

文件编号：

项目名：小学生速算训练营

需求与功能设计规格书

作者：夏天、宋雨涵、许圆圆、欧阳燕璞

部门：AAAA

时间：2026 年 6 月 8 日

改版履历

版数	年月日	改版内容	承认	查阅	作成	备注
V0.1	2026 年 6 月 5 日	创建文档，完成前言、开发概要、外部接口需求草案	刘文洁	团队	团队	基于计划书 V0.1
V0.3	2026 年 6 月 6 日	新增实现方式章节，补充非功能需求和使用方法	刘文洁	团队	团队	基于计划书 V0.3
V0.8	2026 年 6 月 7 日	增加附录，完善注意限制事项和特记事项	刘文洁	团队	团队	基于计划书 V0.8
V1.0	2026 年 6 月 8 日	全文审校定稿，格式统一，交付正式版	刘文洁	团队	团队	基于计划书 V1.0

目录

1. 前言	1
1.1. 输入文档	1
1.2. 相关文档	1
1.3. 用语/术语定义	1
2. 开发概要	2
2.1. 开发的目的和背景	2
2.2. 工作&运行环境	2
2.3. 系统构成	3
2.3.1 系统整体架构	3
2.3.2 系统预期构成	3
2.3.3 开发功能的系统整体位置	4
2.3.4 与其他运行程序的关联关系	4
2.4. 功能整体图	5
2.4.1 功能模块层次图	5
2.4.2 补充事项	5
2.5. 用户分类特征	6
2.5.1 用户一览表	6
2.5.2 各用户角色职责	7
2.5.3 用户与功能模块对应关系	8
2.5.4 用户场景示例	9
2.5.5 补充说明	9
3. 外部接口需求	10
3.1. 用户接口	10
3.2. 硬件接口	14
3.3. 软件接口	14
3.4. 通讯接口	15
4. 实现方式	15
4.1. 学习模式功能实现	15
4.1.1 功能概要	15
4.1.2 功能详细	16
4.1.3 相关功能	18
4.1.4 用户接口	19
4.2. 对战模式功能实现	20
4.2.1 功能概要	20
4.2.2 功能详细	21
4.2.3 相关功能	24
4.2.4 用户接口	25
4.3. 全局与基础功能	26
4.3.1 功能概要	26
4.3.2 功能详细	27
4.3.3 相关功能	28
4.3.4 用户接口	29
5. 非功能需求	29

5.1. 权限需求	29
5.2. 兼容互换性	30
5.3. 安全性	30
5.4. 健壮性	30
5.5. 使用性（操作性）	30
5.6. 效率性（性能）	31
5.7. 维护性	32
5.8. 移植性	32
5.9. 用户文档	32
5.10. 其他	32
6. 使用和操作方法	32
6.1. 环境设定	32
6.2. 使用方法	33
6.2.1 系统启动	33
6.2.2 学习模式使用方法	33
6.2.3 对战模式使用方法	33
6.2.4 系统退出	34
7. 注意限制事项	34
7.1. 制约 / 限制事项	34
7.2. 假定事项	34
7.3. 特记事项	34
8. 附录	35
8.1. 业务规则	35
8.2. 术语表	35
8.3. 数据流图	35
8.4. 数据字典	37

1. 前言

本文档是《小学生速算训练营》项目的需求与功能设计规格书，旨在将开发计划书中定义的功能需求进一步细化，明确系统的外部接口、功能实现方式、非功能需求及操作方法，为后续编码实现和测试验收提供依据。

1.1. 输入文档

编写本需求与功能设计规格书时，主要参考以下输入文档：《小学生速算训练营 开发计划书》（版本 V0.1，2026 年 6 月 2 日）。该文档明确了项目的开发目的、功能概要、作业范围、技术栈、开发日程及成员分工，是本设计书的核心输入依据。

1.2. 相关文档

- 与本设计规格书内容相关联的其他文档如下：
- （1）《小学生速算训练营 开发计划书》（版本 V0.1，2026 年 6 月 2 日）。该文档明确了项目的开发目的、功能概要、作业范围、技术栈、开发日程及成员分工，是本设计书的核心输入文档之一。
 - （2）《小学生速算训练营 Review 记录表》。该表格用于记录项目各阶段（需求、设计、编码、测试）的评审活动。其中包含数据分类统计表和各工程作入 BUG 摘出统计表。评审发现的缺陷将作为 Bug List 的输入来源。
 - （3）《小学生速算训练营 Bug List》。该文档用于记录从 Review、功能测试等活动中发现的所有缺陷，包含缺陷编号、状态、报告者、发生模块、重要度、问题点、再现步骤、bug 原因、修改处理等字段。Bug List 中的缺陷修复情况将最终反映到功能测试报告书中。

1.3. 用语/术语定义

本文档中使用的核心术语定义如下：

术语	定义
学习模式	单人练习模式，用户可选择难度和题型，逐题作答，系统给予即时反馈，支持生命值系统和最终评分。
对战模式	本地双人对战模式，两名玩家通过键盘按键（S/L）抢答，每道题有 10 秒倒计时，答对得分，先达到胜场者获胜。
生命值	学习模式中用户答错题目时的“机会”数量。Easy/Medium 难度初始 3 条命，Hard/Unbeatable 难度初始 2 条命，答错扣 1 条，归零则本轮失败。
难度档位	共四档：Easy（简单）、Medium（中等）、Hard（困难）、Unbeatable（极限），影响题目数值范围、题目数量及初始生命值。

术语	定义
题型	本系统支持的 7 种运算类型：加法 (+)、减法 (-)、乘法 (×)、除法 (÷)、幂运算 (a^n)、平方根 ($\sqrt{\quad}$)、随机混合。
批量校验	学习模式中，用户完成当前轮次所有题目后，一次性提交全部答案，系统给出正误统计并标红错题、给出提示。
抢答互斥锁	对战模式中，当某一玩家正在作答时，另一玩家的抢答按键无效，确保同一时间只有一人答题。
GitHub Pages	GitHub 提供的免费静态网站托管服务，本项目将部署于此，用户通过网址访问，无需下载安装。
MVP 版本	最小可行产品版本，包含随机出题、判断对错、计分、难度切换、对战模式基础抢答等核心功能。

2. 开发概要

2.1. 开发的目的是背景

随着小学教育信息化进程的推进，越来越多的学习场景从纸质练习转向在线平台。然而，市面上现有的数学练习应用普遍存在以下问题：广告过多干扰学习、功能冗余导致操作复杂、缺乏即时反馈机制、以及不具备本地双人互动功能。本项目旨在开发一款界面简洁、即时反馈、支持双人本地对战的在线速算练习平台，面向中小学生及有基础运算练习需求的普通用户，帮助其在趣味中提升四则运算、幂运算和平方根的计算速度与准确率。

2.2. 工作&运行环境

(1) HW

客户端：最低配置为 CPU 1.5 GHz 双核、内存 4 GB、硬盘 8 GB 可用空间。

推荐配置为 CPU 2.0 GHz 四核、内存 16 GB、硬盘 16 GB (SSD 更优)、机器种类为 PC/AT 兼容机、Mac、Chromebook 等支持现代浏览器的设备。

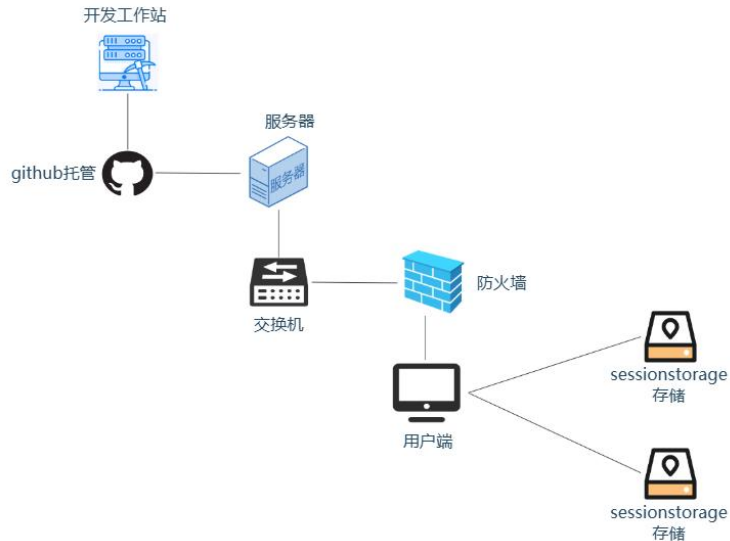
服务器：无需专用服务器，仅使用 GitHub Pages 提供静态托管。

(2) SW

操作系统：Windows 10 / 11、macOS Ventura+、Ubuntu 22.04+、ChromeOS

浏览器：Chrome 90+、Edge 90+、Firefox 88+、Safari 14+

系统部署如下图所示：

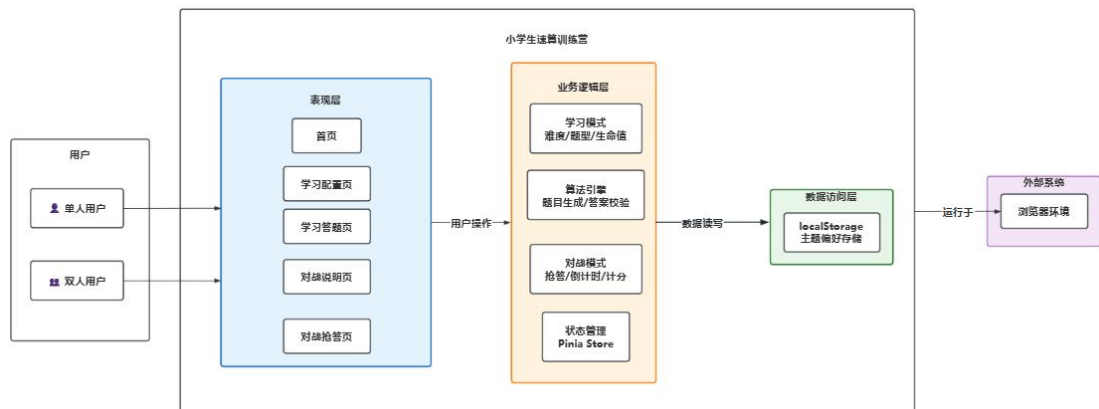


2.3. 系统构成

2.3.1 系统整体架构

本系统为纯前端 Web 应用，采用三层架构设计，从上到下分为表现层、业务逻辑层和数据访问层。各层职责明确，下层为上层提供服务，上层通过调用下层接口实现业务功能。系统不依赖任何后端服务器或外部服务，所有功能均在用户浏览器本地完成。

系统整体架构图如下：



2.3.2 系统预期构成

本产品预期的系统构成包括以下五个核心模块：

模块编号	模块名称	构成内容	职责说明
C1	项目骨架	Vite 配置、Vue Router 路由、Pinia 状态树、全局样式	提供系统运行基础框架
C2	学习模式主框架	难度/题型配置页、答题组件、生命值系统	实现单人练习全流程

模块编号	模块名称	构成内容	职责说明
C3	运算逻辑与算法引擎	7 种题型生成、难度参数映射、答案校验、防重复出题	提供核心算法能力
C4	双人对战模块	规则说明页、抢答界面、键盘监听、倒计时、计分系统	实现双人对战全流程
C5	质量保证	Vitest 单元测试	保障算法正确性

2.3.3 开发功能的系统整体位置

本系统是一个完整的、独立运行的前端 Web 应用，不依赖于任何其他系统或服务。从功能定位来看，本系统处于用户终端与浏览器引擎之间的位置：

- 用户终端：用户通过 PC 的浏览器访问本系统
- 浏览器引擎：负责解析 HTML/CSS/JavaScript，执行本系统代码，渲染界面
- 本系统：运行在浏览器沙箱环境中，调用浏览器提供的 API（如 localStorage 存储主题偏好）

