版次：

《软件工程基础》大作业

对《成绩管理系统》的开发文档（模拟）

信软学院软件工程课程组

2023年10月

**本文档的版本记录**

本文件的版本变动情况如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版次 | 修改原因 | 修改内容 | 修改人 | 日期 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

人员分工

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 主要贡献 | 角色 |
|  |  |  | 组长 |
|  |  |  | 组员 |
|  |  |  | 组员 |
|  |  |  | 组员 |

# 目录

[附录A 软件需求分析报告文档模板](#_引言) 1

[附录B 软件概要设计报告文档模板](#附录B)15

[附录C 软件详细设计报告文档模板](#附录C) [26](#_Toc127782083)

[附录D 软件数据库设计报告文档模板](#附录D)（可选） [33](#_Toc127782083)

[附录E 软件测试](#附录E)报告文档模板 [43](#_Toc127782083)

（提示：请用户根据项目的实际状况，对本报告模板中的要素进行裁剪或增添）

# 附录A 软件需求分析报告文档模板

# 引言（引言部分去掉注释文字，不超过2000字）

提示：请用户根据项目的实际需求状况，裁剪本需求分析报告模板。

***本模板中的蓝色部分是提示性或解释性文字，写作时应删除或替换成实际内容！***

引言是对这份软件产品需求分析报告的概览，是为了帮助阅读者了解这份文档是如何编写的，并且应该如何阅读、理解和解释这份文档。

## 编写目的

本节说明这份软件产品需求分析报告是为哪个软件产品编写的，开发这个软件产品意义、作用、以及最终要达到的意图。通过这份软件产品需求分析报告详尽说明了该软件产品的需求规格，包括修正和（或）发行版本号，从而对该软件产品进行准确的定义。

如果这份软件产品需求分析报告只与整个系统的某一部分有关系，那么只定义软件产品需求分析报告中说明的那个部分或子系统。

## 软件预期用户

## 预期读者和阅读建议

本部分列举本软件报告所针对的各种不同的预期读者，例如，可能包括：

* 用户；
* 开发人员；
* 项目经理；
* 营销人员；
* 测试人员；
* 文档编写人员。
* （其他人员）

并且描述了文档中，其余部分的内容及其组织结构，并且针对每一类读者提出最适合的文档阅读建议。

## 产品范围

说明该软件产品及其开发目的的简短描述，包括利益和目标。把软件产品开发与企业目标，或者业务策略相联系。

描述产品范围时需注意，可以参考项目视图和范围文档，但是不能将其内容复制到这里。

## 文档约定

描述编写文档时所采用的标准（如果有标准的话），或者各种排版约定。排版约定应该包括：

* 正文风格；
* 提示方式；
* 重要符号；

也应该说明高层次需求是否可以被其所有细化的需求所继承，或者每个需求陈述是否都有其自己的优先级。

## 术语与缩写解释

|  |  |
| --- | --- |
| 缩写、术语 | 解 释 |
| SPP | 精简并行过程，Simplified Parallel Process |
|  |  |
|  |  |
|  |  |
| … |  |

## 参考文献

列举编写软件产品需求分析报告时所用到的参考文献及资料，可能包括：

* 本项目的合同书；
* 上级机关有关本项目的批文；
* 本项目已经批准的计划任务书；
* 用户界面风格指导；
* 开发本项目时所要用到的标淮；
* 系统规格需求说明；
* 使用实例文档；
* 属于本项目的其它己发表文件；
* 本软件产品需求分析报告中所引用的文件、资料；
* 相关软件产品需求分析报告；

为了方便读者查阅，所有参考资料应该按一定顺序排列。如果可能，每份资料都应该给出：

* 标题名称；
* 作者或者合同签约者；
* 文件编号或者版本号；
* 发表日期或者签约日期；
* 出版单位或者资料来源。

# 综合描述

这一部分概述了正在定义的软件产品的作用范围以及该软件产品所运行的环境、使用该软件产品的用户、对该软件产品己知的限制、有关该软件产品的假设和依赖。

## 产品的定位

描述了在软件产品需求分析报告中所定义的软件产品的背景和起源。说明了该软件产品是否属于下列情况：

* 是否是产品系列中的下一成员；
* 是否是成熟产品所改进的下一代产品；
* 是否是现有应用软件的替代品（升级产品）；
* 是否是一个新型的、自主型的产品。

如果该软件产品需求分析报告定义的软件系统是：

* 大系统的一个组成部分；
* 与其它系统和其它机构之间存在基本的相互关系。

那么必须说明软件产品需求分析报告定义的这部分软件是怎样与整个大系统相关联的，或者（同时）说明相互关系的存在形式，并且要定义出两者之间的全部接口。

## 产品的主要功能概述

因为将在需求分析报告的第4部分中详细描述软件产品的功能，所以在此只需要概略地总结。仅从业务层面陈述本软件产品所应具有的主要功能，在描述功能时应该针对每一项需求准确地描述其各项规格说明。如果存在引起误解的可能，在陈述本软件产品主要功能的作用领域时，也需要对应陈述本软件产品的非作用领域，以利读者理解本软件产品。

为了很好地组织产品功能，使每个读者都容易理解，可以采用列表的方法给出。也可以采用图形方式，将主要的需求分组以及它们之间的联系使用数据流程图的顶层图或类图进行表示，这种表示方法是很有用的。

参考用户当前管理组织构架，了解各个机构的主要职能，将有助于陈述软件产品的主要功能。

## 运行环境

描述了本软件的运行环境，一般包括：

* 硬件平台；
* 操作系统和版本；
* 支撑环境（例如：数据库等）和版本；
* 其它与该软件有关的软件组件；
* 与该软件共存的应用程序。

## 设计和实现上的限制

确定影响开发人员自由选择的问题，并且说明这些问题为什么成为一种限制。可能的限制包括下列内容：

* 必须使用的特定技术、工具、编程语言和数据库；
* 避免使用的特定技术、工具、编程语言和数据库；
* 要求遵循的开发规范和标准

例如，如果由客户的公司或者第三方公司负责软件维护，就必须定义转包者所使用的设计符号表示和编码标准；

* 企业策略的限制；
* 政府法规的限制；
* 工业标准的限制；
* 硬件的限制

例如，定时需求或存储器限制；

* 数据转换格式标淮的限制。

# 外部接口需求

通过本节描述可以确定，保证软件产品能和外部组件正确连接的需求。关联图仅能表示高层抽象的外部接口，必须对接口数据和外部组件进行详细描述，并且写入数据定义中。如果产品的不同部分有不同的外部接口，那么应该把这些外部接口的全部详细需求并入到这一部分实例中。

