基于面向对象程序设计开发方法的论文写法概要

1 前言

1.1 程序设计两种泛型：结构化程序设计和面向对象程序设计。

- 结构化程序设计

思想：结构化程序设计从系统的功能入手（有的用户也成为面向过程的设计），按照工程的标准和严格的规范将系统分解为若干功能模块，系统是实现模块功能的函数和过程的集合。

- 面向对象程序设计

思想：在进行程序开发之前，先将程序的业务变化进行合理的分析，将程序中的代码与现实中的事物结构特征结合起来考虑。面向对象程序设计是一种围绕真实世界的概念来组织模型的程序设计方法，它使用对象来描述问题空间的实例。

1.2 面向对象程序设计开发方法：UML（统一建模语言）

建筑有建筑图的绘制标准，大家看到建筑图就能知道这个楼如何盖起来。机械绘图有其绘制标准和符号，机械工程师看到机械图就知道这是一个什么机构零件。同理，面向对象开发方法也有自己的一套表达方式方法和标准，这就是 UML。

虽然 UML 叫统一建模语言，但是它不是一门语言，它是一套标准，用于表示面向对象程序设计的建模。也就说，UML 建模工程师，可以不懂 java，不懂 C#，但是他只要懂得 UML，就能用 UML 表示出程序分析和设计模型， java 工程师看到该模型就能写出 java 代码，同样地，c#工程师看到它就能用 C#实现程序。

总之，你的 UML 建模设计拿给码农就能写出代码，这是基本指导原则。如果你的设计拿给另外的人编码，编不出来，就是失败的设计。

2 基于面向对象程序设计开发方法的论文组织

2.1 论文的组织结构

第 1 章 绪论：3-5 页

主要介绍项目的背景、意义、技术和课题的发展国内外现状，论文的主要内容，论文的组织结构。

第 2 章 系统分析：20 页左右

- 问题定义：确定系统的目标、规模和基本任务。问题定义阶段必须回答的关键问题是：“要解决的问题是什么?” 1-2 页（这一部分通常也会放在第一章绪论中）
- 可行性研究：从经济、技术、法律、社会等方面分析确定系统是否值得开发，及时建议停止项目开发，避免人力、物力、时间的浪费。这个阶段要回答的关键问题是：“对于上一个阶段所确定的问题有行得通的解决办法吗?” 1-2 页
- **需求分析（基石）**：该阶段主要解决的问题是“目标系统必须**做什么**”，也就是要**深入描述软件的功能和性能**；确定软件设计的限制和软件与其他系统元素的接口；定义软件的其他有效性需求，并用“需求规格说明书”的形式准确地表达出来，提交管理机构评审。

第 3 章 系统设计 20 页左右

- **概要设计**：确定系统设计方案，软件的体系结构。确定软件由哪些模块组成以及这些模块之间的相互关系。这个阶段必须回答的关键问题是：“概括地说，应该怎样实现目标系统?”总体设计又称为概要设计。2-3 页
- **详细设计（核心）**：描述应该如何具体地实现系统。详细设计每个模块，确定实现模块所需要的算法和数据结构。也就是回答下面这个关键问题：“应该怎样具体地实现这个系统呢?”

第 4 章 系统实现：5 页左右

第 5 章 系统测试：5 页以内

第 6 章 结语（主要介绍课题的总结和需要提高的展望等）1 页

第 7 章 致谢（不写章节号，千万不要写成别人的了）1 页

参考文献（不写章节号，注意文献尽量用近几年的）1 页

2.2 论文的书写规范

2.2.1 文字要求

采用国家语言文字工作委员会正式公布的简化汉字书写，要求语句通顺、论述严谨、程序和实验数据完整、齐全、规范、正确。

2.2.2 字体与字号

- 论文题目　　2 号黑体
- 各章题序及标题　　3 号黑体
- 各节的一级题序及标题　　小 3 号黑体
- 各节的二级题序及标题　　4 号黑体
- 各节的三级题序及标题　　小 4 号黑体
- 款、项　　小 4 号宋体
- 正文（各级标题除外）　　小 4 号宋体
- 中文摘要、参考文献标题　　3 号黑体
- 中文摘要正文、参考文献　　小 4 号宋体
- 英文摘要标题　　3 号 Times New Roman 字体
- 英文摘要正文　　小 4 号 Times New Roman 字体
- 目录标题　　3 号黑体
- 目录内容中章的标题　　小 4 号黑体
- 目录中其它内容　　小 4 号宋体
- 论文页码　　小 5 号宋体、页面底端居中、阿拉伯数字连续编码
- 页眉　　小 5 号宋体
- 阿拉伯数字和字母　　Times New Roman 体

2.2.3 页面设置

页眉：页眉内容左侧为“学年论文”，右侧为学生所完成的论文题目。

页边距：页边距上 2.5cm，下 2.5cm，左侧 2.5cm，右侧 2cm，页眉：2.0cm、页脚：1.5cm。行间距为 1.5 倍行距，左侧装订。

页码：论文页码从论文的绪论开始至附录，用小五号宋体、阿拉伯数字连续编排，页码位于页面底端居中。论文的绪论、正文主体的各章开始、结论以及参考文献、致谢、附录均要另起新页编页码。中英文摘要、目录各自使用罗马数字（Ⅰ、Ⅱ、Ⅲ……）单独编页码，页码位于页面底端居中。封面不编页码。

