

软件需求实习报告

[CUGCTF评测平台需求规格说明书]

学生姓名： 王志辰

学 号： 20161001059

班 号： 111161

指导教师： 周顺平

成 绩：

中国地质大学信息工程学院

2018.11

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **章节** | **修订说明** | **当前版本** | **修订日期** | **修订前版本号** | **修订后版本号** | **修订人** | **审批人** |
| 1 | 起草 |  | 1.0 | 2018-10 | 1.0 | 1.0 | 王志辰 |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

**文档控制页**

目录

[1 文档概述 4](#_Toc462693399)

[1.1 编写的目的 4](#_Toc462693400)

[1.2 项目背景 4](#_Toc462693401)

[1.3 术语和定义 4](#_Toc462693402)

[1.4 参考资料 4](#_Toc462693403)

[2 任务概述 4](#_Toc462693404)

[2.1 业务需求 4](#_Toc462693405)

[2.2 Stakeholder利益分析 4](#_Toc462693406)

[2.3 用户特点分析 4](#_Toc462693407)

[2.4 相关事实与假定 4](#_Toc462693408)

[3 需求概述 4](#_Toc462693409)

[3.1 系统概述 4](#_Toc462693410)

[3.2 主题域 1 5](#_Toc462693411)

[3.2.1 概述 5](#_Toc462693412)

[3.2.2 业务事件 5](#_Toc462693413)

[3.2.3 报表 5](#_Toc462693414)

[4 具体需求 5](#_Toc462693415)

[4.1 主题域 1 5](#_Toc462693416)

[4.1.1 用例模型 5](#_Toc462693417)

[4.1.2 领域模型 7](#_Toc462693418)

[5 补充规约 8](#_Toc462693419)

[5.1 设计约束 8](#_Toc462693420)

[5.1.1 技术选择的限制条件 8](#_Toc462693421)

[5.1.2 运行环境 8](#_Toc462693422)

[5.1.3 预期的使用环境 8](#_Toc462693423)

[5.2 质量属性 8](#_Toc462693424)

[5.2.1 安全性要求 9](#_Toc462693425)

[5.2.2 可靠性要求 9](#_Toc462693426)

[5.2.3 易用性要求 9](#_Toc462693427)

[5.2.4 性能要求 9](#_Toc462693428)

[5.2.5 可维护性要求 10](#_Toc462693429)

[5.2.6 可移植性要求 10](#_Toc462693430)

[5.2.7 其他质量属性要求 10](#_Toc462693431)

[5.3 其他需求 10](#_Toc462693432)

[5.3.1 培训需求 10](#_Toc462693433)

[5.3.2 后勤需求 10](#_Toc462693434)

[5.3.3 包装需求 11](#_Toc462693435)

# 文档概述

## 编写的目的

本文档用于指导开发者为该项目开发软件的过程，通过规范软件项目开发过程而程达到提高软件质量，降低维护成本的目的。开发者应根据本指南进行软件开发和编制软件开发文档。在进行具体软件开发时，开发者可根据实际情况采编写，但必须提供双方约定的文档，文档中约定的内容必须描述清楚。

## 项目背景

CTF（Capture The Flag，夺旗赛）起源于 1996 年 DEFCON 全球黑客大会，是网络安全爱好者之间的竞技游戏。CTFF起源于1996年DEFCON全球黑客大会，以代替之前黑客们通过互相发起真实攻击进行技术比拼的方式。发展至今，已经成为全球范围网络安全圈流行的竞赛形式。

CTF是一种流行的信息安全竞赛形式，其英文名可直译为“夺得Flag”，也可意译为“夺旗赛”。其大致流程是，参赛团队之间通过进行攻防对抗、程序分析等形式，率先从主办方给出的比赛环境中得到一串具有一定格式的字符串或其他内容，并将其提交给主办方，从而夺得分数。为了方便称呼，我们把这样的内容称之为“Flag”。