注意：必须将附加用户类的特征与外部接口需求加以区分，附加用户类的特征描述的是通过接口取得软件产品的数据和服务的人的需求；而外部接口需求描述的是接口本身的需求。

## 用户界面需求

陈述需要使用在用户界面上的软件组件，描述每一个用户界面的逻辑特征。必须注意，这里需要描述的是用户界面的逻辑特征，而不是设计好的用户界面。以下是可能包括的一些特征：

* 将要采用的图形用户界面（GUl）标准或者产品系列的风格；
* 有关屏幕布局或者解决方案的限制；
* 将要使用在每一个屏幕（图形用户界面）上的软件组件，可能包括：
  + 选单；
  + 标准按钮；
  + 导航链接；
  + 各种功能组件；
  + 消息栏；
* 快捷键；
* 各种显示格式的规定，可能包括：
  + 不同情况下文字的对齐方式；
  + 不同情况下数字的表现格式与对齐方式
  + 日期的表现方法与格式；
  + 计时方法与时间格式；
  + 等等。
* 错误信息显示标准；

对于用户界面的细节，例如：一个特定对话框的布局，应该写入具体的用户界面设计说明中（在附录C，详细设计），而不能写入软件需求规格说明中。

如果采用现成的、合适的用户界面设计规范（标准），或者另文描述，可以在这里直接说明，并且将其加入参考文献。

## 硬件接口

描述待开发的软件产品与系统硬件接口的特征，若有多个硬件接口，则必须全都描述。接口特征的描述内容可能包括：

* 支持的硬件类型；
* 软、硬件之间交流的数据；
* 控制信息的性质；
* 使用的通讯协议；

## 软件接口

描述该软件产品与其它外部组件的连接，这些外部组件必须明确它们的名称和版本号以资识别，可能的外部组件包括：

* 操作系统；
* 数据库；
* 工具；
* 函数库；
* 集成的商业组件

说明：这里所说的“集成的商业组件”，是指与系统集成的商业组件，而不是与软件产品集成的商业组件。例如：中间件、消息服务，等等。

描述并且明确软件产品与软件组件之间交换数据或者消息的目的。描述所需要的服务，以及与内部组件通讯的性质。确定软件产品将与组件之间共享的数据。如果必须使用一种特殊的方法来实现数据共享机制，例如：在多用户系统中的一个全局数据区，那么就必须把它定义为一种实现上的限制。

## 通讯接口

描述与软件产品所使用的通讯功能相关的需求，包括：

* 电子邮件；
* WEB浏览器；
* 网络通讯标准或者协议；
* 数据交互用电子表格；

必须定义相关的：

* 消息格式；
* 通讯安全或加密问题；
* 数据传输速率；
* 同步和异步通讯机制；

# 系统用例设计

绘出系统的用例图，粗略说明各个用例的作用，粗略描述参与者对用例的使用以及交互。

建立用例模型的顺序是：

1. 确定谁会直接使用该系统。这些都是参与者（Actor）。
2. 选取其中一个参与者。
3. 定义该参与者希望系统做什么，参与者希望系统做的每件事成为一个用例。
4. 对每件事来说，何时参与者会使用系统，通常会发生什么，这就是用例的基本过程。
5. 描述该用例的基本过程。
6. 考虑一些可变情况，把他们创建为扩展用例。
7. 复审不同用例的描述，找出其中的相同点，抽出相同点作为共同的用例。
8. 重复步骤2~7找出每一个用例。

这些用例是系统功能的体现，在系统功能需求描述中将详细的描述这些用例。

# 系统功能需求

需要进行详细的需求记录，详细列出与该系统功能相关的详细功能需求，并且，唯一地标识每一项需求。这是必须提交给用户的软件功能，使得用户可以使用所提供的功能执行服务或者使用所指定的使用实例执行任务。描述软件产品如何响应己知的出错条件、非法输入、非法动作。

如果每一项功能需求都能用一项，也只需要用一项测试用例就能进行验证，那么就可以认为功能需求已经适当地进行描述了。如果某项功能需求找不到合适的测试用例，或者必须使用多项测试用例才能验证，那么该项功能需求的描述必然存在某些问题。

功能需求是根据系统功能，即软件产品所提供的主要服务来组织的。可以通过使用实例、运行模式、用户类、对象类或者功能等级来组织这部分内容，也可以便用这些元素的组合。总而言之，必须选择一种是读者容易理解预期产品的组织方案。

用简短的语句说明功能的名称，例如：“4.1系统参数管理”。按照服务组织的顺序，逐条阐述系统功能。无论说明的是何种功能，都应该针对该系统功能重复叙述4.1~ 4.3这三个部分。

可以通过各种方式来组织这一部分内容，例如采用：使用实例、运行模式、用户类、对象类、功能等级等，也可以采用它们的组合。其最终目的是，让读者容易理解即将开发的软件产品。一般来说，每个使用实例都对应一个系统功能，因而按照使用实例来组织内容比较容易让用户理解。

必须特别注意的是，在2.2节“产品的功能”中描述的全部需求，以及它们的规格说明；必须在某个系统功能描述中有所反映，而且不应重复。

## 输入／输出数据

列出输入数据（用户输入、来自外部接口的输入或者其它输入）并且定义针对这些输入数据的处理（计算）方法，以及相应地输出数据，描述对应区别：输入数据和输出数据。

当有大量数据需要描述时，也可以分类描述数据，并且注明各项数据的输入、输出属性。

对于每一项数据，均需要描述：

* 数据名称；
* 实际含义；
* 数据类型；
* 数据格式；
* 数据约束；

对于复杂的处理方法，仅仅给出算法原理是不够的，必须描述详细的计算过程，并且列出每一步具体使用的实际算式；如果计算过程中涉及查表、判断、迭代等处理方法，应该给出处理依据和相关数据。如果计算方法很简单，也可以将其从略，不加描述。

