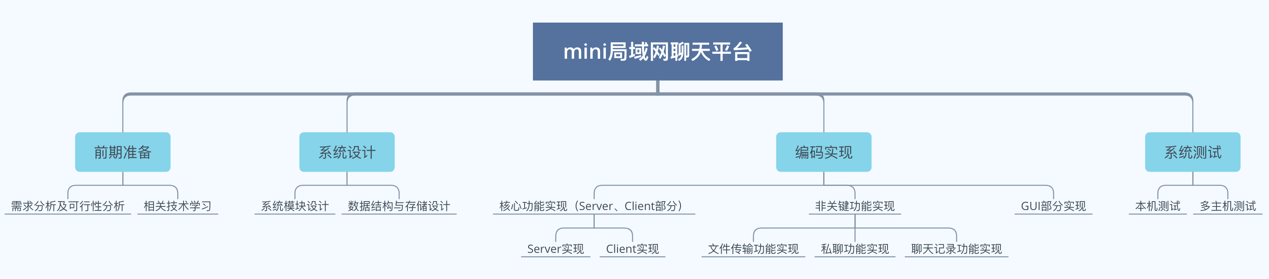
mini局域网聊天室

### WBS任务分解



WBS层次结构图

1. **Mini局域网聊天平台**
   1. **前期准备**
      1. **需求分析及可行性分析**

内容：分析此聊天平台所需具备的功能，及确定实现这些功能所选择的技术、语言及平台

时间：2小时

工作产品：需求分析及可行性分析报告

* + 1. **相关技术学习**

内容：学习实现此聊天平台所需要的一些基础知识，为后期设计与实现打下基础，本次任务中参考书籍《Python网络编程基础》

时间：4小时

工作产品：无

* 1. 系统设计
     1. 系统模块设计

内容：分析系统需求，设计系统的功能模块

时间：2-4小时

工作产品：系统模块设计图及报告

* + 1. 数据结构与存储设计

内容：分析系统内所使用到的数据的形式，进行数据的持久化设计，并选择使用合适的数据库

时间：2-4小时

工作产品：数据库结构图

* 1. 编码实现
     1. 核心功能实现（Server、Client部分）
        1. Server实现

内容：系统服务器部分实现并测试

时间：2小时

工作产品：服务器demo

* + - 1. Client实现

内容：系统客户端部分实现并测试

时间：2小时

工作产品：客户端demo

* + 1. 非关键功能实现
       1. 私聊功能实现

内容：实现并测试两名用户单独聊天的功能

时间：2小时

工作产品：用户单独聊天功能demo

* + - 1. 文件传输功能实现

内容：实现并测试用户之间文件传输功能

时间：2小时

工作产品：文件传输功能demo

* + - 1. 聊天记录功能实现

内容：实现并测试用户的聊天记录本地保存与读取功能

时间：2小时

工作产品：聊天记录功能demo

* + 1. GUI部分实现

内容：为聊天室绘制可视化姐买呢

时间：8小时

工作产品：聊天室可视化界面

* 1. 系统测试
     1. 本机测试

内容：在本机上模拟测试聊天室的功能与可用性，及各功能与可视化界面的易用性

时间：4小时

工作产品： 本机测试报告

* + 1. 多主机测试

内容：测试局域网环境下多主机的聊天室功能的正确性

时间：4小时

工作产品：多主机测试报告

### 规模估算

这里采用国际标准IFPUG组织提供的FP（功能点）估算法对整个系统进行规模估算。

FP = CT\*[0.65+0.01\*]