CTF竞赛模式具体分为以下三类：

一、解题模式（Jeopardy）

在解题模式CTF赛制中，参赛队伍可以通过互联网或者现场网络参与，这种模式的CTF竞赛与ACM编程竞赛、信息学奥赛比较类似，以解决网络安全技术挑战题目的分值和时间来排名，通常用于在线选拔赛。题目主要包含逆向、漏洞挖掘与利用、Web渗透、密码、取证、隐写、安全编程等类别。

二、攻防模式（Attack-Defense）

在攻防模式CTF赛制中，参赛队伍在网络空间互相进行攻击和防守，挖掘网络服务漏洞并攻击对手服务来得分，修补自身服务漏洞进行防御来避免丢分。攻防模式CTF赛制可以实时通过得分反映出比赛情况，最终也以得分直接分出胜负，是一种竞争激烈，具有很强观赏性和高度透明性的网络安全赛制。在这种赛制中，不仅仅是比参赛队员的智力和技术，也比体力（因为比赛一般都会持续48小时及以上），同时也比团队之间的分工配合与合作。

三、混合模式（Mix）

结合了解题模式与攻防模式的CTF赛制，比如参赛队伍通过解题可以获取一些初始分数，然后通过攻防对抗进行得分增减的零和游戏，最终以得分高低分出胜负。采用混合模式CTF赛制的典型代表如iCTF国际CTF竞赛。

本平台适用于**解题模式**

和ACM一样，训练需要一个平台，用来检测选手们的答案是否正确。而在CTF中，往往环境复杂，需要合理的控制权限，进行环境分离，环境下发，以及防作弊机制。而目前的开源项目没有一个基于容器基础进行构造，并且没有防作弊机制，在安全上有很大隐患。

## 术语和定义

CTF- Capture The Flag

## 参考资料

<https://windforce17.github.io/blog/>

<https://baike.baidu.com/item/ctf/9548546>

<https://ctf-wiki.github.io/ctf-wiki/>

# 任务概述

本系统的主要目的是研发一款竞赛平台，实现在线答题。

根据项目计划，项目目标有这几个：

1. 用户合法性校验
2. 手机/微信绑定
3. 题目更新推送
4. 排行榜动态实时更新
5. 比赛计时器
6. 防作弊机制
7. 题目容器化
8. 分布式容器管理
9. 分布式一致性校验
10. Flag随机化处理

## 业务需求

请查看**任务概述**内容

## Stakeholder利益分析

|  |  |  |
| --- | --- | --- |
| 利益相关者 | 权力 | 利益 |
|  | 高低 | 具体表现 |
| 管理员 | 最高 | 设置题目，分数，比赛时间 |
| 参与者 | 最低 | 答题，启动题目容器，提交答案 |
| 其他 | 无 | 禁止登陆 |

