

# 概要设计规约（说明书）

修订历史:

修改或初始编写日期、 SEPG、 版本、 说明、 作者、 评审时间、 评审参与  
人员、 评审后修改批准日期、 确认签字人员

## 目 录

### 1. 引言

#### 1.1. 概要设计依据

#### 1.2. 参考资料

#### 1.3. 假定和约束

### 2. 概要设计

#### 2.1. 系统总体架构设计

#### 2.2. 系统软件结构设计

#### 2.3. 接口设计

#### 2.4. 界面设计

#### 2.5 数据库设计

#### 2.6 系统出错处理设计

### 2.6.1 出错信息

### 2.6.2 补救措施

### 2.6.3 系统维护设计

## 附录

### 软件需求规格(说明书)模板解释及说明：

修订历史：描述修订人、时间及在哪个版本上修订。如果是迭代开发每个版本对应一次迭代。非迭代开发每次正式的修改标记为一个版本。可以用以下的表来体现：

修订历史记录

编写日期	SEPG	版本	说明	作者	。。。。
2019.11.16	All members	1.0.0	基于 Netty 的 mvc	崔焱、蒋伟 博、吴国栋、 杨雨奇	
2019.12.8	All members	1.0.1	基于 websocket 的 client-server 直连	崔焱、蒋伟 博、吴国栋、 杨雨奇	

2020.01.04	All members	1.0.2	基于 websocket 的 client-server 直连  基于 spring 的 mvc  基于 flask 的 ai 服务器	崔焱、蒋伟 博、吴国栋、 杨雨奇	
------------	-------------	-------	---	------------------------	--

## 1. 引言

### 1.1. 概要设计依据

需求分析规约

### 1.2. 参考资料

软件工程：实践者的研究方法（第八版）Roger S. Pressman 著

### 1.3. 假定和约束

开发期限：至 2020.1.8

版本控制：git

## 2. 概要设计

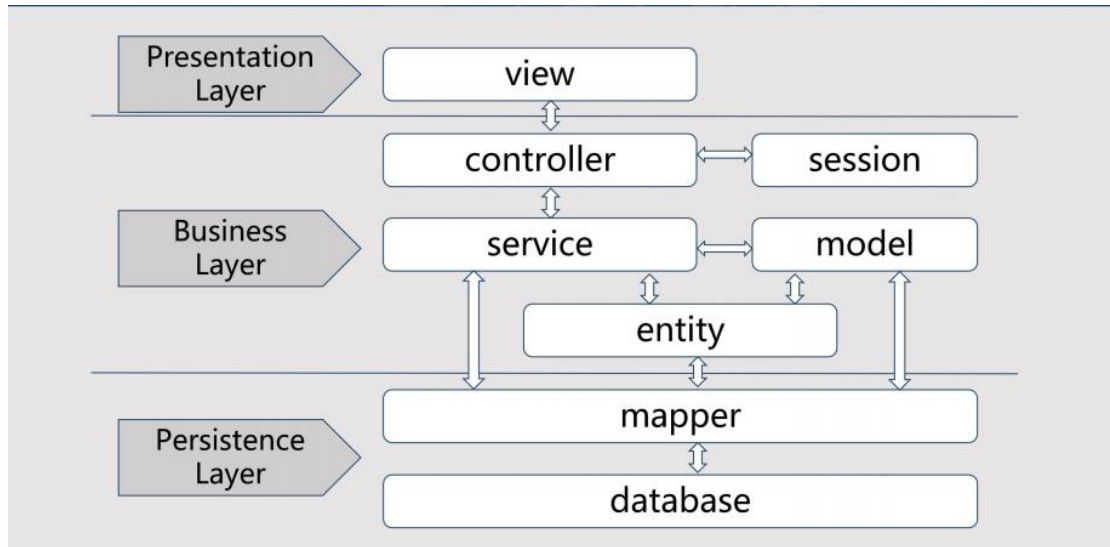
## 2.1. 系统总体架构设计

groupid	artifactId
org.springframework.boot	spring-boot-starter-web
org.springframework.boot	spring-boot-starter-websocket
org.mybatis.spring.boot	mybatis-spring-boot-starter
mysql	mysql-connector-java
com.alibaba	fastjson
org.springframework.boot	spring-boot-devtools