## 用例描述

用例还需要仔细完整的描述，以便后继工作能够以此为基础。用例描述需要从以下方面出发。

1）用例名，一个易记的正确的名字有利于顺畅交流。

2）简述：用简明扼要的话说明用例的意思。

3）主要参与者：用例涉及的主要参与者。

4）情景目标：用例要达到的目的，或是要完成的工作。

5）前提条件：用例能开展的前提条件。

6）触发事件：用例被执行的触发事件。

7）场景描述：这是用例描述的核心，即用例的内部流程。如果用自然语言描述不清楚，可以采用活动图或者顺序图来辅助描述。

8）异常处理：用例流程可能没有考虑各种异常情况，异常处理则描述其中的各种异常情况，以及发生异常时的处理流程。

9）优先级：用例优先实现的紧迫性，通常和用例的价值有关。指出该系统功能的优先级是：高、中、还是低。需要的话，还可以包括对特定优先级部分的评价，例如：利益、损失、费用和风险，其相对优先等级可以从1（低）到9（高）。

10）使用频率：用例被使用的频繁度。

11）次要参与者：非重要的参与者。

12）未明确的问题：用例中还有待厘清的问题。

用例描述的好坏非常重要，直接影响到软件的质量。而且，用例描述的修改也要严格控制，从而确保用例变更得到有效管理。

# 其它非功能需求

在这里列举出所有非功能需求，主要包括可靠性、安全性、可维护性、可扩展性、可测试性等。

## 性能需求

阐述不同应用领域对软件产品性能的需求，并且说明提出需求的原理或者依据，以帮助开发人员做出合理的设计选择。尽可能详细地描述性能需求，如果需要，可以针对每个功能需求或者特征分别陈述其性能需求。在这里确定：

* 相互合作的用户数量；
* 系统支持的并发操作数量；
* 响应时间；
* 与实时系统的时间关系：
* 容量需求
  + 存储器；
  + 磁盘空间；
  + 数据库中表的最大行数。

## 安全性需求

详尽陈述与系统安全性、完整性问题相关的需求，或者与个人隐私问题相关的需求。这些问题将会影响到软件产品的使用，和软件产品所创建或者使用的数据的保护。定义用户身份认证，或备授权需求。明确软件产品必须满足的安全性或者保密性策略。也可以通过称为完整性的质量属性来阐述这些需求。一个典型的软件系统安全需求范例如下：“每个用户在第一次登录后，必须更改他的系统预置登录密码，系统预置的登录密码不能重用。”

## 软件质量需求

详尽陈述对客户和开发人员至关重要的在软件产品其它方面表现出来的质量功能。这些功能必须是确定的、定量的、在需要时是可以验证的。至少也应该指明不同属性的相对侧重点，例如：易用性优于易学性，或者可移植性优于有效性。

## 业务规则

列举出有关软件产品的所有操作规则，例如：那些人在特定环境下可以进行何种操作。这些本身不是功能需求，但是他们可以暗示某些功能需求执行这些规则。一个业务规则的范例如下：“进行达到或者超过10，000，00元人民币的储蓄业务时，必须通过附加的管理员认证。”

列举业务规则时，可以根据规则的数量，选取合适的编目方式。

## 用户文档

列举出将与软件产品一同交付的用户文档，并且明确所有己知用户文档的交付格式或标准，例如：

* 安装指南

纸质文档，16开本；

* 用户手册

纸质文档，16开本；

* 在线帮助
* 电子文档，与软件产品一同分发、配置；
* 使用教程电子文档，与软件产品一同分发、配置。

# 数据定义

数据定义是一个定义了应用程序中使用的所有数据元素和结构的共享文档，其中对每个数据元素和结构都准确描述：含义、类型、数据大小、格式、计量单位、精度以及取值范围。数据定义的维护独立于软件需求规格说明，并且在软件产品开发和维护的任何阶段，均向风险承担者开放。

如果为软件开发项目创建一个独立的数据定义，而不是为每一项特性描述有关的数据项，有利于避免冗余和不一致性。但是却不利于多人协同编写需求分析报告，容易遗漏数据，也不方便阅读。因此还是建议为每个特性描述有关的数据项，汇总数据项创建数据定义，再根据数据定义复核全部数据，使得它们的名称和含义完全一致。必须注意的是，为了避免二义性，在汇总数据项时应该根据数据项所代表的实际意义汇总，而不是根据数据项的名称汇总。

在数据定义中，每个数据项除了有一个中文名称外，还应该为它取一个简短的英文名称，该英文名称应该符合命名规范，因为在软件开发时将沿用该英文名称。可以使用等号表示数据项，名称写在左边，定义写在右边。常见数据项的描述方式如下：

* 原数据元素

一个原数据元素是不可分解的，可以将一个数量值赋给它。定义原数据元素必须确定其含义、类型、数据大小、格式、计量单位、精度以及取值范围。采用以星号为界的一行注释文本，描述原数据元素的定义。

* 选择项

选择项是一种只可以取有限离散值的特殊原数据元素，描述时一一枚举这些值，并用方括号括起来写在原数据元素的定义前。在两项离散值之间，使用管道符分隔。

* 组合项

组合项是一个数据结构或者记录，其中包含了多个数据项。这些数据项可以是原数据元素，也可以是组合数据项，各数据项之间用加号连接。其中每个数据项都必须是数据定义中定义过的，结构中也可以包括其它结构，但是绝对不允许递归。如果数据结构中有可选项，使用圆括号把该项括起来。

* 重复项

重复项是组合项的一种特例，其中有一项将有多个实例出现在数据结构中，使用花括号把该项括起来。如果知道该项可能允许的范围，就按“最小值：最大值”的形式写在花括号前。

# 分析模型（可选）

分析模型通过多种方式展示系统的需求，可以按照以下的方式建模。（注意需求分析中的类图、顺序图是分析型的，和后面的系统设计中的类图、顺序图还是有区别，课上没有仔细讲，大家可以参考教材第八版中的第9、10两章）

## 活动图

活动图是一种用于描述系统行为的模型视图，它可用来描述动作和动作导致对象状态改变的结果，而不用考虑引发状态改变的事件。

活动图的作用主要体现在：

（1）描述一个操作执行过程中所完成的工作。说明角色、工作流、组织和对象是如何工作的。

（2）活动图对用例描述尤其有用。它可建模用例的工作流，显示用例内部和用例之间的路径。它可以说明用例的实例是如何执行动作以及如何改变对象状态。

（3）显示如何执行一组相关的动作，以及这些动作如何影响它们周围的对象。

（4）活动图对理解业务处理过程十分有用。活动图可以画出工作流用以描述业务，有利于与领域专家进行交流。通过活动图可以明确业务处理操作是如何进行的，以及可能产生的变化。