其中：

数据类型功能点：ILF：用户信息文件：含用户账号、密码、用户名

聊天记录文件：含聊天发送人，内容，时间

EIF：本系统不含EIF文件

共2个ILF

人机交互功能点：1、用户修改密码

2、用户修改用户名

3、用户发送消息，影响聊天记录

4、用户查询聊天记录

5、用户创建用户账号、密码、用户名

共4个EI，1个EQ

计算EI复杂度：涉及到2个FTR，6个DET，中

计算EQ复杂度：涉及到1个FTR，3个DET，低

计算ILF复杂度：涉及到涉及到6个DET，2个RET，低

故CT=4\*4+1\*3+2\*7=33

（EQ为4，当复杂度为中时\*4，EI为1，当复杂度为低时\*3，IFL为2，当复杂度为低时\*7）

再对系统进行14项评估

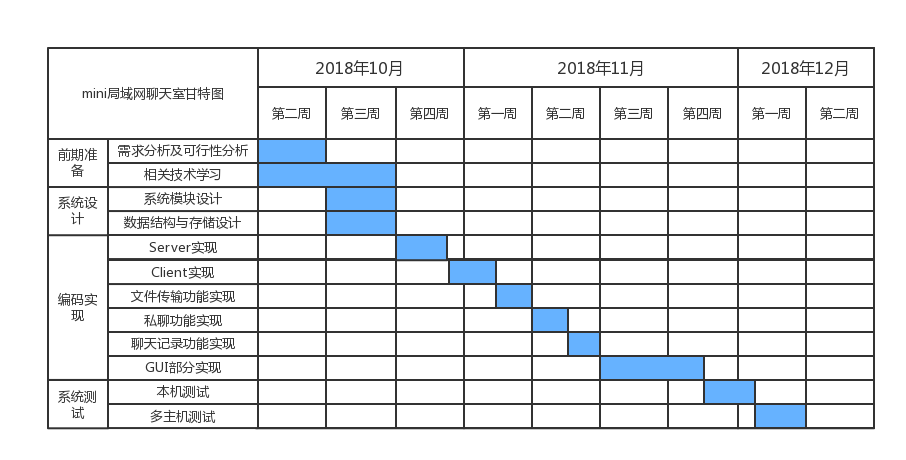
1. 备份与恢复：3
2. 数据通信：5
3. 分布处理：0
4. 性能：4
5. 操作环境：3
6. 联机数据项：5
7. 联机数据项多屏幕输入：0
8. 联机更新主文件：0
9. 查询复杂：0
10. 内部处理复杂：0
11. 代码可复用：2
12. 转换及安装：4
13. 不同组织多次安装：4
14. 方便用户修改：0

