆失败，提示“账号或密码错误！”。  校园卡已经处于“正常”状态，提示“解除挂失失败”。  校园卡不存在，提示“校园卡不存在！”。 |

充值校园卡用例

|  |  |
| --- | --- |
| 执行者: | 学生 |
| 前置条件: | 学生登陆成功。 |
| 基本路径: | 用户点击“校园卡充值”。  系统提示充值金额。  系统根据用户信息增加其校园卡余额信息和消费记录。  系统提示“充值成功！”。 |
| 异常路径: | 登陆失败，提示“账号或密码错误！”。  校园卡处于“非正常使用”状态，提示“校园卡异常，充值失败！”。  校园卡不存在，提示“校园卡不存在！”。 |

注销校园卡用例

|  |  |
| --- | --- |
| 执行者: | 学生 |
| 前置条件: | 学生登陆成功。 |
| 基本路径: | 用户点击“注销校园卡”。  系统显示当前用户校园卡的信息，并提示“确定要注销该卡？”。  用户确认后，系统将此校园卡注销。  系统提示注销成功。 |
| 异常路径: | 登陆失败，提示“账号或密码错误！”。  校园卡不存在，提示“校园卡不存在！”。 |

添加学生信息用例

|  |  |
| --- | --- |
| 执行者: | 管理员 |
| 前置条件: | 管理员登陆成功。 |
| 基本路径: | 管理员点击“添加学生信息”。  系统添加学生信息。  提示添加成功。 |
| 异常路径: | 登陆失败，提示“账号或密码错误！”。  该学生已存在，提示“该学生已存在，不能重复添加”。 |

删除学生信息用例

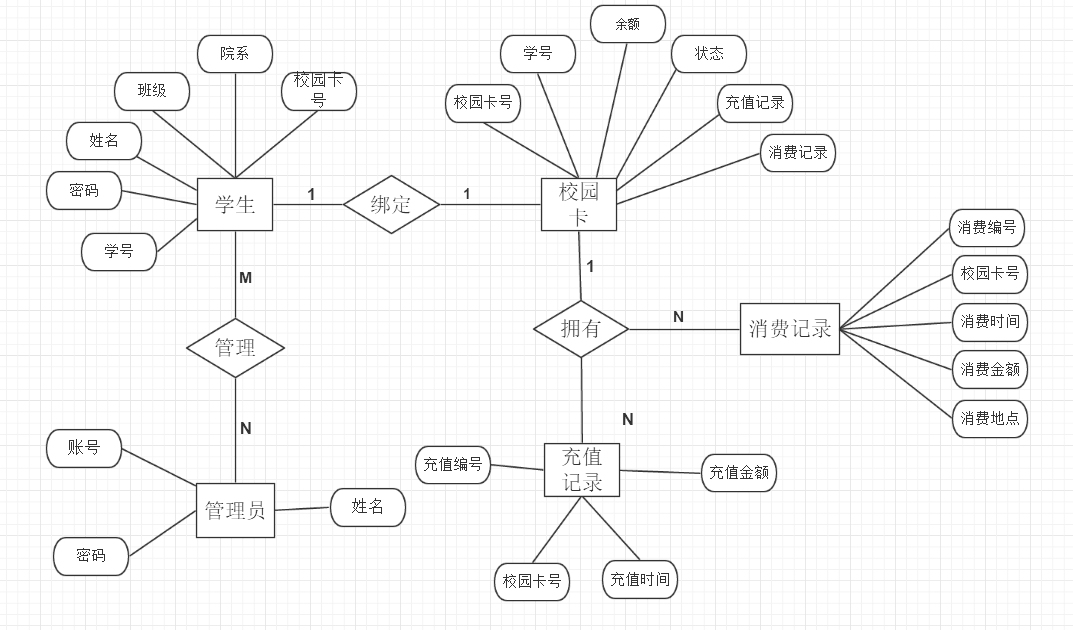
|  |  |
| --- | --- |
| 执行者: | 管理员 |
| 前置条件: | 管理员登陆成功。 |
| 基本路径: | 管理员点击“删除学生信息”。  系统删除学生信息。  提示删除成功。 |
| 异常路径: | 登陆失败，提示“账号或密码错误！”。  该学生不存在，提示“该学生不存在，请确认学生信息再操作”。 |

