

《管理信息系统综合实验》实验指导

1、系统规划

1.1 系统规划方法

1.1.1 企业系统规划法

企业系统规划法（Business System Planning, BSP）主要是帮助企业确定信息系统总体结构。通过 BSP 方法，可以构建企业信息系统的组成，也就是由哪些子系统组成。同时，可以对企业信息化中的数据进行统一的设计和规划，规范各子系统之间数据接口。该方法由于是以企业的业务流程为基础，而不依赖于企业的组织结构，所以设计出的系统总体结构具有很强的适应性和稳定性。即使企业的组织结构发生了变化，信息系统的总体结构依然可以保持稳定。

企业系统规划法主要步骤包括启动阶段、分析结构、构建阶段和报告撰写阶段。

（1）启动阶段。建立系统规划小组，制定系统规划的工作计划。召开启动会，邀请企业一把手参加，对系统规划的意义、重要性、内容、步骤、人员和进度等进行介绍。着重说明需要各部门需要协助完成的工作，并且要求各部门指定专人负责协调工作，使得企业上下均能认识到系统规划工作的重要性，并支持系统规划工作的开展。查阅资料并通过调研，掌握企业各个部门的主要的业务流程、各部门对当前信息系统的意见以及对新系统的期望。

（2）分析阶段。分析阶段主要包括对企业业务流程的分析和对数据类的分析。企业业务流程一般通过输入-处理-输出图进行表示，如图 1 所示。



图 1 企业业务流程

① 业务流程分析。业务流程是指在企业中，为完成某特定目标的一组有序活动的集合。业务流程分析是 BSP 方法的核心和关键。在构建系统总体结构时将以业务流程为基础。业务流程涉及到企业的多个部门，因此调研和分析工作量巨大，并且需要和具体的业务执行人员进行沟通，以确保过程的准确性、完整性和全面性。在对企业的现有的业务流程调研后，通过深入分析，将低效的、不合理的、重复的以及可以用计算机替代的业务流程进行优化。

② 数据类分析。数据类是指在业务执行过程中所需要和产生的数据。基于已经获得企

业业务流程，分析在业务流程各活动执行中，所需要和产生的数据，然后进行整理分析。

（3）构建阶段。基于分析后的企业业务流程和数据类，获得信息系统总体结构也就是子系统的划分。通过研究分析和评价，确定各子系统实施的优先级和先后顺序。对于企业贡献大、需求迫切、独立性较强的应当优先开发。当然，在确定优先顺序时还应考虑资源限制和约束条件。

（4） 报告撰写阶段。

1.1.2 关键成功因素法

对企业的成功具有关键作用的因素称为关键成功因素。各个行业、各个企业的成功因素是不完全相同的。在进行信息系统规划时，应当基于企业的关键成功因素识别出企业所需要的关键信息。关键成功因素法（Critical Success Factors，CSF）是通过企业的关键成功因素，确定信息需求和信息系统需求，从而进行系统规划的方法。

CSF 方法的主要步骤如图 2 所示：

- (1)分析并确定企业的目标。
- (2) 分析识别和企业目标相关的关键成功因素。首先识别出和企业目标相关的所有因素，然后对这些因素逐一进行分析和评价，确定关键成功因素。对于关键成功因素的子因素也应当分析识别。
- (3) 对关键成功因素进行分析，找出和各个关键成功因素有关的性能指标，然后确定和这些性能指标相关的信息需求。

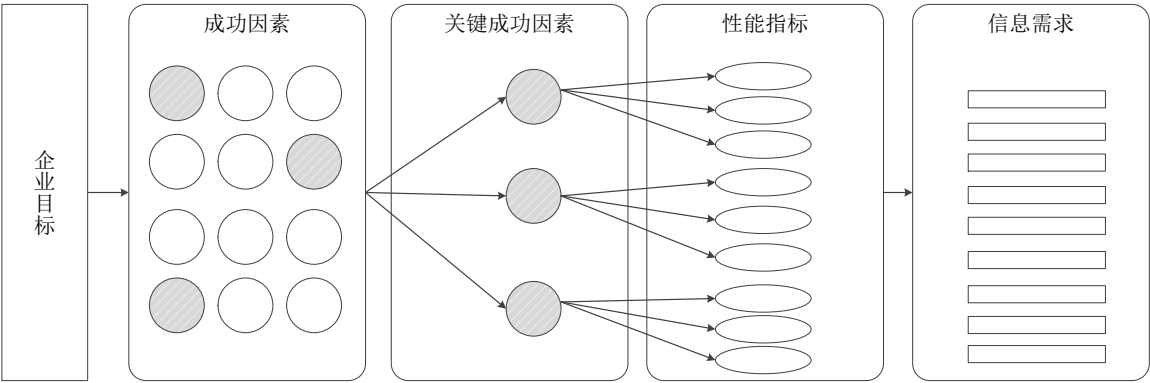


图 2 关键成功因素法步骤

1.1.3 战略目标集转化法

战略目标集转化法（Strategic Set Transformation ， SST）将组织的目标转换为信息系统的目标，从而获得信息系统的关键功能需求。

1.2 可行性分析

可行性分析是对项目在技术上和经济上是否可行所进行的科学分析和论证，是要用最小的代价在尽可能短的时间内确定问题是否能够解决。可行性分析的目的不是解决问题，而是要确定问题是否值得解决。