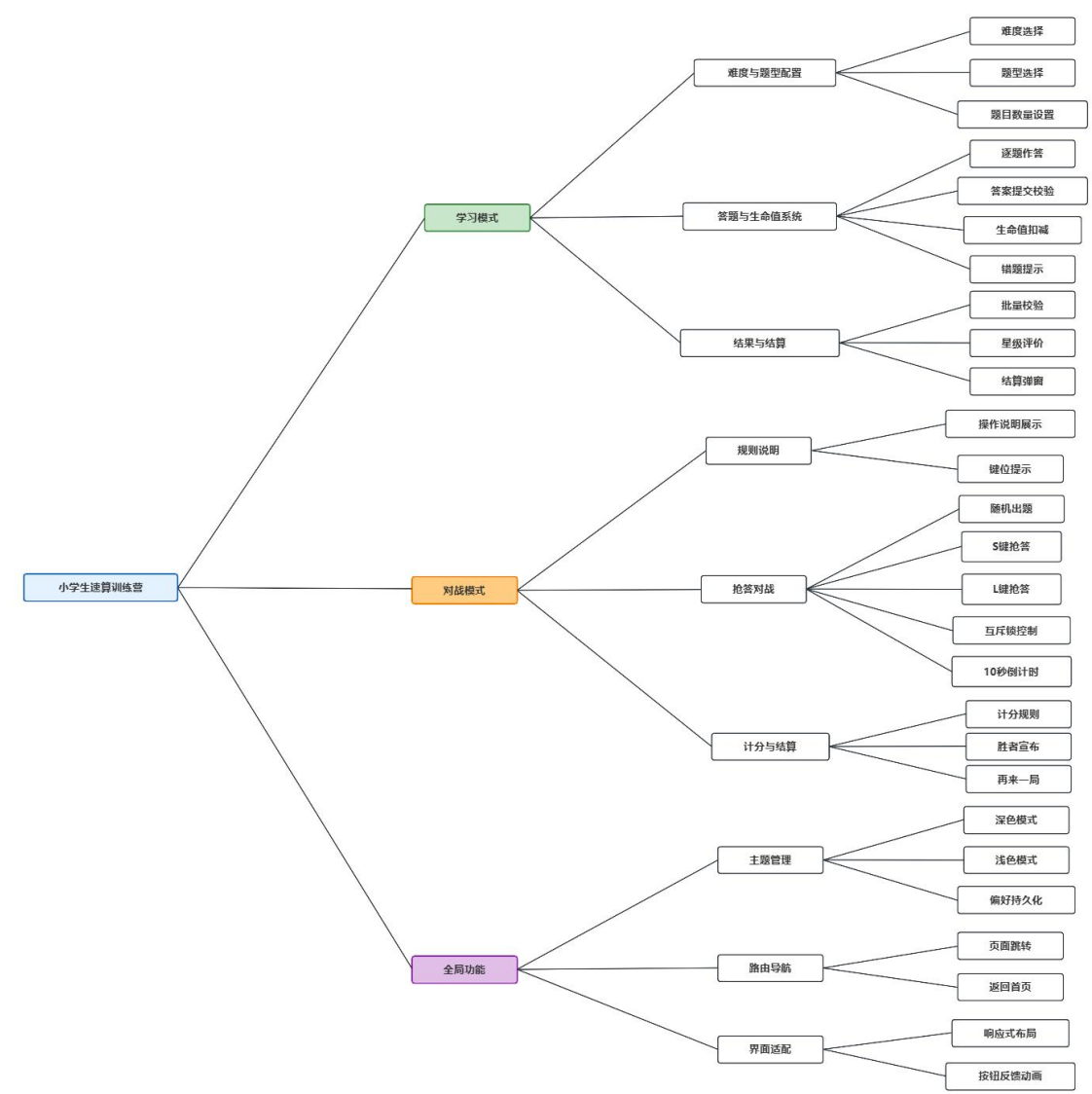
2.3.4 与其他运行程序的关联关系

本系统为独立运行的 Web 应用，与其他运行程序的关联关系如下表所示：

关联类型	关联对象	关联方式	说明
运行时依赖	现代浏览器	HTTP/HTTPS 协议	需 Chrome 90+ / Firefox 88+ / Safari 14+ / Edge 90+
构建时依赖	Node.js 18+/20+	npm 脚本	仅在开发和构建时需要，运行时不需要
本地存储	localStorage	浏览器 API	仅存储主题偏好，不上传任何服务器
外部 API	无	-	所有题目均为前端随机生成
数据库	无	-	不连接任何数据库
第三方服务	无	-	无广告、无追踪、无 Cookie

2.4. 功能整体图

2.4.1 功能模块层次图



2.4.2 补充事项

(1) 难度参数映射规则

难度	数值范围	生命值	特殊规则
Easy	0-20 加减，5×5 乘，除数≤5	3 条	减法结果非负，除法整除
Medium	0-50 加减，10×10 乘，除数≤10	3 条	减法结果非负，除法整除
Hard	0-100 加减，15×15 乘，除数≤20	2 条	可带余数
Unbeatable	负数允许，20×20 乘，除数≤30	2 条	负数结果允许，有余数

(2) 题型生成规则

题型	生成规则
加法	两数相加，和不超过范围上限
减法	被减数 $\geq$ 减数，Unbeatable 允许负数
乘法	两因数相乘，积不超过范围上限
除法	被除数 $\div$ 除数，优先整除
幂运算	底数 ^{指数} ，结果 ≤ 10000
平方根	被开方数为完全平方数，结果为整数
随机混合	从上述 6 种题型中随机选择

(3) 对战模式公平性设计

- 题目范围固定，不随难度变化，避免双方难度差异
- 每局题目数量在 5-7 题间随机，不可预知
- 抢答采用互斥锁，同一题仅首次按键有效
- 超时或答错均由对方得分，确保有胜负

(4) 数据存储说明

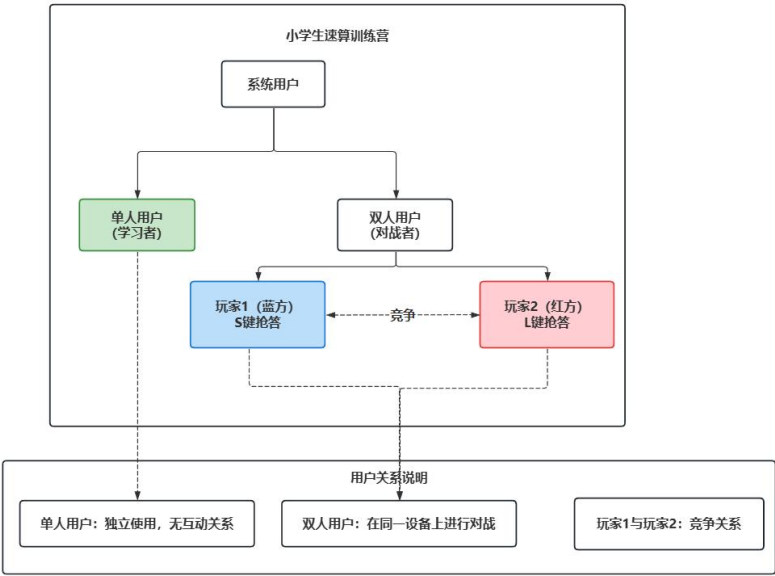
本系统仅使用浏览器 localStorage 存储用户主题偏好（深色/浅色模式），不收集任何个人信息，不使用 Cookie，不向任何服务器发送数据。

2.5. 用户分类特征

2.5.1 用户一览表

编号	用户	别名	说明
U1	单人用户	学习者	独自使用本系统进行速算练习的用户
U2	双人用户 (玩家 1)	蓝方 玩家	参与本地双人对战的玩家 1，使用键盘 S 键抢答
U3	双人用户 (玩家 2)	红方 玩家	参与本地双人对战的玩家 2，使用键盘 L 键抢答

用户角色关系图如下所示：



2.5.2 各用户角色职责

(1) 单人用户（学习者）

独自使用本系统进行速算练习的用户，可以是中小學生或成年人。用户根据自身水平选择难度和题型，系统随机生成题目，用户逐题作答并获得即时反馈。全部答对可获得星级评价，答错过多则生命值归零结束本轮。

主要职责：

- 选择适合自己的难度档位（Easy/Medium/Hard/Unbeatable）
- 选择题型组合（加法/减法/乘法/除法/幂运算/平方根/随机混合）
- 认真作答每道题目
- 查看答题结果和星级评价

(2) 双人用户（玩家 1 / 蓝方玩家）

参与本地双人对战的玩家，使用键盘 S 键进行抢答。两名玩家在同一设备上进行的对战，通过抢答和答题比拼计算速度和准确率。

主要职责：

- 阅读并遵守对战规则
- 在题目出现后快速按下 S 键抢答
- 抢答成功后限时输入答案
- 公平参与竞争

(3) 双人用户（玩家 2 / 红方玩家）

参与本地双人对战的玩家，使用键盘 L 键进行抢答。与玩家 1 形成竞争关系，共同完成 5-7 轮对战。

主要职责：

- 阅读并遵守对战规则
- 在题目出现后快速按下 L 键抢答
- 抢答成功后限时输入答案
- 公平参与竞争

2.5.3 用户与功能模块对应关系

功能模块	子功能	单人用户	双人用户（玩家 1）	双人用户（玩家 2）
学习模式	难度选择	●	-	-
	题型选择	●	-	-
	逐题作答	●	-	-
	答案提交校验	●	-	-
	生命值扣减	●	-	-
	错题提示	●	-	-
	星级评价	●	-	-
	结算弹窗	●	-	-
对战模式	操作说明展示	-	●	●
	键位提示	-	●	●
	随机出题	-	●	●
	S 键抢答	-	●	-
	L 键抢答	-	-	●
	互斥锁控制	-	●	●
	10 秒倒计时	-	●	●
	计分规则	-	●	●
	胜者宣布	-	●	●
全局功能	深色模式	●	●	●
	浅色模式	●	●	●
	偏好持久化	●	●	●
	页面跳转	●	●	●
	返回首页	●	●	●

功能模块	子功能	单人用户	双人用户（玩家1）	双人用户（玩家2）
	响应式布局	●	●	●
	按钮反馈动画	●	●	●

图例：● 表示该用户类型可使用此功能。

2.5.4 用户场景示例

场景一：小学生课后练习（U1 单人用户）

小明是一名小学三年级学生，打开“小学生速算训练营”，选择“学习模式”，将难度设为“Medium”，题型勾选“加法”和“乘法”，点击开始挑战。系统随机生成 10 道题目，小明逐题作答，答对时自动跳转下一题，答错时提示“再大一点~”并扣除 1 点生命值。最终小明答对 8 题，获得 2 星评价。

场景二：课间双人对战（U2 和 U3 双人用户）

课间休息时，小华和小丽想比一比谁算得更快。两人打开“对战模式”，阅读规则后点击开始对战。系统随机生成 6 道题目。第一题“ $15 \times 3 = ?$ ”，小华抢按 S 键抢答成功，输入“45”正确，小华得 1 分。第二题小丽抢答成功并答对。经过 6 轮比拼，小华以 4:2 获胜，系统弹窗宣布“蓝方玩家胜利！”。

2.5.5 补充说明

用户特征总结：

特征维度	单人用户	双人用户
年龄范围	8-15 岁中小學生为主	8-15 岁中小學生为主
使用场景	独自练习	两人在同一设备上对战
使用频率	日常练习	社交性使用
计算机操作水平	基础水平	基础水平
核心需求	提升计算速度和准确率	趣味竞赛、与同伴比拼
关注点	难度是否合适、反馈是否及时	抢答是否公平、胜负判定是否明确

系统不包含的功能：

- 本系统不设登录功能，不区分个人账号
- 不收集任何用户个人信息
- 不支持在线多人对战（仅本地双人）
- 不支持历史记录和排行榜

3. 外部接口需求

3.1. 用户接口

本系统共包含四个主要界面，分别对应首页、学习配置页、学习模式答题页和对战模式答题页。具体用户接口设计如下：

（1）首页

首页界面如下图所示：



首页为用户提供两个主要入口按钮：“学习模式”和“对战模式”。用户通过鼠标点击任一按钮后，系统将跳转至对应的功能页面。首页不涉及用户输入数据的收集，仅作为模式选择的导航入口。

用户输入	系统输出
点击“学习模式”按钮	跳转至学习配置页
点击“对战模式”按钮	跳转至对战模式规则说明页

（2）学习配置页

学习配置页界面如下图所示：

 学习模式配置

选择你需要的学习配置，开启训练之路

难度选择

Easy

Medium

Hard

Unbeatable

题型选择

随机混合

加法

减法

乘法

除法

幂运算

平方根

返回首页

开始挑战

- 学习配置页用于设置本轮练习的参数。用户可在该页面完成以下配置：
- （1）难度选择：从 Easy（简单）、Medium（中等）、Hard（困难）、Unbeatable（极限）四档中选择其一，选中项以高亮样式反馈。
 - （2）题型选择：从加法、减法、乘法、除法、幂运算、平方根、随机混合七种题型中勾选需要练习的类型。
 - （3）页面底部提供“返回首页”和“开始挑战”两个按钮。

用户输入	系统输出
点击难度档位选项	当前选中难度高亮显示
勾选/取消勾选题型	当前题型状态更新
点击“返回首页”按钮	跳转回首页
点击“开始挑战”按钮	跳转至学习模式答题页，并根据所选难度和题型生成题目