故Fi的和为30

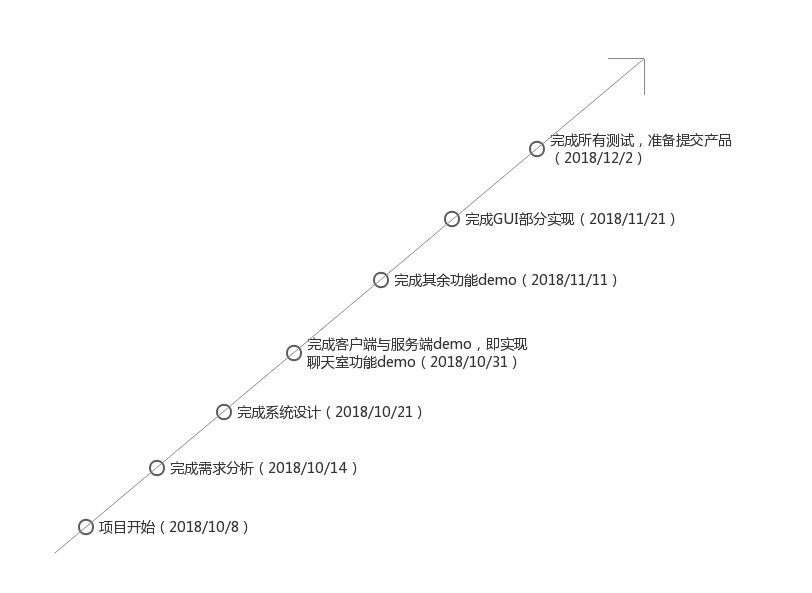
故最终的FP功能点估算为33\*（0.65+0.01\*30）=31.35

### 进度估算

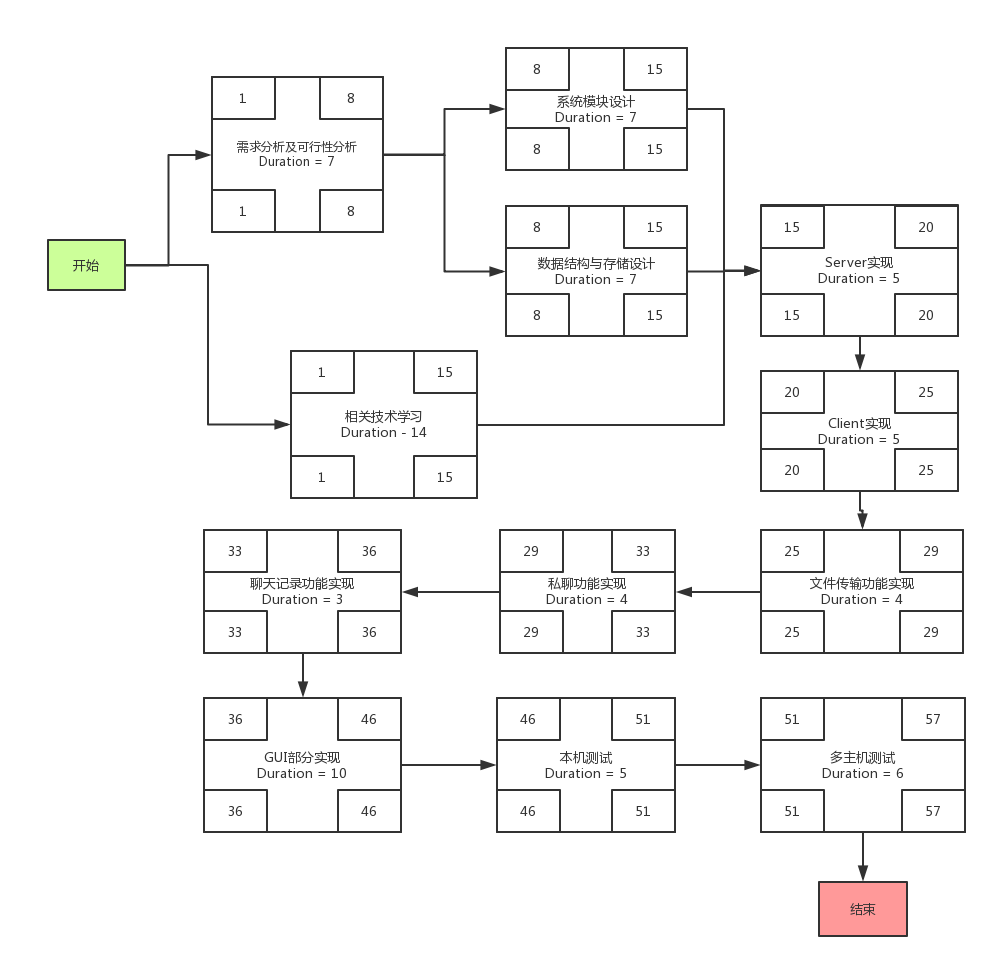
甘特图：



里程碑图：



CPM：



关键路径：需求分析->系统模块设计->Server实现->Client实现->文件传输功能实现->私聊功能实现->聊天记录功能实现->GUI部分实现->本机测试->多主机测试

关键路径长度：56

### 风险管理

风险标识：

|  |  |
| --- | --- |
| 问题 | 风险程度（0…5） |
| 开发人员水平如何 | 2 |
| 技术是否配套 | 3 |
| 系统的性能如何（可支持多少用户同时使用） | 4 |
| 系统的耦合性如何，是否有利于后续添加新功能 | 2 |
| 开发人员能否集中精力投入开发 | 1 |
| 系统的安全性如何 | 4 |
| 开发过程中，是否会有其他事件导致开发过程受阻 | 2 |

预测风险：

风险表

|  |  |  |  |
| --- | --- | --- | --- |
| 风险 | 种类 | 概率 | 影响 |
| 系统最终支持同时使用的用户数量可能不理想 | 性能风险 | 60% | 会导致聊天室可支持的用户数较低，十分影响用户体验 |
| 系统的安全性可能欠佳 | 性能风险 | 55% | 会导致无法很好的保护用户的隐私 |
| 系统的耦合性过高 | 支持风险 | 40% | 过高的耦合性使后续难以修改系统，难以添加新功能 |
| 开发人员不擅长这方面的编程任务 | 进度风险 | 35% | 有可能导致项目进展缓慢，延长开发周期 |
| 开发过程中，碰到其他活动使开发人员抽不开身，影响开发进度 | 进度风险 | 30% | 影响开发进度，演唱开发周期 |
| 开发人员水平不足 | 进度风险 | 20% | 导致项目失败，无法按照实现预定功能，满足客户需求 |

