

基于B/S架构的在线数独对战平台

项

目

总

结

报

告

课 程： 软件工程

题 目： OurSudoku

成 员： SE-2020-G02

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件标识： | SE2020-G02-项目总结报告 |
| 当前版本： | 1.0.0 |
| 作者： | 刘羽佳、张鑫、潘言 |
| 完成日期： | 2021.01.20 |

项目开发总结报告(PDSR)

说明：

项目开发总结报告的编制是为了总结本项目开发工作的经验，说明实际取得的开发结果以及对整个开发工作的各个方面的评价。

**版本历史**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 作者 | 参与者 | 审核人 | 起始日期 | 备注 |
| 0.1.0 | 刘羽佳 | 刘羽佳 | 刘羽佳 | 2021.01.10-2021.01.19 |  |
| 1.0.0 | 刘羽佳 | 刘羽佳 | 刘羽佳 | 2021.01.20 |  |

目录

[项目开发总结报告(PDSR) 2](#_Toc11230)

[1引言 4](#_Toc10091)

[1.1标识 4](#_Toc26874)

[1.2系统概述 4](#_Toc27864)

[1.3文档概述 4](#_Toc6162)

[2引用文件 4](#_Toc152)

[3实际开发结果 5](#_Toc7258)

[3.1产品 5](#_Toc19786)

[3.2主要功能和性能 5](#_Toc24903)

[3.3基本流程 7](#_Toc22166)

[3.4进度 8](#_Toc12740)

[4开发工作评价 14](#_Toc16921)

[4.1对生产效率的评价 14](#_Toc29165)

[4.2对产品质量的评价 14](#_Toc11433)

[4.3对技术方法的评价 14](#_Toc6083)

[4.4出错原因的分析 14](#_Toc22199)

[4.5风险管理 15](#_Toc25217)

[5缺陷与处理 15](#_Toc18659)

[6经验与教训 16](#_Toc3435)

[7注解 17](#_Toc23031)

[附录 17](#_Toc25174)

# 1引言

## 1.1标识

标题：《项目总结报告》

简称：PDSR

版本号：1.0.0

说明：本文档支持Office、WPS等word文档阅读软件

## 1.2系统概述

本文档适用的工程：在线数独对战平台，Beta1.0.0版本

系统的名称：OurSudoku

产品所有权：浙大城市学院计算机与计算科学学院软件工程专业软件工程课程G02小组

任务提出人：杨枨教授

用户：广大数独爱好者

开发方：软件工程课程小组成员

支持机构：浙大城市学院计算机与计算科学学院

## 1.3文档概述

本文档的主要内容是《软件工程》课程实践项目总结，包含了对实践项目的一个基本状况，按阶段性划分，有项目的初始提出与介绍，项目的发展过程，项目最后的结果，项目最后的总结与反思等。除此之外，还包括组员之间的评价与相关寄语等。

