# 一、软件名称:

**忆途 (TripMemory)**

# 二、软件需求

**2.1 用户群体**

**(1)学生群体：**涵盖中小学生、大学生及职业院校学生。他们在学习成长阶段，需记录学习成果、课外活动、社会实践等经历，用于规划学习目标、申请奖学金、升学或求职时整理资料。

**(2)职场人士：**包括职场新人与有经验的专业人员。他们可借助软件记录工作成就、项目经历、技能提升等，助力职业规划、明确发展路径，为求职或晋升提供证明。

**(3)个人成长爱好者：**注重自我提升、热衷记录生活点滴、兴趣爱好发展、心理状态变化等，通过软件分析建议更好了解自己，制定合理成长目标。

## 2.2 用户需要什么

**(1)全面记录：**

渴望有一个便捷的工具来记录自己在不同领域取得的成就，如学习上的奖项、工作中的业绩成果等。

能够详细记录生活中做过的各类事情，包括事件的关键信息，以便日后准确回顾。

随时记录当下的心情状态，便于分析情绪变化规律，更好地管理情绪。希望记录去过的地方，留存旅行记忆，分享旅行感悟。对个人基本资料进行集中管理，方便随时查看和更新。

**(2)深度分析：**

希望软件能对自己记录的大量信息进行深度挖掘和分析，从中发现自身的成长规律、优势与不足，为未来的发展提供有价值的参考。

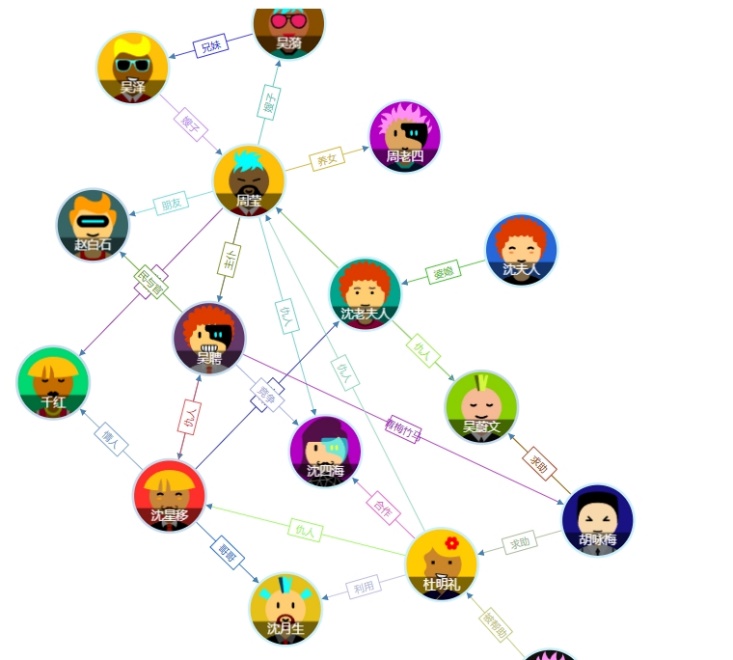
**(3)人际关系：**

**a.记录联系人信息**：姓名、关系类型、认识时间、联系方式d等。

**b.追踪互动事件**：记录互动内容、时间、地点、重要程度、附件，支持互动事件分类。

**c.社交网络分析：**生成人际关系图谱，展示联系人之间的关系密度，自动识别核心联系人与潜在关系网。

**d.互动频率统计：**按周/月/年维度统计高频互动对象，标记长期未联系的重要联系人并推送维护提醒。



**(4)合理规划：**

根据自身的成长情况和软件分析结果，制定出符合自身实际的、切实可行的个人目标，并能得到关于如何实现这些目标的具体建议。

**(5)数据安全：**

确保自己记录的所有信息得到妥善保护，不会被他人非法获取或篡改。

## 2.3 软件能帮用户做什么

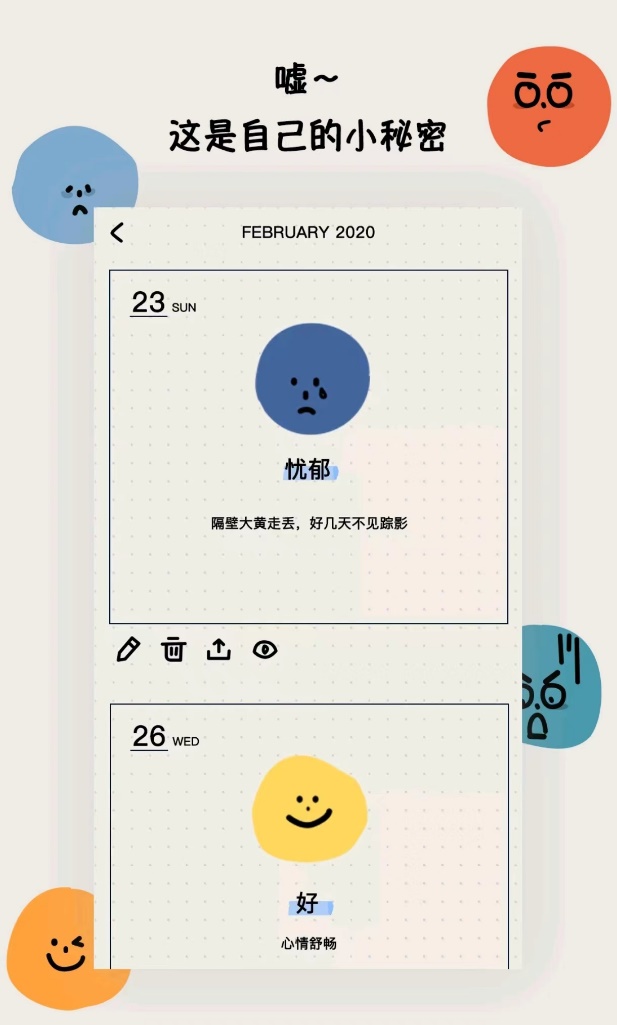
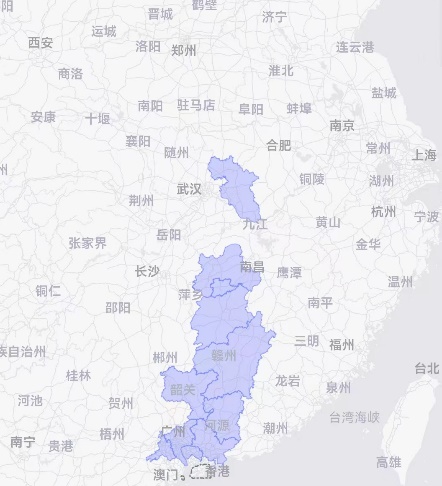
### 2.3.1.全面记录个人信息

