
架构设计说明书

项目名称：[项目名称]

项目代号：[项目代号]

编制人：[编制人]

编制日期：[编制日期]

修订记录			
版本	日期	撰写 / 修订人	备注 / 修订内容

目 录

架构设计说明书	1
1. 引言	5
1.1. 编写目的	5
1.2. 系统目标	5
1.3. 术语和缩写词定义	5
1.4. 参考资料	5
2. 需求规定	5
2.1. 系统功能	5
2.2. 系统性能	5
2.3. 故障处理要求	6
2.4. 软硬件要求	6
2.5. 其他需求限制条件	6
3. 总体结构设计	6
3.1. 系统体系结构	6
3.2. 系统开发的基础平台和关键组件	6
3.2.1. 外部基础平台和关键组件	6
3.2.2. 内部基础平台和关键组件	7
3.3. 总体结构	7
4. 子系统设计	7
4.1. 功能结构图 /类图	7
4.2. 功能定义	7
4.3. 功能需求与系统模块的关系	7
5. 接口设计	8
5.1. 用户接口	8
5.2. 外部接口	8
5.3. 内部接口	8
6. 系统数据结构设计	8
6.1. 逻辑结构设计	8
6.2. 物理结构设计	9
6.3. 配置文件结构设计	9
6.4. 数据结构与程序的关系	9
7. 算法设计	9
8. 运行设计	9
8.1. 运行模块组合	9
8.2. 运行控制	10
8.3. 运行时间	10
9. 系统安全	10
9.1. 8.1 系统安全	10
9.2. 8.2 数据安全	10
9.3. 8.3 备份与恢复	10

9.4.	8.4 出错处理	10
10.	系统维护设计	11
11.	系统部署	11
11.1.	软硬件环境	11
11.2.	系统部署	11
11.3.	系统升级	11
12.	遗留问题	11

1. 引言

1.1. 编写目的

说明编写这份架构设计说明书的目的，指出预期的读者。

1.2. 系统目标

本系统要达成的目标，应描述架构设计的细度及为下一步模块设计提供的支持。

1.3. 术语和缩写词定义

列出本文件中用到的专门术语的定义和外文首字母组词的原词组。

1.4. 参考资料

列出与本文密切相关的参考资料，可能包含如下内容：

本项目经核准的计划任务书或合同、上级主管的批文；

属于本项目的其它已发表的文件；

本文件中各处引用的文件、资料，包括所要用到的软件开发标准。

列出这些文件资料的标题、文件编号、发表日期和出版单位，说明如何能够得到这些资料。

2. 需求规定

说明影响架构设计的系统主要功能、输入输出、性能要求等的需求限制条件。

2.1. 系统功能

系统的功能性需求，主要列出对架构设计产品影响的典型功能要求、数据要求、输入输出要求和界面要求等。

2.2. 系统性能

主要包括

数据精度

执行效率

数据处理能力等的要求。

2.3. 故障处理要求

系统出现故障时的处理机制，比如：故障转移等。

2.4. 软硬件要求

软件需求规格中规定必须支持的软硬件环境。

2.5. 其他需求限制条件

包括：

事先约定必须要使用的商业上架系统 / 组件（比如：必须在 MS Office 上做二次开发）；

必须应用的技术（比如：必须使用 B/S 结构）；

必须要与某商业软件做接口（比如：必须与用友软件做接口）；

与旧版本的兼容升级关系等

3. 总体结构设计

3.1. 系统体系结构

确定系统采用何种体系结构，如：

单机版还是网络版？

C/S 结构还是 B/S 结构？

两层结构还是多层结构？

分布式数据库还是集中式数据库？

分布式应用程序服务器还是集中式应用程序服务器？

是采用 .Net 应用程序框架还是 J2EE 应用程序框架？

3.2. 系统开发的基础平台和关键组件

3.2.1. 外部基础平台和关键组件

确定系统开发的外部基础平台和关键组件，如：

是在 Windows 平台、Unix 平台或者 Linux 平台上开发？还是平台无关性？

是否是在某些软件上作二次开发？如 AutoCAD。

是否采用一些第三方的关键组件。

3.2.2. 内部基础平台和关键组件

- 确定系统开发所使用的公司内部开发平台、组件 / 控件、公共函数 / 单元等公共资源，如：
- 是否在 GSP平台上开发；
- 是否采用公司的报表控件；
- 是否复用公司其他产品的架构或资源等。