## 用户特点分析

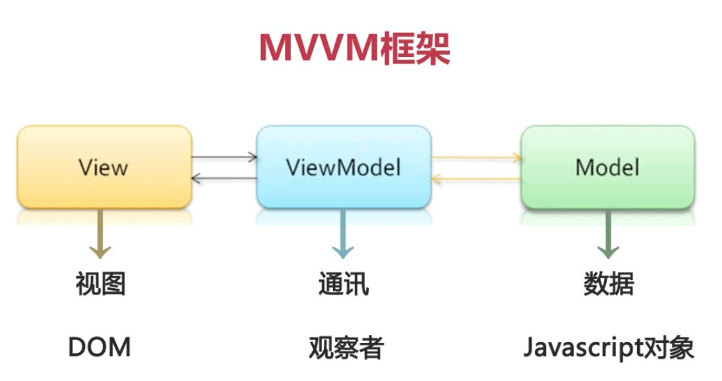
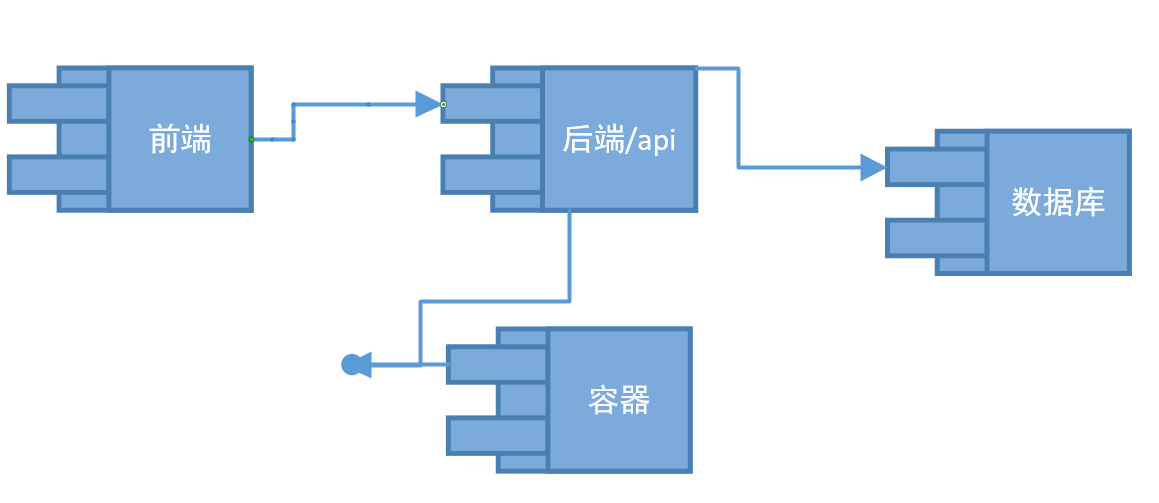
无