**(1)成就记录：**支持用户详细录入各类成就信息，如获奖时间、颁奖机构、奖项名称、奖项级别等。例如，学生可记录学科竞赛、艺术比赛奖项；职场人士可记录工作业绩、创新成果奖励等。提供图片上传功能，可上传获奖证书照片作为证明。



**(2)事件记录：**允许用户记录做过的事情，包括事件发生的时间、地点、参与人员、事件详情等。支持添加事件标签，方便分类和检索。例如，用户可以记录参加的商务会议、学术讲座、社交聚会等。

**(3)心情记录：**用户能随时记录当下心情，提供多种心情状态选项可选多种**基础情绪**标签（如开心、焦虑、平静），并拖动强度滑块（0100%）量化混合情绪比例并可添加简短描述，分析情绪产生的原因。同时，支持语音输入心情记录，提高记录效率。

**(4)地点记录：**记录去过的地方，可添加地点名称、到达时间、离开时间、地点照片、评价和旅行感悟等。软件可根据记录生成个人旅行地图，直观展示去过的地方。

### 2.3.2数据分析与洞察

**(1)生成个人成长报告：**软件依据用户记录的信息，自动生成详细的个人成长报告。报告内容包括不同时间段的成就分布、活动频率统计、心情变化趋势等。例如，以图表形式展示学习成绩的提升曲线、工作业绩的增长趋势。

**(2)发现优势与不足：**通过对用户记录的大量信息进行深度分析，精准找出用户的优势领域和不足之处。例如，分析工作项目经历，发现用户在项目管理、技术创新等方面的优势，以及沟通协作、时间管理等方面的不足。

**(3)提供个性化建议：**根据个人成长报告和优势不足分析结果，为用户提供针对性的建议和目标方案。例如，对于学习成绩有待提高的学生，提供学习方法建议、推荐相关学习资源；对于职业发展遇到瓶颈的职场人士，建议参加培训课程、拓展人脉资源。



### 2.3.3目标设定与跟踪

**(1)设定个人目标：**用户可根据自身成长需求和软件建议，设定学习、职业、健康等多方面的目标。目标可分为短期、中期和长期，并分解为具体的任务和步骤。支持为每个目标设置优先级和截止日期。

**(2)跟踪目标进度：**实时跟踪目标进度，以直观的图表形式展示目标的完成情况和剩余时间。用户可随时更新任务完成状态，软件自动计算目标完成比例。当目标进度滞后时，提供提醒和建议。

**(3)调整目标方案：**随着用户成长和环境变化，软件可根据实际情况帮助用户调整目标方案。例如，当用户在某方面取得重大突破时，自动调整相关目标的难度和时间节点。

### 2.3.4.保护用户的隐私

**(1)法律保障体系优化**：用户协议明确禁止数据用于广告推送及未经授权披露，违规者追究刑事责任，公司对员工或合作方分别处罚巨额违约金。

**(2)技术隐私保护**：采用 AES256 加密存储 + TLS 1.3 传输，区块链存证操作记录。动态匿名化隐藏身份及敏感信息，零信任架构与设备指纹行为分析实时监测异常登录。

**(3)用户数据自主管理**：支持一键导出全部数据（加密压缩包）、选择性删除特定记录（如某时间段的心情记录）。

**(4)敏感信息模糊化**：自动隐藏身份证号、银行卡号等字段的部分数字（如 "5101234"），用户可自定义脱敏规则。

## 2.4软件的使用

### 2.4.1记录方式设计

**(1) 智能全自动记录**

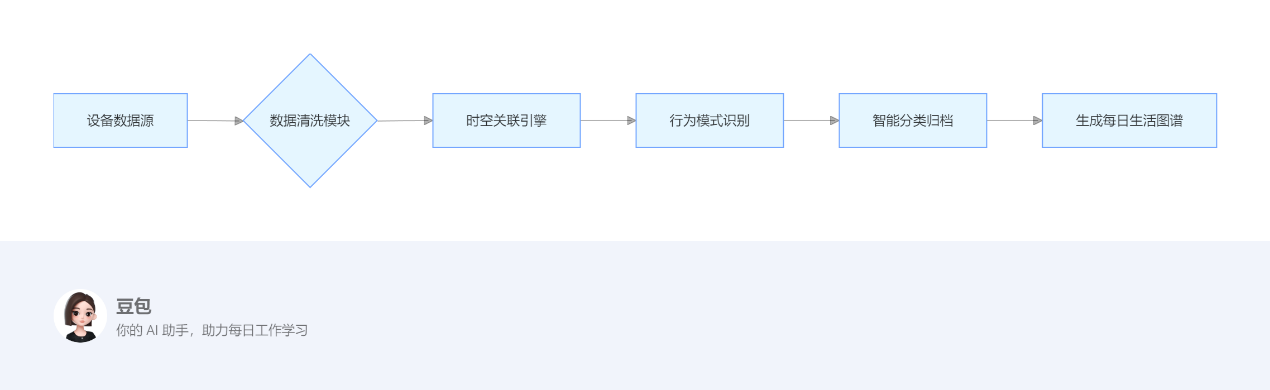
**多端数据采集**

**1.手机/平板**：通过系统API获取应用使用时长、社交动态、支付记录等行为数据

**2.电脑端**:支持浏览器插件记录网页浏览轨迹,以及本地软件使用时长统计

**3.智能穿戴：**同步手环/手表的运动数据、心率变化、睡眠质量等生物信息，通过定位用户的位置来推算用户的行程与做的事情

**4.智能家居：**接入电视/音箱的观看记录、语音指令等生活数据



**(2) 灵活手动记录**

**触发方式：**

1.语音指令："忆途，记录今天的项目会议内容"

2.快捷手势：三指下滑唤醒记录界面

3.桌面小组件：拖拽添加至主屏幕的快速入口

**多模态输入(支持用各种形式来记录)**

1.文本输入：支持Markdown格式排版以及各类文本形式

2.语音形式：支持用户用语音来记录当天特殊时段的事情，并且实时转写并生成文字摘要以语音形式保留

3.图片上传：自动识别图片中的信息，用图片更生动的展示获奖证书，旅游景点之类的

4.视频录制：支持视频直接上传进行展示

