<项目名称>

软件需求规约

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **修订说明** | **作者** |
| <yyyy-mm-dd> | <x.x> | <详细信息> | <姓名> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[1引言 1](#_Toc446674967)

[1.1编写目的 1](#_Toc446674968)

[1.2适用范围 1](#_Toc446674969)

[1.3定义 1](#_Toc446674970)

[1.4参考资料 1](#_Toc446674971)

[1.5概述 1](#_Toc446674972)

[2目前系统 1](#_Toc446674973)

[3建议的系统 2](#_Toc446674974)

[3.1概述 2](#_Toc446674975)

[3.2功能需求 2](#_Toc446674976)

[3.3非功能需求 2](#_Toc446674978)

[3.3.1可用性 2](#_Toc446674979)

[3.3.2可靠性 2](#_Toc446674980)

[3.3.3性能 3](#_Toc446674981)

[3.3.4可支持性 3](#_Toc446674982)

[3.3.5设计约束 3](#_Toc446674983)

[3.3.6 接口 4](#_Toc446674984)

[3.3.7 法律、版权及其他声明 4](#_Toc446674985)

[3.3.8 适用的标准 4](#_Toc446674986)

[3.4 系统模型 5](#_Toc446674987)

[3.4.1场景 5](#_Toc446674988)

[3.4.2用例模型 5](#_Toc446674989)

[3.4.3用户界面 5](#_Toc446674990)

软件需求规约

# 1引言

## 1.1编写目的

说明：编写这份软件需求规约的目的，并指出预期的读者。

## 1.2适用范围

列出此文档适用的软件应用程序、特性或其他子系统分组、与其相关的用例模型，以及受到此文档影响的任何其他产品。

## 1.3定义

列出本文件中用到的专门术语的定义和外文的首字母缩写词。可以引用项目词汇表来提供。

## 1.4参考资料

列出本文中各处引用的文档资料，包括每个文档的标题、文档编号、发表日期和出版单位并列出能够得到这些文件资料的来源。

## 1.5概述

说明本文件中其他各部分包含的内容，与本文件的内容组织方式。

# 2目前系统

描述目前的状态。如果目前软件项目是开发一个新的系统替代以前的系统，那么此章节可以用来描述老系统的功能和存在的问题。如果目前并没有老系统，那么该章节就描述在实际业务中遇到的一些问题。

# 3建议的系统

## 3.1概述

简要介绍本项目包含的所有软件需求。

## 3.2功能需求

说明：列出本项目的功能性需求。对于应用程序来说，此节应重点描述。在表述时，可以一一列举各条功能需求，如<功能需求一>

建议仔细考虑此节的组织方式，如按照软件特性、用户集、子系统等组织方式进行说明。通常需要说明特性、性能和安全性。

当利用应用程序开发工具（如需求工具、建模工具等）来获取功能性时，此节文档将引用获取相应数据的方法，并指出用来获取数据的工具的位置和名称。

### 3.2.1 <功能需求一>

对该条功能需求进行详细说明。

## 3.3非功能需求

### 3.3.1可用性

说明：列出所有可能影响本项目可用性的需求，例如：

* + 普通用户和高级用户熟练操作所需培训时间。
  + 可根据用现有系统来确定新系统的可用性需求。
  + 符合公认的可用性标准（如IBM的CUA标准和Microsoft的GUI标准）方面的需求。

在表述时，可以一一列举各条可用性需求，如<可用性需求一>，以下各节非功能性需求均可如此表述。

#### 3.3.1.1 <可用性需求一>

对该条可用性需求进行详细说明。

### 3.3.2可靠性

列出所有关于本项目系统可靠性的需求，例如：

* + 系统开放时间与操作权限。
  + 平均故障间隔时间、平均修复时间与精确度需求。
  + 最高代码错误率，通常表示为每千行代码的错误数目 (bugs/KLOC) 或每个功能点的错误数目 (bugs/function-point
  + 错误率，按照小错误、大错误和严重错误来分类。需求中必须对“严重”错误进行界定，例如：数据完全丢失或完全不能使用系统的某部分功能。

