

设计文档

2009.9.17

系统模型

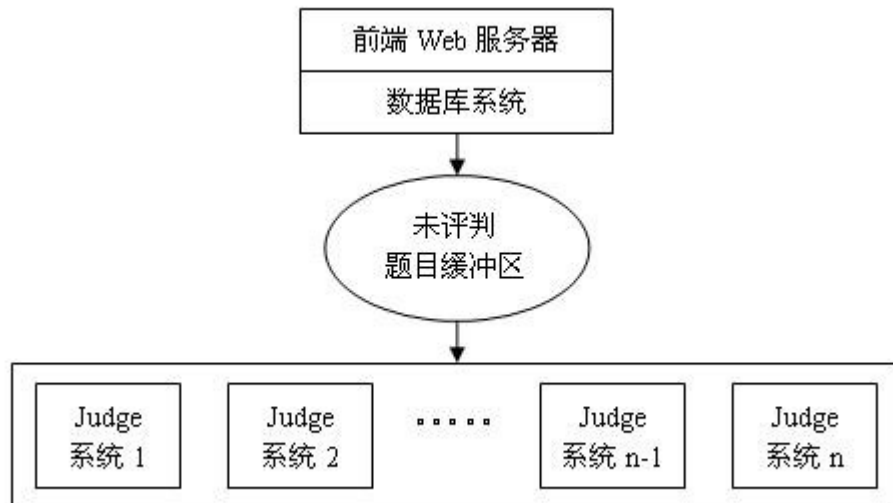
• 前端 Web 系统

- 即给每个用户使用的客户端，负责接受用户的命令，将程序提交给服务器，并将服务器返回的结果显示给用户。对于这种类型的应用来说，有两种模式可以选择，一为客户/服务器（C/S）模式，二为浏览器/服务器（B/S）模式。
- C/S 模式有很多优点，比如能够实现更丰富的用户操作方式，给用户更加丰富的用户体验。但它也存在很根本的缺点，即非常不容易管理，需要在每一个客户端安装程序。相对来说，B/S 模式就要方便很多，只需要客户端有一个浏览器就行。在线评判系统的目标是要在校园网上开展大型的练习或竞赛，同时，用户的复杂性操作的需求不显著，所以 B/S 模式是在线评判系统的最佳选择。

• 评判模块

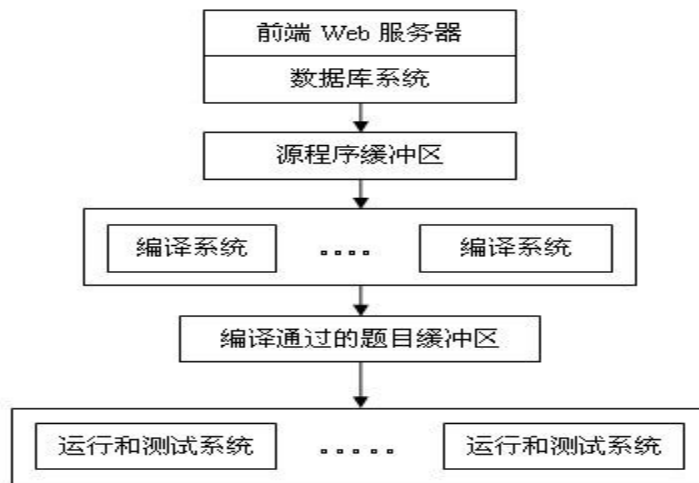
- 对用户提交的解答进行评判。在目前的设计中，包含了两大类的评判器：
 - （1）本地评判器。本地评判器和评判系统框架执行在同一个机器上。
 - （2）远程评判器。远程评判器可以运行在不同的机器上，形成一个由多台计算机组成的集群，以大大提高系统的可伸缩性。
- 服务器在评判过程中，不管是编译阶段，程序运行阶段，还是用例测试阶段，其工作量是相当大的。特别是当程序提交数量很多时，可能会造成提交的问题需长时间的等待才能够给予评判，这不仅影响了评判进度，还会影响前端 Web 服务器的执行速度。解决该问题的一种方案是提升服务器的硬件性能指标，如使用工作站，甚至巨型机当服务器，或分布式系统计算机。另一种解决方案是利用基于网络的多服务器系统并行执行。不管采用哪一种方案，在 Judge 服务器设计过程中，应尽量提高评判过程中的并行执行特性，如多线程，多进程，多服务器，流水线系统结构等。
- 并行执行的系统结构是通过多线程，多进程或多 Judge 服务器并行执行一个