### 2.4.2记录内容管理

**1. 结构化数据存储**

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **必选项** | **可选项** | **扩展属性** |
| 成就 | 时间 / 名称 / 类型 | 证书编号 / 关联目标 | 贡献值 (110 分) |
| 事件 | 地点 / 参与人 | 任务进度 / 文件附件 | 影响力指数 (热力图) |
| 心情 | 情绪类型 / 强度 | 触发原因 / 缓解措施 | 生理指标 (心率 / 血压 |

**2. 多媒体资产管理**

**文件**格式支持

1.文档：PDF/Word/Excel

2.图片：JPG/PNG/HEIC（支持EXIF元数据）

3.音视频：MP3/M4A/MP4/AVI

4.特殊格式：CAD图纸/PSD源文件（预览缩略图）

### 2.4.3数据存储与管理

**(1) 存储策略:**

分层存储架构：

**a.近 30 天数据：**本地 SSD + 云端主集群双备份，采用 RAID 10 技术保障高可用性，每日增量备份至异地数据中心。

**b.历史数据：**自动归档至云端冷存储（支持 S3 Glacier），归档前通过 AI 算法压缩冗余数据（如重复照片、未修改文档），存储成本降低 70%。

**c.敏感数据**：采用加密分片存储（每片独立 AES256 密钥），分片分散存储于不同物理服务器，密钥由硬件安全模块（HSM）管理，支持零知识证明验证数据完整性。

**(2) 智能设备联动**

**跨设备协同场景：**

a.手环检测到运动结束 → 自动生成运动记录（配速,卡路里,轨迹地图），同步至健康分析模块并关联当日心情数据。

b.手机拍摄照片 → 自动匹配地理位置信息（基于 EXIF + 基站定位），识别照片内容（如 "会议现场"" 旅游景点 "）并建议关联事件类型。

c.电脑端完成文档 → 自动同步至成就系统，提取文档关键词（如 "项目方案"" 创新点 "）生成标签，触发相关目标进度更新。

d.智能家居传感器检测到异常 → 触发紧急事件记录（如 "20240520 家中无人时警报触发"），自动标记为高优先级。

# 三、功能描述：

（提供哪些函数？各函数的输入数据有哪些？每个数据的类型和格式是什么？各函数的输出数据有哪些？每个数据的类型和格式是什么？各函数分别能完成什么任务？）

## 3.1功能一：成就记录系统(基于AchievementTracker.java)

### 3.1.1 成就记录函数 AchievementTracker()

输入：获奖时间、颁奖机构、奖项名称、奖项级别、备注

输出：将输入内容数据保存起来

任务：创建成就记录界面，包含5列表格（获奖时间、颁奖机构、奖项名称、奖项级别、备注）和「添加/编辑/删除」按钮组。

绑定按钮事件：点击「添加」直接弹出空白对话框（传入null），点击「编辑/删除」需先选中表格行。

### 3.1.2 展示成就函数showAchievementDialog(Object[] existingData)

输入：同上

输出：将数据库里的成就内容展示出来

任务：弹出模态对话框。

### 3.1.3 编辑成就函数editSelectedAchievement()

输入：对成就内容进行编辑

输出： 将编辑修改后的成就展示出来

任务： 检查表格选中行：若未选中，提示「请先选择一行」；若选中，提取该行数据（Object[]）并调用showAchievementDialog()进入编辑模式， 对内容进行修改。

## 3.2功能二：心情日历系统（基于 MoodCalendar.java）

### 3.2.1 记录心情函数 MoodCalendar()

输入: emotion 在对应日期上输入心情

输出: 将日历上这天心情状态记录下

任务: 初始化窗口标题[心情日历]，currentDate设为当前日期，moodMap存储日期心情键值对。

### 3.2.2 初始化日历页面函数initUI()

输入：无 (初始化功能)

输出：初始化日历页面，构建界面布局

任务：顶部导航栏：按钮绑定月份切换（修改currentDate并触发重绘重新初始化），monthLabel显示当前年月（如May2024）。

日历容器：calendarPanel使用7列网格布局，每行代表一周。

### 3.2.3 生成日期对应星期函数renderCalendar()

输入：year年-month月-day日 (例如：2025-4-29)