#### 3.3.2.1 <可靠性需求一>

对该条可靠性需求进行详细说明。

### 3.3.3性能

此节应概述系统的性能特征。其中需包括具体的响应时间。如果可行，按名称引用相关用例。例如：

* + 对事务的响应时间（平均、最长）
  + 吞吐量，例如每秒处理的事务数
  + 容量，例如系统可以容纳的客户或事务数
  + 降级模式（当系统以某种形式降级时可接受的运行模式）
  + 资源利用情况，如内存、磁盘、通信等

#### 3.3.3.1 <性能需求一>

对该条性能需求进行详细说明。

### 3.3.4可支持性

此节应列出将提高所构建系统的可支持性或可维护性的所有需求，其中包括编码标准、命名约定、类库、维护访问权和维护实用程序。

#### 3.3.4.1 <可支持性需求一>

对该条可支持性需求进行详细说明。

### 3.3.5设计约束

此节应列出所构建系统的所有设计约束。设计约束代表已经批准并必须遵循的设计决定。其中包括软件语言、软件流程需求、开发工具的指定用途、构架及设计约束、购买的构件、类库等

#### 3.3.5.1 <设计约束一>

对该条设计约束进行说明。

### 3.3.6 接口

此节规定应用程序必须支持的接口/界面。它应非常具体，包含协议、端口和逻辑地址等，以便于按照接口/界面需求开发并检验软件。

#### 3.3.6.1用户界面

说明软件将实现的用户界面方面的要求。

#### 3.3.6.2 硬件接口

此节指出软件所支持的所有硬件接口，其中包括逻辑结构、物理地址、预期行为等

#### 3.3.6.3 软件接口

此节说明软件系统中与其他构件之间的软件接口。这些构件可以是购入的构件、取自其他应用程序重新利用的构件，也可以是为此需求范围之外的子系统开发，但该软件应用程序必须与之交互的构件

#### 3.3.6.4通信接口

说明与其他系统或设备（如局域网、远程串行设备等）的所有通信接口

### 3.3.7 法律、版权及其他声明

此节说明软件涉及的所有必需的法律免责声明、保证、版权声明、专利声明、字标、商标或徽标符合性问题

### 3.3.8 适用的标准

此节说明了所有适用的标准以及适用于所述系统的相应标准的具体部分。例如，其中可以包括法律、质量及法规标准；业界在可用性、互操作性、国际化、操作系统相容性等方面的标准

## 3.4 系统模型

该节使用UML中的概念、方法和模型图来详细表达需求

### 3.4.1场景

该节中针对每一个发起用例的参与者，选择代表性的场景进行描述。按照如下格式进行：

<场景一>:

* 场景名字: 为每一个场景起的具体名字
* 参与者实例：场景中涉及到的具体参与者人员
* 事件流程：按照步骤列出详细的流程

### 3.4.2用例模型

该节编写的具体步骤如下：

1. 列出系统中的所有参与者，并对每一个参与这做出解释。按以下格式进行填写：

|  |  |
| --- | --- |
| 参与者名称 | 参与者解释 |
| 参与者1 | 对参与者1的解释 |
| …… | …… |

1. 列出系统中所有用例，并给出简单介绍。按以下格式进行填写：

|  |  |  |
| --- | --- | --- |
| 用例 | 用例级别 | 用例描述 |
| 用例1 | 用户目标级别还是子功能级别 | 用例1的简单描述 |
| …… | …… | …… |

1. 详细介绍系统中的每一个用例。
2. 画出完整的用例图。

### 3.4.3对象模型

[分类别列出每一个类的介绍]

表 实体类定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 实体类名称 | 属性 | 关联类 | 定义 |
|  |  |  |  |
|  |  |  |  |

表 边界类定义

|  |  |
| --- | --- |
| 边界类名称 | 定义 |
|  |  |
|  |  |

表 控制类定义

|  |  |
| --- | --- |
| 控制类名称 | 定义 |
|  |  |
|  |  |

[列表方式介绍完类后，画出一张类图。]

### 3.4.4动态模型

1. 系统顺序图与操作契约

(1)用例X

* 为每一个用例画出一张系统顺序图
* 为该用例中的系统消息定义操作契约

1. 顺序图

[围绕系统消息画顺序图，为其中每一个系统消息单独画一张顺序图。]

1. 状态图

[给出某些具备复杂状态的类的状态图。]

### 3.4.5 用户界面

在需求获取阶段，可以给出部分用户界面，并在分析阶段继续完善。在介绍界面时，可以通过文字描述界面之间的转换关系。