删除学生信息用例

|  |  |
| --- | --- |
| 执行者: | 管理员 |
| 前置条件: | 管理员登陆成功。 |
| 基本路径: | 管理员点击“修改学生信息”。  管理员对学生信息进行修改。  系统修改学生信息。  提示修改成功。 |
| 异常路径: | 登陆失败，提示“账号或密码错误！”。  该学生不存在，提示“该学生不存在，请确认学生信息再操作”。 |

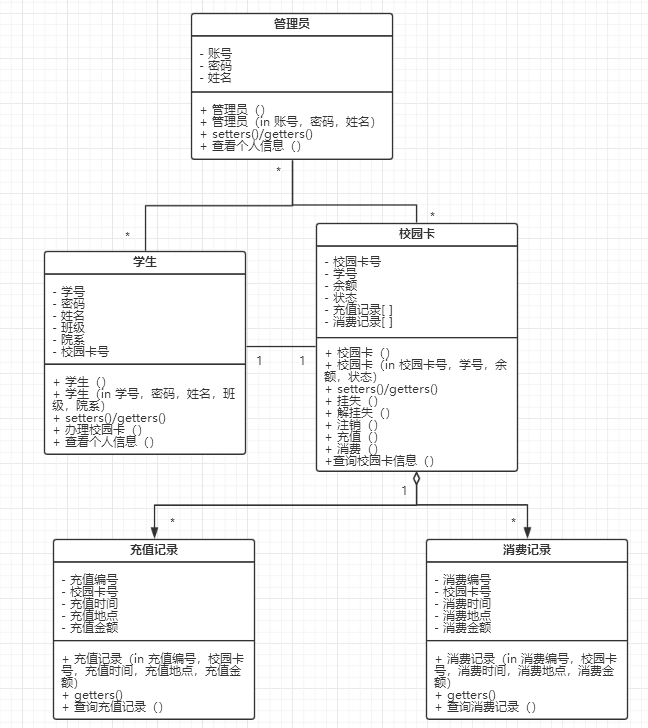
### 1.3数据库设计

### 1.3.1 E-R图



1.3.1-1校园卡管理系统E-R图

### 1.3.2 实体关系建模



1.3.2-1校园卡管理系统实体关系模型

### 1.3.3 数据库设计

Student表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| xuehao | number | 4 | 主键 | 学号 |
| name | char | 20 |  | 姓名 |
| password | varchar | 50 |  | 密码 |
| class | char | 20 |  | 班级 |
| yuanxi | char | 50 |  | 院系 |
| kahao | number | 10 | 外键 | 卡号 |

Card表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| kahao | number | 10 | 主键 | 卡号 |
| xuehao | number | 4 | 外键 | 学号 |
| yue | float | 50 |  | 余额 |
| type | char | 20 |  | 状态 |
| chonngzhijilu | char | 200 |  | 充值记录 |
| Xiaofeijilu | char | 200 |  | 消费记录 |

Manager表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| zhanghao | number | 10 | 主键 | 账号 |
| mname | char | 20 |  | 姓名 |
| mpassword | varchar | 50 |  | 密码 |

CharageRecord表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| bianhao | number | 10 | 主键 | 编号 |
| kahao | number | 10 | 外键 | 卡号 |
| date | DATE |  |  | 时间 |
| didian | varchar | 200 |  | 地点 |
| jine | float | 200 |  | 金额 |

ConsumptionRecord表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 约束 | 说明 |
| bianhao | number | 10 | 主键 | 编号 |
| kahao | number | 10 | 外键 | 卡号 |
| date | DATE |  |  | 时间 |
| didian | varchar | 200 |  | 地点 |
| jine | float | 200 |  | 金额 |

