文件编号：MAC-SWE-TMP-06 密级：■ 保密 □ 通用

**体系结构文档**

# Preliminary Design Description

本程序属MAC公司所有，未经书面许可，

不得以任何形式复印或传播。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
|  |  |  |  | |  |
|  |  |  |  | |  |
|  |  |  |  | |  |
|  |  |  |  | |  |
| V1.0 | 楼康 | 孙潇桐 | 孙潇桐 | |  |
| 版 本 | 编写/修改 | 审 核 | 批 准 | | 修改原因 |
| 受控编号： | | 正文页数： | | 附件页数： | |
| 存储编号： | | 管理部门： | | 审查期： | |

**修 改 记 录**

|  |  |  |  |
| --- | --- | --- | --- |
| 日 期 | 版 本 | 修 改 内 容 | 修 改 人 |
| 2023/05/12 | <1.0.0> | 初步完成文档 | 楼康 |
| 2023/06/20 | <1.0.1> | 检查并完善文档内容 | 孙潇桐 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

文件编号： 密级：■ 保密 □ 通用

**体系结构文档**

|  |  |  |
| --- | --- | --- |
| 项目名称：OBattle *源神斗* | 拟制：楼康 | 审核：孙潇桐 |
| SQA：测试小组 | 批准：孙潇桐 |
| 项目代号：OBattle | 收文： | |
| 产品版本：1.0 | 抄送： | |

**1．0 引言**

**1．1 编写目的**

目的:

1. 沟通和传达：体系结构文档作为一个重要的沟通工具，用于将系统的设计和结构概念传达给项目团队、利益相关者和其他相关方。它帮助各方了解系统的整体架构以及各个组件之间的关系，促进共同理解和协作。
2. 文档化设计决策：体系结构文档记录了系统的设计决策、关键设计模式、技术选择和解决方案。它提供了对系统设计背后的原理和考虑因素的理解，以及设计决策的合理性和合规性。
3. 支持系统演化和维护：体系结构文档为系统的演化和维护提供了指导。它使开发团队和维护团队能够理解系统的整体结构和组件之间的依赖关系，从而更好地进行修改、增强和维护。
4. 帮助评估和分析：体系结构文档可以用于评估和分析系统的质量属性，例如性能、可靠性、安全性和可扩展性。它提供了一个基础框架，用于进行系统级别的评估和分析，帮助发现和解决潜在的问题和风险。
5. 支持决策制定：体系结构文档为项目管理和决策制定提供了信息和依据。它可以帮助项目管理人员和决策者了解系统的整体状况，评估项目进展和风险，并支持制定合适的决策和策略

总之：确保源神斗软件的高质量和可维护性。该文档将会提供源神斗软件的整体架构、各个子系统之间的关系与交互，以及子系统的详细设计、接口规范、数据模型、算法实现等详细说明，以便开发人员能够更好地理解系统的设计和需求，从而顺利地完成开发工作。

预期的读者包括以下人员：

1. 软件开发人员：他们需要了解联机斗地主软件的整体结构和各个子系统之间的关系与交互，以便开发出高质量、符合要求的代码。
2. 审计人员和监管机构：在某些情况下，体系结构文档可能需要被审计人员和监管机构审查和评估，以确保系统的合规性、安全性和稳定性。
3. 软件测试人员：在测试阶段，他们需要了解联机斗地主软件的整体结构和各个子系统之间的关系与交互，以便更有效地设计和执行测试用例。

**1．2 背景**

说明：

a)

名称：Origin War of Spirit 源神斗

b)

任务提出者:

孙潇桐，俞家宝

开发者:

孙潇桐，俞家宝，任庆华

测试小组：

孙潇桐，俞家宝，任庆华，纪国梁，楼康以及部分组外成员

文档小组：

楼康，孙潇桐

用户:

1.个人玩家：对游戏感兴趣的普通玩家.

2.团队玩家：例如朋友、家人、同事等，他们可以组队进行游戏。

3.竞技玩家：喜欢参加各种比赛和竞赛，挑战自我，展示自己的棋牌技能。

运行软件的计算站:

数据库服务器:负责存储和管理斗地主游戏数据和玩家信息

Web服务器：同步用户的应用内信息，控制用户行为。支持玩家的登录与注册。

**1．3 定义**

①软件体系结构：指软件系统的组织结构、组件和连接方式等方面的描述，用于指导软件的设计和开发。

②模块化：指将软件系统划分为独立的模块或组件，以便于管理和维护。

③分层结构：指将软件系统按照功能分为不同层次，每层之间通过明确定义的接口进行交互，以实现代码的复用和模块化。

④客户端/服务器结构：指软件系统将功能划分为客户端和服务器两部分，客户端负责显示界面和用户交互，服务器负责后台逻辑和数据管理。

⑤MVC架构：指软件系统采用Model-View-Controller的架构，将业务逻辑、用户界面和逻辑控制分别独立分层，以实现模块化和代码的复用。

⑥API：指软件系统提供的应用程序接口，用于与其他系统进行交互和数据传输。

⑦SDK：指软件系统提供的软件开发工具包，用于帮助开发者快速地开发和集成软件系统。

⑧GUI：指软件系统的图形用户界面，用于与用户进行交互和数据显示。

⑨JavaFX：Java的GUI构造套件

**1．4 参考资料**

①斗地主的基本规则

②原神人物建模

③JavaFX官方文档 <https://openjfx.io/javadoc/20/>

④Java8官方文档 <https://docs.oracle.com/javase/8/docs/api/overview-summary.html>

⑤MySQL官方文档<https://docs.oracle.com/javase/8/docs/api/overview-summary.html>

**2．0 总体设计**

**2．1 要求规定**

*用户可以通过源神斗软件注册登录，体验斗地主与原神元素的交融，还可以和好友联机开黑斗地主，并且还有聊天框增加互动性。前端GUI界面极富原神元素，次元感十足。*

**2．2 运行环境**

客户端：

Win7以上操作系统。

JRE1.8 虚拟机

服务端：

CentOS 7.0及以上

JDK18 及以上

MySQL数据库管理系统。

**2．3 结构**

给出系统的体系结构，即要说明系统被分解成哪些模块，这些模块之间如何交互。

前端:注重GUI展示渲染(聊天框,玩家游戏界面(出牌,胜负)).

后端:服务器处理游戏中的数据，例如牌组分发，聊天同步, 地主确定。同时提供登录和注册的接口，让前端可以有限制的访问数据库。

面向对象:卡牌(54张牌的图形展示)，玩家(3个用户的联机交互)

界面分离:登录界面，牌桌选择界面，游戏界面，结束界面。

前后端利用socket串联，通过传递我们定义好的结构化指令实现交互。

**2．4 功能需求与程序的关系**

本条用一张如下的矩阵图说明各项功能需求的实现同各组件的分配关系：

**GB 8567—88**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
|  | Beans | Scenes | GameRun | Utils |
| 卡牌对象 | √ |  |  |  |
| 玩家对象 | √ |  | √ |  |
| 牌组对象 | √ |  |  |  |
| 游戏界面 |  | √ | √ | √ |
| 登陆界面 |  | √ | √ | √ |
| 聊天框展示 |  | √ | √ | √ |
| 压牌逻辑控制 |  | √ |  |  |
| 前后端交互 |  |  |  | √ |

**3．0 接口设计**

**3．1 外部接口**

1. 登录

Input: username, password

Output: success/failure

2. 匹配玩家

Input: playerID

Output: success/failure

3. 游戏开始

Input: playerID

Output: success/failure

4. 出牌

Input: cardList, playerID

Output: success/failure

5. 特殊牌型（如飞机、炸弹）识别

Input: cardList

Output: success/failure

**3．2 内部接口**

*- 后端模块与数据库访问模块的接口：*

1. 添加玩家

Input: username, password

Output: success/failure

2. 查询玩家

Input: username

Output: player info

3. 删除玩家

Input: username

Output: success/failure

4. 添加游戏记录

Input: game info

Output: success/failure

5. 查询游戏记录

Input: game ID.

Output: game info

**4．0 系统数据结构设计**

**4．1 逻辑结构设计要点**

扑克类ADT设计:

①扑克花色

②扑克大小

③扑克中原神人物

玩家类ADT设计:

①玩家UID

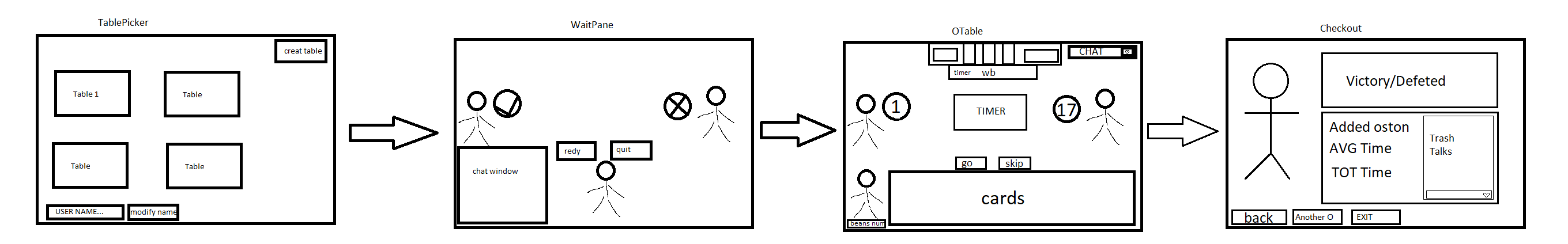
②地主判断位

③当前手牌

④是否轮到出牌

⑤玩家账号密码

界面前期设计:



**4．2数据结构（或表）与程序的关系**

**GB 8567—88**

|  |  |  |
| --- | --- | --- |
|  | User | Puke |
| 用户对象 | √ |  |
| 扑克对象 |  | √ |

**5．0 系统出错处理设计**

| **错误/故障情况** | **输出信息形式** | **含义** | **处理方法** |
| --- | --- | --- | --- |
| 网络连接中断 | "网络连接已中断" | 表示与服务器的网络连接断开 | 提供重连选项并尝试重新连接 |
| 服务器故障 | "无法连接到服务器" | 表示无法与游戏服务器通信 | 等待服务器恢复并尝试重新连接 |
| 数据库出错 | "数据库错误" | 表示无法读取或写入数据库 | 记录错误并尝试重新连接 |
| 游戏客户端异常退出 | "游戏客户端异常退出" | 表示玩家异常退出游戏 | 将玩家转化为机器人或终止游戏 |
| 等待超时 | "等待超时" | 表示等待其他玩家的响应超时 | 结束当前回合或从游戏中删除等待超时的玩家 |