阳光学院

思政课社会实践（一）调研报告

院 系: 人工智能学院

专业班级: 计算机科学与技术

姓 名: 李阳坚

学 号: 2227010157

任课老师: 金莉 \_\_\_

# QIHan的Mytxt程序开发文档

目录

[QIHan的Mytxt程序开发文档 2](#_Toc177988256)

[一、 项目名称 2](#_Toc177988257)

[二、 开发环境 2](#_Toc177988258)

[三、 项目概述 2](#_Toc177988259)

[四、 具体实现 2](#_Toc177988260)

[五、 主要功能 2](#_Toc177988261)

[六、 数据库设计： 3](#_Toc177988262)

1. 项目名称

QiHan在线文档

1. 开发环境

版本1.0

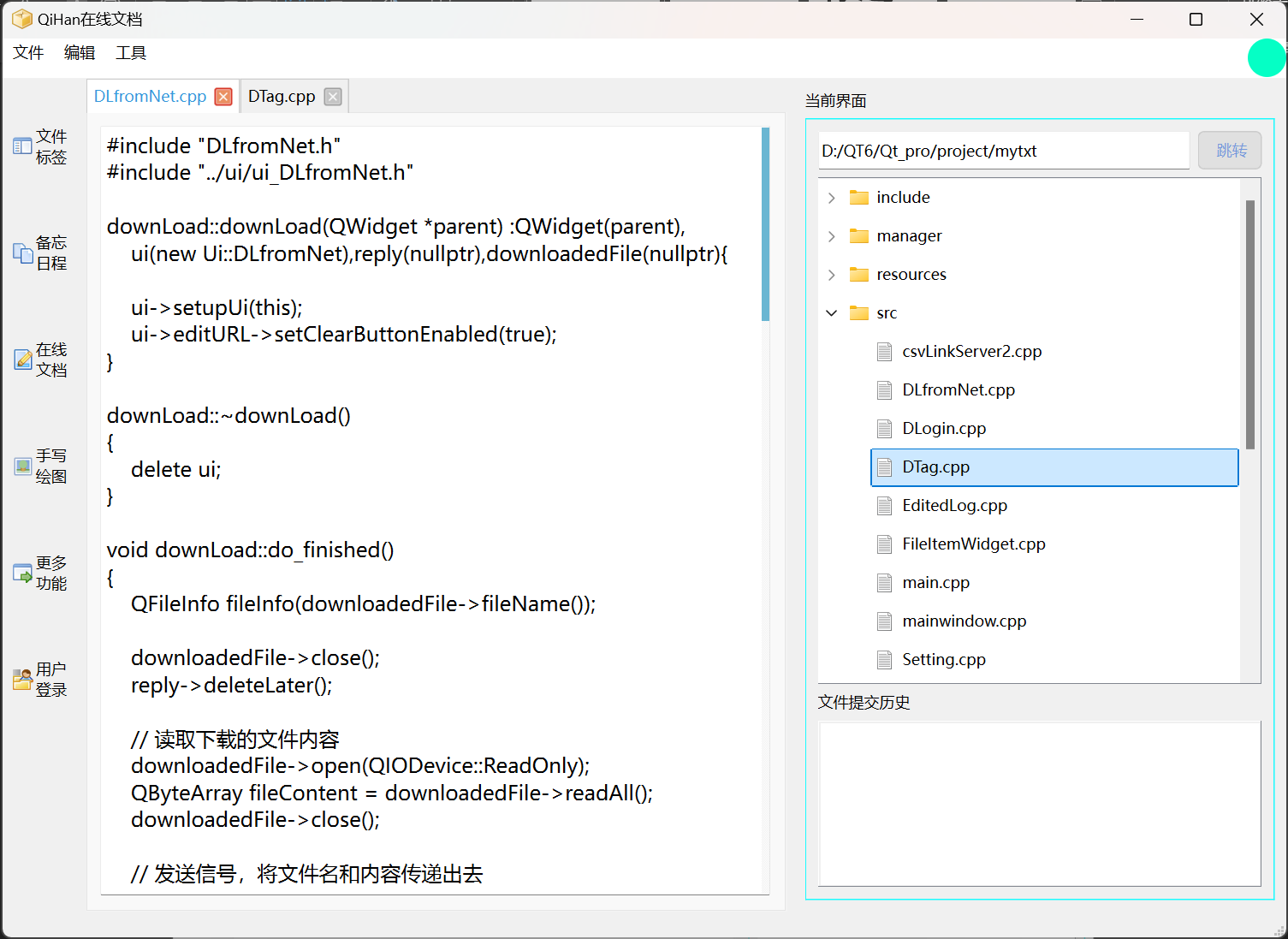
* IDE: Qt Creator
* 编程语言: C++
* Qt版本: 6.6.0

