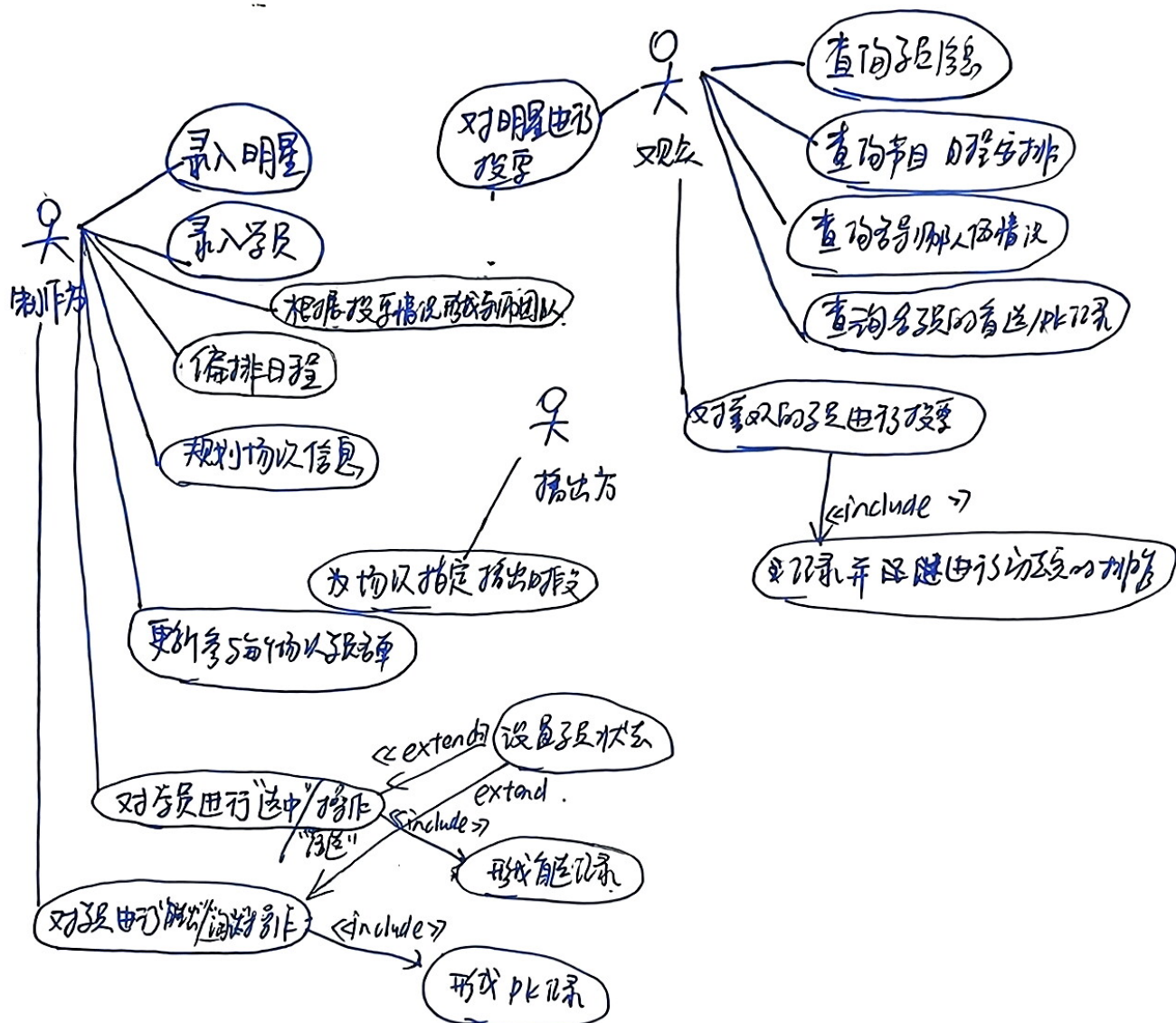
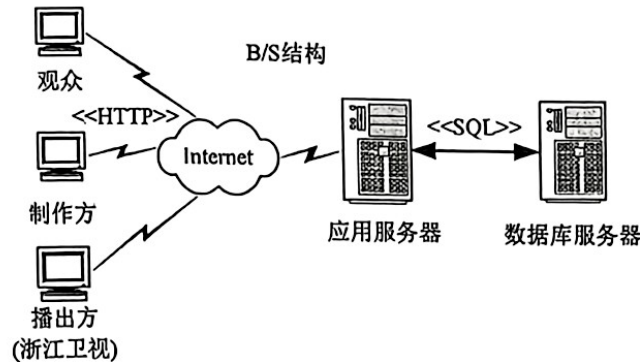


