# 测试计划

## 1．引言

### 1.1编写目的

   测试计划是描述将要进行测试活动的范围、方法、资源和时间进度的文档。它确定了测试项，要测试的特征、测试任务，谁来完成每项任务以及需要制定应急方案的所有风险等。

本文档的读者是系统编程人员和系统测试人员。

### 1.2背景

A、待开发软件名称：课程作业管理系统

B、项目提出者：米新江教授

   开发者：张凯超

   用户：廊坊师范学院全体教职工和学生

   实现该软件的计算中心或计算机网络：廊坊师范学院局域网

C、该软件系统同其他机构的基本的相互来往关系：由廊坊师范学院信息技术提高班做技术支持。

### 1.3定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 注册 | 消费金额 | 充值 | 值班 | 退卡 |
| Login | Consume | Recharge | On work | Cancel |
| 结账 | 基本数据 | 购卡 | 上机 | 下机 |
| Statement | Basic\_date | Buycard | Login | logout |

### 1.4参考资料

[1]《软件工程事务》刘学俊李继芳 刘汉中 编著  浙江大学出版社

[2]测试计划（GB8567——88）

## 2．计划

### 2.1测试接口计划

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口 | 功能 | 输入 | 输出 | 说明 |
| 查询接口（Search  Function  Service） | 根据抽象作业中的属性：task\_number、task\_title、course\_number、t\_number、class\_number返回所有相关抽象作业（Abstract\_task）的所有信息  getAbstractTasksByTask(Abstract\_task task, String sortOrder,Set<String> sortParams) | Abstract\_task t task=new Abstract\_task();  t task.task\_number=”1001”  String sortOrder=””; | 获取所有相关抽象作业（Abstract\_task）的所有信息 | Search  Function  Service接口用于实现该作业管理的查询功能，实现的功能有：教师查询学生成绩、作业提交情况、学生信息、查询班级情况，学生查询作业、查询作业成绩、查询课程的作业等等 |
|  | 根据作业号返回单个线下型作业的所有信息  getDelineTasksByTask\_number(String taskNumber) | String taskNumber ="1001"; | 获取到线下型作业号1001的所有信息 |
|  | 根据作业号返回单个在线型作业的所有信息  getOnlineTasksByTask\_number(String taskNumber) | String taskNumber ="1002" | 获取到在线型作业号1002的所有信息 |
|  | 根据作业号和学号返回该学生提交的线下型（文件型）作业的信息  getOneStudentCompletion\_DelineTask\_  ByTask\_numberAndS\_number(String task\_number, String s\_number) | String task\_number ="1001";  String task\_number ="1001"; | 获取到作业号1001和学号1001学生的线下作业类型信息 |
|  | 根据作业号返回单个在线型作业的所有信息  getOneStudentCompletion\_OnlineTask\_  ByTask\_numberAndS\_number(String task\_number,String s\_number) | String task\_number ="1002"; | 获取在线作业号1002的作业信息 |
|  | 根据班级的班级编号或班级名称获取本班所有学生的数据  getStudentsByClass(Student\_class studentClass) | setClass\_name("软件工程");  setClass\_number("1001"); | 正确获取本班所有学生的数据 |
|  | 根据班级的课程编号或课程名称获取本课程所有学生的数据  getStudentsByCourse(Course course); | Course c=new Course();  c.course\_name="html";  c.course\_number="1001"; | 正确获取本课程所有学生的数据 |
|  | 根据作业号返回单个学生的信息  getStydentsByTask\_number(String taskNumber) | String task="1001" | 获取到作业号1001的学生信息 |
|  | 根据作业号获取改作业的所属课程的信息  getOtherMessageByTask\_number(String taskNumber); | String task="1001" | 输出1001的所属作业课程信息 |
|  | 根据学生学号获取作业完成后，教师的给的分数和评语  getStudentScoreAndCommentByS\_number(); | String studentNumber=“1001” | 根据学生学号1001获取该学生所完成作业的分数 |

### 2.2测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识符 | 进度安排 | 内容和目的 |
| 基本信息测试 | Basicdata\_Info | 1天 | 正确设定基本信息 |
| 用户信息 | User\_info | 1天 | 添加删除用户 |
| 学生信息 | Student\_info | 3天 | 保证学生正确注册、充值、上机、退卡等操作 |
| Online\_Info |
| Recharge |
| Cancle |
| 结账 | Checkday | 2天 | 能够根据日期正确进行日结账和周结账操作 |
| Checkweek |