目前，在进行可行性分析时，多从如下方面进行研究判断，

- （1）技术可行性，现有的技术储备能否支撑整个项目的完成。
- （2）经济可行性，项目完成后的经济效益是否客观，能否超过投入的成本。
- （3）操作可行性，最终的用户群体能否适应项目的操作方式。

此外，一般还应该从法律/社会效益等更广泛的方面研究每种方案的可行性，还应该为每个可行的方案制定一个大致的实现进度。

可行性分析最根本的任务是为以后的行动方针提出建议。如果方案不可行，应及时中止项目的开展，以避免时间、资源、人力和物力的浪费；如果方案可行，应该从多个备选方案中选择出整体最优的解决方案。在进行可行性分析时，可以从如下方面展开，

（1）再次确认系统目标

通过访问关键人员，再次分析相关资料，确认系统规划阶段所提交的各类文档中的内容，改正存在的不确切或不清晰的表述，准确的说明目标系统中存在的限制和约束。

（2）研究已有的系统

对于已有的系统中存在的缺陷，需要能够在新系统中解决。新系统的运行维护费用不能高于已有系统，同时，已有系统的操作方式需要纳入到新系统的设计中，避免由于用户对已有系统的使用习惯变化而影响对新系统的接受。同时，新系统和已有系统的接口和数据关联方法要确定。

为了能够全面准确的获得已有系统的信息，在对已有系统的研究中，要阅读系统相关的文档，并进行实地操作与分析，同时与相关的人员进行深入沟通交流。

（3）分析出多个解决方案并评估其可行性

根据调查的结果，结合技术、经济、操作和法律等多个方面，判断是否应该继续项目的开发。同时，尽量提供多种方案，给出可行性分析的详细信息，向用户说明，双方人员共同确定最优的解决方案。

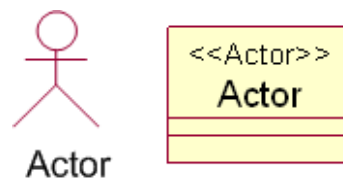
2、系统分析

2.1 用例图

用例是指描绘一个系统外在可见的需求情况，是代表系统中各个项目相关人员(风险承担人, **Stakeholder**)之间就系统的行为所达成的契约。开发人员首先捕获客户的需求，并以用例的形式组织成用例模型。然后分析并设计系统来满足这些用例，因此在用例模型之后就是分析模型，接着是设计模型和实施模型。在实现了整个系统之后，还将根据用例模型设计出测试模型来对系统进行验证

这些模型之间并不是线性转变的，它们是一个迭代、增量的开发过程。也就是在整个项目开发周期中，将会多次经过这五个模型的迭代，每次都将越来越精化。

- 参与者是为了完成一个事件而与系统交互的实体，是用户相对系统而言所演的角色



- 用例实例是在系统中执行的一系列动作，这些动作将生成特定参与者可见的价值结果。一个用例定义一组用例实例
- 用例是由一组用例实例组成的，用例实例也就是常说的“使用场景”，就是用户使用系统的一个实际的、特定的场景



- 图中的元素包括：参与者、用例、一个方框和一些表示关系的连接线
- 所有的用例都位于方框之内，该方框称为“系统边界”
- 参与者与用例的关系：在参与者和用例之间的关联是用一根带箭头的线来表示的
- 用例之间的关系：
 - 1) 包含关系
 - 2) 扩展关系
 - 3) 泛化关系

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]

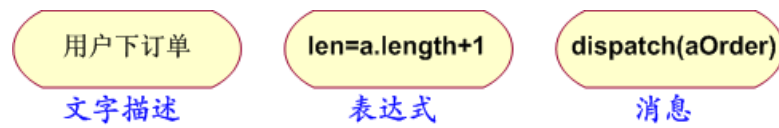
前置条件	[即启动该用例所应该满足的条件。]	
后置条件	[即该用例完成之后，将执行什么动作。]	
成功保证	[描述当前目标完成后，环境变化情况。]	
基本事件流	步骤	活动
	1	[在这里写出触发事件到目标完成以及清除的步骤。]
	2(其中可以包含子事件流，以子事件流编号来表示)
扩展事件流	1a	[1a 表示是对 1 的扩展，其中应说明条件和活动]
	1b(其中可以包含子事件流，以子事件流编号来表示)
子事件流	[对多次重复的事件流可以定义为子事件流，这也是抽取被包含用例的地方。]	
规则与约束	[对该用例实现时需要考虑的业务规则、非功能需求、设计约束等]	

2.2 活动图

活动图（activity diagram，动态图）是阐明了业务[用例](#)实现的工作流程。业务工作流程说明了业务为向所服务的业务主角提供其所需的价值而必须完成的工作。业务用例由一系列活动组成，它们共同为业务主角生成某些工件。工作流程通常包括一个基本工作流程和一个或多个备选工作流程。工作流程的结构使用活动图来进行说明。

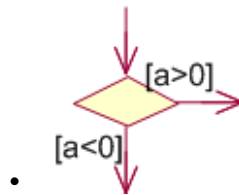
初始节点和活动终点：用一个实心圆表示初始节点，用一个圆圈内加一个实心圆来表示活动终点