1. (10 分) 针对上述功能需求，识别外部参与者、用例和通讯关联，给出用例模型。用例名称请用中文表述。



2. (13 分) 该系统设计之初使用了下图所示的 B/S 架构，各功能模块均由 PHP 语言书写，每个 PHP 程序由用户界面代码、业务逻辑代码、与后台数据库交互的代码构成，都部署在应用服务器上，观众等通过浏览器请求 PHP 页面以访问系统。



随着节目进展，系统需求出现了以下变化：

(1) 新浪网与制作方合作，当中国好声音系统中的比赛相关数据发生更新时，新浪网可自动获取到变化的信息而无需人工参与；

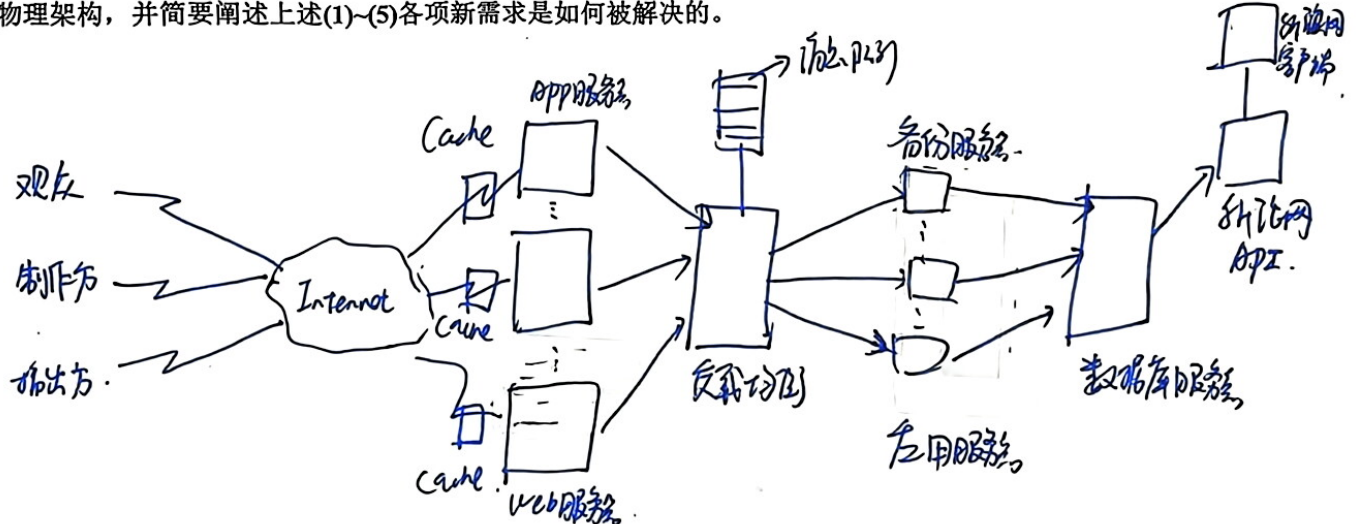
(2) 观众希望在看电视时通过智能手机来访问网站，并可通过手机短信与系统互动（例如为支持的学员或导师投票），而不是只能傻乎乎待在电脑屏幕前；