输出：当天日历表格

任务：生成当前月份日历：

1. 标题行：[Sunday-Saturday]标签。

2. 日期偏移：计算当月第一天的周偏移（如5月1日是周三，前置3个空白单元格）。

3. 日期按钮：每个日期生成按钮，文本格式<日期>心情，点击弹出输入框编辑心情，更新moodMap后强制重绘。

**基姆拉尔森（Kim Larsen）计算公式：**

（注意:除法向下取整）

**符号说明表**

|  |  |
| --- | --- |
| 符号 | 说明 |
| W | 星期（1~7） |
| D | 对日期中的日 |
| M | 月份 |
| Y | 年份 |

例如：

以2025年4月29号为例，确定参数：d = 29，m = 2，y = 2025

代入公式： W = (29 + 8 + 3 + 2025 + 506 – 20 + 5 + 1)mod7= (2557) mod 7 = 2 ，即 W = 2 即周二。

**注意：**W = 0是星期天，对应日历中第1列

## 3.3功能三：事件记录系统(基于LogRecord.java)

### 3.3.1 记录事件函数：recordEvent()

输入：日期,事件类型（如“学习”、“娱乐”等）,事件描述

输出：将事件信息存储进事件列表或数据库

任务：构建事件记录界面，包含列表（列：日期、事件类型、事件描述）,提供“添加事件”按钮，点击后弹出输入框填写事件,保存数据到eventList或数据库表中

### 3.3.2 修改事件函数：editEvent()

输入： idx事件索引（来自表格选中需要修改的事件行）

输出： 更新该事件记录的结果

任务：若无选中行，弹出提示[请先选择一条事件记录],然后读取该条数据，调用showEventDialog(existingData)进行编辑，保存编辑结果更新表格和数据源

### 3.3.4 查询事件函数：searchEvents(String keyword)

输入： keyword关键字（事件类型或内容）

输出： 过滤并展示匹配的事件记录

任务：提供一个搜索框 + 查询按钮，更新列表展示符合条件的结果

### 3.3.4 统计事件类别函数：countEventTypes()

输入： 需要统计的事件类别

输出： 返回或展示每种事件类型的数量

任务： 顺序遍历存储的事件数据，统计各类事件出现次数（如“娱乐”10次，“考试”3次等） 可用柱状图、饼图等可视化（可选）

# 四、性能描述：

（完成每个功能允许的时间和空间是多少？）

## 4.1成就记录系统（AchievementTracker）

|  |  |  |  |
| --- | --- | --- | --- |
| **功能** | **时间限制（响应时间）** | **空间限制（内存 / 存储）** | **说明** |
| **新增成就** | ≤50ms | 单条记录：≈1KB（5 个字符串） | 直接操作内存中的DefaultTableModel，无 IO 延迟。10 万条记录≈100MB 内存。 |
| **编辑成就** | ≤30ms | 无额外空间（复用原有记录内存） | 仅修改表格模型的行数据，无需重建界面。 |
| **删除成就** | ≤20ms | 释放单条记录内存（≈1KB） | 调用removeRow()直接从数组移除，时间复杂度 O (1)。 |
| **启动时间** | ≤500ms | 初始内存：≈20MB（JVM 基础 + Swing 组件） | 界面组件初始化耗时主要在 Swing 渲染，与记录数量无关。 |

## 4.2. 心情日历（MoodCalendar）

|  |  |  |  |
| --- | --- | --- | --- |
| **功能** | **时间限制（响应时间）** | **空间限制（内存 / 存储）** | **说明** |
| **日期选择** | ≤20ms | 无额外空间（仅记录选中日期） | 直接获取LocalDate，无复杂计算。 |
| **心情标记（渲染）** | ≤80ms（单月标记≤31 天） | 每个标记：≈80B（LocalDate+int） | 遍历moodMap绘制圆点，时间复杂度 O (n)，n 为单月标记天数（≤31）。 |
| **月份切换** | ≤50ms | 无额外空间（复用日历组件内存） | SwingJCalendar预加载相邻月份，切换时仅重绘当前月。 |
| **数据持久化（扩展）** | ≤200ms（100 条记录） | 单条记录：≈50B（加密存储） | 假设扩展为文件存储，AES 加密写入时间与记录数线性相关（≤100 条 / 200ms）。 |
| **启动时间** | ≤600ms | 初始内存：≈25MB（含日历组件） | 加载自定义渲染器和按钮组件，比成就系统多 5MB 左 |

## 4.3 事件记录系统(LogRecord)

|  |  |  |
| --- | --- | --- |
| **功能** | **时间限制（响应时间）** | **空间限制（内存 / 存储）** |
| 新增事件记录 | ≤50ms | 单条记录：≈1KB（日期+类型+描述） |
| 编辑事件记录 | ≤30ms | 无额外空间（复用原有记录内存） |
| 删除事件记录 | ≤20ms | 释放单条记录内存（≈1KB） |
| 查询事件（关键词匹配） | ≤100ms（≤1000 条记录） | 临时结果集合占用 ≈1~2MB |
| 分类统计（事件类别） | ≤150ms（≤5000 条记录） | Map结构占用 ≈100KB（<20 类别） |
| 启动时间 | ≤550ms | 初始内存：≈22MB（基础组件+事件表） |