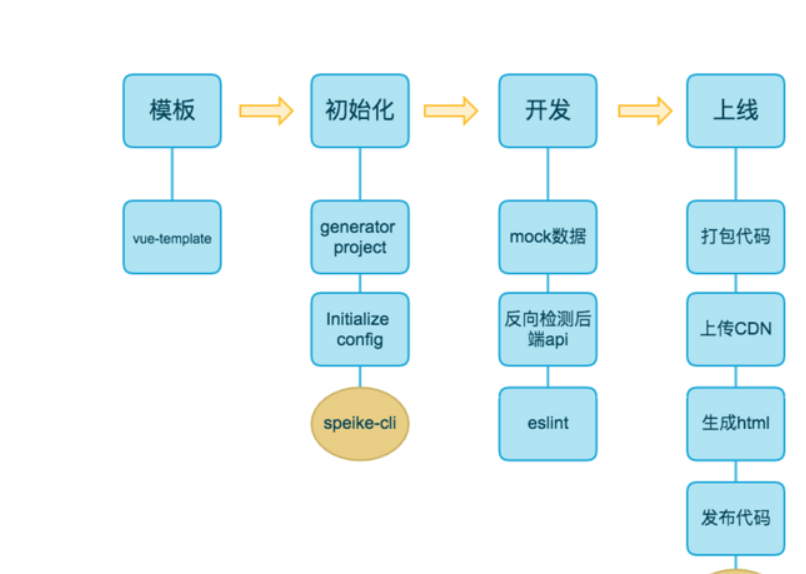
## 相关事实与假定

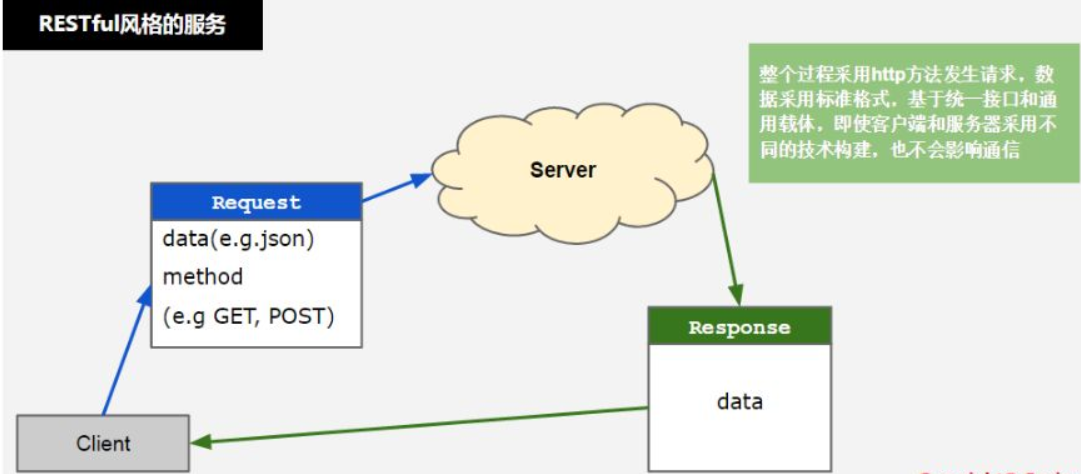
暂无

# 需求概述

## 系统概述

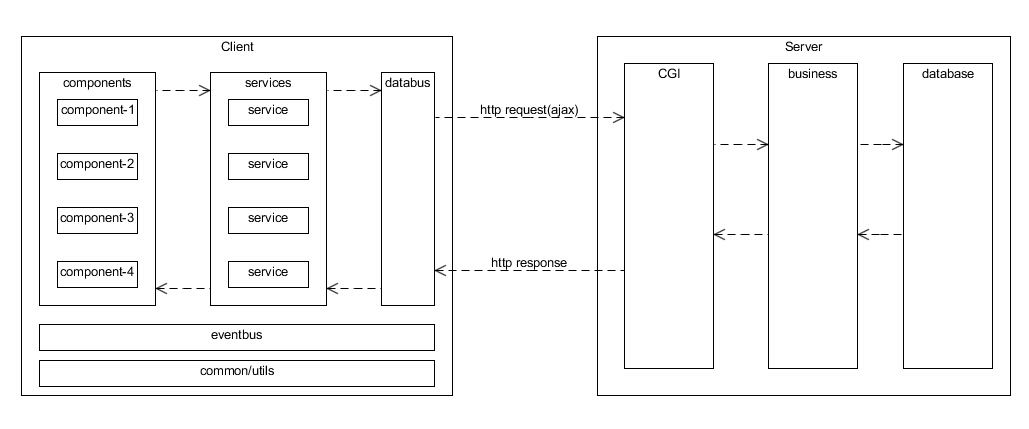






## 主题域

### 概述



### 业务事件

#### 登陆界面



#### 用户界面



# 具体需求

## 主题域 1

### 用例模型

#### UC\_B\_（B类）

##### 概述

|  |  |  |
| --- | --- | --- |
| 利益相关者 | 权力 | 利益 |
|  | 高低 | 具体表现 |
| 管理员 | 最高 | 设置题目，分数，比赛时间 |
| 参与者 | 最低 | 答题，启动题目容器，提交答案 |
| 其他 | 无 | 禁止登陆 |

##### 事件流描述

前置条件：登陆前



后置条件：登陆后





##### 相关需求与功能点

需求：杜绝安全问题

功能：v2版本支持微信/qq登陆。

#### UC\_I\_login（I类）

##### 使用者

名称：login

目的：登陆

##### 内容与格式

Json格式

POST /{api version}/login

psername string

password string

vaid\_code string

#### UC\_I\_logout（I类）

##### 使用者

名称:logout

目的:登出

##### 内容与格式

Json格式

POST /{api version}/logout

uuid string

#### UC\_I\_get\_problem\_list（I类）

##### 使用者

名称:get\_problem\_list

目的：得到题目列表

##### 内容与格式

Json格式

GET /{apt version}/get\_problem\_list

返回值：

{Problem\_name ,score} string[]

#### UC\_I\_get\_score（I类）

##### 使用者

名称：get\_score

目的：得到所有人的分数

##### 内容与格式

json

GET /{apt version}/get\_score

返回是

{username，score} string[]

