

# 25 春程设 75 组大作业项目报告

## 一、程序功能介绍

我们小组基于 PyQt 进行设计，项目由日程管理和日记系统两大主要功能组成，各自集成了如下的具体实现：

### 1、日记系统

(1) 写日记——点击界面的“Diary”按钮进入日记系统，界面将显示最近一周的日期，并为每个日期提供日记编辑保存功能：

日历视图显示默认为显示当前日期所在的星期，方便查看并选择具体日期修改、保存日记；用户可点击“SelectDate”选择指定日期进行日记编辑，选择后点击“go and see”按钮，日历视图会自动调整为该日期所在的星期，确保清晰看到相关信息；与之相应配备有直接跳转回当天的接口，在编辑完指定日期的日记后，用户可以通过点击“back to today”按钮一键返回到当前日期，随时切换。

(2) 情绪监测——集成了情绪监测功能，基于外部 API (DEEPSEEK) 进行情感分析并打分：

使用者可以在编辑日记后，点击右侧的“see report”按钮，程序会调用 DEEPSEEK API 分析当天日记的情感色彩，并给出评分（0-100 分）。分析结果和情绪评分会显示在界面上，并自动保存到对应日期的日记记录中。

用户点击“view statistics from past 7 days”按钮，可查看选定日期（对应功能

(1) 中的“SelectDate”选择的指定日期，若未选择默认显示时间的当日日期）前 7 天的情绪评分汇总。经过可视化处理，评分结果以柱状图形式展示，每个评分区间使用不同的颜色进行区分，从而直观地展现过去一周的情绪波动。

### 2、日程管理

(1) 待办：点击“To-Do”进入待办界面，可选择日期添加待办。每个待办事项由 title 名称、info 具体信息、category 分类（可选择）组成，同时可以选择该事项开始和截止时间，点击“save”添加，随后该待办的开始时间、title、category 分类将显示在左侧区域；也提供更便捷的添加方式——按下“Input events freely”自由输入，如“后天上午十点去邱德拔打羽毛球，一个小时”，程序会自动解析保存、添加到左侧区域显示。双击

事项，可改变代办前的符号“x”为“√”，用以标记完成。

(2) 添加讲座：我们借助 python 爬虫的功能，在北大数学科学学院官网以及小北同学网站上获取了不同学院规划的讲座、活动信息。点击 “addLectures” 按键后，这些讲座将以“具体开始时间-讲座/活动名称-来源（数院/小北同学）”出现在对应日期的待办显示栏目中。

(3) 日历视图：更可视化的方式查看每日待办。点击“Calendar”进入待办的日历视图界面，视图默认展示最近一周的日期，并将每一天按小时划分。所有添加的待办与讲座活动将在对应日期和时间段内显示；当用户将鼠标悬停在某个时间段的任务或活动上时，系统会自动在光标旁显示该任务的详细信息，如标题、描述、地点等，提供便捷的查看方式。

### 3、其他

借助 json 完成账号用户管理，确保不同用户登录后只能编辑自己的日记待办事项；以及左上角的 ui 设计加入了用户名与头像的实现。

## 二、项目各模块与类设计细节

MainWindow 类:

初始化函数中绑定了按钮和对应的槽函数，定义了 CalDayView 类

各个成员函数实现对应的功能

setuser:

在 main.py 中接收登陆的 username，加载情绪日记信息，绑定 CalendarController 类对象（内部初始化函数加载对应用户的待办事项列表）

change\_date:

输入 date\_str 更换日历内容为对应日期

show\_page 一系列函数:

在 show\_page{i} 函数中, 执行函数(如 show\_page3 中执行 self.loadCalendar() 更新每个 Frame 的内容) 更新第 i 页的内容，并切换到 StackedWidget 的对应页

refresh\_list:

被绑定到 dateChanged 按钮，当切换日期的时候，更新待办事项的显示内容（通过 CalendarController 的 get\_events 接口）

addlectures:

接受来自 comboBox 的选项，调取对应爬虫，接受返回的 event 列表，并通过调用

calender 的 add\_event 接口，把对应院系的学术报告信息加入到待办中。

loadEventsOfWeek:

    调取一周所有事项存入字典，传给 loadCalendar

loadCalendar:

    加载所在周每一天的事项列表，并调用 calendarFrames (CalDayView 对象) 的 setDate、addEvent、finalyzeLayout 接口更新 Frame 的显示

toggle\_event\_done:

    绑定到 QListWidget 的双击事件，当双击时候，调用 CalendarController 的 toggle\_done 接口，将事件标记为已完成。

goto\_now 和 goto\_past:

    切换到当前日期的时候，更新整个日记视图的显示

generate\_report:

    从 json 获取当前用户输入的日记信息，调用 evaluator(emo\_eval) 的接口获取评分和寄语，更新显示

score\_renew\_info:

    显示上一次记录和评分时间

load\_emotion\_scores:

    读取文件（存有上一次评分的事件和分数），return 给 score\_new\_info

save\_emotion\_scores:

    更新 json 文件，存储评分、

从 diary 中 import 了部分函数（负责储存情绪日记，读取情绪日记）

plot\_emotion\_scores:

    显示一周情绪变化图

open\_ai\_input\_dialog:

    创建 AIInputDialog 对象，用 QMessageBox 弹出弹窗，将用户输入传递给 DeepSeek 的 API (AIInputDialog 内部处理)，分析后返回格式化的数据，更新到待办事项中。

convertTime:

    将 QDateTime 转为 python 的 datetime 方便处理。

模块一：日程表

    一. CalendarController 类:

1. `add_event` 函数: 添加事件
2. `get_event` 函数: 获取事件
3. `toggle_done` 函数: 在待办列表视图下, 双击标记事件已完成
4. `save_to_file` 函数: 存储该用户的所有待办 (更新 json 文件)
5. `load_from_file` 函数: 读取事件

## 二. `CalenderEvent` 类:

包含 `title, time, end_time, category, note, done` 成员变量

## 三. `Day` 类:

记录每一天的日期和事件

## 四. `event_eval` 类:

封装对接 OpenAI (DeepSeek) API, 解析用户的事件描述文本, 返回 structured JSON 数据。

关键属性: `API_KEY, MODEL, prompt, temperature`

核心方法: `eval(text)`: 将文本发送给 AI, 返回解析后的字典结构。

## 模块二: 情绪日历

### 一. `emo_eval` 类:

功能: 调用 DeepSeek Chat API 对输入文本进行情绪分析并返回评分和反馈语句

属性: `prompt`: 包含评分规范与生成指导语的 `prompt`

`api_key`: API 密钥 (默认为硬编码)

`model_name`: 所使用的模型名

`client`: OpenAI 客户端对象

`temperature`: 控制生成文本的多样性 (默认为 0.5)

方法: `eval(text)`: 与模型进行对话, 输入用户文本, 接收并解析返回值 (使用 @ 分割为分数和文本, 返回 `(float, str)` 格式的情绪得分和回应文本)

## 模块三: 学院讲座爬虫

### 一. `DepartmentsCrawler` 类 (`sms, eecs, xiaobei` 派生类):

功能: 从不同学院的官网和“小北同学”活动信息平台

#### 1. `departmentsCrawler` (基类)

属性:

`dep`: 部门名称 (默认是“元培学院”)

events: 存储讲座事件的列表

headers: HTTP 请求头 (模拟浏览器 User-Agent)

方法:

create\_events(): 占位方法, 由子类重写

2. sms (数学科学学院爬虫类, 继承自 departmentsCrawler)

重写属性:

dep = "数学科学学院"

重写方法:

create\_events():

调用 fetch\_info\_all 爬取讲座列表

生成结构化事件 (标题、时间、讲师、详情页、类型分类)

fetch\_info\_all(page=1):

支持多页爬取 (构造 URL)

fetch\_info(url, today):

爬取并解析页面讲座信息, 正则匹配讲座时间、标题等

3. eecs (信科院爬虫类, 继承自 departmentsCrawler)

重写属性:

dep = "信息科学技术学院"

重写方法:

create\_events(): 爬取页面生成 CalendarEvent 对象列表

fetch\_info(): 爬取主页, 提取讲座标题、时间、详情链接

4. xiaobei ("小北同学"活动信息平台爬虫类, 继承自 departmentsCrawler)

