

华中科技大学

《软件工程》项目报告

题目：基于 AI 的心理疗愈软件——自留地 SelfPlanet

<https://github.com/sakaaanaYu/SelfPlanet>

课程名称： 软件工程

专业班级： 传播 2201 班

组 名： 热干面土豆小组

同组成员： 学号： U202217034

姓名： 余易昕

学号： U202217024

姓名： 江思怡

学号： U202217026

姓名： 李昕豫

学号： U202211780

姓名： 陈芊聿

指导教师： 冯琪

报告日期： 2025.5.7

任 务 书

一 总体要求

1. 综合运用软件工程的思想，协同完成一个软件项目的开发，掌握软件工程相关的技术和方法；
2. 组成小组进行选题，通过调研完成项目的需求分析，并详细说明小组成员的分工、项目的时间管理等方面。
3. 根据需求分析进行总体设计、详细设计、编码与测试等。

二 基本内容

根据给出的题目任选一题，自行组队，设计与开发中软件过程必须包括：

1. **问题概述、需求分析：**正确使用相关工具和方法说明所开发软件的问题定义和需求分析，比如 NABCD 模型，Microsoft Visio，StarUML 等工具（20%）；
2. **原型系统设计、概要设计、详细设计：**主要说明所开发软件的架构、数据结构及主要算法设计，比如墨刀等工具（35%）；
3. **编码与测试：**编码规范，运用码云等平台进行版本管理，设计测试计划和测试用例（30%）；
4. **功能创新：**与众不同、特别吸引用户的创新（10%）；
5. **用户反馈：**包括用户的使用记录，照片，视频等（5%）。

目 录

任务书	I
1 问题定义	1
1.1 项目背景及意义	1
1.2 项目基本目标	3
1.3 可行性分析	3
1.4 人员管理	5
2 需求分析	6
2.1 需求分析概述	6
2.2 UML 相关需求分析图	7
2.3 原型系统设计	14
3 概要设计和详细设计	22
3.1 系统结构	22
3.2 类图等	25
3.3 关键数据结构设计	27
3.4 关键算法设计	36
4 实现与测试	40
4.1 实现环境和代码管理	40
4.2 关键函数说明	42
4.3 测试计划和测试用例	44
4.4 结果分析	47
5 总结	48
5.1 用户反馈	48
5.2 总结	49
6 体会	50

1 问题定义

1.1 项目背景与意义

1.1.1 市场痛点

根据《2022 年国民心理健康调查报告》，2022 年我国国民抑郁风险检出率达 10.6%，焦虑风险检出率为 15.8%，其中 18-34 岁的青壮年群体占近四成。当今社会节奏快、压力大，许多年轻人面临着焦虑、孤独和抑郁等负面情绪。而在对心理健康领域知识需求的调研中，七成国民表露了对“自我调节”类服务的需求。正规的心理治疗对他们来说也许尚不必要，但在工作与生活的社交场以外，他们的确需要一个能独处、能疗愈的“温暖一隅”来接收自己的负面情绪，并恢复内心的平静。由此，一款安抚性强、私密性高的心理疗愈类 App，无疑是最高效、便捷的解决方案。

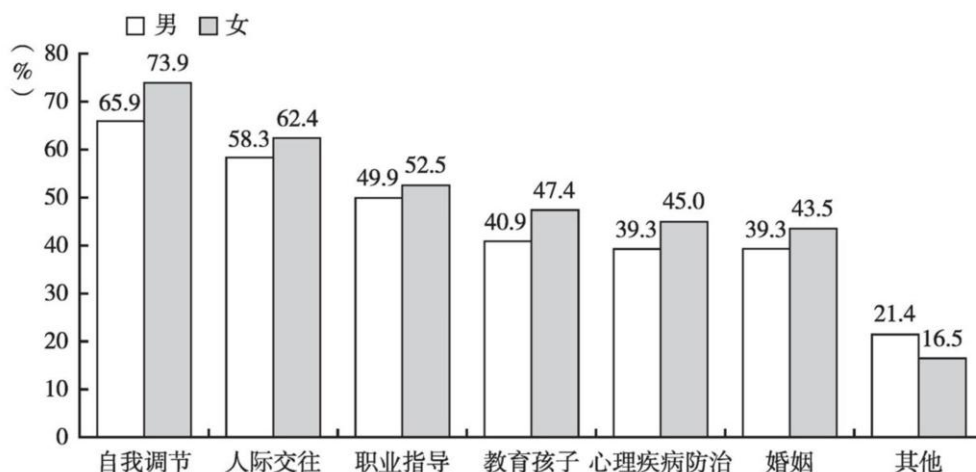


图 1 不同性别对不同心理健康领域的知识需求

(数据来源：2020 年国民心理健康调查报告)

1.1.2 竞品分析

目前，市场上流行的心理疗愈应用通常存在着功能性不强、私密性不足和同质化严重等问题。以下就接受度较高的“月食”和“心岛日记”两款应用分析。在“聊天疏导”这一核心功能点上，“月食”均为付费咨询，价格高昂，对于仅有负面情绪但并不至于罹患心理疾病的群体来说，此功能性价比较低。且其咨询对象的专业性不详，难以保证咨询的有效性及合理性。“心岛日记”将聊天疏导功能设计为卡

通动物，并接入类似“Siri”“小爱同学”的智能语音助手，其回复均套用“复述问题+名人名言”这一模板，极为生硬，实效甚微。综合来看，两款热门疗愈型应用均难以保证心理疏导的高功能性与有效性。



图 2 竞品应用功能界面

在“抒发感受”这一核心功能点上，“月食”和“心岛日记”两款应用均**仍为较为开放的社交平台，不够专注自我**。除了在公开广场发布、浏览帖子以外，两款应用均设有语音聊天室功能，使用者可与陌生人连麦聊天、探讨问题。但就现阶段情况而言，这些聊天室多以“交友”和“付费咨询”为目的，在一定程度上已背离其“发泄情绪、畅所欲言”的初衷，**难以为使用者提供私密安心的抒情空间**。且公开、非匿名式的社交增加了受骚扰、诈骗的风险，**安全性难以保障**。

综合来看，我们希望在**提升功能性、加强私密性和破除同质性**的基础上，借助人工智能内容生成技术打造一款专注于年轻人内心的，**沉浸式、私密式、灵活式**心理疗愈应用。

1.1.3 用户画像

2022 年国民心理健康调查报告显示，18-24 岁青年人群体抑郁风险检出率最高，占总数的 24.1%，25-34 岁壮年群体紧随其后，占比 12.4%。在快节奏、强竞争的社会环境下，面临负面情绪和心理压力的年轻人并不鲜见。

立足于此，本项目将目标用户定位于年龄在 18-35 岁的年轻人，他们在社交场所中面临一定的压力和负面情绪，但尚未达到需要专业心理治疗的程度，需要

一个独立温暖的空间，来发泄情绪、获得安慰并进行自我探索。

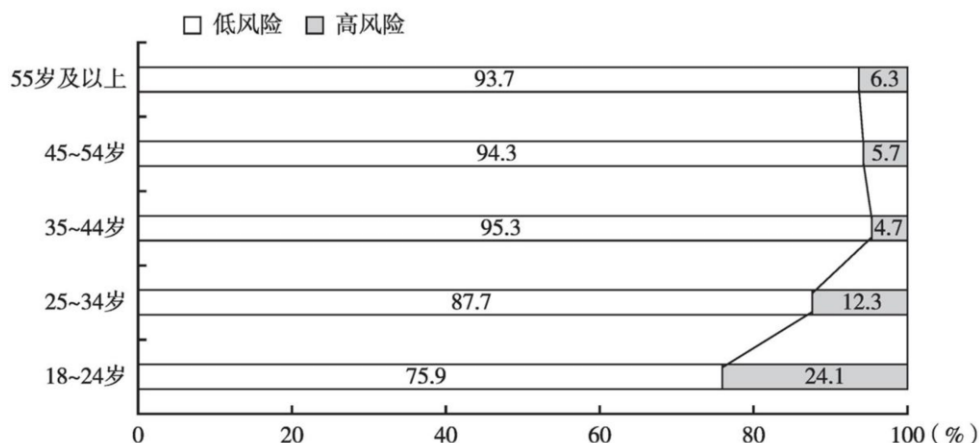


图 3 不同年龄组的抑郁风险检出率
(数据来源：2022 年国民心理健康调查报告)

1.2 项目基本目标

项目主题集中于情感抒发与心理疗愈。旨在为用户提供一个安全、私密的情感表达空间，通过 AI 插件等工具和可视化交互方式帮助用户处理并减轻负面情绪，以实现内心平静和情感疗愈。

本项目的战略目标是满足年轻用户群体的情感需求，尤其是在日常生活中经历负面情绪，有罹患心理疾病风险的群体。它采用人工智能插件与个性化 IP 定制相结合的方式为用户提供个性化、自适应的情感支持，以帮助用户更好地疗愈心理问题。

另一目标是为用户提供一个自我探索情感世界的空间，用户可不受拘束地表达内心情感、释放压抑的情绪。这一相对私密的窗口有助于用户合理抒发负面情绪，探索内心平静和情感满足的方式。

1.3 可行性分析

1.3.1 技术可行性

Woebot 是基于行为认知疗法 (CognitiveBehaviorTherapy, 简称 CBT)，帮助用户改善心理焦虑和抑郁的一款人工智能聊天机器人，它通过分享心理学理论和引导用户改变自己的语言引导用户平和客观地接纳生活中的负面事件，帮助用

户获得拥有好心情的能力。

项目依托其现有成熟的 AI 技术框架，结合自然语言处理与行为认知疗法，技术实现路径清晰。开发团队可利用开源工具及 github 管理代码降低服务器成本，个性化虚拟形象的实时互动功能通过轻量化设计可优化性能需求。但团队缺乏 AI 开发经验，AI 模型的本地化适配、多模态交互（表情/动作反馈）及隐私数据加密需专业技术支持，在算法调优与系统稳定性上面临着较多挑战。

UI 设计方面，小组成员均可熟练使用 figma、墨刀等原型图设计工具，能够快速进行页面 demo 设计。

总体而言，在现有技术生态下，项目具备可落地性。

1.3.2 经济可行性

该应用的目标用户为 18-35 岁心理健康需求旺盛的年轻群体，市场规模庞大（2023 年中国心理咨询行业规模超 480 亿元），付费意愿较强。此外，O2O 模式（线上虚拟+线下实体）能提升用户粘性，形成差异化竞争力。潜在风险在于市场竞争激烈，需通过精准营销（社交媒体+IP 运营）快速抢占用户心智，确保长期盈利。

初期成本集中于 AI 接口调用、UI/UX 设计，人力成本占比高；若产品后期正式上线，可通过虚拟装扮（能量币充值）、文创周边（公仔、贴纸）和联名广告实现稳定收益。

1.3.3 社会可行性

产品契合国家“十四五”心理健康政策导向，符合 Z 世代情绪管理需求，具有社会公益性。用户隐私通过匿名社区与加密通信保障，符合《个人信息保护法》要求。设计上采用温暖治愈的视觉风格，降低心理压力群体的使用门槛。

但仍需注意 AI 疗愈需避免替代专业心理咨询的伦理争议，产品需明确功能边界并加入防沉迷提示。潜在风险在于内容审核机制需严格，防止负面情绪扩散。

总体而言，项目符合社会道德与法律规范，具备可持续运营基础。

1.4 人员管理

本小组共有 4 名成员，前期背景分析、竞品分析与 UML 类图设计等由大家共同完成，开发部分主要由两名同学完成。具体分工如下：

- 余易昕：前期市场及竞品调研、UML 图设计、原型图设计、前端开发、报告撰写与整合
- 江思怡：前期市场及竞品调研、UML 图设计、原型图设计、前端开发、报告撰写
- 李昕豫：前期市场及竞品调研、UML 图设计、原型图设计、报告整合
- 陈芊聿：前期市场及竞品调研、UML 图设计、原型图设计、报告撰写

2 需求分析

2.1 需求分析概述

现针对市场痛点进行如下 NABCD 分析：

● N (Need, 需求)

当代年轻人面对快节奏生活与社交压力，亟需一个私密、温暖的情感释放空间。根据《2022 年国民心理健康调查报告》，目标用户群体中抑郁风险检出率高达 24.1%，需求刚性显著。

传统心理咨询成本高、门槛高，而现有心理疗愈类应用（如“心岛日记”）存在三大痛点：**功能性不足**（如 AI 回复机械化）、**私密性欠缺**（开放社交增加风险）与**同质化严重**（缺乏创新互动形式），难以满足用户对私密性、个性化和专业性的需求。

● A (Approach, 方法)

瞄准上述痛点，自留地聚焦 18-35 岁群体，精准定位用户“轻量级心理疏导”需求，通过 AI 驱动的个性化虚拟形象 (AiMonster) 和熟人社交模块 (“小天地”)，为用户提供 24 小时陪伴、情绪能量转化与安全分享空间提供基于 AI 的轻量化情绪管理工具。

技术层面采用策略模式（商品分类）与工厂模式（怪兽形象生成），确保系统可扩展性。功能设计上结合个性化虚拟形象与封闭式社交空间，采用“AI+情感化设计”双核驱动模式：

- 1. AI 自适应疗愈：**基于行为认知疗法 (CBT) 的 Woebot 框架，结合讯飞星火大模型的多模态理解能力，实现文本/语音交互、表情动作反馈及情绪能量转化功能；
- 2. 沉浸式体验设计：**用户可定制虚拟小怪兽形象（毛发、服饰、配饰），通过“投喂情绪-获取能量币-解锁装扮”机制增强参与感；
- 3. O2O 闭环生态：**线上虚拟商品（装扮）与线下文创周边（公仔、贴纸）结合，延伸情感陪伴至现实场景；
- 4. 安全与合规保障：**匿名社区、端到端加密通信及防沉迷提示系统，规避隐私泄露与过度依赖风险。