（3）学习模式答题页

学习模式答题页界面如下图所示：



学习模式答题页是用户进行单人练习的核心界面。页面布局及交互说明如下：

- （1）顶部区域：左侧显示当前生命值，右上角显示退出按钮
- （2）中部区域：显示当前题目序号，以进度条形式表示答题进度
- （3）底部区域：显示题目内容，提供答案输入框和“提交”按钮。用户输入答案后点击提交，系统即时判断正误并给出反馈

用户输入	系统输出
在输入框中输入答案	输入框显示用户键入的数字
点击“提交”按钮	判断答案正误；正确时绿色提示并自动跳至下一题；错误时红色提示并给出“更大/更小”提示，同时扣除一点生命值
生命值归零	弹出失败弹窗，提供“返回配置页”按钮
全部题目答完	弹出评分弹窗，展示最终得分和题目回顾，提供“返回配置页”按钮
点击右上角退出键	直接返回学习配置页

（4）对战模式答题页

对战模式答题页界面如下图所示：



对战模式答题页支持两名玩家在同一设备上进行抢答比赛。页面布局及交互说明如下：

- （1）顶部区域：左右两侧分别显示“玩家 1”和“玩家 2”的当前得分，右上角显示退出
- （2）中部区域：居中显示当前题目内容。
- （3）底部区域：提供“抢答”按钮（作为键盘抢答的备用方案），以及倒计时进度条。
- （4）抢答机制：玩家 1 按 S 键抢答，玩家 2 按 L 键抢答。抢答成功后，系统弹出输入弹窗，同时激活 10 秒倒计时进度条。另一方抢答键被禁用，直至当前答题结束。

用户输入	系统输出
玩家 1 按 S 键	激活玩家 1 的输入弹窗，启动 10 秒倒计时，禁用玩家 2 的抢答键
玩家 2 按 L 键	激活玩家 2 的输入弹窗，启动 10 秒倒计时，禁用玩家 1 的抢答键
点击“抢答”按钮	激活当前回合玩家的输入弹窗
在弹窗中输入答案并提交	判断答案正误：答对则当前玩家得 1 分，答错则对方得 1 分；自动跳至下一题
倒计时归零	当前题超时判负，对方得 1 分，自动跳至下一题
全部题目答完	弹出结算弹窗，宣布胜者，提供“返回首页”按钮
点击右上角退出键	直接返回学习配置页

3.2. 硬件接口

本系统不涉及与任何外部硬件设备的直接数据交换。所有硬件交互均通过浏览器层间接完成，具体包括：

硬件设备	描述
键盘输入	系统监听键盘按键事件（S 键、L 键、Enter 键），由浏览器 DOM 事件 API 提供，本系统不直接操作硬件
显示输出	系统通过浏览器渲染引擎将界面输出至显示器，本系统不直接控制显示硬件

3.3. 软件接口

（1）与外部组件的连接

本系统为纯前端静态 Web 应用，不依赖任何后端服务或外部 API，无需配置软件接口。具体说明如下：

项目	说明
外部 API	无。所有题目均为前端随机生成，不调用任何第三方接口
数据库	无。不连接任何数据库，仅使用浏览器 localStorage 存储用户主题偏好
消息服务	无
第三方服务	无。不接入广告、支付、推送等任何第三方服务

运行时依赖： 仅需现代浏览器（Chrome 90+ / Firefox 88+ / Safari 14+ / Edge 90+），无需安装任何其他软件。

（2）内部组件之间的通信

接口名	描述
视图层 ↔ 状态层	Vue 组件通过 Pinia 的 state/actions 读写全局状态，实现响应式更新
视图层 ↔ 逻辑层	组件调用算法函数生成题目、校验答案，通过 import/export 完成
对战互斥锁	一方抢答后，Pinia state 中的锁状态更新，另一方按键被禁用

（3）组件之间共享的数据

数据名	描述
主题偏好	深色/浅色模式，通过 localStorage 持久化，所有组件共享

数据名	描述
学习模式状态	包含难度、题型、题目列表、当前题号、生命值、得分
对战模式状态	包含玩家 1/2 得分、当前题目、抢答锁、倒计时剩余秒数

3.4. 通讯接口

接口名	描述
HTTPS	本系统通过 HTTPS 协议从 GitHub Pages 加载所有静态资源（HTML、CSS、JavaScript、图片等），要求 TLS 1.2 或更高版本，确保传输过程中数据不被窃听或篡改
无数据传输	本系统不向任何服务器发送用户数据。所有用户输入（题目答案、难度选择、主题切换等）仅在浏览器本地处理，不经过网络传输，因此不涉及消息格式定义
无外部 API	本系统不调用任何第三方 API（如支付、登录、社交分享、数据库等），无电子邮件、电子表格等外部通信需求
无实时通信	本系统不涉及 WebSocket、SSE、WebRTC 等实时通信协议。对战模式为本地双人对战，所有交互（抢答、倒计时、计分）均在单台设备的浏览器内完成，不通过网络同步数据
数据传输速率	不适用。本系统仅在首次加载时从服务器获取静态资源，后续运行过程中无持续数据传输。正常网络环境下，首次加载时间不超过 3 秒
同步机制	不适用。本系统无客户端与服务器之间的数据同步需求

4. 实现方式

4.1. 学习模式功能实现

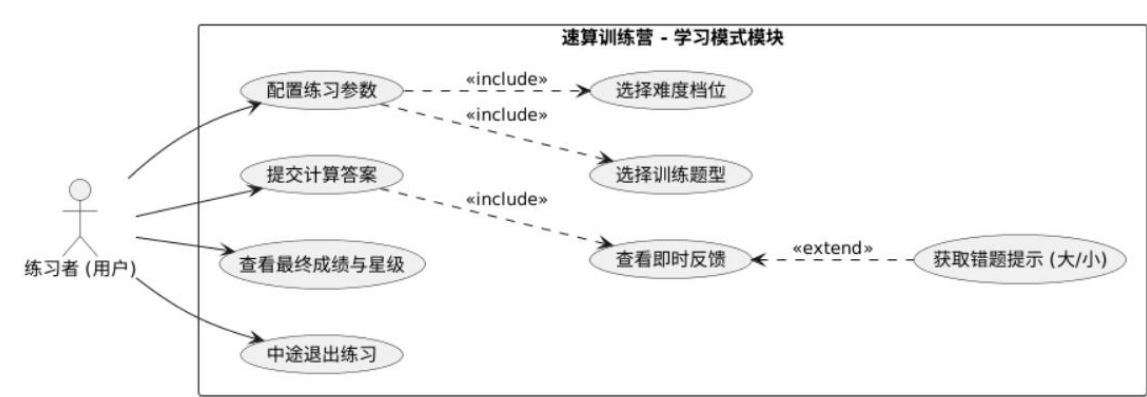
4.1.1 功能概要

学习模式面向单人速算练习场景，用户选择题型和难度后，系统生成题目列表，并逐题答题、判断正误、显示反馈及生命值变化，最终显示结果。用例表如下：

项目	内容
用例名称	学习模式训练
参与者	练习者（学生/普通用户）
简要说明	用户选择难度和题型，逐题进行口算答题。系统提供即时校验与错题提

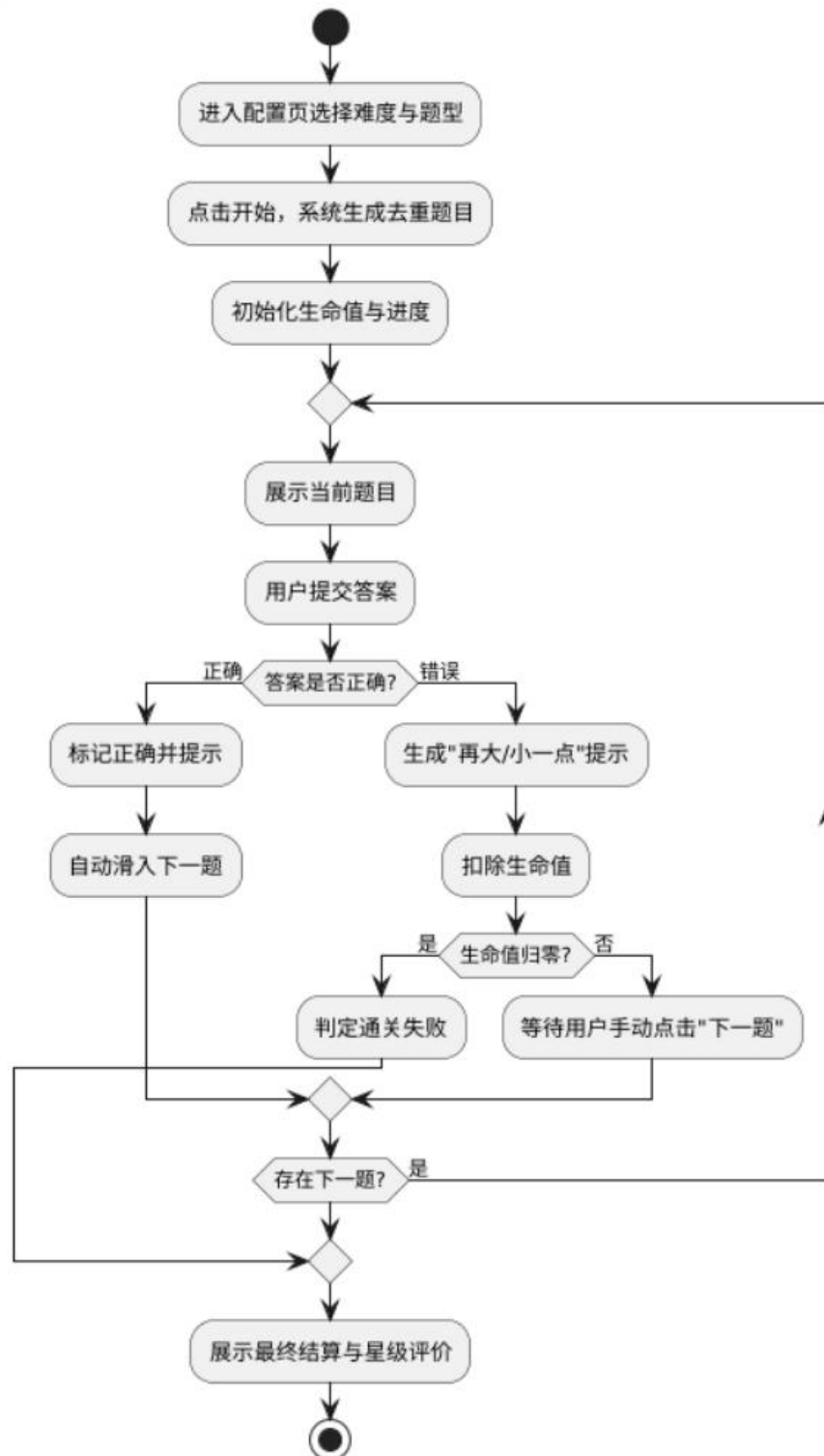
	示，答错扣除生命值，直至完成全部题目或生命值归零时弹出结果结算窗口。
前置条件	练习者已进入学习模式配置页
备注与异常流	1. 空值提交：用户未输入任何内容时尝试点击提交，系统无响应（前端拦截提交操作），不扣除生命值。 2. 非法输入拦截：前端输入框直接限制仅允许数字输入，用户无法键入非数字字符，从源头规避非法数据。 3. 生命值归零：练习者答错导致生命值耗尽，系统提前终止本次练习并弹出失败结算弹窗。 4. 中途退出：练习者在答题过程中随时可点击“退出”按钮返回配置页，当前答题进度不予保存。
后置条件	弹出结果窗口，展示本次练习的得分和通关星级评价