(3) 制作方工作人员认为通过浏览器访问系统的效率太低了，跟不上舞台上快速发生的各种事件，造成系统中的数据与现实中有某些延迟；

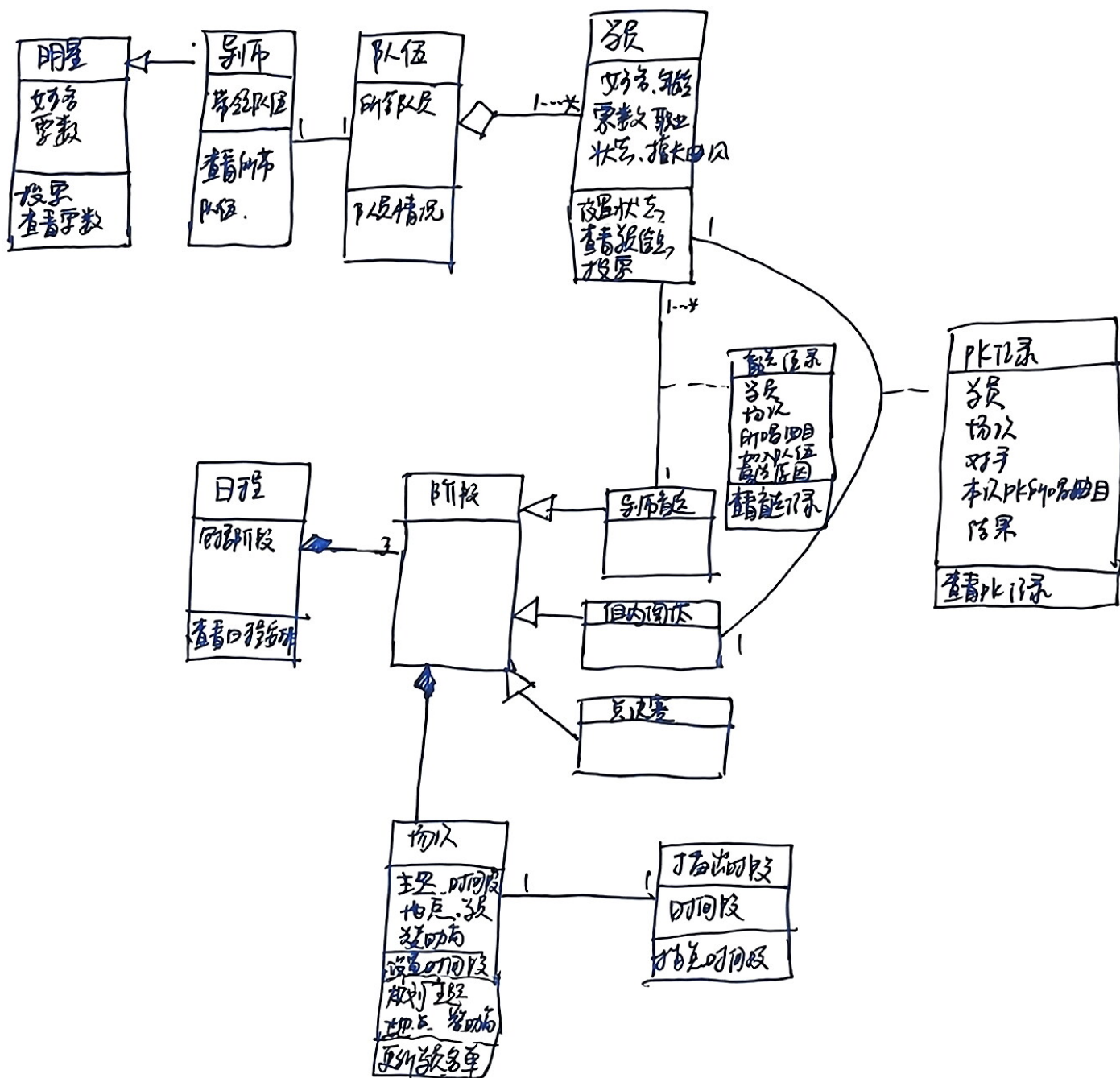
(4) 总决赛期间，全国观众的热情将异常高涨，会在节目播出的 2 个小时内高频度的集中访问系统，这可能造成系统宕机的严重后果；

(5) 比赛规则容易发生变化，开发该系统的软件公司希望以尽量少的代码更改来适应变化。

针对上述情况，考虑采用哪些设计决策对上图架构进行改进，使用 UML 部署模型描述系统新的物理架构，并简要阐述上述(1)~(5)各项新需求是如何被解决的。



3. (13 分) 识别该系统中所用到的实体类(概念类), 分析它们之间的静态结构关系, 建立领域模型, 以类图的形式展现出来。模型中需包含各类的关键属性和方法, 并详细标识出类间关联关系的必需信息。类的名称、属性、方法名称均用中文表述。



二 测试题 (22 分)

1. (10 分) 观众可通过手机向系统发送短消息，表示自己对特定学员或导师的支持。系统中有一个短消息接收模块，其参数为客户发来的短消息，格式遵循以下规则：

- 分为三部分，依次为：对象、编号、文字，三部分之间直接相连；
- “对象”部分的长度为 1，取值范围为 {X, D}，分别表示学员、导师；
- “编号”部分：若对象部分取值为 X，则该部分由三位数字构成，取值范围为 [001, 120]，表示学员编号；若对象部分取值为 D，则该部分取 {NY, LH, YCQ, YK} 之一，表示导师身份；
- 文字部分无任何限定，可以为空，也可以任意长度。