活动节点：是活动图中最主要的元素之一，它用来表示一个活动

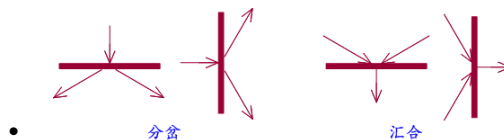


转换：当一个活动结束时，控制流就会马上传递给下一个活动节点，在活动图中称之为“转换”，用一条带箭头的直线来表示

分支与监护条件：分支是用菱形表示的，它有一个进入转换（箭头从外指向分支符号），一个或多个离开转换（箭头从分支符号指向外）。而每个离开转换上都会有一个监护条件，用来表示满足什么条件的时候执行该转换。



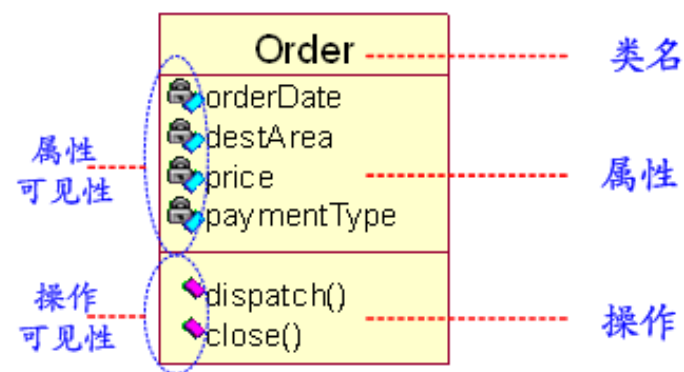
分岔与汇合：



2.3 类图

类图(Class diagram)是显示了模型的静态结构，特别是模型中存在的类、类的内部结构以及它们与其他类的关系等。类图不显示暂时性信息。

- 名称：每个类都有一个惟一的名称，通常采用 CamelCase 格式表示
- 属性：是已被命名的类的特性，它描述该类实例中包含的信息
- 操作：是类所提供的服务，它可以由类的任何对象请求以影响其行为
- 属性名和操作名也通常采用 CamelCase 格式表示，只不过首字母通常为小写。

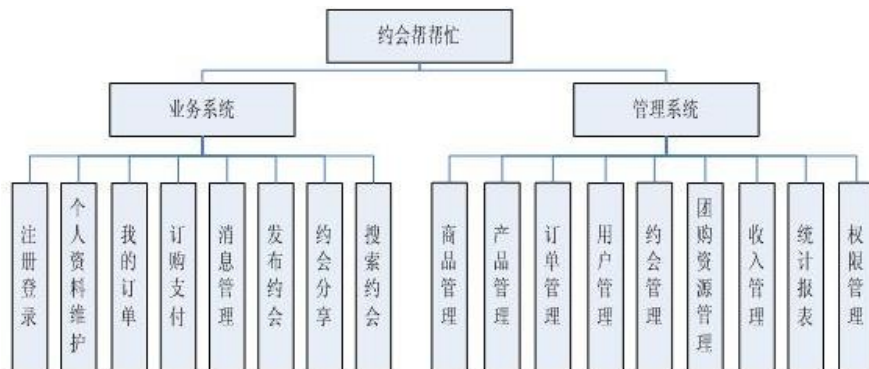


3、系统设计

3.1 总体设计

3.1.1 功能结构图

功能结构图就是将系统的功能进行分解，按功能从属关系表示的图表。管理信息系统的各子系统可以看作是系统目标下层的功能，对其中每项功能还可以继续分解为第三层、第四层……甚至更多的功能。



3.1.2 架构图

信息系统中的数据与程序的分布方式被成为计算模式。在信息系统中，最为常用三种计算模式是客户机/服务器计算模式、浏览器/服务器计算模式。

在信息系统部署时，可以将数据库层、业务逻辑层等分开部署，如果业务规则复杂，计算量大，可以将其部署在高性能服务器上，而如果对数据库读取频繁，那么可以提高数据库服务器的性能。

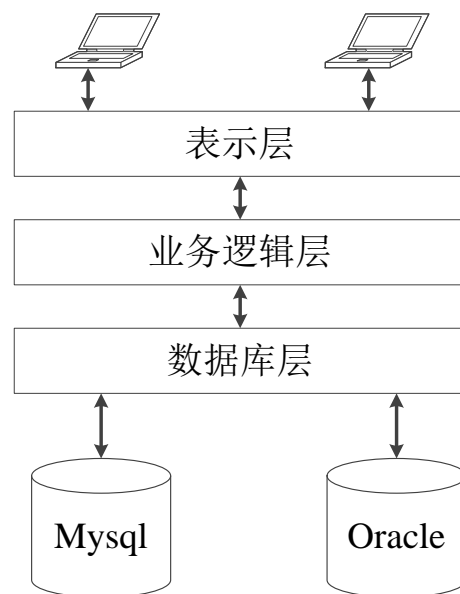


图 浏览器/服务器计算模式

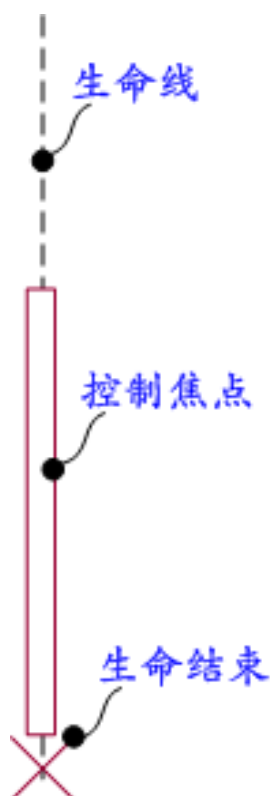
3.2 详细设计

3.2.1 顺序图

顺序图是将交互关系表示为一个二维图。纵向是时间轴，时间沿竖线向下延伸。横向轴代表了在协作中各独立对象的类元角色。类元角色用生命线表示。当对象存在时，角色用一条虚线表示，当对象的过程处于激活状态时，生命线是一个双道线。