## 1.4 系统配置以及代码组织结构

**1.4.1系统配置**

技术栈：spring-mvc+bootstrap+jquery

开发环境：Eclipse+SQL Server+Tomcat

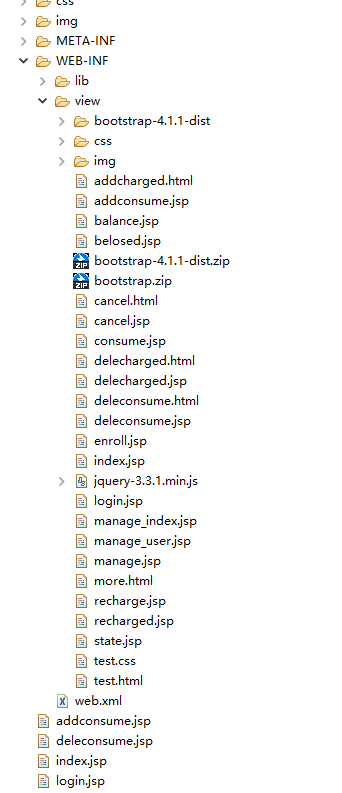
运行环境：Windows10

**1.4.2代码组织结构**

# P{6HA)WRC6K{T@ZU]S4LER8

1.4.2-1 Java Resources结构

1. com.spring.Controller中的类均为Controller（业务控制器），完成对核心控制器截获的请求进行处理。
2. com.spring.Dao中定义了接口。
3. com.spring.DaoImp中的类实现了com.spring.Dao中定义的接口，完成功能实现。
4. com.spring.Entities中定义了实体类。



1.4.2-2 WebContent结构

WebContent中的jsp界面用来显示给用户。

# 2.本人工作

## 2.1　小组分工以及本人工作说明

15205236 曹金鹏（组长，工作量40%）：

主要负责系统前端界面的设计，以及充值模块（查询充值、在线充值、充值统计）。

15205214 曹柳文（工作量30%）：

主要负责后端整体框架设计，以及消费模块（查询消费、更新消费、消费统计）。

15205233 蔡晓宇（工作量30%）：

负责系统数据库设计，以及登录注册、校园卡操作模块（校园卡挂失、注销）。

**2.2 前端的设计与框架**

**2.2.1 LOGO设计**



2.2.1 LOGO

在LOGO的设计中，我采用了本校校徽的三个飞行器三角作为主体，在外侧用线条组合成矩形来表示“校园卡”。在右侧，用较大、较粗的字体突出“校园卡”三个字，并用较小的字体说明本系统的具体功能“管理系统”。LOGO整体采用了绿紫渐变的颜色，突出LOGO的现代感。

**2.2.2 页面风格设计**



2.2.2-1 页面风格示例1



2.2.2-2 页面风格示例2

在页面实现过程中，我使用了CSS框架Bootstrap提供的样式。在页面头部，采用了暗色的导航条，配合绿紫渐变的LOGO，凸显科技感。使用浅灰色作为页面主体颜色，以区别功能模块，让页面层次更清晰。功能模块使用了Bootstrap提供的默认样式，以白色为主。

**2.2.3 前端结构**

用index.jsp和manage\_index.jsp编写导航栏，并使用iframe标签，通过JQuery改变其src属性来链接其他功能页面。在“注销校园卡”选项中加入了点击模态框，使用CSS3提供的功能来显示警告。具体代码如下，以index.jsp为例：

**index,jsp**

<!doctype html>

<%@ page language="java" contentType="text/html; charset=utf-8"%>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="bootstrap-4.1.1-dist/css/bootstrap.min.css" >

<!--引入外部css-->

<link rel="stylesheet" href="css/index.css" >

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="bootstrap-4.1.1-dist/js/jquery-3.3.1.min.js"></script>

<!---->

<script src="bootstrap-4.1.1-dist/js/bootstrap.min.js"></script>

<link rel="icon" href="img/logo1.png" type="image/x-icon" />

<link rel="shortcut icon" href="img/logo1.png" type="image/x-icon" />

<script>

$(function(){

var $iframe = $("#mainif");

//查询余额

$("#balance").click(function(){

$iframe.attr("src","balance");

});

// 查询消费

$("#consume").click(function(){

$iframe.attr("src","consume");

});

//查询充值

$("#recharged").click(function(){

$iframe.attr("src","recharged");

});

//在线充值

$("#recharge").click(function(){

$iframe.attr("src","recharge");

});

//挂失校园卡

$("#belosed").click(function(){

$iframe.attr("src","belosed");

});

//注销校园卡

$("#cancel").click(function(){

$iframe.attr("src","cancel");

});

//更多功能

$("#more").click(function(){

$iframe.attr("src","more");

});

//点击效果

var $Lis = $("#sel li");

$Lis.click(function(){

$Lis.removeClass("active");

$(this).addClass("active");

});

});