● B (Benefit, 收益)

1. **用户价值：**提供 24 小时陪伴式疗愈，降低心理疏导门槛，同时通过可视化情绪记录帮助用户自我觉察。
2. **商业价值：**虚拟装扮、文创周边、联名广告构成多元盈利模式，能量币体系（情绪转化机制）巧妙引导付费转化。
3. **社会价值：**响应国家心理健康政策，减少社会心理服务资源压力，助力构建心理健康“预防-干预”体系。

● C（Competitors，竞争）

目前，市场上流行的心理疗愈应用通常存在着功能性不强、私密性不足和同质化严重等问题。相比竞品（如“心岛日记”“月食”），自留地的差异化优势在于：

1. **技术深度：**AI 模型自适应优化与多模态交互（表情/动作）提升疗愈真实感，优于“心岛日记”的模板化回复；
2. **私密性设计：**熟人社交模块（“小天地”）与匿名机制降低社交压力，规避“月食”开放式聊天室的安全隐患；
3. **IP 衍生潜力：**小怪兽形象具备强延展性，可通过联名合作与文创开发形成品牌壁垒。

● D（Delivery，推广）

潜在威胁来自大厂竞品（如腾讯“小 Q”或字节跳动心理类应用）的技术与流量优势，需通过快速迭代与垂直用户运营巩固市场地位，通过 IP 运营快速建立品牌认知。产品推广策略如下：

1. **推广策略：**依托社交媒体（小红书、B 站）进行 IP 形象传播，联合高校与职场社群开展心理健康主题活动；
2. **合作生态：**与心理咨询机构合作提供专业内容，增强用户信任；接入第三方支付与物流系统，保障交易流畅性；
3. **迭代计划：**基于用户反馈优化 AI 对话逻辑，扩展白噪音库与装扮品类，逐步上线“情绪日历”等桌面组件功能，提升日常使用频次。

2.2 UML 相关需求分析图

2.2.1 参与者识别

表 1 参与者识别

种类	细项	参与者
人	外部	用户
	内部	管理者
系统	其它系统	内容审核系统
	数据库	用户语料库
其他	插件	AI 聊天机器人插件 Woebot

2.2.2 各系统用例图

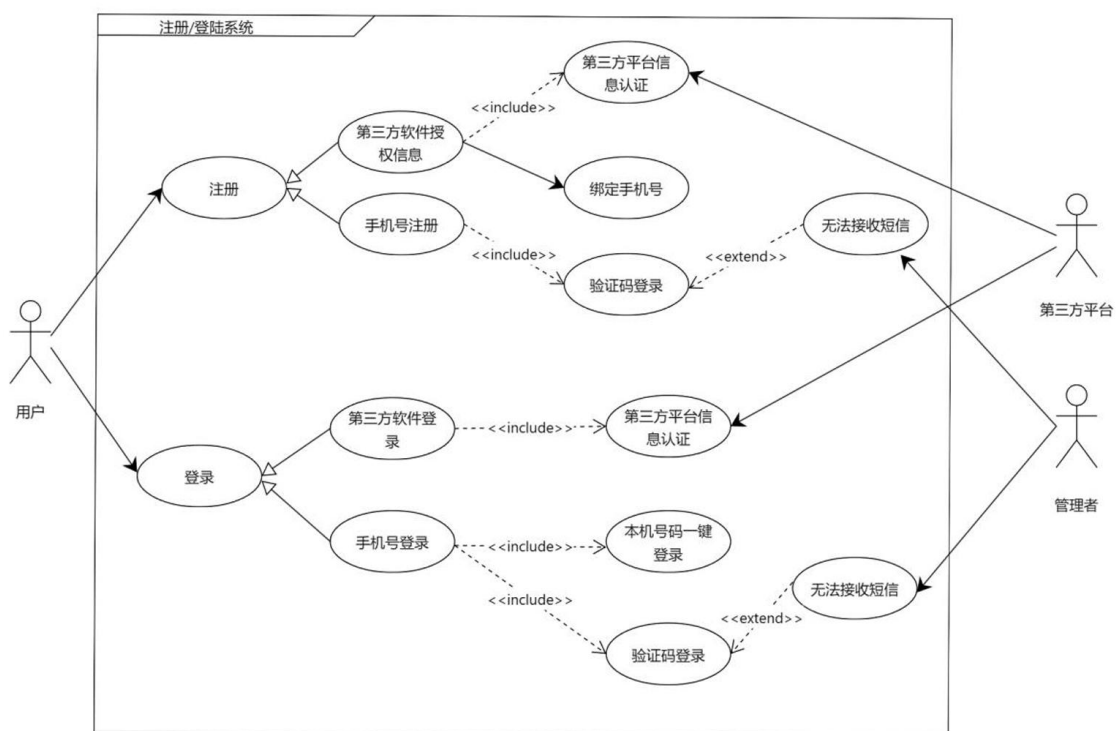


图 4 注册/登陆系统

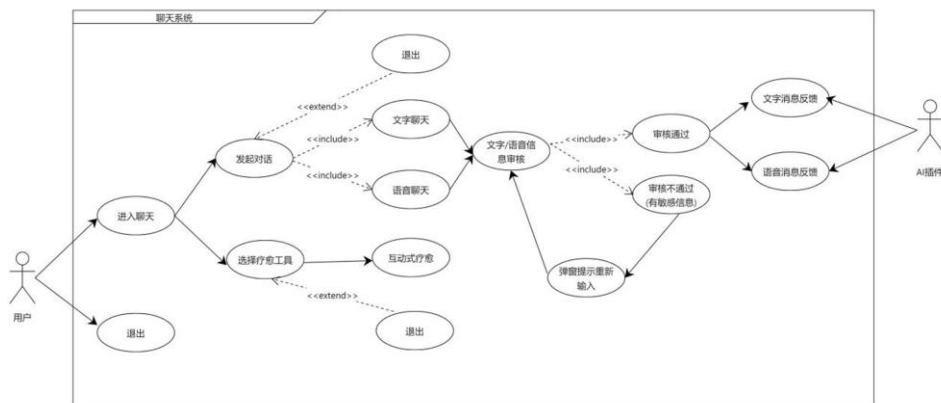


图 5 聊天系统

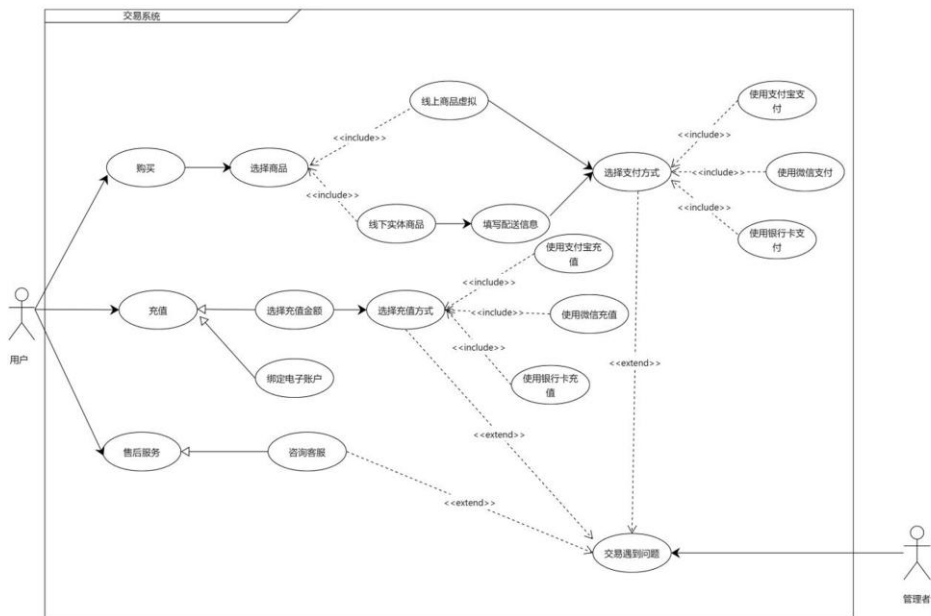


图 6 交易系统

2.2.3 协作图

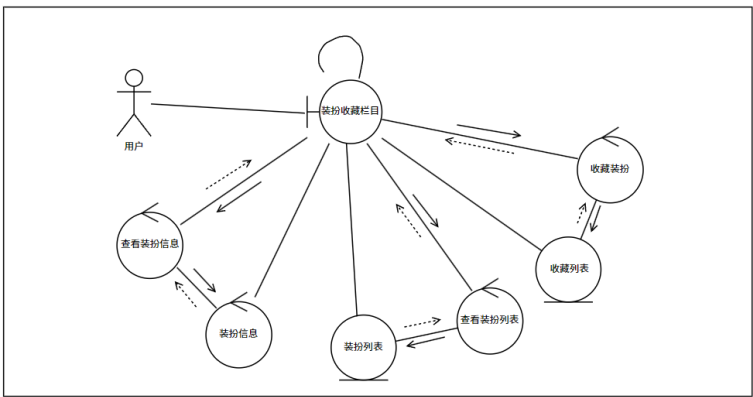


图 7 装扮收藏协作图

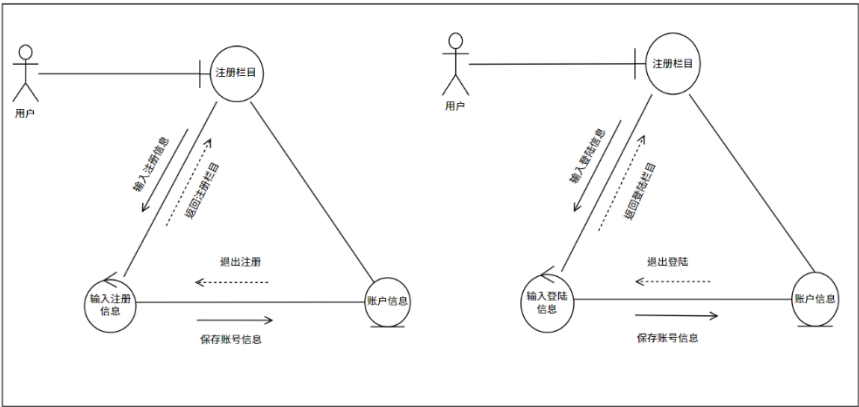
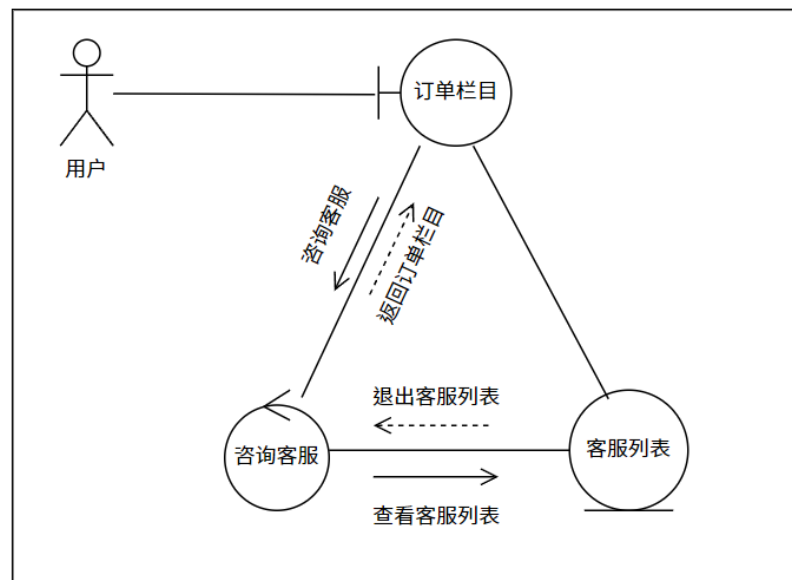
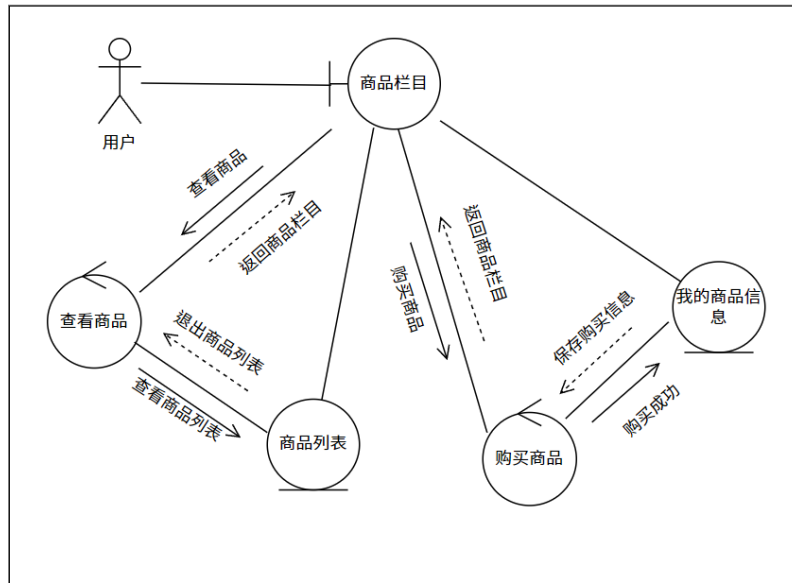


