

第 3 章

需求分析

从本章开始进入项目文档的编写,需求分析文档是我们要写的第一个文档,是所有文档里最重要的文档,为后续的项目开发起到指导的作用,也是与客户建立良好关系的一座桥梁。

3.1 需求分析的作用

现今所使用软件出现的问题大部分源于收集记录、客户协商和修改产品的过程不够规范,包括客户没有明确提出需要的要求,或者收集信息的方法不当,或者与客户进行业务沟通的过程中理解错误以及相应需求变更等,大部分项目中出现的错误都是由需求分析阶段不够仔细、不够规范导致的,所以需求说明书是软件开发和项目管理活动的基础,遵循有效的需求过程将在开发中有以下好处。

- 在需求阶段(也就是开发初期)确保规范完善,可以减少返工,提高生产效率。
- 通过控制范围扩大和需求变更来满足项目的进度目标。
- 多与客户交流,达到更高的客户满意度。
- 做好计划,降低维护成本和技术支持的成本。

3.2 需求分析的结构

本章将通过“通讯录系统(ContactList System)”这个假想的小型项目阐述需求文档和图,包括以下内容。

- 应用的前景和使用的范围文档。
- 业务用例表图和用例描述。
- 软件需求规格说明。
- 需求分析模型。
- 相应的数据字典。
- 相应的业务规则。

通讯录系统是一个范例,主要给读者提供一种做需求分析的思路。也许不同的企业会有不同的规格要求,但大同小异。各种类型的需求信息之间是存在关联的,要求我们尽可能编写文档中每一部分的内容,基本目标是确保需求文档清晰明了、完整、易使用。

3.3 通讯录系统业务需求

1. 背景、业务机会和客户需求

随着网络和信息技术的发展,很多陌生人之间有了或多或少的联系,如何更好地管理通讯录成了问题。基于上述原因,开发一套通讯录管理系统势在必行。

2. 业务目标(Business Objective, BO)和成功标准(Success Criteria, SC)

本系统主要是为了更好地管理每个人的通讯录,给每个人提供一个井然有序的管理平台,防止手工管理混乱,造成不必要的麻烦。

3. 业务风险(Risk)

业务风险指移动端业务的冲击。

4. 系统边界

系统边界即系统包含的功能与系统不包含的功能之间的界限,一般在系统分析阶段定义,只有明确了系统边界才能继续进行下面的分析、设计等工作。本系统为计算机的桌面单机软件,不具备网络连接功能。用户通过本地软件进行通讯录数据的基本管理,包括数据的添加、修改、删除和查询。本软件主要是计算机用户用来管理通讯录中联系人信息的工具。

5. 术语定义

- 用户信息: 包括用户的姓名、性别、联系方式等基本信息。
- 通讯录: 用于存储和管理用户的基本信息。
- 计算机: 个人计算机。
- 添加信息: 通过软件的操作界面进行联系人信息的添加操作。
- 删除信息: 通过软件的操作界面进行联系人信息的删除操作。
- 修改信息: 通过软件的操作界面进行联系人信息的修改操作。
- 查询信息: 通过软件的操作界面进行联系人信息的查询操作。
- 用户界面: 呈现在用户面前的图形化用户界面(窗口)。
- 菜单栏: 位于用户界面上方的文字菜单项目,通过鼠标单击操作完成特定功能。
- 工具栏: 位于菜单栏下方,以工具按钮的形式呈现,通过鼠标单击操作完成特定功能。
- 按钮: 位于用户界面的特定位置,以按钮的形式呈现,通过鼠标单击操作完成特定功能。
- 查询框: 位于功能按钮上方,以输入框的形式呈现,通过键盘输入操作完成查询功能。

6. 系统说明

一个完整的通讯录管理系统应该包括系统管理模块、主窗体模块、通讯录数据管理模块、信息查询模块。本系统主要实现信息管理,包括联系信息的添加、修改、查询和删除,以及一些辅助项目的管理,例如登录系统用户管理和用户组管理。整个系统模块的划分如图 3-1 所示。