对象与角色：最顶上一排矩形框。在交互图中，参与交互的对象既可以是具体的事物，又可以是原型化的事物。作为具体的事物，一个对象代表现实世界中的某个东西。

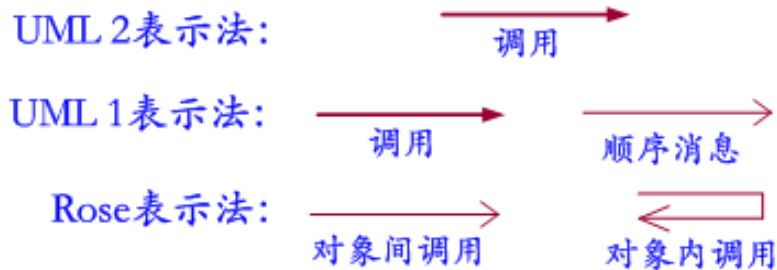
生命线与控制焦点：每个对象都有自己的生命线，对象生命线是一条垂直的虚线，用来表示一个对象在一段时间内存在。



消息：用来描述对象之间所进行的通信的，该信息带有对将要发生的活动的期望。当传送一个消息时，它所引起的动用是一个通过对计算过程的抽象而得到的可执行语句。

消息分为五种：调用、返回、发送、创建和销毁

调用：表示调用某个对象一个操作



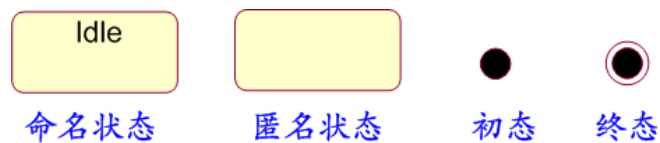
顺序编号：整个消息的传递过程就形成了一个完整的序列，因此通过在每个消息的前面加上一个用冒号隔开的顺序号来表示其顺序。

3.2.2 状态图

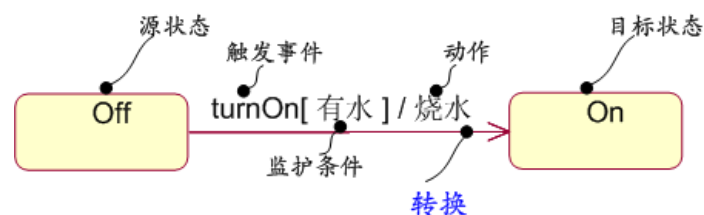
状态图(Statechart Diagram)是描述一个实体基于事件反应的动态行为，显示了该实体如何根据当前所处的状态对不同的事件做出反应。

状态是指在对象生命周期中满足某些条件、执行某些活动或等待某些事件的一个条件和状况

一个状态通常包括名称、进入/退出活动、内部转换、子状态和延迟事件等五个部分组成



最为核心的元素无外乎是两个：一个是用圆角矩形表示的状态（初态和终态例外）；另一个则是在状态之间的、包含一些文字描述的有向箭头线，这些箭头线称为转换



源状态：即受转换影响的状态

目标状态：当转换完成后对象的状态

触发事件：用来为转换定义一个事件，包括调用、改变、信号、时间四类事件

监护条件：布尔表达式，决定是否激活转换、

动作：转换激活时的操作

3.2.3 ER 图和范式

(1) ER 图

E-R 图也称实体-联系图(Entity Relationship Diagram)，提供了表示实体类型、属性和联系的方法，用来描述现实世界的[概念模型](#)。它是描述现实世界概念结构模型的有效方法。是表示概念模型的一种方式，用矩形表示实体型，矩形框内写明实体名；用椭圆表示实体的属性，并用无向边将其与相应的实体型连接起来；用菱形表示实体型之间的联系，在[菱形](#)框内写明联系名，并用无向边分别与有关实体型连接起来，同时无向边旁标上联系类型

(1:1, 1:n 或 m:n)。

实体在 E-R 图中用矩形表示，矩形框内写明实体名。属性在 E-R 图中用椭圆形表示，并用无向边将其与相应的实体连接起来；联系在 E-R 图中用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时无向边旁标上联系类型 (1 : 1, 1 : n 或 m : n)

(2) 数据库范式

设计关系[数据库](#)时，遵从不同的规范[要求](#)，设计出合理的关系型数据库，这些不同的规范要求被称为不同的范式，各种范式呈递次规范，越高的范式数据库冗余越小。目前[关系数据库](#)有六种范式：第一范式 (1NF)、第二范式 (2NF)、第三范式 (3NF)。