评估风险：

[系统支持用户数量不理想，60%，较低]：加强对服务器并发能力的检测，在server、client的demo完成后，对服务器进行压力测试，若不理想，则查询资料提升服务器的并发处理能力。

[系统安全性欠佳，55%，较低]：提前学习目前主流的加密算法，保护用户敏感数据

[系统耦合性过高，40%，较低]：在系统设计阶段，注重高内聚，低耦合的原则，尽量避免此情况的发生。若已发生，在时间允许的情况下考虑重构。

[开发人员不擅长，35%，中]：开发人员提前学习储备相关知识，并在开发期间与相同选题同学多多交流，共同进步完成任务。

[开发过程中碰上其他事项，30%，中]：在安排开发时间时，预留一定的缓冲时间，为突发情况提供缓冲余地。同时，严格按照甘特图等实施开发计划，加强监管，精力允许情况下适当提前完成任务。

开发人员水平过低

开发进度延迟

[开发人员水平不足，20%，严重]：加强相关技能的学习，在开始阶段先研究相关背景，保证项目的完成。

风险参考水准：

### 质量管理

QC：创建一个反馈循环，在每个里程碑完成时进行评测，确保提交物满足预期的规范。具体规约如下：

1. 立项阶段：提交一个完整的文档，描述项目内容，及各阶段时间规划，以及预测项目过程中可能遇到的风险等。
2. 需求分析：提交一个完整的需求分析文档，详细描述本项目所要满足的需求及使用场景。
3. 系统设计：提交一个关于系统设计的文档，清晰描述系统分为几个模块，各模块需要完成的功能，以及模块之间所需要使用的接口，设计过程要注重高内聚低耦合的思想。
4. 客户端与服务器demo：测试客户端与服务器demo功能，是否能够实现聊天室功能并可以稳定使用。
5. 其余功能demo：测试文件传输、私聊、聊天记录功能是否可以满足用户需求，并可稳定使用。
6. GUI功能demo：完成程序的可视化界面实现，并邀请非专业用户评价界面的易用性及美观程度。
7. 提交产品：对产品进行综合测试，并根据测试结果给出报告，检查是否已满足需求，并在美观程度、稳定性等方面给出评价。

QA：邀请同学进行监督，在QC的每个环节予以监督评审，具体示例见下表。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 项目名称 | Mini聊天室项目 | | | | | | |
| 要求评审的工作产品的名称 | 项目立项报告书 | | | | | | |
| 负责人员 | 唐啸 | | | 评审人员 | | 矫杨 | |
| 要求评审的工作产品所属  开发阶段 | 规划阶段 □ 需求分析阶段 □ 系统设计阶段  □ 实现与测试阶段 □ 系统验收阶段 □ 安装运行阶段 □ 其它 | | | | | | |
| 评审准则 | 检查文档是否合规，计划分解是否合理，进度估算是否合理，风险管理是否可行。 | | | | | | |
| 评审需提交  的资料 | WBS分解图  WBS分解任务列表  项目进度估算报告  项目风险管理报告 | | | | | | |
| 技 术 评 审 意 见 及 结 果 | | | | | | | |
| 评审时间 | 自2018 年10月12日08时 至 2018 年10月14日08时 | | | | | | |
| 评审  建议 | 文档基本符合规范  任务分解合理，控制在4个工作时内  项目选题明确，具有可实现度 | | | | | | |
|  | 记录人签名 | | 矫杨 | | 日 期 | | 2018 年10月 14日 |
| 评 审  人员签名 | 矫杨 | | | | | | |
| 其他参与  人员签名 | 无 | | | | | | |
| 补充说明 |  | | | | | | |
| 评审结论 | | □**评审通过：**工作产品合格，“无需修改”或“需要轻微修改但不必再审核”；  **评审基本通过：**工作产品基本合格，需要作少量修改，之后通过审核即可；  □**评审不通过：**工作产品不合格，需要作比较大的修改，之后必须重新对其评审。 | | | | | |

### 配置管理

本项目中，使用Github进行配置的管理。在SCI被确定时，评审其是否遵守已制定的规程与规格、其实现是否有效以及它与其他SCI的一致性，确定无误后进行上传保存。通过Github，对项目进行版本控制，方便对系统的不同版本进行标识和跟踪。

Github地址：https://github.com/a39153973/mini\_chatroom