长 沙 学 院

软件工程基础实训Ⅱ 实训

|  |  |
| --- | --- |
| **题目** | **基于C/S架构的餐饮管理系统** |
| **学院** | **计算机科学与工程**  **请注意**：  蓝色字体内容为示例或解释，仅供参考。 |
| **专业(班级)** | **软件工程（23软件0x）** |
| **姓名** |  |
| **学号** |  |
| **指导教师** |  |
| **起止日期** | **2025.06.09～2025.06.20** |

实训任务书

**课程名称：软件工程基础实训II**

**实训题目：基于C/S架构的餐饮管理系统**

### 已知技术参数和设计要求：

1. 问题描述（功能要求）：

本次案例将完成一个基于C/S架构的餐饮管理系统。

（1）.进行餐饮管理系统的功能设计。

（2）.设计餐饮管理系统的数据库。

（3）.利用JAVA的GUI类设计开发友好的人机接口。

（4）.利用JAVA语言设计监听用户操作和JDBC进行数据库的处理。

餐饮管理系统主要有6大模块：

（1）.员工管理：对员工实现增删改查。

（2）.客户管理：对客户实现增删改查。

（3）.餐台管理：对餐台实现增删改查。

（4）.菜品管理：对菜品分类、菜品实现增删改查。

（5）.点菜管理：服务员对某客户、某一空闲餐台实行开台，同时实现点菜，将餐台号与所点的菜品对应起来，分别显示出来，并记录开台时间。

（6）.结账管理：收银员对某一餐台通过统计显示消费的菜品清单统计出消费金额，通过手动输入实收金额进行找零的计算，并显示，完成结账的操作，并记录成统计数据。

基础实训II有2周时间40学时，学生基本必做任务：独立完成基于C/S架构的餐饮管理系统。

选做扩展：菜品推荐、销售统计。

**2. 运行环境要求：**

（1）客户端：

windows操作系统，JDK1.6，Idea或Eclipse。

（2）服务器：

MySQL5.6或 SQL Server 数据库服务器。

3. 技术要求：

1） 掌握软件工程的需求分析和系统设计方法。

2) 掌握数据库的分析与设计，完成餐饮管理系统数据库的设计。

3) 掌握JAVA的GUI设计方法，完成登录、主界面、菜品分类管理界面、菜品管理界面等人机接口。

4） 掌握JAVA面向对象程序设计方法，设计系统所需的各种类。

5) 掌握JDBC方法，对数据库进行操作，完成主程序和菜品分类、菜品管理等功能。

### 实训工作量：

40课时

### 工作计划：

1.班级

23软件1-4

2.课时及教室安排

见实训课程工作计划表。

实训报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 |  | 学号 |  | 班级 |  |
| 专业 | 软件工程 | | 指导教师姓名 |  | |
| 项目名称 | 基于C/S架构的餐饮管理系统 | | | | |
| **实训过程**：（说明实训各阶段的工作过程，包括需求分析、系统设计、编码实现、系统测试、交付实施等阶段的主要工作。）  本栏正文排版要求：宋体5号，行距18磅，段首缩进2字符。  本栏可跨多页。 | | | | | |
| **实训成果**：  参见基于C/S架构的餐饮管理系统设计说明书。 | | | | | |
| **实训总结**：（包括心得体会、存在的问题和改进方向。）  本栏正文排版要求：宋体5号，行距18磅，段首缩进2字符。  本栏可跨多页。 | | | | | |

基于C/S架构的餐饮管理系统

设计说明书

作者：xxx(学号)

计算机科学与工程学院

2025年 06 月 20日

**摘要**

摘要应概括反映出课程设计的内容、方法、成果和结论。结果和结论性字句是摘要的重点，在文字论述上要多些，以加深读者的印象；用精炼、概括的语言来表达，每项内容不宜展开论证或说明，要客观陈述，不宜加主观评价；要独立成文，选词用语要避免与全文尤其是前言和结论部分雷同；摘要中不宜使用公式、图表，不标注引用文献编号。避免将摘要写成目录式的内容介绍。

关键词是供检索用的主题词条，应采用能覆盖全文主要内容的通用技术词条(参照教材附录A的名词索引或其它技术术语标准)。关键词一般列3~5个，按词条的外延层次排列（外延大的在前面）。

**关键词：**java ，C/S架构，餐饮管理系统

目录

[1 引言 1](#_Toc11882504)

[1.1 编写目的 1](#_Toc11882505)

[1.2 参考资料 1](#_Toc11882506)

[2 需求规约 2](#_Toc11882507)