</script>

<title>校园卡管理系统-学生</title>

</head>

<body class="bg-light">

<!--头-->

<nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-top">

<a class="navbar-brand" href="#">

<img src="img/logo1.png" alt="Logo" style="width:70px;">

</a>

<ul id="sel" class="navbar-nav">

<li id="balance" class="nav-item active">

<a class="nav-link" href="#">查询余额</a>

</li>

<li id="consume" class="nav-item">

<a class="nav-link" href="#">查询消费</a>

</li>

<li id="recharged" class="nav-item">

<a class="nav-link" href="#">查询充值</a>

</li>

<li id="recharge" class="nav-item">

<a class="nav-link" href="#">在线充值</a>

</li>

<li id="belosed" class="nav-item">

<a class="nav-link" href="#">挂失校园卡</a>

</li>

<li id="cancel" class="nav-item">

<a class="nav-link" href="#" data-toggle="modal" data-target="#mymodal">注销校园卡</a>

</li>

<li id="more" class="nav-item">

<a class="nav-link disabled" href="#">更多功能</a>

</li>

</ul>

<div id="login" style="margin-left: 200px;">

<img style="border: 1px solid black;border-radius: 50%;background-color: white;margin-left: 16px" width="40px" height="40px" src="img/logo1.png" alt="">

<a style="margin-left: 16px" href="login">注销</a>

</div>

</nav>

<div id="maindrop" class="centerbox1">

<iframe id="mainif" src="balance" frameborder="0" scrolling="no" allowfullscreen="yes" style="width: 100%;height: 100%;display: block;"></iframe>

</div>

<!-- 模态框 -->

<div class="modal fade" id="mymodal">

<div class="modal-dialog modal-sm">

<div class="modal-content">

<div class="modal-header">

<h4 class="modal-title text-danger">警告</h4>

<button type="button" class="close" data-dismiss="modal">&times;</button>

</div>

<div class="modal-body">

<p>此操作确认后无法恢复，确认要注销吗？</p>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">已经了解</button>