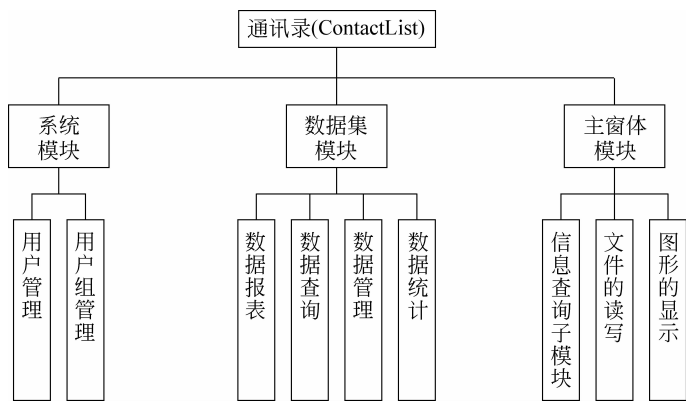


图 3-1 通讯录结构图

3.4 通讯录系统需求分析范例

1. 需求说明

本系统分为系统模块、数据集模块和主窗体模块。系统模块包含用户管理和用户组管理两个子模块；数据集模块包含数据报表、数据查询、数据管理和数据统计 4 个子模块；主窗体模块包含信息查询、文件读写和图形显示 3 个子模块。本系统的使用人员主要是拥有个人计算机并且需要管理大量联系人信息的用户。

2. 功能要求

- 联系人信息的显示功能：包括联系人列表、联系人头像和联系人性别统计信息的显示。其中，联系人列表应显示联系人的所有基本信息(姓名、性别、手机号、QQ 号、分组)；联系人头像应以图片形式显示；联系人性别统计信息应以柱状图的形式显示。
- 联系人信息的管理功能：包括菜单、工具按钮和功能按钮。其中，菜单应当实现的功能包括联系人基本信息的添加、编辑、删除、复制、备份和恢复，系统的退出以及系统基本信息的显示(显示系统的版本号等)；工具按钮应当实现的功能包括联系人基本信息的添加、编辑、删除、复制，以及系统基本信息的显示；功能按钮应当实现的功能包括联系人基本信息的添加、编辑、删除、复制、备份和恢复。
- 联系人信息的查询功能：此功能应当实现联系人基本信息(姓名、性别、手机号、QQ 号、分组)的模糊查询，查询结果以列表的形式显示。

3. 可用性

对于系统的用户而言，使用系统所需的培训时间应不超过 1 小时，即系统的使用方法应当足够简单，方便用户使用；对于系统的每一项功能，应当给出足够详细的用户操作指南。联系人信息的添加和编辑操作的响应时间应当在毫秒级；联系人查询操作的响应时间也应当在毫秒级。

4. 可靠性

软件可靠性是指在给定时间内特定环境下软件无错运行的概率。

软件可靠性包含以下 3 个要素。

- 规定的时间：软件可靠性只是体现在其运行阶段，所以将“运行时间”作为“规定的时间”的度量。“运行时间”包括软件系统运行后工作与挂起（开启但空闲）的累计时间。由于软件运行的环境与程序路径选取的随机性，软件的失效为随机事件，所以运行时间属于随机变量。
- 规定的环境条件：环境条件指软件的运行环境，它涉及软件系统运行时所需的各种支持要素，例如支持硬件、操作系统、其他支持软件、输入数据格式和范围以及操作规程等。不同的环境条件下软件的可靠性是不同的。具体来说，规定的环境条件主要是描述软件系统运行时计算机的配置情况以及对输入数据的要求，并假定其他一切因素都是理想的。有了明确规定的环境条件还可以有效判断软件失效的责任在用户方还是研制方。
- 规定的功能：软件可靠性还与规定的任务和功能有关。由于要完成的任务不同，软件的运行剖面会有所区别，所以调用的子模块不同（即程序路径的选择不同），其可靠性也可能不同，所以要准确度量软件系统的可靠性必须首先明确它的任务和功能。

5. 性能要求

本系统的使用需要达到如表 3-1 所示的性能要求。

表 3-1 性能要求

项 目	指 标	项 目	指 标
系统平台服务连续性	$\geq 99.98\%$	操作系统	Windows 7/8/10(含最新补丁)
系统的可用度	$\geq 99.98\%$		

