

【XXX 系统】

——XXX 子系统

需求说明书

组长:

组员:

日期:

版本:

目录

——XXX 子系统	1
1. 文档介绍	3
1.1. 编写目的	3
1.2. 文档范围	3
1.3. 读者对象	3
1.4. 术语与缩写解释	3
1.5. 参考资料	3
2. 项目介绍	4
2.1. 项目说明	4
2.2. 项目背景	4
2.3. 项目目标	4
2.4. 项目用户	4
3. 需求说明	4
3.1. 整体需求	4
3.2. 功能需求	5
3.3. 性能需求	5
3.4. 安全性需求	5
3.5. 可维护性需求	6
4. 用户场景	6
4.1. 用例	6
4.2. 用例图	8
5. 数据流图	8
5.1 顶层图	8
5.2 0 层图	9
5.3 1 层图	9
6. 数据词典	10
6.1 数据流	10
6.2 数据项	11
6.3 数据存储	11
6.4 处理逻辑	11
7. 状态图	12
8. CRC 卡	12
9. 运行环境规定	13
9.1 服务器配置要求	13
9.2 客户端配置要求	13
9 软件依赖	13

1. 文档介绍

1.1. 编写目的

本文档描述软件产品需求规格说明书（SRS）的目的是：

- 定义软件总体要求，作为用户和软件开发人员之间相互了解的基础；
- 提供性能要求、初步设计和用户影响的信息，作为软件人员进行软件结构设计和编码的基础；
- 作为软件总体测试的依据。

1.2. 文档范围

XXX 系统需求规格说明书主要包含了该系统整体需求及功能性需求的详细介绍，XXX，XXX……

1.3. 读者对象

编写详细设计人员及程序开发人员、XXXX、YYYY……

1.4. 术语与缩写解释

（对文档编写中可能遇到的专业词汇或陌生词汇进行说明）【以下斜体红色文字不应该出现在最终的文档里】

缩写、术语及符号	解释
SOA 架构	面向服务的体系结构。
元数据 Metadata	描述数据的内容、质量、状况和其他有关特征的数据。
数据中心 Data Center	以各类数据为核心，依托成熟的存储、数据库、GIS、网络等技术，按照统一标准，建立的具有信息管理、分析、查询、统计及服务的一体化数据管理体系。
数据管理 Data Management	利用数据库、数据仓库、元数据和网络等技术，建立分布式、集中式或集中加分布式数据管理系统，开展数据接收、组织存储、运行维护、更新、共享交换等工作，实现对数据资源的有效组织和应用。

1.5. 参考资料

（包含文档编写过程中参考的材料以及版本）

序号	文档名称	文档编号	版本	发布日期
			

2. 项目介绍

2.1. 项目说明

介绍产品的名称、任务提出者、开发者、用户群

项目名称：XXX 系统。

任务提出者：XXX。

开发者：XXX。

用户群：XXX

2.2. 项目背景

XXX

2.3. 项目目标

XXX

2.4. 项目用户

XXX

3. 需求说明

3.1. 整体需求

XXX

3.2. 功能需求

（主要根据业务和展示功能划分，分为A 模块、B 模块、……。附带文字说明。每个模块需要根据实验大纲进行需求分析，明确模块涉及到的所有功能）

（注意这里的A、B、C 模块不是每个小组负责的顶层模块，而是进一步划分之后的中间层模块）

模块名称		A 模块		
模块简介		相关功能描述 ABCD		
模块功能列表				
序号	一级功能		二级功能	
	功能名称	功能编号	功能名称	功能编号
1		xxdt001		
2		xxdt002		
3		xxdt003		

3.3. 性能需求

（性能需求分析也要做，涉及到相关模块的访问频率的分析，对应模块要承载多少访问量和并发量。例如管理员模块需要对职责范围内的所有数据进行管理，可能对并发量要求较低，但对数据处理的效率要求较高[信息展示、检索、数据导入导出等]）

(1) 系统配置

系统要具有良好的反应速度，课题要求在良好的网络情况下，本系统应该具有如下时间特性要求：

单个用户在线时：1. Web 响应用户动作时间小于 1 秒。2. 信息搜索操作响应用户动作时间小于 2 秒。500 个用户同时在线时：1. Web 响应用户动作时间小于 2 秒。2. 信息搜索操作响应用户动作时间小于 5 秒。