第一范式 (1NF) 指数据库表的每一列都是不可分割的原子数据项，而不能是集合，数组，记录等非原子数据项。即实体中的某个属性有多个值时，必须拆分为不同的属性。

第二范式 (2NF) 指在 1NF 的基础上，非码属性必须完全依赖于候选码 (在 1NF 基础上消除非主属性对主码的部分函数依赖)

第二范式 (2NF) 要求实体的属性完全依赖于主关键字。所谓完全依赖是指不能存在仅依赖主关键字一部分的属性，如果存在，那么这个属性和主关键字的这一部分应该分离出来形成一个新的实体，新实体与原实体之间是一对多的关系。为实现区分通常需要为表加上一个列，以存储各个实例的唯一标识。简而言之，第二范式就是在第一范式的基础上属性完全依赖于主键。

第三范式（3NF）指在 1NF 基础上，任何非主属性不依赖于其它非主属性（在 2NF 基础上消除传递依赖）。

4、界面设计与程序开发

4.1 输入设计

输入设计的主要任务是根据业务要求，确定适当的输入形式，使信息系统获取业务执行过程中所需要的信息。输入设计的目的是提高输入效率，减少输入错误。

4.1.1 输入设计的原则

（1）降低用户的输入量

是指控制用户的输入量，而非是控制数据的输入量。数据的输入是信息系统运行的基础，只有进行大量的数据输入，信息系统才能够有效发挥其价值。

在用户输入时，系统大部分时间将处于等待状态，并且耗费大量的人力。因此，应当缩短输入时间，优先选择由系统自动输入或者系统辅助用户进行半自动化的输入。例如，在进行基础数据输入时，可以使用数据文件批量导入的方式自动的完成数据的输入。对于必须用户输入的数据，则可以根据用户输入的内容自动的进行补充或推荐，由用户进行选择输入内容，一方面可以提高输入效率，同时也降低了输入错误的可能性。

（2）提高输入的反应速度

在用户输入时，应当及时的对输入的结果进行反馈，以能够知晓是否输入正确，或者是否已经输入完毕。输入反应不及时将有可能导致数据的重复输入，当前经常对输入数据进行校验，校验的速度也影响着输入的反应速度。特别是在移动终端上录入数据时，反应速度受到终端设备、网络等的影响，因此在进行输入设计时，还应当考虑到不同设备的特点。

（3）提高输入的正确率

在输入设计中，应当从多个方面入手，提高用户输入的正确性。例如，尽量多的多采用选择而非填写的方式进行数据的输入。在必须的数据输入时，应当提供完备高效的校验方法，尽可能的降低输入错误。针对不同的输入设备，应当设计有针对性的校验方法。例如，在传统计算机上，可以使用二次校验方式，而在

手持终端上，由于不易输入，可以使用明码的方式进行静态校验。

（4）提高输入的效率

在输入设计时，应当尽量降低不必要的输入内容和步骤。同时，考虑到不同类型用户的需求，设计个性化的易于输入的方式，从多方面提高输入的效率。例如，在输入身份证信息后，应当可以自动的获得省份、城市、性别以及出生日期，而无需用户再次的重复输入。

在进行输入设计时，还应当研究已有系统以及常用软件的输入方式，尽量和用户的习惯保持一致，进一步提高用户的输入效率。

4.1.2 数据输入设备的选择

（1）键盘

通过键盘输入数据是最常使用的输入方式，也是最容易出错的输入方式，多用于用户的直接输入。

（2）鼠标

通过点击界面的控件完成数据的输入。由于多是通过选择的方式进行数据输入，相比键盘输入，可以提高数据输入的效率 and 准确性。

（3）语音

语言包括两种，一种是按键声音，一种是语言表达。按键声音多是来自电话的输入，而语言表达则依赖于语音识别技术。目前由于语音识别技术的准确性需要进一步提高，因此语言表达类的语音输入尚未普及。

（4）光标记

数据以格式化的字符或孔存储在光扫描纸上。通过专用的光学识别设备，对相应的光扫描纸进行识别进而完成数据输入。

（5）生物识别

生物识别主要是根据唯一的人类特征来进行。从最初的指纹识别，到目前常用的人脸识别、视网膜识别等，均得到了广泛的应用。

（6）触摸屏

新兴的输入方式，在移动终端的输入中获得了广泛的应用。由于可以手写、通过输入法录入以及直接选择，无需过多的电脑技能，非常适用于对计算机不熟悉人员的数据输入。

4.1.3 输入检验

为了降低错误的输入信息，避免将错误的数据存储到系统中，在输入数据后，应当进行输入数据的校验。在进行输入校验时，可以分为前端校验和后端校验两种方式，前端校验主要是进行初步的校验，包括格式、位数是否正确，而不考虑是否符合业务逻辑。后端校验是在满足前端校验的基础上，通过与数据库交互进行校验，主要是判断该输入是否符合业务逻辑规则。