图 8 注册登陆系统协作图



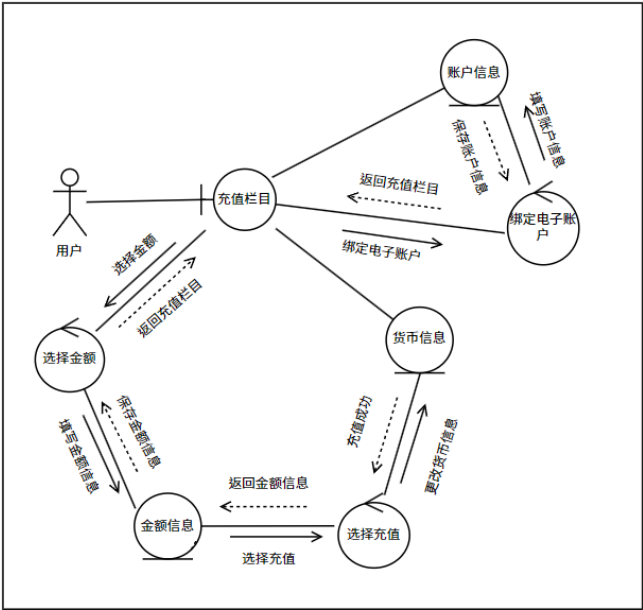


图 9 交易系统协作图

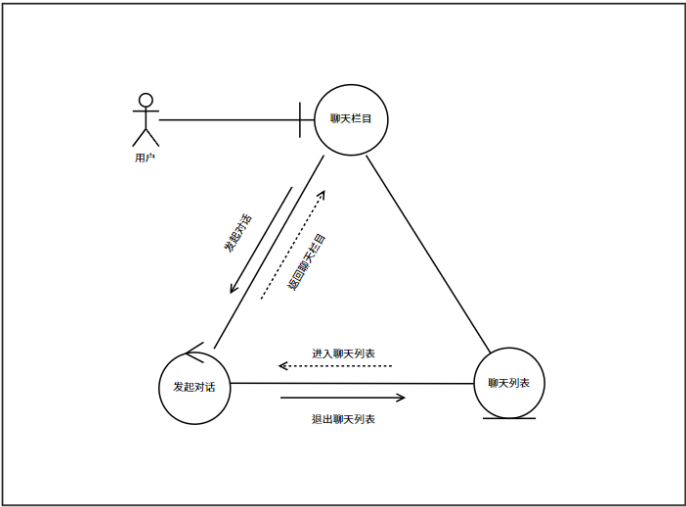
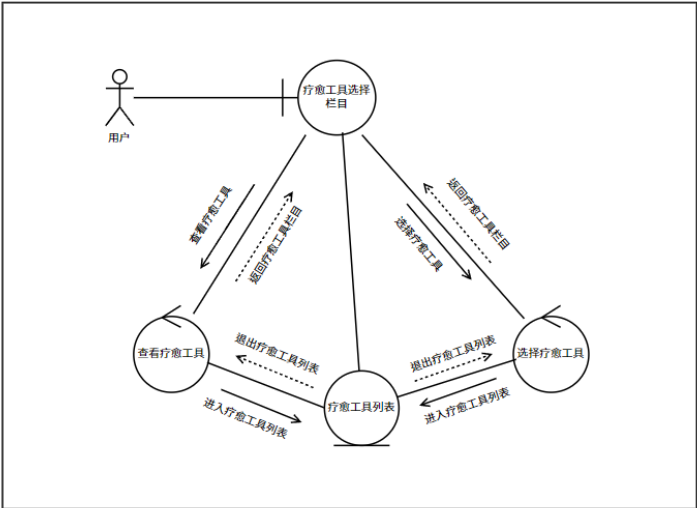


图 10 聊天栏目协作图

2.2.4 状态图

用户进入聊天后处于正常的聊天状态，若用户在聊天过程中呈现出沉迷或当用途的趋势，则聊天变为正常的状态，系统给予警告后用户整改，则恢复正常的聊天状态，若用户不整改则终止聊天。

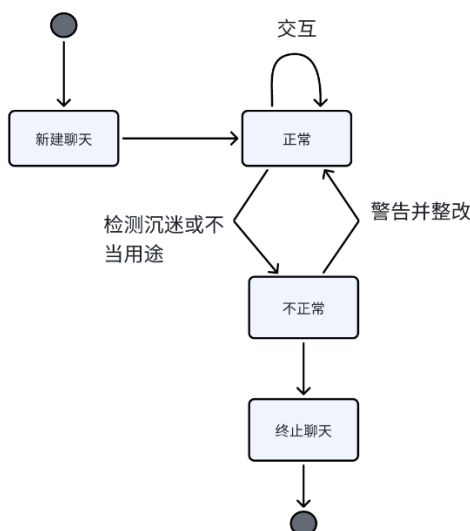


图 11 聊天异常状态检测

帖子发布后处于正常状态，若系统在帖子中检测到了敏感信息，则帖子变为不正常状态，系统给予提示警告后用户整改，则恢复帖子的正常发布，若用户不整改，则撤回帖子。

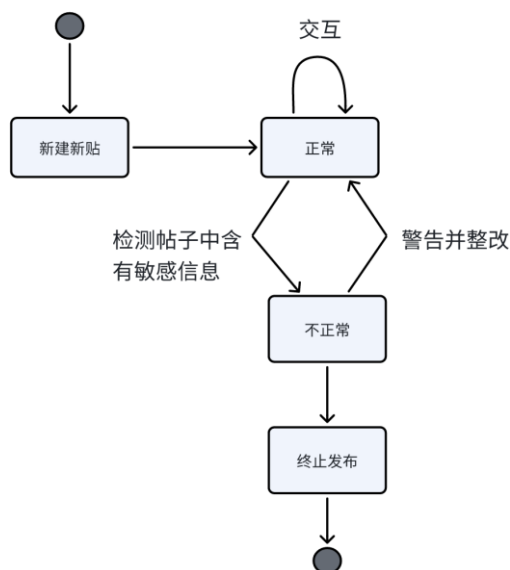


图 12 帖子异常内容检测

2.2.5 活动图

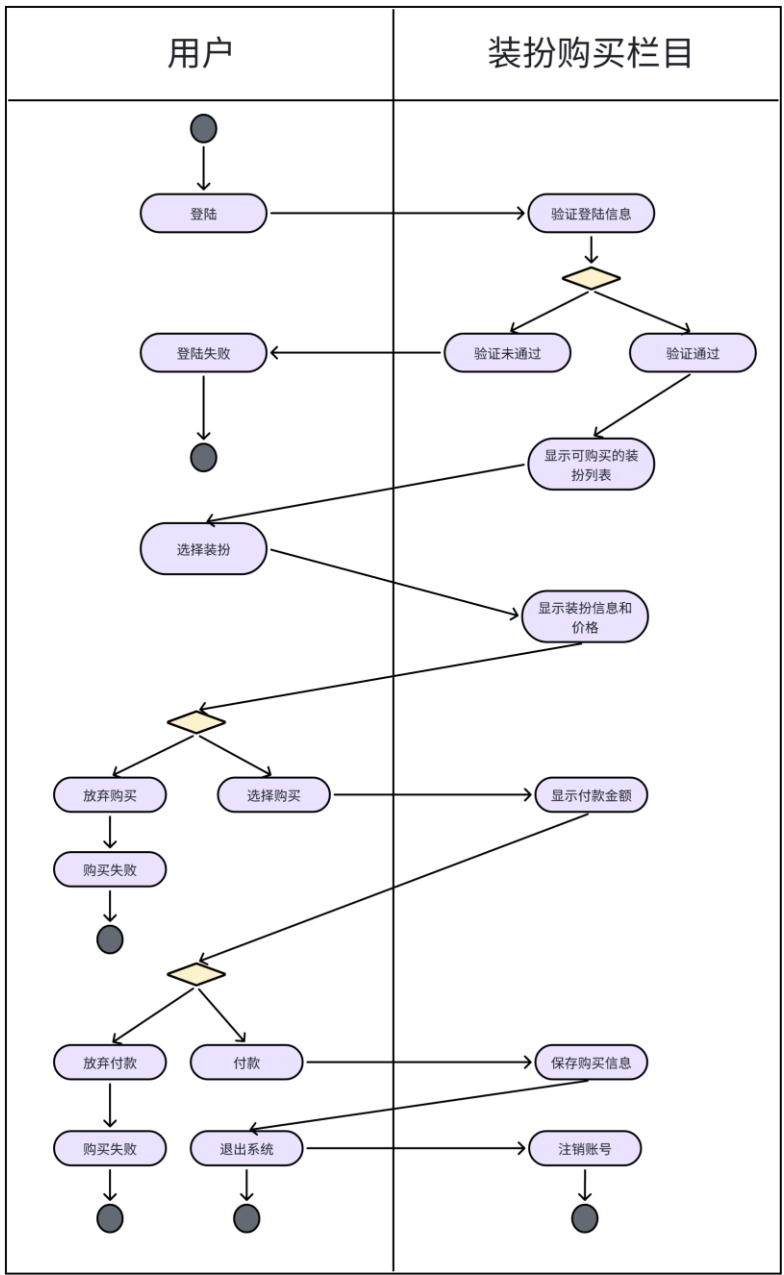


图 13 购买装扮操作

1. 用户登陆,在装扮购买栏目验证登陆信息,若信息验证未通过,则登陆失败,若信息验证通过,则继续活动流程;
2. 装扮购买栏目显示可购买的装扮列表,用户在列表中选择装扮,栏目中显示装扮信息和价格,若用户选择放弃购买,则购买终止,若用户选择购买则活动流程继续;
3. 栏目中限制付款金额,若用户放弃付款,则购买终止,若用户付款则活动流

程继续；

4. 系统保存购买信息，用户退出系统，购买流程完成，后台注销账号信息。

2.3 原型系统设计

产品的交互界面在原型制作工具“即时设计”和“Figma”中完成，其中包含 5 个一级界面，共 15 个界面。为了丰富产品想要提供的温暖、私密的使用体验，紧扣“探索你的内心宇宙”核心概念，在界面设计上我们以杏黄色和橙棕色作为主视觉色系，配上色彩丰富的小怪兽形象，给人以温暖又明快的感觉。

在软件的 icon 及交互 UI 上，产品使用了扁平动画风格的设计，让页面设计更生动可爱，为用户营造出一种亲切感，同时也能很好的融合小怪兽的个性化形象。在以疗愈功能为主的页面设计里，均尽可能采用简洁、直观的设计排布，突出疗愈功能的同时也体现了探索、净化内心的核心概念。在按钮的设计上，产品多使用直观简约的 icon 设计；对于用户发布的不同帖子，为直观展现各用户当前的情绪状态，帖子呈现方式为瀑布流。

具体的交互功能将在本部分进行详细介绍。在下方的图像中，紫色圆形代表操作流程，可与文字说明一一对照。绿色圆形代表页面中高亮的部件，可在下方表格中找到其说明。

2.3.1 开屏登陆界面

开屏动画持续 3 秒，小怪兽和它站立的星球自下而上慢慢浮现，随即跳出“想和我倾诉一下吗”和“我这里有很多疗愈工具”的默认对话框。点击对话框，对话框内容将进行更新，浮现更多怎样的疗愈语句。



图 14 开屏登录界面

在开屏界面的设计上，我们采取了明亮温暖的杏黄色，展示了我们标志性的形象--小怪兽，和暗合产品名称“SelfPlanet”的星球形象。同时，明确了自留地的两个核心功能--语聊疗愈和疗愈工具，给用户带来统一连续的使用体验。

用户在登录界面可以选择微信登录和手机号登录，首次登录将自动创建账号。

2.3.2 首页相关功能

首页、侧拉栏、收藏相关功能的交互图如下图所示。

用户进入首页，点击左上角的菜单按钮，可以唤出侧拉栏(流程 1)。在侧拉栏中，用户可以查看当前的能量币数量、账号头像和用户名称等信息，可以点击账户设置、添加好友等图标进入相应的界面，也可以进入设置、联系客服界面或登出账号。这里用“我的收藏”功能举例。

用户点击“我的收藏”图标后，进入“我的收藏”界面(流程 2)，选购收藏商品后，点击右下角购买按钮进行结算。用户也可以点击小红心将商品移除收藏名单。

表 2 首页界面相关功能描述

序号	描述
1	用户点击菜单按钮，弹出侧边栏。
2	用户点击账户设置、添加好友等图标进入相应的界面。
3	用户选购收藏商品后，点击右下角购买按钮进行结算。



图 15 首页界面

2.3.3 聊愈相关功能

语聊疗愈、疗愈工具相关功能的交互图如下图所示。



图 16 疗愈工具相关功能界面

用户进入首页，可以看到小怪兽的形象和当前小怪兽的情绪值。用户通过点击导航栏的“聊愈”按钮进入选择界面(流程 1)。在选择界面，用户如果选择“聊天”则跳转聊天界面(流程 2)，用户可以在对话框中输入内容，点击发送按钮发送，也可以语音和小怪兽聊天。点击“聊天记录”按钮可以查看历史聊天记录(流程 3)，历史聊天记录的界面上方显示对话的主题、时间和用户与小怪兽的昵称。如果用户在选择界面选择“疗愈工具”，则跳转白噪音播放界面(流程 4)。用户在该界面可以选择播放的白噪音中所包含的元素，并调整它们在声音中占据的比例。

表 3 疗愈工具相关功能描述

序号	描述
1	用户可以在首页和小怪兽交互，以获取情绪值。
2	用户可以在对话框中输入内容，点击发送按钮发送，也可以语音和小怪兽聊天。
3	用户在该界面可以选择播放的白噪音中所包含的元素，并调整它们在声音中占据的比例。

2.3.4 小天地相关功能

小天地相关功能的交互图如下所示。用户进入首页，点击“小天地”按钮，进入小天地界面(流程 1)。在此页面中，用户可以上下滑动查看朋友们的心情记录，可以点击爱心按钮进行点赞，点击评论按钮进行评论。点击屏幕上方的加号，唤出心情转盘(流程 2)，选择对应心情，再次点击中央表情，进入心情记录界面(流程 3)。