本文档符合国家相关法律法规，并受严格保护，具备良好的隐私性和保密性。

# 2引用文件

[1]《GB T-8567-2006计算机软件文档编制规范》

[2]张海藩，牟永敏.软件工程导论(第6版)[M]北京：清华大学出版社

# 3实际开发结果

## 3.1产品

本产品名为OurSudoku，是以网站作为主要的展示形式。

## 3.2主要功能和性能

**3.2.1功能**

**3.2.1.1 用户**

1.用户登录（达成）

2.用户注册（达成）

3.修改密码（达成）

4.忘记密码（达成）

5.退出登录（达成）

6.个人记录汇总（未达成）

**3.2.1.2 比赛**

1.创建房间（达成）

2.加入比赛（达成）

3.退出比赛（达成）

4.进行比赛（达成）

**3.2.1.3 排名**

1.查看排名（达成）

2.排名浮动情况展示（达成）

**3.2.1.4 悬赏**

1.悬赏做题（达成）

2.自主出题（达成）

**3.2.1.5 练习**

1.练习做题（达成）

2.题解查看（达成）

3.编写题解（达成）

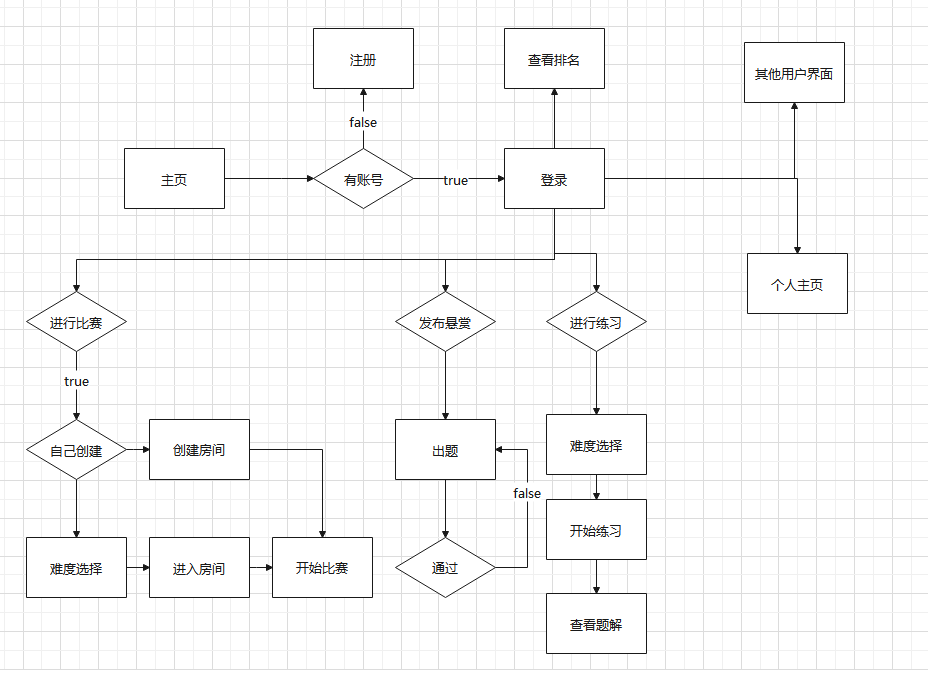
4.不良信息反馈（达成）

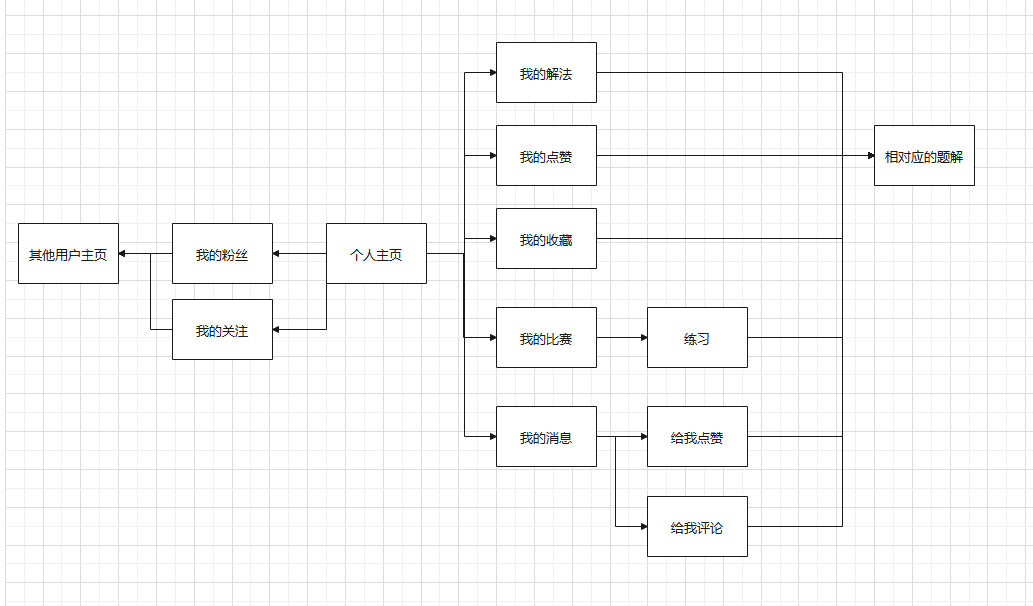
**3.2.2性能**

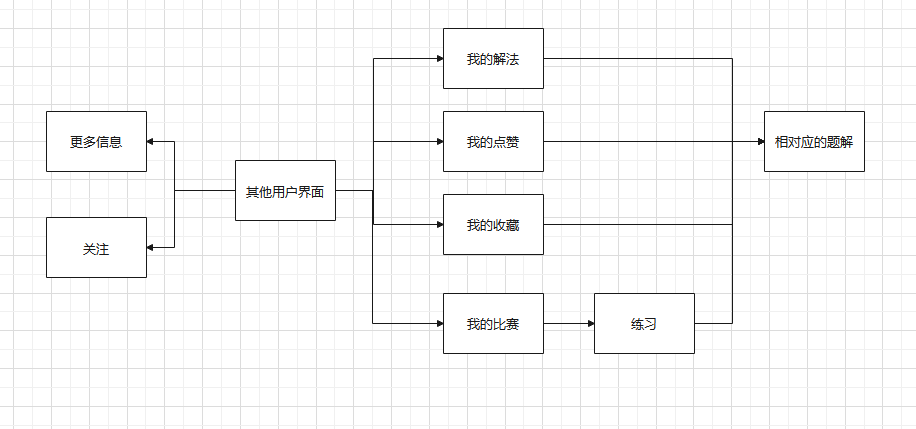
|  |  |  |
| --- | --- | --- |
| 质量特性 | 质量子特性 | 要求 |
| 功能性 | 适宜性 | 合格 |
| 准确性 | 合格 |
| 互用性 | 合格 |
| 依从性 | 合格 |
| 安全性 | 合格 |
| 可靠性 | 成熟型 | 合格 |
| 容错性 | 合格 |
| 可恢复性 | 合格 |
| 可用性 | 可理解性 | 合格 |
| 易学性 | 合格 |
| 可操作性 | 合格 |
| 效率 | 时间特性 | 合格 |
| 资源特性 | 合格 |
| 可维护性 | 可分析性 | 合格 |
| 可修改性 | 合格 |
| 稳定性 | 合格 |
| 可测试性 | 合格 |
| 可移植性 | 适应性 | 合格 |
| 易安装性 | 合格 |
| 一致性 | 合格 |
| 可替换性 | 合格 |

同时作为一个在线数独游戏对战平台，应保证满足一定量人数的比赛访问，一般来说，一场数独游戏的人数一般在10-15人以内，一段时间内同时进行的游戏场数一般是3-5场，所以服务器起码要能满足最少30人，最多75个人同时进行游戏的压力。（达成）