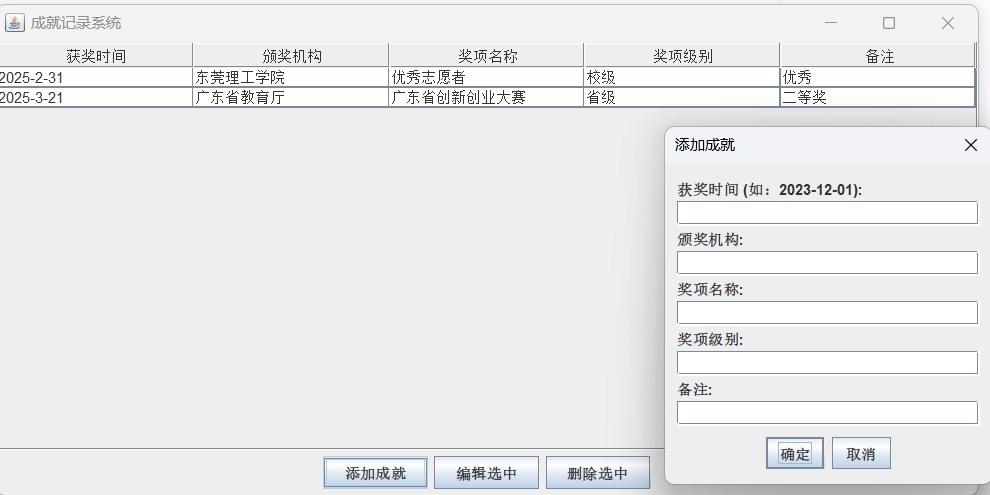
# 五、测试数据：

（为测试每个功能的正确性分别提供若干组输入数据和对应的输出数据）

### 功能一：成就记录系统（基于 AchievementTracker.java）

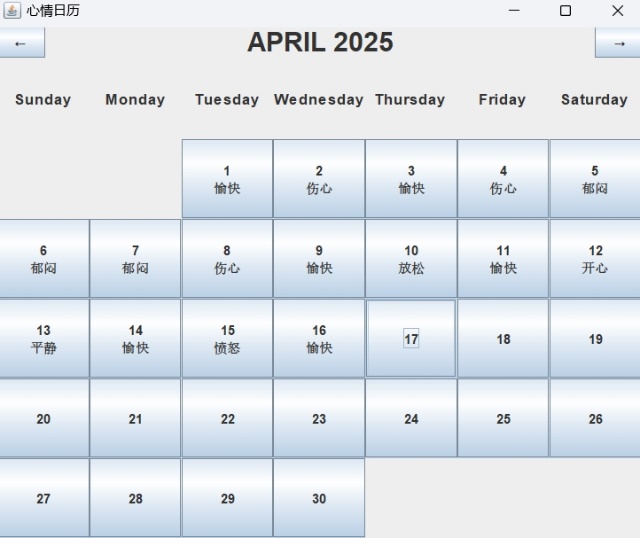
**测试数据：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **获奖时间** | **颁奖机构** | **奖项名称** | **奖项级别** | **备注** |
| 2025-6-5 | 北京科技协会 | 科技小能手奖 | 市级 | 积极参与科技活动 |
| 2025-6-5 | 上海某高校 | 校园文学奖 | 校级 | 文章优秀 |
| 2025-8-20 | 杭州市教育局 | 青少年艺术奖 | 市级 | 绘画类一等奖 |
| 2025-7-25 | 清华大学 | 优秀志愿者 | 校级 | 服务时长超 100 小时 |
| 2025-10-4 | 广东省文化厅 | 文化传承奖 | 省级 | 非遗传承项目 |
| 2025-11-15 | 成都市教育委员会 | 创新思维大赛奖 | 市级 | 团队协作优秀 |



### 功能二:描述：心情日历系统（基于 MoodCalendar.java）

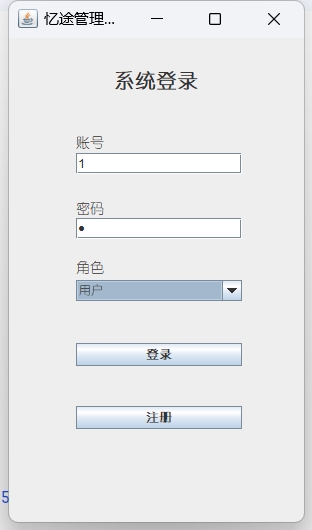
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sunday（周日）** | **Monday（周一）** | **Tuesday（周二）** | **Wednesday（周三）** | **Thursday（周四）** | **Friday（周五）** | **Saturday（周六）** |
| **（空）** | **（空）** | 1（愉快） | 2（伤心） | 3（愉快） | 4（伤心） | 5（郁闷） |
| 6（惬意） | 7（舒畅） | 8（伤心） | 9（愉快） | 10（放松） | 11（愉快） | 12（开心） |
| 13（平静） | 14（愉悦） | 15（愤怒） | 16（愉快） | 17（难过） | 18（烦闷） | 19（愁闷） |
| 20（轻松） | 21（愉悦） | 22（欢喜） | 23（恼怒） | 24（平和） | 25（愉悦） | 26（惬意） |
| 27（失落） | 28（愉快） | 29（开心） | 30（烦躁） | **（空）** | **（空）** | **（空）** |



功能三：记录事件系统（基于 MoodCalendar.java）

|  |  |  |  |
| --- | --- | --- | --- |
| **分类** | **主题** | **内容** | **发布时间** |
| 旅游 | 古城漫步 | 游览丽江古城，体验民俗风情 | 20250416 |
| 运动 | 瑜伽练习 | 完成一小时瑜伽体式训练 | 20250416 |
| 学习 | 编程学习 | 学习 Python 函数编写基础 | 20250416 |
| 旅游 | 海滨之旅 | 青岛海边踏浪，参观栈桥 | 20250417 |
| 学习 | 历史研读 | 梳理唐朝兴衰历史脉络 | 2025041 |





