

灵感日历 Qt-c++项目

北京大学 2025 春 • 程序设计实习 • 大作业项目报告
徐海毓 陈彦恺 王诺初

一、程序功能介绍

1. 概述：

“灵感日历”（Inspiration Calendar）是一款融合日历、天气、每日灵感与历史事件的工具类应用，主打简约设计和内容激励。核心功能包括：日历展示（突出“今日”视觉）、实时天气集成（通过 API 获取气温、风力等数据）、每日灵感推送（名言金句与图片）、历史上的今天事件展示。整体采用卡片式布局和图文混排的 UI 风格，旨在为用户提供实用性与美学兼具的日常陪伴体验。

2. 用户账户系统



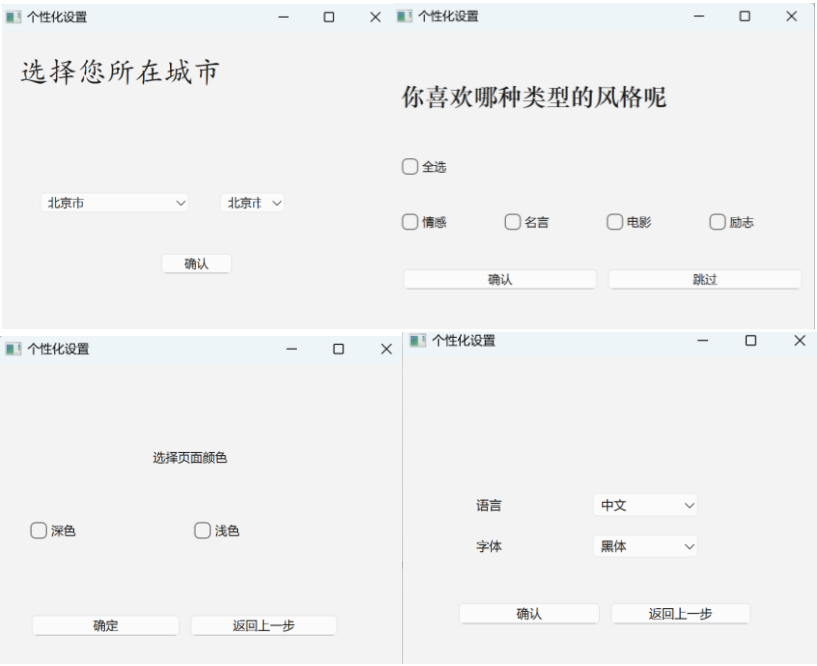
（1）完整的登录/注册流程：利用 SQLite 本地数据库记录用户信息，若已有账户可直接登录；若用户名与密码不能对应则提示错误，需要注册。注册时，账号不能与已有账号重复。

（2）密码安全策略：

①加密存储与长度限制（默认非明文，但选择“显示”后可显示）

②密码二次确认机制

3. 个性化设置



(1) 四级设置流程:

①城市选择: 省份→地级市二级联动

②每日一句类型偏好:

1) 情感/影视/名言/励志

2) 选择框设置: 点击“全选”可便捷全选, 再次点击“全选”则全不选, 若只选择部分选项则显示半选。

③视觉模式: 深色/浅色主题

④语言文字: 中英语言→字体集与语言联动

示例:

浅色+黑体模式



深色+楷体模式:



4. 灵感日历系统

(1) 日历界面：既提供逐月变动选项，同时使用下拉框可便捷选择月份，周末日期特殊着色。

(2) 核心交互：

①每日笔记：点击日历界面的某个日期，可撰写今日笔记（笔记自动保存）。切换页面或再次登录当前账号，仍可加载对应日期的笔记。

②今日心情：点击笔记区域上方表情，即可将表情与表情说明文字添加至今日笔记。

③灵感值标记：当日首次交互触发，同时通过光晕效果（金色/橙色）区分灵感强度。

(3) 灵感三连：

①生成随机组合：

1) 名言警句（根据个性化设置中的每日一句类型偏好随机推荐）、关联图片、思考问题

2) 数据源：inspirations.json

②弹窗组件：

1) 自适应布局

2) 图片加载容错机制



(4) 历史上的今天：

①根据今天日期，自主显示历史上该日期发生过的大事件，并设置问题



②时空模拟器：

1) 集成 DeepSeek 的 API，与之交互，在选择选项后，deepseek 将会给予反馈。

2) 结构化响应：事实、短期影响分析、长期影响分析、追问问题、预测验证问题。

(5) 灵感星图：

①可视化算法：基于日历网格布局，可看到本月有灵感记录的日期上有黄色光点。

②灵感星图可导出为 PNG 文件并保存至本地。

(6) 天气信息服务

①实时数据显示：集成网站 API，每小时更新数据（由 WeatherManager 信号驱动）

②页面显示温度范围、天气状况、湿度、风力

二、项目各模块与类设计细节

1. 用户系统模块

(1) cal_login

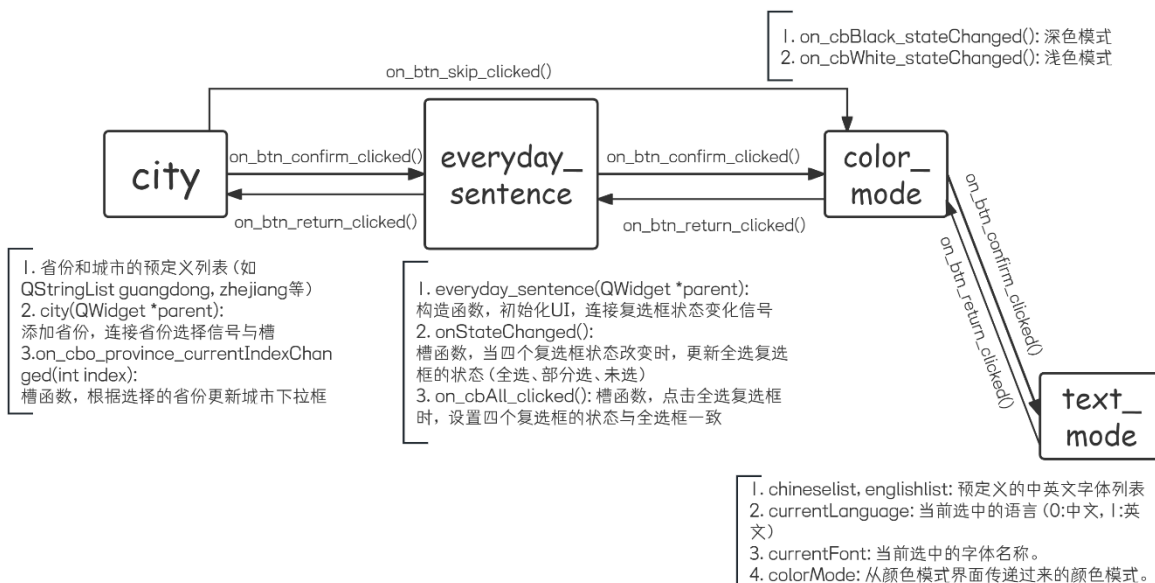
cal_login(QWidget *parent)	初始化 UI，设置窗口标题和密码输入框模式；调用 initStyle() 设置整体蓝色简约样式；调用 initDatabase() 初始化数据库，若无 userInfo 表则创建并插入默认用户。
void initStyle()	设置窗口控件 (QWidget、QLineEdit、QPushButton) 的样式表 QSS，实现蓝白风格、圆角输入框和按钮、高亮边框等视觉效果。
void initDatabase()	初始化并连接 SQLite 数据库 user.db，检查 userInfo 表是否存在，如不存在则自动创建。
void on_btn_reg_clicked()	槽函数：点击“注册”按钮时，跳转到注册界面 (reg 窗口)，并隐藏当前登录窗口。
void on_btn_login_clicked()	槽函数：点击“登录”按钮时，从数据库中验证输入的用户名和密码是否匹配，若正确则跳转到主窗口 (MainWindow)，否则弹出提示框。
void on_btn_quit_clicked()	槽函数：点击“退出”按钮时，关闭当前窗口。
void on_checkBox_clicked(bool checked)	槽函数：点击“显示密码”复选框时，切换密码框显示模式 (明文 / 密文)。