常见的前端校验方法包括二次输入校验法、静态校验法、值域校验法、格式校验法等，具体如表所示。

表前端校验方法

名称	内容	适用范围	举例	说明
重 复 输 入 校 验 法	将同一数据重复输入，判断多次的输入是否一致。	较重要的数据输入，如账户密码等。	如注册账号时需要将密码填写两次。	在移动终端不建议使用二次输入校验，由于移动终端不易于输入，容易输入错误，多使用明码校验。
静 态 校 验法	待数据输入后，通过静态观察的方法确认是否正确。	较简单或逻辑性较弱的数据输入。	在填写注册表后，将所有录入信息弹出，让用户确认。	静态校验法多是根据用户的输入信息，产生确认界面，由用户再次确认输入信息。
值 域 校 验法	通过判断输入数据是否超出了值域。	存在上限和下限的数据。	例如，输入13月，则输入有误。	在进行值域校验法时，特别要注意逻辑关系，譬如闰年和平年的二月的天数是不同的。

格 式 校 验法	从输入数据的格 式、类型和长度 等进行校验。	大多数数据。	姓名类型为 字符型，其 他类型为错 误。手机号 码为 11 位， 其他长度为 错误。	特别要注意校 验汉字中的全 角字符、以及中 英文字符的区 别。要考虑升级 的情况，例如固 定电话位数升 级，将有可能带 来长度校验法 的失效。
-------------	------------------------------	--------	--	--

常见的后端校验方法包括平衡校验法、逻辑校验法等，这些校验方法的具体情况如表所示。

表后端校验方法

名称	内容	适用范 围	举例	说明
平衡校验法	根据数量关系来检查 输入数据的正确性。	存在数量关 系的数据。	譬如进货 数量与库 存数量的 差应当和 销售数量 一致。	该方法的 应用将有 可能要求 用户重复 输入数据。
逻辑校验法	检查输入的数据是否 符合逻辑。	存在有效性的 数据。	譬如输入 物料编码 查询物料 信息。	该校验方 法由于需 要操作数 据库，因此 应当尽量 在必要的 时候使用， 降低服务

				器的负担。
--	--	--	--	-------

4.2 输出设计

输出设计的目的是根据用户的信息需求，将用户所需要的信息，及时、准确的提供给用户。信息的输出是用户使用信息系统的主要目的，输出设计直接关系到用户对系统的接受程度。

4.2.1 输出设计的原则

（1）提高可阅读性和可理解性

应当尽可能的提高输出信息的可阅读性和可理解性。包括多使用各种图表对输入信息进行展示，以能够使用户快速直观的获取所需要的信息。对于输出信息的描述信息，应当符合用户的习惯，对于必要的缩写词语的使用，要及时的提供全称，便于用户的理解。

（2）提高输出信息的准确性

在进行输出设计时，特别当数据量大时，应当把相关的信息放在一起进行输出展示，避免由于输出信息的不完整，降低了输出信息的准确性。在进行大数据量的信息输出时，应当将待输出的信息分为多个独立的模块，而不是将所有的信息一并展示。

（3）提高输出信息的灵活性

避免将所有的信息千篇一律的向所有用户展示，允许用户根据需求，定制输出的信息内容和展示方式，能够实现所看到的信息就是所需要的信息，避免用户在输出的信息中进行二次过滤查询。

（4）提高输出信息的友好性

对于输出的信息，要能够考虑到不同用户的需求，对于不易理解的信息，可以使用多种方式从多个角度进行输出展示。输出的设计要符合用户的习惯，同时兼顾不同操作系统，不同终端设备的需求。

4.1.2 输出设备的选择

（1）屏幕

最为常用的输出设备，将输出结果显示在屏幕上，由用户选择下一步的操作。但是在屏幕上显示的输出信息是暂时的，当信息离开屏幕时，信息将随之消失。

（2）打印机

打印机多是输出表格。该方式输出的信息，将较长时间内保存在介质上。

（3）多媒体

包括图像、声音和图片等，多是作为屏幕输出的补充。通过多媒体可以使用户更快速的掌握输出信息。

（4）磁盘

磁盘容量大，复制方便，易用长期保存，是目前常用的一种输出设备。

（5）绘图仪

相比打印机，绘图仪精度高，更适用于各类设计图纸的精确输出。

4.1.3 用户界面设计

用户界面是指用户和计算机交互的窗口。通过用户界面，实现信息的计算机表现形式和用户理解接受形式之间的转换。

（一）用户界面设计的原则

（1）一致性原则

在进行用户界面设计时，要使得整个系统的色彩布局、操作模式、文字大小等保持一致。该一致性不仅仅指系统内部的一致性，还包括新系统和企业常用系统、常用软件和行业软件的一致性。这样可以降低用户的学习接受成本，减少操作失误。因此在进行用户界面设计时，要深入分析企业已有系统、常用软件、行业软件的界面设计，在此基础上完成新系统的界面设计。例如，文字处理模块的界面设计要和 OFFICE 软件相似，使得用户无需专门学习就可以直接掌握操作。

（2）层次性原则

要考虑主菜单和二级三级菜单的层次关系，使得用户能够快速识别多窗口之间的关系。对于界面中各元素的展示方式和展示位置，需要结合用户访问的频率。例如，对于经常跳转的两个界面，应当是用户无需寻找定位，直接可以进行界面的切换。特别是在涉及业务流程顺序、查看顺序等具有明显前后层次关系的界面设计中，要能够通过界面设计，体现出各界面之间的逻辑关系。

（3）用户至上的原则

仔细分析用户的类型、用户的特征、用户的需求，要针对不同类型的用户，进行相应的界面设计。同时，在界面设计过程中，要不断的提出原型，供用户有针对性的提出意见，持续的优化界面设计。在此过程中，也可以有针对性的培养