(2) 访问容量

该系统至少在同一时间内支持 200 个用户并发访问。

(3) 服务器配置最低要求

CPU 4 核 2.6G，内存 8.0G，硬盘 7200 转。

(4) 可用性

该系统应实现多 Web 浏览器支持：在大多数流行的 Web 浏览器中正确显示和执行，包括 Firefox、Chrome、Edge 等。

3.4. 安全性需求

（安全性需求包括隐私数据的安全性，已经应对爬虫、DDOS、SQL 注入等常见攻击手段的处理）

用于身份验证的用户名和密码应防止未经授权的用户访问系统。应构建访问控制以防止合法用户非法使用系统资源。某些敏感数据（如用户名，密码）在交换时应加密。密码在存储之前应加密。

在用户登录期间，应该防止 SQL 注入，密码强制破解和伪造会话入侵。

1. 完整性：防止非法用户对数据进行无意或恶意的修改、插入、删除，防止数据丢失。

2. 约束性：为数据库加上一定的约束，对关键性操作如删除、修改进行限制，并对用户进行警示。不同身份所拥有的权限不同，只可以进行自己权限内的操作。

3. 账户信息安全性：着重账户信息安全性设计，做到外界人员无法入侵到系统本身。内部人员操作需要留下操作痕迹，使用权管理层可以定期或不定期地维护系统

3.5. 可维护性需求

(对代码维护性提出要求)

作为一个成熟的系统，在开发初期就应该充分考虑系统的可维护性。对此，我们提出以下几点要求：

(1) 高内聚、低耦合的系统模块划分。开发者需要充分考虑模块内部结构的紧密型及模块间联系的独立性。

(2) 完备、清晰、可读的文档。文档是影响软件可维护性的一个决定因素，一个好的文档应具有简明性和书写风格的一致性，从而提高系统的可读性和可修改性。设计系统时应准备好各类相关文档，方便操作人员的对功能的快速查阅及维护人员的对架构的系统掌握。交付时应文档齐全，说明详尽，且文档描述符合相关标准。

(3) 良好的编程风格：程序内部应有详细的注释和统一的编程格式，结构清晰、注释明确，使调试、测试人员能快速定位各种错误。对编程风格的具体要求如下：不使用令人捉摸不定或含糊不清的代码；使用有意义的变量名和过程名；适当的、格式正确的注释；使用模块化、结构化的设计方法；具有正确、一致和完整的文档。

(4) 严谨的单元测试：对核心模块应编写单元测试，在交互时保证各子模块和系统整体的正常运作。对可测试性的要求如下：具有模块化和良好的结构；具有可理解性、可靠性；能显示任意的中间结果；以清楚的描述方式说明系统的输出，根据要求显示所有的输入。