学习模式用例图如下：

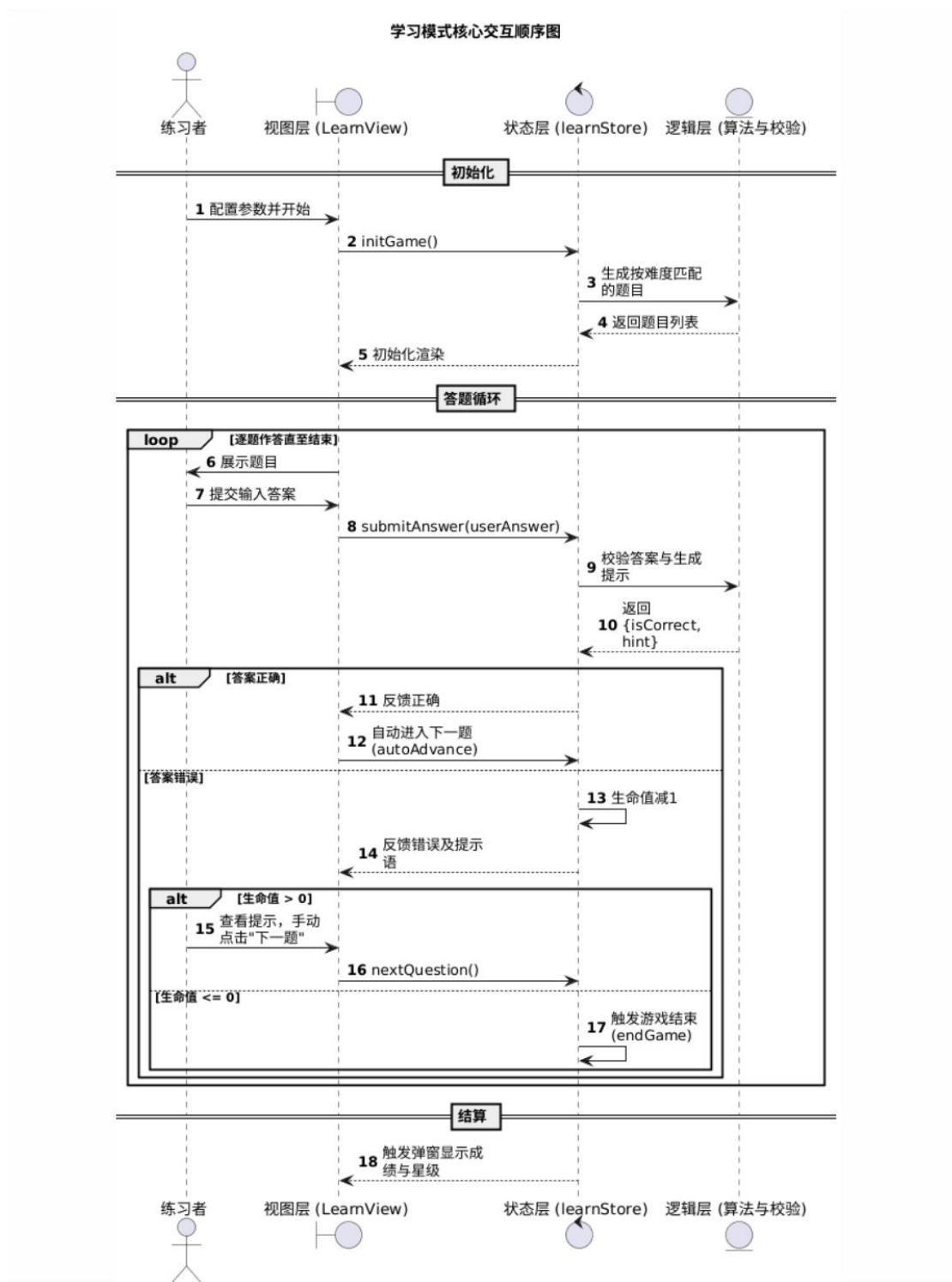


4.1.2 功能详细

学习模式的生命周期遵循“状态初始化 -> 题目循环作答 -> 即时校验反馈 -> 终止条件触发结算”的控制流。用户点击开始后，系统利用前端算法引擎生成去重数组并重置指针。在答题循环中，系统根据用户的输入进行实时对错判断，并动态调整生命值。具体的底层控制流向与条件决策节点如以下业务活动图所示：



在具体业务执行中，系统采用响应式状态层（Store）与逻辑层（Composable）解耦的协作机制，以确保 UI 渲染与底层算法低耦合。用户在视图层（LearnView）的每一次提交，都会通过状态层分发至逻辑层进行算式比对与动态文本生成。若答错则通过全局状态自减生命值，直至触发游戏结束判定并唤起结算弹窗。各核心运行对象之间的具体消息协作与交互逻辑如以下时序顺序图所示：

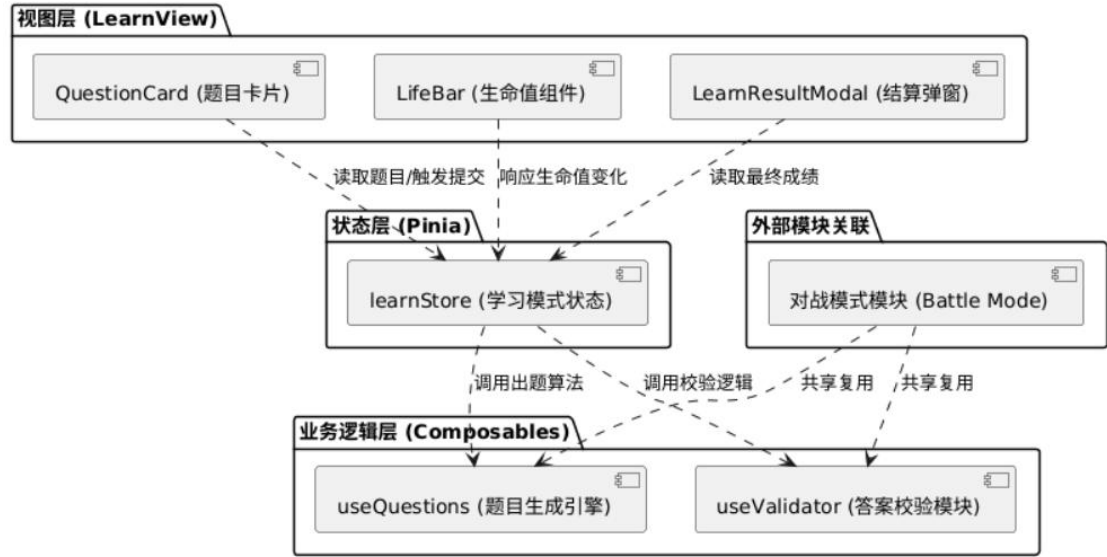


4.1.3 相关功能

学习模式功能的完整实现采用高度组件化的设计思想，确保了 UI 渲染与业务逻辑的低耦合、高内聚。该模块并不孤立，其核心的题目生成与校验逻辑与“本地双人对战模式”存在直接的复用关系，同时内部依赖多个基础 UI 组件协同完成界面的动态展示。具体的关联关系如下表所示：

组件/模块名	功能说明	关联关系
learnStore	全局状态管理：保存当前题目序列、难度、题型配置、生命值及完成状态。	被 LearnView 视图调用，作为学习模式的单一数据源和控制中枢。
useQuestions	核心算法：生成 7 种题型的题目列表及数值防重计算。	跨模块复用：学习模式和对战模式共用该底层出题引擎。
useValidator	校验逻辑：判断用户输入正误，并生成“再大/小一点”的反馈文本。	跨模块复用：学习模式和对战模式共用该校验逻辑。
QuestionCard	UI 组件：负责单道题目的展示与输入框的交互绑定。	被 LearnView 父组件调用，依赖 learnStore 传递的单题数据进行渲染。
LifeBar	UI 组件：负责当前剩余生命值的红心渲染及扣血动画。	被 LearnView 父组件调用，实时响应 learnStore 中 lives 的状态变化。
LearnResultModal	UI 组件：展示最终得分、星级评价及重新开始/退出操作。	被 LearnView 父组件调用，仅在触发游戏结束（isGameOver）条件时显示。

学习模式组件关联图



4.1.4 用户接口

学习模式的用户接口由“学习配置页（LearnConfigView）”、“答题主页（LearnView）”以及“结算弹窗（LearnResultModal）”三个核心视图交互区域构成。界面视觉遵循大字体、高对比度的设计原则，确保学生能够直观操作。以下通过用户接口数据映射表，明确系统在各个阶段所涵盖的用户输入与系统输出数据：

界面状态	需要用户输入的数据	需要给用户输出的数据
学习配置页	1. 难度档位：点击选择 Easy/Medium/ Hard/Unbeatable 四档之一。 2. 训练题型：单选具体算法题型（或勾选“随机混合”）。	1. 高亮视觉反馈：当前被激活选中的难度与题型组件的外框色彩高亮。 2. 难度默认为 Medium，题型选择默认随机混合，用户点击开始挑战即可答题。
答题主页	1. 数值答案：在卡片文本框中键入单题口算结果。 2. 校验触发：点击“提交”按钮或按下键盘 Enter 键。 3. 手动推进：答错时，点击显式弹出的“下一题”交互按钮。	1. 当前生命值：LifeBar 组件中实时渲染的红心剩余数量及抖动反馈。 2. 核心算式文本：QuestionCard 中动态生成的口算文本（如 $12 \times 5 = ?$）。 3. 当前进度条：ProgressSteps 显示当前题号指针、总题数以及历史作答题目的对错颜色图例。 4. 动态错题提示：答错时输出 getHint 计算出的“再大一点~”或“再小一点~”中文引导文本。 5. 即时正误反馈：作答后输入框边缘瞬间切换的绿色（正确）或红色（错误）视觉边框。
结算弹窗	1. 退出流向：点击“返回配置”交互按钮，系统将路由跳转回配置页，供用户重新选择关卡难度和题型。	1. 最终作答得分：以“正确数 / 总题数”形式展示的定量成绩。 2. 通关定性状态：展现“挑战成功”或“挑战失败”的视觉大标题。 3. 综合星级评价：依据正确率百分比动态亮起的 2~3 颗金色五角星特效。

4.2. 对战模式功能实现

4.2.1 功能概要

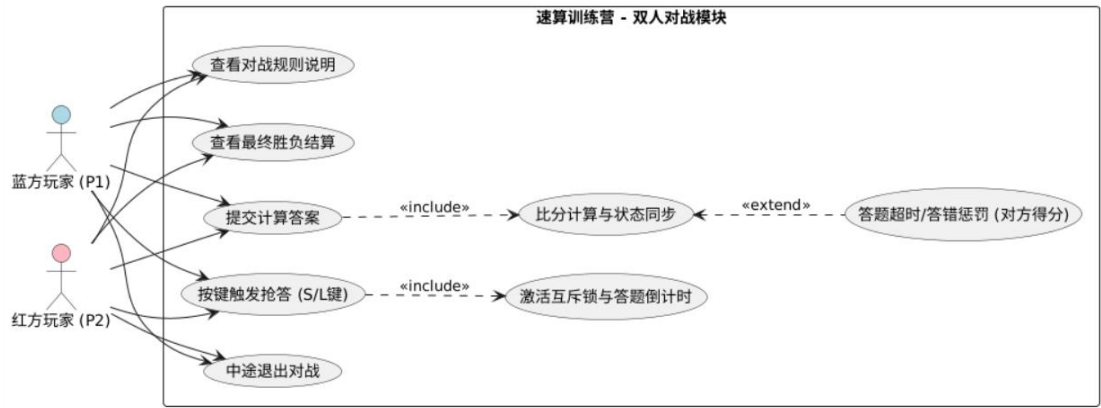
对战模式是“速算训练营”中面向同屏互动的双人竞技模块。该模块将单人枯燥的算术练习转化为具有时间压力的抢答比赛。对战双方在同一台设备上进行操作，蓝方玩家（Player 1）使用键盘 S 键抢答，红方玩家（Player 2）使用键盘 L 键抢答。每局对战系统会随机生成 5~7 道题目。题目出现后，双方有 10 秒的抢答时间。一方抢答成功后，系统会通过互斥锁锁定对方按键，并启动 10 秒的答题倒计时。答对者得 1 分；若答错或超时未作答，则对方得 1 分。当规定题目全部作答完毕后，系统将进行总分结算并弹出成绩视窗，宣布最终胜

者或平局。

用例表如下：

项目	内容
用例名称	对战模式双人同屏竞技
参与者	蓝方玩家（P1）、红方玩家（P2）
简要说明	两名玩家阅读规则后进入对战。系统随机出题并启动倒计时，玩家通过独立按键进行抢答与作答。系统根据抢答互斥结果及答题正误实时计算双方比分，最终判定胜负。
前置条件	玩家访问对战说明页，阅读规则并点击“开始对战”按钮初始化游戏。
备选与异常流	1. 无人抢答（抢答超时）：10 秒内双方均未按键抢答，系统判定本轮流局，双方均不加分，自动跳至下一题。 2. 答题超时：玩家抢答成功后，未能在 10 秒内提交答案，系统判定超时出错，对方得 1 分。 3. 输入错误答案：抢答玩家提交错误答案，对方得 1 分。 4. 中途退出：任何玩家点击底部的“退出对战”按钮，系统清空定时器与比分数据，退回对战说明页。
后置条件	题目全部耗尽，系统结束对战并弹出结算窗口，展示双方比分及最终胜负结果（蓝方胜 / 红方胜 / 平局）。