至于架构,要实现的架构是用源/架构编写的。让我们看看项目中包含的依赖项。

1. Spring Boot Starter 网站。 它允许我们创建一个基于 Spring Boot 的 MVC Web 程序。
2. 网络套接字启动器。 它允许会话进行全双工连接,这意味着浏览器和服务端并发送消息种子主动消息,只要保持会话连接活。
3. mybatis。 这是一个广泛使用的持久层框架,用户包括阿里巴巴等主流公司。
- 4.数据库我们使用 MySQL 作为存储库,使用 jdbc 进行 Java 连接。
- 5.我们以 json 格式来回发送消息。 因此,我们还导入了 json 依赖项。
6. Dev-tools 用于热部署,编译器可以在其中重新构建并重新运行项目时间很短。

## 2.2. 系统软件结构设计



上图是我们的主要架构。我们采用了经典的 3 层 Web 框架。

当消息从表示层到达业务层时，第一个模块是控制者。首先负责取消请求并处理消息开始。它还用于控制网络连接和响应发送。例如，在 Web 套接字，控制器需要记住会话，以便以后使用消息交换。然后，请求进入服务层。它用作操作逻辑处理。最简单的在不涉及特定数据的情况下，它可以计算然后返回结果而无需持久层。但是，这种情况越来越少，因为我们可以考虑到 BS，几乎可以使用 JavaScript 在表示层完成所有这些操作框架项目。当涉及数据库连接时，使用映射器进行数据库访问管理，有时不同线程中的服务可能需要数据库同时。关于发送和接收对象，除了普通对象（如 int 或字符串，一个实体对象适用于此功能。对于一些不需要的临时对象存储在数据库中，将它们存储到模型中，它们就可以进行数据自我管理。另外，模型可以与持久层交互。

### 2.3. 接口设计

对 2.2 节中的类之间的接口，模块之间的接口，对接口作详细描述。

本系统和其他系统的接口，对接口作详细描述。

2.3.1 登陆接口

<b>/login</b>				
接口描述 : 验证用户登陆,返回用户登陆状态				
Request Method : POST				
Data	Located in	Description	Required	Schema
username	json	用户名	YES	String
password	json	密码	YES	String
返回结果				
CODE	Description	Schema		
true	Successful response	String		

2.3.2 注册接口

<b>/register</b>				
接口描述 : 验证用户注册,返回用户注册状态以及个人战绩概况				
Request Method : POST				
Data	Located in	Description	Required	Schema
username	json	用户名	YES	String
password	json	密码	YES	String
返回结果				
CODE	Description	Schema		
true	Successful response	String		
win	胜场	Integer		
coins	欢乐豆	Integer		
lose	败场	Integer		

2.3.3 历史战绩接口

<b>/getHistory</b>				
接口描述 : 返回历史战绩详情				
Request Method : POST				
Data	Located in	Description	Required	Schema
id	json	用户id,自动分配	YES	Integer
返回结果				
Data	Description	Schema		
state	战绩是否存在	String		
coins	获利情况	Integer		
gameDate	对局日期	String		

### 2.3.4 好友列表接口

<b>/getFriendlist</b>				
接口描述 : 返回好友战绩				
Request Method : POST				
Data	Located in	Description	Required	Schema
id	json	用户id,自动分配	YES	Integer
返回结果				
Data	Description	Schema		
username	好友用户名	String		
coins	欢乐豆数量	Integer		
win	胜场	Integer		
lose	败场	Integer		

2.3.5 任务列表接口

/getTasklist				
接口描述 : 返回任务列表				
Request Method : POST				
Data	Located in	Description	Required	Schema
id	json	用户id,自动分配	YES	Integer
返回结果				
Data	Description	Schema		
taskId	任务号	String		
currentProgress	任务完成度	Integer		

2.3.6 简单 AI 出牌接口

/shotPokers/general				
接口描述 : 得到简单AI的出牌信息				
Request Method : POST				
Data	Located in	Description	Required	Schema
table_id	json	对战桌号	YES	Integer
robot_id	json	机器人编号	YES	Integer
返回结果				
Data	Description	Schema		
nextshot	出牌数组	Integer[]		