### 2.3基本信息测试（Basicdata\_Info）

     主要测试系统管理员对基本数据的设定是否合理，符合实际逻辑。

#### 2.3.1进度安排

对操作人员进行1个小时的培训，了解正常的基本数据设定标，保证系统能够修改和执行基本数据的设定工作。

#### 2.3.2条件

陈述本项测试工作对资源的要求，包括：

1. 至少使用2台电脑，大约测试1个小时。

2. 以用户对基本数据的要求为标准，正确设定基本数据。

#### 2.3.3测试资料

列出本项测试所需的资料

暂无

#### 2.3.4测试培训

对操作人员进行1个小时的培训，了解正常的基本数据设定标，保证系统能够修改和执行基本数据的设定工作。

### 2.4用户信息测试（User\_Info）

#### 2.4.1进度安排

由于本系统的主要管理用户为学生，这一部分主要设计到对学生信息的管理工作，包括注册、充值、上机和退卡等工作。这一部分要测试的内容相对较多，耗时较长。大概需要3天的测试时间。

#### 2.4. 2条件

陈述本项测试工作对资源的要求，包括：

至少需要两台电脑，一个刷卡机。

至少需要两个测试人员，一个刷卡上机，另一个作为操作员实现对学生的注册、充值、上机、退卡等工作。

#### 2.4.3测试资料

暂无

#### 2.4.4测试培训

两个测试人员要熟悉不同种类的测试方法，选取正确的测试用例对学生注册窗口的测试。另外，要采取恰当的测试方法实现对注册、充值和退卡的操作，保证操作的合理性。

### 2.5结账（Check）

#### 2.5.1进度安排

    这一部分主要是管理人员的任务部分，由于涉及到用户单位的利益关系，所以比较关键。大概需要两天的时间完成该部分的测试工作。

#### 2.5.2条件

陈述本项测试工作对资源的要求，包括：

a.至少使用3台电脑，3个测试人员

b.测试人员要熟悉系统的基本运行流程和各个模块的功能。

#### 2.5.3测试资料

暂无

#### 2.5.4测试培训

测试人员要熟悉系统的基本运行流程和各个模块的功能。一个刷卡上机、另一个执行操作员的充值、退卡工作，最后一个测试管理员的结账功能。

## 3．测试设计说明

### 3.1基本信息测试（Basicdata\_Info）

#### 3.1.1控制

人工设定基本系统基本信息。

#### 3.1.2输入

输入基本信息要求的半小时基本费用、递增时间、最少上机时间、上机准备时间、卡内最少余额。

#### 3.1.3输出

暂无

#### 3.1.4过程

使用管理员身份登录该系统后，就拥有了该系统的基本信息设定的权限，根据实际情况设定完基本数据后，检验是否能够修改和正常保存即可。

### 3.2用户信息测试（User\_Info）

#### 3.2.1控制

采用刷卡器和人工输入同时进行的方式。

#### 3.2.2输入

学生上机时刷卡上机，对学生进行注册、充值、退卡时手工输入卡号。

#### 3.2.3输出

学生上机输出学生的相关基本信息，以及相对应的时间。对学生进行注册充值时，如果成功，系统会弹出提示框，表明注册、充值或退卡的卡号、姓名、充值/退卡金额，操作教师等。

#### 3.2.4过程

先对学生完成注册测试，然后完成充值测试或者上机测试，最后完成退卡测试。

### 3.3结账（Check）

#### 3.3.1控制

主要有管理员进行该项功能，测试时需要手动点击鼠标进行结账及相关查询。

#### 3.3.2输入

主要输入的是查询日期的范围，如果是日结账则只需要点击日结账单中的结账即可，如果是周结账则需要选择要查询的日期进行相关操作。

#### 3.3.3输出

日结账单和周结账单。

#### 3.3.4过程

 首先保证本系统在投入运行，有学生上机及操作人员的相关操作，最后测试管理人员的结账模块。

## 4．评价准则

### 4.1范围

所选择的测试用例可以基本覆盖本系统的功能，其局限性在于不能覆盖所有的数据，不能测试多台电脑同时访问数据库的情况。

### 4.2数据整理

在整理数据方面系统中引用了报表控件，在日结账单和周结账单模块使用到报表一方面对结账情况的汇总和打印。

### 4.3尺度

1.要求关于学生充值、上机、下机、退卡等涉及到金额的部分精确度不低于0.1元。

2.在涉及到系统操作反应时间方面不超过2s.

3.遇到在测试过程中中断或停机的状况，需找出问题所在，保证系统正常工作的运行，否则系统测试不通过。