1. 项目概述

本项目旨在开发一个在线文档编辑工具，支持多种文档格式（文本文件和CSV文件）的编辑、保存和下载。用户可以通过该工具轻松管理文件，支持多标签的界面操作。

1. 具体实现

mainwindow类



左侧菜单栏包含文件标签，备忘日程，在线文档，手写绘图，更多功能，用户登录，信息展示

前五个功能在点击触发后均在右侧“当前界面展示”（widgetr类），用户登录，信息展示采用模态的浮动对话框。

中间是文件展示窗口，当用户点击所需打开文件时，槽函数openFile会触发新建选项卡的操作。

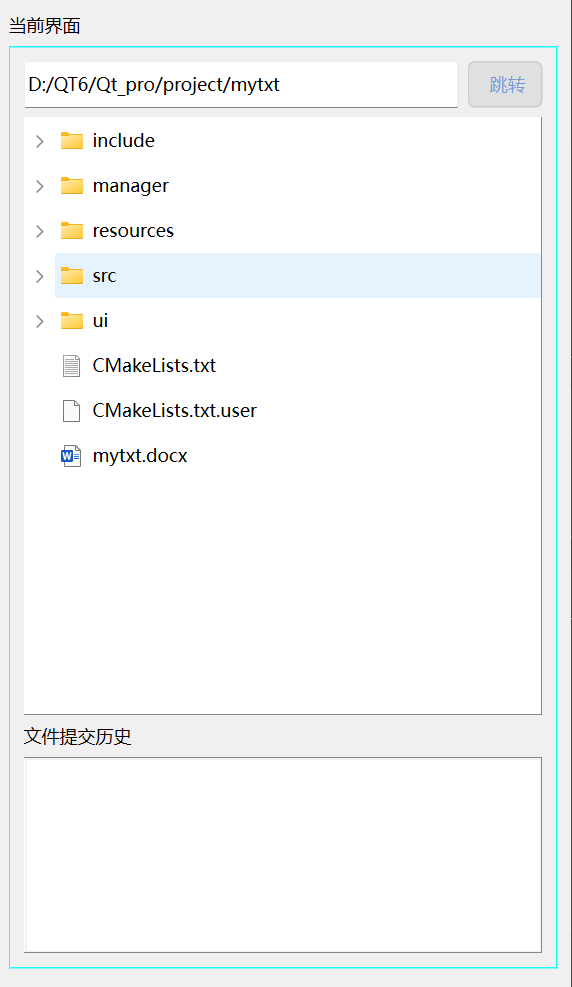
DdbSqlite， dbMysql已在mainwindows初始化，子类想要获得数据库操作权都要接收他们的实例化指针。

RecentFilesManager类

RecentFilesManager 是一个管理最近打开文件的类。它支持添加文件到历史记录、加载和保存历史记录，以及填充菜单以显示最近文件。

他的实现放在mainwindows的工具栏中，每一份打开过的文件路径会保存到注册表recentFilesManager->populateRecentFilesMenu(ui->recentFile);用户可以通过工具栏打开曾经打开过的文件，

WFileTag类



WFileTag 类是一个文件标签。它提供与文件交互、查看文件元数据和处理，标签相关的到期日期的功能

DBSQlite：这里没有使用，主要传递给TagItemDelegate 委托构造函数；

QFileSystemModel作为一个数据模型，它提供了文件和目录的信息（如名称、路径、大小等），供视图（如 QTreeView）使用。

TagItemDelegate：用于以自定义方式渲染标签，增强标记项的视觉表现。

updateFileList函数耦合了ServerManager的onListFilesFinished

主要在用户点击文件时，会获取历史提交过的文件展示在listWidget中，这些文件在点击后一样能展示到tabwidget

TagItemDelegate 类

1. 实现效果

TagItemDelegate 类是一个自定义的视图委托，用于在 Qt 应用程序中处理文件标签的显示和交互。其主要功能包括：

绘制项：重载 paint 方法，让用户在视图内就能触发标签和远程传送操作。

通过重载 editorEvent 方法，处理鼠标点击等事件

上下文菜单：实现 showContextMenu 方法，在用户右键单击文件项时弹出上下文菜单

DTag：用于创建和编辑标签的对话框。 用addTag 方法与 DTag 进行交互，以允许用户添加标签。

DBSQlite：提交标签增删改查的sql语句支持

ServerManager：负责与远程服务器的交互。TagItemDelegate 通过 serverManager 提交文件和获取目录信息。

ServerManager 类

ServerManager类是一个用于处理与远程服务器交互的组件，主要功能包括：

提交文件 commitToServer  获取提交历史 getCommitHistory 下载文件downloadFile  获取目录文件列表 getFilesInDirectory