2.2.4 中英文摘要

中文摘要包括：“摘要”字样、摘要正文和关键词。摘要正文下空一行顶格打印“关键词”三字。关键词之间用“;”隔开，最后一个关键词后不打标点符号。

英文摘要另起一页。英文摘要内容及其关键词应与中文摘要内容及其关键词一致，在语法、用词和书写上应正确无误，语句通顺，文字流畅。英文摘要关键词之间用“,”隔开，最后一个关键词后不打标点符号。

2.2.5 目录

目录的三级标题，建议按（1…...、1.1……、1.1.1……）的格式编写，目录中各章题序的阿拉伯数字用 Times New Roman 体，文字字号及字体见第六条。

2.2.6 论文正文

（一）章节和各章标题

正文分章节撰写，每章应另起一页。各章标题要突出重点、简明扼要。字数一般在15字以内，不得使用标点符号。标题中尽量不采用英文缩写词，必须采用时，应使用本行业的通用缩写词。

（二）层次

层次以少为宜，根据实际需要进行选择。正文层次的编排和代号要求统一，层次如下：章（如“第1章”）、节（如“1.1”）、条（如“1.1.1”）、款（如“1、”）、项（如“（1）”），层次用到那一层视需要而定，若节后无“条”可直接到“款”、“项”。

2.2.7 引用文献

引用文献标示方式应全文统一，并采用所在学科领域内通用的方式，用上标的形式置于所引内容最末句的右上角，用小4号字体。所引文献编号用阿拉伯数字置于方括号中，如：“…成果[1]”。当提及的参考文献为文中直接说明时，其序号应该用小4号字正文排齐，如“由文献[8，10～14]可知”。

摘要、各级标题处不得出现引用文献标示。

2.2.8 名词术语

科技名词术语及设备、元件的名称，应采用国家标准和部颁标准中规定的术语或名称。标准中未规定的术语或名称要采用行业通用术语或名称。全文名词术语必须统一。一些特殊名词或新名词应在适当位置加以说明或注解。

采用英文缩写词时，除本行业广泛应用的通用缩写词外，文中第一次出现的缩写词应该用括号注明英文全文。

2.2.9 计量单位

物理量计量单位及符号一律采用《中华人民共和国法定计量单位》（GB3100～3102-1993），不得使用非法定计量单位及符号。计量单位符号，除用人名命名的单位第一个字母用大写之外，一律用小写字母。论文中某一物理量的名称和符号应统一。

表达时刻应采用中文计量单位，如：“上午8点3刻”，不能写成“8h45min”。

2.2.10 数字

按国家语言文字工作委员会等六单位1987年发布的《关于出版物上数字用法的规定》，除习惯用中文数字表示的以外，一般均采用阿拉伯数字。

2.2.11 外文字母的正、斜体用法

按照GB3100～3102－1986及GB7159－1987的规定使用，即物理量符号、物理量常量、变量符号用斜体，计量单位符号均用正体。

2.2.12 公式

公式须用公式编辑器编辑公式，一般不得截图作为公式。

公式应另起一行居中书写，若公式前有文字（如“解”、“假定”等），文字顶格书写，

公式仍居中写。公式末不加标点。公式和编号之间不加虚线。公式较长时最好在“=”前转行；如难实现，则可在“+、－、×、÷”运算符号处转行，运算符号应在转行后的行首，公式的编号用圆括号括起来放在公式右边行末。

公式序号按章编排，如第 1 章第一个公式序号为“(1.1)”，附录 2 中的第一个公式为（②-1）等。

文中引用公式时，一般用“见公式（1-1)”或“由公式（1.1)”。

公式中用斜线表示“除”的关系时应采用括号，以免含糊不清，如 a/(bcosx)。通常“乘”的关系在前，如 acosx/b 而不能写成（a/b）cosx。

2.2.13 插表

表格一般采用三线表，不加左、右边线，居中排写。

表序一般按章编排，如第 1 章第一个插表的序号为“表 1－1”等。表序与表名之间空一格，表名中不允许使用标点符号，表名后不加标点。表序与表名置于表上，表头设计应简单明了，尽量不用斜线。表头中可采用化学符号或物理量符号。

全表如用同一单位，将单位符号移到表头右上角，加圆括号。

表中数据应正确无误，书写清楚。数字空缺的格内加“—”字线（占 2 个数字宽度)。表内文字和数字上、下或左、右相同时，不允许用“〃”、“同上”之类的写法，可采用通栏处理方式。

文管类的插表在表下一般根据需要可增列补充材料、注解、附记、资料来源、某些指标的计算方法等。

表内文字说明，起行空一格，转行顶格，句末不加标点。表题用五号字，表内文字及表的说明文字均用五号字，中文用宋体。

2.2.14 插图

插图应与文字紧密配合，文图相符，技术内容正确。

每个图均应有图题（由图号和图名组成)。图号按章编排，如第 1 章第一图的图号为“图 1.1”等。图题置于图下。有图注或其他说明时应置于图题之上。图名在图号之后空一格排写。引用图应说明出处，在图题右上角加引用文献编号。图中若有分图时，分图号用 a)、b)等置于分图之下。

图中各部分说明应采用中文（引用的外文图除外）或数字项号，各项文字说明置于图题之上（有分图题者，置于分图题之上)。

图题用五号字，图内文字及说明均用五号字，中文用宋体。

插图与其图题为一个整体，不得拆开排写于两页。插图应编排在正文提及之后，插图处的该页空白不够排写该图整体时，则可将其后文字部分提前排写，将图移到次页最前面。

论文中的照片图均应是原版照片粘贴（或数码像机图片)，照片可为黑白或彩色，应主题突出、层次分明、清晰整洁、反差适中。照片采用光面相纸，不宜用布纹相纸。对金相显微组织照片必须注明放大倍数。