</div>

</div><!-- /.modal-content -->

</div><!-- /.modal-dialog -->

</div><!-- /.modal -->

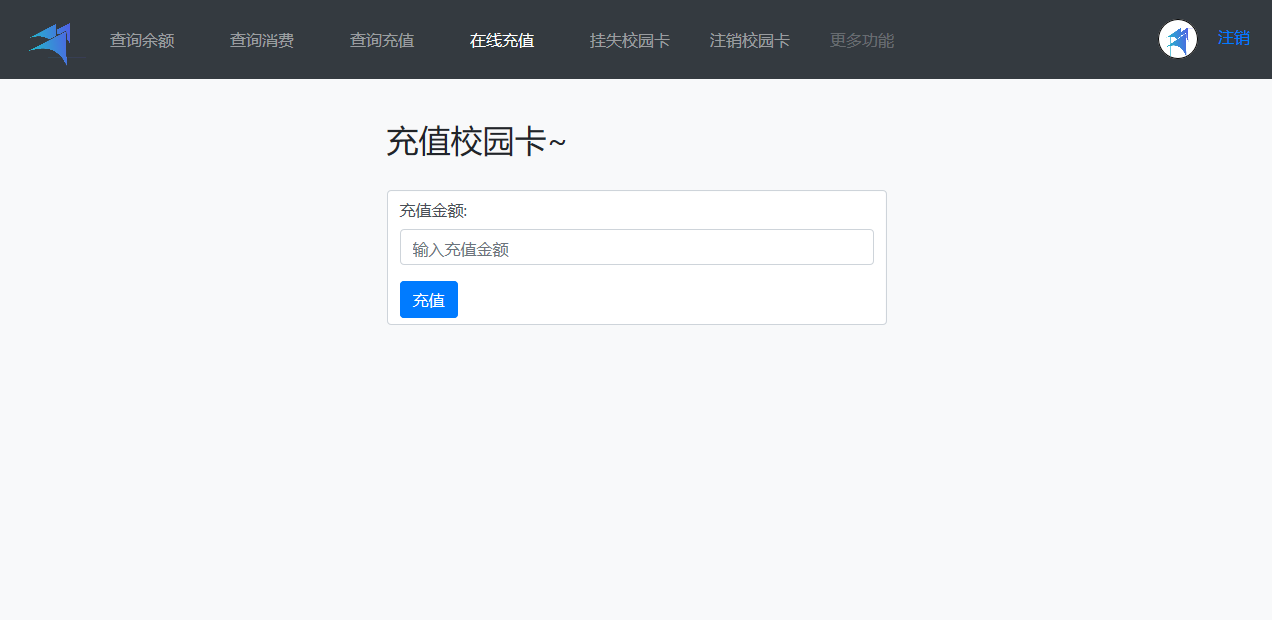
</body>

</html>

**2.3 在线充值(增加余额并添加消费记录)功能的实现**

**2.3.1 界面设计**

在线充值界面：



2.3.1-1 在线充值界面

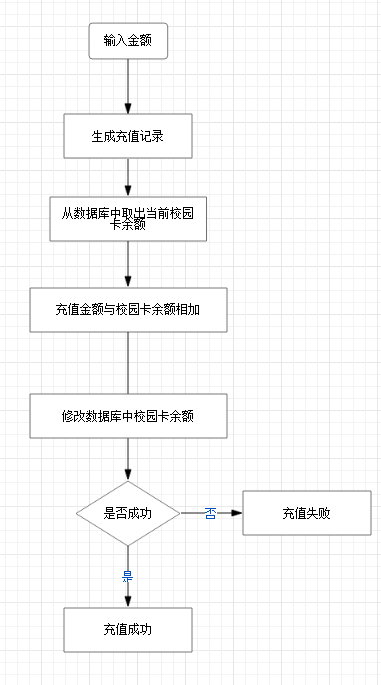
# 学生登陆成功后，点击在线充值，进入在线充值界面。

1. springDispatcherServlet核心控制器拦截到/recharge请求，调用RechargeController处理请求，完成在线充值操作。

# 学生输入充值金额，完成在线充值操作。

1. 再次查询个人信息，余额增加。

**2.3.2** **在线充值(增加余额并添加消费记录)组件及其工作过程**



2.3.2-1 充值模块组件及工作过程

**2.3.3 在线充值(增加余额并添加消费记录)实现**

**RechargeController.java**

package com.spring.Controller;

import java.sql.SQLException;

import java.text.SimpleDateFormat;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.servlet.ModelAndView;

import com.spring.Dao.AddRechargedRecordsDao;

import com.spring.Dao.RechargeDao;

import com.spring.Entities.RechargedRecords;

@Controller

public class RechargeController{

@Autowired

private AddRechargedRecordsDao AddRechargedRecordsDao;

@Autowired

private RechargeDao RechargeDao;

@RequestMapping(value="/recharge")

public void a(){

}

@RequestMapping(value="/addrecharge")

@ResponseBody

public ModelAndView recharge(String rechargedAmount,HttpSession session) throws SQLException{

int studentId =(int)session.getAttribute("studentId");

ModelAndView mv = new ModelAndView();

System.out.println(rechargedAmount);

RechargedRecords rechargedRecords = new RechargedRecords();

rechargedRecords.setStudentId(studentId);

rechargedRecords.setRechargedStyle("线上");

rechargedRecords.setRechargedAmount(rechargedAmount);

SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");

rechargedRecords.setRechargedTime(df.format(System.currentTimeMillis()));

AddRechargedRecordsDao.AddRechargedRecords(rechargedRecords);

RechargeDao.Recharge(rechargedRecords);

mv.setViewName("recharge");

mv.addObject("code", "1");

return mv;

}

}

**RechargeDao.java**

package com.spring.Dao;

import com.spring.Entities.RechargedRecords;

/\*\*

\* 来自 曹金鹏的骚操作

\* 操作时间：2018年6月20日下午8:29:13

\*/

public interface RechargeDao {

public void Recharge(RechargedRecords rechargedRecords);

}

**RechargeDaoImp.java**

package com.spring.DaoImp;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.stereotype.Repository;

import com.spring.Entities.RechargedRecords;

/\*\*

\* 来自 曹金鹏的骚操作

\* 操作时间：2018年6月20日下午8:31:56

\*/

@Repository("rechargeDao")

public class RechargeDaoImp implements com.spring.Dao.RechargeDao {

@Autowired

private JdbcTemplate jdbcTemplate;

@Override

public void Recharge(RechargedRecords rechargedRecords) {

// TODO Auto-generated method stub

String sql;

sql = "select b.balance from student as a,schoolCard as b where a.studentId=? and a.cardId=b.cardId";

//RowMapper<ConsumeRecords> rowMapper = new BeanPropertyRowMapper<>(ConsumeRecords.class);

String balance1 = jdbcTemplate.queryForObject(sql, new Object[]{rechargedRecords.getStudentId()}, java.lang.String.class);

float sum = Float.parseFloat(rechargedRecords.getRechargedAmount()) + Float.parseFloat(balance1);

