# 1 引言

[引言提出的是一个整体介绍，有助于读者了解SRS是如何组织的，如何使用它。]

## 1.1 目的

[对产品或应用进行定义，在这个文档中说明产品或应用程序的需求，包括修订或者发行版本号。如果这个SRS只与某个复杂系统的一部分有关，就只定义这个部分或子系统。描述文档所针对的不同读者类型，如开发人员、项目经理、营销人员、用户、测试人员和文档作者]

例如：

**本软件需求规格说明书描述了自助餐厅系统（COS）1.0版本软件中的功能性和非功能需求。此文档由项目团队成员使用，以实现并检验正确的系统功能。除非另有说明，否则1.0版本中承诺包含这里所规范的所有需求。**

## 1.2 文档约定

[描述所用的标准或排版约定，包括具体的文本风格、高亮或符号的意思。如果是在手动标注需求，也许还要在此说明你采用的格式，以便他人后期再添加内容。]

例如：

**本SRS中未使用特定书面约定。**

## 1.3 项目范围

[简要描述所制定的软件及其目的。将软件与用户、公司目标及业务目标和策略相关联。如果有一个独立愿景和范围或其他类似文档可用，可以将其作为参考，但不要将内容复制到此处。如果软件需求规范说明规定要对一个演化产品进行增量式发布，那么就要将其自身的范围说明包含进来，并将其作为长期战略产品愿景的一部分。可能还要针对发布所包含的主要特性或者要完成的重要功能，提供一个概要性的小结摘要。]

例如：

**COS系统将允许Process Impact公司员工在线从公司自助餐厅订餐并送至园区指定的位置。在《自助餐厅系统愿景与范围文档》[1]中有详尽的描述，并且阐述了在此版本中规划的需要完整或部分实现的特性。**

## 1.4参考文献

[列举本软件需求规格说明所参考的文件或其他资源。如果参考文献的位置不变，也可以列出其超链接。这可能包括用户界面风格指南、合同、标准、系统需求规格说明、界面规格说明或者相关产品的软件需求规格说明。给出的信息要足够详尽，包括标题、作者、版本号、日期、出处、存储位置或URL，以便读者查阅每一个参考文献。]

例如：

**[1] 自助餐厅系统愿景与范围文档, Wiegers、Karl,**

[www.processimpact.com/project/COS/COS Vision and Scope.docx](http://www.processimpact.com/project/COS/COS%20Vision%20and%20Scope.docx)

**[2] Process Impact公司内网开发标准（1.3版）, *Beatty, www.processimpact.com***

***/corporate/standards/PI Intranet Development Standard.pdf***

**[3] Process Impact公司互联网应用用户界面标准（2.0版）, Rath、Andrew,**

***www.processimpact.com/corporate/standards/PI Internet UI Standard.pdf***

# 2 总述

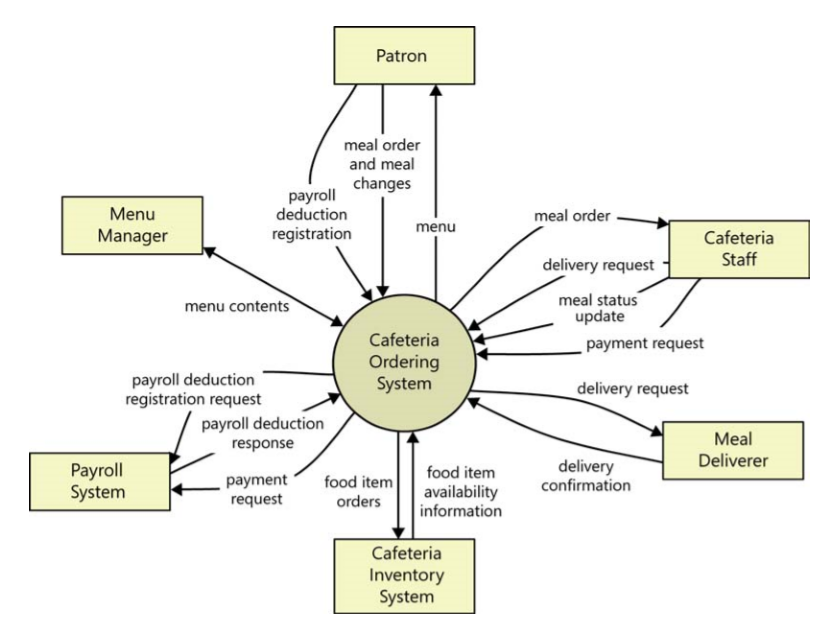
[这一部分高度概述产品及其使用的环境、预期的用户和已知约束、假设及依赖。]

