**即时聊天软件：吾亦聊**

项目结题报告v1.3

**修订历史记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2018/12/12 | V1.0 | 初步完成内容 | 杨健威 |
| 2018/12/17 | V1.1 | 审阅 | 张媚 |
| 2018/12/17 | V1.2 | 完整版 | 卢越兴 |
| 2018/12/18 | V1.3 | 排版校对 | 魏芸 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**发行记录**

|  |  |  |
| --- | --- | --- |
| **日期** | **副本编号** | **持有者及角色** |
| 2018/12/17 | 1 | 项目经理 杨健威 |
| 2018/12/17 | 2 | 项目组成员 杨玉卿，罗锦昊，魏芸，张媚，卢越兴 |
| 2018/12/17 | 3 | 客户代表 罗比·金斯顿(Lobby Kingston) |
|  |  |  |
|  |  |  |
|  |  |  |

**目录**

[1 引言 3](#_Toc532896700)

[1.1 目的 3](#_Toc532896701)

[1.2 范围 3](#_Toc532896702)

[1.3 目标读者 3](#_Toc532896703)

[1.4 定义 3](#_Toc532896704)

[1.5 参考资料 4](#_Toc532896705)

[2 组织架构与角色安排 4](#_Toc532896706)

[2.1 项目组织 5](#_Toc532896707)

[2.2 领导组织 5](#_Toc532896708)

[3 结果、计划、质量、工作量分析 5](#_Toc532896709)

[3.1 结果分析 5](#_Toc532896710)

[3.2 里程碑和重大事件 6](#_Toc532896711)

[3.3 计划分析 6](#_Toc532896712)

[3.4 质量分析 6](#_Toc532896713)

[3.5 工作量估计分析 6](#_Toc532896714)

[3.6 暂停与变更的需求分析 6](#_Toc532896715)

[4 开发方法和开发过程 7](#_Toc532896716)

[4.1 开发环境和工具 7](#_Toc532896717)

[4.2 技术途径 7](#_Toc532896718)

[5 风险分析 7](#_Toc532896719)

[5.1 风险1 7](#_Toc532896720)

[5.2 风险2 7](#_Toc532896721)

[6 经验与教训 7](#_Toc532896722)

[6.1 成功经验 7](#_Toc532896723)

[6.2 教训 8](#_Toc532896724)

[7 对未来项目开发的建议 8](#_Toc532896725)

[8 整体评价 8](#_Toc532896726)

1. 引言
   1. 目的

本文档旨在总结吾亦聊系统开发过程。对项目进行结题报告。

* 1. 范围

本文应包括组织介绍、项目结果和进度、部署方案、风险分析和经验总结。

* 1. 目标读者

所有与系统项目开发相关的人员。

* 1. 定义

|  |  |
| --- | --- |
| 术语 | 解释 |
| DCD | Domain Class Diagram，显示了模型的静态结构，特别是模型中存在的类、类的内部结构以及它们与其他类的关系。 |
| Web | 网络。 |
| UI | User Interface，即用户界面。 |
| 房间 | 一个服务端程序可以有多个不同的房间，不同的房间中的模型可互不相同。在房间一中的用户无法看见房间二中的用户，房间二中的用户可以接收到来自房间一中用户的信息仅为私聊频道消息和世界频道消息。 |
| 广场 | 对所有用户开放的多人在线聊天房间。 |
| 家园 | 由用户所私有的多人在线聊天房间。 |
| 虚拟角色 | 用户所控制的3D人物。 |
| 上线/在线 | 表明用户已连接到服务器。与之相对的是“下线/离线”。 |
| 同屏其他用户 | 指可以观察到当前用户的虚拟角色的用户们。 |
| 外观 | 虚拟角色的体型、涂装、装饰等可进行自定义的造型组合。 |
| 涂装/皮肤/预设 | 指虚拟角色3D模型的贴图，可自定义。 |
| 消息 | 聊天的基本单元，由文字、表情图片、其他附件组成。 |
| 转发消息/推送消息 | 指服务器将收到的用户发出的消息，再将消息分发给应当收到此消息的用户们。 |
| 聊天记录 | 可以存储在本地，仅显示上线后收到的消息。 |
| 频道 | 聊天记录的分类，分为“区域”、“世界”、“私聊”、“全部”四个频道。 |
| 区域 | 区域是指一个以用户为中心的几何区域，象征用户的听力范围。当用户在室外时，区域是半径一定的圆形区域，减去建筑截面；当用户在室内时，区域是该层建筑截面和半径一定的圆形的交集。 |
| 区域频道 | 同区域的其他用户发出的消息进入此频道。 |
| 世界频道 | 其他用户发出的世界消息进入此频道。世界消息指有一定限制（如需要使用道具才能发布）、推送到同一服务器服务端程序中所有房间的用户的消息。 |
| 私聊频道 | 与所有其他用户的跨房间私聊往来消息进入此频道。 |
| 全部频道 | 区域、世界、私聊频道的消息全部汇总后的频道。 |
| 动作 | 虚拟角色依照预设的动画做出动作。 |
| 房间物体 | 房间中的可进行互动的3D模型。 |
| 互动 | 执行脚本，改变虚拟角色的动态，或改变房间的样子。 |
| 管理员 | 具有超越普通用户权限的用户，辅助维护社区秩序。 |
| 全服公告 | 管理员发布的特殊消息，具有标识，将转发到同一服务器所有用户处。 |
| 家园主人 | 家园的拥有者。 |
| 访客 | 指目前停留在家园中的其他用户，与家园是否公开无关。 |
| NPC | 指非用户可控角色，在房间中提供某些互动选项。 |

