<项目名称>

软件架构设计

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](#_Toc448817922)

[1.1编写目的 1](#_Toc448817923)

[1.2适用范围 1](#_Toc448817924)

[1.3定义 1](#_Toc448817925)

[1.4参考资料 1](#_Toc448817926)

[1.5概述 1](#_Toc448817927)

[2 目前软件系统体系架构 1](#_Toc448817928)

[3 软件系统架构设计目标 2](#_Toc448817929)

[4 建议的软件系统架构 2](#_Toc448817930)

[4.1 概述 2](#_Toc448817931)

[4.2 用例视图 2](#_Toc448817932)

[4.3 系统逻辑视图 2](#_Toc448817933)

[4.4系统运行视图 3](#_Toc448817934)

[4.5系统实现视图 3](#_Toc448817935)

[4.6系统物理视图 3](#_Toc448817936)

[4.7 边界条件设计 3](#_Toc448817937)

[4.8数据管理设计 3](#_Toc448817938)

[4.9 其他设计 3](#_Toc448817939)

软件需求规约

# 1引言

## 1.1编写目的

说明：编写这份软件架构设计的目的，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，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 目前软件系统体系架构

对目前存在的系统进行描述，如果原来没有系统，则对相类似的系统进行描述。指出目前系统存在的问题，新系统希望得到的改进。

对原来系统架构的描述可以采用UML包图，或者部署图。

# 3 软件系统架构设计目标

本节说明对构架具有某种重要影响的软件需求和目标，例如：安全性、保密性、市售产品的使用、可移植性、分销和重复使用。还应记录可能适用的特殊约束：设计与实施策略、开发工具、团队结构、时间表、遗留代码等

# 4 建议的软件系统架构

## 4.1 概述

对整个软件的架构进行概述，给出软件架构采用的架构模式及其选择理由。

描述软件架构中重用的框架、中间件和类库。

简略描述包含哪些子系统，每个子系统的功能是什么。

1）客户机／服务器模式（ ｃ ｌ ｉ ｅｎ ｔ ／ ｓ ｅ ｒ ｖ ｅ ｒ，Ｃ／Ｓ）

２）模型视图控制器模式（ ｍｏｄ ｅ ｌ，ｖ ｉ ｅｗａｎｄｃ ｏｎ ｔ ｒ ｏ ｌ ｌ ｅ ｒ，ＭＶＣ）

## 4.2 用例视图

本节选择列出所有用例中的某一些用例或场景，这些用例或场景应体现最终系统中重要的、核心的功能；或是在构架方面涉及范围很广（使用了许多构架元素）；或与构架中某一特别的设计有关系。

## 4.3 系统逻辑视图

1. 系统架构

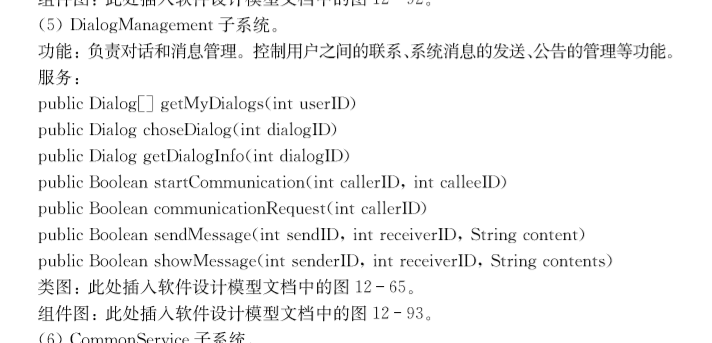
首先利用UML的包图，给出一个系统架构图，简要介绍该系统架构的特点，各个子系统的功能；

Xxxxxxxxxxxx(copy)

1. 子系统

针对每一个子系统描述其：

* 子系统的功能
* 子系统向外界提供的服务的详细描述（定义其接口具体形式）
* 给出每一个子系统的组件图、类图



1. 用例实现

针对选择的每一个核心用例，用顺序图刻画其具体实现(copy)

1. 子系统协作

以交互图的形式围绕核心用例，刻画子系统之间的协作过程。此时，每一个子系统是一个黑盒子，在交互图中作为一个对象。

|  |
| --- |
| CreateRoom |
| JoinRoom |
| SelectRole |
| Prepare |
| GameProcess |
| Move |
| Attack |
| Use |
| EndOfGame |
| ShowResults |

## 4.4系统运行视图（copy）

该部分首先描述系统的控制流结构的选择，例如是采用数据流驱动、事件驱动，还是采用多线程。

如果软件中涉及到并发的情形，提供相应的进程视图给以相应的解释。

## 4.5系统实现视图(copy)

该部分分两部分。

1. 系统开发模型：首先说明开发环境、开发语言、版本控制工具的选择，在开发环境中一个项目包含哪些目录，包含哪些源文件，用组件图来说明源文件包含的类，以及源文件之间的依赖关系。
2. 系统实现模型：定义编译后的可执行文件及其依赖关系。通过组件图来表示可执行文件及其依赖关系。

## 4.6系统物理视图

利用UML的部署图描述系统的软硬件部署方式。列出硬件配置的规格要求以及选择理由。

## 4.7 边界条件设计

描述了系统中如何启动、关闭以及进行错误处理。描述的方式是采用用例的方式来进行说明，即提供启动用例、关闭用例、错误处理用例的分别说明，并通过交互图来说明这些用例是如何实现的。

打开server

关闭server

打开app

打开app

异常

## 4.8数据管理设计

采用UML的类图表示那些需要持久化的数据。

说明选择的数据持久化保存的方式及理由。

如果是数据库保存，确定所选的数据库管理系统，并给出数据库的表设计，如果是文件系统，说明文件的目录结构，文件的格式。

写表格

## 4.9 其他设计

列出针对设计目标提出的特别的设计考虑，比如说访问控制和安全，可靠性设计等。

（1）访问控制和安全设计

采用表格列出不同的操作者对不同对象的权限；

描述用户认证的方式；

如果需要，给出数据的加/解密方式、给出接口调用的安全认证；

以及其他安全问题。

（2）可靠性设计

如果有特定的可靠性要求，给出在可靠性方面采取的特定设计方案。

