《模拟桌游》需求说明书

**课题名称： 模拟桌游网页游戏**

小 组： 第1组

组 长： 谢子渊

组 员： 蔡祺辉

李欢

冉启航

指导教师： 杨波

**评阅成绩：**

提交报告时间：2025年3月26日

目录

[《模拟桌游》需求说明书 1](#_Toc581)

[一、 引言 3](#_Toc26319)

[二、 项目概述 3](#_Toc1823)

[三、 功能需求 4](#_Toc16638)

[四、 性能需求 5](#_Toc18710)

[五、 非功能需求 5](#_Toc19514)

[六、 接口需求 6](#_Toc28579)

[七、 技术架构 7](#_Toc22831)

[附录 需求跟踪列表 9](#_Toc11828)

###### **引言**

**1.1 编制目的**

本文档旨在详细说明《模拟桌游》项目的功能需求、非功能需求及相关技术架构，为开发团队提供明确的开发指导，并向用户说明项目的需求规格。

**1.2 定义、缩略语和参考文献**

**1.2.1 UNO**

UNO游戏开始时，每位玩家随机获得7张牌，牌面朝上置于手中，同时在牌堆上方翻开一张牌作为起始牌，形成公共牌堆。玩家需根据公共牌堆顶牌的颜色或数字匹配出自己的手牌，若无法匹配则需从公共牌堆抽牌。特殊功能牌包括“跳过”、“反转”、“+2”等，使用后会触发相应效果。当某位玩家出完手牌时，游戏结束，该玩家获胜。若游戏过程中无人出完牌，则继续进行，直至某位玩家获胜。

**1.2.2 海龟汤**

在海龟汤游戏中，玩家从一系列神秘的汤面中选择一个开始游戏，每个汤面背后都隐藏着一个独特的故事。游戏过程中，玩家通过提问来探索和还原汤面背后的故事内容，但提问必须遵循特定规则：只能针对汤面相关的内容提问，且不能直接询问汤底（即故事的核心谜底）。AI会根据问题与汤底的相关性，给出“是”“否”或“无关”的回答，帮助玩家逐步缩小范围、推理出完整的故事。当玩家成功还原出汤底时，游戏结束。