**（初版页面设计）**

# 六、设计思想：

（说明软件系统的总体结构和各部分的调用关系，分别说明每个函数中采用的数据结构和关键算法，并阐述选用的理由）



## 6.1 成就记录系统（基于 AchievementTracker.java）

### 6.1.1 函数调用关系（序号标注）：

**(1)初始化流程：**

构造函数AchievementTracker()启动时，首先调用initUI()方法初始化界面，创建包含 5 列表格（获奖时间、颁奖机构、奖项名称、奖项级别、备注）和「添加 / 编辑 / 删除」按钮组的界面布局，表格数据通过DefaultTableModel进行管理。

**(2)添加成就：**

点击「添加」按钮时，触发无参的showAchievementDialog()方法，弹出空白对话框供用户输入成就信息。用户提交后，系统调用addAchievement()方法将输入数据封装为Object[]数组，并插入到DefaultTableModel中，实现表格数据的更新。

**(3)编辑成就：**

点击「编辑」按钮时，首先通过getSelectedRow()获取用户选中的表格行索引。若未选中行，提示用户；若选中行，则提取该行数据（Object[]）并调用带参的showAchievementDialog(existingData)进入编辑模式。用户修改完成并提交后，系统调用updateAchievement()方法更新DefaultTableModel中的对应行数据。

**(4)删除成就：**

点击「删除」按钮时，通过getSelectedRow()获取选中行索引，直接调用DefaultTableModel的removeRow(int)方法删除对应行，触发表格自动刷新。

### 6.1.2 数据结构与算法：

**(1)数据结构:**

**DefaultTableModel：**用于存储表格数据，包含列名（获奖时间、颁奖机构、奖项名称、奖项级别、备注）和具体记录。每条记录以Object[]数组形式存储，数组元素依次对应各列数据。

**Object数组：**单条成就记录的存储形式，如new Object[]{date, org, name, level, note}，与表格行一一对应。

**(2)关键算法：**

主要依赖 Swing 表格模型的内置方法实现数据操作，如addRow(Object[])用于添加行，removeRow(int)用于删除行，

时间复杂度均为 O (1)（基于数组的直接操作）。

### 6.1.3 选用理由

**(1)DefaultTableModel的选用：**Swing 框架的原生表格模型，与JTable组件天然绑定，支持数据与界面的双向同步。无需手动处理界面渲染与数据存储的解耦逻辑，降低开发成本，且提供高效的行级操作（添加、删除、修改），适合结构化数据的表格展示场景。

**(2)Object 数组的选用：**

简单轻量，能够直接适配表格行的格式要求。每条记录的各字段通过数组索引一一对应，便于在按钮事件处理中快速提取和传递数据（如编辑时获取选中行数据），避免复杂的数据封装类实现。

**(3)Swing 组件的选用：**

针对桌面端应用场景，Swing 提供成熟的 GUI 组件库，无需额外引入第三方框架，兼容性强。按钮事件处理逻辑清晰，通过监听机制实现功能解耦，适合快速原型开发和中小型应用。

## 6.2 心情日历系统（基于 MoodCalendar.java）

### 6.2.1 函数调用关系（序号标注）：

心情日历系统以日历为核心界面，支持按日期记录心情，包含日历初始化、月份切换、心情标记等功能，结构如下：

**(1)初始化流程：**

构造函数MoodCalendar()启动时，设置窗口标题、尺寸和位置，初始化当前日期currentDate和用于存储日期 - 心情映射的moodMap（HashMap<LocalDate, Mood>），随后调用initUI()方法构建界面。

**(2)界面布局：**

initUI()方法创建顶部导航栏和日历容器calendarPanel（采用 7 列网格布局）。切换按钮绑定事件监听，点击时调用updateCurrentMonth()方法变更年月，并触发renderCalendar()方法重绘日历。

**(3)日历渲染：**renderCalendar()方法根据当前年月生成日历表格：

计算当月第一天的周偏移由TemporalAdjusters.firstDayOfMonth()和getDayOfWeek()，确定日历中前置空白日期的数量。

为每个日期生成按钮，按钮文本显示日期和心情（若有记录）。点击按钮时弹出输入框，用户输入心情后更新moodMap，并强制重绘日历以反映心情变化（如按钮背景色根据情绪类型变化）。

### 6.2.2 数据结构与算法：

**(1)数据结构**

**HashMap<LocalDate, Mood>（moodMap）**：键为 Java 8 的LocalDate对象（表示日期），值为自定义的Mood对象（包含情绪类型和强度），用于存储每个日期的心情记录。

**GridLayout(0, 7)**：日历容器calendarPanel的布局管理器，固定 7 列（对应一周七天），行数自动根据日期数量调整，确保日期按钮按周排列。

**(2)关键算法**

**日期偏移计算：**

通过LocalDate.now().with(TemporalAdjusters.firstDayOfMonth())获取当月首日，再通过DayOfWeek.getValue()获取首日是一周中的第几天（如周日为 1，周一为 2，...，周六为 7），计算前置空白日期的数量（如首日为周三，则前置 2 天空白）。

**心情标记渲染：**遍历moodMap中的日期，为对应的日历按钮设置背景色或图标，直观展示心情分布（如红色表示悲伤，绿色表示愉快）

### 6.2.3 选用理由

**(1)HashMap 的选用：**

利用其 O (1) 的平均读写复杂度，实现快速的日期 - 心情查询和存储。对于日历场景，每天的心情记录是独立的键值对，哈希表能够高效处理频繁的插入、查询操作，尤其适合单月最多 31 条记录的小规模数据场景。

**(2)GridLayout 的选用：**

网格布局管理器简化了日历界面的排版逻辑，无需手动计算每个按钮的位置，只需按顺序添加日期按钮，即可自动形成周行结构。与日历的周期性（7 天一周）天然匹配，确保界面整齐统一。

**(3)Java 标准库日期算法的选用：**

使用LocalDate和TemporalAdjusters等 Java 8 新增的日期 API，避免手动处理日期计算的复杂性（如不同月份的天数、闰年判断），提高代码的准确性和可读性。标准库算法经过充分测试，稳定性强，减少开发错误。