表 4 小天地相关功能描述

序号	描述
1	用户可以上下滑动查看朋友们的心情记录，可以点击爱心按钮进行点赞，点击评论按钮进行评论。
2	用户记录自己的心情，点击确认发布到小天地。



图 17 小天地相关功能界面

2.3.5 装扮相关功能

装扮、充值相关功能的交互图如下所示。用户进入首页，点击“装扮”按钮，进入装扮界面(流程 1)。点击屏幕下方上拉框中的装扮，可以查看饰品在小怪兽身上的效果。点击屏幕中部的分类栏，可以分类查看饰品。点击右下方的“就 TA 了”按钮，弹出购买确认弹框(流程 2)。如果用户点击“确认”按钮，则弹出“购买成功”弹窗(流程 3),点击右上角关闭弹窗;如果用户点击“取消”按钮，则关闭弹窗。点击上方的加号，弹出能量币充值弹窗(流程 4)。

表 5 小天地相关功能描述

序号	描述
1	用户点击屏幕中部的分类栏，可以分类查看饰品。
2	用户可以选择充值能量币的数值。

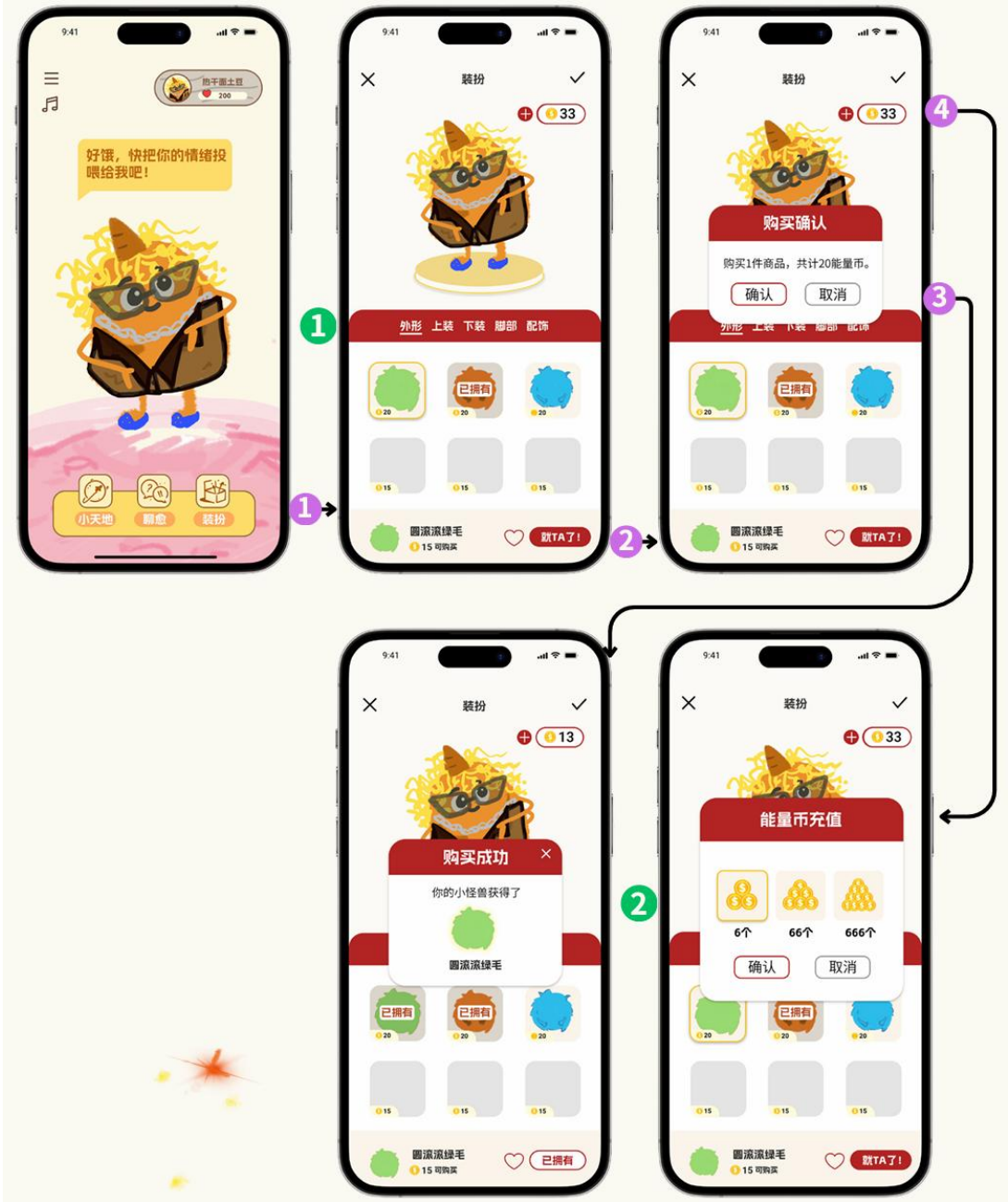


图 18 装扮相关功能界面

2.3.6 特色功能：桌面小组件交互

2024 年 4 月，鸿蒙系统正式升级为鸿蒙 4.0，随之而来的是桌面万象小组

件功能的单独上架，用户可根据自己的想法将桌面万象小组件自由组合，打造个性化面。除鸿蒙系统以外，安卓、IOS 系统也均支持桌面小组件功能，各应用的核心功能可被缩至小方块内，用户即使不点开应用，也可于小组件完成交互。

基于此考量，本产品单独设计出两款桌面小组件，方便用户在空闲时间与本产品完成简单交互：

1. 情绪日历

用户在使用“小天地”功能发布帖子前，会在“今天过得怎么样”转盘中选择代表当天情绪的表情。情绪日历小组件将一个月中每天的情绪记录下来，并浓缩至一个小日历中展现。将此小组件放置于手机桌面，用户能直观地观察到当月自己的情绪变化，且画风可爱的日历能为手机桌面增添一抹亮色。



图 19 情绪日历 UI 设计

2. 触摸卡片

触摸卡片将用户的私人定制化小怪兽放置于手机桌面，方便用户在不打开应用的情况下，就可以与小怪兽完成简单交互。

卡片上提供触摸引导，当用户按照引导触摸小怪兽时，小怪兽会做出相应肢体动作予以回应，同时，用户与小怪兽之间的“亲密能量值”也会增加。点击小怪兽，便可直接进入应用的聊天界面，与小怪兽完成更多交互。



图 20 触摸卡片 UI 设计

3 概要设计和详细设计

3.1 系统结构

3.1.1 产品简介

“自留地 SelfPlanet”是一款基于人工智能技术的心理疗愈软件，旨在为在日常生活中积累有负面情绪，但尚未达到专业心理治疗需求的年轻人提供一个独立的情感表达和疗愈的空间。“自留地”这一名称象征着用户可以在这个私密、温暖的小天地里畅所欲言，释放内心的情感，找到内心平静与自我满足的途径。

“探索你的内心宇宙”是自留地的slogan，它传达了该应用的初衷:帮助用户处理日常生活中的负面情绪，减轻社交压力，并为他们提供一个自我探索情感世界的平台。应用logo风格较为可爱，由官方小怪兽IP形象与星球元素组成，小怪兽代表着应用核心功能点AiMonster，星球寓意着“每个人都能找到独属于自己的小星球”，也暗合应用名称“SelfPlanet”。

3.1.2 核心功能说明

产品核心功能包含两个板块:“AiMonster”人工智能聊愈，和“小天地”熟人社交分享。

● AiMonster——人工智能自适应聊愈功能

AiMonster是一个接入AI聊天机器人Woebot的、可个性化定制的虚拟小怪兽IP形象。Woebot是基于行为认知疗法帮助用户改善心理焦虑和抑郁的一款人工智能聊天机器人，接入Woebot后，AiMonster可24小时给予用户聆听与陪伴，通过语音或文字的形式回应用户所抒发的情绪或困惑，同时也可借助白噪音、轻音乐等工具达到更优质的舒缓效果。

区别于传统AI聊天机器人，AiMonster被设计为可个性化定制的虚拟小怪兽形象。用户可根据自己的喜好定制小怪兽的身体形态、毛发颜色等，打造一款“独属于自己”的心理疗愈伙伴。在互动过程中，小怪兽除了提供言语上的慰藉，还会依据对话内容进行表情变化与肢体互动，如微笑、招手、拥抱等。每当用户向小怪兽倾诉情绪，小怪兽都会将情绪“吃掉”，并将坏情绪转换为用户的“情绪能量”。

● 小天地——熟人社交分享模块

小天地是一个专注于熟人社交，让用户能尽情释放分享欲的“情绪帖”广场。在编辑分享内容时，用户需先在“情绪转盘”处选择当日心情，代表该种心情的图案会显示于帖子最上方，让朋友们直观地了解到用户今天的情绪状态如何。选择心情后，用户即可借助文字、图片、画笔等自由编辑所感所想，畅所欲言。同时，用户也可在小天地中浏览其他密友的帖子，并进行点赞、评论等互动行为。

3.1.3 产品系统架构图

对全局功能模块进行进一步细化，我们把自留地SelfPlanet的主要功能设计分为三个模块: AiMonster、小天地、装扮。

在“**AiMonster**”板块，用户可以与自己的个性化小怪兽进行互动，对它的外观进行个性化装扮，与它聊天对话，选择疗愈工具进行疗愈，对他投喂“情绪”使其恢复能量;在“**小天地**”板块，用户可以记录自己每日的情绪状态，在“今日心情”页面记录下自己的情绪，也可以前往广场，看看“大家的一天”。在“**装扮**”板块，用户可以浏览、收藏或购买小怪兽的装扮，从而打造个性化定制的小怪兽。

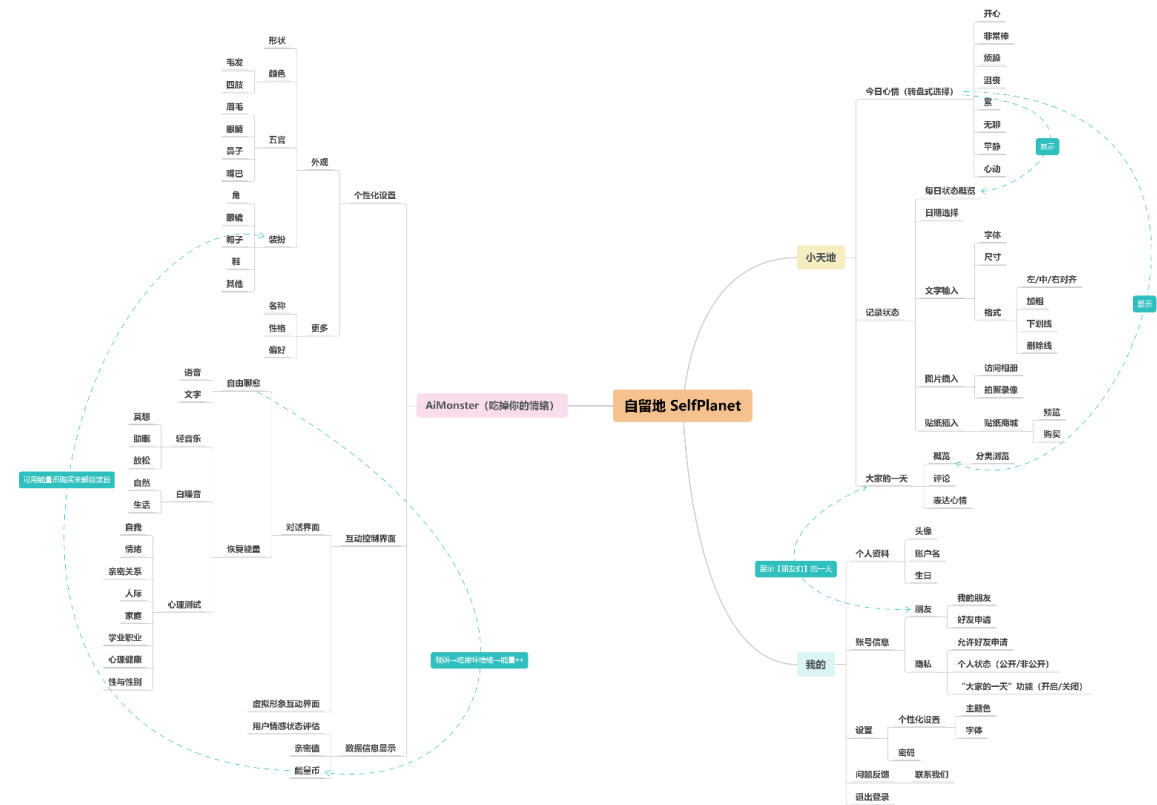


图 21 “自留地”产品信息架构图

3.1.4 接口设计

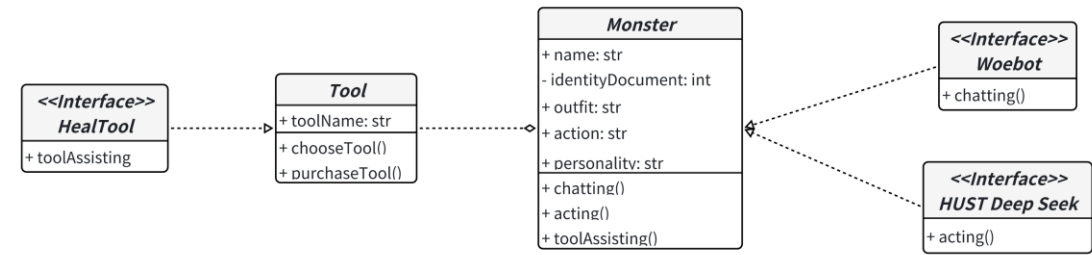


图 22 三种接口使用类图说明

本设计模式中共涉及三个接口，分别是Woebot、华中科技大学 DeepSeek 和 HealTool。

1. Woebot

Woebot是一款基于行为认知疗法（Cognitive Behavior Therapy），帮助用户改善心理焦虑和抑郁的人工智能聊天机器人，它通过分享心理学理论和引导用户改变自己的语言引导用户平和客观地接纳生活中的负面事件，帮助用户获得拥有好心情的能力。