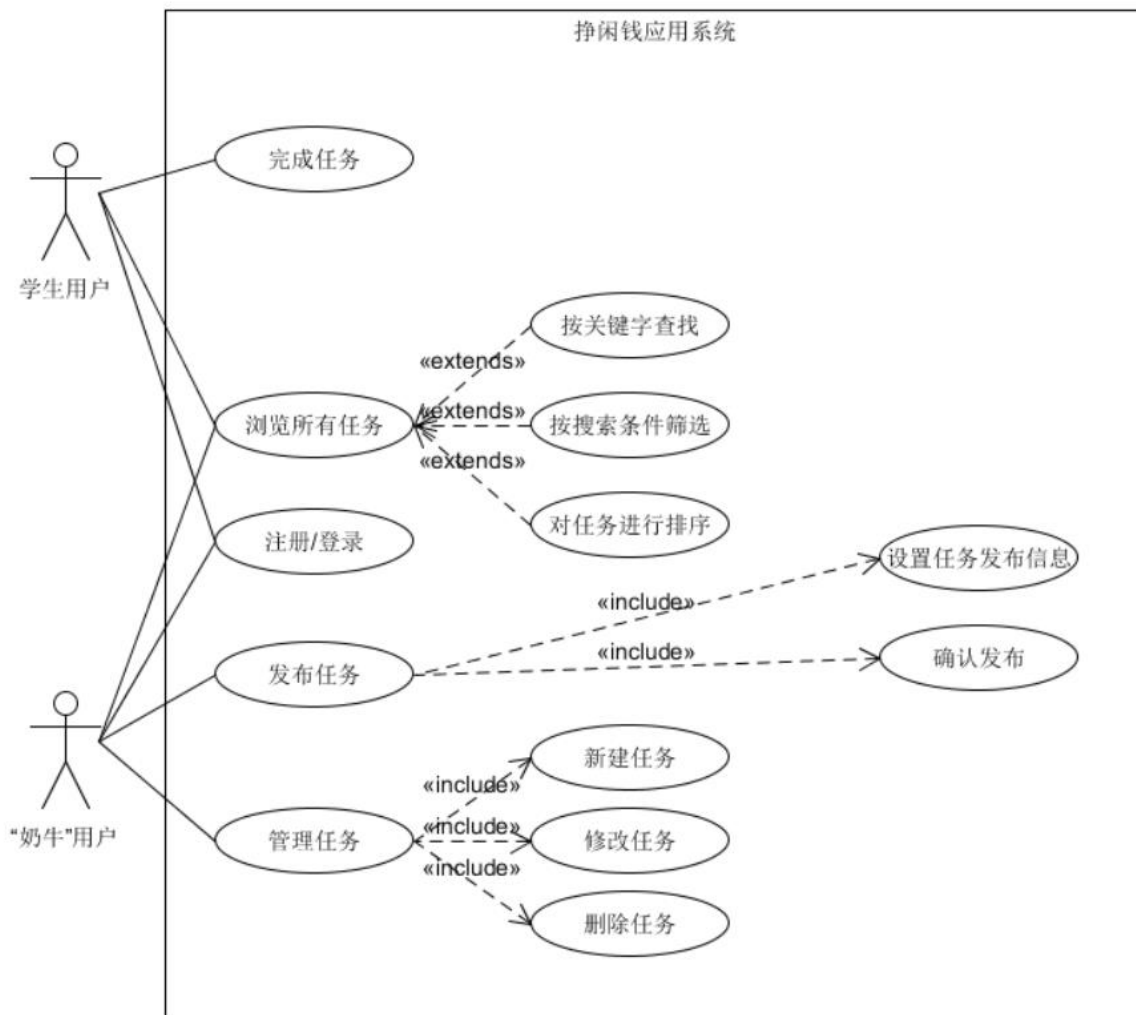
4. 用户场景

4.1. 用例

用例	添加教室信息
主要参与者	管理员
目标	添加一个新的可用教室资源
前提条件	管理员已登录
触发器	管理员决定添加教室
工作流程	<div>1. 管理员点击进入添加教室信息功能的按钮</div> <div>2. 系统显示添加教室信息的界面</div> <div>3. 管理员通过选择框选择新增教室的校区、教学楼、楼层、类型，输入门牌号，并核对无误</div> <div>4. 管理员点击确认添加按钮</div> <div>5. 系统通过弹窗再次提示确认添加教室的信息</div> <div>6. 管理员再次点击确认添加</div> <div>7. 系统为此新教室信息生成一个独有的编号，并将其插入到教室数据库</div> <div>8. 系统根据数据库的操作结果向管理员反馈添加成功/失败</div> <div>回到添加教室信息界面，管理员可以选择继续添加教室或退出此功能</div>
异常	<div>1. 管理员点击按钮试图进入添加教室信息的页面时，系统没有反应</div> <div>2. 管理员无法提交添加教室的信息</div> <div>3. 点击确认后，显示的操作状态和实际不一致，比如显示为操作成功但实际没有添加到数据库</div> <div>4. 管理员输入的教室已经存在</div> <div>9. 页面加载失败</div>
优先级	必须实现
何时可用	首次增量
使用频率	一般
使用方式	通过浏览器
次要参与者	无

4.2. 用例图

(以下为一个用例图示例，绘制用例图注意功能层次关系，并且和前文用例一致)