**(4)事件驱动架构：**

通过按钮点击事件触发心情输入和日历重绘，解耦界面交互与数据逻辑，符合 MVC（模型-视图-控制器）设计模式，便于后续功能扩展（如导出日历、心情统计）。

## 6.3 事件记录系统（基于 LogRecord.java）

### 6.3.1 函数调用关系（序号标注）：

**(1)初始化流程：**

系统启动时调用loadFromFile()方法（事件数据存储在文件或数据库中），将历史事件加载到eventList（ArrayList<Event>）中，并绑定到表格的DefaultTableModel，实现数据初始化。

**(2)添加事件：**

点击[添加事件]按钮时，弹出输入框让用户填写日期、事件类型、事件描述等信息。用户提交后，创建Event对象并添加到eventList，同时更新表格显示。

**(3)编辑事件：**

用户选中表格中的某条事件记录后，点击「编辑」按钮，系统通过getSelectedIndex()获取选中行索引，提取对应的Event对象并进入编辑模式。修改完成后，调用eventList.set(index, newEvent)更新指定索引的事件数据，刷新表格。

**(4)查询事件：**

用户在搜索框输入关键词（如事件类型或内容），点击查询按钮后，系统调用filterEvents(keyword)方法遍历eventList，筛选出包含关键词的事件，更新表格显示查询结果。

**(5)统计事件类别：**

点击「统计」按钮时，调用countEventTypes()方法遍历eventList，统计各事件类型的出现次数，生成HashMap<EventType, Integer>，并可通过柱状图或饼图可视化统计结果。

### 6.3.2 数据结构与算法：

**(1)数据结构**

**ArrayList<Event>（eventList）：**顺序存储事件对象，每个Event包含日期、事件类型、事件描述等字段。

**HashMap<EventType, Integer>：**用于统计事件类型数量，键为事件类型（如 “会议”“学习”“旅游”），值为该类型事件的计数。

**(2)关键算法**

**关键词模糊匹配：**使用String.contains(keyword)方法对事件描述或类型进行线性遍历匹配，筛选出包含关键词的事件。

**(3)事件分类统计：**

遍历eventList，对每个事件的类型，使用HashMap.merge(type, 1, Integer::sum)方法进行计数，避免重复判断类型是否已存在，简化统计逻辑。

### 6.3.3 选用理由

**1.ArrayList 的选用：**

适合按时间顺序添加和访问事件的场景，支持快速的追加操作（O(1)时间复杂度）和按索引访问（O(1)），满足事件记录的时序性需求。对于中小型应用（如万级以下事件记录），线性遍历的性能足够，无需引入更复杂的数据结构（如链表或树结构）。

**2.HashMap 统计的选用：**

利用哈希表的高效键值对操作，实现 O (n) 时间复杂度的事件类型统计。merge方法简化了计数逻辑，一行代码即可完成 “若类型存在则计数加 1，否则初始化为 1” 的操作，代码简洁且高效。

**3.简单模糊匹配的选用：**

针对用户可能的基础搜索需求（如快速查找包含某关键词的事件），使用contains方法足够满足需求，避免引入正则表达式或全文检索引擎的复杂性。在数据规模较小时，线性遍历的性能损失可忽略，且实现成本低。

**4.与表格组件的兼容性：**

ArrayList与 Swing 的DefaultTableModel兼容，可直接将事件对象转换为表格行数据（如Object[]数组），便于界面展示和交互，保持数据层与界面层的简单映射关系。

## 6.4 总体设计思想总结

### 6.4.1 架构设计原则

**(1)模块化设计**：

将系统拆分为成就记录、心情日历、事件记录等独立模块，每个模块对应一个 Java 类（AchievementTracker、MoodCalendar、LogRecord），模块内部封装数据结构和业务逻辑，模块间通过界面交互解耦，便于单独开发、测试和维护。

**(2)数据与界面分离：**

采用 Swing 的模型-视图架构（如DefaultTableModel与JTable绑定），数据存储在模型中，界面仅负责渲染和用户交互，确保数据更新时界面自动同步，减少手动更新界面的代码冗余。

**(3)适配小规模数据场景：**

针对学生、职场人士等用户群体的日常记录需求，数据规模通常较小（单模块记录数在万级以下），因此选用简单高效的数据结构（如 ArrayList、HashMap）和基础算法（线性遍历、简单匹配），在保证性能的同时降低实现复杂度。

### 6.4.2 技术选型依据

**(1)Swing 框架的选用：**

目标用户为桌面端用户，Swing 作为 Java 原生 GUI 框架，无需额外依赖，适合快速开发轻量级应用。对于非复杂界面（表格、日历、表单），Swing 组件足够满足需求，避免引入 JavaFX 等更复杂框架的学习成本。

**(2)Java 标准库的充分利用：**

日期处理（LocalDate）、集合框架（ArrayList、HashMap）、布局管理器（GridLayout）等均使用 Java 标准库，确保代码的可维护性和兼容性。

**(3)用户体验优先：**

数据展示直观（日历标记、表格列表），符合用户对 “记录工具” 的核心需求 —— 便捷记录与快速查询。数据结构和算法的选择均以 “易用性” 为前提，避免为追求技术复杂度而牺牲用户体验。

# 七、实现细节：

（采用的编程环境及语言，并阐述选用的理由；说明运行平台的考虑）

**7.1.1编程语言：Java**

**(1)选用技术：**Java 是一种广泛使用的高级编程语言。

**(2)原因：**

**a.跨平台特性：**Java 具有“一次编写，到处运行”的特性，这意味着基于 Java 编写的软件可以在不同的操作系统（如 Windows、MacOS、Linux）上运行，无需针对每个系统进行单独的修改。这使得软件能够轻松适配多样化的用户群体，无论是学生、职场人士，还是使用不同操作系统的用户，都能顺利使用软件。