6. 可维护性与可扩展性

考虑到应用的多变性，本系统采用层次化设计思想，每层实现特定的功能，并通过标准接口向上层提供透明的服务，因此最大限度地实现了系统中各模块功能的独立性，每一层的改动都不影响其他层次，这样可以方便地添加、修改、删除应用，提高系统的可维护性；管理系统采用图形化用户界面，方便管理员管理、维护系统。

系统在设计之初要考虑到以后的扩展，除了采用层次结构，保证功能实现与通信接口最大限度的独立以外，在与通信平台的接口设计上也应该尽量标准化，以实现和新系统的无缝连接。

7. 安全性

系统的管理系统与维护系统设置了用户管理权限，采用账号+操作日志，对管理员采取有效的管理，最大限度地降低内部管理、操作失误带来的不必要损失。

根据不同的需要，可以对数据库采取实时热备份或定时备份的策略。

- 软件故障：处理软件异常，不能产生无效数据以及造成数据丢失。
- 硬件故障：可快速更换硬件，以保证系统正常，且不可造成数据丢失。

8. 设计约束

1) 设备

- 奔腾 4 及以上。

- 内存 128MB 以上。
- 硬盘 1GB 以上。
- 10/100Mbps 网卡。

2) 支持软件

- 操作平台：Windows 7/8/10。
- 开发工具：Visual Studio 2012。
- 设计工具：Rose 2010。

9. 用户使用手册

在本书第 12 章列出了详细的用户使用手册,该手册以具体的操作实例介绍了通讯录系统中各项功能的使用方法。

10. 界面要求

- 用户界面：对话框。
- 硬件接口：无。
- 软件接口：微软基础类库。
- 通信界面：无。

11. 验收标准

- 软件可以正常运行。
- 实现项目软件需求说明书要求的各项功能(联系人信息的管理、显示和查询)。
- 软件界面友好,易于交互。

12. 参考资料

- 软件需求管理过程。
- 软件需求采集指南。

3.5 小结

本章对通讯录案例进行了简单的需求分析,对系统构架、程序环境、网络环境、成本预算等进行了定义。准确的需求定义可以提高开发效率,如果需求定义不完整,在开发过程中的变化会很多,特别是客户提的意见,所以需求收集是个很烦琐的过程,在进行需求分析时一定要和客户或相关部门一一确认,从客户的角度以及企业发展的角度把能考虑的都考虑进去,把功能加以强化、细化,防止返工。

第 4 章

概要设计

本章介绍概要设计在整个项目开发过程中的作用,以及该阶段涉及的元素,并通过通讯录系统(ContactList System)项目实际练习如何完成一个基本的概要设计,为后续的编码实现做好规范。

4.1 概要设计要求

概要设计把项目按照一定的业务逻辑关系分解为模块层次,赋予每个模块一定的任务,并确定模块间的调用关系和接口。

- 考虑内部实现,主要集中于划分模块、分配任务、定义调用关系。
- 模块间的接口与参数传递,编写数据字典,避免后续设计时产生不解或误解。
- 反复检查讨论,进行结构调整。
- 最大限度地提取可以重用的模块,建立合理的结构体系,节省后续环节的工作量。

4.2 概要设计结构

概要设计包括组成模块、模块的层次结构、模块的调用关系、每个模块的功能等,同时还要规划好数据类型,绘制好该项目的应用系统的总体数据结构和数据库结构关系图,即应用系统要存储什么数据,这些数据是什么样的结构,它们之间有什么关系。其作用如下。

- 为后续编码人员提供依据,或者为后期修改、维护提供条件。
- 做阶段性和总结性的质量验证与确认。

4.3 通讯录概要设计范例

1. 设计背景

随着网络与信息技术的发展,很多陌生人之间有了或多或少的联系,用户拥有的联系人的数量越来越大,联系人的信息也越来越复杂,如何更好地管理通讯录成为问题。基于上述需求,有必要开发一套通讯录管理系统。

2. 项目定义

本项目开发的软件约定如下。

中文全称：通讯录信息管理系统。
中文别称：通讯录。
英文全称：ContactList Information Management System。
本报告用到的术语符合国家标准《软件工程术语(GB/T 11475—1995)》。

3. 引用文档

与本文直接相关的国家标准(中国标准出版社,1996 年)如下。
GB 8566—1995：软件生存期过程。
GB 8567—88：计算机软件产品开发文件编制指南。
GB 9385—88：计算机软件需求说明编制指南。
GB/T 11457—1995：软件工程术语。