5. 数据流图

(以下为一个数据流图示例，以竞赛管理系统为例子，边是数据流，注意和流程图区分)

5.1 顶层图

顶层图主要描述了该系统的主要的输入数据，输出数据，相关实体。采用的 IPO 模型。

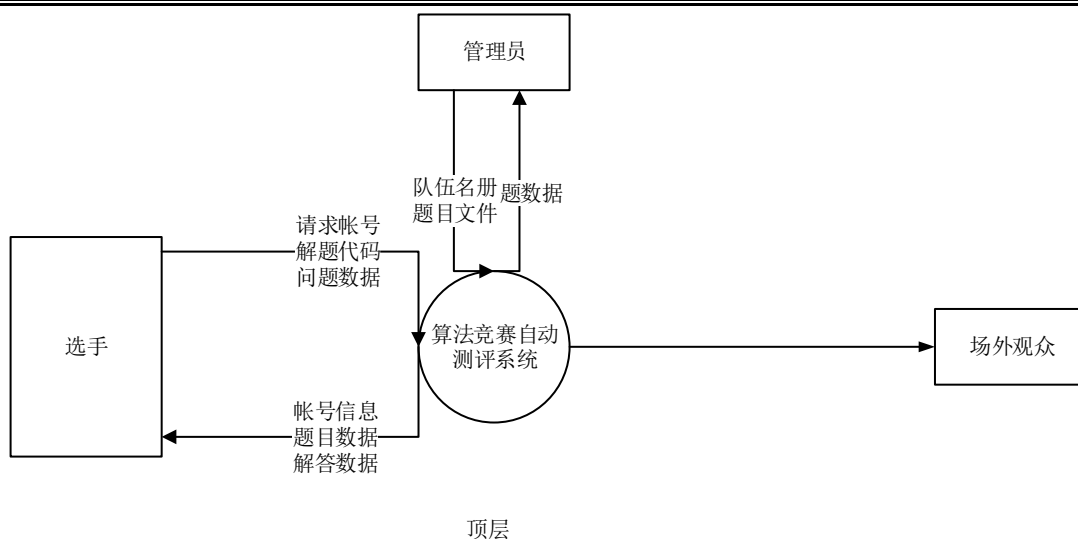
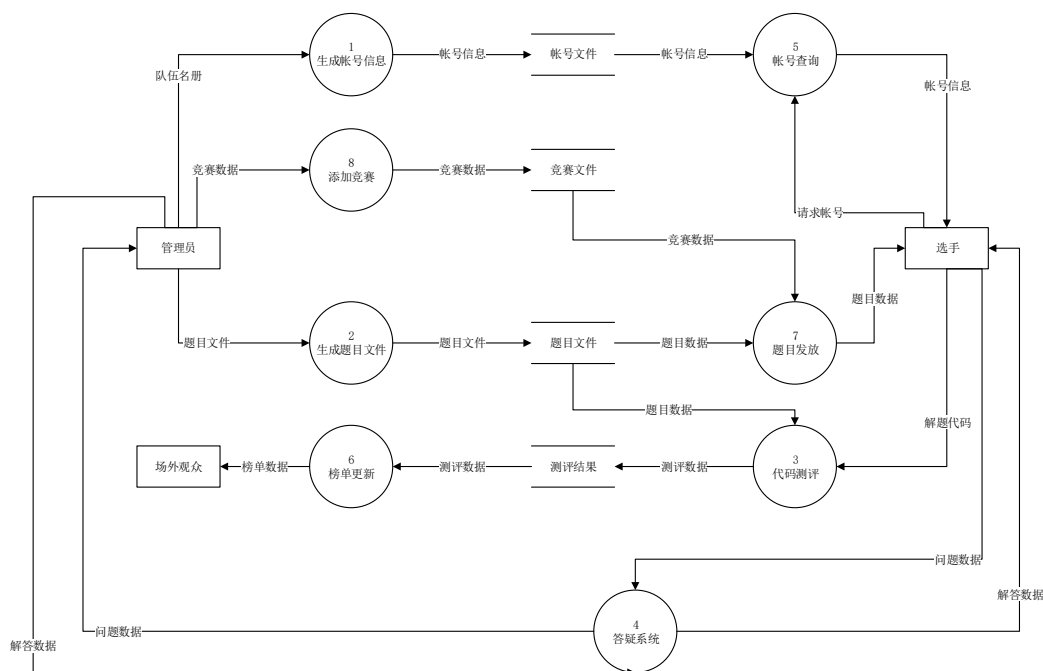