## 2.1 产品概述

[描述产品的背景和来源。该产品是仍在发展中的产品系列中的下一成员？成熟系统的下一版本？还是现有应用程序的替代器？或是一个全新的产品？如果软件需求规格说明定义了大型系统的一个组成部分，那么就要说明该软件是如何与整个系统相关联的，并且要确定两者之间的主要接口。还要考虑将视觉模型包含进来，如上下文图或生态系统图，展示产品与其他系统之间的关联。]

例如：

**自助餐厅订餐系统是一套新的软件系统，用于替代Process Impact自助餐厅现在通过人工和电话处理订餐到取餐的方式。图R－1中的上下文图绘制了1.0版本的外部实体和系统接口。预期系统将经过多个版本的演化，并且最终为一些当地餐馆连接到互联网订餐服务以及支持借记卡和信用卡的授权服务。**



图R－1 自助餐厅订餐系统1.0版本的上下文图

## 2.2用户类别和特征

[确定你觉得可能使用该产品的不同用户类别并描述其相关特征。有些需求可能只与特定的用户类别相关。确定出你的重要用户。用户类别代表的只是干系人的一部分，这点在愿景和范围文档中会有所描述。对用户类的描述是一种可再利用的资源。如果你手头上有一份重要用户类别目录，就可以合并对用户类的描述，只需在目录中简单提及，无需在此复制信息。]

例如：

**表R－1 COS系统的用户类别和特征（下页）**

## 2.3 运行环境

[描述软件运行环境，包括硬件平台；操作系统和版本；用户、服务器和数据库的地理位置；容纳相关数据库、服务器和网站的组织。列出系统必须与之共存的其他软件组件或应用程序。如果还需要做一些额外的技术基础设施工作以便与开发中的新系统相结合，就要考虑创建一个单独的基础设施需求规范说明来细化工作。]

例如：

**OE-1：COS系统将可以在如下网页浏览器中正常使用：Windows Internet Explorer版本7、8、9；Firefox版本12至26；谷歌Chrome（全部版本）；苹果Safari版本4.0至8.0。**

**OE-2：COS系统将运行于当前企业审批通过版本的RedHat Linux服务器和Apache HTTP服务器。**

**OE-3：COS系统将允许用户从企业内网、VPN互联网连接或使用Android、iOS、Windows智能手机和平板电脑进行访问。**

|  |  |
| --- | --- |
| **用户类别** | **描述** |
| **顾客（重要）** | **顾客是希望从公司进行订外卖的Process Impact公司员工。潜在顾客大约有600名，预计其中平均每周每人使用COS系统5次。有时，顾客会为团队会议或访客预订多份。估计60%的订餐由公司内网下单，40%的订餐从家中或使用智能电话和平板电脑下单。** |
| **餐厅工作人员** | **Process Impact公司餐厅雇了约20名餐厅工作人员，他们将从COS系统接收订单、备餐，为外卖打包并请求送餐。大多数餐厅工作人员都需要接受培训，学习如何使用COS系统的软硬件。** |
| **菜单管理员** | **菜单管理员是一名餐厅雇员，负责创建和维护每天餐厅可供应菜品的菜单。有些菜单项可能无法外送。菜单管理员还会定义每天的特色菜。菜单管理员需要定期编辑已有的菜单。** |
| **送餐人员** | **当餐厅工作人员准备外卖订单时，他们会向送餐人员的智能手机发送送餐请求。送餐人员提取订餐送至顾客。送餐人员与COS系统的其他交互包括确认订餐是否送达。** |