用户的使用习惯，提高用户对于界面的熟悉程度，以利于后续的信息系统实施。

5、 系统测试

系统测试包括功能测试、性能测试等方面。在进行测试时，要编写测试用例，以便于测试的实施。

5.1 功能测试用例编写原则

（1）单元测试功能用例的编写目的

单元测试用例的目的在于验证单个模块是否达到了详细设计说明书中规定的功能，由于是单个模块所以无法检验关联性，可能会牵扯到数据库的操作，例如：删除时，需要查看数据库是否完全删除了数据。

（2）集成测试功能用例的编写目的

集成测试功能用例的目的在于验证软件连接时，模块的连接是否正确（及数据的传递是否正确）。我们的软件中体现出来的是，是否正确调用界面，界面之间显示的数据是否正确，特别是财务、费用、数据方面的。

集成测试用例的编写过程中，经常将功能用例与业务流程混合编写，因为在集成测试时需验证业务流程中的数据正确性，以及界面之间的数据传递的准确无误。

（3）系统测试业务流程用例的编写目的

系统测试业务流程用例的目的在于验证软件最终数据的准确性，我们的软件体现为，手工数据与报表数据的一直性。用例与用例之间有着一定的关系，目的性十分明确。

5.2 测试用例设计的原则

5.2.1 全面性

指编写的测试用例应该覆盖所有的“概要设计文档”或“需求文档”以及“测试申请文档”中描述的功能。

（1）数据库程序基本的增、删、改功能

增、改测试用例重点在于数据合法性、正确性的检验和提示信息正确性的检验。输入的数据可能有无限种组合，此时可以采用等价类划分和边界值法，下面有较详细的说明。

删除的测试用例比较简单，只有操作没有数据的输入，但是应该在“备注”中注明，删除的限制条件，以及数据库中应该删除的表的情况，有条件限制时，测试用例应该包含各种删除条件，必要时在添加或修改的测试用例后面或中间紧跟删除的测试用例。

(2) 对于无输入的操作，应该详细描述其具体的操作步骤和结果

例如：选择“用户帐号”，可以通过多种途径进行，此时应具体描述程序从何处进入，通过何种操作，达到查看“用户帐号”界面。对于报表的测试用例，最好紧跟在输入数据的后面，并且应该给出报表输出的数据的界面图(含数据)。

对于不便书写测试用例的情况，应该在备注中说明，并写出可能的操作步骤。例如：对于文件夹的拖动，说明左右拖还是上下拖，结果如何就可以了。

(3) 单元测试用例的书写是使用一条数据，多种可能的情况考虑。但是对于其余各阶段的测试用例，必须考虑多条数据时的情况，此时主用是针对新增多条数据后，进行删、改、拖等情况的考虑。

(4) 针对有结算的测试用例，数据的应考虑存在跨年、跨月的数据。

5.2.2 正确性

包括数据的正确性和操作的正确性。

首先保证测试用例的数据正确。其次预期的输出结果应该与测试数据发生的业务吻合。操作的预期结果应该与业务流程在程序执行的结果吻合。

5.2.3 符合正常业务惯例

测试数据应符合用户实际工作业务流程，实际就是测试用例的先后顺序，先新增，后修改或删除，不能将删除放在第一位。

5.2.4 仿真性

人名、地名、电话号码等应具有模拟功能，符合一般的命名惯例；

5.2.5 可操作性

测试用例中应写清测试的操作步骤，不同的操作步骤相对应的操作结果不同，达到的是，任何人，均可以根据测试用例，单独进行测试。

5.3 测试用例设计的方法

测试用例的核心是输入数据。预期输出是依据输入数据和程序功能来确定的，也就是说，对于某一程序，输入数据确定了，预期输出也就可以确定了，至于生成/销毁被测试对象和测试的方法等，是所有测试用例都大同小异的，一般只有操作而没有输入。因此，我们讨论测试用例设计方法时，重在输入数据。

输入数据包括四类：参数、成员变量、全局变量、IO 媒体。IO 媒体是指文件、数据库或其他储存或传输数据的媒体，例如，被测试函数要从文件或数据库读取数据，那么，文件或数据库中的原始数据也属于输入数据。

这四类数据中，只要所测试的程序需要执行读操作的，就要设定其初始值，显然，把输入数据的所有可能取值都进行测试，是不可能也是无意义的，我们应该用一定的规则选择有代表性的数据作为输入数据，主要有三种：正常输入、边界输入、非法输入。每种输入还可以分类，也就是平常说的等价类法，每类取一个数据作为输入数据，如果测试通过，可以肯定同类的其他输入也是可以通过的。下面举例说明：

（1）正常输入

例如“字符串 Trim 函数”，功能是将字符串前后的空格去除，那么正常的输入可以有四类：前面有空格；后面有空格；前后均有空格；前后均无空格。

（2）边界输入

上例中空字符串可以看作是边界输入。

再如一个表示年龄的参数，它的有效范围是 0-100，那么边界输入有两个：0 和 100。

（3）非法输入

非法输入是正常取值范围以外的数据，或使代码不能完成正常功能的输入，如上例中表示年龄的参数，小于 0 或大于 100 都是非法输入，再如一个进行文件操作的函数，非法输入有这么几类：文件不存在；目录不存在；文件正在被其他程序打开；权限错误。