String sql1;

sql1 = "UPDATE schoolCard SET balance = ? from schoolCard,student WHERE student.cardId=schoolCard.cardId and student.studentId = ? ";

jdbcTemplate.update(sql1,new Object[]{sum,rechargedRecords.getStudentId()});

}

}

**AddRechargedRecordsDao.java**

package com.spring.Dao;

import com.spring.Entities.RechargedRecords;

/\*\*

\* 来自 曹金鹏的骚操作

\* 操作时间：2018年6月18日下午5:31:23

\*/

public interface AddRechargedRecordsDao {

public void AddRechargedRecords(RechargedRecords rechargedRecords);

}

**AddRechargedRecordsImp.java**

package com.spring.DaoImp;

import java.sql.Types;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.stereotype.Repository;

import com.spring.Entities.RechargedRecords;

/\*\*

\* 来自 曹金鹏的骚操作

\* 操作时间：2018年6月18日下午5:34:31

\*/

@Repository("addRechargedRecordsDao")

public class AddRechargedRecordsImp implements com.spring.Dao.AddRechargedRecordsDao{

@Autowired

private JdbcTemplate jdbcTemplate;

public void AddRechargedRecords(RechargedRecords rechargedRecords){

String sql = "insert into rechargedRecordsTable(studentId,rechargedStyle,rechargedTime,rechargedAmount) values(?,?,?,?)";

jdbcTemplate.update(sql,

new Object[] {rechargedRecords.getStudentId(),

rechargedRecords.getRechargedStyle(),

rechargedRecords.getRechargedTime(),

rechargedRecords.getRechargedAmount()},

new int[]{ Types.VARCHAR, Types.VARCHAR, Types.TIMESTAMP, Types.NCHAR });

System.out.println("aaaaa");

}

}

**2.4 查询充值记录功能的实现**

**2.4.1 界面设计**

查询充值记录界面**：**

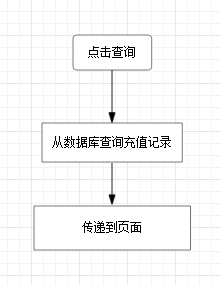


2.4.1-1 查询充值记录界面

# 学生登陆成功后，点击查询充值，进入查询充值记录界面。

1. springDispatcherServlet核心控制器拦截到/recharged请求，调用RechargedController处理请求，完成查询充值记录操作。

**2.4.2 查询充值记录组件及其工作过程**



2.4.2-1 查询充值记录组件及工作过程

**2.4.3 查询充值记录实现**

# RechargedController.java

package com.spring.Controller;

import java.sql.SQLException;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.servlet.ModelAndView;

import com.spring.Dao.SearchRechargedRecordsDao;

import com.spring.Entities.RechargedRecords;

@Controller

public class RechargedController{

@Autowired

private SearchRechargedRecordsDao SearchRechargedRecordsDao;

@RequestMapping(value="/recharged")

public ModelAndView recharged(HttpServletRequest request,HttpSession session) throws SQLException{

int studentId =Integer.parseInt((String)session.getAttribute("studentId"));

ArrayList<RechargedRecords> rechargedInfo = SearchRechargedRecordsDao.AllRechargedRecords(studentId);

ModelAndView mv = new ModelAndView();

mv.setViewName("recharged");

mv.addObject("rechargedInfo", rechargedInfo);

return mv;

}

}

**SearchRechargedRecordsDao.java**

package com.spring.Dao;

import java.util.ArrayList;

import com.spring.Entities.RechargedRecords;

/\*\*

\* 来自 曹金鹏的骚操作

\* 操作时间：2018年6月18日下午4:43:31

\*/

public interface SearchRechargedRecordsDao {

public ArrayList<RechargedRecords> AllRechargedRecords(int studentId);

}

**SearchRechargedRecordsImp.java**

/\*\*

\*

\*/

package com.spring.DaoImp;

import java.util.ArrayList;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.jdbc.core.BeanPropertyRowMapper;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.stereotype.Repository;

import com.spring.Entities.RechargedRecords;

/\*\*

\* 来自 曹金鹏的骚操作

\* 操作时间：2018年6月18日下午4:51:31

\*/

@Repository("searchRechargedRecordsDao")

public class SearchRechargedRecordsImp implements com.spring.Dao.SearchRechargedRecordsDao{

@Autowired

private JdbcTemplate jdbcTemplate;