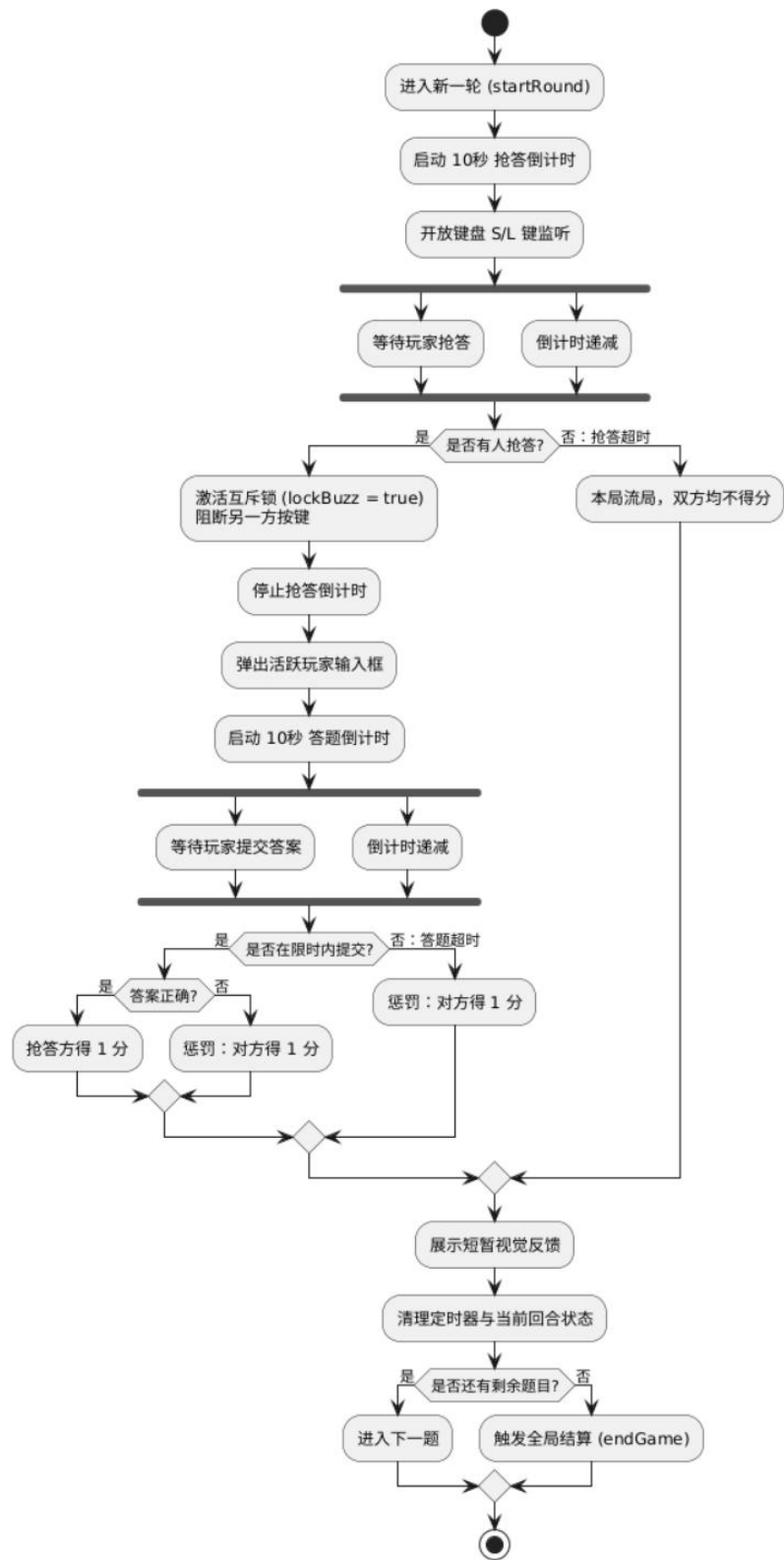
用例图如下：



4.2.2 功能详细

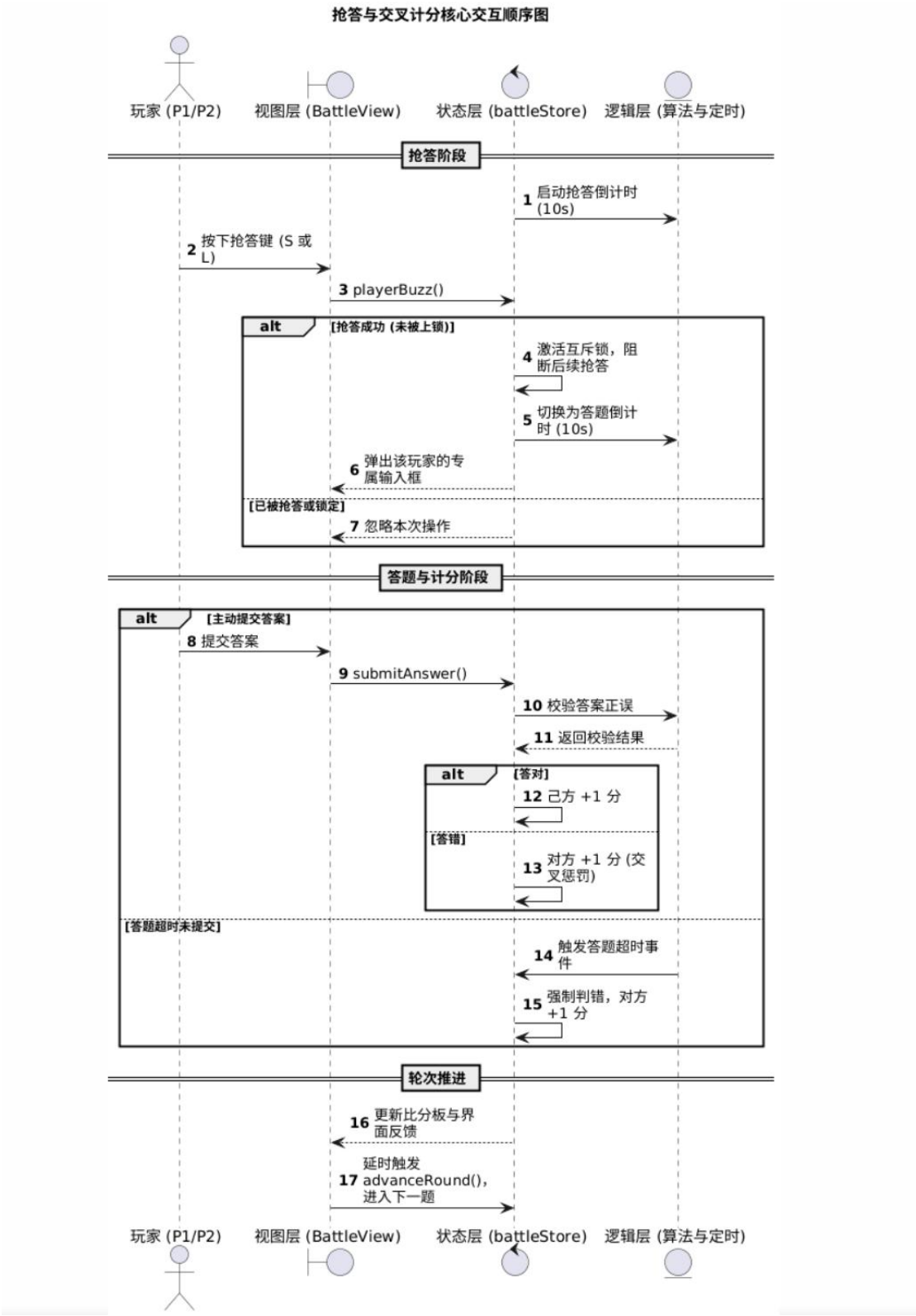
本地双人对战模式的底层逻辑围绕“状态隔离、互斥抢答、双重定时器与交叉计分”展开。当对战初始化完成后，系统进入循环的单局业务生命周期：每轮自动启动 10 秒的抢答倒计时，并开放物理键盘（蓝方 S 键 / 红方 L 键）的监听。一旦有玩家抢答，系统将立即锁定

抢答权，转而启动 10 秒的答题倒计时。在答题阶段，若玩家按时提交且答案正确，则己方得 1 分；若提交错误答案或答题超时，则触发“交叉惩罚”机制，对方自动得 1 分。若抢答倒计时结束依然无人按键，则本局流局，双方均不得分。具体的业务控制决策流如下图所示：



在具体的技术实现与对象协作上，双人对战模式通过 useKeyboard 组合式函数实现了基

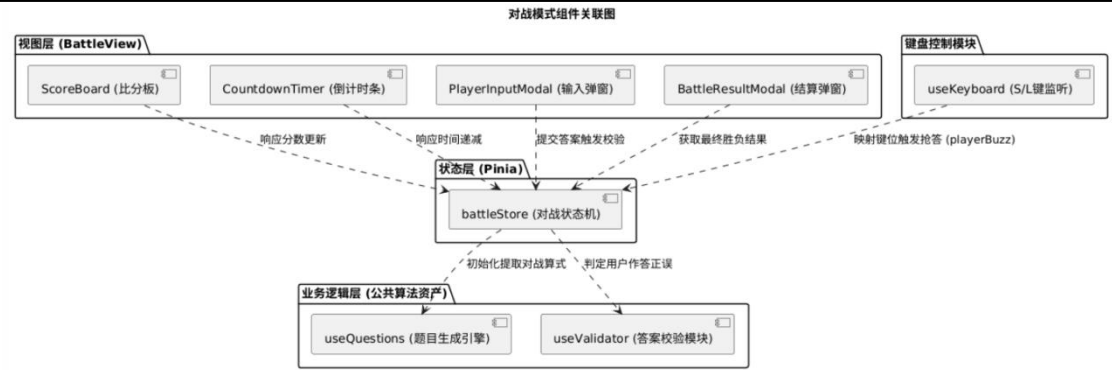
于物理键盘的底层按钮捕获，并与视图层和状态层深度解耦。玩家按下按键后，视图层触发状态层的抢答动作，状态层通过加设互斥锁来绝对阻断另一方的后续按键干扰。作答提交后，状态层调用逻辑层进行精准比对，并根据校验结果及定时器状态执行分数的加减流转，最终延时触发进入下一题。核心对象间的消息传递与交互顺序如下图所示：



4.2.3 相关功能

本地双人对战模式的实现依赖于多层次组件的紧密协作。为保证系统性能与工程结构的清晰，对战模块在独立管理其特有的抢答状态与并发定时器的同时，充分复用了学习模式底层的算式生成与答案校验引擎。此外，对战模式额外引入了独立的键盘监听模块来处理同屏物理按键的冲突。这些组件间的依赖与功能边界划分十分明确，确保了模块之间的高内聚与低耦合。具体的相关组件及其协同关系如下表及组件图所示：

组件名	功能说明	关联关系
battleStore	对战模式的状态机，集中管理双人比分、活跃玩家、互斥锁、双重倒计时定时器以及轮次推进。	被视图层 BattleView 及各 UI 组件调用，作为双人对战的数据总线与逻辑调度中心。
useKeyboard	独立的底层键盘监听模块，专门负责捕获并隔离物理键盘的 S 键与 L 键事件。	被 BattleView 引入，在监听开启期间捕获按键并直接触发 battleStore 中的 playerBuzz 抢答动作判定。
useQuestions	核心算法引擎，负责按对战参数生成 5~7 道的去重算式。	跨模块共享：被 battleStore 在 initGame 时调用，与“学习模式”完全共享该底层算法逻辑。
useValidator	核心校验组件，验证用户提交的答案与实际计算结果是否一致。	跨模块共享：被 battleStore 在判题时调用，依据其返回的正误布尔值来执行加分或交叉惩罚。
ScoreBoard & CountdownTimer	UI 展示组件，分别负责渲染双人比分板以及 10 秒倒计时进度条特效。	被 BattleView 父组件调用，单向绑定并实时响应 battleStore 中 scores 与 timeLeft 的状态变化。
PlayerInputModal	UI 交互组件，为抢答成功的活跃玩家提供专属的居中数字输入面板。	在互斥抢答成功后由 BattleView 渲染，收集用户答案并通过事件抛出给状态层进行校验计算。



4.2.4 用户接口

本地双人对战模式的用户接口（UI）由“对战说明页（BattleIntroView）”、“对战主页（BattleView）”以及“结算弹窗（BattleResultModal）”三个核心视图交互区域构成。为了适应同屏双人竞技的特殊场景，界面采用了高度对称的视觉布局，并通过红蓝双色（蓝方 P1，红方 P2）进行严格的视觉隔离与身份提示。