（5）描述复杂过程的算法。在这种情况下使用的活动图和传统的程序流程图的功能是差不多的。



## 类图

在需求分析阶段，寻找系统中参与活动的实体，并描述这些实体是重要的工作之一。同时，这些实体的交互也体现了系统功能的实现过程。在面向对象编程框架下，实体（也可能是抽象概念），就是对象。具有相同特征的对象可以抽象为类。类与类间的关系也就描述了对象之间的关系。类图就是要描绘这种关系。

## 顺序图

如果需要展示多个实体之间交互的过程，尤其是体现交互的时间顺序，则可以使用顺序图。顺序图是强调消息时间顺序的交互图。顺序图描述了对象之间传送消息的时间顺序，用来表示用例中的行为顺序。顺序图将交互关系表示为一个二维图。即在图形上，顺序图是一张表，其中显示的对象沿横轴排列，从左到右分布在图的顶部；而消息则沿纵轴按时间顺序排序。创建顺序图时，以能够使图尽量简洁为依据布局。

可以用顺序图来描述复杂用例的处理流程，体现用例实现过程中各个参与者的交互关系。

# 词汇表（可选）

列出本文件中用到的专业术语的定义，以及有关缩写的定义（如有可能，列出相关的外文原词）。为了便于非软件专业或者非计算机专业人士阅读软件产品需求分析报告，要求使用非软件专业或者非计算机专业的术语描述软件需求。所以这里所指的专业术语，是指业务层面上的专业术语，而不是软件专业或者计算机专业的术语。但是，对于无法回避的软件专业或者计算机专业术语，也应该列入词汇表并且加以准确定义。

# 待定问题列表（可选）

编辑一张在软件产品需求分析报告中待确定问题时的列表，把每一个表项都编上号，以便跟踪调查。

# 附录B 软件概要设计报告文档模板

# 设计概述（概述部分不超过1500字）

本节描述现有开发条件和需要实现的目标，说明进行概要设计时应该遵循的设计原则和必须采用的设计方法。

## 设计原则和设计要求

描述对本软件系统进行概要设计的原则，通常可以考虑以下几方面的内容：

* 命名规则；
* 模块独立性原则：
* 边界设计原则；
* 数据库设计规则；
* 必须的安全措施；
* 安全性和保密原则；
* 系统灵活性要求；
* 系统易操作性要求；
* 系统可维护性要求；
* 等等。

## 限制和约束

简要描述起到限制和约束作用的各种可能存在的条件，例如：

* 技术条件；
* 资金状况；
* 开发环境（包括：工具和平台）；
* 时间限制；
* 等等。

并且说明在上述条件下，应该实现的系统目标，

## 项目风险

具体说明本软件开发项目的全部风险承担者，以及各自在本阶段所需要承担的主要风险，首要风险承担者包括：

* 任务提出者；
* 软件开发者；
* 产品使用者。

## 支撑环境

### 开发工具、中间件以及数据库接口（若有）

描述所选用的工具软件和中间件的名称、版本号，以及开发工具与数据库或者中间件接口的情况。如果使用了多种开发工具、辅助开发工具、第三方软件部件、多种中间件、多种接口、等答应该逐项分别描述，并且说明每一项的适用范围。需要描述的内容可能包括：

* 产品名称以及发行厂商

同2.1中产品名称以及发行厂商。

* 版本号

同2.1中版本号。

* 补丁包版本号

同2.1中补丁包版本号。

* 语言或代码集

同2.1中语言或代码集。

* 数据库接口名称

描述数据库接口的名称，如果使用别名时，应同时描述使用的别名。

* 数据库接口方式

描述与数据库接口的方式，并说明该接口方式的特点；如果需要，还应该说明使用时的注意事项。

* 数据库接口设置

描述各种接口设置，包括：协议、端口号等等。

### 硬件环境

描述所选用的硬件环境，各种机型，例如：服务器、工作站，应该分别描述。需要描述的内容可能包括：

* 机型；
* 主频；
* 内存容量；
* 磁盘容量；
* 特殊部件；
* 操作系统；
* 使用位置；
* 等等。

### 网络环境

描述可能影响应用软件访问数据库的各种网络环境，如果存在加密传输、VPN链路等情况，也必须描述。对于结构复杂的网络，还应该提供网络拓扑图和数据流向示意图。需要描述的内容可能包括：

* 网络结构；
* 网络操作系统；
* 网络带宽；
* 路由组织；
* 加密传输方式；
* VPN链路连接方式；
* 等等。

### 多种支撑环境开发要点

当软件产品将来可能遇到的多种运行环境时，应该分别按照3.1节至3.4节的内容列表描述。如果软件产品各个子系统的运行环境不完全一样时，应该分子系统按照3.1节至3.4节的内容列表描述。

遇到上述情况时，不仅需要详细描述各种软件开发、调试、测试的环境，为了确实保证软件产品将来能够在各种可能的运行环境中正常运行，还需要对软件产品进行严格的配置管理。

# 系统逻辑设计

（本节的表格只是一种示例，不要硬套，可以根据项目实际情况另外设计表格）

本节内容主要根据软件产品需求规格说明书和软件产品数据字典建立系统的逻辑模型。此种模型暂时与系统的物理因素（例如：计算机、数据库管理系统）无关。它是系统需求与物理实现的中间结构，它的主要结果可以包括：系统结构图、系统界面结构图、系统开发技术说明。

说明：如果进行系统设计时尚未编写软件数据字典：应首先编写软件数据字典。在完成软件数据字典后，再进行系统设计。

## 系统组织设计

系统组织设计通过系统组织表描述本系统由哪些子系统（模块）组成，这些子系统与业务职能之间的关系，以及各个子系统的安装地点。系统组织表的格式如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 子系统编号 | 英文名称 | 中文名称 | 业务职能 | 安装地点 | 备注 |
|  |  |  |  |  |  |

其中：

* 子系统编号

给出本系统中指定子系统的顺序编号。如果本系统末划分为多个子系统，仅由一个运行模块组成；则本项内容仍需要描述，但是本表内容只有一行。

说明：在一个系统中有可能安装若干个相同的子系统，在这种情况下，应该视为一个子系统，并且对多个安装地点分别进行描述。如果相同的子系统通过系统设置，实现的业务职能具有明显差异时，应该采用多行进行分别描述，并且在备注中说明其差异所在。

* 子系统英文名称

给出本子系统的英文名称，该名称是在应用软件中实际使用的可执行文件名称，必须能够说明该子系统的特点。若本系统中只有一个子系统，则本项内容仍需要描述，但是本表内容只有一行。