@Override

public ArrayList<RechargedRecords> AllRechargedRecords(int studentId) {

String sql;

sql = "select rechargedId, rechargedStyle, rechargedTime, rechargedAmount from rechargedRecordsTable where studentId=?";

ArrayList<RechargedRecords> rechargedRecords1 =(ArrayList<RechargedRecords>) jdbcTemplate.query(sql, new BeanPropertyRowMapper<RechargedRecords>(RechargedRecords.class),studentId);

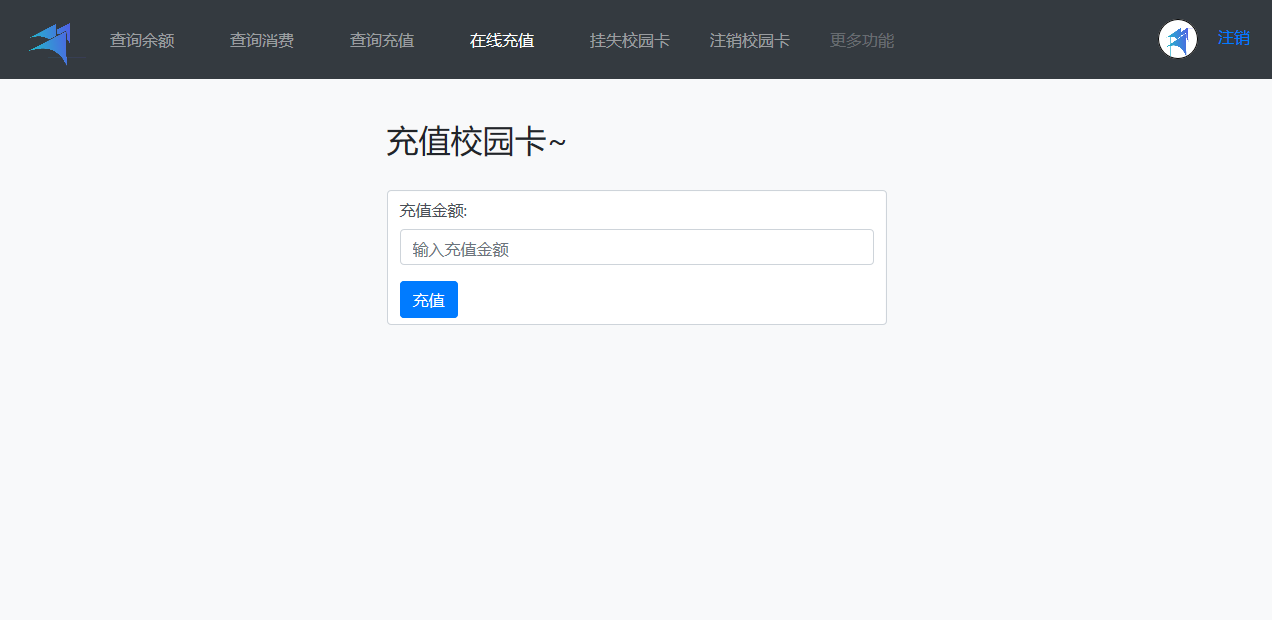
return rechargedRecords1;

}

}

# 3 .系统测试

**3.1在线充值功能**



3.1-1 在线充值界面



3.1-2 充值成功界面 （余额增加10）

**3.2查询充值记录功能**



3.2-1 查询充值记录界面

# 4.个人工作总结

在此次项目中，使用了CSS框架Bootstrap，清晰的感受到它的优势。

1. 扩展性强，能够更好的与现实的web开发项目结合。
2. 该框架不断适应web技术的发展。
3. 比较成熟，在大量的项目中充分的使用和测试。
4. 拥有完善的文档，使用起来更方便。
5. Bootstrap提供了非常豐富的组件与插件，组件包含小图标、按钮组、菜单导航、标签页等等，并接受定制。
6. 扩展性强，兼容各种脚本插件。
7. 拥有现成UI组件，可以快速搭建网页页面。
8. 前端处理简洁，在Blog系统搭建过程中使用方便。

也感受到了JS框架Jquery的设计理念：用更少的代码实现更多的功能。

使用了spring-mvc，也清晰的感受到了它的优势。

但是在项目本身的开发中还是遇到了许多的的问题，如组员数据库字段未能统一、对JDBCTemplet不熟悉等。但最后都一一解决。