论文中的插图不得采用复印件。对于复杂的引用图，可采用数字化仪表输入计算机打

印出来的图稿。

2.2.15 参考文献

参考文献的序号应按在正文中出现的顺序排列，并用数字加方括号表示，如[1]，[2]，…。每一参考文献条目的最后均以“.”结束。按照引用文献类型不同使用不同的方法，具体参照 GB/T 7714-2015《文后参考文献著录规则》规定。

2.2.16 附录

理工类论文附录的序号采用“附录 1”、“附录 2”等；非理工类论文附录的序号相应采用“附录一”、“附录二”等。

3 需求分析的组织

需求分析是基石，包括功能需求和非功能性需求。

3.1 功能需求

采用 UML 的用例分析方法。主要说明“什么角色做什么事情”。角色就是 actor，做什么事情就是“用例”，英文叫 usecase。从本质上讲，一个用例是用户与计算机之间为达到某个目的而进行的一次典型交互作用。需要注意的是用例能捕获功能性需求，它描述的是关于系统将做什么。整个系统的功能需求是由一系列用例的集合组成的。

（1） 用例图

如图 3.1 基本合理，存在一些细节处理的问题。（注意，**图必须有编号和标题，居中。图的编号一般只有二级标题，且在图的下面。图的标题和图本身必须在同一页上**）。

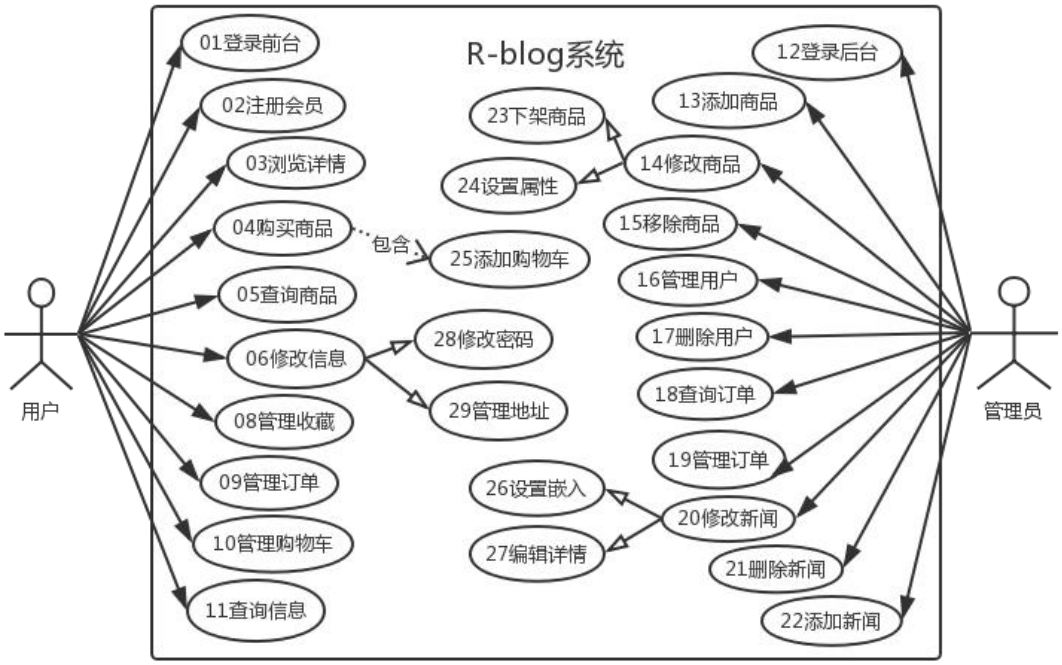

图 3.1 某案例用例图的示意图

四个基本元素：参与者、用例和关系和方框表示的系统边界。

用例图可视化地描述了系统外部的使用者（抽象为参与者）和使用者使用系统时，系统为这些使用者提供的一系列服务（抽象为用例），并清晰地描述了参与者和参与者之间的泛化关系、用例和用例之间的包含关系（或泛化关系、或扩展关系）以及用例和参与者之间的关联关系。所有的用例都位于方框之内，该方框称为“系统边界”。

A. 椭圆里面的用例必须是动宾结构；

B. 请特别注意用例之间的包含和扩展关系的含义；

C. 请特别注意用例图中关系的表示形式，都是有规定的。

（２）和用例说明（对每一个用例）

（注意，**表必须有编号和标题，居中。编号一般只有二级编号，且在表的上方。虽然表可以跨页，但是标题和表本身必须在同一页上，也就说说，不能标题单独一页，下一页是表**）。

表 3.1 前台登录用例表

用例名称	登录前台	标识号	01
参与者	用户		
用例描述	用户可在前台进行登录		
用例需求	用户名唯一		
前置条件	用户打开浏览器进入欢迎界面		
后置条件	无		
基本操作流程	1.用户点击登录进入登录界面； 2.用户输入账号； 3.用户输入密码； 4.用户点击登录；		
异常处理流程	2.1 输入非数字要求重新输入；		

表 3.2 用户注册用例表

用例名称	注册账号	标识号	02
参与者	用户		
用例描述	未注册用户可在欢迎页进行注册		
用例需求	1.登录使用邮箱； 2.密码非明文显示； 3.手机和邮箱进行格式校验。		
前置条件	用户打开浏览器访问欢迎页		
后置条件	无		
基本操作流程	1.在用户登录界面点击注册按钮进入注册页面； 2.填写账号，密码等基本信息； 3.点击注册按钮；		
异常处理流程	2.1 账号或密码格式错误提示无法进行注册； 3.1 已经注册用户账号提示错误。		