* 子系统中文名称

给出本子系统的中文名称，该名称必须能够说明该子系统的特点。若本系统中只有一个子系统，则本项内容仍需要描述，但是本表内容只有一行。

* 业务职能

描述该子系统完成的核心业务。

* 安装地点

描述该子系统实际安装的部门、或者某个具体地点。

* 备注

针对该子系统，需要说明的其它有关问题。

## 系统结构（或功能）设计

本节将对系统特性作较为详细的描述，并给出系统特性结构图。

### 系统用例模块映射表

（这种表属于结构化的描述方式，酌情选用）

系统用例是系统中完成某项具体操作的基本单元，它由入口参数，出口参数以及处理过程等组成，是系统要实现的对象。

系统用例可以具有操作界面，也可以没有操作界面；可以被其它操作界面、或者系统用例调用，也可以调用其它操作界面、非操作界面、或者系统用例；但是不允许递归调用（调用自己），包括间接递归调用。

系统用例模块映射表说明各个系统用例和各个子模块的关系，即每个子模块实现那些用例或者用例的一部分。

当系统由多个子系统（模块）组成时，每个子系统分别使用一张系统用例模块映射表进行描述。系统用例模块映射表的格式如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 子系统编号：  子系统英文名称：  子系统中文名称： | | | | | | |
| 用例编号 | 系统用例  英文名称 | 系统用例  中文名称 | 操作功能 | 调用对象 | 被调用  对象 | 备注 |
|  |  |  |  |  |  |  |
| 说明： | | | | | | |

其中

* 子系统编号

含义同上。

* 子系统英文名称

含义同上。

* 子系统中文名称

含义同上。

* 用例编号

整个系统所有用例的统一编号。

* 系统用例英文名称

系统用例的英文正式名称，将来用于软件开发中，必须符合命名规范。

* 系统用例中文名称

系统用例的中文正式名称，来源于需求规格说明书中，系统用例一节中的有关描述。

* 操作功能

是指该用例实际完成的操作说明。

* 调用对象

是指调用该系统用例的系统对象，这里的系统对象可以是系统用例、也可以是操作界面。

* 被调用对象

是指被该系统用例调用的系统对象，这里的系统对象可以是系统用例、也可以是操作界面。说明：某些较低层的系统用例，可能不存在被调用对象。

* 备注

描述与该系统用例有关的其它注意事项。

* 说明

描述与该系统用例模块映射表有关的其它注意事项。

### 系统体系结构图或功能模块图

系统体系结构图给出系统各模块在逻辑层面上相互之间的关系。如果系统划分为多个子系统，应分别给出系统与子系统、以及各个子系统与系统功能的结构图。

## 系统接口设计

（设计中的接口含义广泛，既可以指硬件接口，也可以指软件或代码的接口。具体情况具体分析）

系统接口是一种非可视的系统界面，在多数情况下，它对用户是透明的。

本节将对系统接口作较为详细的描述，并给出接口说明清单。

### 系统接口表

接口作为系统的一种输入／输出形式，分为网络接口、数据库接口、RS-232串行通讯接口、IEEE—485串行总线接口、并行I/O接口等等多种类型。

对于一些为可视界面服务的接口，例如：打印机接口、显示器接口等，因为这类接口对应用软件是透明的，所以不在本节描述范围内。

当系统由多个子系统（模块）组成时，每个子系统分别使用一张系统接口表进行描述。系统接口表的格式如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 子系统编号  子系统英文名称  子系统中文名称 | | | | | | |
| 接口  编号 | 接口  名称 | 接口  类型 | 接口  性质 | 接口  速率 | 接口  协议 | 备注 |
|  |  |  |  |  |  |  |
| 说明： | | | | | | |

其中：

* 子系统编号

含义同上。

* 子系统英文名称

含义同上。

* 子系统中文名称

含义同上。

* 接口编号

整个系统所有接口的统一编号。

* 接口名称

系统接口的正式名称，必须符合通常习惯。

* 接口类型

指出该接口所传输的数据在该模块中起到的作用。

* 接口性质

指出该接口在通讯中起到的作用，这里的作用可以是：

* 输入；
* 输出；
* 双向。
* 接口速率

指出该接口的传输速率。如果该接口依赖于其它通讯方式，那么传输速率将不高于它所依赖的其它通讯方式的速率。

* 接口协议

给出该接口实际使用的通讯协议。

* 相关对象

给出直接使用本接口的系统对象，这里的系统对象，可以是操作界面，也可以是系统特性。

* 备注

描述与该系统接口有关的其它注意事项。

* 说明

描述与该系统接口表有关的其它注意事项。

### 系统接口传输协议说明

逐项详细描述系统接口表中所列出各个系统接口使用的传输协议，以及其它相关内容，例如：驱动程序、动态连接库等等。

# 技术设计

（本节的表格只是一种示例，不要硬套，可以根据项目实际情况另外设计表格）

系统技术设计描述系统各个特性实际使用的开发技术，以及具体开发技术使用时应该注意的事项。

## 系统开发技术说明表

本表描述系统各个特性开发时实际使用的具体技术，只有一些不太常用的技术需要在这里描述。一些常用技术，例如：通过数据库接口调用存储过程，则不必冗述。

当系统由多个子系统（模块）组成时，每个子系统分别使用一张系统开发技术说明表进行描述。系统开发技术说明表的格式如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 子系统编号：  子系统英文名称：  子系统中文名称： | | | | | | |
| 技术编号 | 开发技术 | 开发技术 | 处理功能 | 系统特性编号 | | 备注 |
| 英文名称 | 中文名称 |
|  |  |  |  |  |  |  |
| 说明： | | | | | | |

其中：

* 子系统编号

含义同上。

* 子系统英文名称

含义同上。

* 子系统中文名称

含义同上。

* 技术编号

这个系统所使用各种技术的统一编号。

* 开发技术英文名称

该开发技术的英文正式名称，可以便用缩写。该名称应该是常用的，并且为人们所普遍接受的。

* 开发技术中文名称

该开发技术的中文正式名称，是该开发技术英文名称的中文说明。该名称应该是常用的，并且为人们所普遍接受的。

* 处理功能

描述本开发技术的处理目的。

* 系统特性编号

含义同上。由于一项开发技术可能在多处使用，因此针对一项开发技术，有可能存在多个系统特性编号，在此必须一一列出。

* 备注

描述与该系统开发技术相关的其它注意事项。

* 说明

描述与该系统开发技术说明表有关的其它注意事项。

## 开发技术应用说明

逐项详细描述系统开发技术说明表中所列出各项系统开发技术使用的技术要点，以及其它相关内容，例如：所需的服务、使用的动态连接库、调用的组件、等等。

# 数据库或数据文件设计

（如果下面的附录D数据库设计没有涉及的话，可在这一节作为一个补充，作为对数据建模的描述）

如果该软件产品需要使用数据库，不论是使用数据库平台支撑的，还是采用由软件产品开发者自行定义的；都应该在完成软件产品需求分析报告后，开始进行软件产品详细设计之前，按照软件产品数据库设计说明文档模板完成数据库设计工作。