[2.1 功能需求 2](#_Toc11882508)

[2.2 界面需求 3](#_Toc11882509)

[2.3 数据需求 3](#_Toc11882510)

[3 系统设计 4](#_Toc11882511)

[3.1 运行环境 4](#_Toc11882512)

[3.2 系统静态结构设计 4](#_Toc11882513)

[3.3 人机接口设计 5](#_Toc11882514)

[3.4 XXX模块设计说明 7](#_Toc11882515)

[3.5 XXX模块设计说明 7](#_Toc11882516)

[4 数据库设计 8](#_Toc11882517)

[4.1 数据库环境说明 8](#_Toc11882518)

[4.2 数据库的命名规则 8](#_Toc11882519)

[4.3 逻辑结构设计 8](#_Toc11882520)

[4.4 物理结构设计 9](#_Toc11882521)

[4.5 安全性设计 10](#_Toc11882522)

[5 测试用例设计 12](#_Toc11882523)

[附录 14](#_Toc11882524)

[附录1 程序运行结果 14](#_Toc11882525)

[附录2 程序源代码 15](#_Toc11882526)

# 1 引言

## 1.1 编写目的

本文档是“xxx”的软件设计说明书。编写目的是：明确软件需求、概要设计、详细设计和功能测试用例，为软件后续开发和维护提供指导。本文档的读者对象是需求分析人员、系统设计人员、系统开发人员、系统测试人员。

## 1.2 参考资料

列出有关资料的名称、作者、文件编号或版本等。

[1] 文东. 数据库系统开发[M]. 北京: 北京科海电子出版社，2011．

[2]

参考资料格式要求：

[1] 袁庆龙，候文义．Ni-P合金镀层组织形貌及显微硬度研究［Ｊ］．太原理工大学学报，2001，32(1)：51-53.（连续出版物：［序号］ 主要责任者．文献题名［J］．刊名，出版年份，卷号(期号)：起止页码）

[2] 刘国钧，王连成．图书馆史研究［Ｍ］．北京：高等教育出版社，1979．（专著：［序号］ 主要责任者．文献题名［Ｍ］．出版地：出版者，出版年．）

[3] 孙品一．高校学报编辑工作现代化特征［Ｃ］．中国高等学校自然科学学报研究会．科技编辑学论文集(2)．北京：北京师范大学出版社，1998：10-22．（论文集：[序号］ 主要责任者．文献题名[Ｃ]∥主编．论文集名．出版地：出版者，出版年：起止页码．）

[4] 张和生．地质力学系统理论［Ｄ］．太原：太原理工大学，1998．(学位论文:［序号］ 主要责任．文献题名［Ｄ］．保存地：保存单位，年份．)

[5] 冯西桥．核反应堆压力容器的LBB分析［Ｒ］.北京：清华大学核能技术设计研究院，1997．(报告:［序号］ 主要责任．文献题名［Ｒ］．报告地：报告会主办单位，年份．)

# 2 需求规约

## 2.1 功能需求

提示：详细描述程序的功能需求，分析功能需求，写出每个功能的说明。画出功能模块图，如图2.1所示。

（**图名： 仿宋，小四。**

**图内文字：宋体，五号。）**

图2.1 图书管理系统功能模块图

2.1.1 借书登记用例说明

用例名称：借书登记。

用例描述：图书管理员对读者借阅的图书进行登记。读者借阅图书的数量不能超过规定的数量。如果读者有过期未还的图书，不能借阅新图书。

前置条件：读者请求借阅登记。

后置条件：读者取得借阅的图书。

活动步骤：

1. 读者请求借阅图书。
2. 系统检查读者的状态。
3. 系统检查图书的状态。
4. 系统标记图书为借出状态。
5. 读者获取图书。

扩展点：

1. 图书续借，执行续借用例。

异常处理：

1. 如果用户借阅数量超过规定数量，或者有逾期未还的图书，则用例终止。
2. 如果借阅的图书不存在，则用例终止。

2.1.2 还书登记用例说明

[解释] 要逐一说明每个基本用例。其中活动步骤（流程）、扩展点、异常处理是重点描述内容。对于扩展用例和包含用例可以不单独列出，也可以单独列出。

## 2.2 界面需求

提示：说明用户对交互界面的要求。

## 2.3 数据需求

提示：说明程序需处理的数据实体及其属性，数据名称，数据格式，数据结构，数据取值范围，数据存储要求，数据处理过程。例如，用户实体

学生实体属性有学号，姓名,…。表格说明范例如表2.1所示。

表2.1 用户实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| 姓名 | 字符串 | 32 | 否 | 否 | 用户名 |
|  |  |  |  |  |  |