表 3.3 列出一般的用例说明模板。

表 3.3 用例规约的定义模板

用例编号	**[为用例制定一个唯一的编号，通常格式为UCxx]**	
用例名称	**[应为一个动词短语，让读者一目了然地知道用例的目标]**	
用例概述	**[用例的目标，一个概要性的描述]**	
范围	**[用例的设计范围]**	
主参与者	**[该用例的主Actor，在此列出名称，并简要的描述它]**	
次要参与者	**[该用例的次要Actor，在此列出名称，并简要的描述它]**	
项目相关人利益说明	**项目相关人**	**利益**
	[项目相关人员名称]	**[从该用例获取的利益]**
	……	**……**
前置条件	**[即启动该用例所应该满足的条件。]**	
后置条件	**[即该用例完成之后，将执行什么动作。]**	
成功保证	**[描述当前目标完成后，环境变化情况。]**	
基本事件流	**步骤**	**活动**
	1	**[在这里写出触发事件到目标完成以及清除的步骤。]**
	2	**……(其中可以包含子事件流，以子事件流编号来表示)**
扩展事件流	**1a**	**[1a表示是对1的扩展，其中应说明条件和活动]**
	1b	**……(其中可以包含子事件流，以子事件流编号来表示)**
子事件流	**[对多次重复的事件流可以定义为子事件流，这也是抽取被包含用例的地方。]**	
规则与约束	**[对该用例实现时需要考虑的业务规则、非功能需求、设计约束等]**	

（3）用例举例

下面举例几个不好的例子。

图 3.2a 使用包含关系，意味着住户管理用例必须完成添加、同时也必须完成删除、查询和修改。显然不对。

如果改为扩展，理论上说的过去。但是对于用例“住户管理”不好描述（它的主事件流是什么？），并且用例应该是动宾结构。

修改为图 3.2b。画出系统边界，住户管理成为子系统（模块）的名称。

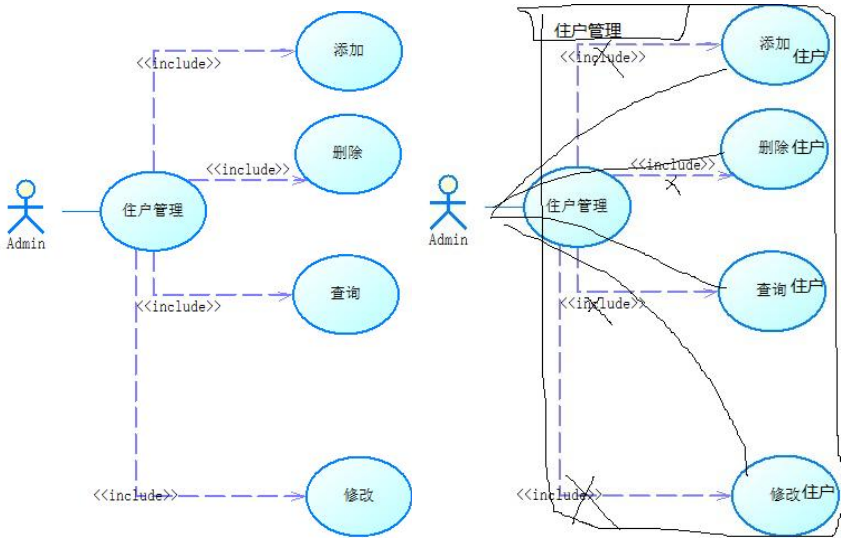

图 3.2a 错误的用例图　图 3.2b 修改后的用例图

图 3.3 也存在很多问题，太多问题。。。怎么改？大家试一下。

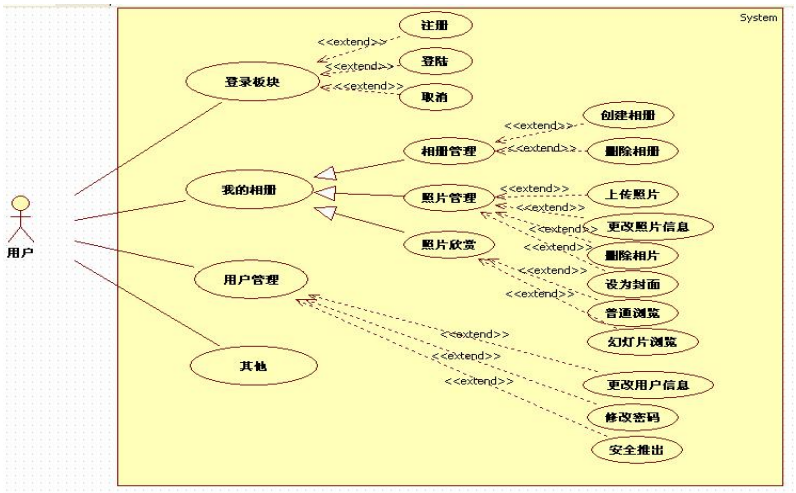

图 3.3 存在问题的用例图

再举一个例子：某酒店管理系统，客人退房功能，在用例图中出现了“客人”角色和“出示押金条”用例，请大家考虑是否合适？

3.2 性能需求

简单举俩例说明，看下面不好的例子：

（1）浏览器响应快捷，支持快速响应服务器页面，速度能达到业务的要求；

（2）服务器并发性强，在达到同时连接多台终端时能够不卡顿；

修改后的，可测可度量：

（1）浏览器响应快捷，支持快速响应服务器页面，速度能达到业务的要求，单个页面的加载时间应当控制在 2 秒以内为最佳响应速度；

（2）服务器并发性强，在达到同时连接超过 1000 台终端时能够不卡顿；

下面是一些其他非功能性需求的示例（部分，截图）

- **3.9 安全性需求**

  （1）权限控制：最高管理员权限账户应当只有一个，并且
  只能由最高管理员发放，管理员不支持个人添加行为，并且应
  级；

  （2）传递数据加密或隐藏处理：本系统对一些重要的传递
  法进行隐式加密，如用户操作 op 参数、重要的具有标示性意义
  不需要进行直接展示的数据应当以隐藏的方式在页面中加载；