# 进度计划

列出进度计划，包括各子系统、各子模块完成进度计划，人员配备计划等。

# 附录C 软件详细设计报告文档模板

# 引言（引言不超过500字）

（提示：请用户根据项目的实际设计状况，裁剪本详细设计报告模板）

引言是对这份软件系统详细设计报告的概览，是为了帮助阅读者了解这份文档如何编写的，并且应该如何阅读、理解和解释这份文档。

## 文档约定

描述编写文档时所采用的标准（如果有标准的话），或者各种编写约定。编写约定应该包括：

* 部件编号方式；
* 界面编号方式；
* 命名规范：
* 等等。

# 部件（或类、方法）详细设计

（这里的“部件”指能够完成特定功能、相对独立的代码或功能组件，可以指代任何具有“模块”特点的东西，包括但不限于：类、接口、方法/函数、多个类组成的子系统、界面、界面中的控件、动态连接库、库文件等。具体呈何种形态，取决于实际采用的开发语言、工具和软件的体系结构。

用部件表对设计进行描述，属于结构化的描述风格，而不是面向对象的。本模板有些内容较老，大家酌情采用，也可以不用，不要迷信模板。本章关于“部件”描述的表格其实设计得并不好，对于面向对象的设计表达起来比较别扭。同学们完全可以自己设计一种表）

按照合适的顺序，逐个描述软件部件的详细情况。描述的顺序可以是按层次横向进行描述，也可以是按模块纵向进行描述，总之描述的方式必须有利于读者理解软件结构。

部件（或类、方法）表格式：

|  |  |  |  |
| --- | --- | --- | --- |
| 部件编号 |  | 部件名称 |  |
| 所属子系统 |  | | |
| 部件调用者 |  | | |
| 部件被调用者 |  | | |
| 部件入口参数 |  | | |
| 部件出口参数 |  | | |
| 算法或流程描述（也可放到后面再描述） | | | |
| 表示性能 |  | 运行环境 |  |
| 性能要求 |  |  |  |

说明：如果软件不见使用一张表表述不完时，可以采用续表描述，但是必须注明是那张表的续表。每个部件采用一张软件部件表进行描述，软件部件表的格式见上面附表，其中；

每个部件采用一张软件部件表进行描述，软件部件表的格式见附表一，其中；

* 部件编号

软件部件的统一顺序编号；对于实行配置管理的软件开发项目来说，该编号必须与该部件在配置管理中的编号相同。

* 部件名称

软件部件的正式英文名称，该名称是程序中使用的实际名称，必须符合国家相关软件命名标准。

* 所属子系统

指该部件所属的子系统；对于不分为多个子系统的软件来说，不必填写该栏。

* 部件调用者

指调用该部件的部件（或界面参数）的编号和名称。

* 部件被调用者

指被该部件所调用的部件的编号和名称。

* 部件入口参数

指该部件入口数据类名称或者数据名称，以及对这些数据的描述；如果部件没有入口参数，该栏为空。

* 部件出口参数

指该部件出口数据类名称或者数据名称，以及对这些数据的描述；如果部件没有出口参数，该栏为空。

* 算法

指该部件的算法形式表示，如果很简单、或者不存在，也可以为空。

* 流程描述

指该部件的处理流程的详细表示或描述。

* 部件表示形式

指该部件完成开发后的最终表示形式，具体形式取决于开发工具和软件结构，表示形式可能是：

* 插件、组件、控件，
* 函数、过程、子程序，
* 存储过程，
* 动态连接库，
* 等等。
* 运行环境

描述该部件所适合的运行环境，即说明该部件是针对何种运行环境所开发的；

可以直接描述运行环境，也可以描述运行环境的编号；对于实行配置管理的软件开发项目来说，该描述必须与该部件在配置管理中的描相同。

* 性能要求

指开发该部件时必须满足的专门要求，这些要求可以是：

* 精度
* 灵活性
* 响应时间
* 可重用性
* 等等。

提出的要求一般不宜超过3项，以排列的先后顺序表示优先级。

## 类数据成员说明

2.1.1 成员变量列表（示例）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 变量 | 类型 | 含义 | 修饰符 |
| 1 | id | String | 用户的账号id | 私有 |
| 2 | pwd | String | 用户的密码password | 私有 |
| 3 | name | String | 用户名name | 公有 |

## 类方法成员的签名、算法和使用说明

这部分既可以放到前面的部件表中，也可以单列。建议单列。前面的部件表设计不是很合理，它适合表现较大的子系统或模块。

2.2.1 成员方法/函数列表（示例）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 方法名 | 参数列表 | 返回类型 | 用途 | 修饰符 |
| 1 | User | 无 |  | 构造函数 | 公有 |
| 2 | User | String id, String pwd, String name |  | 构造函数，进行赋值 | 公有 |
| 3 | getId | 无 | String | 获取用户账号 | 公有 |
| 4 | setId | String id | void | 设置用户账号 | 公有 |
| 5 | getPwd | 无 | String | 获取用户密码 | 公有 |
| 6 | setPwd | String pwd | void | 设置用户账号 | 公有 |

2.2.2 成员方法实现细节

成员方法不仅要列出来，还应对其中重要方法的算法或流程的描述（过于简单的不用，比如getId( ) ）。不推荐用伪代码来表现，因为很容易用ChatGPT生成。当然大家可以用ChatGPT生成伪代码或解释辅助理解是可以的，但最后在文档中宜用流程图、顺序图、判定表等图表的形式来表现。

# 界面详细设计

界面表格式：

|  |  |  |  |
| --- | --- | --- | --- |
| 界面编号 |  | 部件名称 |  |
| 界面性质 |  | 界面介质 |  |
| 表示形式： | | | |
| 界面参数 | | | |
| 参数名 | 内容 | | 说明 |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |

说明：如果软件不见使用一张表表述不完时，可以采用续表描述，但是必须注明是那张表的续表。

# 附录D 软件数据库设计报告文档模板

（可选，若本节没有应删除）

# 引言（引言不超过1000字）

（提示：请用户根据项目的实际状况，裁剪本数据库设计报告模板）

引言是对这份数据库设计说明书的概览，是为了帮助阅读者了解这份文档是如何编写的，并且应该如何阅读、理解和解释这份文档。

## 文档约定

描述编写文档时所采用的各种排版约定。排版约定应该包括：

* 命名方法；
* 提示方式；
* 通配符号：
* 等等。

## 数据库命名规则

完整并且清楚的说明本数据库的命名规则，在《南京市交通局信息化数据库建设规范》中已经给出了一个完整的数据库命名规则，开发者应遵守执行，如果本数据库的命名规则与该规范不完全一致，应作出解释。

## 数据库管理系统概述

描述数据库管理系统、以及安装配置情况，需要描述的内容可能包括：

* 产品名称以及发行厂商

这里的产品名称指的是数据库发行厂商发布产品时公布的正式商品名称，不应该使用别名、简称、研发代号等非正式名称，以免混淆；同样的道理，发行厂商的名称也应该使用正式名称。

* 版本号

数据库管理系统的准确版本号，必须按产品的实际情况描述到最细节的版本号。

* 补丁包版本号

描述实际上将要使用的数据库管理系统补丁包的版本号，必须注意，在某些情况下该版本号不一定是最新的版本号。

* 语言或代码集

对于只支持一种语言或者一个代码集的数据库管理系统来说，该项描述不具意义。对于支持多种语言或者多个代码集的数据库管理系统来说，该项描述指的是实际使用的语言或者代码集。

* 安装位置

描述数据库管理系统的实际安装位置，应该分别对管理系统安缺位置和数据存放位置进行描述，应该指明服务器名和安装卷号（盘号）。对于分布式数据库，必须分别描述每一个数据库管理系统。

* 配置参数

描述数据库管理系统在实际安装时应该配置的各个参数，对于分布式数据库，必须分别描述每一个数据库管理系统的配置参数。

* 等等。

# 数据库设计说明

## 数据库逻辑设计

数据库设计人员根据《软件需求分析报告》，创建与数据库相关的实体关系图（E-R图）。如采用面对对象的分析和设计方法，则此处的实体相当于类。

在此处，应给出逻辑设计的完整的E-R图。

## 数据库物理设计

在此处应给出完整的数据库物理结构E-R图。开发者应根据逻辑设计的结果，进行数据库的物理设计，并对表结构进行规范化处理（第一范式，第二范式，第三范式）。

## 数据库分布

数据库分布采用一张表格进行描述，其格式如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据库编号 | 数据库管理系统名称 | 数据库管理系统版本号 | 数据库英文名称 | 数据库中文名称 | 数据库安装物理位置 |
|  |  |  |  |  |  |

其中：

* 数据库编号

给出本系统中指定数据库的顺序编号。

若本系统中只有一个数据库，则本项内容不需要描述，本表内容也只有一行。

说明： 在一个系统中可能安装若干个相同的或者不同的数据库管理系统，

一个数据库管理系统也可能安装一个或者多个数据库。

* 数据库管理系统名称

给出本系统中指定数据库管理系统的商品名称。

若本系统中只有一种数据库管理系统，则本项内容不需要描述。

* 数据库管理系统版本号

给出本系统中指定数据库管理系统的版本号。

若本系统中只有一个版本的数据库管理系统，则本项内容不需要描述。

* 数据库英文名称

给出本数据库的英文名称，该名称是在应用软件中实际使用的名称，必须符合《南京市交通局信息化数据库建设规范》中相关命名规范。

* 数据库中文名称

给出本数据库的中文名称，该名称是本数据库英文名称的说明。

* 数据库安装物理位置

给出本数据库安装的实际位置，必须描述清楚该位置是在那个物理设备的哪一

个逻辑存储设备上，以及存储文件的名称。

## 基表设计

每个基表采用一张表格进行描述，其格式如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据库编号： | | | | |
| 基表编号： | | | | |
| 基表英文名称： | | | | |
| 基表中文名称： | | | | |
| 字段编号 | 英文字段名 | 中文字段名 | 字段类型 | 备注 |
|  |  |  |  |  |
| 说明： | | | | |

其中

* 数据库编号

含义同上。

* 基表编号

给出本基表的顺序编号。

* 基表英文名称

给出本基表的英文名称，该名称是在应用软件中实际使用的名称，必须符合命

名规范。

* 基表中文名称

给出本基表的中文名称，该名称是本基表英文名称的说明。

* 字段编号

该基表中，各个字段的顺序编号。

* 英文字段名

该基表中，各个字段的英文名称，该名称必须符合《南京市交通局信息化数据库建设规范》中相关命名规范。

* 中文字段名

该基表中，各个字段的中文名称，该名称是英文字段名的说明。

* 字段类型

该基表中，各个字段的类型；如果需要，在说明类型时，还需要说明字段长度。

* 备注

该基表中，各个字段有关的限制性说明，需要描述的内容可能包括：

* 值域；
* 缺省值；
* 空字段限制；
* 显示格式与小数位数；
* 有效性规则与约束；
* 标题；
* 等等
* 说明

说明一些有关本表的、必须描述清楚的问题，需要描述的内容可能包括：

* 主关键字；
* 索引、排序方式和类型；
* 触发器；
* 数据复制；
* 等等

## 视图设计

每个视图采用一张表格进行描述，其格式如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据库编号： | | | | | |
| 视图编号： | | | | | |
| 视图英文名称： | | | | | |
| 视图中文名称： | | | | | |
| 相关基表和视图： | | | | | |
| 字段编号 | 英文字段名 | 中文字段名 | 字段类型 | 字段源 | 备注 |
|  |  |  |  |  |  |
| 说明： | | | | | |