Figure 3: 顶层图

5.2 0 层图

0 层图对于系统进行了模块的划分，分为了 7 个部分：生成帐号信息，添加竞赛，生成题目文件，榜单更新，帐号查询，题目发放，代码测评，答疑系统。其中在该层图中也显示了相关的存储数据。



5.3 1 层图

1 层图中对 0 层图中划分的 2 个模块分别进行了详细地描述。

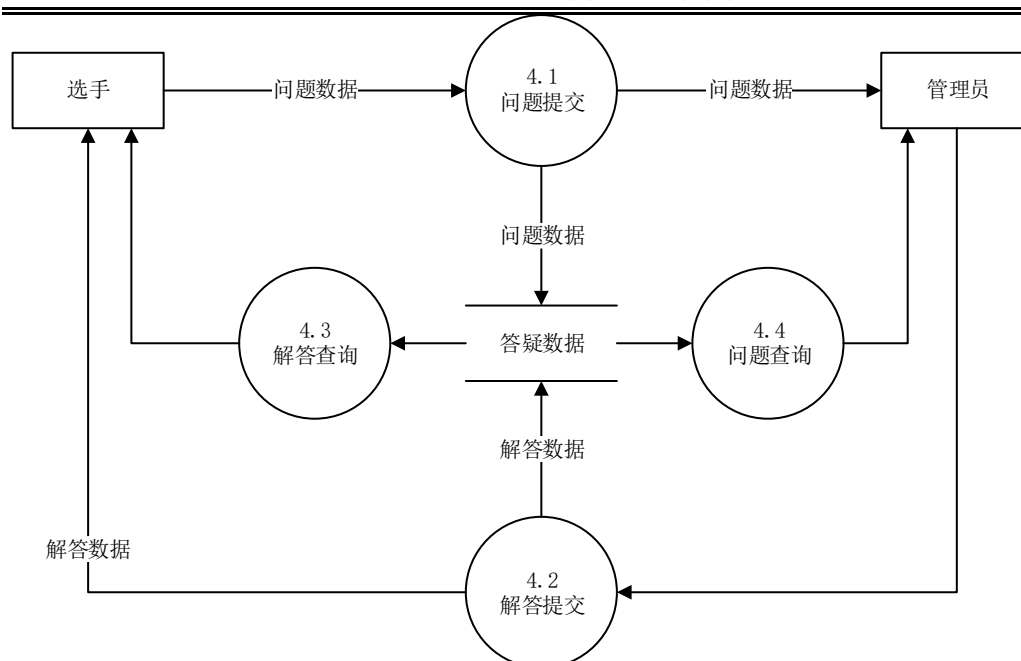
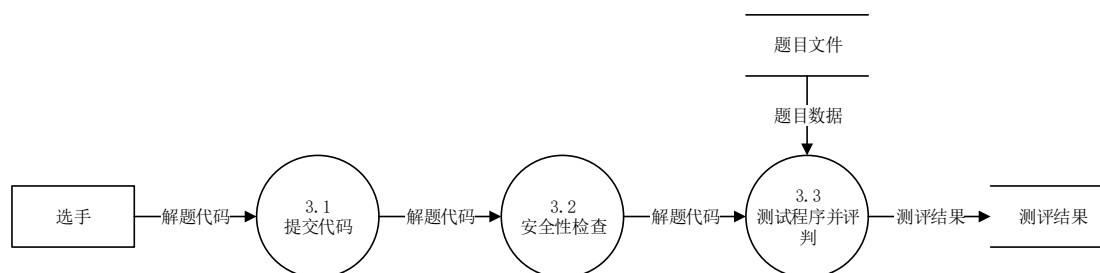