# 3 系统设计

这里陈述总体设计思路。说明系统需求中的每个功能实现思路。

## 3.1 运行环境

简要说明对本系统的运行环境（包括硬件环境和支持软件环境）的规定。

## 3.2 系统静态结构设计

采用类图方式给出系统静态结构设计，说明本系统的各软件类的划分。类图要求详细描述，必须有属性和操作，不能仅仅是类名。如果一张图画不下，可以分模块叙述，也可以先画类的简图，在简图之后逐个画出每个类的类图。

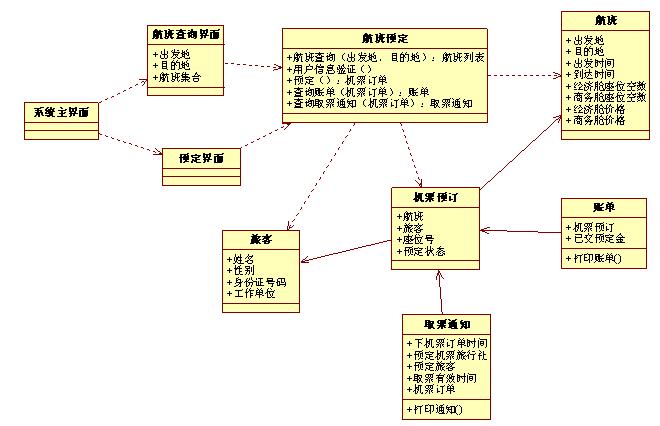
例如，图2.1的航空售票系统中，要实现预售票，则系统的软件静态结构如下：

图3.1 航空售票系统总体类图

系统静态结构说明

针对3.1中绘制的类图说明在类设计是如何体现功能需求和非功能需求的。

## 3.3 人机接口设计

3.3.1 设计应用类

给出了应用类模型，进一步说明有多少边界类。如有可能，最好进一步设计控制类和持久类，并通过应用类模型来说明它们之间的关系。示例：图3.2。

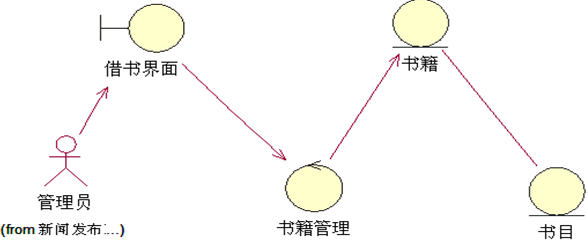


图3.3 图书借书的设计类图

3.3.2 用户接口1（具体标识符）设计

说明本接口的界面设计，给出界面布局。例如：图3.1

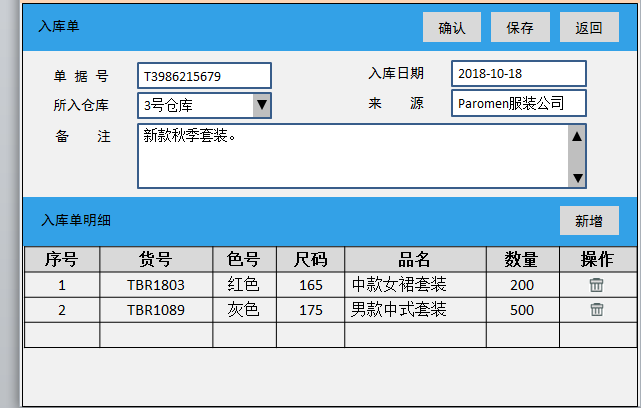


图3.2 新增入库单界面布局设计

说明界面元素主要控件，包括控件类型、控件名称、控件作用、控件属性（仅说明使用的属性）。说明本用户接口所实现的功能，例如是否设计有密码位数控制，用户名是否设计有限制等等。

3.3.3 用户接口2（具体标识符）设计

## 3.4 XXX模块设计说明

从小本节开始，详细说明每个模块的详细设计，逐个地给出需求规格的每个用例的设计考虑。要求给出所有用例的实现方法。

## 3.5 XXX模块设计说明

用类似第3.4小节的方式，说明第2个模块乃至第N个模块的设计考虑。

如果有更多模块，

往后继续添加。

# 4 数据库设计

## 4.1 数据库环境说明

提示：说明所采用的数据库系统，设计工具，编程工具等，说明其详细配置。

## 4.2 数据库的命名规则

提示：完整并且清楚的说明本数据库的命名规则。如果本数据库的命名规则与机构的标准不完全一致的话，请作出解释。

## 4.3 逻辑结构设计

提示：设计逻辑模型。说明把上述原始数据进行分解、合并后重新组织起来的数据库全局逻辑结构，包括所确定的关键字和属性、重新确定的记录结构和文卷结构、所建立的各个文卷之间的相互关系，形成本数据库的数据库管理员视图。