程序评判的全过程。多个 Judge 服务器通过缓冲区与评判系统的前端 Web 服务器耦合。缓冲区是一个临界资源，必须采取相应的机制（信号量，互斥锁）使得多 Judge 服务器的同步与互斥。其参考模型如下图所示。



并行系统结构

- 其中每台 Judge 服务器都相对独立的执行一个程序的评判全过程（即编译，运行和测试）。
- 流水线式系统结构借鉴计算机组成原理中的流水线式系统设计模式。它是将一个任务分成相对独立的几个阶段，系统同时运行各个阶段，每个任务连续的从第一个阶段流向最后一个阶段完成一个任务的整体。
- 该评判系统后端 Judge 服务器分为两个阶段：编译系统模块，运行和测试系统模块。各个模块通过缓冲区衔接过度，其中个缓冲区也是临界资源。系统参考模型如下图所示。



流水线系统结构

- 从图中可以看出，每个系统相对独立地执行一个程序评判过程的相应阶段，但是一个题目的评判有可能编译或运行没有正常通过，因此并不一定流过该评判系统流水线的每一个阶段。

功能规格说明：

- 用户：注册，修改密码，登录，提交程序，显示结果。。。
- 管理员：添加题目，编辑题目，添加比赛。。。

性能规格说明：

- 精度：对输出结果要求与标准结果完全相同。
- 时间特性要求在 2000ms 内反应出结果。
- 灵活性可使用多种语言（Java、C、C++，Pascal 等语言）提交代码。
- 安全性可安全处理用户提交的恶意代码。
- 可维护性：可以由系统管理员对数据库进行维护。
- 可扩展性：可以添加多种其他的功能，如：增加虚拟币，购买他人的代码

- 输入输出要求：输入：以某种语言（java、C、C++等）编写的可以解决评测系统内特定问题的程序。输出：评测结果（right、wrong）、排名等统计信息。

详细设计

实体类的提取

1 用户提交问题

Candidate key abstraction	Eliminated for the following reason	Selected component name
User Name	Attribute of user	
Problem ID	Attribute of problem	
Source Code	Attribute of status	
Status		status

2 系统判题

Candidate key abstraction	Eliminated for the following reason	Selected component name
Result	Attribute of status	
Deamon Process		Deamon
Rank		Rank

3 用户注册

Candidate key abstraction	Eliminated for the following reason	Selected component name
User		User
User name	Attribute of user	
Sex	Attribute of user	
Password	Attribute of user	
E-Mail	Attribute of user	
Register time	Attribute of user	

4 查看当前状态

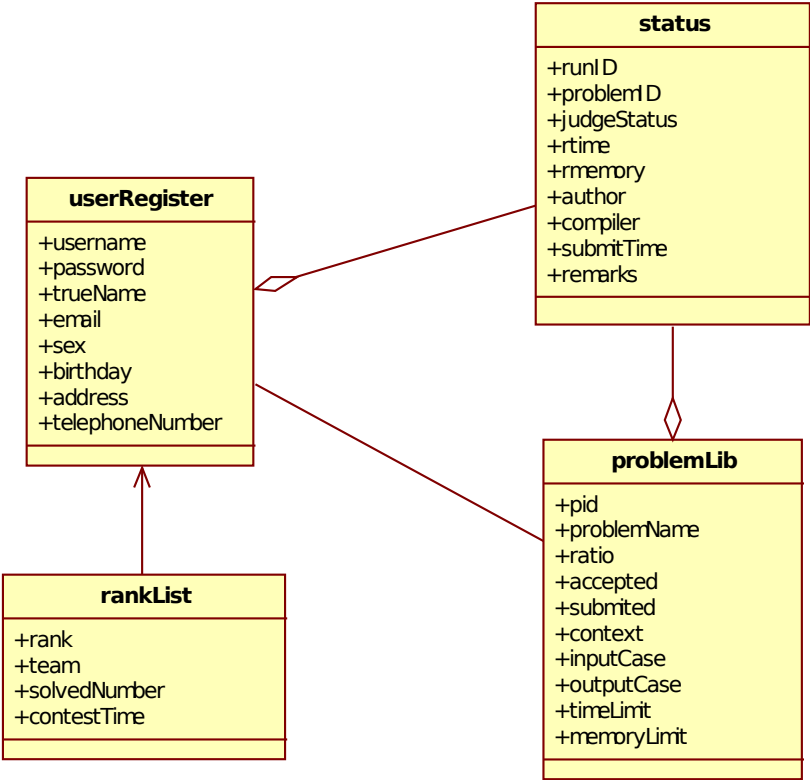
Candidate key abstraction	Eliminated for the following reason	Selected component name
Status		Status
Status ID	Attribute of status	
User ID	Attribute of user	