4. 设计方法和原则

软件系统进行概要设计的原则如下。
命名规则：系统模块、变量等按照功能的英文名命名,并且采用下画线命名法。
必需的安全措施：用户名、密码以及传送数据包进行必要的加密,采用对称的加密算法。

安全性和保密原则：数据文件备份到不同服务器,数据文件系统包括安全防护、安全检测与安全恢复机制。

系统灵活性要求：当需求发生变化时采用增减模块的方式具有一定的灵活性。
系统易操作性要求：系统输入尽可能采用“选择”而非直接输入。
系统可维护性要求：应维护简单。

5. 总体设计

在充分理解《通讯录系统需求说明书》的基础上,根据实际的联系人信息管理需要设计此方案来最大限度地满足用户的需要。

根据模块调用关系,给出图 4-1 所示的系统调用结构框图和图 4-2 所示的类继承关系图。



图 4-1 系统调用结构框图

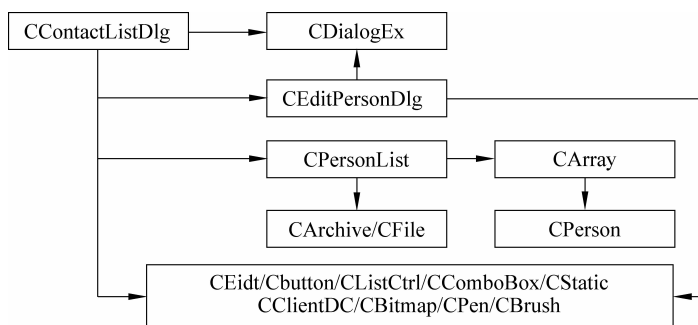


图 4-2 类继承关系图

6. 接口设计

用户界面设计规则：按照 MFC 界面设计符合用户使用的大方、美观的界面。

内部接口设计：系统采用同一个数据文件共享实现功能，数据库内部不需要接口，各子系统之间按照需要提取数据库中的数据。

硬件之间的接口：无。

软件之间的接口：与现有的管理系统之间的接口。

硬件接口：在输入方面，对于键盘、鼠标的输入，可用微软的标准输入/输出进行处理；在输出方面，可用微软的标准输入/输出进行处理。

内部接口：各模块之间采用函数调用、参数传递、返回值的方式进行信息传递，具体参数的结构将在数据结构设计的内容中说明。接口传递的信息是以数据结构封装的数据，以参数传递或返回值的形式在各模块间传输。

7. 出错和维护设计

出错信息：所有出错信息均以字符串的形式在弹出式窗口中显示。所有出错信息分为 3 种，一是由于输入超出或不符合预定格式的误差，属于处理错误；二是由于系统的预设不能执行的误差，属于设定错误；三是由于服务器响应超时而导致的误差，属于系统错误。

对于处理错误，需在操作成功判断及输入数据验证模块进行数据分析，判断错误类型，再生成相应的错误提示语句送到输出模块中；对于设定错误，应在开始提交信息类别中依据权限等判定错误类别，再生成相应的出错信息语句，输出到输出模块中；对于系统错误，应根据 Resin 服务器的响应内容判断错误类别输出。

补救措施：所有服务器都必须安装不间断电源，以防止停电或电压不稳造成的数据丢失的损失。若真断电，对服务器会有较大的影响，在断电后的恢复过程中可采用 Windows 的日志文件对其进行恢复处理，对数据进行恢复。在硬件方面要选择较可靠、稳定的服务器机种，以保证系统运行时的可靠性。

系统维护设计：在维护方面主要对服务器上数据文件中的数据进行维护，可使用 Windows 的数据文件维护功能机制。例如，定期为数据文件进行备份(Backup)，维护管理数据文件中数据的一致性。

4.4 小结

本章对通讯录项目进行了简单的概要设计,确定了软件的总体结构以及软件系统的组成部分和模块间的相互关系,其基本目的是简要地说明软件系统是如何实现的,所以概要设计也称为初步设计或总体设计。对应的详细设计是对概要设计的内容进一步细化和丰富,主要的作用是确定软件系统中各级模块内部的数据结构以及算法。