以下通过用户接口数据映射表，明确系统在该模块各个阶段涵盖的用户输入与系统输出数据：

界面状态	需要用户输入的数据	需要给用户输出的数据
对战说明页	1. 动作触发：点击“开始对战”按钮，初始化双人游戏数据并进入抢答界面。	1. 规则文本：展示双人对战的抢答键位（S 键/L 键）、倒计时规则与交叉计分规则。 2. 身份标识：展示“  蓝方玩家”与“  红方玩家”的视觉标识。
对战主页	1. 物理抢答指令：玩家通过键盘按下 S 键（蓝方）或 L 键（红方）。 2. 数值答案：抢答成功后，活跃玩家在弹出的输入框中键入数字答案。 3. 动作指令：点击提交答案，或点击顶部“退出”按钮中止对战。	1. 全局比分板：顶部实时渲染蓝方（P1）与红方（P2）的当前得分对比。 2. 双重倒计时条：顶部渲染动态递减的进度条（10 秒抢答倒计时 或 10 秒答题倒计时）。 3. 对战算式文本：屏幕中央展示当前轮次的算式（如 $15 \times 4 = ?$ ）。 4. 即时状态反馈：算式下方输出当前局势文本（如“蓝方抢答成功！”、“回答正确”、“回答错误”或“超时”）。 5. 专属输入面板：仅对抢答成功方弹出带有其专属图标的数字输入视窗。
结算弹窗	1. 确认与返回流向：点击“返回对战大厅”等关闭弹窗按钮，清理对战数据并路由跳转回说明页。	1. 最终胜负宣告：基于总分对比，输出高亮大字标题（“蓝方胜”、“红方胜”或“平局”）。 2. 最终比分数据：展示双方完成全部 5~7 道题目后的最终得分对比。

4.3. 全局与基础功能

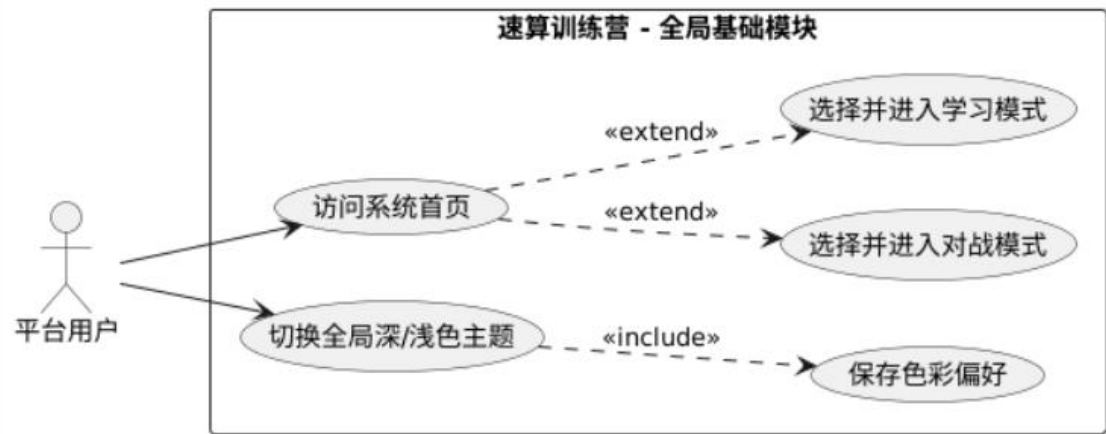
4.3.1 功能概要

全局与基础功能模块构成了“速算训练营”的底层业务骨架与入口导航。该模块主要提供两项核心能力：一是通过首页（HomeView）实现对“学习模式”和“对战模式”的双路路由分发；二是提供全局的深色/浅色（Dark/Light）主题切换机制，满足不同环境光线下的视觉需求，并将用户的色彩偏好持久化存储至本地浏览器中。

用例表如下：

项目	内容
用例名称	全局导航与主题管理
参与者	平台用户
简要说明	用户通过首页选择期望的训练模式进行跳转；或通过顶部导航栏的主题切换按钮，全局一键更改系统的色彩外观，系统自动记录偏好。
前置条件	用户通过浏览器成功访问系统网址。
备选与异常流	1. 无本地主题偏好：首次访问的新用户（或清理了缓存），系统将自动读取浏览器或操作系统的原生色彩偏好（prefers-color-scheme）进行默认渲染。
后置条件	成功跳转至对应模式视图，或全站 UI 样式无缝切换为深/浅色模式并完成本地持久化存储。

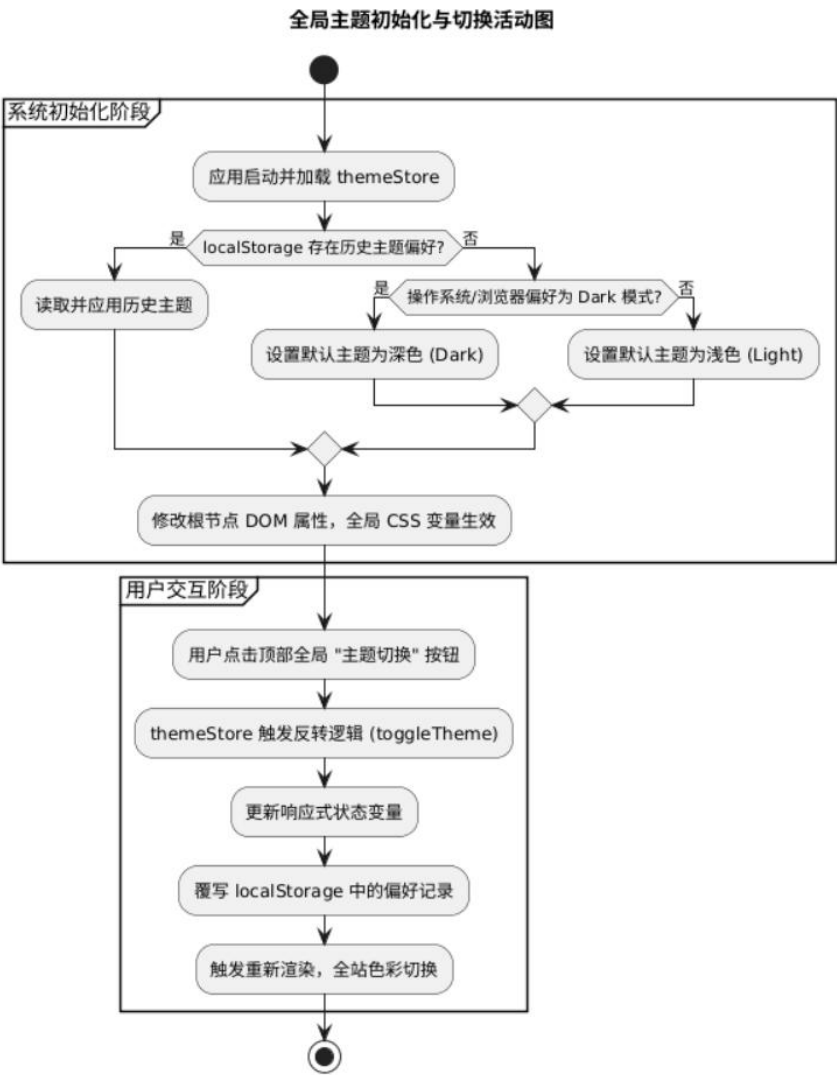
用例图如下：



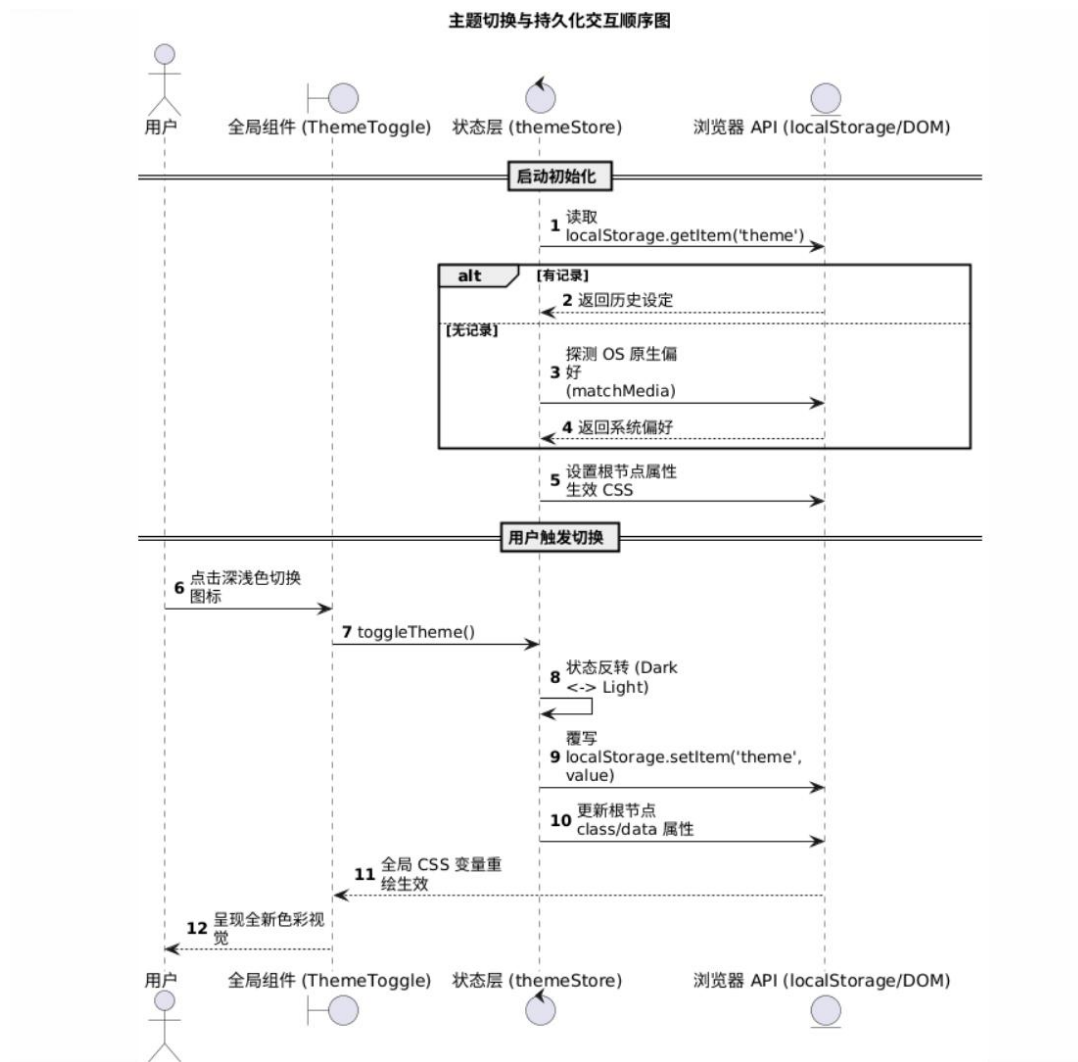
4.3.2 功能详细

全局功能的底层控制流主要体现在“状态的持久化读取与响应式重绘”上。当系统初次挂载时，themeStore 会优先校验本地 localStorage 中是否存在历史配置，若存在则直接应用；若不存在，则利用 window.matchMedia 探测操作系统的明暗偏好进行初始化设定。用户在任意页面点击主题切换控件时，状态层会反转当前的色彩标识，同步将新值覆写至本地缓存，并通过底层修改 HTML 根节点的属性或类名，进而触发全局 CSS Variables 的色彩替换，实现全站所有组件视觉的瞬间无缝切换。