  （3）数据备份：允许管理员进行数据的备份和恢复操作，
  和丢失。在一定的时间段内，能够自动提示将数据由数据库数
  环境，成为实体文档进行独立的存储保护操作；

  （4）拦截器与过滤器：防范一些攻击操作，如 sql 注入式
  过拼接数据访问后台，应当设计对应的关键词过滤操作，预防
  入。

- **3.10 软件质量属性**

  记录日志：本系统应该能够记录系统运行所发生的错误，
  同本软件一起发行的用户文档包括：

  （1）浏览器操作帮助手册：html 页面或可下载的 Word 格

  （2）管理员后台数据操作手册：html 可访问页面以及 Wo

- **3.11 其它需求**

  （1）软件开发维护后期可使用标准化的申请 excel 模板，

○○○○○○○○○○○○○○○○

4 核心是系统设计

系统设计分为总体设计（概要设计）和详细设计两大部分。设计的最高指导原则，你的设计拿给编码人员可以写出代码。

4.1 总体设计

确定系统设计方案，软件的体系结构。

其中，一个重要的内容是确定软件由哪些模块组成以及这些模块之间的相互关系。

图 4.1 给出了一个示意图，该图中的模块有编号，是一种良好的习惯。表 4.1 给出了对模块的说明。当然，表 4.1 也可以不用表的形式，直接用文字一个个罗列描述（这样做的好处是，不会让论文看起来全是图表，让论文看起来有图表有文字，真真做到图文并茂）。

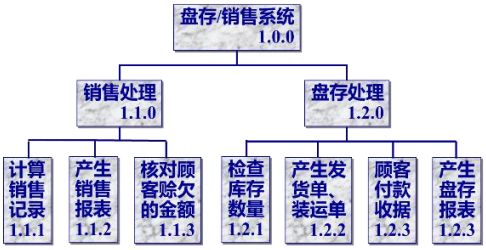

图 4.1 某系统功能结构图

表 4.1 系统的功能描述

编号	说　　明
1.0.0	销售/盘存处理框图
1.1.0	顾客订单检查，核对顾客赊欠金额，产生销售报表
1.1.1	用工作文件的盘存项目号，对顾客订单进行核对和排序
1.1.2	以地区和人员为单位，编制销售报表，计算销售佣金
1.1.3	检验顾客赊欠金额，计算折扣，确定支付项目
1.2.0	处理顾客盘存管理报表，顾客付款收帐，处理发货、包装、托运
……	

4.2 功能设计

这一部分主要对 4.1 节列出的主要功能（也可以参考需求分析的用例），作出其实现过程的详细说明。

注意：用于说明“过程”的表示方法，传统上是流程图，UML 中主要是活动图、顺序图和状态图。**使用哪种图没有固定要求**，只要能说明一个活动过程，都可以。图 4.2 到图