###### **项目概述**

**2.1 产品背景**

开发一款集成了UNO和海龟汤两款经典小游戏的多人在线网页游戏，支持多平台浏览器访问（Chrome、Firefox、Safari等）。

**2.2 核心功能**

**2.2.1 系统配置**

用户账户信息管理（注册/登录/账号信息/房间历史）。

游戏管理系统（辅助玩家进行游戏选择）。

**2.2.2 UNO**

房间创建（创建/加入房间）。

实时对战（标准UNO规则下的游戏/可选AI加入模式）。

**2.2.3 海龟汤**

下滑式选择列表（浏览/选择汤面）。

AI主持的游戏过程。

###### **功能需求**

**3.1系统配置**

**3.1.1 用户账户信息系统**

注册/登录：支持第三方（QQ）登录，建立账号进入游戏。阻止未登录的游客直接访问游戏系统。优先级低。

账号信息保存：对玩家的账号信息（账号、密码等）进行保密储存。优先级低。

个人信息修改：支持用户自定义昵称和头像。优先级低。

房间历史查询：用户可以查询自己曾进入过的游戏房间。优先级低。

**3.1.2 游戏管理系统**

游戏选择：用户登录后可以在首页选择要进入的游戏。优先级中。

**3.2 UNO**

**3.2.1 房间创建**

创建房间：用户可以作为房主创建新的游戏房间，并生成房间码和链接以邀请他人加入房间。优先级中。

加入房间：可以通过房间的房间码/分享房间链接加入房间。优先级中。

房间设置：房主可以确定以当前房间内玩家开始游戏。优先级高。

AI加入模式：房主可在房间中选择加入一位AI玩家参与游戏。优先级低。

**3.2.2 游戏流程**

**3.2.2.1 核心规则实现**

发牌：根据UNO标准规则与场上玩家数量，随机为每个玩家发牌。优先级高。

出牌验证：在游戏进行阶段，确保玩家的出牌符合规则，即花色/数字/功能牌与场上的牌堆顶相匹配。优先级高。

特殊牌处理：在玩家打出特殊牌时，根据标准规则改变游戏的相应状态，如出牌顺序、当前花色等。优先级高。

**3.2.2.2 游戏状态管理**

回合开始：提示当前出牌的玩家，激活相应玩家的手牌操作，检测可出的牌。优先级高。

出牌阶段：玩家可点击拖拽手牌以出牌。出牌应当符合出牌条件。玩家也可选择摸牌。出牌阶段有一定时间限制，超过时间将触发自动摸牌并结束回合。优先级高。

回合结束：更新牌堆顶显示，将回合结果广播给所有玩家。优先级高。

**3.2.2.3 胜负判定**

游戏结束：结束对局，弹出结算动画，提示胜利玩家。优先级高。

再来一局选项：结束游戏时提供“再来一局”选项，可以回到原房间。优先级低。

退出游戏选项：结束游戏时提供“退出游戏”选项，退出当前房间回到房间创建阶段。优先级中。

**3.3 海龟汤**

**3.3.1 汤面列表**

汤面推送：将后端储存的所有汤面以一定推送方式显示在列表上。显示内容包括汤面缩略文案、汤面类型。优先级低。

选择汤面：玩家可点击汤面进入详情页查看汤面信息。优先级低。

**3.3.2 游戏流程**

开始游戏：用户确定开始游戏后提示后端接入AI游戏相关内容，使用对话框开始游戏。优先级高。

用户提问：用户提问汤面相关内容，接入AI给出“是”“否”“无关”的回答。优先级高。

游戏结束：当游戏推进到用户基本还原汤底时，AI判断游戏已经结束，提示用户并给出完整汤底。优先级高。

###### **性能需求**

**4.1 响应速度**

客户端操作反馈小于0.5秒。

服务端逻辑计算延迟小于50毫秒。

**4.2 并发能力**

服务器支持100同时在线用户。