## 3.3基本流程

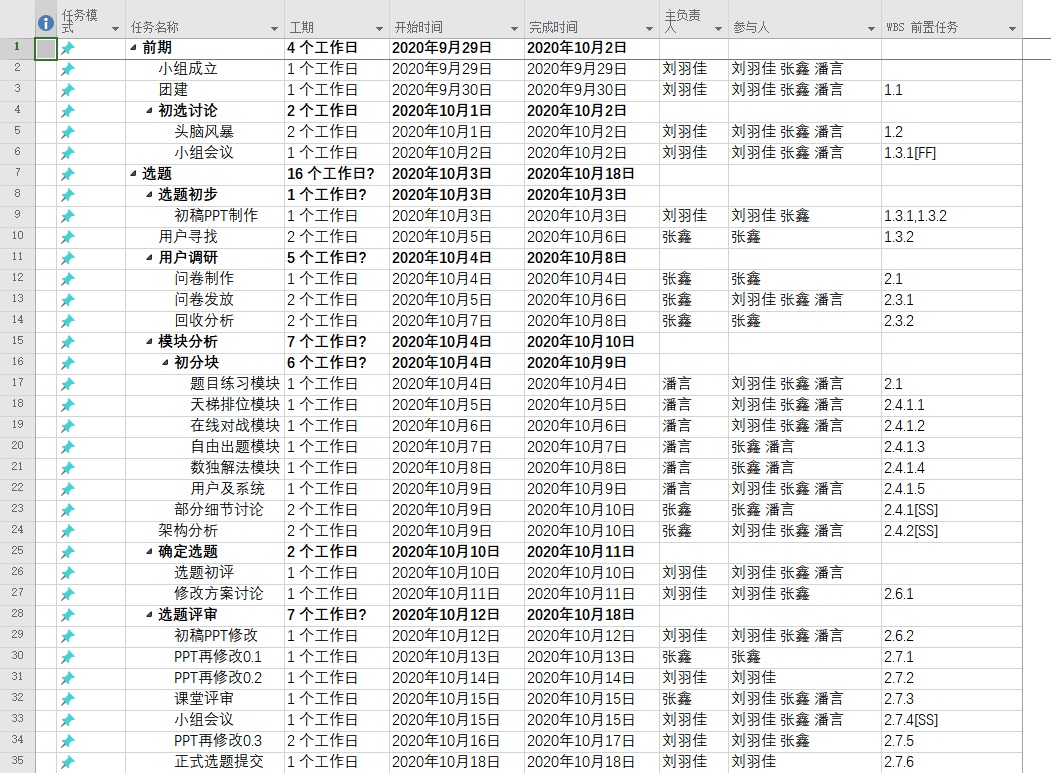


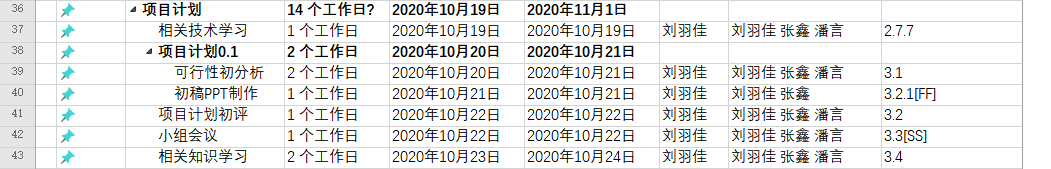


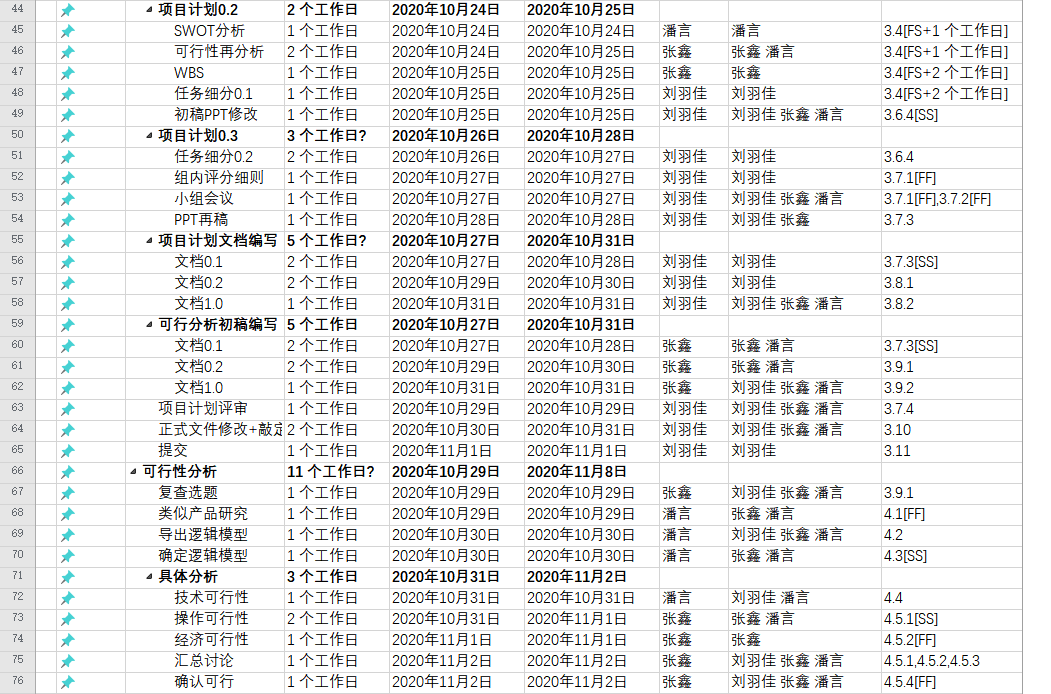