E-R图

关系模型

范式分析

逻辑模型图

## 4.4 物理结构设计

提示：建立系统程序员视图，包括：数据在内存中的安排，包括对索引区、缓冲区的设计；所使用的外存设备及外存空间的组织，包括索引区、数据块的组织与划分；访问数据的方式方法。下列必选项必须说明：

4.4.1 数据库表

说明数据库的表的设计。

表4.1 表汇总

|  |  |
| --- | --- |
| 表 名 | 功 能 说 明 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

表4.2 XX表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表 名 | |  | | | | | |
| 数据库用户 | |  | | | | | |
| 主 键 | |  | | | | | |
| 其他排序字段 | |  | | | | | |
| 索引字段 | |  | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| sql脚本 | |  | | | | | |
| 备注 | |  | | | | | |

4.4.2 视图的设计（如果有的话）

4.4.3存储过程、函数及触发器的设计（如果有的话）

## 4.5 安全性设计

提示：提高软件系统的安全性应当从“管理”和“设计”两方面着手。这里仅考虑数据库的安全性设计。必须说明在数据库的设计中，将如何通过区分不同的访问者、不同的访问类型和不同的数据对象，进行分别对待而获得的数据库安全保密的设计考虑。

4.5.1 防止用户直接操作数据库的方法

（1）用户标识和鉴定

（2）存取控制

（3）数据加密

4.5.2 用户帐号密码的加密方法

防止用户直接操作数据库的方法。用户只能用帐号登录到应用软件，通过应用软件访问数据库，而没有其他途径操作数据库。（3）用户帐号密码的加密方法。对用户帐号的密码进行加密处理，确保在任何地方都不会出现密码的明文。

4.5.1 角色与权限

可以考虑：（1）角色与权限。确定每个角色对数据库表的操作权限，如创建、检索、更新、删除等。每个角色拥有刚好能够完成任务的权限，不多也不少。在应用时再为用户分配角色，则每个用户的权限等于他所兼角色的权限之和。如表4.3所示，表格仅供参考。

表4.3 角色与权限

|  |  |  |
| --- | --- | --- |
| 角色 | 可以访问的表与列 | 操作权限 |
| 角色A |  |  |
|  |  |
|  |  |
| 角色B |  |  |
|  |  |
|  |  |

列出角色与权限管理SQL脚本代码。

# 5 测试用例设计

对照需求中功能，写出功能的测试用例和测试结果。应考虑边界测试、默认测试、空测试、非法值测试。每个用例应说明下列内容：

(1)测试用例名称

(2)测试项：规定并简要说明本测试用例所要涉及的项和特性、对于每一项、可考虑引用需求说明书和设计说明书。

(3)输入说明：规定执行测试用例所需的各个输入。有些输入可以用值（允许适当的误差）来规定。而另一些输入，如常数表或事务文件可以用名来规定。规定所有合适的数据库、文件、终端信息、内存常驻区域和由操作系统传送的值。规定各输入间所需的所有关系（如时序关系等）。

(4)输出说明：规定测试项的所有输出和特性（如：响应时间）。提供各个输出或特性的正确值（在适当的误差范围内）。

(5)环境要求：规定执行本测试用例所需的硬件特征和配置、系统软件和应用软件。

(6)规程说明：描述对执行本测试用例的测试规程的一切特殊限制。这些限制可以包括特定的准备、操作人员干预、确定特殊的输出和清除过程。

(7)用例间的依赖关系：列出必须在本测试用例之前执行的测试用例名称，归纳依赖性质。建议采用表格描述每个用例，示例见表5.1。

表5.1 登录-注册测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 登录 | yl-001 | 输入‘用户名’和‘密码’ | admin，123456 | 登录成功 | 通过 |
| yl-002 | admin，1234567 | 登录失败 | 通过 |
| yl-003 | admin1，123456 | 登录失败 | 通过 |
| yl-004 | admin,空 | 登录失败 | 通过 |
| yl-005 | 空,123456 | 登录失败 | 通过 |
| yl-006 | admin,一个空格 | 登录失败 | 通过 |
| yl-007 | 一个空格，123456 | 登录失败 | 通过 |
| yl-008 | admin，密码长度超过32 | 登录失败 | 通过 |
| yl-009 | 用户名超过36，123456 | 登录失败 | 通过 |
| 注册 | yl-010 | 用户名，密码，确认密码，真是姓名，性别，出生日期，地址，邮政编码，电话，email，密保问题，密保答案，验证码 | Tony,123456,123456,test1,男，1991-10-10，长沙，410014，85577424，505@qq.com，我是谁，test1. | 创建成功 | 通过 |
|  |  |  |  |  |  |