(2) reg

reg(QWidget *parent)	初始化 UI，设置窗口标题，设置密码框为密码模式。
on_btn_reg_clicked()	槽函数，点击注册按钮时进行输入验证，通过则插入新用户到数据库，并跳转到城市选择界面。
on_btn_return_clicked()	槽函数，点击返回按钮时跳转到登录界面。

2. 个性化设置模块

四个模块的链接见如下示意图，通过槽函数实现合理的跳转、自身功能与返回。



3. 核心功能模块

(1) MainWindow:

①关键成员:

- 1) inspirationScoreMap: 存储日期对应的灵感值 (0 表示无, 1 表示有)。
- 2) notes: 存储日期对应的笔记内容 (QMap<QDate, QString>)。
- 3) calendarWidget: 自定义日历控件 (InspirationCalendarWidget)。
- 4) weathermanager: 天气管理对象。
- 5) m_aiService: AI 服务对象 (用于历史事件分析)。
- 6) currentLanguage: 当前语言。
- 7) currentFontName: 当前字体名称。

②方法:

- 1) MainWindow(QWidget *parent): 构造函数, 初始化 UI, 加载灵感数据, 替换默认日历为自定义日历, 设置布局, 连接信号槽, 初始化天气和 AI 服务。
- 2) loadStyleSheet(const QString &fileName): 加载并应用样式表 (根据颜色模式)。
- 3) setColorMode(const QString &mode): 设置颜色模式 (深色或浅色) 并加载对应样式表。
- 4) setLanguage(int languageIndex): 设置语言。
- 5) setFontName(const QString &fontName): 设置字体。
- 6) dataFilePath(): 返回数据文件路径 (用于存储笔记和灵感值)。
- 7) loadInspirationData(): 从 JSON 文件加载笔记和灵感值数据。
- 8) saveInspirationData(): 保存笔记和灵感值数据到 JSON 文件。
- 9) onInspirationClicked(): 槽函数, 点击“灵感三连”按钮时随机展示一条灵感 (名言、图片、问题)。
- 10) onHistoryEventClicked(): 槽函数, 点击“历史事件”按钮时模拟历史事件分析 (调用 AI 服务)。
- 11) handleAiAnalysis(const QJsonObject &result): 处理 AI 返回的历史事件分析结果并显示。
- 12) handleAiError(const QString &errorMessage): 处理 AI 服务错误。
- 13) onExportStarMapClicked(): 槽函数, 导出星图 (将当前月份的日历以星空背景和光晕效果绘制成图片)。
- 14) weatherUpdatedUI(): 更新天气信息到 UI。

(2) AIService:

①关键成员:

- 1) m_networkManager: 网络访问管理器。
- 2) m_apiEndpoint: API 端点 URL。
- 3) m_apiKey: API 密钥。
- 4) m_currentReply: 当前网络回复对象。
- 5) m_progressDialog: 进度对话框。

②方法:

- 1) setApiKey(const QString &apiKey): 设置 API 密钥。
- 2) setApiEndpoint(const QString &endpoint): 设置 API 端点。
- 3) generateHistoricalAnalysis(const QString &eventTitle, const QString &question, const QString &choice,