表R－1 COS系统的用户类别和特征

## 2.4 设计与实现约束

[有时我们必须使用某种特定的编程语言，以及需要对特定代码库花费时间再次开发以便可以使用等等。描述限制开发人员选择的因素，并说明每个约束的基本原理。如果需要包含或者写入解决方案思路，而不是以需要为前提，就是在施加无意义的设计约束，所以要引起我们的警觉。]

例如：

**CO-1：系统的设计、代码、维护文档应遵循《影响力过程内网开发标准（1.3版）》。**

**CO-2：系统将使用当前企业标准的Oracle数据库引擎。**

**CO-3：所有HTML代码将遵循HTML5.0标准。**

## 2.5 假设与依赖

[假设就是在没有确凿证据或明确知识的情况下假定为真的说明。如果假设是错误的、过时的、不能共享或发生了变化，问题就会随之而来，因此某些假设会给项目带来风险。软件需求规格说明的某一个读者可能假设产品符合一个特殊的用户界面约定，而另一个读者却可能不这样认为；开发人员可能假设某组特定的功能是专门为这个应用程序而写，业务分析师可能假设该功能在以前的项目中使用过，而项目经理可能希望获取一个商业性功能库。这里所包含的假设要与系统功能相关；与业务相关的假设包含在愿景和范围文档中。

确定开发中的项目或者程序对外部因素或者其控制之外的组件存在的所有依赖关系。例如，在程序运行之前，必须安装Microsoft .NET Framework 4.5或更新版本，这就是依赖。]

例如：

**AS-1：认为员工在工作日在线，因此餐厅会在每个工作日开放早、中、晚三餐。**

**DE-2：COS系统的运行依赖于为了薪资系统能够接受COS系统的订单支付请求而做的修改。**

**DE-3：COS系统的运行依赖于为了在COS系统接受订餐时更新菜品项的有效性而对餐厅库存系统所做的修改。**

# 3 系统特性

[此模板采用的是由系统特性所组织的功能需求，而这只是众多组织方式中的一种。其他组织性选项还包括按照功能领域、工艺流程、用例、操作模式、用户类别、刺激和响应来排列功能需求。我们还可以对这些要素进行层级组合，例如将用例和用户类相结合。只要选择的组织方法便于读者理解产品的预期功能。]

**3. x 系统特性x**

[简单几个词就可以说明特性名称，如“3.1 拼写检查”。]

**3.x.1 描述**

[对系统特性进行简要描述，表明它级别是高、中还是低。优先级是动态的，往往在项目过程中不断变化。]

**3.x.2 功能需求**

[列出与此特性相关的具体功能需求。这些软件性能必须先完成，用户才能执行特性的服务或者完成用例。描述产品如何响应可预知的错误条件以及无效的输入和动作。]