Figure 5: 答疑系统 1 层图



6. 数据词典

(以下为数据词典示例，以竞赛管理系统为例子，数据词典与数据流图关键部分一致)

6.1 数据流

名称	队伍名册
描述	管理员将队伍的信息录入系统
组成	队伍编号+队伍名称
来源	管理员
终点	生成帐号信息 1

数据流名称	帐号信息
描述	系统将生成的帐号信息保存到帐号文件中
组成	队伍编号+队伍帐号+帐号密码
来源	生成帐号信息 1
终点	帐号文件

6.2 数据项

数据项名称	队伍编号
描述	参加比赛的队伍唯一编号，不重复
相关文件或记录	队伍名册，账号文件
相关处理	处理名称 DFD 参考号 结构图号
	生成账号信息 1
	请求账号 5
数据特征	整数
取值范围	>0, <=9999

数据项名称	队伍名称
描述	参赛队伍的名称
相关文件或记录	账号文件
相关处理	处理名称 DFD 参考号 结构图号
	生成账号信息 1
	账号查阅 5
数据特征	字符串
取值范围	长度大于 0，小于 256

6.3 数据存储

文件或数据库名	账号文件
描述	含有比赛队伍的信息和账号密码信息
组成	队伍编号+队伍名称+账号名称+账号密码
相关处理	处理名称 DFD 参考号 结构图号
	生成账号信息 1
	账号查询 5

文件或数据库名	题目文件
描述	记录了一场比赛题目的所有信息
组成	题目编号+题目描述+测试数据
相关处理	处理名称 DFD 参考号 结构图号
	生成题目文件 2
	代码测评 3
	题目发放 7

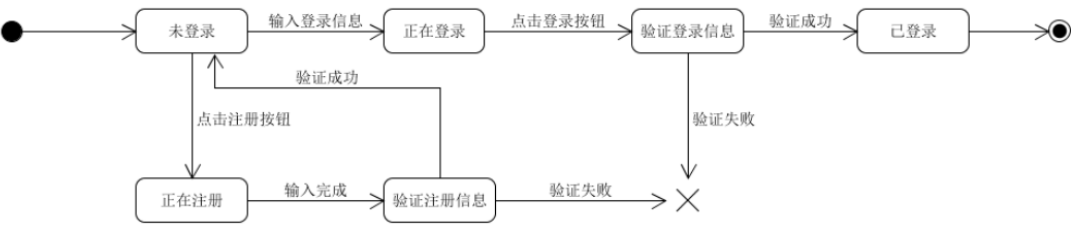
6.4 处理逻辑

处理名	生成账号信息
处理表示	DFD1
描述	根据管理员录入的比赛队伍信息，自动生成账号和密码
输入数据流	队伍名称

输出数据流	账号信息
存取的数据库	账号文件
处理逻辑	<pre>while(输入比赛队伍未完) do 读入一个比赛队伍信息 检索账户文件里是否有相关记录 IF 记录不存在 THEN 生成新的账户的密码，添加到账户文件</pre>

7. 状态图

用户登录注册过程中的状态转移过程
(以下为状态图示例，注意状态图针对某一个对象的状态，和流程图区分开来)



8. CRC 卡

(以下为CRC 卡示例)

Class: login_administrator	
Description: 管理员登录类	
Responsibility:	Collaborator:
获得账号、密码	
验证身份	account_storage
账户不存在，提示“账号或密码错误”	account_storage
密码不正确，清空密码框，提示“账号或密码错误” 给出密码找回、密码重置选项	account_storage
检查账号是否锁定	account_storage

账户锁定，提示“账户被锁定”	database_update
验证成功，进入管理员的主界面	account_storage
验证失败超过一定次数锁定账户	account_storage

9. 运行环境规定

9.1 服务器配置要求

9.2 客户端配置要求

CPU:不小于 2.0GHz

内存：不小于 2.0GB

9 软件依赖

操作系统：Windows Vista/7/8/8.1/10, Mac OS，Linux

数据库平台：MySQL Web

服务器：Nginx

MySQL 管理软件：PHPMyAdmin 或 MySQL WorkBench 等

开发工具：能支持网页开发的工具均可（如 PyCharm）