每秒处理20+游戏操作事件。

**4.3 资源占用**

浏览器内存占用小于300MB。

###### **非功能需求**

**4.2 可靠性需求**

断线恢复：网络中断10秒内重连可恢复游戏进度。

数据持久化：关键操作需99.9%成功写入。

服务可用性：全年可用性大于99%。

**4.3 安全需求**

数据安全：用户敏感信息传输加密。

DDoS防护：抵御10Gbps以下流量攻击。

**4.4 兼容性需求**

支持多平台浏览器访问。

**4.5 可维护性需求**

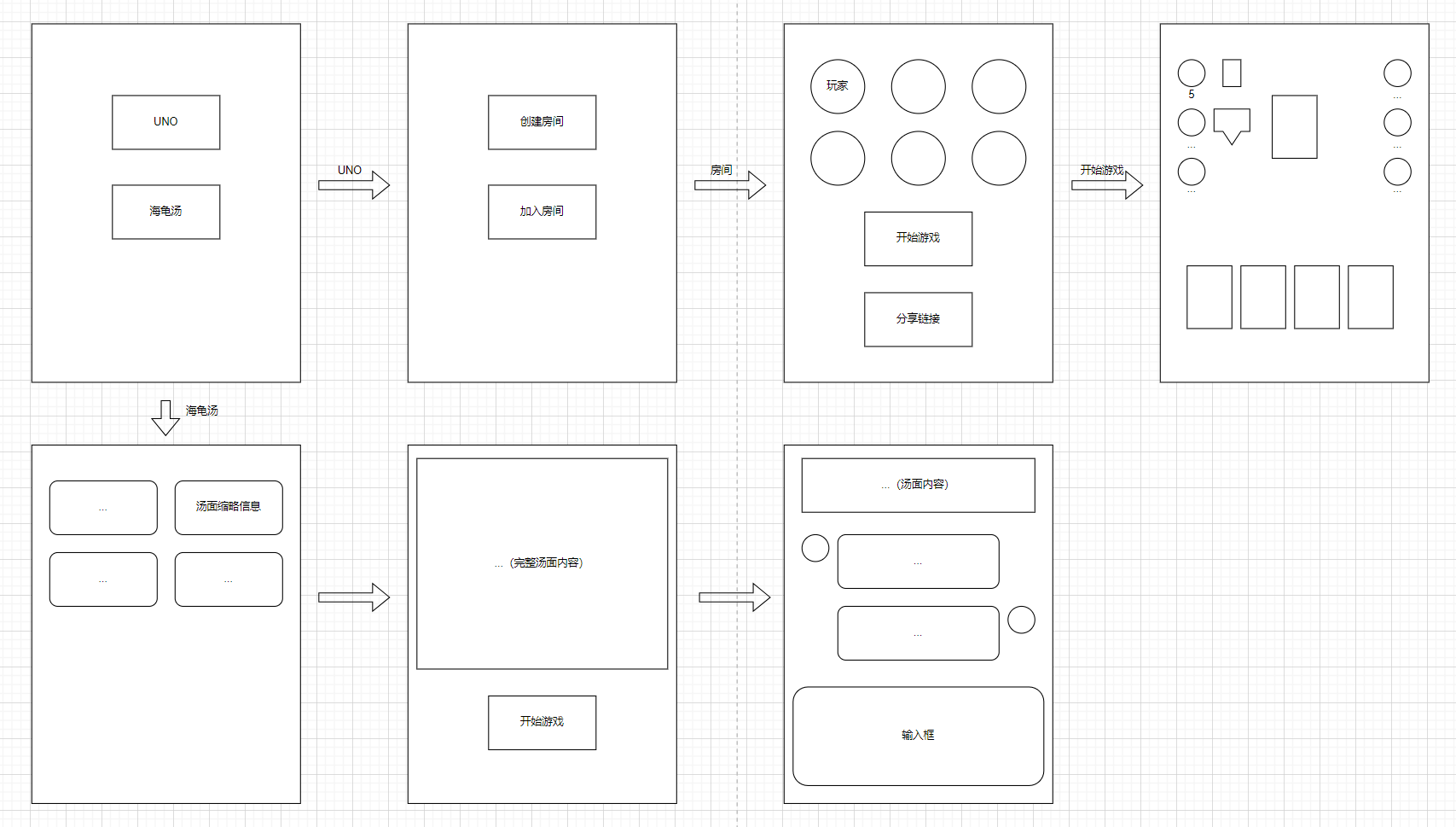
操作日志保留。

非数据库变更支持不停机更新。

###### **接口需求**

**6.1 UI接口**

原型图示意：



**6.2 外部接口**

**6.2.1 用户登录所用的第三方登录接口**

使用QQ等第三方登录API辅助进行用户认证。

与用户认证模块的接口：将第三方登录相关的认证信息发送给用户认证模块，用于验证用户信息。

**6.2.2 接入AI的接口**

使用deepseek R1模型等第三方AI API辅助UNO和海龟汤的游戏进行。

与UNO模块的接口：UNO模块选择AI玩家模式时，使用后端预设的提示词从AI获取AI玩家的游戏操作，并提交到UNO模块。

与海龟汤模块的接口：开始海龟汤游戏时，使用后端预设的提示词和相关信息提示AI主持游戏过程，并根据玩家的对话内容让AI给出回答和判断游戏是否结束。

**6.3 内部接口**

**6.3.1 用户认证模块与其他模块的接口**

与前端的接口：接收前端传来的用户登录请求，验证用户信息后，返回认证结果给前端。

与数据库的接口：查询、插入用户数据，确保用户信息的准确性和唯一性。

**6.3.2 游戏管理模块与其他模块的接口**

与前端的接口：接收前端的游戏选择请求，返回进一步的游戏信息给前端展示。

**6.3.3 UNO模块与其他模块的接口**

**6.3.3.1 房间相关模块**

**6.3.3.1.1 与前端的接口**

接收前端传来的房主创建房间的请求，将房间信息存储在服务器内存中。

接收前端传来的玩家加入房间的请求，通过WebSocket协议验证房间号的有效性，并将玩家加入对应的房间。

实时将房间状态推送给前端，确保所有玩家的界面同步。

**6.3.3.1.2 与通信模块的接口**

通过WebSocket协议建立和维护玩家与服务器间的连接。当玩家加入房间时，将玩家加入房间的信息传递给通信模块。

**6.3.3.2 游戏内相关模块**

与前端的接口：将游戏过程中的牌局状态、玩家手牌等信息发送给前端展示。

与通信模块的接口：将玩家的操作通过WebSocket实时同步给其他玩家。

**6.3.4 海龟汤模块与其他模块的接口**

与前端的接口：将汤面信息、玩家提问和AI回答等信息发送给前端展示。

与数据库的接口：从数据库获取汤面信息。

**6.3.5 数据库模块与其他模块的接口**

与用户认证模块的接口：提供用户信息的增删改查操作。

与海龟汤模块的接口：提供汤面数据的存储和查询操作。

###### **技术架构**

**7.1 前端**

**7.1.1开发框架**

Vue.js：作为核心框架，利用其简洁的语法和高效的响应式系统，快速构建用户界面。通过Vue的组件化开发模式，将页面拆分为多个可复用的组件，提高代码的可维护性和开发效率。使用Vue CLI或Vite进行项目初始化和构建，确保项目的规范性和可扩展性。