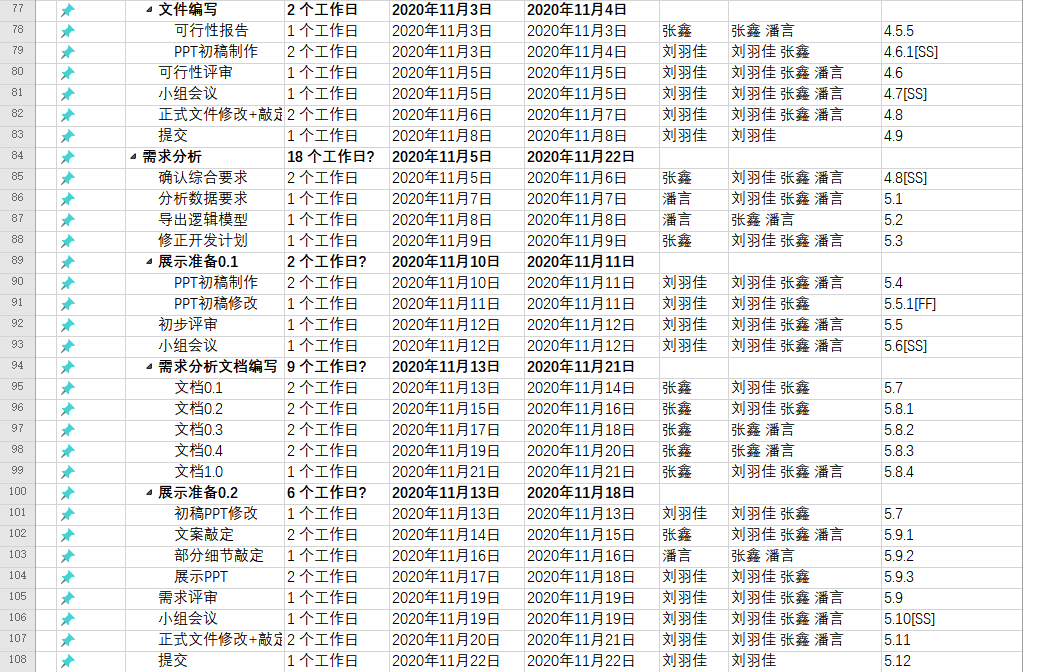
## 3.4进度

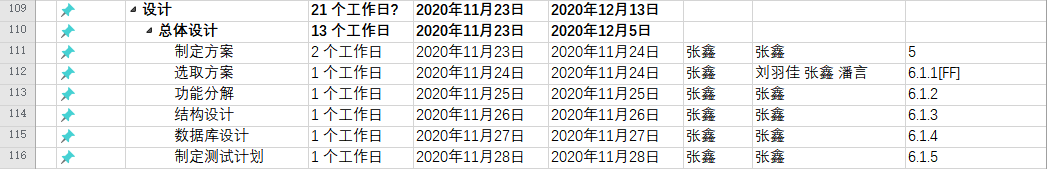
原计划进度：