例如：

## 3.1 从餐厅订餐

### 3.1.1 描述

**身份经过验证的餐厅顾客能够订餐，并要求送餐至指定的公司位置或到餐厅自取。顾客如果还没有准备好，可以取消或修改订餐订单。优先级＝高。**

### 3.1.2 功能性需求

|  |  |
| --- | --- |
| **Order.Place:**  **.Register:**  **.No:**  **.Date:**  **.Cutoff:** | **下订单**  COS系统将确保顾客注册了薪资。  如果顾客未注册薪资抵扣，COS系统就会让顾客选择立即注册并继续下单、下餐厅取餐订单（而不送餐）或者退出。  COS系统会提示顾客选择用餐日期。  若用餐日期时当天且当前时间晚于下单截止时间，COS系统就会告知顾客今天太晚以致无法下单。顾客可以变更用餐日期或取消订单。 |
| **Order.Deliver:**  **.Select:**  **.Location:**  **.Notimes:**    **.Times:** | **送餐或自取**  顾客将指定对订单进行送餐还是自取。  若需送餐且送餐日尚有可用送餐时间，顾客需要提供有效的送餐地点。  若送餐日已无可用送餐时间，COS系统将告知顾客。顾客可以取消订单或选择前往餐厅取餐。  COS系统将显示用餐日期剩余的可用送餐时间。COS系统允许顾客请求一个次送餐、修改订单为自取或者取消订单。 |
| **Order.Menu:**  **.Date:**  **.Available:** | **查看菜单**  COS系统将根据顾客指定的日期显示菜单。  所指定日期的菜单将只显示那些餐厅库存系统中至少有一个单位可用以及可送餐的菜品。 |
| **Order.Untils:**  **.Multiple:**  **.TooMany:** | **订多餐或多个菜品**  COS系统允许用户订多份相同的餐，但不能超过任何订单中菜单项的最低可用数量。  若顾客订购的数量超过当前餐厅库存系统中的数量，COS系统将告知顾客所能能订购的菜品的最大数量。 |
| **Order.Confirm:**  **.Display:**  **.Prompt:**  **.Response:**  **.More:** | **确认订单**  当顾客表示他不再需要订更多的菜品，COS系统就会显示所订的菜品项、每个菜的价格以及按照BR-12计算得到的应付金额。  COS系统将提示用户进行订餐订单的确认。  顾客可以对订单进行确认、编辑或取消。  COS系统将让顾客进行相同或不相同上期的订餐，BR-3和BR-4适用于单个订单中的多重订餐。 |

## 3.2 从饭馆订餐

[3.1 从餐厅订餐的描述的许多功能都可以复用。所以本节应当只规格说明针对饭馆的额外功能。 ]

### 3.2.1 描述

略。

### 3.2.2 功能性需求

略。

## 3.3 创建、查看、修改以及删除订下的菜品

### 3.3.1 描述

略。

### 3.3.2 功能性需求

略。

## 3.4 创建、查看、修改以及删除餐厅菜单

### 3.4.1 描述

略。

### 3.4.2 功能性需求

略。

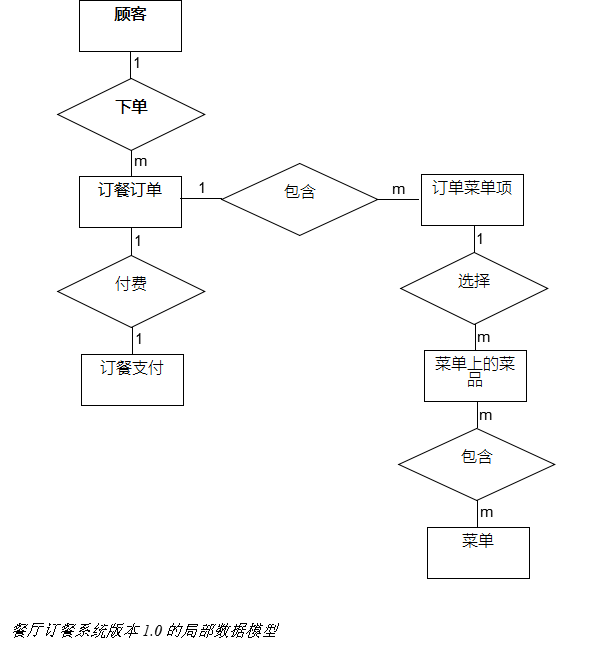
# 4 数据需求

[信息系统通过处理数据来提供价值。使用模板中的这一部分来描述各方面的数据，系统会将其作为输入消耗，将其以某种形式来加式，或者将其作为输入来创建。]