**7.1.2状态管理**

Vuex：负责管理应用的全局状态，如用户登录状态、游戏房间信息等。通过定义状态、mutations、actions和getters，实现状态的集中管理和有序变更。在用户登录、房间创建和游戏进行过程中，使用Vuex存储和更新相关数据，确保组件间的数据一致性。

**7.1.3样式框架**

SCSS：作为CSS的预处理器，提供变量、嵌套、混合等高级功能，简化样式编写和维护。通过定义全局变量和混合宏，统一项目的样式风格，提高样式的可复用性。针对不同组件和页面，创建相应的SCSS文件，进行模块化管理。

Tailwind CSS：作为辅助样式框架，提供丰富的预设样式类，快速实现响应式布局和基础样式。在页面布局、组件间距和字体调整等方面，利用Tailwind的类名直接应用样式，提高开发效率。同时，结合SCSS进行自定义样式扩展，满足项目的特定需求。

**7.1.4交互与动画**

Vue Transitions：利用Vue内置的过渡系统，为页面切换和组件状态变化添加动画效果。通过定义进入和离开的CSS动画或JS钩子，实现平滑的过渡效果，提升用户体验。

GSAP (GreenSock Animation Platform)：对于复杂的动画需求，如游戏中的卡牌翻转、角色移动等，使用GSAP进行实现。GSAP提供高性能的动画控制和丰富的动画功能，能够精确地控制动画的 timing、缓动函数和属性变化，创建出流畅且引人注目的动画效果。

**7.1.5 通信与实时工具**

Axios：作为HTTP客户端，处理与后端服务器的RESTful API通信。封装Axios请求，统一处理请求的拦截、响应和错误，确保数据交互的可靠性和规范性。在用户登录、房间操作和游戏数据同步等场景中，使用Axios发送请求和接收响应，实现前后端的数据交互。

Socket.io-client：用于实现与后端的实时通信，支持WebSocket和轮询两种方式。在游戏过程中，通过Socket.io-client与服务器建立连接，实时接收其他玩家的操作和游戏状态更新，确保游戏的同步性和实时性。在连接建立、消息接收和断线重连等方面，进行相应的处理和状态管理。

**7.1.6 测试工具**

Jest：作为单元测试框架，对项目的各个模块和组件进行单元测试。通过编写测试用例，验证函数、组件方法和Vuex actions等的正确性，确保代码的质量和稳定性。在开发过程中，定期运行测试用例，及时发现和修复潜在的问题。

Vue Test Utils：作为Vue官方的测试工具，用于测试Vue组件的行为和渲染结果。结合Jest使用，模拟组件的挂载、触发事件和检查DOM输出，验证组件在不同场景下的功能和交互是否符合预期。通过组件测试，提高组件的可靠性和可复用性。

Cypress：作为端到端测试工具，模拟用户在浏览器中的操作流程，对整个应用进行集成测试和验收测试。编写Cypress测试脚本，覆盖用户注册、登录、游戏创建、加入和进行等关键业务流程，确保应用在实际使用中的完整性和正确性。在项目发布前，运行Cypress测试，发现并解决可能存在的问题，提高应用的质量和用户体验。