4.5 分别给出了示例。

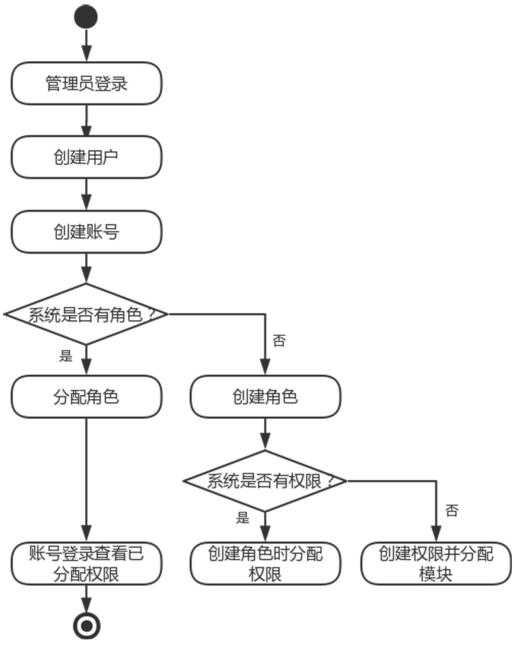

图 4.2 系统基础管理业务流程图

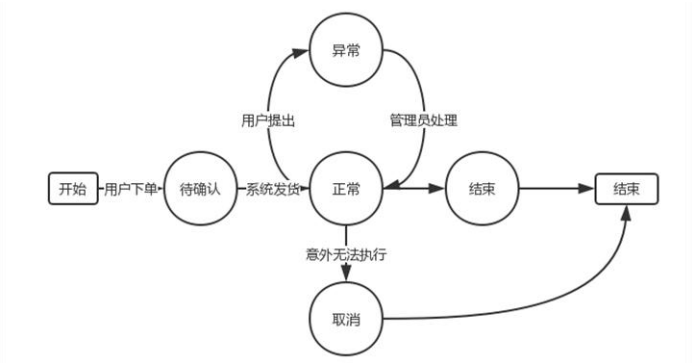

图 4.3 订单状态转换图

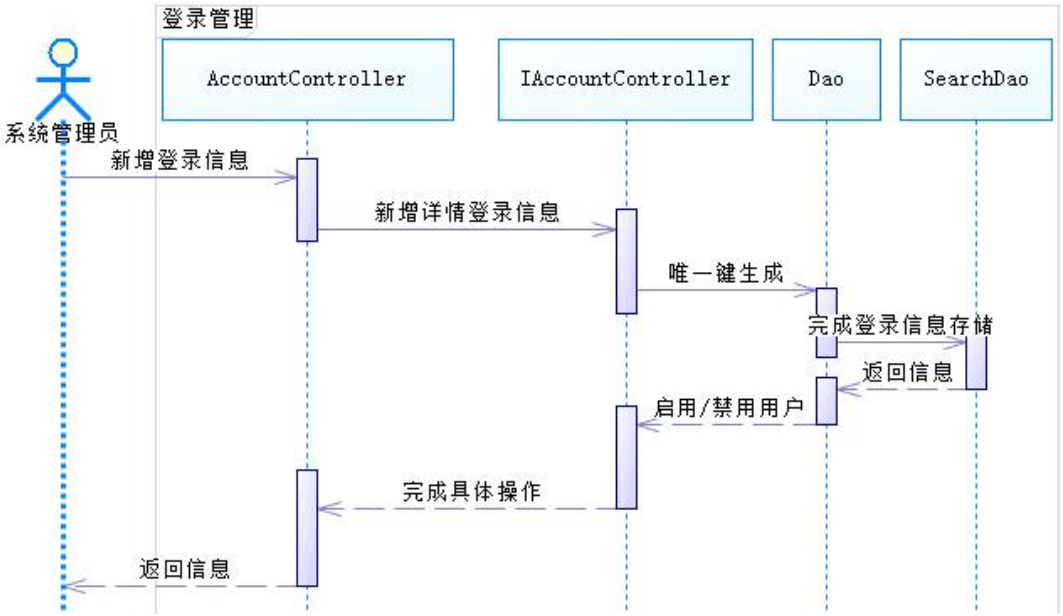

图 4.4 基础管理时序图

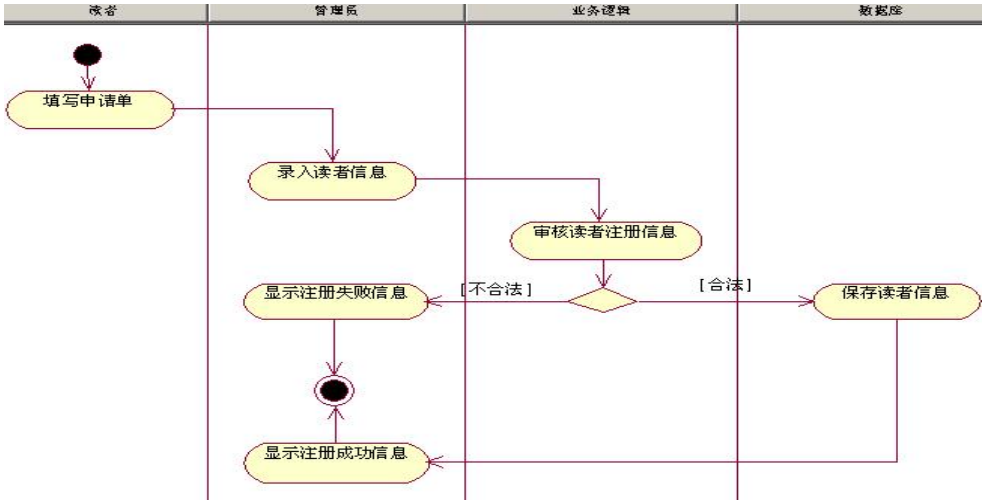

图 4.5 带泳道的活动图

4.3 数据库设计

这一部分也可以单独做成一章。包括 ER 图，PDM 图和 CDM 图。包括数据表。注意这一部分不要全部是图表，注意要有足够的文字说明，使得论文图文并茂。

另外，关于 ER 图中的条目，应该用数据字典加以详细描述。

4.4 类的设计

类图和类的设计是系统详细设计阶段必须要有的东西。分析确定系统中有哪些类以及类的样子，以及类之间的关系（类图）。

4.4.1 类图

确定系统中有哪些类，类可以分为三种类型：实体类(Entity)、边界类(Boundary)和控制类(Control)。

（1） 从用例视图中寻找类，一般是从用例的事件流开始，查看事件流中的名词来获得类；

（2） 也可以检查顺序图和通信图中的对象，通过对象的共性来寻找类；

（3） 从 4.2 节中功能中发现类。例如图 4.5 我们觉得应该有“管理员”类和“读者”类；

（4） 从 4.3 的数据库中发现类，比如学生信息表对应“学生类”；

（5） 其他控制业务流程的类；

（6） 界面类。

确定了类后，应该画出这些类之间的关系（类图）。

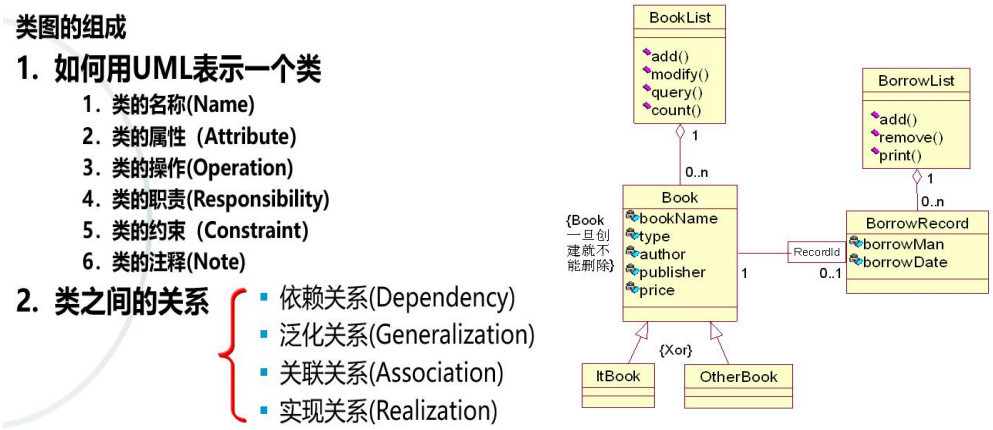

图 4.6 某系统类图

4.4.2 类的详细设计

这一部分给出类具体的样子，否则码农写不出代码。表 4.1 给出一个例子。

（1）BorderList 类的设计

表 4.1 BorderList 类详细设计

类名	BorderList
父类	●ActionSupport 类：继承了 Struts 框架的下的 action 支持类，在框架下将会根据配置文件调用对应的方法，默认调用 execute（）方法
接口	●SessionAware 接口：Struts 下的 session 自动填充接口，需要 override 对应的 set 方法和声明 map 私有化变量。 ●ServletRequestAware 接口：Struts 下的 httpservletRquest 自动填充接口，需要 override 对应的 set 方法和声明 map 私有化变量。
属性	●private Map session 属性：struts 框架下的 session 变量，为类中使用 session 接口必须声明的存储变量。 ●private HttpServletRequest request 属性：struts 框架下的 session 变量，为类中使用 servletRquest 接口必须声明的存储变量。

方法	●public String execute()：action 在无特别声明调用方法下的默认调用方法，其返回值对应配置文件中的返回值。 ●public void setSession(Map)：实现在调用 action 时的对私有变量 session 的初始化。 ●public void setRequest(HttpServletRequest)：实现在调用 action 时的对私有变量 httpservletrequest 的初始化。

BorderList 类主要实现了对应 action 操作的方法调用，需使用 Struts 框架配根据配置文件和前台请求调用对应的 action 操作，其调用的方法可在配置文件中填写或默认调用 excute（）方法，可通过配置文件在 acion 前后实现拦截器的调用，action 中所需要的接口的具体实现，通过编写 @autowrite 自动注入注解，由 spring 框架进行实例化。

注意：在方法的说明中，如果方法的实现比较简单，可以用语言简要描述。如果方法的实现比较复杂（比如排序），那也可以用流程图、自然语言、顺序图、活动图等表达出来。

（2）BorderDao 类的设计

类的详细设计也可以不用表的形式，采取如下形式：

- 类名:BorderDao
- 父类:BaseHibernateDAO 类：继承了 hibernate 框架的的基本 dao 类方法，如数据的基本处理和数据库请求以及对应的缓存数据处理等。
- 接口:无
- 属性(这一部分可以做成表，包含属性名，类型，可见性，长度，默认值和备注等列)