3.3. 总体结构

给出系统结构总体框图（包括软件、硬件结构框图），说明本系统的各模块的划分，扼要说明每个系统模块的标识符和功能，分层次地给出各模块之间的控制与被控制关系。

如果采用结构化分析方法，通常采用下面两种模型来描述总体结构：

- 结构图（层次图）：对系统自顶向下进行分解。
- 系统流程图：描述子系统内部的控制流和数据流

如果采用面向对象的分析方法，通常采用包（package）图来描述总体结构。包（package）图描述了子系统之间的包含关系和依赖关系。

4. 子系统设计

在总体结构设计中已经将系统划分为若干个子系统，在此需要对所有的子系统进行设计。一般应包括下列内容。

4.1. 功能结构图 / 类图

- 结构化设计：对结构图进行细化，分解到功能一级。
- 面向对象设计：对包（package）图进行细化，确定各包中包含的类。

4.2. 功能定义

说明上一步分解的所有功能组件（功能模块或者类）的功能。

序号	功能 / 类名称	功能描述
1		
2		

4.3. 功能需求与系统模块的关系

本条用一张矩阵图说明各项功能需求的实现同各模块的分配关系，用来实现 RTM(需求跟踪)表，如果已经采用了外部工具来做，此处可以省略。

需求跟踪表

	[系统模块 1]	[系统模块 2]	[.....]	[系统模块 m]
[功能需求 1]				
[功能需求 2]				
[.....]				
[功能需求 n]				

5. 接口设计

5.1. 用户接口

说明将向用户提供的命令和它们的语法结构，以及相应的回答信息。例如：脚本命令语法等。

5.2. 外部接口

说明本系统同外界的所有接口的安排包括：

- 软件与硬件之间的接口；
- 本系统与各支持系统之间的接口；
- 本系统支持扩展性预留的外部接口及扩展的方式。
- 如果需要对外提供 API 接口，列出 API 接口。

5.3. 内部接口

说明本系统之内的各个系统模块或元素之间的接口的安排，包括：

- 系统涉及的全局变量与接口函数；
- 本系统支持扩展性预留的内部扩展点及扩展的方式等。

6. 系统数据结构设计

数据库结构设计遵循数据库设计规范，此规范文档单独提供。

6.1. 逻辑结构设计

给出本系统内所使用的每个数据结构的名称、标识符以及它们之中每个数据项、记录、文卷和系的标识、定义、长度及它们之间的层次的或表格的相互关系。

6.2. 物理结构设计

给出本系统内所使用的每个数据结构中的每个数据项的存储要求，访问方法、存取单位、存取的物理关系、设计考虑和保密条件。

6.3. 配置文件结构设计

给出本系统内所使用的每个配置文件中数据结构中的每个数据项的存储要求，访问方法、具体的含义和设计考虑等。

6.4. 数据结构与程序的关系

说明各个数据结构与访问这些数据结构的各个程序之间的对应关系。

	[模块 1]	[模块 2]	[.....]	[模块 m]
[数据结构 1]				
[数据结构 2]				
.....				
[数据结构 n]				

7. 算法设计

算法设计也可以在详细设计或实现阶段进行；

对涉及到大量数据运算和实时性要求高的组件，需对将采用的算法进行阐述、论证，以确保合理使用系统资源、满足需求规格说明书中的系统性能要求。

8. 运行设计

运行设计主要考虑软件在对外发布时所提供的根据用户特点的定制能力。比如：通过修改配置文件来显示或隐藏某些模块，通过修改配置文件来组合某些模块等

8.1. 运行模块组合

说明对系统施加不同的外界运行控制时所引起的各种不同的运行模块组合，说明每种运行所历

经的内部模块的支持软件。

模块集合	运行条件	支持软件

8.2. 运行控制

说明每一种外界的运行控制的方式方法和操作步骤。

运行名称	控制方法	操作步骤

8.3. 运行时间

说明每种运行模块组合将占用各种资源的时间，此项一般可以省略。

运行名称	所占资源	时间

9. 系统安全

9.1. 系统安全

- 1．系统安全控制和物理保护措施，比如：程序加密（软件锁） 。
- 2．用户身份鉴别机制；
- 3．用户对系统的访问权限和范围。

9.2. 数据安全

根据《需求规格说明书》中的系统保密要求，对系统进入及数据保密功能进行设计。通常指：数据加密、数据用户身份鉴别、数据库本身的安全（比如异常情况的数据保护、并发控制）等。

9.3. 备份与恢复

- 1．数据备份；
- 2．数据恢复；
- 3．恢复及再启动技术。说明将使用的恢复再启动技术，使软件从故障点恢复执行或使软件从头开始重新运行的方法。

9.4. 出错处理

- 1) 出错的情况；

-
- 2) 出错时信息输出的形式、信息的含义和反馈处理方法；
 - 3) 出错失效后的后备处理措施。

10. 系统维护设计

维护功能是系统的重要组成部分，应提供系统维护和管理的方法和工具。此处说明为了系统维护的方便而在程序内部设计中作出的安排，包括在程序中专门安排用于系统检查与维护的检测点和专用模块及工具。

11. 系统部署

11.1. 软硬件环境

列出运行该软件所需要的软硬设备及其专门功能，如果是多层结构需要分别列出各层所需的软硬件环境。

11.2. 系统部署

列出系统的部署方式，绘出系统的部署图。

11.3. 系统升级

旧版本的升级方式；

当前版本的未来版本的升级方式，比如：智能客户端等。

12. 遗留问题

说明在架构设计过程中尚未解决而设计者认为在系统完成之前必须解决的各个问题。

13. 需要模块设计文档的模块清单

公共类、公共函数清单

复杂模块、复杂算法清单

特别影响性能的模块 / 算法清单（重点在架构设计中考虑）