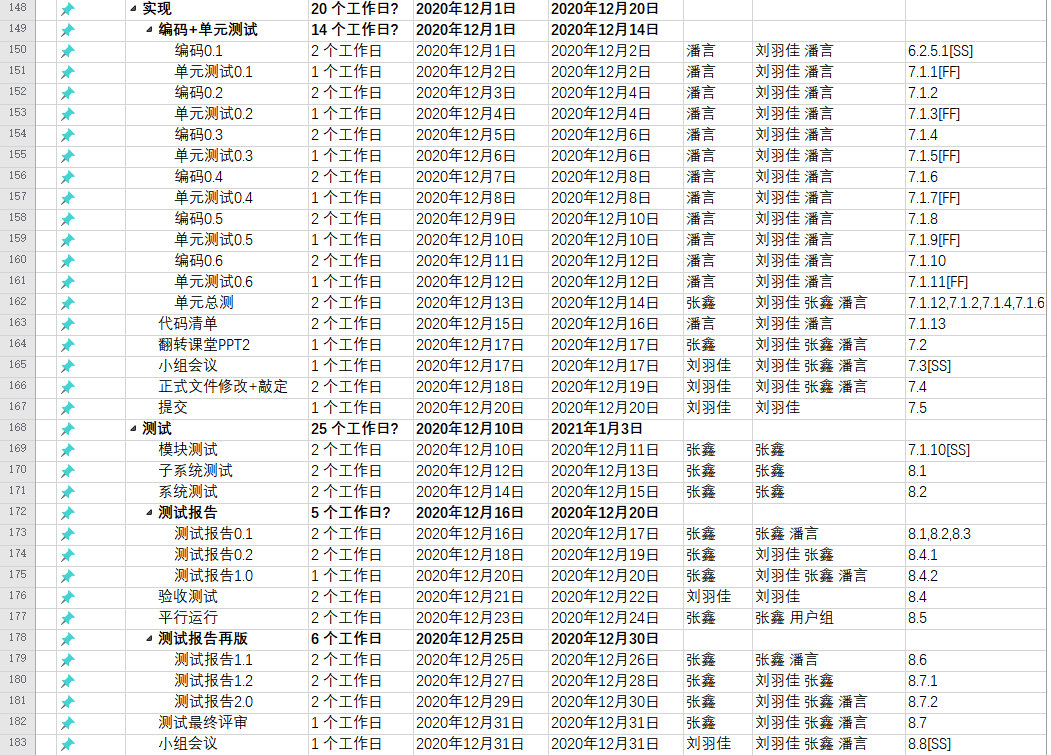


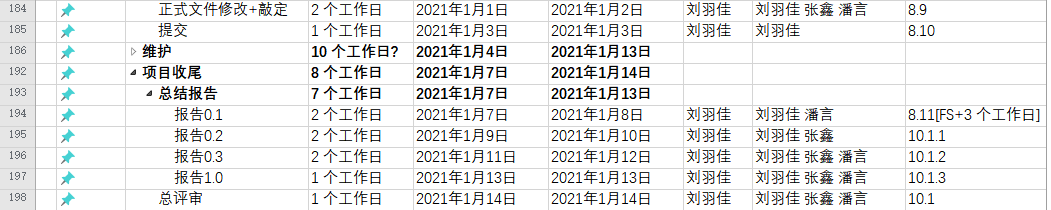




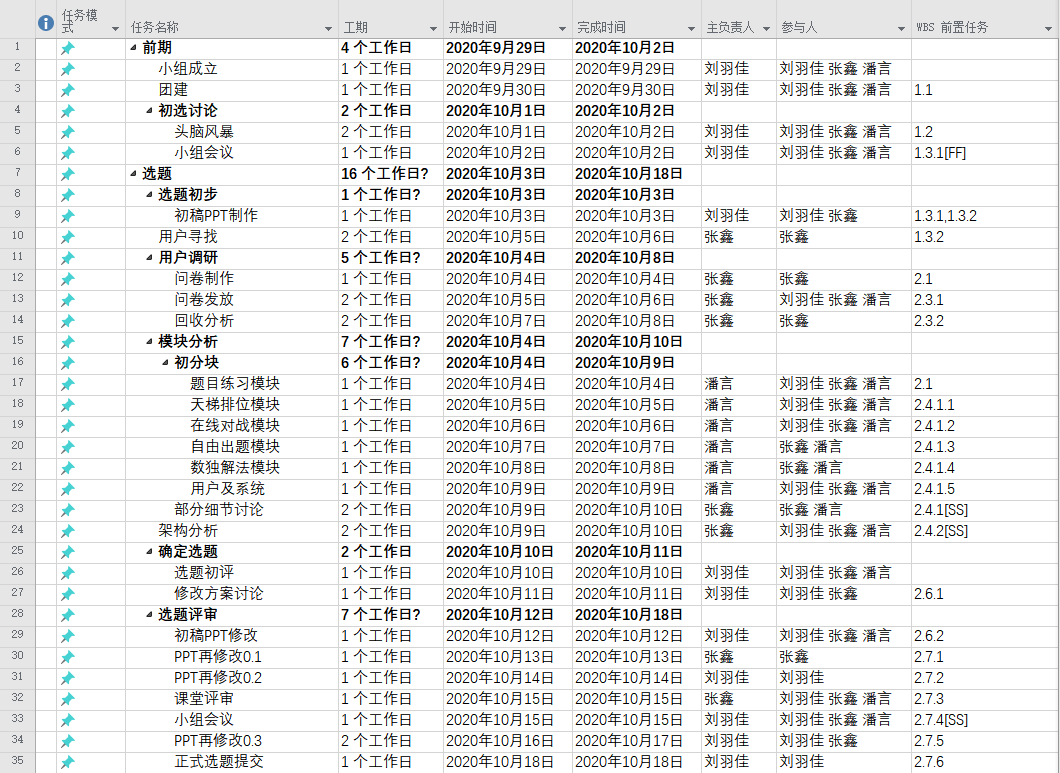