具体的持久化读取与切换业务活动决策流如下：



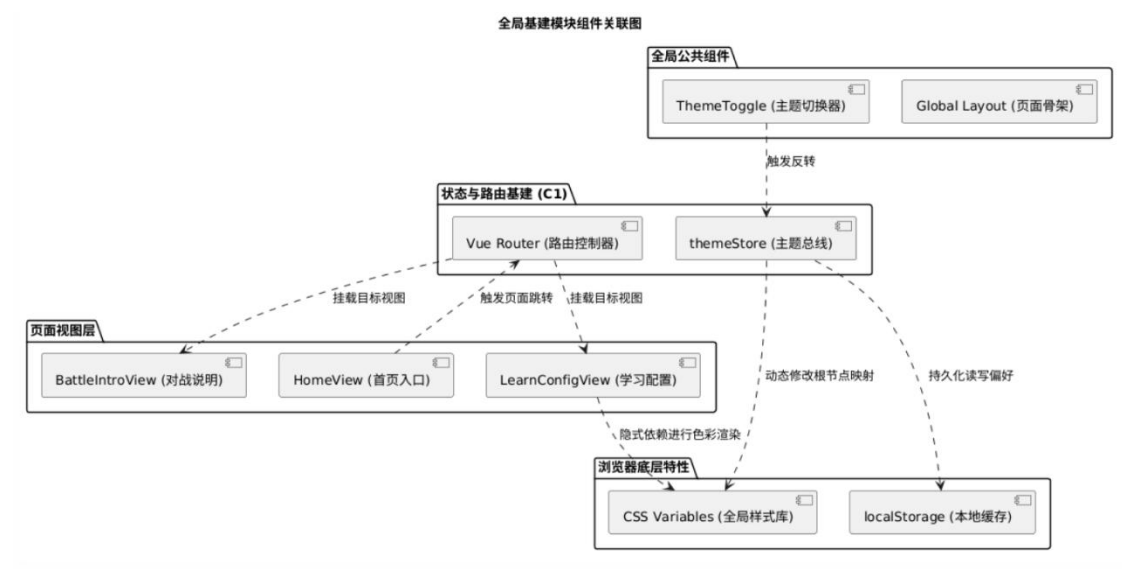
在系统各对象的消息协作上，视图层（如顶部控制栏组件）仅作为交互触发器，真正的读写逻辑与 DOM 隔离均由 themeStore 统一封装处理，确保了代码的高可维护性。其交互顺序如下图所示：



4.3.3 相关功能

全局与基础功能处于系统架构的最底层，它是其他所有业务模块得以正常渲染和跳转的公共基建。其涉及的具体组件与关联依赖如下表及组件图所示：

组件/模块名	功能说明	关联关系
HomeView	纯视图组件：系统的默认落地页，利用大色块卡片展示两种模式的特色并提供入口。	调用 Vue Router 暴露的路由方法，负责向配置页或对战大厅分发流量。
themeStore	状态控制中枢：维护全局的 <code>isDark</code> 状态变量，并封装本地缓存读写逻辑。	跨模块供应：被整个应用的所有页面与组件隐式依赖（通过 <code>CSS</code> 变量），并被顶部栏组件显式调用。
Vue Router	基础架构：前端路由控制器，负责管理各个页面视图的映射关系与浏览器的历史栈。	核心依赖： HomeView 以及各个弹窗返回动作均强依赖其 <code>push</code> 或 <code>replace</code> API 实现页面流转。



4.3.4 用户接口

全局基础功能主要集中在 HomeView（首页视图）以及全局页面骨架（Layout）的导航栏部分。其主要的输入输出映射如下：

界面状态	需要用户输入的数据	需要给用户输出的数据
系统首页	1. 模式选择：点击屏幕中央的“学习模式”卡片按钮或“对战模式”卡片按钮。	1. 产品信息：输出对应 Logo、系统副标题及四大特性标签（即时反馈、生命值系统等）。 2. 模式概览：卡片上直观输出各模式的简短说明与规则特点。
全局导航栏	1. 模式切换指令：点击右上角的主题切换按钮。	1. 全局色彩呈现：输出整体界面的背景色、卡片表面色、文字色等系统性视觉反馈。 2. 当前状态图标：切换器自身展现代表当前色彩模式的视觉图标（太阳代表浅色，月亮代表深色）。

5. 非功能需求

5.1. 权限需求

本项目为纯前端公开应用，不设用户登录与权限管理。所有访问者均为匿名用户，拥有完全相同的功能访问权限（首页、学习模式、对战模式、主题切换）。无需区分用户角色或进行权限控制。

5.2. 兼容互换性

无。本项目为首次开发，无历史版本需要兼容；且为独立前端应用，不与其他产品存在接口或数据交换依赖。

5.3. 安全性

本产品为纯前端公开应用，无后端、无数据库、无用户登录，因此：

- 不收集任何用户个人信息，符合《个人信息保护法》
- 无数据泄露、丢失风险
- 无外部攻击风险（无后端接口）
- 通过 HTTPS 传输，保证内容完整性
- 无需用户身份确认、密码管理或授权机制
- 无其他特殊安全需求。

5.4. 健壮性

本产品为纯前端静态应用，故障域仅限于客户端浏览器环境。

（1）成熟性

核心功能（题目生成、答案校验）经单元测试验证，预期故障率低于 0.1%。对战模式的计时器与互斥锁管理存在一定复杂度，预期故障率约 1%，但不会导致数据损坏，仅影响当前对局。

（2）障害兼容性

当单题生成异常时，系统跳过该题并重新生成，不影响整体流程。当状态管理发生错乱或计时器卡死时，用户刷新页面即可完全重置，虽当前进度丢失，但可立即开始新对局。对战模式提供屏幕按钮作为键盘失效的备用方案。

（3）恢复性

本产品不存储持久化数据，故障恢复极为简单：用户刷新页面后，系统在 3 秒内完全重置并恢复正常使用，无需任何数据修复操作。

（4）障害解析与对应

故障定位依赖浏览器开发者工具的控制台错误信息。修复完成后通过 git push 触发 GitHub Actions 自动部署，用户刷新页面即可获得修复版本。

5.5. 使用性（操作性）

本产品为 Web 应用，用户通过浏览器访问即可使用，无需安装。

（1）用户界面

整体界面采用卡片式布局，主色调区分深色/浅色两种主题，用户可一键切换。首页提供“学习模式”和“对战模式”两个大尺寸入口按钮，适合儿童点击。学习模式包含难度选择（Easy/Medium/Hard/Unbeatable）和题型选择（七种运算类型）的配置页面，配置完成后进入答题页面。答题页面顶部展示步骤条和剩余生命值，中部展示题目和输入框，底部提供提交按钮。对战模式包含规则说明页和同屏抢答页，双方得分和当前题号始终可见。

（2）操作性

本产品所有操作均通过鼠标点击完成，学习模式的答案输入需使用键盘数字键，对战模式的抢答可使用键盘 S/L 键或点击屏幕按钮。无需命令行操作，无需记忆快捷键。页面间的导航通过按钮触发路由跳转，无复杂菜单结构。

（3）理解性

首页两个入口按钮的文字说明清晰，用户无需学习即可理解各模式的用途。学习模式的难度档位采用常见英文标识（Easy/Medium/Hard/Unbeatable），题型选择使用中文标签（加法、减法、乘法等），用户可根据自身水平选择。答题时的正误反馈通过颜色区分（绿色正确、红色错误）和文字提示（“再大一点”“再小一点”）双重表达，易于理解。

（4）掌握性

用户首次使用时，从首页进入任一模式后按界面引导操作即可完成一轮练习，无需阅读说明文档。对战模式在进入前设有单独的规则说明页，明确告知 S/L 键的使用方法和计分规则，用户阅读一次即可掌握。整体学习成本约为 1-2 分钟。

（5）运用性

本产品无需用户进行任何配置或维护操作。主题偏好自动跟随浏览器本地存储，下次访问时保持用户选择。无需登录、无需更新、无需备份。

5.6. 效率性（性能）

（1）硬盘容量

无。本产品为纯前端 Web 应用，无需安装，不占用用户硬盘空间（浏览器缓存除外，该缓存由浏览器自动管理）。

（2）内存容量

本产品运行于浏览器环境，预期内存占用约 50MB 以内。以现代 PC 普遍配备 8GB 以上内存计，占用率不足 1%，无性能风险。

（3）处理速度

题目生成：< 10ms

答案校验：< 1ms

路由跳转：< 100ms

对战抢答响应：< 30ms

（4）响应性能

用户点击按钮或提交答案后，界面反馈在 50ms 内完成。10 秒倒计时进度条精度控制在 ± 0.1 秒以内。

（5）执行效率性

采用 Vue3 响应式系统配合 Pinia，仅在数据变更时触发相关组件更新。题目生成使用 Set 结构去重，避免低效遍历。对战结束后立即清理计时器，释放资源。

（6）资源效率性

无后端请求，不占用网络带宽（仅初始加载）。无定时轮询、无后台持续运行任务。

5.7. 维护性

（1）版本升级方式

本产品采用 GitHub 进行版本管理。当需要进行功能更新或 Bug 修复时，开发人员在本地完成代码修改并推送至 GitHub 仓库，GitHub Actions 将自动触发构建和部署流程，约 2-3 分钟后新版本上线。用户无需任何手动更新操作，刷新页面即可获得最新版本。版本升级不涉及数据库迁移或后端服务重启，风险极低。

（2）分析性

本产品无后端日志系统，故障分析依赖前端信息收集手段，开发阶段：通过浏览器开发者工具的控制台查看错误信息和警告；代码层面：关键逻辑（如题目生成、答案校验）已编写单元测试，故障时可快速定位到具体函数。

由于本产品无持久化数据、无后端依赖，故障复现相对容易。开发人员在本地即可模拟用户操作环境，通过断点调试或添加临时日志进行问题定位。

5.8. 移植性

无。本产品为 Web 应用，基于标准 HTML/CSS/JavaScript 开发，采用 Vue 3 框架和 Vite 构建工具，最终产出为标准静态网页文件，运行于现代浏览器中。浏览器作为运行时环境已屏蔽底层操作系统差异，因此不存在向其他 OS 移植的需求。本产品在当前技术选型下已具备跨平台通用性。

5.9. 用户文档

- 开发计划书
- 需求与功能设计规格书

5.10. 其他

没有

6. 使用和操作方法

6.1. 环境设定

本系统为纯前端 Web 应用，无需安装任何软件或配置复杂环境。用户使用前需确保以下条件：

- （1）硬件环境：终端设备具备正常的显示与输入能力。
- （2）网络环境：首次访问时需连接互联网，以加载系统静态资源。加载完成后，后续答题过程不依赖网络连接（对战模式为本地双人对战，无需网络同步数据）。
- （3）浏览器要求：使用现代浏览器访问，推荐 Chrome 90+、Edge 90+、Firefox 88+、Safari 14+。不支持 IE 浏览器。
- （4）系统部署：本项目部署于 GitHub Pages，用户通过项目组长提供的网址（格式为 <https://用户名.github.io/仓库名>）即可访问。无需注册、登录或提供任何个人信息。

(5) 主题偏好：系统默认跟随浏览器偏好设置（浅色/深色模式）。用户可在页面右上角或底部手动切换主题，切换后偏好将保存至浏览器本地存储，下次访问时自动恢复。

6.2. 使用方法

6.2.1 系统启动

用户在浏览器地址栏输入项目部署网址，按回车键即可进入系统首页。首页展示“学习模式”和“对战模式”两个入口按钮。

6.2.2 学习模式使用方法

步骤一：配置练习参数

在首页点击“学习模式”按钮，进入学习配置页。用户需完成以下设置：

- (1) 选择难度：点击 Easy（简单）、Medium（中等）、Hard（困难）或 Unbeatable（极限）四档中的一档。不同难度影响题目数值范围、初始生命值。
- (2) 选择题型：从加法、减法、乘法、除法、幂运算、平方根、随机混合七种题型中勾选需要练习的类型，可多选。
- (3) 确认无误后，点击“开始挑战”按钮进入答题页。如需返回首页，点击“退出”按钮。

步骤二：逐题作答

- (1) 答题页顶部显示当前生命值，右上角显示退出按钮。中部显示当前题目序号与总题数（如“第 3/10 题”）及题目内容（如“ $12 + 7 = ?$ ”）。
- (2) 用户在输入框中输入答案，点击“提交”按钮（或按 Enter 键）。系统即时判断正误：
 - 回答正确：题目区域显示绿色提示，自动跳至下一题。
 - 回答错误：题目区域显示红色提示，并给出“更大/更小”建议，同时扣除一点生命值。
- (3) 若生命值归零：系统弹出失败弹窗，展示题目正误汇总，用户点击“返回配置页”按钮可重新开始。
- (4) 若完成全部题目：系统弹出评分弹窗，展示最终得分（如“8/10”），用户点击“返回配置页”按钮可继续练习。
- (5) 练习过程中如需退出，点击右上角退出键，直接返回学习配置页。

6.2.3 对战模式使用方法

步骤一：阅读规则

- (1) 对战规则已在首页展示，用户直接点击“开始对战”按钮。

步骤二：双人对战

- (1) 答题页顶部左右两侧分别显示“蓝方玩家”和“红方玩家”的当前得分。中部居中显示当前题目内容。
- (2) 抢答阶段：
 - 玩家 1 按下键盘上的 S 键，玩家 2 按下 L 键，可激活己方抢答。
 - 抢答成功后，系统弹出输入弹窗，同时启动 10 秒倒计时进度条。另一方抢答键被禁用，直至当前答题结束。
 - 若用户不便使用键盘，可点击屏幕上的“抢答”按钮（作为备用方案）。
- (3) 答题阶段：