Woebot接口被接入Monster抽象类，用于用户与小怪兽的聊天功能。用户向小怪兽输入文本信息后，文本信息将输入至Woebot端机器人，Woebot将根据文本输入做出语言信息反馈，并以小怪兽说话的形式输出给用户，进而实现用户与小怪兽的聊天功能。

2. 华中科技大学 DeepSeek

华中科技大学 DeepSeek 是基于 DeepSeek 训练的智能大模型，拥有跨领域的知识和语言理解能力，能够基于自然对话方式理解与执行任务。

该接口同样被接入Monster抽象类，用于帮助小怪兽理解用户对话内容并做出肢体动作反馈。用户和Woebot的文本信息会实时输入至 DeepSeek，DeepSeek将对对话内容做出整体理解，并给出肢体动作建议。小怪兽内置程序将根据DeepSeek 所提供的动作建议，对用户做出动作反馈，如拥抱、擦眼泪等。

3. HealTool

Heal Tool接口全面地集成了当前市面上较为常用的聊愈类工具，包括海量的白噪音音频、助眠音频、放松类播客等。Heal Tool接口被接入Tool抽象类，用户选择使用“疗愈工具”时，该接口将为用户提供大量疗愈类工具以供选择。

3.2 类图等

3.2.1 实体类与抽象类

本系统中的核心实体类有:User(用户)、Outfit(装扮)、Post(帖子)、Currency(货币)等。其中装扮类包含Appearance(外观)、Top(上装)、Bottom(下装)、Footwear(鞋)与Accessory(配饰)五个具体策略类。

本系统中的抽象类有:Monster(小怪兽)、Commodity(货物)、Tool(疗愈工具)和Factory(工厂)。其中小怪兽类包含DecoMonster、MonsterRex、MonsterFelix和MonsterHazel四个实体类，工厂类包含OfficialFactory和用户Factory两个实体工厂类。

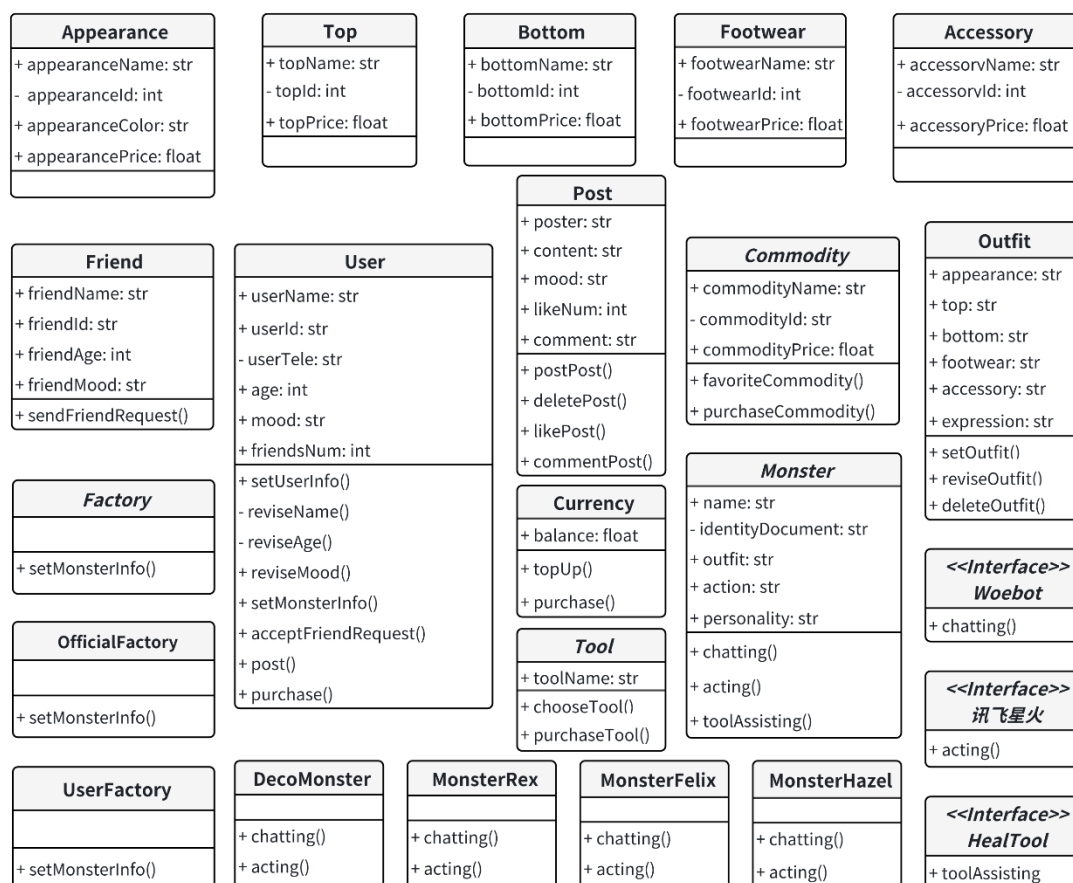


图 23 UML 类图汇总

3.2.2 商品类与策略模式

策略模式被用于定义商品类。由于商品包含 Appearance（外观）、Top（上衣）、Bottom（下装）、Footwear（鞋）和 Accessory（配饰）五个相似的类，故

将它们定义为五个**具体策略**，并为他们设置了一个能够实现所有类功能的**策略接口** **Commodity**（商品）。**Commodity** 抽象类中所包含的全部操作算法，即“收藏商品”和“购买商品”，均可运用于五个具体策略，且算法可独立于具体策略端的变化。

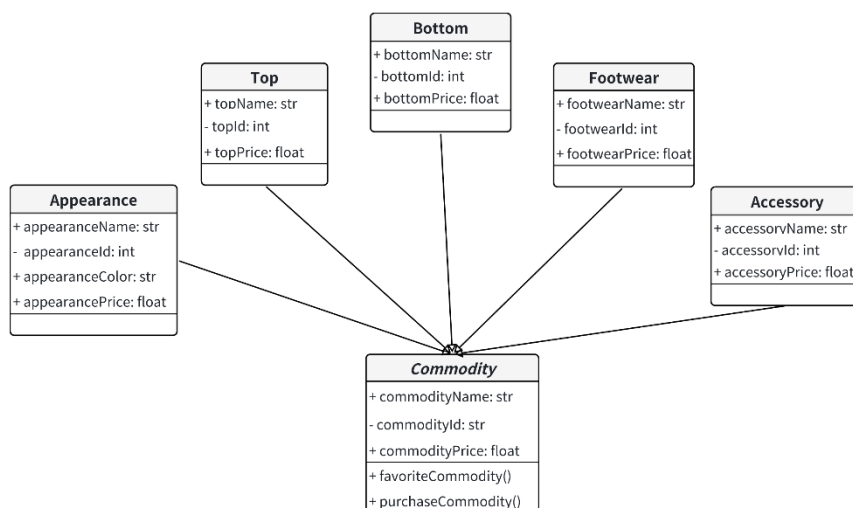


图 24 商品类策略模式

Commodity 抽象类中所包含的全部操作算法，即 **favoriteCommodity()** (收藏商品)和 **purchaseCommodity()** (购买商品)，均可运用于五个具体策略，且算法可独立于具体策略端的变化。

3.2.3 工厂类与工厂模式

工厂方法模式被运用于小怪兽的创建，它包含一个抽象工厂类 **Factory**，和两个具体工厂类 **UserFactory** 和 **OfficialFactory**，其中“**UserFactory**(用户工厂)”被用于根据用户需求自主创建定制化小怪兽，而“**OfficialFactory**(官方工厂)”被用于创建官方预设的小怪兽——**Rex**，**Felix**和**Hazel**。

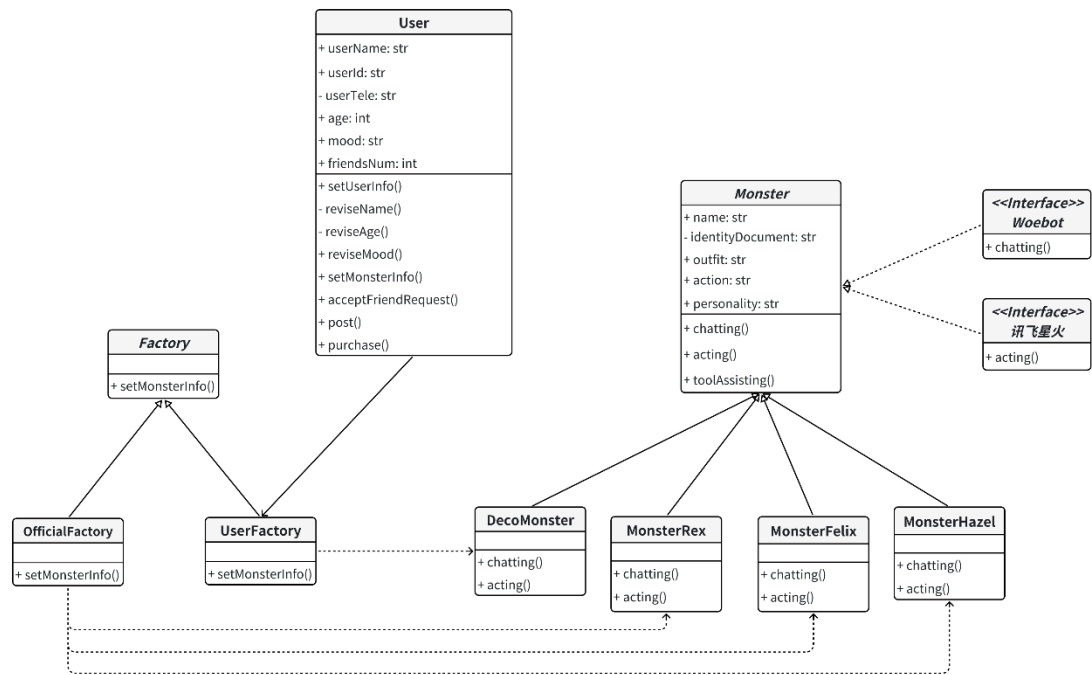


图 25 工厂类工厂模式

用户在需要一个新的“Monster”对象时，会根据不同需求选择使用用户工厂或官方工厂。无论选择哪一工厂，工厂均会调用 `setMonsterInfo()` 方法设置小怪兽信息，构建并返回一个正确配置的“Monster”对象。

3.3 关键数据结构定义

3.3.1 用户

1. 实体类图

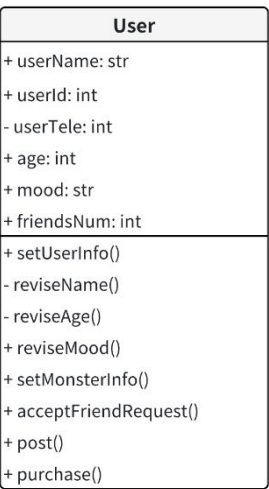


图 26 User 实体类类图

2. 属性设计

表 6 User 类属性设计

属性	类型	说明	初始值
用户名	String	用户昵称 2-15 位字符串	
用户 ID	String	每位用户唯一的 ID 号 8 位字符串	
电话号码	String	用户电话号码 11 位字符串	
年龄	int	用户年龄	0
心情	String	用户当前心情	"愉悦"
好友数	int	依据用户好友个数累加	0

3. 方法设计

表 7 User 类方法设计

操作	说明
设置用户信息 ()	reviseName(); reviseAge();
更改用户名 ()	String userName = 用户名; if (更改用户名) { age = input.next(); } return age ;
更改年龄 ()	int age = 年龄; if (更改年龄) { age = input.nextInt(); } return age ;
更改心情 ()	String mood = 心情; if (更改心情) { mood = input.next(); } return mood ;

设置怪兽信息 ()	<pre>String monsName = 怪兽名称; if (设置怪兽信息) { monsName = input.next(); } return monsName;</pre>
接受好友申请 ()	<pre>int friendNum = 用户原始好友数; if (用户接受了一个好友申请) { friendNum += 1; } return friendNum;</pre>
发布帖子 ()	<pre>ArrayList post = 帖子内容列表; String content = 空字符串; String mood = 用户当前心情; if (用户发布帖子) { content = input.next(); post.add(content); post.add(mood); } return post;</pre>
购买商品 ()	<pre>ArrayList commodities = 拥有商品列表; float balance = 金币余额; String commodityName= 将购买的商品名称; float commodityPrice = 将购买的商品价格; if (用户购买商品) { commodities.add(commodityName); balance -= commodityPrice; } return commodities, balance;</pre>

3.3.2 AiMonster 小怪兽

1. 实体类图

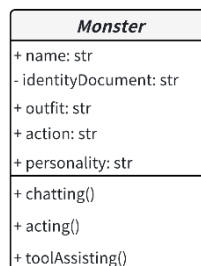


图 27 AiMonster 实体类图

2. 属性设计

表 8 AiMonster 类属性设计

属性	类型	说明	初始值
怪兽名称	String	怪兽名称 2-15 位字符串	
怪兽 ID	String	每只怪兽唯一的 ID 号 8 位字符串	
怪兽装扮	String	当前装扮名称	
怪兽动作	String	怪兽当前动作	"微笑"
怪兽性格	String	怪兽性格	"温柔"

3. 方法设计

表 9 AiMonster 类方法设计

操作	说明
聊天 ()	<pre> interface woebot = woebot 接口; String contentSend = 发送的聊天内容; String contentReceive = 接收的聊天内容; if (用户与小怪兽聊天) { contentSend = input.next(); contentReceive.woebot(contentSend); } return contentReceive; </pre>
做出行为 ()	<pre> interface xinghuo = 讯飞星火接口; String contentSend = 发送的聊天内容; String action = 怪兽的行为; if (聊天时小怪兽做出行为反馈) { contentSend = input.next(); action.woebot(contentSend); } return action; </pre>
使用聊愈工具 ()	<pre> interface healTool = Heal Tool 接口; ArrayList tools = 聊愈工具列表; if (用户需选择聊愈工具) { tools.add(healTool); } return tools; </pre>