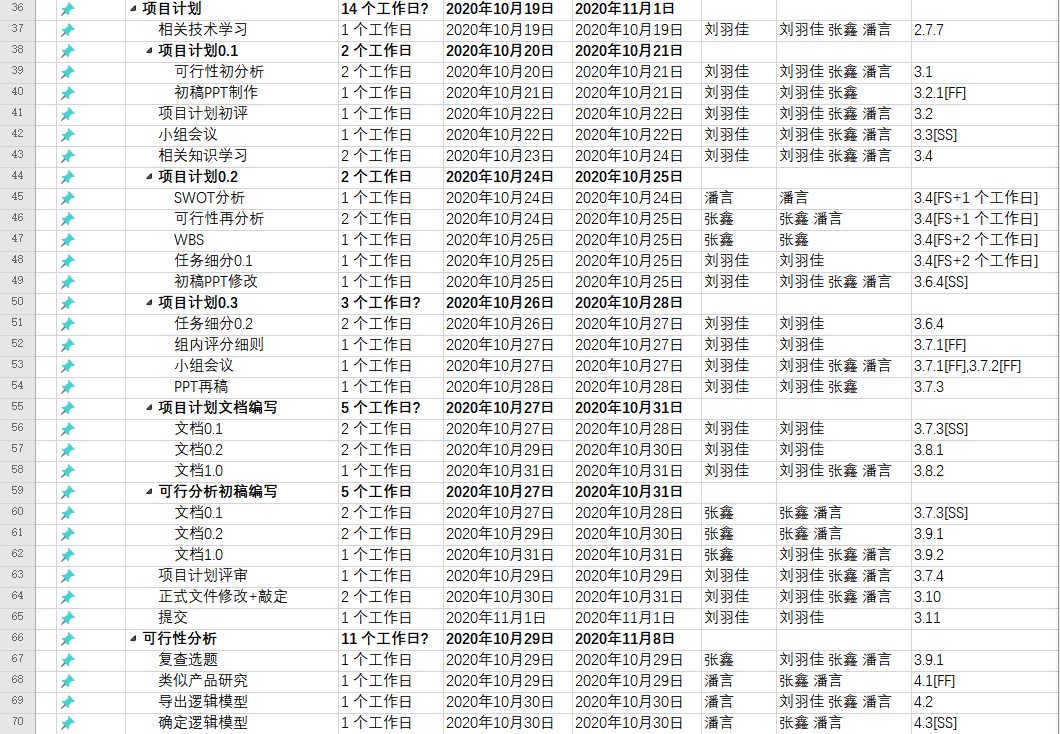


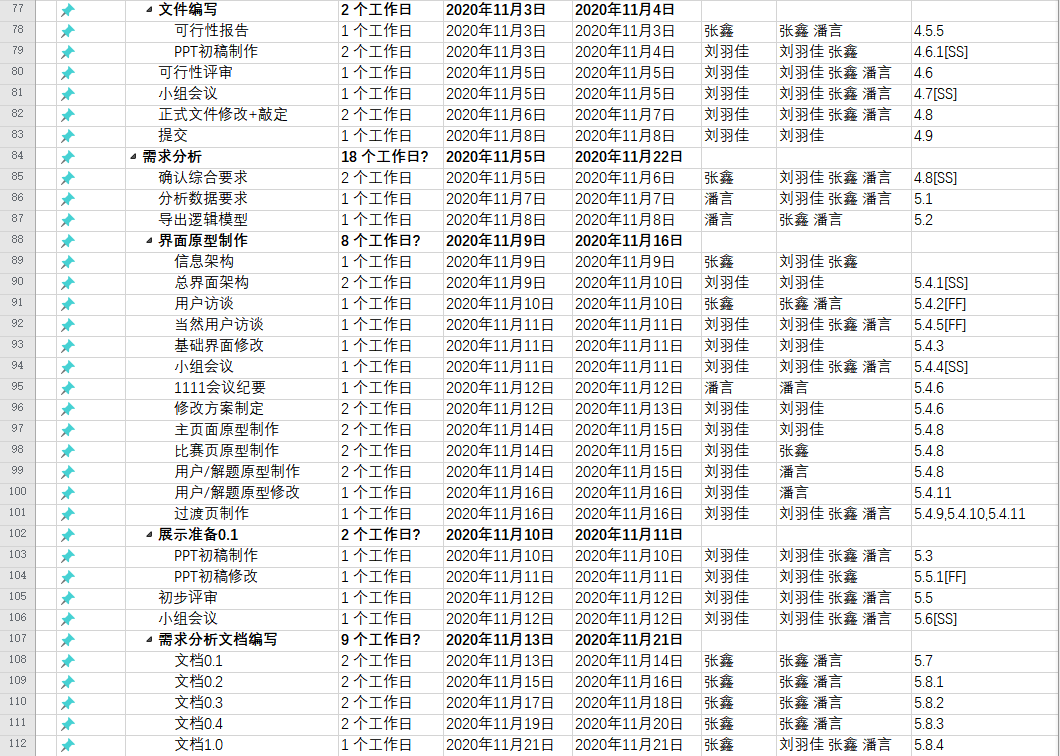
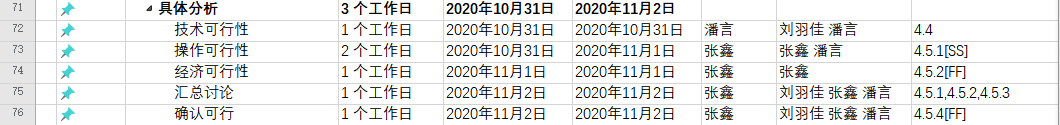