#### UC\_I\_start\_problem（I类）

##### 使用者

名称：start\_problem

目的：开启一个题目实例

##### 内容与格式

Json格式

POST /{api version}/ start\_problem/:id

url 中id string

post数据：

uuid string

#### UC\_I\_ submit\_flag（I类）

##### 使用者

名称：submit\_flag

目的：提交题目答案

##### 内容与格式

Json格式

POST /{api version}/submit\_flag

problem\_id string

flag string

#### UC\_I\_ del\_problem（I类）

##### 使用者

名称：del——problem

用途：删除一个题目实例

##### 内容与格式

Json格式

POST /{api version}/del\_problem

uuid string

problem\_id strin

#### UC\_I\_change\_password（I类）

##### 使用者

名称：change\_password

目的：更改密码

##### 内容与格式

Json格式

POST /{api version}/change\_password

uuid string

old\_password string

new\_password string

### 领域模型

#### admin领域类

##### 概述

类名称：admin

Id int64

Username string

Password string

Email string

##### 数据窗口分析

业务事件：

AddUsers

ChangeUserPasswd

SetUserProblem

RandomAllUsersProblem

ChangeGCTFMode

UploadProblem

ChangeProblem

DeleteProblem

#### user领域类

##### 概述

名称：user

Id int64

Username string

Password string

Email string

RegisterTime time.Time

ProblemsID string

SolvedProblems string

Score int

IsAdmin bool

##### 数据窗口分析

业务事件：

Register

Logout

Login

GetScore

SubmitFlag

UserDelProblem

#### problem领域类

##### 概述

名称：problem

Name string

Description string

Value int

Category string

Hidden bool

Location string

Scale int

##### 数据窗口分析

业务事件：

StartProblem

GetProblemList

#### hint领域类

##### 概述

名称：

Id int64

ProblemsId int64

Hint string

Cost int

##### 数据窗口分析

业务事件：

GetHint

# 补充规约

## 设计约束

### 技术选择的限制条件

使用go语言，支持mysql，postgresql，Oracle数据库。

前端使用vue框架。

使用容器技术进行整合。

禁止使用Jre相关环境。

### 运行环境



## 质量属性

【本部分建议直接分解成需要开发的技术功能点】

### 安全性要求

#### 访问安全性要求

需要邮箱校验，防止重放攻击，做好权限隔离等。

#### 数据安全性要求

所有密码采用bcrypto进行加密，防止碰撞攻击。

#### 通信安全性要求

页面采用https协议

#### 其他安全性要求

即使更新组件，预防漏洞

### 可靠性要求

#### 容错性要求

数据库采用冗余备份，防止不一致问题

#### 可恢复性要求

采用lvm镜像层，出错时可以做到秒级恢复

#### 其他可靠性要求

无

### 易用性要求

#### 界面友好性要求

用户界面的组织结构应该要完整、清晰，呈现给用户的通用操作序列、术语和信息的措辞，界面元素的布局、颜色搭配方案和排版样式等都要保持一致

界面布局应当体现用户操作时的一般顺序和被使用到的频繁程度。图形界面的布局应当符合人们通常阅读和填写纸质表单的顺序

### 性能要求

#### 数据访问性能要求

支持同步在线的用户数量不低于1000人，平均相应时间不超过50毫秒，同时支持的并发数量不低于10k

#### 数据传输性能要求

带宽不少于2M

#### 其他性能要求

储存器空间不低于20G，CPU不少于2个核心。

### 可维护性要求

#### 公共数据要求

禁止共享任何用户数据

#### 公共框架开发要求

使用相关框架需要写入到相关文档，并且要充分做好测试。

#### 公共程序库开发要求

在注释中需要生命函数的参数，返回值的类型以及意义。必要时给出样例。

尽量避免循环依赖。

#### 其他可维护性要求

无

### 可移植性要求

#### 适应性要求

要求支持Linux 4.14以上版本。可选择支持windows10

#### 易安装性要求

要求使用docker打包，提供部署脚本

#### 其他可移植性要求

无