# 附录

## 附录1 程序运行结果

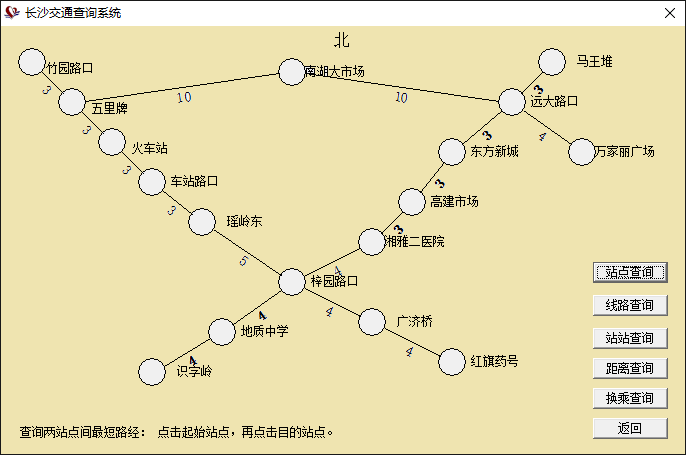


图1 查询界面

## 附录2 程序源代码

列出自己设计的主要代码

1. Am\_dij.c

#include "amgraph.h"

//获取最短路径(采用迪杰斯特拉算法)

void ShortestPath\_DIJ(MGraph G,int v0,int v1,PathMatrix P,ShortPathTable D)

{ // 用Dijkstra算法求有向网G的v0顶点到其余顶点v的最短路径P[v]及带权长度

// D[v]。若P[v][w]为TRUE，则w是从v0到v当前求得最短路径上的顶点。

// final[v]为TRUE当且仅当v∈S，即已经求得从v0到v的最短路径 算法7.15

int v,w,i,j,min;

Status final[MAX\_VERTEX\_NUM]; // 辅助矩阵，为真表示该顶点到v0的最短距离已求出，初值为假

int path[100];//adding 存放前驱结点

for(v=0;v<G.vexnum;++v)

{

final[v]=FALSE; // 设初值

D[v]=G.arcs[v0][v].adj; // D[]存放v0到v的最短距离，初值为v0到v的直接距离

for(w=0;w<G.vexnum;++w){

P[v][w]=FALSE; // 设P[][]初值为FALSE，没有路径

path[w]=v0;//adding

}//adding

if(D[v]<INFINITY) // v0到v有直接路径

P[v][v0]=P[v][v]=TRUE; // 一维数组p[v][]表示源点v0到v最短路径通过的顶点

}

D[v0]=0; // v0到v0距离为0

final[v0]=TRUE; // v0顶点并入S集

for(i=1;i<G.vexnum;++i) // 其余G.vexnum-1个顶点

{ // 开始主循环，每次求得v0到某个顶点v的最短路径，并将v并入S集

min=INFINITY; // 当前所知离v0顶点的最近距离，设初值为∞

for(w=0;w<G.vexnum;++w) // 对所有顶点检查

if(!final[w]&&D[w]<min) //在S集之外的顶点中找离v0最近的顶点，并将其赋给v，距离赋给min

{

v=w;

min=D[w];

}

final[v]=TRUE; // 将v并入S集

for(w=0;w<G.vexnum;++w) // 根据新并入的顶点，更新不在S集的顶点到v0的距离和路径数组

if(!final[w]&&min<INFINITY&&G.arcs[v][w].adj<INFINITY&&(min+G.arcs[v][w].adj<D[w]))

{ // w不属于S集且v0→v→w的距离＜目前v0→w的距离

D[w]=min+G.arcs[v][w].adj; // 更新D[w]

for(j=0;j<G.vexnum;++j) // 修改P[w]，v0到w经过的顶点包括v0到v经过的顶点再加上顶点w

P[w][j]=P[v][j];

P[w][w]=TRUE;

path[w]=v;//adding

}

}

DisplayPath(G,path,D,v0,v1);

}

注：源代码以文件为单位附列在此。