**7.2 后端技术选型**

**7.2.1 语言与框架**

使用 Golang 语言，基于高性能框架Gin（轻量级 HTTP 框架）和 gorilla/websocket（实时通信）。

**7.2.2 数据库**

MySQL/PostgreSQL: 存储用户账户、房间历史、汤面数据等结构化数据。

Redis: 缓存会话信息（如 JWT Token）、房间状态、实时游戏数据（如玩家手牌、回合状态）。

消息队列: NSQ 或 RabbitMQ，用于异步处理高并发游戏事件（如出牌、提问）。

###### **附录 需求跟踪列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 需求编号 | 需求名称 | 描述 | 优先级 |
| SRS-0101 | 用户注册/登录 | 支持第三方（QQ）登录，建立账号进入游戏。阻止未登录的游客直接访问游戏系统。 | 低 |
| SRS-0102 | 账号信息保存 | 对玩家的账号信息（账号、密码等）进行保密储存。 | 低 |
| SRS-0103 | 个人信息修改 | 支持用户自定义昵称和头像。 | 低 |
| SRS-0104 | 房间历史查询 | 用户可以查询自己曾进入过的游戏房间。 | 低 |
| SRS-0105 | 游戏选择 | 用户登录后可以在首页选择要进入的游戏。 | 中 |
| SRS-0106 | 创建房间 | 用户可以作为房主创建新的游戏房间，并生成房间码和链接以邀请他人加入房间。 | 中 |
| SRS-0107 | 加入房间 | 可以通过房间的房间码/分享房间链接加入房间。 | 中 |
| SRS-0108 | 房间设置 | 房主可以确定以当前房间内玩家开始游戏。 | 高 |
| SRS-0109 | AI加入模式 | 房主可在房间中选择加入一位AI玩家参与游戏。 | 低 |
| SRS-0110 | 发牌 | 根据UNO标准规则与场上玩家数量，随机为每个玩家发牌。 | 高 |
| SRS-0111 | 出牌验证 | 在游戏进行阶段，确保玩家的出牌符合规则，即花色/数字/功能牌与场上的牌堆顶相匹配。 | 高 |
| SRS-0112 | 特殊牌处理 | 在玩家打出特殊牌时，根据标准规则改变游戏的相应状态，如出牌顺序、当前花色等。 | 高 |
| SRS-0113 | 回合开始 | 提示当前出牌的玩家，激活相应玩家的手牌操作，检测可出的牌。 | 高 |
| SRS-0114 | 出牌阶段 | 玩家可点击拖拽手牌以出牌。出牌应当符合出牌条件。玩家也可选择摸牌。出牌阶段有一定时间限制，超过时间将触发自动摸牌并结束回合。 | 高 |
| SRS-0115 | 回合结束 | 更新牌堆顶显示，将回合结果广播给所有玩家。 | 高 |
| SRS-0116 | 游戏结束 | 结束对局，弹出结算动画，提示胜利玩家。 | 高 |
| SRS-0117 | 再来一局选项 | 结束游戏时提供“再来一局”选项，可以回到原房间。 | 低 |
| SRS-0118 | 退出游戏选项 | 结束游戏时提供“退出游戏”选项，退出当前房间回到房间创建阶段。 | 中 |
| SRS-0119 | 汤面推送 | 将后端储存的所有汤面以一定推送方式显示在列表上。显示内容包括汤面缩略文案、汤面类型。 | 低 |
| SRS-0120 | 选择汤面 | 玩家可点击汤面进入详情页查看汤面信息。 | 低 |
| SRS-0121 | 开始游戏 | 用户确定开始游戏后提示后端接入AI游戏相关内容，使用对话框开始游戏。 | 高 |
| SRS-0122 | 用户提问 | 用户提问汤面相关内容，接入AI给出“是”“否”“无关”的回答。 | 高 |
| SRS-0123 | 游戏结束 | 当游戏推进到用户基本还原汤底时，AI判断游戏已经结束，提示用户并给出完整汤底。 | 高 |
| SRS-0201 | 响应速度 | 客户端操作反馈小于0.5秒。服务端逻辑计算延迟小于50毫秒。 | 中 |
| SRS-0202 | 并发能力 | 服务器支持100同时在线用户。每秒处理20+游戏操作事件。 | 中 |
| SRS-0203 | 资源占用 | 浏览器内存占用小于300MB。 | 中 |
| SRS-0301 | 断线恢复 | 网络中断10秒内重连可恢复游戏进度。 | 低 |
| SRS-0302 | 数据持久化 | 关键操作需99.9%成功写入。 | 低 |
| SRS-0303 | 服务可用性 | 全年可用性大于99%。 | 低 |
| SRS-0304 | 数据安全 | 用户敏感信息传输加密。 | 低 |
| SRS-0305 | DDoS防护 | 抵御10Gbps以下流量攻击。 | 低 |
| SRS-0306 | 兼容性需求 | 支持多平台浏览器访问。 | 低 |