public static final String USERID 属性：与 Border 表中的“userid”列对应。

public static final String GOODSID 属性：与 Border 表的“goodsid”列对应。

public static final String AMOUNT 属性：与 Border 表中的“amount”列对应。

public static final String ADDRESS 属性：与 Border 表中的“address”列对应。

public static final String NAME 属性：与 Border 表中的“name”列对应。

public static final String TELLPHONE 属性：与 Border 表中的“tellphone”列对应。

public static final String TRANSWAY 属性 ：与 Border 表中的“transway”列对应。

public static final String PAYWAY 属性：与 Border 表中的“payway”列对应。

public static final String NOTE 属性：与 Border 表中的“note”列对应。

public static final String DATA 属性：与 Border 表中的“data”列对应。

public static final String SUMPRICE 属性：与 Border 表中的“sumprice”列对应。

public static final String FLAG 属性:，与 Border 表中的“flag”列对应。

- 方法

① public void save(Border transientInstance)：hibernate 框架中的数据对象保存方法，若 hibernate session 中已经有这个对象的存在，则会报错，需先对数据进行融合处理。

① public void delete(Border persistentInstance)：hibernate 框架中的数据对象删除的方法，调用该方法删除的是 session 中的对象数据，要永久保存到数据库中仍需开始事务进行处理。

② public Border findById(java.lang.Integer id)：使用 hql 语句基于 hibernate 框架下根据 id 从数据库中做相应的查询,其返回结果为对应的 border 对象。

③ public List findByProperty(String property, Object value) ：使用 hql 语句基于 hibernate 框架下根据传入的对象名和值从数据库中做相应的查询，其返回结果为对应的 border 对象的列表 lit。

④ public List findAll()：使用 hql 语句基于 hibernate 框架下返回当前 border 表内的全部记录的 list

⑤ public List findAllOrderby(String order):使用 hql 语句基于 hibernate 框架下返回当前 border 表内的全部记录的 list，基于 order 的值进行做降序排序。

⑥ public Border merge(Border detachedInstance)：数据融合，若 hibernate session 中已经有该数据，则与传入对象融合成新数据，避免单一数据的重复，返回值为融合后的对象。

4.5 原型设计（人机界面设计）

这一部分很多同学放在“系统实现”章中，倘若这样，那就是截图和粘代码，不算工作量。强烈建议这一部分放在系统设计的“原型设计”。

贴图，并配上关于这个图中各个组件元素的说明，以及其他关于这个图的说明。

5 系统实现（本课程不需要）

根据 4.5 的描述，这一部分不要截图。这一部分可以：

（1） 列出系统的部署等信息；

（2） 粘贴小部分代码（关键代码），并配文字说明；

（3） 系统测试的内容。

下面给出本部分结构组织的一些示意（仅供参考，不是标准）。

5.1 配置部署

SSH 框架的配置文件均采用 xml 文件,基于 MVC 的 J2EE 平台下的 Web 应用，使用 XML 进行部署扩展，使框架更加灵活、可扩展、易于维护[12]。而服务器我们使用了 tomcat，主要是因为 Tomcat 高并发优化处理为跨平台通用的 Web+APP 网络软件系统提高网速、改善用户体验、保证信息传递的及时性、增强大数据并发处理能力,提供了有力的支撑[13]。