其中：

* 数据库编号

含义同上。

* 视图编号

给出本视图的顺序编号。

* 视图英文名称

给出本视图的英文名称，该名称是在应用软件中实际使用的名称，必须符合

命名规范。

* 视图中文名称

给出本视图的中文名称，该名称是本视图英文名称的说明。

* 相关基表和视图

列出建立该视图时，所用到的基表和视图。

* 字段编号

该视图中，各个字段的顺序编号。

* 英文字段名

该视图中，各个字段的英文名称，该名称必须符合《南京市交通局信息化数据库建设规范》中相关命名规范。

* 中文字段名

该视图中，各个字段的中文名称，该名称是英文字段名的说明。

* 字段类型

该视图中，各个字段的类型；如果需要，在说明类型时，还需要说明字段长度。

* 字段源

该视图中，各个字段的来源，即该字段原来是那个表或者那个视图中的那个字

段；在某些情况下，字段可能来自一个特定的表达式。

* 备注

该视图中，各个字段有关的限制性说明，包括：

* 值域；
* 缺省值；
* 空字段限制；
* 显示格式与小数位数；
* 有效性规则与约束；
* 标题；
* 等等。
* 说明

说明一些有关本视图的、必须描述清楚的问题，需要描述的内容可能包括：

* 索引；
* 权限；
* 等等

## 索引设计

每个数据库的所有采用一张表格进行描述，其格式如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据库编号： | | | | |
| 索引编号 | 基表名称 | 索引名称 | 字段集名称 | 备注 |
|  |  |  |  |  |

其中：

* 数据库编号

含义同上。

* 索引编号

给出本项索引的顺序编号。

* 基表名称

给出本项索引所在的基表名称。

* 索引名称

给出本项索引的名称。

* 字段集名称

给出本项索引所在的字段名称或者字段集名称。

* 备注

描述有关本项索引中，其它需要说明的事项，例如：排序方式、等等。

## 完整性约束

每个数据库的完整性约束采用一张表格进行描述，其格式如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据库编号： | | | | |
| 索引编号 | 基表名称 | 索引名称 | 字段集名称 | 备注 |
|  |  |  |  |  |

其中：

* 数据库编号

含义同上。

* 约束编号

给出本项完整性约束的顺序编号。

* 完整性约束名

给出本项完整性约束的名称。

* 基表名

给出本项完整性约束所在的基表名称。

* 字段名

给出本项完整性约束所在的字段名称。

* 约束表达式

给出本项完整性约束的逻辑表达式。

* 备注

描述有关本项完整性约束中，其它需要说明的事项。

## 授权设计

每个数据库的授权采用一张表格进行描述，其格式如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据库编号： | | | | |
| 授权编号 | 用户名称 | 对象名称 | 权限 | 备注 |
|  |  |  |  |  |

其中：

* 数据库编号

含义同上。

* 授权编号

给出本项授权的顺序编号。

* 用户名称

给出本项授权的用户名称，这里的用户不一定是具体用户，也可以是用户组。

* 对象名称

给出本项授权的对象名称，例如：基表、字段、等等。

必须注意到，一个用户可能存在多项授权，应该逐项描述。

* 权限

被授权用户在该对象上拥有的访问权限，例如：查询权、修改权、等等。

* 备注

描述有关本项授权中，其它需要说明的事项。

## 触发器设计

* 数据库编号

含义同上。

* 触发器编号

给出本触发器的顺序编号。

* 触发器英文名称

给出本触发器的英文名称，必须符合《南京市交通局信息化数据库建设规范》中相关命名规范。

* 触发器中文名称

给出本触发器的中文名称，该名称是本触发器英文名称的说明。

* 触发器条件

给出该触发器产生触发的条件。

* 触发器结果

给出该触发器被触发后所执行的动作内容。

## 存储过程设计

每个数据库的授权采用一张表格进行描述，其格式如下：

|  |
| --- |
| 数据库编号： |
| 存储过程编号： |
| 存储过程英文名称： |
| 存储过程中文名称： |
| 存储过程内容： |
| 说明： |

其中：

* 数据库编号

含义同上。

* 存储过程编号

给出本存储过程的顺序编号。

* 存储过程英文名称

给出本存储过程的英文名称，该名称是在应用软件中实际使用的名称，必须符

合命名规范。

* 存储过程中文名称

给出本存储过程的中文名称，该名称是本存储过程英文名称的说明。

* 存储过程内容

给出该存储过程算法或者描述详细内容，如果需要，应该辅以流程图说明。

* 说明

描述本存储过程需要说明的一些事项。

## 数据复制设计

每项数据复制采用一张表格进行描述，其格式如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 数据复制编号： | | | |
| 复制英文名称： | | | |
| 复制中文名称： | | | |
| 源数据库编号： | | | |
| 目标数据库编号： | | | |
| 复制说明： | | | |
| 执行方式： | | | |
| 源数据库名称 | | 目标数据库名称 | |
| 基表名称 | 字段名称 | 基表名称 | 字段名称 |
|  |  |  |  |
| 备注： | | | |

其中：

* 数据复制编号

给出本数据复制的顺序编哥

* 数据复制英文名称

给出本数据复制的英文名称，该名称是在应用软件中实际使用的名称，必须符

合命名规范。

* 数据复制中文名称

给出本数据复制的中文名称，该名称是本数据复制英文名称的说明。

* 源数据库编号

作为复制数据源的数据库编号，编号含义同上。

* 目标数据库编号

作为复制目标的数据库编号，编号含义同上。

* 复制说明

给出该复制的详细描述，如果需要，应该辅以示意图说明。

* 执行方式

给出该复制的执行方式，描述时应该说明：

* 自动执行

必须说明执行周期或者执行条件。

* 调用执行

必须说明被那个模块调用，以及是手动调用，还是条件调用。

* 源数据库名称

给出对应源数据库编号的源数据库名称。

* 目标数据库名称

给出对应目标数据库编号的目标数据库名称。

* 基表名称

分别给出源数据库和目标数据库中，进行对应复制的源基表名称和目标基表名

事例。

* 字段名称

分别给出源基表和目标基表中，进行对应复制的源字段名称和目标字段名称。

* 备注

描述本复制中需要说明的一些特殊事项。

# 附录E 软件测试报告文档模板

# 1. 引言（引言不超过500字）

（提示：请用户根据项目的实际测试状况，裁剪本测试报告模板）

**2. 单元测试**

**2.1 被测试对象（单元）的介绍**

**2.2 测试范围与目的**

**2.3 测试环境与测试辅助工具的描述**

**2.4 基本路径覆盖法设计单元测试用例**

## 2.5 测试记录和分析

**3. 集成测试**

**3.1 被测试对象的介绍**

**3.2 测试范围与目的**

**3.3 测试环境与测试辅助工具的描述**

**3.4 集成策略与驱动/桩测试模块的设计**

## 3.5 测试记录和分析

**4. 验收测试**

**4.1 被测试对象的介绍**

**4.2 测试范围与目的**

**4.3 测试环境与测试辅助工具的描述**

**4.4 黑盒测试策略设计验收测试用例**

### 4.4.1 等价类划分

### 4.4.2 边界值分析

## 4.5 测试记录和分析