

## 一. 程序功能介绍

1. 注册和登录界面与机制
2. 日记部分可以切换字号, 字体, 加黑, 斜体, 更改颜色, 插入图片
3. 随写部分可以选择事件图标并补充相关信息, 并可以自由增加删减事件
4. 日记和随写的储存和加载 (包括意外退出时的储存)
5. 回忆部分可以切换日期并可以点击加载之前的日记 (如果当日有写日记)
6. 信息部分可以查看用户名和日记随写的存储地址
7. 程序主界面下方每隔 30s 滚动名人名言
8. 在程序主界面摁住鼠标左键拖动会有墨迹效果

## 二. 项目各模块与类设计细节

### 1. Page\_Login: 登录窗口

- a) 数据库交互: 使用 QSqlDatabase 来处理用户信息的存储和验证。
- b) 数据持久化: 用户信息存储在本地 SQLite 数据库中, 通过标准 SQL 操作进行管理。
- c) on\_button\_login\_clicked(): 处理登录逻辑, 验证用户凭证。
- d) on\_button\_sign\_clicked(): 跳转到注册界面。
- e) on\_pushButton\_clicked()与 on\_checkBox\_stateChanged(): 控制密码的显示隐藏。

### 2. regis: 注册窗口

- a) 数据库交互: 利用 QSqlDatabase 来处理用户注册信息的存储和验证。
- b) on\_button\_register\_clicked(): 执行注册流程, 包括密码验证、数据库操作, 并在成功后跳转回登录界面。
- c) on\_button\_back\_clicked(): 返回登录界面。

### 3. Diaries: 日记界面

- a) 用户界面: 通过 Ui::Diaries 与.ui 文件关联, 管理日记编辑和格式化的各种工具。
- b) 类初始化的时候加载今日已经写好的日记内容。
- c) 文本格式化 (如粗体、斜体、下划线) 的槽函数处理文本编辑器中的文本格式更改。
- d) on\_toolButtonSave\_clicked(): 保存日记内容到文件。
- e) on\_toolButtonPicture\_clicked(): 插入图片到文本编辑器。
- f) closeEvent: 关闭窗口时触发, 负责保存所有文本框中的数据到文件。

### 4. Jottings: 随写界面

- a) 构造函数中, 初始化了用户界面并设置了当前日期和文件路径。此外, 还构建了多个布局来组织界面元素, 如按钮和文本框, 并连接了相关的槽函数。
- b) mainLayout, layout: 垂直布局, 用于组织文本框和按钮
- c) textBoxes, comboBoxes: 分别存储多个文本框和下拉菜单, 以便管理动态生成的界面元素。
- d) contextMenu: 右键菜单, 用于文本框的删除。
- e) closeEvent: 关闭窗口时触发, 负责保存所有文本框中的数据到文件。
- f) contextMenuEvent: 处理右键菜单事件, 允许用户对特定文本框执行操作, 如

删除。

- g) onAddButtonClicked, onSaveButtonClicked, onDeleteTextBox: 槽函数, 分别处理添加、保存和删除文本框的功能。
- h) createTextBox 和 createComboBox 方法用于动态创建并配置文本框和下拉菜单。这些方法支持动态界面的实时更新和元素配置。

#### 5 . Recollections: 回忆界面

- a) 用户界面: 通过 Ui::Recollections 与.ui 文件关联, 配置日历组件以显示和管理回忆录。
- b) 使用 QCalendarWidget, 提供可视化日期选择, 并对特定日期进行自定义格式化。
- c) 日历自定义: 日历通过样式表自定义视觉样式, 如按钮图标、背景颜色、文本颜色等。
- d) onDataClicked(): 处理日历日期点击事件, 用于显示特定日期的日记。

#### 6 . Settings: 信息界面

- a) 信息显示: 界面主要用于显示静态信息, 如用户名、存储地址和开发者信息。

#### 7 . Form: 程序主界面, 包含四个功能界面

- a) 通过 Ui::Form 与.ui 文件关联, 使用 QStackedWidget 来管理多个页面 (日记、随笔、回忆、信息)。
- b) 多功能页面: 包括墨迹绘制部件 (InkWidget) 和各种子页面如 Diaries、Jottings、Recollections 和 Settings。
- c) updateQuote(): 定时更新显示的名言。
- d) closeEvent(): 确保所有子页面在窗口关闭时也正确关闭。

#### 8. DiaryDialog: 加载历史日记的界面

- a) 用户界面: 使用 QTextEdit 来显示日记内容, 设置为只读模式以防止修改。
- b) loadHtmlFile: 从文件路径读取 HTML 内容, 并在文本编辑器中显示。

#### 9 . InkWidget: 实现墨迹效果的部件

- a) 点追踪与颜色处理: 使用 QVector 来存储墨点位置和颜色信息。
- b) mousePressEvent()、mouseReleaseEvent(): 处理鼠标点击释放事件。
- c) mouseMoveEvent(): 捕捉鼠标移动并添加墨迹点。
- d) paintEvent(): 绘制界面上的墨迹。
- e) updateInkPoints(): 定期更新墨迹的透明度, 模拟墨迹逐渐消失的效果。

### 三 . 小组成员分工情况

张致铨:

- 明确, 细化目标并分配任务
- Diaries 的编写
- Settings 的编写
- InkWidget 的编写
- 绝大部分代码的完善, 整合工作, 包含多个页面之间的串联, Form 中各个功

能界面的整合添加，修改不满足要求的细节等等。

杨灵凡：

- 创意的提出
- Page\_Login 的编写
- regis 的编写
- Form 的 ui 界面和名人名言部分的代码实现

羊瑞：

- Recollections 的编写
- Jottings 的编写

#### 四 . 项目总结与反思

1. 尽管做了很多很多使用 github 进行协作的尝试，但最终还是遇到了很多困难，没能成功。如果老师能有一些对 github 的指导就好了，新手真的感觉上手有难度
2. 项目总体推进很顺利，归功于所有组员的不懈努力。大家不仅会分工写完各自的部分，还会相互帮助解答队友遇到的问题。
3. 本项目的 cpp 文件代码数超过 1000 行，并且大量内容是通过 ui 文件实现的，同时我们需要花很多时间进行美化（如寻找图片），所以我们认为我们的这个大作业的工作量还是比较大的。
4. 总的来说我们的程序界面美观，还有很多像鼠标墨迹，名人名言，按键反馈这些细节（非常费心费力!），完成度应该算非常高了。

完蛋我被单身狗包围了队（组号 49）

张致铨

杨灵凡

羊瑞