鉴于项目的实际部署情况，项目需要对以下框架或服务进行配置部署。

(1) Tomcat 配置

- 设置默认欢迎为 R-Blog 的欢迎页；
- 指定开放端口为具体端口号，默认 8080；
- 将项目发布到 tomcat 服务器下的 webapp 上。

<Connector port="8080"URLEncoding="UTF-8" redirectPort="8443" connectionTimeout="20000" protocol="HTTP/1.1"/>

(2) Web 项目部署配置

- 设置项目 Encoding 编码格式为 utf-8；
- 设置直接访问项目的空间下的自动访问指定页面设置；
- 编写 web.xml 设置项目启动时自动启动 spring 框架和 struts 框架；
- 编写 web.xml 设置过滤器将所有的.action 请求转转到 struts 核心控制器；
- 导入 struts2，hibernate 3，spring2.5 以及 apache 等相关的 jar 包。

<display-name>RBlogV2</display-name>
<welcome-file-list>
<welcome-file>jsp/welcome.jsp</welcome-file>
</welcome-file-list>
<filter>
<filter-name>SqlFilter</filter-name>
<filter-class>interceptor.SqlFilter</filter-class>
</filter>
<filter>
<filter-name>struts2</filter-name><filter-class>struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>

```
    </filter>
    <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>*.action</url-pattern>
    </filter-mapping>
<filter-mapping>
<filter-name>SqlFilter</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>
<listener><listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>classpath:applicationContext.xml</param-value>
</context-param>
```

(3) Mysql 配置置

- 编写 my.ini 文件设置 sql 事务链接的编码格式为 utf-8；
- 设置对应的数据链接地址以及对应的连接账户，并授予管理权限。

```
character-set-server = utf8
character_set_client = utf8
```

(4) Struts 配置置

- 编写 struts.xml 文件设置 action 并设置对应的 result；
- 编写 struts.xml 文件配置拦截器栈并部署到对应的 action 下；
- 编写 struts.xml 文件设置过滤器，其中最为重要的是文件流过滤器。

```
<bean type="org.apache.struts2.dispatcher.multipart.MultiPartRequest" scope="default"
    name="myRequestParser" class="interceptor.RequestParseWrapper" optional="true" />
    <constant name="struts.multipart.handler" value="myRequestParser" />
    <package name="default" namespace="" extends="struts-default">
    <interceptors>
            <interceptor name="logincheck" class="interceptor.LoginIntercept" />
            <interceptor name="Blogincheck" class="interceptor.BLoginIntercept" />
            <interceptor-stack name="intStack">
                <interceptor-ref name="logincheck" />
                <interceptor-ref name="defaultStack" />
            </interceptor-stack>
            <interceptor-stack name="BintStack">
                <interceptor-ref name="Blogincheck" />
                <interceptor-ref name="defaultStack" />
            </interceptor-stack>
        </interceptors>
        <action name="login" class="action.LoginAC">
            <result name="success">jsp/welcome.jsp</result>
        </action>
```

5.2 系统部署

每个软件系统都会有相应的部署图，用来显示系统中软件和硬件的物理架构。而一个系统模型只会有一个部署图，对硬件的配置及其软件如何部署到网络结构中进行展示。如图 5.1 所示，展示了本系统的部署图。

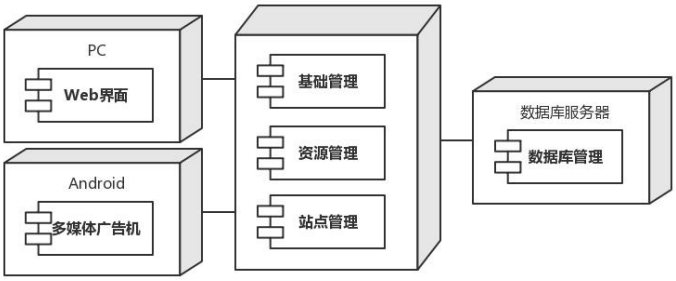

图 5.1 系统部署图

合理的系统网络架构是本系统和硬件设备在数据安全同步方面的保障，同时在连接本系统与硬件设备时起着至关重要的作用。如图 5.2 所示，展示了本系统的网络架构。

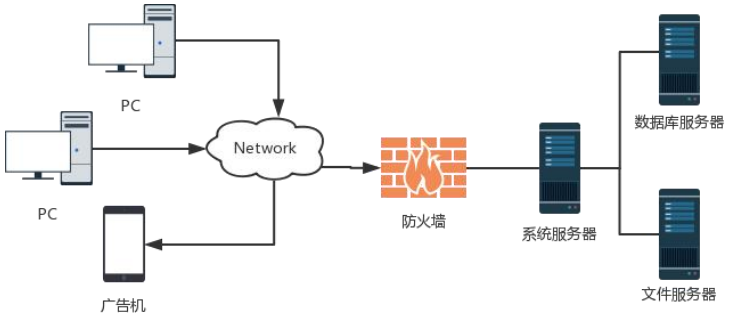

图 5.2 系统网络架构图

6 系统测试（本课程不需要）

测试计划/测试用例设计和测试的结果分析。