本类可以直接通过小北同学的 api 获取数据。

模块四: 待办事项的日历视图

CalDayView 类:

功能: 托管每个日期的日历显示, 初始化函数中四个成员变量分别为 show\_date, no\_time, has\_time. eventWidgets. 前三个为 ui 中对应的 Frame

成员函数:

clearList: 清除当前 Frame 中的所有事件, 清除日历日期显示。保证多次添加不会重叠

setDate: 获取 date\_str 所在的周的每个日期字符串, 对应到具体星期 N, 并更新日历

视图上方显示的日期

`addEvent`: 输入 `event`, 将 `event` 的各个详细信息添加到 `_pending_event` 列表中, 以便 `finalizeLayout` 进行处理。

`finalyzeLayout`: 输入当前 `Frame` 的宽度 (动态调整), 实现当 `event` 时间段重叠时, 分列显示。动态把时间绑定到新建的 `QPushButton`, 把 `QPushButton` 加入到 `Frame` 中, 并设置 `tooltip` 以在鼠标悬停于上方时显示事件详情。

静态成员函数:

`calculateY`: 传入 `time_str`, 计算当天时间对应的日程应当在 `Frame` 中占有多大的 `y`

### 三、小组成员分工情况

不同功能实现所需的代码量与函数逻辑各不相同, 但小组成员分工合作, 最终确保大家各自负责部分投入的时间相当: 吕鸿坤主要负责登录与主页面基础框架搭建、头像功能、网站爬虫与 `todo` 添加事件逻辑、全部日历视图的待办事项构建, 李欣悦主要负责日记部分全部界面与功能的搭建编写, 王韵博负责后端 `CalendarController`、`Day`、`UserManager` 类和前端 `StyleSheet` 设计, 以及 AI 自动识别事件功能的实现, 最终小组成员共同美化 `ui` 界面设计, 互相提出修改意见进行改进, 并一同撰写作业报告。

### 四、项目总结与反思

在项目开发的过程中, 我们充分运用了程序设计与实习课堂上教授的多种知识: 面向对象编程的思想贯穿整个项目, 例如我们定义了不同的类来处理日程、日记和讲座信息, 每个类负责特定的功能, 实现了代码的模块化; 此外 `Python` 爬虫技术的应用让我们能够从学院网络上抓取讲座信息, 丰富系统的内容, 并通过 `BeautifulSoup` 库解析网页内容, 我们能够准确提取所需的讲座标题、时间和详情链接等信息, 并将其整合到日程安排中。此外, 数据处理和 `matplotlib` 画图功能也为整体系统增色不少。我们利用数据处理技术对用户的日记内容进行情绪分析, 并将分析结果以直观的图表形式展示出来, 让用户能够清晰地了解自己的情绪变化。

然而, 项目开发该过程中, 我们遇到了一些技术难题和团队协作问题。比如技术方面, 虽然经过课堂上的讲解, 但团队成员普遍针对 `Qt` 开发的经验不足, 槽函数和信号使用生疏, 只好大家现学现用。在团队协作方面, 由于成员之间分工完善不同功能的前端后端, 又在同

时作业，常常出现同时出现改动处各不一致的多个版本，需要反复整合，推进效率受到不少影响。

幸运的是，经过团队小组反复磨合、修改，我们最终版本的项目成功完成了预设的功能。同时，我们也意识到项目仍存在一些改进的空间。在程序设计方面，虽然已经进行了一定的优化，但“打包处理”仍可以进一步完善，以提高代码的复用性和可扩展性。在日历待办功能中，不同类别事件的颜色显示还不够完善，用户在查看日程时可能会感到不够直观。此外，UI 界面的美学设计还有很大的提升空间，目前的界面尽管经过调整，较为整洁简约，但缺乏一定的视觉吸引力。

通过这次项目，我们积累了丰富的开发经验和教训。在未来的项目开发中，我们将更加注重代码的规范性和可维护性，提前规划好项目架构，避免出现零散碎片化的问题。同时，我们会加强团队协作，合理分工提高沟通效率，确保项目能够顺利推进。我们也会更加注重用户体验，对 UI 界面进行更加精心的设计，提升系统的整体品质。这次项目的经验将成为我们未来项目开发的宝贵财富，激励我们不断进步和创新。