## 4.1 逻辑数据模型

[数据模型从视觉上呈现了系统要处理的数据目标和集合以及它们之间的关系。数据建模中含有大量的符号，包括实体关系图和UML类图。你可能还要为系统所强调的业务运作纳入一个数据模型，或者针对系统要处理的数据展示其逻辑关系。这与纳入一个数据模型不是一回事，这样的模型将会以数据库设计的形式来实现。]

例如：



## 4.2 数据字典

[数据字典定义数据结构的组成及其意义、数据类型、长度、格式以及组成这些结构的数据元素的允许值。商业数据建模工具通常包括一个数据字典组件。在大多数情况下，都应将数据字典存储为一个独立的工件，而不是将它嵌入在软件需求规范说明之中。这样一来，其他项目也可以重用它。]

例如：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据元素 | 描述 | 数据构成或数据类型 | 数据长度 | 数据取值 |
| 化学品申请 | 向化学品库房或供应商提出关于新化学品的申请 | 申请ID+申请人  申请日期  账户编号  +1：10{要申请的化学号} |  |  |
| 投递位置 | 被申请的化学品将要发往的目的地 | 建筑物  实验室编号  实验室部门 |  |  |
| 容器数量 | 指定化学品容器的数量或正在申请的化学品容器的数量 | 正整数 | 3 |  |
| 数量 | 化学品的申请数量 | 数字类型 | 6 |  |
| 计量单位 | 化学品的申请数量单位 | 字母表示的字符 | 10 | 克，千克，毫克 |
| 申请ID | 每个申请的唯一标识 | 整数 | 8 | 系统生成的序列号，是一个以1开头的整数 |
| 申请的化学品 | 申请的化学品的描述 | 化学品ID  容器数量  等级  重量  计量单位  （供应商） |  |  |
| 申请人 | 提出化学品的申请的申请人个人信息 | 申请人姓名  申请人编号  部门  发送地址 |  |  |
| 申请人姓名 | 申请人的姓名 | 字母表示的字符串 | 40 | 允许包含空格、横杠、句号、单引号 |

## 4.3 报告

[不管应用程序形成什么报告，都要将其在此确定出来并描述特征。如果报告必须要与某个具体的预定义的布局相吻合，可以将其定义为一个约束，可能还要有一个示例。否则，就将重点放在报告内容，排列顺序、总体水平等的逻辑描述上，并将详细的报告布局推迟到设计阶段。]

例如：

### 4.3.1 订餐历史报表

|  |  |
| --- | --- |
| 报表ID | COS－RPT－1 |
| 报表标题 | 订餐历史 |
| 报表目的 | 顾客希望查看此前6个月至今的某个时间段内，他曾在Process Impact公司餐厅或当地饭馆预定的所有订餐列表，以便能够再次订购自己喜爱的菜品 |
| 优先级 | 中 |
| 报表用户 | 顾客 |
| 数据源 | 之前存储订餐订单的数据库 |
| 频率和计划 | 报表根据用户需要进行生成。报表中的数据是静态的。报表能够在计算机、平板电脑或智能手机的浏览器中显示。如果显示设备允许打印，则可以打印报表。 |
| 延时 | 必须在顾客请求报表之后的3秒内展现给顾客 |
| 可视化布局 | 横向方式 |
| 页眉和页脚 | 报表页眉将包括报表标题、顾客姓名和指定的时间范围。打印时报表页脚将显示页码 |
| 报表主体 | 展现的字段和列头包括：   * 订单号 * 用餐日期 * 订餐自（“餐厅”或饭馆名） * 订购项（订餐订单中的所有项的列表及名项的数量、价格） * 餐品总价 * 税费 * 送餐费 * 订单总价（餐品总价、税费、送餐费之和） * 筛选条件：顾客指定的日期范围，含终点 * 排序条件：按先后顺序倒序 |
| 报表结束符 | 无 |
| 交互性 | 顾客可以进行下钻来查看订单中每个菜的原料和营养信息 |
| 安全访问限制 | 顾客只能查看自己的订餐历史 |