QWidget *parentWidget): 生成历史事件分析（构造请求，发送，并处理响应）。

4) constructApiRequest(const QString &prompt): 构造 API 请求的 JSON 数据。

5) handleApiResponse(QNetworkReply *reply): 处理 API 响应，解析 JSON 并发出信号。

6) showError(const QString &message, QWidget *parent): 显示错误消息

（3）InspirationCalendarWidget（日历组件）

①关键成员：scoreMap: 存储日期和灵感值的映射。

②方法：

1) setScoreMap(const QMap<QDate, int> &map): 设置灵感值映射并触发重绘。

2) paintCell(QPainter *painter, const QRect &rect, QDate date) const: 重写绘制单元格，如果日期有灵感值则绘制光晕。

（4）WeatherManager（天气管理）：

①功能：管理天气数据，提供天气更新信号。

②信号：weatherUpdated(): 当天气数据更新时发出。

③方法：提供获取温度、天气、湿度、风力等数据的接口。

4. 数据管理模块

（1）SQLite 数据库（user.db）

①表：userInfo

②字段：username（文本），password（文本）

③存储用户账户信息

（2）JSON 文件

①inspiration_data.json: 存储笔记内容和灵感值

结构：{"notes": {"日期": "笔记内容", ...}, "scores": {"日期": 灵感值, ...}}

②parallel_cards.json: 存储历史事件分析结果（灵感卡片）

③inspirations.json: 存储灵感三连的内容（名言、图片路径、问题）

三、小组成员分工情况

代码部分：

1. 徐海毓（组长）：主界面全部功能（包括个性化设置中颜色、字体与主页面联动，每日笔记和心情功能，每日灵感三连，日历灵感值显示、灵感星图绘制与本地储存、DeepSeekAPI 的接入），界面美化和按钮响应。

2. 陈彦恺：天气获取模块、天气展示模块。

3. 王诺初：登录和注册模块，用户账户系统。

项目文档由三人共同构思，陈彦恺完成；展示程序功能的视频录制由徐海毓完成，作业功能说明文档由王诺初完成初稿，徐海毓完成最终文档。

四、项目总结与反思

1. 技术亮点：

在平行时空推演模块，我们与 AI 引论联动，集成 DeepSeek API 构建“历史上的今天”平行时空分析，既增强了用户与项目的可交互性，同时更加凸显了“灵感”日历的特殊性。另外，这个项目还进行了灵感值的可视化表达，相比于死板的日历，我们加入了“灵感值”的元素，让用户在使用软件的过程中更有趣也更有收获感。无论是在日期当天首次点击日历，还是写下每日笔记，用户都能够获得灵感值的奖励，每月还能够生成每月灵感星图的照片进行纪念保存。

2. 遇到的问题和收获：

- （1）对“面向对象”的理解更加深入：从用户系统中理解类的封装，从接口设计中理解多态。
- （2）学会使用 QT 创建项目，从公式化使用到真正理解信号与槽函数，也学会了在 ui 界面无法实现一些选择关联时手搓代码实现复杂功能。
- （3）对于 Github 的用法理解更加深入。
- （4）遇到问题时更加冷静，报错时学会看 QT 错误提示，学会看官方文档。
- （5）体验到模块化的好处，在组织代码时有架构使思路更加清楚。
- （6）个性化设置：根据在登录时的选择，实现主页面的颜色和字体的更改。

3. 反思：

在 QT 教学课之后，我们汲取上届学长学姐们的经验，坚持每周线下开会碰一下进度，非常好地 push 我们尽可能早点把项目的基础功能做完。总的来说我们并没有赶 ddl，较为顺利的完成了大部分的功能，很大程度上也得益于我们充分利用了五一假期的时间完成主要模块的搭建任务。

当然，我们的项目仍有一些可以改进和优化的部分，如说页面的美化、异常处理机制，在与 DeepSeek 交互方面还未实现多轮交互与过往聊天记录的保存。对于“灵感值”，我们也希望可以增加类似于“灵感值商店”的功能，让用户兑换礼品奖励。

在项目发布之后，我们依然会进一步处理其中的问题并且改进更多的功能。再次感谢所有的老师和助教，也感谢我们小组的全部成员，真正实现了一次从无到有的项目开发过程