3.3.3 装扮

1. 实体类图

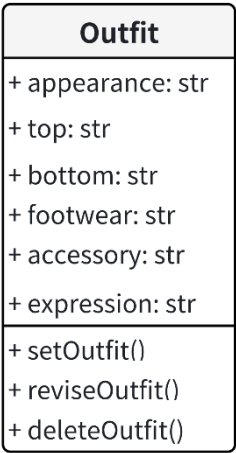


图 28 装扮实体类图

2. 属性设计

表 10 装扮类属性设计

属性	类型	说明	初始值
外形	String	小怪兽整体外形，包括轮廓样式和颜色等	"圆滚滚黄毛"
上装	String	小怪兽上半身服饰	
下装	String	小怪兽下半身服饰	
脚部	String	小怪兽脚部装扮	
配饰	String	小怪兽配饰，如角、尾巴，或眼镜、帽子等	
表情	String	小怪兽面部表情	"微笑"

3. 方法设计

表 11 装扮类方法设计

操作	说明
----	----

设置装扮 ()	<pre> ArrayList outfits = 全体装扮列表（二维列表）； ArrayList newOutfit = 新装扮列表； String appearance = 装扮外型； String top = 装扮上衣； String bottom = 装扮下装； String footwear = 装扮鞋子； String accessory = 装扮配饰； if（设置新装扮）{ newOutfit.add(appearance); newOutfit.add(top); newOutfit.add(bottom); newOutfit.add(footwear); newOutfit.add(accessory); outfits.add(newOutfit); } return outfits; </pre>
修改装扮 ()	<pre> ArrayList reviseOutfit = 需更改的装扮列表； String appearance = 装扮外型； String top = 装扮上衣； String bottom = 装扮下装； String footwear = 装扮鞋子； String accessory = 装扮配饰； if（修改装扮）{ reviseOutfit[0] = appearance; reviseOutfit[1] = top; reviseOutfit[2] = bottom; reviseOutfit[3] = footwear; reviseOutfit[4] = accessory; } return reviseOutfit; </pre>
删除装扮 ()	<pre> ArrayList outfits = 全体装扮列表（二维列表）； ArrayList deleteOutfit = 需删除的装扮列表； if（删除装扮）{ outfits.delete(deleteOutfit); } return outfits; </pre>

3.3.4 商品

1. 实体类图

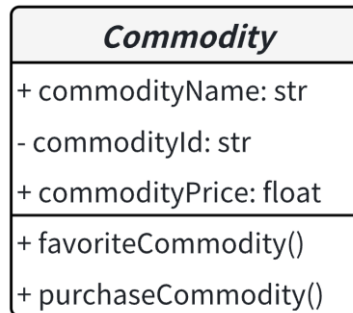


图 29 商品实体类图

2. 属性设计

表 12 商品类属性设计

属性	类型	说明	初始值
商品名称	String	2-15 位字符串	
商品 ID	String	每个商品唯一的 ID 号 6 位字符串	
商品价格	float	购买商品所需金币数量	0.0

3. 方法设计

表 13 商品类方法设计

操作	说明
收藏商品 ()	ArrayList favorite = 收藏列表; String commodityName = 装扮名称; if (收藏商品) { favorite.add(commodityName); } return favorite;
购买商品 ()	ArrayList commodities = 拥有商品列表; float balance = 金币余额; String commodityName = 将购买的商品名称; float commodityPrice = 将购买的商品价格; if (用户购买商品) { commodities.add(commodityName); balance -= commodityPrice; } return commodities, balance;

3.3.5 帖子

1. 实体类图

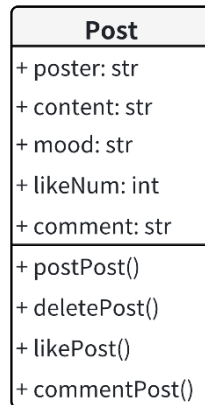


图 30 帖子类实体类图

2. 属性设计

表 14 帖子属性设计

属性	类型	说明	初始值
发布者名称	String	2-15 位字符串	
发布内容	String	帖子具体内容	
发布者心情	String	发布者当前心情	"愉悦"
点赞数	int	帖子收到的点赞数	0
评论	String	帖子下的评论内容	

3. 方法设计

表 15 帖子类方法设计

操作	说明
发布帖子 ()	ArrayList posts = 总帖子列表; ArrayList post = 帖子内容列表; String poster = 发布者名称; String content = 空字符串; String mood = 用户当前心情; if (用户发布帖子) { content = input.next(); post.add(poster); post.add(content); }

	<pre> post.add(mood); posts.add(post) } return post; </pre>
删除帖子（）	<pre> ArrayList posts = 总帖子列表; ArrayList deletePost = 需删除的帖子内容列表; if （用户删除帖子）{ posts.delete(deletePost); } return posts; </pre>
点赞帖子（）	<pre> bool like = 点赞状态; if（点赞帖子）{ like = True; } </pre>
评论帖子（）	<pre> ArrayList comments = 总评论列表; String comment = 评论内容; if（评论帖子）{ comment = input.next(); comments.add(comment); } return comments; </pre>

3.3.6 好友

1. 实体类图

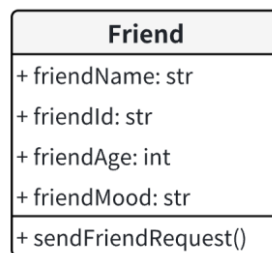


图 31 好友类实体类图

2. 属性设计

表 16 好友类属性设计

属性	类型	说明	初始值
好友用户名	String	好友昵称 2-15 位字符串	
好友 ID	String	每位用户唯一的 ID 号 8 位字符串	

好友年龄	int	好友年龄	0
好友心情	String	好友当前心情	"愉悦"

3. 方法设计

表 17 好友类方法设计

操作	说明
发送好友申请（）	<pre>String request = 申请内容; if（发送好友申请）{ request = input.next(); } return request;</pre>

3.4 关键算法设计

3.4.1 用户登录算法

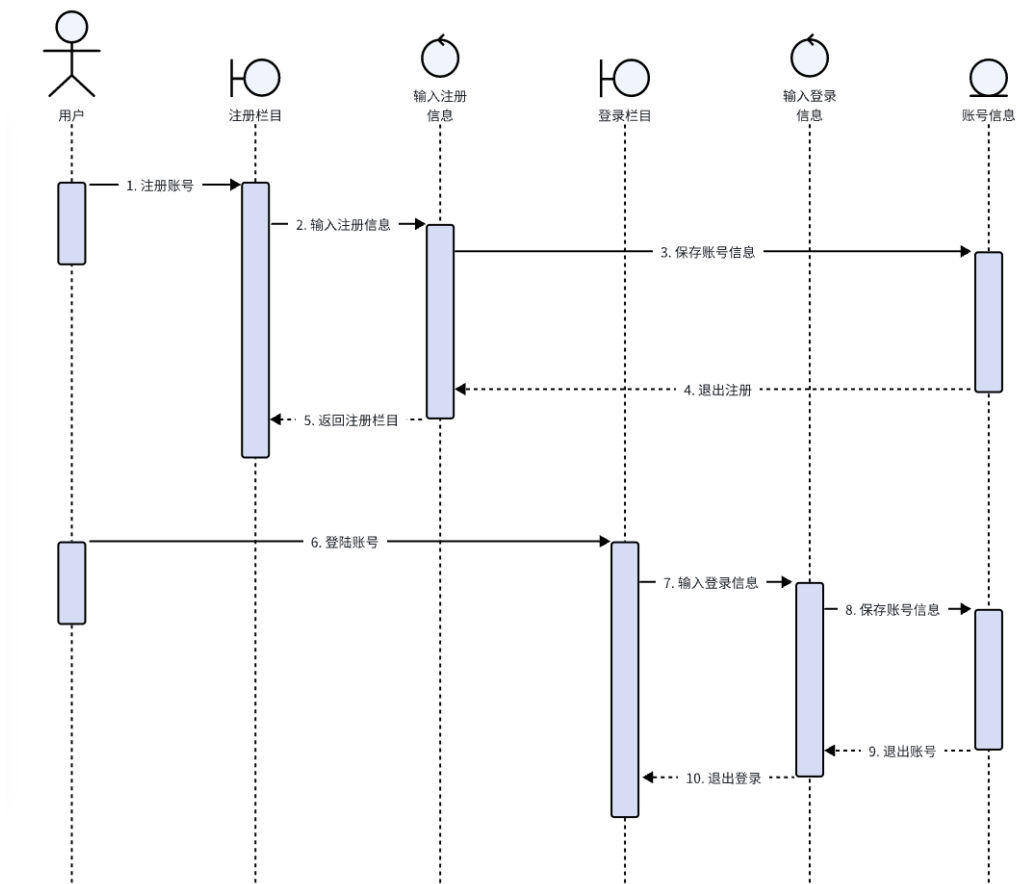


图 32 注册登录序列图

- 1)用户在注册栏目选择注册账号
- 2) 边界类将注册账号的消息传递给控制类
- 3) 控制类将输入注册信息的消息传递给实体类
- 4) 实体类将输入注册信息反馈给控制类
- 5) 控制类将消息反馈给边界类，用户得知账号已经注册。
- 6) 用户登陆账号
- 7) 边界类将登陆账号的消息传给控制类
- 8) 控制类将输入登陆信息的信息传给实体类
- 9) 实体类将输入登陆信息反馈给控制类
- 10) 控制类将消息反馈给边界类，用户得知账号已登陆。

3.4.2 用户交易算法

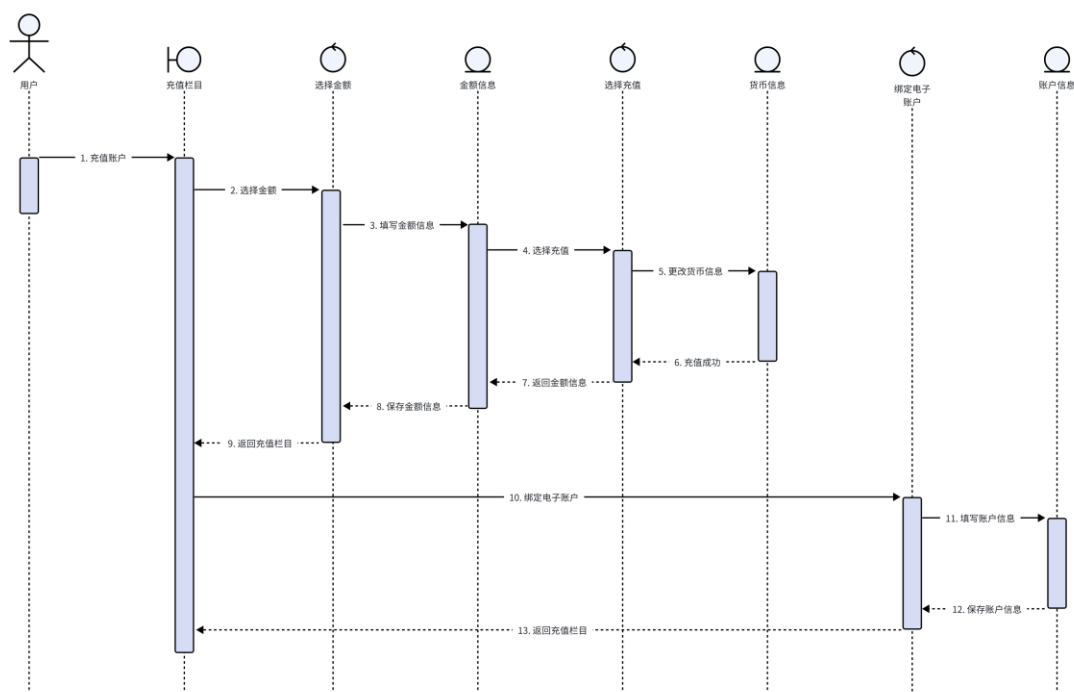


图 33 交易系统序列图

- 1) 用户选择充值账户
- 2) 边界类将选择金额的消息传递给控制类
- 3) 控制类将填写金额信息的信息传递给实体类
- 4) 用户选择充值，将选择充值的消息传递到控制类

- 5) 控制类将更改货币信息的信息传递到货币信息实体类
- 6) 货币信息实体类将充值成功的消息反馈给选择充值的控制类
- 7) 选择充值的控制类将金额信息反馈给金额信息实体类
- 8) 实体类将金额信息反馈给选择金额控制类
- 9) 控制类将消息反馈给边界类，用户得知充值成功
- 10) 边界类将绑定电子账户的消息传递给控制类
- 11) 控制类将填写账户信息的信息传递给实体类
- 12) 实体类将账户信息反馈给控制类
- 13) 控制类将消息反馈给边界类，用户得知账户绑定成功