Problem ID	Attribute of problem	
Result	Attribute of status	

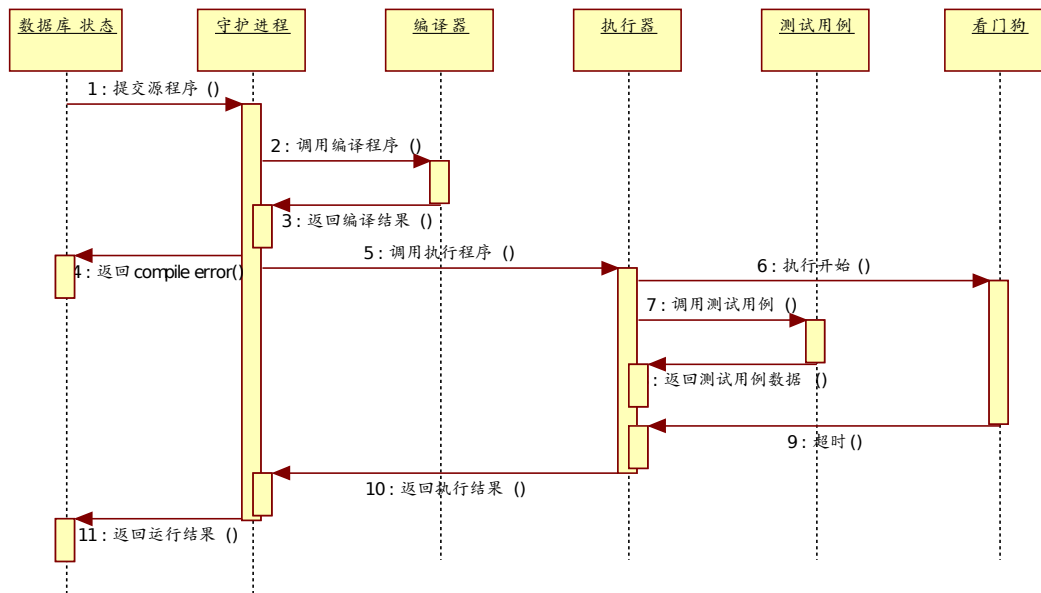
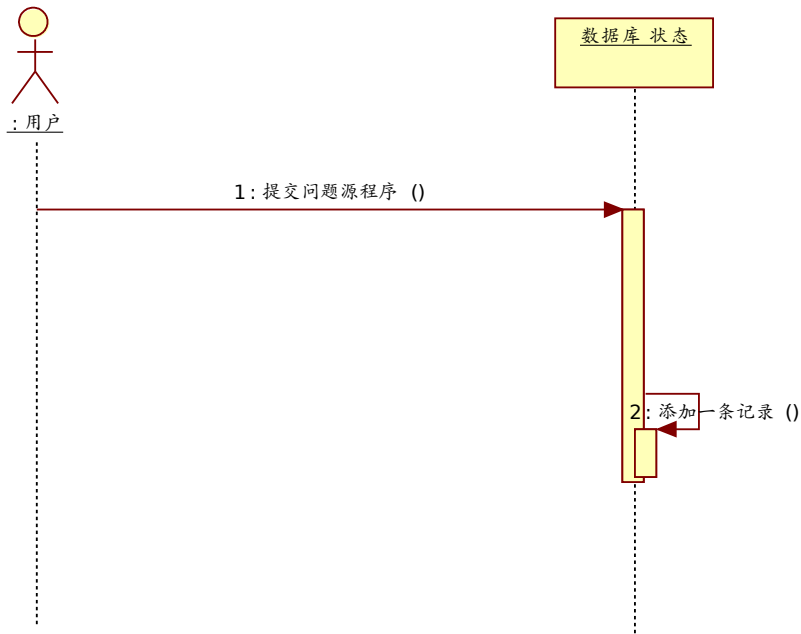
5 查看用户排名