在弹窗中输入答案并点击“提交”（或按 Enter 键），系统判断正误：

- 答对：当前玩家得 1 分，得分区域更新。
- 答错：对方玩家得 1 分。

若倒计时归零仍未提交答案，当前题超时判负，对方得 1 分。

（4）每题结束后，系统自动跳至下一题，重新开启抢答阶段。

（5）全部题目完成后，系统弹出结算弹窗，宣布胜者。用户可选择“返回首页”。

6.2.4 系统退出

本系统为 Web 应用，无需正式退出流程。用户关闭浏览器标签页即可离开。系统不保留任何用户个人信息或答题记录。

7. 注意限制事项

7.1. 制约 / 限制事项

本项目采用 Vue3、JavaScript、Vite 和 Pinia 技术栈进行前端开发，系统主要运行在浏览器环境中。因此，本系统的使用需要依赖现代浏览器环境，例如 Google Chrome、Microsoft Edge 等。对于版本过低或不支持现代 JavaScript 特性的浏览器，可能无法保证页面显示效果和功能运行完全正常。

本系统的数据主要用于前端页面状态管理和本地交互展示，当前版本不连接后端服务器，也不使用数据库进行长期数据存储，数据仅在本地浏览器中临时存储。因此，用户在练习过程中，刷新页面或关闭浏览器后练习记录会丢失，不提供长期数据持久化。

题目生成规则仅面向课程实验需求设计，题型覆盖 7 种基础运算类型，不支持复杂公式、文字应用题、分数运算或方程求解。答案判断主要提供即时正确/错误提示及生命值扣减机制，以保证系统稳定性和基本交互体验为目标。

7.2. 假定事项

1. 用户具备基础计算机操作能力，能在浏览器中完成练习模式选择、作答和结果查看。
2. 双人对战模式假定两名用户在同一设备或浏览器中进行操作，不涉及远程网络同步。
3. 系统假定运行环境已安装 Node.js 与 npm，可通过 Vite 启动本地前端项目。
4. 系统假定使用现代桌面操作系统（Windows 11/macOS Ventura+/Ubuntu 22.04+）及兼容浏览器。

7.3. 特记事项

1. 本地双人对战模式中，两名用户在同一设备上交替答题，需要特别注意状态切换时的数据隔离，避免 A 用户的作答结果被 B 用户的操作意外覆盖。所以后续通过 Pinia 分别管理两名用户的独立状态。

2. 开发计划书中规定的四档难度（Easy / Medium / Hard / Unbeatable）对应的具体参数范围，在本次需求与功能设计规格书中尚未明确定义。上述参数将在 C3 模块开发时由开发人员根据实际实现复杂度进行确定和调整。

3. 开发计划书中规定本地双人对战模式“每局随机生成 5-7 道题”，但各类题型（加、减、乘、除）的具体分布比例在本次需求与功能设计规格书中尚未明确定义。后续将采用均匀随机策略实现，即每道题从四种运算类型中随机抽取，各题型出现概率相等。

4. 开发计划书中规定学习模式的校验方式为“批量校验：全部作答完毕后统一校验”。经开发团队讨论决定，实际实现采用逐题即时校验方式，即用户每完成一道题提交答案后立即判断正误，正确显示绿色通过，错误标红并给出“更大/更小”提示。该调整旨在提供更及时的反馈体验，便于用户在答题过程中即时了解对错并进行调整，无需等待全部题目作答完毕。

8. 附录

8.1. 业务规则

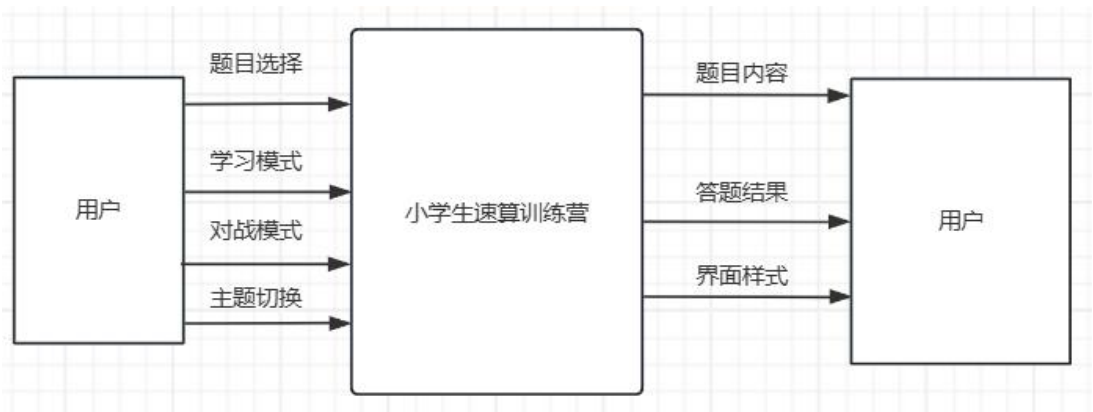
无。本产品为公开访问的速算练习工具，无角色区分、无权限控制、无金额或敏感操作。所有用户均可无限制地使用全部功能。

8.2. 术语表

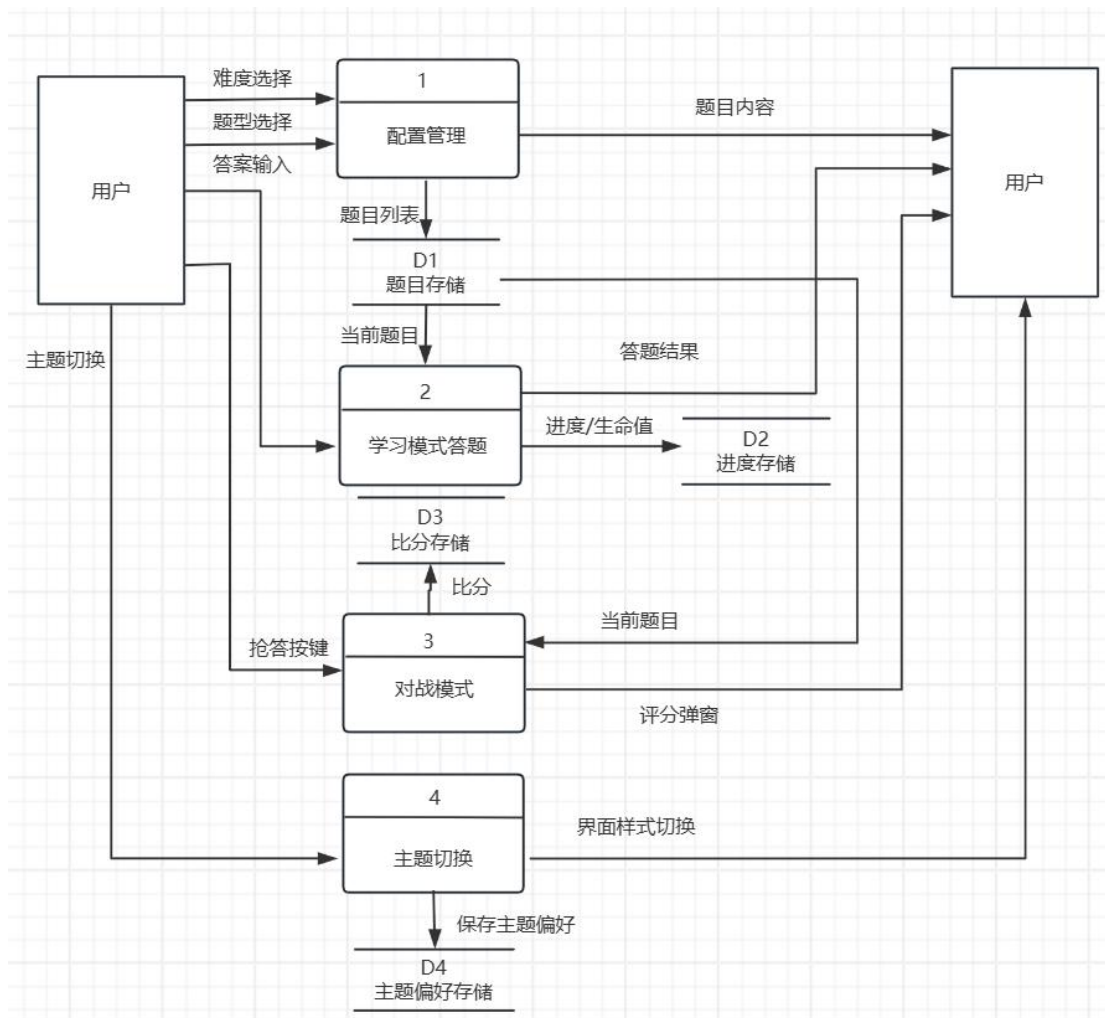
无。本产品面向普通小学生用户，所有界面用语均为日常词汇（加法、减法、得分、难度等），无需要额外解释的专业术语。

8.3. 数据流图

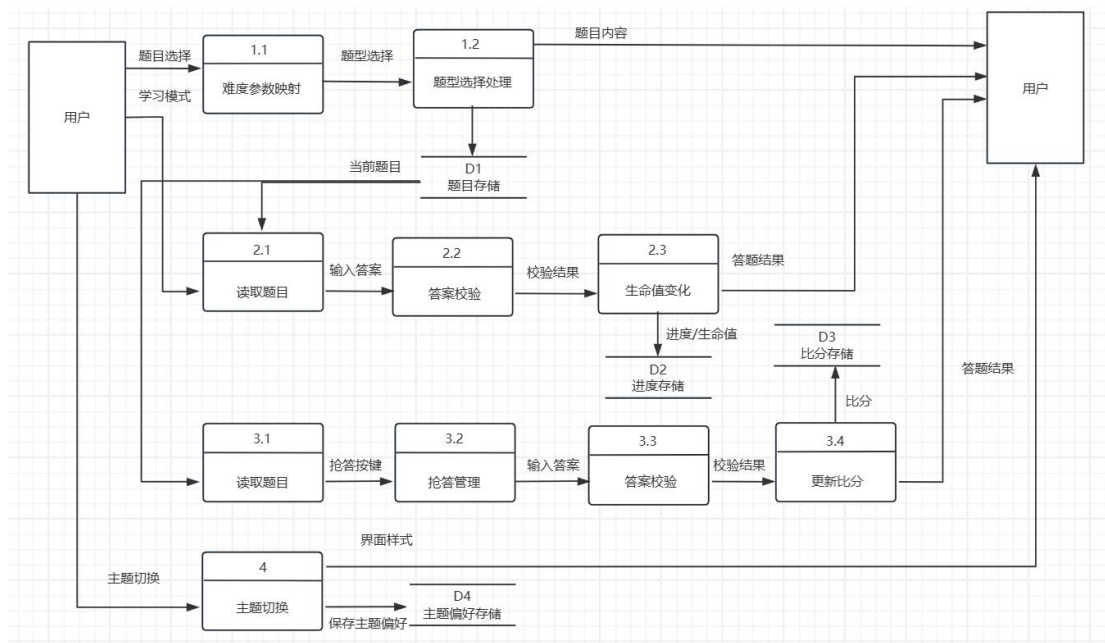
（1）顶层数据流图



（2）第 0 层数据流图



(3) 第1层数据流图



8.4. 数据字典

对象	属性	说明	取值范围
难度档位	难度	用户选择的练习 难度级别	Easy, medium, hard, unbeatable
题型类型	题型	用户选择的运算 题型	加法、减法、乘法、除法、幂运算、平方根、 随机混合
题目数量	题目 数量	每轮练习的题目 总数	学习模式 10 题，对战模式 5-7 题
题目内容	题目 文本	题目展示内容	字符串
正确答案	正确 答案	题目的正确答案	整数
用户答案	用户 答案	用户输入的答案	整数
校验结果	校验 结果	答案是否正确	正确、错误
生命值变 化	生命 值	当前剩余生命值	Easy/Medium: 0-3; Hard/Unbeatable: 0-2
答题结果	答题 结果	学习模式下，用 户的答题结果	三星、两星
读取题目	获取 题目	根据当前索引从 D1 中读取题目	D1 题目存储
抢答管理	抢答 互斥 锁	控制同一时间只 有一人能抢答成 功	已锁定、未锁定

更新比分	比分 变化	更新玩家的比分	整数：整数
主题偏好	主题 偏好	用户选择的主题 模式	日间模式、夜间模式
学习模式	答题 控制	逐题校验答案、 扣减生命值、切 换题目	进行中、结束
对战模式	对战 控制	管理抢答互斥、 计分、题目切换	进行中、结束