3.4.3 聊天系统算法

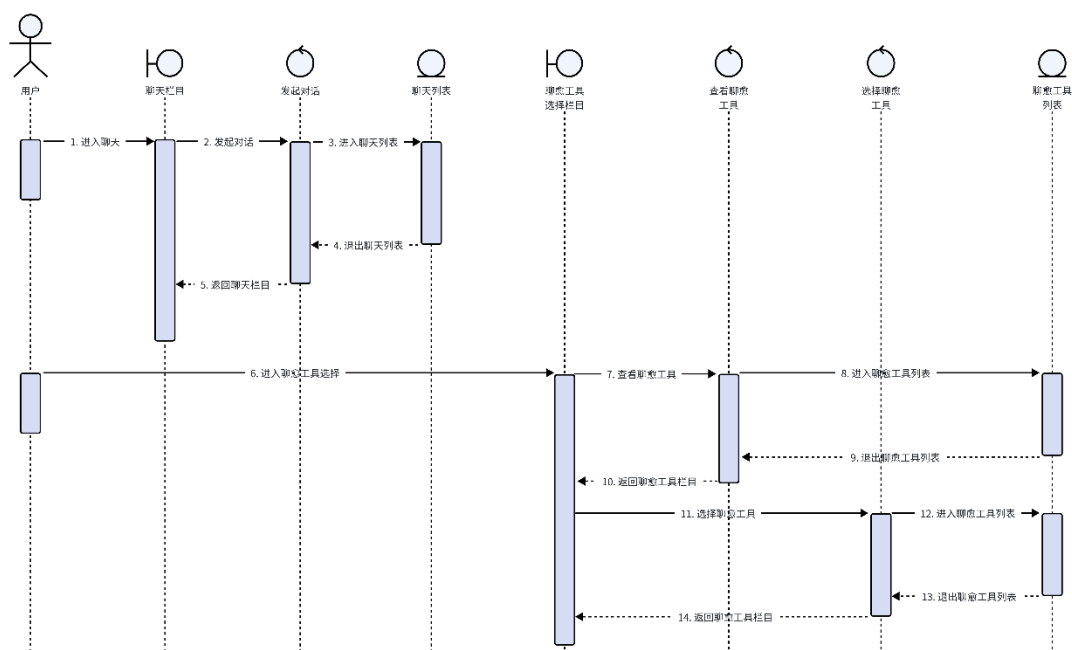


图 34 聊天系统序列图

- 1) 用户进入聊天栏目
- 2) 边界类将发起对话的消息传递给控制类
- 3) 控制类将进入聊天列表的消息传递给实体类
- 4) 实体类将退出聊天列表的消息反馈给控制类
- 5) 将消息反馈给边界类，用户得知聊天结束

- 6) 用户进入疗愈工具选择栏目
- 7) 边界类将查看疗愈工具的消息传递给控制类
- 8) 控制类将进入疗愈工具列表的消息传递给实体类
- 9) 实体类将退出疗愈工具列表的消息反馈给控制类
- 10) 控制类将消息反馈给边界类，用户得知疗愈工具的查看结束
- 11) 边界类将选择疗愈工具的消息传递给控制类
- 12) 控制类将进入疗愈工具列表的消息传递给实体类
- 13) 实体类将退出疗愈列表的消息反馈给控制类
- 14) 控制类将消息反馈给边界类，用户得知疗愈工具体验结束

3.4.4 消息发布算法

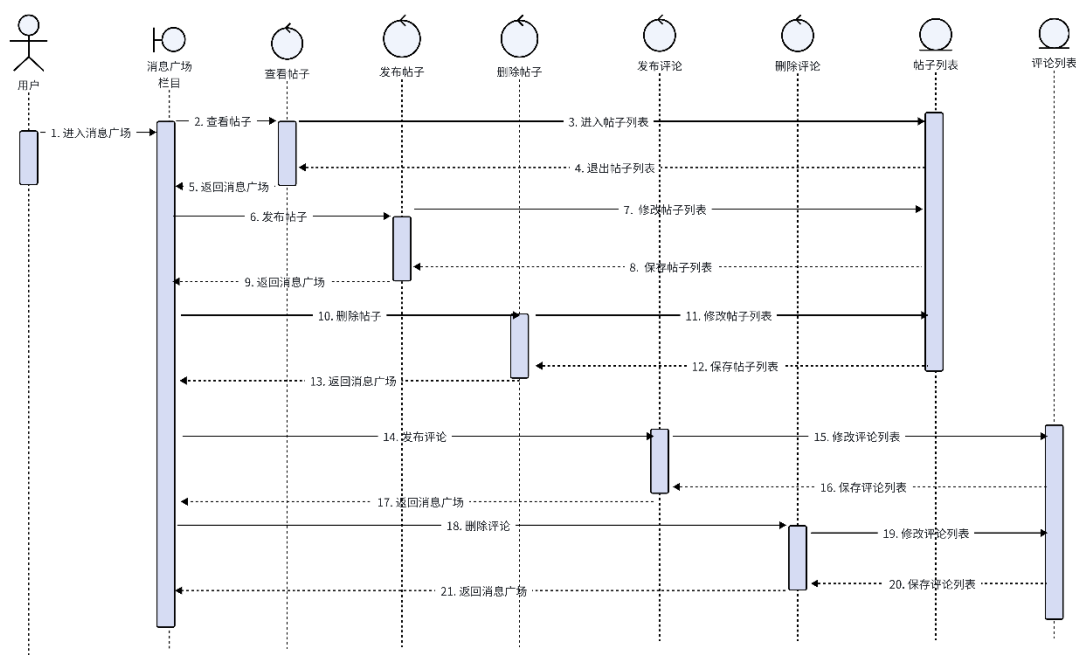


图 35 消息广场

4 实现与测试

4.1 实现环境与代码管理

4.1.1 实现环境

1. 前端技术栈

- 框架：React.js
- 样式解决方案：CSS-in-JS
- 状态管理：React Hooks (useState, useEffect)

2. 后端技术栈

- 服务器：Node.js
- API 实现：RESTful API 设计

3. 开发环境

- 操作系统：Windows 10
- 开发工具：Cursor IDE
- 版本控制：Git
- 包管理器：npm/yarn

4. 项目结构

```
project/
├── src/
│   ├── components/
│   │   ├── MoodWheel.js
│   │   ├── ChatInterface.js
│   │   └── MessageList.js
│   ├── styles/
│   │   └── main.css
│   └── App.js
├── server/
│   ├── index.js
│   └── routes/
│       └── chat.js
├── public/
│   └── index.html
└── package.json
```

5. 依赖项

```
{
  "dependencies": {
    "react": "^18.x",
    "react-dom": "^18.x",
    "express": "^4.x",
    "node-fetch": "^2.x"
  },
  "devDependencies": {
    "webpack": "^5.x",
    "babel": "^7.x",
    "eslint": "^8.x"
  }
}
```

4.1.2 代码管理

本项目在 GitHub 平台管理。开发部分由余易昕、江思怡两名同学完成，故有两名贡献者。整个项目代码一共有 25 条 commits，具体如图 36 Github 平台

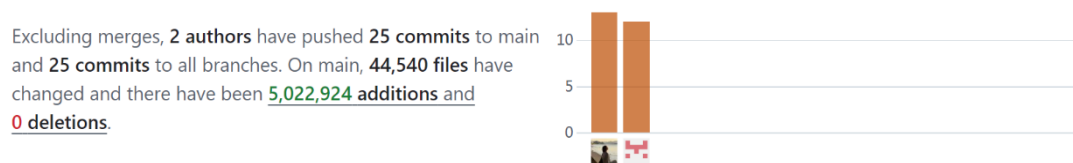


图 36 Github 平台代码统计

代码统计图 36所示。

代码推送记录如图 37所示。

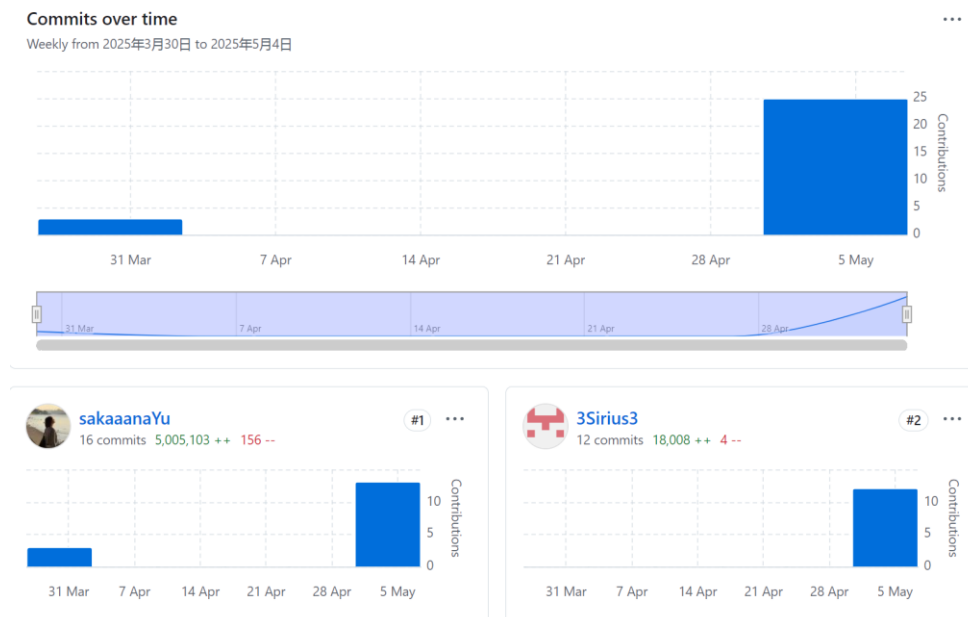


图 37 代码推送记录

4.2 关键函数说明

4.2.1 “小天地”部分心情转盘算法

心情转盘算法通过精确的数学计算和动画控制，实现了一个流畅的心情选择轮盘。为了增加趣味性，轮盘会至少旋转两圈，最后通过缓动函数实现自然的减速效果。

6. 基础角度计算

```
const anglePerMood = 360 / moods.length; // 每个心情占用的角度
const rotation = -90; // 初始旋转角度，使第一个选项位于顶部
```

- 表示心情的五个 icon 均匀分布在轮盘上
- 确保第一个 icon 位置在轮盘顶部

```
const randomDegrees = Math.floor(Math.random() * 360) + 720;
```

- $\text{Math.random()} * 360$: 生成0-359度的随机角度
- $+ 720$: 确保至少旋转两圈（720度）

7. 选中心情计算

```
const selectedIndex = Math.floor(((360 - (randomDegrees % 360)) / anglePerMood));
const selectedMood = moods[selectedIndex];
```

- $\text{randomDegrees} \% 360$: 获取最终停止角度（0-359）
- $360 - (\text{randomDegrees} \% 360)$: 反转角度，因为轮盘是顺时针旋转

- / anglePerMood: 将角度转换为心情索引
- Math.floor(): 向下取整, 确保得到有效的索引

8. 动画效果实现

```
wheelRef.current.style.transition = 'transform 3s cubic-bezier(0.17, 0.67, 0.83, 0.67)';
wheelRef.current.style.transform = `rotate(${randomDegrees}deg)`;
```

- transition: 定义动画持续时间和缓动函数
- 3s: 动画持续3秒
- cubic-bezier(0.17, 0.67, 0.83, 0.67): 自定义缓动函数
 - 创造先快后慢的效果, 使轮盘旋转看起来更自然
 - 模拟物理世界中的减速效果

9. 轮盘旋转函数实现

```
const spinWheel = () => {
  // 禁用按钮, 防止重复点击
  setIsSpinning(true);

  // 计算随机旋转角度
  const randomDegrees = Math.floor(Math.random() * 360) + 720;

  // 计算选中的心情
  const selectedIndex = Math.floor(((360 - (randomDegrees % 360)) / anglePerMood));
  const selectedMood = moods[selectedIndex];

  // 应用旋转动画
  wheelRef.current.style.transition = 'transform 3s cubic-bezier(0.17, 0.67, 0.83, 0.67)';
  wheelRef.current.style.transform = `rotate(${randomDegrees}deg)`;

  // 动画结束后更新状态
  setTimeout(() => {
    setIsSpinning(false);
    setSelectedMood(selectedMood);
  }, 3000);
};
```

4.2.2 “聊愈”部分大模型 API 调用算法

本部分采用异步方式处理请求。算法首先构建一个标准的HTTP请求, 包含必要的请求头和消息体。使用try-catch结构确保错误处理, 如果发生异常, 会返

回一个友好的错误提示。

```
const handleSendMessage = async (message) => {
  try {
    const response = await fetch('/api/chat', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ message })
    });
    const data = await response.json();
    return data.response;
  } catch (error) {
    return "抱歉，我现在无法回应。请稍后再试。";
  }
};
```

4.3 测试计划和测试用例

软件测试常用的方法有静态测试、动态测试、黑盒测试与白盒测试。

静态测试方法指软件代码的静态分析测验，此类过程中应用数据较少，主要过程为通过软件的静态性测试（即人工推断或计算机辅助测试）测试程序中运算方式、算法的正确性，进而完成测试过程。

计算机动态测试的主要目的为检测软件运行中出现的问题，较静态测试方式相比，其被称为动态的原因即为其测试方式主要依赖程序的运用，主要为检测软件中动态行为是否缺失、软件运行效果是否良好。

黑盒测试即为将软件测试环境模拟为不可见的“黑盒”。通过数据输入观察数据输出，检查软件内部功能是否正常。测试展开时，数据输入软件中，等待数据输出。数据输出时若与预计数据一致，则证明该软件通过测试，若数据与预计数据有出入，即便出入较小亦证明软件程序内部出现问题，需尽快解决。

白盒测试相对于黑盒测试而言具有一定透明性，原理为根据软件内部应用、源代码等对产品内部工作过程进行调试。测试过程中常将其与软件内部结构协同展开分析，最大优点即为其能够有效解决软件内部应用程序出现的问题，测试过程中常将其与黑盒测试方式结合，当测试软件功能较多时，白盒测试法亦可对此类情况展开有效调试。



我们采用黑盒测试的方法测试每个模块的功能是否能够正确实现预期目标。由于代码能力的限制，本次开发仅简单实现“聊愈”和“小天地”两个板块的功

能，故仅对这两个板块进行黑盒测试。

4.3.1 “聊愈”部分对话功能测试

聊愈部分对话功能的测试样例、预期测试结果与实际测试结果如表 18所示。

表 18 “聊愈”部分对话功能测试表

测试样例	预期结果	实际结果截图
用户选中文本输入框，向小怪兽发送消息。	经过一段时间的加载，小怪兽通过调用的大模型 API 为用户提供回复。	
用户用鼠标滑动消息记录。	用户可以浏览历史消息记录。	