Candidate key abstraction	Eliminated for the following reason	Selected component name
User ID	Attribute of user	
Solved Problems	Attribute of rank	
Rank		Rank

实体类关系图

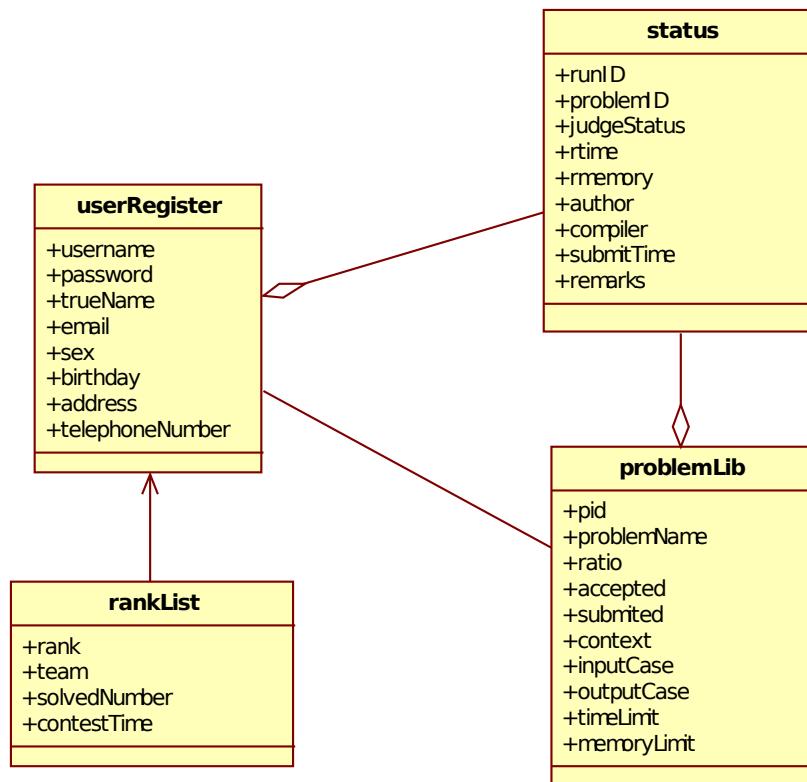


各用例鲁棒性分析

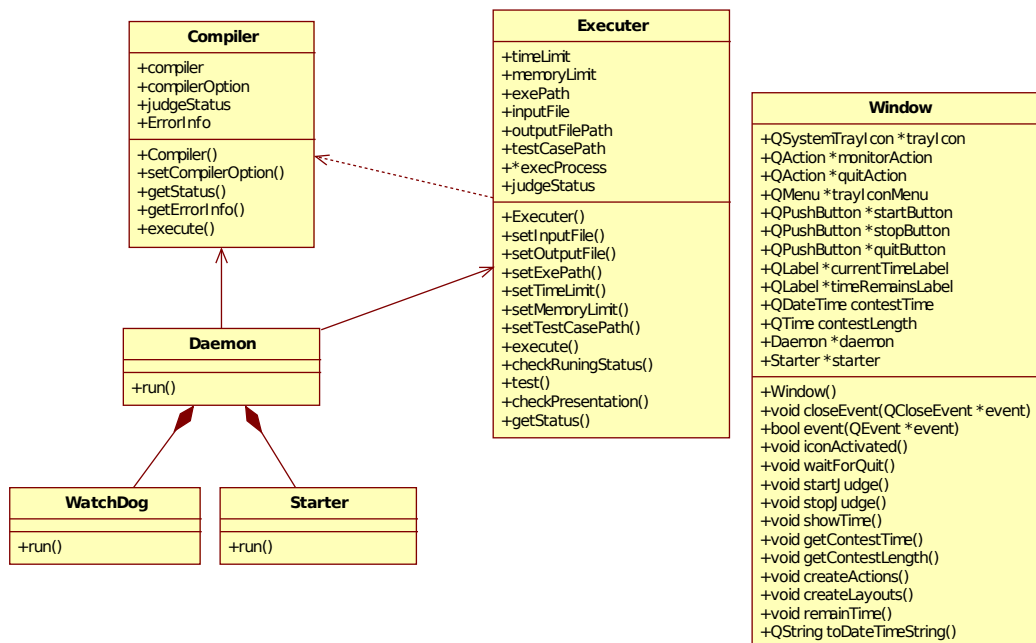


类详细设计

实体类的关系：



控制类的关系：



类详细说明：

实体类：

1、

userRegister
+username +password +trueName +email +sex +birthday +address +telephoneNumber

属性： username 用户名（包括管理员与普通用户）

 Password 密码

 trueName 用户真实姓名

 email 用户有效邮箱地址

 sex 用户性别

 birthday 出生日期

 address 用户住址

 telephoneNumber 用户电话

2、

problemLib
+pid +problemName +ratio +accepted +subnited +context +inputCase +outputCase +timeLimit +memoryLimit

属性： pid 问题 id

ProblemName 问题名称

ratio 该题正确率（双精度）

accepted 该题当前是否被选中

submitted 该题是否被提交

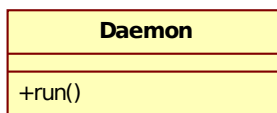
context 该题题干内容

timeLimit 时间限制

memoryLimit 运行内存限制

控制类：

1、



属性：

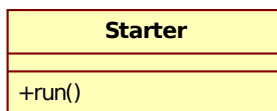
操作：run() 运行守护进程

2、



操作：run() 运行看门狗进程

3、



操作：run() 运行计时器

4、

Compiler
+compiler +compilerOption +judgeStatus +ErrorInfo
+Compiler() +setCompilerOption() +getStatus() +getErrorInfo() +execute()

属性： compiler 编译器

compilerOption 编译模式选择

judgeStatus 判定状态

ErrorInfo 出错信息

操作： compiler()

setCompilerOption() 设定编译模式

getStatus() 获取当前编译状态

getErrorInfo() 获取出错信息

execute() 执行

5、

Executer
+timeLimit +memoryLimit +exePath +inputFile +outputFilePath +testCasePath +*execProcess +judgeStatus
+Executer() +setInputFile() +setOutputFile() +setExePath() +setTimeLimit() +setMemoryLimit() +setTestCasePath() +execute() +checkRuningStatus() +test() +checkPresentation() +getStatus()