**b.丰富的类库：**Java 拥有庞大且丰富的类库，涵盖了基础数据处理、网络通信、图形界面开发等各个领域。这些类库提供了许多现成的功能和工具，极大地提高了开发效率。例如，Java 8 引入的 LocalDate 等日期时间 API，提供了简洁且线程安全的日期操作方式，避免了开发者手动进行复杂的日期计算和格式化处理。

**c.面向对象特性：**Java 是一门面向对象的编程语言，具有良好的封装性、继承性和多态性。这些特性使得代码结构清晰，易于模块化开发和维护。开发者可以将不同的功能封装成独立的类和对象，通过继承和多态实现代码的复用和扩展，降低了软件的耦合度，提高了代码的可维护性和可扩展性。

**7.1.2集成开发环境（IDE）：IntelliJ IDEA**

**(1)选用技术**：

IntelliJ IDEA 是一款功能强大的 Java 集成开发环境。

**(2)原因：**

**a.智能代码补全：**IntelliJ IDEA 提供了智能的代码补全功能，能够根据上下文自动提示可能的代码选项，大大减少了开发者的输入工作量，提高了编码速度。

**b.高效的代码导航与搜索：**它具备强大的代码导航和搜索功能，开发者可以快速定位到所需的类、方法和变量，方便代码的查看和修改。例如，通过快捷键可以快速跳转到类的定义、方法的实现等。

**c.集成调试工具：**IDE 集成了功能强大的调试工具，支持单步调试、断点设置等操作。开发者可以在调试过程中逐步执行代码，观察变量的值和程序的执行流程，方便排查代码中的问题。

**d.丰富的插件生态系统：**IntelliJ IDEA 拥有丰富的插件生态系统，如 Maven、Git 插件等。Maven 插件可以方便地进行项目的依赖管理，自动下载和管理项目所需的依赖库；Git 插件则支持版本控制，方便开发者进行代码的提交、分支管理等操作，显著提升了项目管理效率。

**7.2 页面布局：Swing 框架**

**(1)选用技术：**

Java Swing 是 Java 基础类库（JFC）的一部分，用于构建桌面应用程序的用户界面。

**(2)原因：**

**a.丰富的轻量级组件：**Swing 提供了丰富的轻量级组件，如 JFrame（窗口）、JPanel（面板）、JTable（表格）、JCalendar（日历）等。这些组件可以方便地组合和使用，能够快速搭建出功能齐全、交互性良好的桌面应用界面。

**b.满足功能模块界面展示需求：**对于忆途软件的成就记录、心情日历、事件记录等功能模块，Swing 组件能够很好地满足其界面展示需求。例如，使用 JTable 组件可以方便地展示成就记录和事件记录列表，实现数据的可视化呈现与交互操作；JCalendar 组件则为心情日历系统提供了直观的日期选择和展示方式。

**c.与 Java 紧密集成：**Swing 与 Java 语言紧密集成，开发者无需额外学习复杂的语法和框架，降低了开发门槛。开发者可以直接使用 Java 语言的特性和类库来操作 Swing 组件，实现界面的功能逻辑。

**7.3 存储数据：MySQL 数据库**

**(1)选用技术：**

MySQL 是一款开源的关系型数据库管理系统。

**(2)理由：**

**a.稳定性和可靠性：**MySQL 具有高度的稳定性和可靠性，经过多年的发展和优化，能够在各种环境下稳定运行，确保数据的安全和完整性。

**b.强大的数据存储和管理能力：**它对数据的存储和管理能力强大，能够高效地处理忆途软件中的结构化数据，如成就记录、事件记录、用户信息等。可以根据不同的数据类型和业务需求，设计合理的数据库表结构，实现数据的高效存储和查询。

**c.支持标准 SQL 语言：**MySQL 支持标准的 SQL 语言，方便开发者进行数据的查询、插入、更新和删除操作。开发者可以使用熟悉的 SQL 语句来操作数据库，提高了开发效率。

**d.与 Java 的 JDBC 无缝集成：**MySQL 与 Java 的 JDBC（Java Database Connectivity）技术无缝集成。通过 JDBC，Java 程序可以方便地连接到 MySQL 数据库，实现数据的持久化存储与读取。这确保了软件在用户关闭后数据不会丢失，为用户提供持续的记录和分析服务。

**7.4 运行平台的考虑**

**(1)支持平台：**

基于 Java 的跨平台特性开发软件，支持 Windows、MacOS 和 Linux 操作系统。

**(2)优势：**

**a.广泛覆盖用户群体：**Java 的跨平台特性使得软件能够在不同的操作系统上稳定运行，无需针对每个系统进行单独的开发和适配。这使得软件能够覆盖更广泛的用户群体，无论是使用 Windows 系统的大多数普通用户，注重用户体验和简洁性的 MacOS 用户，还是对开源软件接受度较高的 Linux 用户，都能顺利使用软件。

**b.满足不同平台用户需求：**

不同的操作系统有其自身的特点和用户偏好。对于 Windows 系统，其广泛的用户基础和兼容性确保了大多数用户能够顺利安装和使用软件；MacOS 用户注重用户体验和软件的简洁性，Java Swing 的界面设计能够满足这一需求，提供美观、易用的界面；Linux 系统的用户对开源软件的接受度较高，忆途软件基于 Java 和 MySQL 的开源技术栈，符合 Linux 用户的偏好，且能在不同的 Linux 发行版上正常运行。

通过充分考虑不同运行平台的特点，软件能够为各类用户提供便捷的个人记录与分析服务。