实际进度：

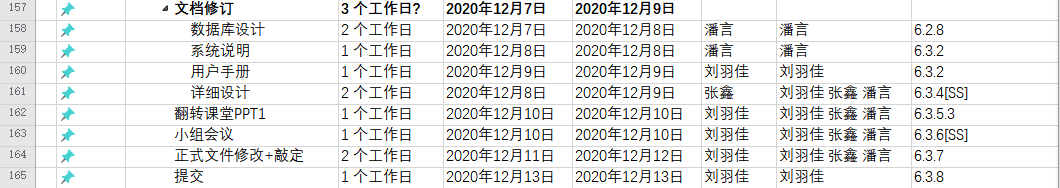


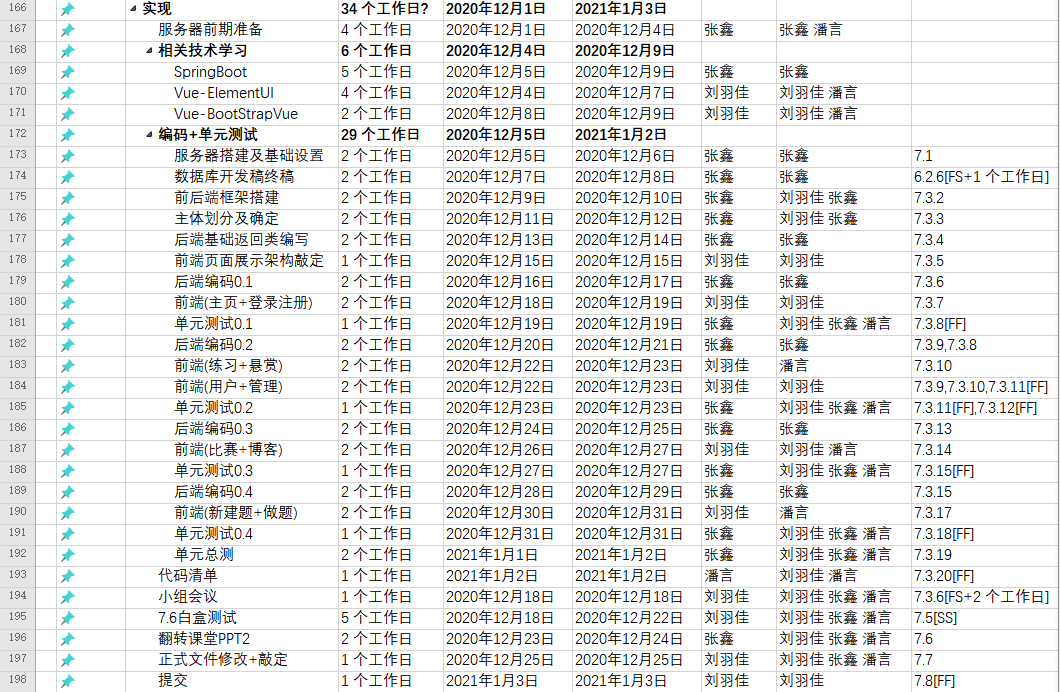


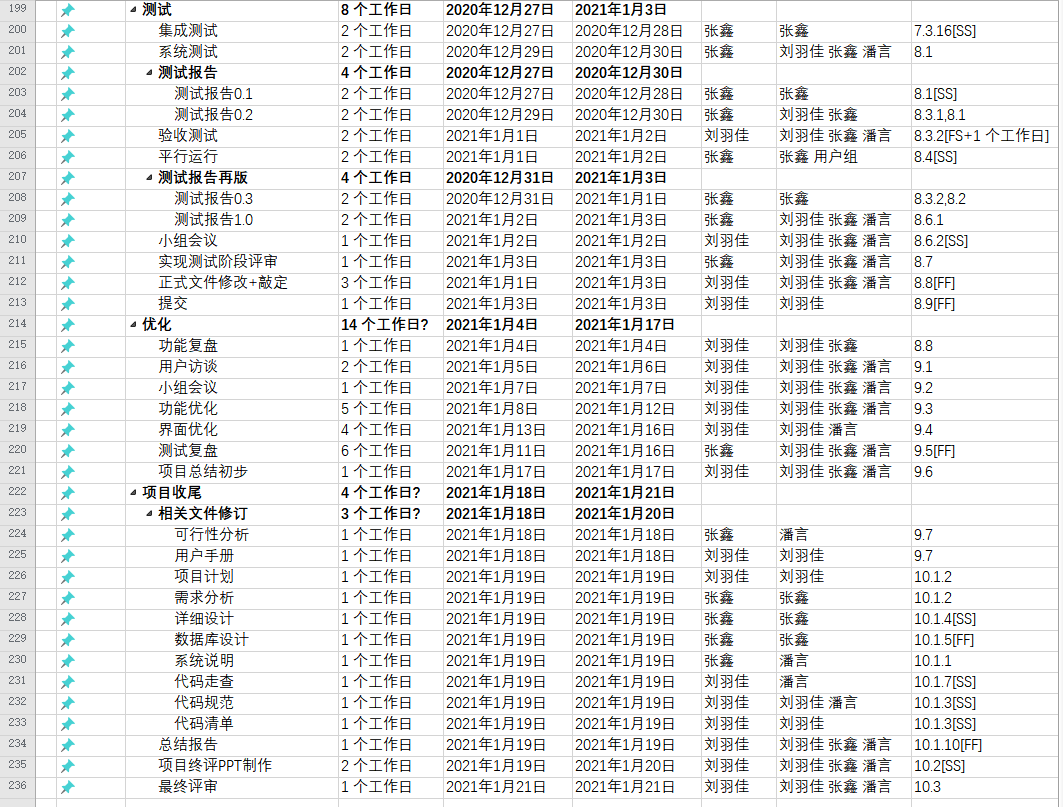












从大阶段进度来看，并没有问题，但在预计的维护阶段遗留下了很多前阶段实现测试部分的任务，直接导致了项目进度延后。很多的原因在于设计阶段有一些过于混乱了，致使很多本该在总体设计和详细设计部分完成的内容，都需要在实现阶段推翻重来，加上小组成员本身缺乏项目经验，在前期学习上浪费了很多时间，导致实现阶段的进度严重滞后，由此影响了之后所有的项目计划。

# 4开发工作评价

## 4.1对生产效率的评价

据统计计算，前后端代码总计25419行，由于实际开发时长为1个月，因此程序的平均产生效率为8473行/人月；产生工作文档21份，共计148038字，历时3个月，平均效率为16.449千字/人月。

## 4.2对产品质量的评价

每千条指令(或语句数)中的错误指令数(或语句数)在40-50。在我们成员组看来已经达到了预期，对于第一次投入开发的我们来说已经满意。

## 4.3对技术方法的评价

我们在整个项目中使用的技术：前端使用vue，vue.js作为比较流行的前端框架，拥有现成的组件和模板，可以实现较为复杂但优美的界面。后端使用springboot，springboot作为比较流行的后端框架，由于采用了java语言，所以其在数据库操作以及服务的计算上，都有着很大的先天优势，方便开发者合作开发，分块开发。

主要使用的开发工具：IDEA、vscode、Webstrom、HBuilderX、Axure RP、Git、Visio、Project、MySQL、Navicat、Postman等。