如果函数使用了外部数据，则正常输入是肯定会有，而边界输入和非法输入不是所有函数都有。一般情况下，即使没有设计文档，考虑以上三种输入也可以找出函数的基本功能点。实际上，一个合格的功能模块测试，编码时对上述三种输入都是必须考虑的，否则代码的健壮性就会成问题。

5.4 测试用例编写格式

（1）用例编号

测试用例编号是由字母和数字组合而成的，用例的编号应该具有唯一性，易识别性，比如可以采用统一的约定，产品编号_ST_系统测试项名_系统测试子项名_编号。这样看到编号就可以知道是做的什么测试，测试的对象是什么，也方便维护。

（2）测试项目

现在这个测试用例所测的项目名，可以是测试用例所属的大类，被测需求，被测的模块，或者是被测的单元。例如：计算器加法功能

（3）用例标题

测试标题是对测试用例的简单描述。用概括的语言描述该测试用例的测试点。每个测试用例的标题不能够重复，因为每个测试用例的测试点是不一样的。例如：手机在没有 SIM 卡的情况下，拨打 119。

（4）重要级别

重要级别分为高中低三等：

高：保证系统基本功能、重要特性、实际使用频率比较高的用例；

中：重要程度介于高和低之间的测试用例；

低：实际使用频率不高，对系统业务功能影响不大的模块或功能的测试用例。

注：一般情况下，重要级别为高的测试用例，一个测试子项里有且仅有一个，大多数都是重要级别为中的测试用例。因为一般我们会进行一个系统测试预测试项，如果重要级别为高的太多，则就失去了预测试的实际意义。

（5）预置条件

就是执行当前测试用例的前提描述，如果不满足这些条件，则无法进行测试

（6）测试输入

测试用例执行时，需要输入的外部信息。例如：某一个文件，数据记录等

（7）操作步骤

执行当前测试用例所要经过的操作步骤，需要给出每一步操作的详细描述，测试人员根据测试用例操作步骤，完成测试用例的执行

（8）预期结果

当前测试用例的预期输出结果，用来与实际结果比较，如果相同则该测试用例通过，否则该测试用例失败。

（9）作者：谁写的

（10）创建日期：写用例的日期

（11）修改日期：最后一次修改用例的日期

（12）测试结果：执行用例后的结果 Pass、Fail、Block

5.5 测试用例举例

此功能测试用例着重于站在用户的角度对测试对象的功能测试。应侧重于所有可直接追踪到用例或业务功能和业务规则的测试需求。这种测试的目标是核实数据的接受、处理和检索是否正确，以及业务规则的实施是否恰当。主要测试技

术方法为用户通过 GUI（图形用户界面）与应用程序交互，对交互的输出或接受进行分析，以此来核实需求功能与实现功能是否一致。

项目名称	XXXX 系统			程序版本			
模块名称	登录/退出			子模块名称			
编制人		编制时间	2010-7-22	执行人员		执行时间	
测试目的及描述	测试 XXXX 系统客户端软件在正常、异常情况下的用户登录功能以及系统退出功能						
预置条件	启动服务器端的系统服务						
用例编号	用例说明	输入数据（步骤）		预期结果	测试结果	缺陷编号	备注 (发现日期, 严重程度)
DL-01	用户正常登录	启动客户端软件，在用户登录界面输入正确的用户名，密码和数据库名，点击“登陆”。		系统验证用户名与密码后，进入系统初始化界面，初始化通过后进入业务管理系统主界面。	通过		
DL-02	用户异常登录	1. 启动客户端软件，在用户登录界面输入不匹配的用户名和密码，输入正确的数据库名，点击“登陆”。 2. 启动客户端软件，在用户登录界面输入正确的用户名和密码，输入不正确的数据库名，点击“登陆”。		1. 系统不能启动，提示用户名与密码不正确。 2. 系统不能启动，提示数据库名不正确。	不通过	WTHG-01	
DL-03	登录取消	启动客户端软件，在用户登录界面输入用户名，密码和数据库名，点击“取消”		用户登录界面自动关闭，系统不能启动。	通过		
DL-04	界面规范性测试——【用户名】	1. 【用户名】输入为空，点击“登录”。 2. 【用户名】输入超长字符串，点击“登录”。		1. 系统不能启动。 2. 系统对输入框的输入长度有限制。 3. 系统能够自动	不通过	WTHG-02	

		3. 【用户名】输入包含空格的字符串，密码和数据库名正确，点击“登录”。	过滤字符串首尾的空格，并成功登录系统; 对于字符串中间包含空格的用户名，无法登录。			
DL-05	界面规范性测试——【密码】	1. 【用户名】输入正确，【密码】输入为空，点击“登录”。 2. 【用户名】输入正确，【密码】输入超长字符串，点击“登录”。 3. 【用户名】输入正确，【密码】输入包含空格的字符串，点击“登录”。	1. 系统不能启动。 2. 系统对输入框的输入长度有限制。 3. 系统不能启动。	不通过	WTHG-03	
DL-06	退出系统	点击系统窗口右上角的按钮 	用户退出系统，该用户占用的资源被释放。	通过		
DL-07	随机界面测试	对登录/退出界面随机测试	系统不会出现异常	不通过	WTHG-03	