对符合上述规则的短消息，模块返回结果为“合法”，否则返回“非法”。例如，你强烈支持刘欢的短消息为“DLH 刘欢你是最棒的导师！”，你简单支持学员金志文(002)的短消息为“X002”，均属合法的短消息。

利用黑盒测试的相关知识，完成以下题目：

- (5 分) 使用等价类划分方法，以表格形式给出所有的有效等价类和无效等价类；
- (5 分) 设计一组黑盒测试用例（包括输入数据、覆盖的等价类、期望的输出结果）。

c)

有效等价类

无效等价类

- “对象”部分为 X，编号为 001-120，后跟任意文字
- “对象”部分为 D，编号为 NY, LH, YCQ, YK, 后跟任意文字
- D, ---LH, ---YCQ, ---YK, 后跟任意文字
- D, ---YCQ, ---YK, 后跟任意文字
- D, ---YK, 后跟任意文字

- 对象部分缺失
- 对象部分为非 D, X
- 对象 X，编号不是三位数字
- 对象 X，编号不在 001-120
- 对象 D，编号不是 NY, LH, YCQ, YK
- 对象 D，编号缺失

(17).	X 001 X	合法 ①
	D NY X	合法 ②
	D LH X	合法 ③
	D LCR X	合法 ④
	D YK X	合法 ⑤
	LH X	非法 ⑥
	NY D	非法 ⑦
	X X X X	非法 ⑧
	A 000 X	非法 ⑨
	X 123	非法 ⑩
	D NX	非法 ⑪

2. (12 分) “中国好声音”系统中有一个函数用于从观众发来的短消息序列中找出受支持最多的学员, 其伪代码如下所示。输入一个整型数组, 其值为观众支持的学员编号, 输出一个字符串, 其值为“学员编号@被支持次数”的格式。

```

1  String getChampionFromSMS (int[] sms) {
2      sms.sort();    //按升序对 sms 中字符串排序

3      int count = 0, value = sms[0]-1;
4      int count_new = 0, value_new = sms[0]-1;

5      int i = 0;
6      while(i < sms.length()) {    //sms.length() 返回 sms 数组的长度
7          if(sms[i] == value_new)
8              count_new ++;
9          if(sms[i] != value_new || i == sms.length-1) {
10             if(count_new > count) {
11                 count = count_new;
12                 value = value_new;
13             }
14             value_new = sms[i];
15             count_new = 1;
16         }
17         i++;
18     }

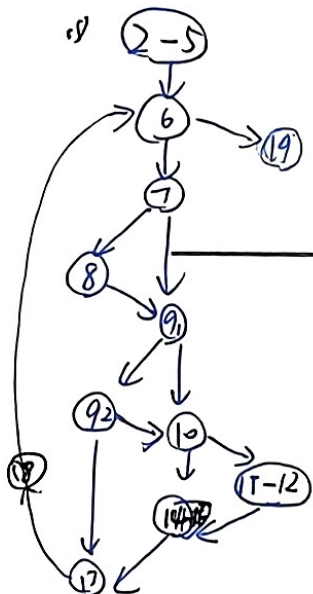
19     return value + "@" + count;
20 }

```

基于上述源代码, 利用白盒测试方法, 完成以下题目:

- (1) (4 分) 绘制该程序的控制流图;
- (2) (2 分) 计算其环形复杂度 (圈复杂度);
- (3) (3 分) 列出所有的基本路径;
- (4) (3 分) 针对每条基本路径来设计测试用例, 测试用例中至少应包含输入数据和期望结果。

答题中请务必使用程序中的代码行号 (1~20)。若需要将某一行拆分为多行, 请以 $n1$ 、 $n2$ 、 $n3$ 的形式表示, n 为需要拆分的行号。



(2) 6个。 $15 - 11 + 2 = 6$

(3) 11. $2-5 \rightarrow 6 \rightarrow 19$

12. $2-5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 14-16 \rightarrow 17 \rightarrow 18 \rightarrow 6 \rightarrow 19$

13. $2-5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 14-16 \rightarrow 17 \rightarrow 18 \rightarrow 6 \rightarrow 19$

第 9 页 (共 12 页)

14. $2-5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 14-16 \rightarrow 17 \rightarrow 18 \rightarrow 6 \rightarrow 19$

15. $2-5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11-12 \rightarrow 14-16 \rightarrow 17 \rightarrow 18 \rightarrow 6 \rightarrow 19$

16. $2-5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 17 \rightarrow 18 \rightarrow 6 \rightarrow 19$