WidgetFunctional 类设计与实现

1. 概述

WidgetFunctional 类是基于 Qt 框架开发的自定义 QWidget，旨在为图形用户界面（GUI）提供功能按钮的管理与交互。该类通过信号和槽机制响应用户操作，允许用户在不同功能之间快速切换，适合在各种应用程序中使用。

2. 类的定义

* *Dlogin\* dlogin*: 指向登录功能的对象。
* *Dlogin\* dlogin*: 指向登录功能的对象。

1. 主要功能
2. 文件管理
   * 打开文件（支持TXT和CSV格式）
   * 保存文件
   * 关闭文件标签
   * 下载文件
3. 多标签支持
   * 用户可以同时打开多个文件，每个文件在不同的标签中显示。
   * 标签可以关闭，且关闭后会更新最近打开的文件列表。
4. 文件操作
   * 对CSV文件的行和列进行添加和删除操作。
5. 设置界面
   * 提供用户设置选项。

代码结构

1. 主窗口类 MainWindow

* 构造函数: 初始化用户界面和主要组件。
* 初始化方法:
  + initFunc(): 初始化功能组件，包括文件标签、在线文档、日程等。
  + initSpli(): 初始化分割器，布局主要组件。

2. 文件操作

* openFile(const QString &filePath): 打开文件并加载到新标签。
* on\_actionopen\_triggered(): 响应打开文件菜单项的点击事件。
* on\_actionsave\_triggered(): 响应保存文件菜单项的点击事件。

3. 标签管理

* createNewTab(...): 创建新标签并添加到标签控件。
* getCurrentTab<T>(): 获取当前选中标签，并进行类型转换。

4. 信号与槽

* 使用Qt的信号和槽机制进行不同组件间的交互。例如，通过信号传递文件路径，实现文档的打开和管理。

界面设计

* 菜单栏: 包含文件操作相关的菜单项。
* 工具条: 提供常用操作的快捷按钮。
* 标签控件: 显示当前打开的文档，每个文档在不同的标签中。

使用说明

1. 启动应用程序。
2. 通过菜单栏的“文件”选项打开文档。
3. 对文档进行编辑，使用“保存”功能保存更改。
4. 使用“关闭”选项关闭当前文档标签。

未来计划

* 增加更多文件格式支持。
* 提升界面用户体验。
* 实现云端存储功能，支持在线协作。

参考文献

* Qt Documentation
* C++ Standard Library

1. 数据库设计：

**表结构**

1. **FilePaths**
   * id (主键)：唯一标识每个文件路径。
   * file\_path (唯一文件路径)：存储文件的路径，确保路径的唯一性。
   * expiration\_date (过期日期)：可选字段，用于标记文件的过期时间。
2. **Tags**
   * id (主键)：唯一标识每个标签。
   * file\_id (外键)：关联到 FilePaths 表的 id，表示该标签属于哪个文件。
   * tag\_name (标签名称)：存储标签的名称。
3. **Annotations**
   * id (主键)：唯一标识每个批注。
   * file\_id (外键)：关联到 FilePaths 表的 id，表示该批注属于哪个文件。
   * annotation (批注内容)：存储批注的具体内容。
4. **Submissions**
   * id (主键，自增)：唯一标识每个提交记录。
   * file\_path (文件路径)：记录提交的文件路径。
   * submit\_time (提交时间)：记录提交的时间，默认为当前时间。

**1. users**

* **username** (VARCHAR(10), 主键)：唯一标识用户的用户名。
* **password** (VARCHAR(15), NOT NULL)：用户的密码，不能为空。
* **avatar** (BLOB)：存储用户头像的二进制数据。

**2. user\_info**

* **id** (INT, AUTO\_INCREMENT, 主键)：唯一标识每个用户信息记录。
* **username** (VARCHAR(10), 外键)：与 users 表中的 username 字段关联，表示该信息属于哪个用户。
* **name** (VARCHAR(20))：用户的真实姓名。
* **motto** (VARCHAR(50))：用户的座右铭或个性签名。
* **gender** (ENUM('Male', 'Female', 'Other'))：用户的性别。
* **birthday** (DATE)：用户的生日。
* **location** (VARCHAR(100))：用户的居住地。
* **company** (VARCHAR(100))：用户的工作单位。

**外键约束**

* **FOREIGN KEY (username)** REFERENCES users(username) ON DELETE CASCADE：如果用户被删除，相关的用户信息也会被自动删除。