4.3.2 “小天地”部分日记功能测试

小天地部分日记的测试样例、预期测试结果与实际测试结果如表 19所示。

表 19 “小天地” 部分日记功能测试

测试样例	预期结果	实际结果截图
用户转动情绪轮盘，选择某种情绪。	情绪被成功选中，并高亮显示在轮盘下方。	
用户选中文本输入框，输入日记内容，完成后点击“保存日记”。	日记被成功保存，文本内容、情绪icon和日期一同显示在下方日记记录栏。	
用户用鼠标滑动日记记录。	用户可以浏览所有的日记内容。	

4.4 结果分析

由于代码能力限制，本次开发并未完全按照原型图实现所有功能，但两大核心、基本功能可以正常实现：

本次开发可以实现正常的“聊愈”功能。消息发送出去后成功到达所调用大模型 API 的服务器，经过处理后，相应回复被成功返回。

本次开发可以实现正常的“小天地”功能。用户可以丝滑地通过转盘选择情绪，并对心情进行相应记录，也可以浏览历史情绪。

5 总结

5.1 用户反馈

本次开发的产品已经过产品试用，并收获用户反馈如下：

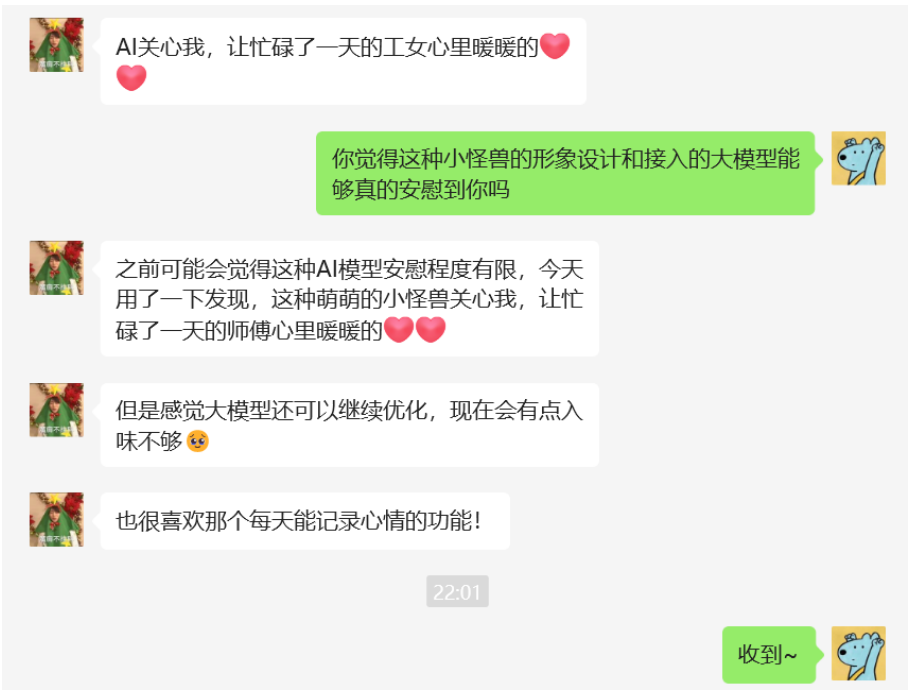


图 38 用户反馈 1

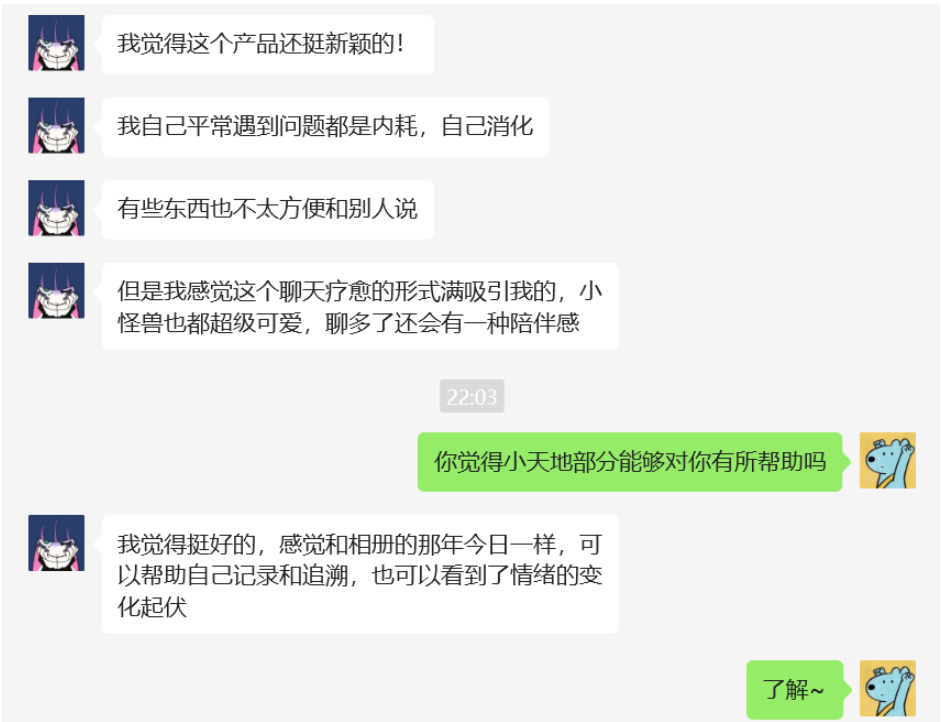


图 39 用户反馈 2

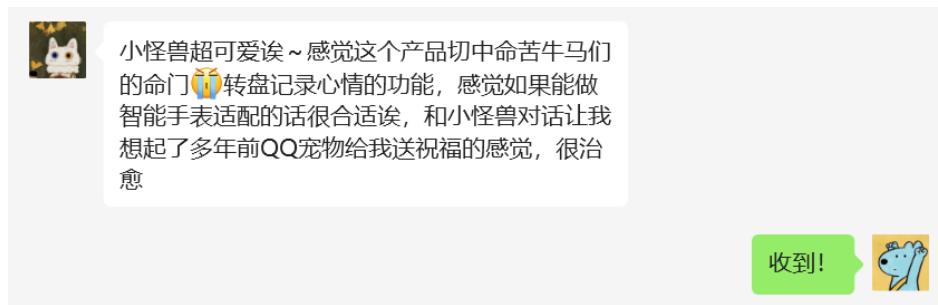


图 40 用户反馈 3

5.2 全文总结

“自留地 SelfPlanet”是一款针对 18 - 35 岁年轻人开发的基于 AI 的心理疗愈软件,旨在为用户提供一个安全、私密的情感表达空间,帮助他们缓解负面情绪,实现情感疗愈和自我探索。项目团队由四名成员组成,通过前期的市场调研、竞品分析以及对目标用户需求的深入理解,明确了产品的定位和核心功能,并采用软件工程的思想和方法,完成了从需求分析到设计、编码、测试以及最终实现的全过程。

5.2.1 功能实现

项目成功实现了两大核心功能模块:

1. “AiMonster” 人工智能聊愈

用户可以与虚拟小怪兽进行互动,小怪兽接入了大模型,能够 24 小时聆听用户倾诉,通过语音或文字回应情绪或困惑

2. “小天地” 日记记录 :

用户在记录每日情绪状态时,需先在“情绪转盘”选择当日心情,然后可借助文字、图片、画笔等编辑所感所想。

5.2.2 技术与设计

在技术选型上,前端采用 React.js 框架,结合 CSS-in-JS 进行样式设计,利用 React Hooks 进行状态管理;后端基于 Node.js 搭建服务器,设计 RESTful API 实现前后端通信。开发过程中,团队成员熟练运用了 figma、墨刀等原型图设计工具,以及 StarUML 等 UML 设计工具,完成了详细的系统架构设计、类图设计等,为项目的顺利开发提供了有力保障。

在设计方面,紧扣“探索你的内心宇宙”核心概念,以杏黄色和橙棕色作为主

视觉色系，搭配色彩丰富的小怪兽形象，营造出温暖、明快且富有亲和力的界面风格。同时，采用了扁平动画风格的 icon 及交互 UI 设计，让页面更加生动可爱，为用户带来亲切感。

5.2.3 困难与解决方案

在原型图设计阶段，初期设计的小怪兽形象与整体界面风格融合不够自然，影响了用户体验。经过多次讨论和修改，团队重新调整了小怪兽的设计细节，使其更加贴合产品的温暖、治愈风格，并优化了界面布局和交互流程，提高了产品的易用性和美观度。

在开发方面，团队成员缺乏 AI 开发经验，在 AI 模型的本地化适配、多模态交互以及隐私数据加密等方面面临挑战。为此，成员们通过查阅大量资料、学习相关技术文档，并积极寻求专业技术支持，逐步攻克了这些难题。虽然功能开发得不够完善，但对于团队成员而言也是一次极大的突破。

6 体会

1. 余易昕同学

➤ 分工

前期市场及竞品调研、UML 图设计、原型图设计、前端开发、报告撰写与整合。

➤ 体会

通过参与该项目，我对心理疗愈类应用的市场需求有了更深入的了解。了解到用户对于一个能够提供个性化、私密性高且功能丰富的心理疗愈应用有着强烈的需求。这让我认识到，在产品设计过程中，必须始终以用户为中心，关注用户的真实需求和痛点，才能设计出真正有价值的产品。

在 UML 图设计阶段，我学会了如何运用 UML 工具进行系统设计，将抽象的需求转化为具体的系统架构和模块划分。这个过程不仅锻炼了我的逻辑思维能力，也让我对软件工程的整体流程有了更清晰的认识。通过不断地优化和完善 UML 图，我能够更好地理解系统各部分之间的关系和交互流程，为后续的开发工作提供了明确的指导。

前端开发是本次项目中我投入精力最多的一部分。在这个过程中，我熟练掌握了 React.js 框架的使用方法，并深入了解了前端开发的各种技术和工具。通过解决实际开发中遇到的各种问题，我的前端开发技术得到了显著提升，同时也积累了丰富的项目经验。

看到曾经只是一个构想的东西一步一步变得具体，是一件很有成就感的事情。我们感谢 SelfPlanet 的诞生，在完善 SelfPlanet 的同时，我们也在完善我们自己。虽然小组成员们已经付出了很多，但 SelfPlanet 目前也只是一个简单的空中楼阁，仍然有待继续完善。毕竟，不管什么事，什么人，在自我完善的这条路上，都是走不到终点的。我们祝你，在自我完善的道路上，也能找到一个温暖治愈的小天地，能让这条路走起来轻松一些。

2. 江思怡同学

➤ 分工

前期市场及竞品调研、UML 图设计、原型图设计、前端开发。

➤ 体会

SelfPlanet 的诞生源于我们对“情感疗愈”的共鸣。在讨论中，团队成员敞开心扉，发现许多人需要这样一个温暖、私密的空间来安放情绪。于是，我们构想了一个治愈系的小天地，用“小怪兽”的形象承载用户的喜怒哀乐。

从概念到雏形，每一步都充满挑战，但也让我们不断成长。我们学会了如何平衡功能与情感，如何让技术服务于人性需求。尽管 SelfPlanet 仍有待完善，但这个过程让我们明白：好的产品不仅是功能的集合，更是情感的载体。

最终，SelfPlanet 不仅是一个工具，也成了我们自我疗愈的旅程。愿每个寻求成长的人，都能找到属于自己的温暖角落。

3. 李昕豫同学

➤ 分工

前期市场及竞品调研、UML 图设计、原型图设计、报告整合。

➤ 体会

在“自留地 SelfPlanet”项目中，我主要负责了前期市场及竞品调研、部分 UML 类图设计和原型图设计与最终报告整合。这段经历最大的收获是，软件工程中每一个环节的决策都需兼顾用户需求与技术落地的可行性，而设计者的角色更像是在理想与现实的夹缝中寻找平衡点。

市场调研的本质是通过数据反推需求，而非单纯堆砌信息。我们通过分析“心岛日记”用户对 AI 回复模板化的吐槽，团队决定引入 Woebot 框架并加入表情动作反馈，以提升交互真实感。调研过程中，团队多次召开需求对齐会，用可视化图表呈现用户画像和功能优先级，最终砍掉存在隐私风险的“陌生人连麦”功能，转而聚焦熟人社交模块的设计。

UML 类图设计阶段，如何将抽象需求转化为清晰的系统架构是一大挑战。初期设计时，我曾为“小怪兽”设定了复杂的继承关系，导致类图冗余，开发组反馈此类设计可能增加代码维护成本。通过与后端同学反复讨论，我们重新梳理核心场景，将“装扮系统”独立为策略模式，减少类之间的强耦合。Outfit 类最初与小怪兽属性混杂，调整后其外观、配饰等字段被剥离为独立模块，通过组合模式实现灵活扩展。工具辅助也发挥了重要作用，使用 StarUML 检测类依赖关系时，发现 User 与 Post 的双向关联可能引发循环调用，最终改为单向依赖。后续原型图设计过程中，团队首先制定了统一的 UI 规范文档，明确配色、字体和动效时长，既保障了设计一致性，也降低了开发适配的复杂度。

回顾整个项目，最大的收获是学会在各种约束中创造价值。未来希望更早介入技术实现环节，通过持续沟通减少设计与开发的摩擦，让每一个创意都能扎实地生长为可触达用户的功能。

4. 陈芊聿同学

➤ 分工

前期市场及竞品调研、UML 图设计、原型图设计、报告撰写。

➤ 体会

在本次项目的开发过程中,我深刻体会到“技术服务于人”产品观的重要性。

项目伊始,我们致力于探索“心灵疗愈”的新的可能性,构思了一个以克服日常小情绪为目的的疗愈助手,以提供情感支持回应用户的情绪需求。初期设计时,我曾为聊天模块设计了过多的子功能,导致界面冗余,用户路径杂乱。通过与团队的反复沟通,我重新梳理了核心功能,将一些非必要的功能删除,简化了用户的操作路径。技术是服务用户不可或缺的工具,让产品更好地服务于用户才是最终目的。