属性： timeLimit 运行时间上限

memoryLimit 运行空间上限

exePath 执行文件的路径

inputFile 输入文件路径

outputFilePath 输出文件路径

testCasePath 测试用例路径

*execProcess 执行程序指针

judgeStatus 测试状态

操作： execute() 执行

setInputFile() 设置输入文件路径

setOutputFile() 设置输出文件路径

setExePath() 设定可执行文件的路径

setTimeLimit() 设置运行时间上限

setMemoryLimit() 设置运行可用内存上限

setTestCasePath() 设置测试用例路径

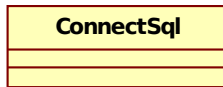
checkRunningStatus() 检查运行状态

test() 测试

checkPresentation() 检查结果表示

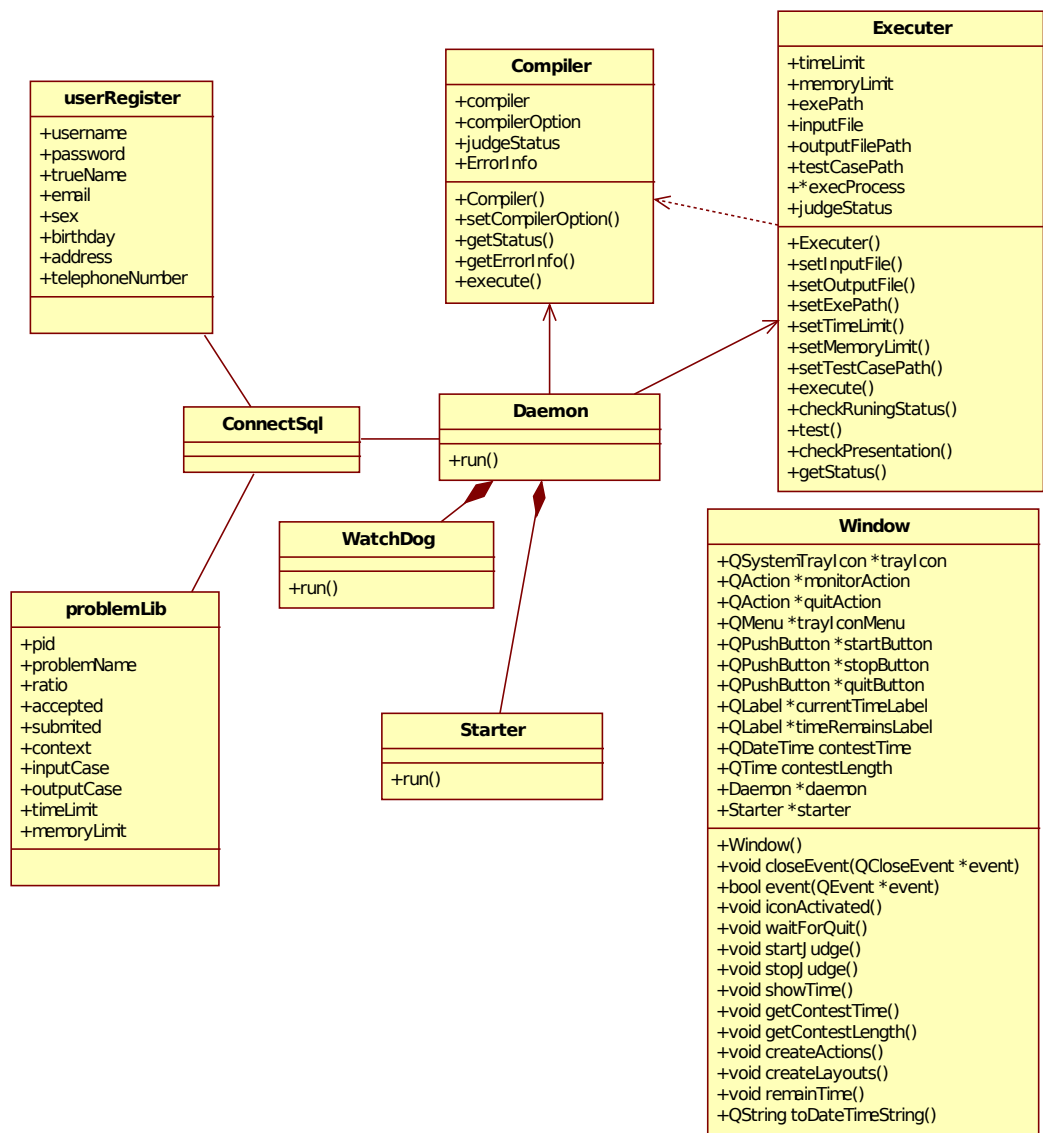
getStatus() 获取当前状态

数据库类：

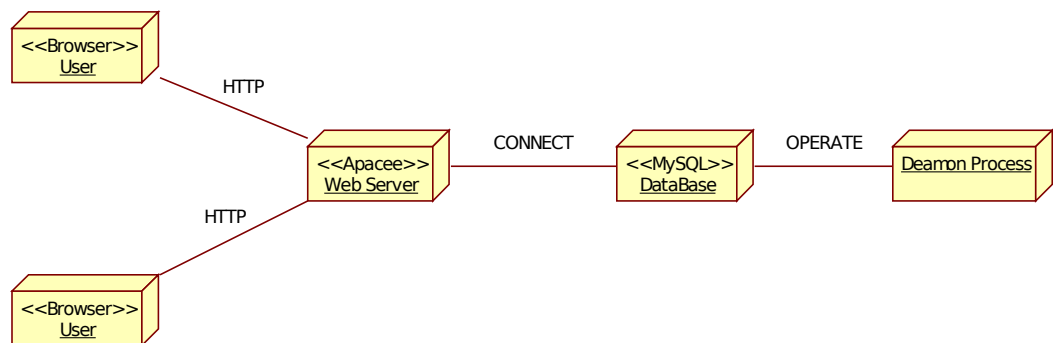


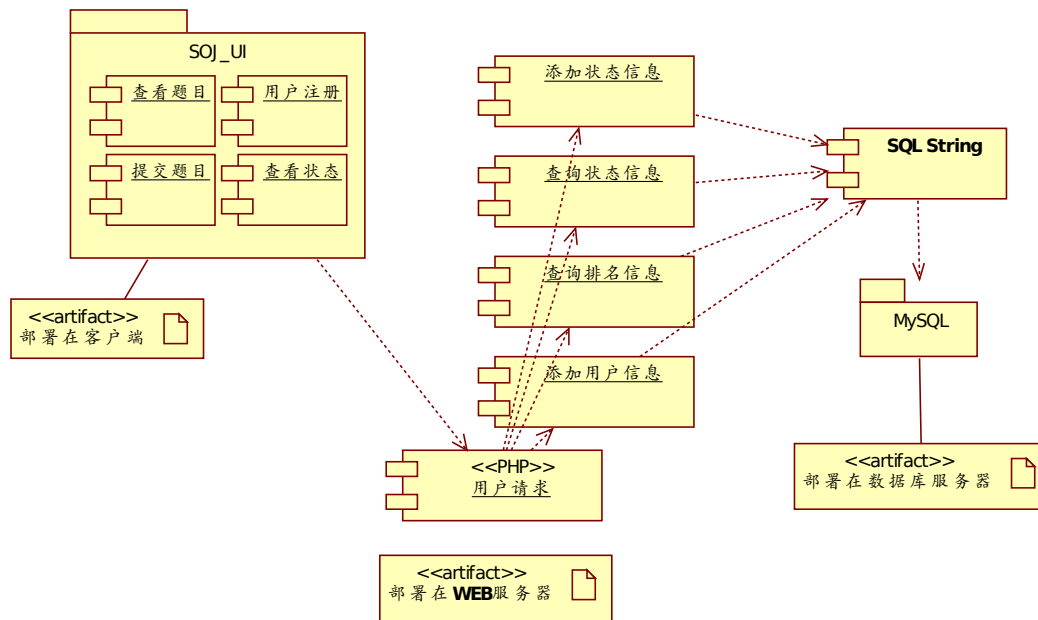
类间关系图

类关系总图：



7. 系统架构部署图





附录：

术语定义（遵循国际通用的标准）：

- **Waiting:** The judge is so busy that it can't judge your submit at the moment, usually you just need to wait a minute and your submit will be judged.
- **Judging:** The judge is judging your problem now.
- **Accepted:** OK! Your program is correct!
- **Presentation Error:** Your output format is not exactly the same as the judge's output, although your answer to the problem is correct. Check your output for spaces, blank lines, etc. against the problem output specification.
- **Wrong Answer:** Correct solution not reached for the inputs. The inputs and outputs that we use to test the programs are not public (it is recommendable to get accustomed to a true contest dynamic).
- **Runtime Error:** Your program failed during the execution (illegal file access, stack overflow, pointer reference out of range, floating point exception, divided by zero, etc.).
- **Time Limit Exceeded:** Your program tried to run during too much time.
- **Memory Limit Exceeded:** Your program tried to use more memory than the judge default settings.

- **Compile Error:** The compiler could not compile your program. Of course, warning messages are not error messages. Click the link at the judge reply to see the actual error message.
- **No Such Problem:** Either you have submitted a wrong problem ID or the problem is unavailable.
- **Out Of Contest Time:** This message can only appear during a contest, if a program is submitted out of contest time.
- **Restricted Function:** Your program tried to call restricted functions. For example, maybe you have tried to open a file which is forbidden.