## 4.4出错原因的分析

交流不够、有误解。因为软件几乎是分模块开发，没有沟通好很容易出bug，后面修改耗费时间也长，造成延期。

软件复杂性较高（对于目前的我们来说）。

程序个人理解错误，导致必然的编码错误，并且需要很多时间去重整重构。

设计阶段过于混乱，导致花很多时间做的设计不能直接投入开发，需要重做。

需求变化，直接影响项目的结束时间，同时增加开发的工作量。

在实现阶段后期由于时间压力，使开发误码率增大，且测试不够全面。

## 4.5风险管理

1、需求不明确

解决：1）开发用户界面原型；2）需求讨论会议；3）强化需求分析与评审。

2、项目缺少可见性

解决：1）迭代开发；2）技术评审；3）持续集成。

3、性能不足

解决：1）性能规划；2）性能测试（尽可能模拟现实使用环境，搭建测试平台）；3）保留充足的调试时间。

4、时间仓促

解决：重视前期规划。

5、可用性不足

解决：1）了解用户；2）用户参与设计；3） 竞争性分析。

# 5缺陷与处理

**需求评审阶段：**

我们做了初步的用户问卷调查，但由于问卷设计本身并不尽善尽美同时样本量不够大，样本种类较为单一，无法提取到足够多的有用信息，因此我们对基础用户调研得来的用户特点进行了分类处理，大致分为了数独资深爱好者、数独小白及可能会想要了解数独的路人（此处包含最终用户杨枨老师），以选取代表用户的方式进行了深度访谈，并对需求的基础分析进行了整改。

**设计评审阶段：**

由于原定阶段负责人对项目的熟悉程度不足以及一些其他个人原因，造成该阶段半持续性的混乱，直接表现在进度不合理、设计阶段文件不可用等后果。临时更换负责人之后情况稍有好转，但依旧没有完全解决，直接影响了下一阶段的正常工作。

**代码实现阶段：**

由于设计阶段的不合理安排，许多本应在设计阶段完成的任务不得不放在实现阶段重新讨论甚至返工；加之组内成员均是第一次实践开发任务，对于需要使用的技术熟悉度不够使用并不熟练，学习曲线十分难看，致使开发进度严重滞后、成员心理濒临崩溃。由于当前的成员组并没有能力解决这样的问题，经组内商议决定，向课程导师杨枨老师寻求帮助。在杨枨老师的帮助下，我们重新审视了所需开发项目，修改调整了当前开发计划，并强硬执行新策略，情况有所缓解。但由于前期工作的欠账太多，依旧影响了原有的项目计划。

**系统测试阶段：**

由于实现阶段的进度延后，导致测试阶段的进度安排顺期延后。但测试阶段的工作依旧顺利完成，有关测试的最重要的文档，测试计划，在最开始时因为对待测试项的了解不够导致测试计划的内容缺失严重，测试计划的补充修订花费了一段时间。最终确定了测试部分的各个阶段，分为七个阶段，其中最开始的接口测试由于工具的使用不熟练导致只能手动测试，知道进行压力测试的时候，才知道阿里云PST对测试的一些阶段内置有工具，于是在压力测试后的阶段，我们采用了阿里云的PST同时结合postman进行了后续的测试，终于在项目总结前完成。

# 6经验与教训

**刘羽佳：**

从大一开始一直被强调人月神话的概念，也一直不以为意，直到现在，可以说是几乎毫无准备地掉进软件工程泥潭。项目管理团队沟通真的是门大学问。作为一个并不合格的项目经理，对组内成员的具体情况了解不足，做出不少错误的决定。简单说来在不该强硬的时候强硬，在该当机立断的时候反而想要再给一次机会。间接造成了几乎无法挽回的错误，致使团队陷入阶段性混乱，导致项目进度严重滞后，这是作为项目经理非常糟糕的一点。

其次是对项目实现所需技术的学习曲线预估错误，导致团队在已经进入实现阶段之后，成员还对要使用的技术基本没有了解。由此还有另外一个问题：前期已经明确要使用的技术之后，尚未进入实现阶段之前的很大一片时间，是拿给你快速学习相关技术的，而不是回寝室躺下；越快投入学习越好，而不是等到设计阶段都要结束了还没开始学。

**张鑫：**

通过这次软件工程课程项目，实实在在的了解到了一个项目的管理与运行是及其复杂的。具体体现在很多方面，但感觉其中最大的问题就是沟通合作问题。一直以为做互联网项目时最大的问题是技术问题，比如某种技术无法实现导致项目崩盘，但知道自己亲手参与一个项目的时候，才知道沟通合作才是最大的问题。最主要的体现就是项目成员对自己项目的不了解或者某些名词的认知不统一，导致实现阶段实际开发时，每个人的对项目的认知程度不同造成了这张图的结果：



所以在合作过程中，沟通交流一定是最需要解决的问题。

通过本次项目，我学到了如何和小组成员进行沟通，如何让小组成员知道我在做什么，我做了什么，我为什么要这么做。这点是学到的最重要的技能。

其次我学到了，如何和用户沟通进行需求分析，如何设计实现一个具体的项目，还学会了如何使用springboot构建后端项目，如何使用tomcat部署前端项目，如何使用git管理项目的文件。

今后从事开发工作，一定要和同项目组的组员做好充分的沟通，确保我们对开发的项目的认知是一致的，避免造成上述图片的后果。

**潘言：**

在这次的小组合作过程中，我认识到了一个项目开发过程是如何的复杂，并不是只需要敲代码就能完成的，需要经过一系列的分析，最终决定后再动手写代码。在一个项目的开发过程中不能逞强，自己干不好的事要及时与其他组员沟通，否则拖下去只会造成严重的后果。

对名词的解释一定要与组员先行确定，因为不同的人对一个名词可能有不同的理解，有可能会因为理解的偏差导致最终做出来的成果不尽人意。

这次因为我个人的疏忽导致全组的进度被拖后，最终需要整组成员熬夜赶工，所以一个团队里不能有任何人掉链子，一个人掉链子会牵连整组的进度，轻则熬夜赶工，重则项目延期甚至无法完成。

通过这次的代码编写，我自学了VUE，学会了VUE中的路由设置、组件调用、动态绑定等基础内容，学会了基本的网页制作。

# 7注解

# 附录