2.4. 界面设计



好友列表 历史战绩

用户名	金币	胜场	负场
dandanroubing	1000	3	3
danchaofan	1060	2	2
computer	1000	0	0
rttttt	1000	0	0
spring	1000	0	0

欢迎回来 pay 金币:1100 胜率:0.71






等待玩家加入


房间:

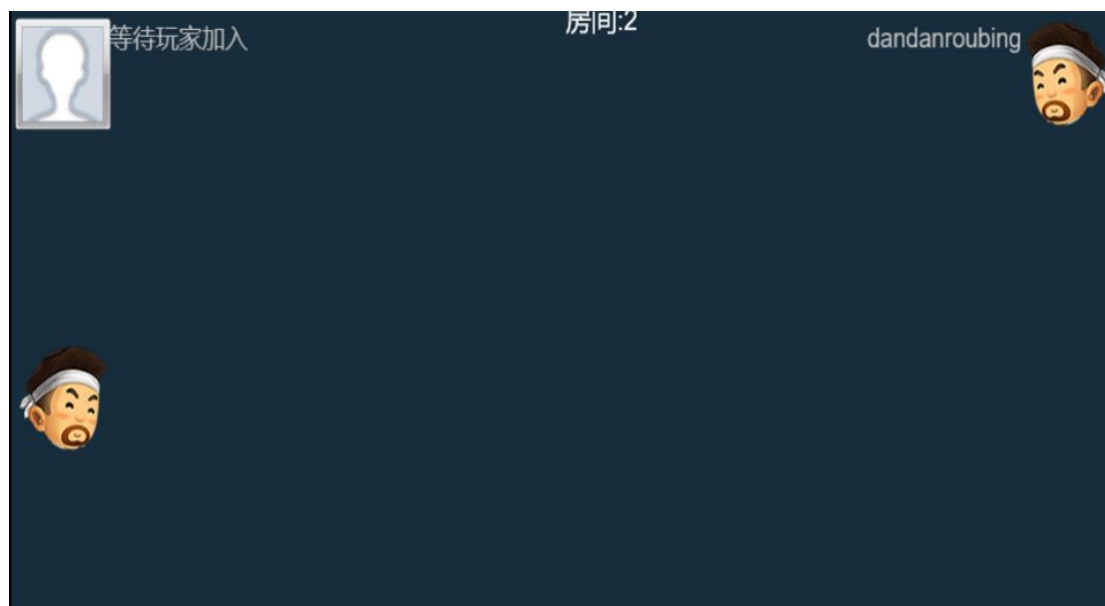
等待玩家加入





新建房间





## 2.5 数据库设计

No.	name	property
1	login	(userName, pwd, userId)
2	user	(userId, nickName, money, ...)
3	friend	(userId, friendId)
4	proverties	(userId, proverty1, proverty2, ...)
5	assignment	(userId, assignment1, assignment2,...)
6	garment	(userId, garment, garment2, ...)
7	matchInfo	(matchId, totalhands, userId1, userId2, userId3, matchTime)
8	handsRecord	(matchId, handId, bigPoker, smallPoker, handType, time)

Login: 用于登录和注册功能。

User: 有关用户的基本信息。 大多数都可以在之后查看和修改用户进入游戏。

Friend: 这是一个连接表，用于将彼此的朋友关系与用户 ID 的格式。

Task: 这是一个实时记录玩家任务完成情况的表。

Proverties: 对于这些奖励和未使用的衣服，它们将被存储在这里。

Assgnment: 它记录分配的共谋性。

Garment: 对于仍在使用但尚未淘汰的持久服装, 时间开始使用服装记录在这里。

MatchInfo: 对于记录的特定比赛，一些基本信息存储在此表。

HandsRecord: 在一场比赛中，每位球员进行很多手牌。 这里我们会记录时间，扑克信息，以重新显示比赛。

## 2.6 系统出错处理设计

### 2.6.1 出错信息

可以用一览表的形式说明每种可能出现错误或故障情况发生时,系统输出信息的形式、含义及处理的方法。

### 2.6.2 补救措施

说明故障出现后可能采取的变通措施。

### 2.6.3 系统维护设计

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的检查与维护的检测点和专用模块(或)类。