* 1. 参考资料

1. 参考书目：《UML和模式应用》第三版，Craig Larman著
2. 参考书目：《IT项目管理》第7版，Kathy Schwalbe著
3. 参考文档：《软件构架实践》第3版，Len Bass, Paul Clements, Rick Kazman著
4. 组织架构与角色安排
   1. 项目组织

|  |  |  |
| --- | --- | --- |
| 名字 | 角色 | 职责 |
| 杨健威 | 项目经理 | 管理项目 |
| 杨玉卿 | 开发人员 | 实现UI和前后端对接 |
| 罗锦昊 | 开发人员 | 实现房间模型 |
| 魏芸 | 开发人员 | 实现人物编辑 |
| 张媚 | 开发人员 | 实现Web后端 |
| 卢越兴 | 开发人员 | 实现管理员端页面 |

* 1. 领导组织

|  |  |  |
| --- | --- | --- |
| 名字 | 角色 | 职责 |
| 杨健威 | 项目经理 | 管理项目 |

1. 结果、计划、质量、工作量分析
   1. 结果分析

#### 完成的交付物

1. 需求文档

2. 概要设计

3. 详细设计

4. 测试报告

5. 用户手册

6. 系统数据模型

7. 项目开题报告与计划

8. 项目章程

9. 周报告

10. 项目范围

11. 项目更改报告

12. 成本估算

13. 结题报告

14. MS Project文件

15. StarUML文件

16. 吾亦聊（Chatstyle）即时通讯聊天系统

#### 与计划的差别

1. 吾亦聊未具有房间系统。

2. 吾亦聊未完成用户系统。

3. 吾亦聊未完成管理员系统。

4. 吾亦聊未完善聊天系统。

5. 吾亦聊未较好的达到质量标准。

* 1. 里程碑和重大事件

1. 项目立项

2. 完成需求分析

3. 成员结束学习阶段，开始进入代码实践

4. 实现第一次前后端接口对接

5. 实现前端服务端和前端客户端互相通讯

* 1. 计划分析

项目进度仅进行到第一次迭代。

比计划延迟了两个月的时间，到交付期时比计划少了两次迭代。

主要原因是所有成员没有全身心投入，皆为非全职员工，被其他项目所耽搁。

* 1. 质量分析

项目没有达到质量需求，原因：

1. 测试使用的是基于无线网络的局域网，传输延迟较有线网络更高。

2. 网络传输方面采用项目组自研的解决方案，没有借鉴市场上比较成熟的方案。

3. 没有良好的开发用途的硬件设备支持。

4. 没有稳定强大的服务器做部署测试。

* 1. 工作量估计分析

实际项目开发和计划工作量出现偏差，实际项目开发工作量偏大：主要原因有两点：

1. 成员对开发技术并不熟悉，在排除错误的时候花费了大量时间。

2. 成员非全职工作，导致很多互相有依赖关系的工作不能顺畅完成。

* 1. 暂停与变更的需求分析

中止了关于家园系统的需求，原因：

1. 实现家园系统需要投入大量的精力对现有的系统架构进行升级。

2. 如果集中精力开发家园系统则会大大拖延交付期。

1. 开发方法和开发过程
   1. 开发环境和工具

开发平台：Windows 10

开发环境：Node V8，Unity引擎

开发工具：VSCode、Visual Studio 2017、Unity Editor

* 1. 技术途径

1. 前端使用Unity引擎进行开发。

2. 前端引入网络公开素材和插件美化视觉效果。

3. 后端采用Koa框架进行Web服务器构建。

4. 管理员端采用Vue框架进行开发。

1. 风险分析
   1. 风险1

描述：前后端对接的过程中，后端暴露的接口向前端妥协。

采取的规避措施：精简前端发送步骤和内容，一次请求尽可能涉及较少的数据表。

风险：导致数据库表项因为前端工程师的需求而变得非常冗余，出现不必要的字段。

* 1. 风险2

描述：Node和NPM版本不同，在下载依赖包时造成开发环境不同。

采取的规避措施：借助docker来进行开发，所有开发人员使用同样的docker镜像进行开发。

风险：不同的开发环境产出的产品与实际部署的环境出现不兼容的现象，影响系统功能。

1. 经验与教训
   1. 成功经验

使用docker统一了开发环境，使后端程序一直在Ubunutu16.04操作系统下运行。

成员使用了同样版本的MySQL，Node和NPM。

使用git和企业微信交流进度。

* 1. 教训

例会周期设置得太长，容易给成员造成过多的空档时间。

在检查周期内完成任务的成员，在下一次例会到来之前的时间没有被利用上。

1. 对未来项目开发的建议

对前端程序的系统架构进行重新设计，采取成熟的网络传输方案。

1. 整体评价

项目立项的设想比较合理，计划中的实现目标也在能力范围之内，但是在项目过程中缺乏有效的监管和控制手段，导致后期时间越来越紧促，开发人员的空闲时间也没有利用好。项目组的水平有待提升。