CashNote项目报告

程序功能介绍

CashNote 是一款为学生设计的记账软件。该软件能轻松记录“饮食”，“购物”等日常开销，
统计本月收支（因为学生党的收入一般是家长每月提供生活费），
并且用条形图实时显示收支比，可直观地估计每月预算与花销。

此外，CashNote 可以将最近7天的花费转化为柱状图，清晰地显示近期消费情况。
同时可以按月或者按日显示各个类别开销的占比，并且利用大语言模型根据该占比提出合理的建议。

各模块与类细节

DatabaseHandler

`DatabaseHandler` 提供了与 SQLite 数据库交互的功能，
主要用于管理主目录下名为 `cashnote.db` 的数据库，
用于学生记账应用的数据存储和管理。

数据库连接和初始化：

在构造函数中，使用 `QSqlDatabase` 建立了与 `SQLite` 数据库的连接，
并打开了名为 `cashnote.db` 的数据库文件。如果连接失败，会输出相应的调试信息。
我们的数据库由两个表组成， `category` 记录了账单的类别，例如“饮食”，“娱乐”，“收入”等；
`record` 记录了账单的具体数额，日期，备注等信息。

获取当月总支出和总收入：

`getThisMonthTotalSpent()` 和 `getThisMonthTotalEarned()` 方法分别返回当前月份的总支出和总收入。
这些方法执行 SQL 查询并返回相应的整数值。

获取当月记账记录数量：

`getThisMonthCount()` 方法返回当前月份的记账记录数量。

按日期获取总支出：

`getTotalSpentGroupByDate()` 方法返回最近7天内每天的总支出，包括日期和支出金额。
如果某一天没有记账记录，会返回0。

按类别获取支出统计：

`getSpentGroupByCategory()` 和 `getSpentGroupByCategoryForMonth()` 方法分别按指定日期或月份返回每个记账类别的总支出。

获取当月各类别支出百分比：

`getCategorySpentPercentageForMonth()` 方法返回当前月份每个记账类别的支出占比，以字符串列表形式展示。

插入、更新和删除记账记录：

- `insertRecord()` 方法用于插入新的记账记录。
- `updateRecord()` 方法用于更新现有的记账记录。
- `deleteRecord()` 方法用于删除指定ID的记账记录。

获取所有记账类别：

`getAllCategories()` 方法返回数据库中所有的记账类别，包括类别名称和对应的ID。

一些细节

将几乎所有的字符串查询都集中到了头文件中，并且放在命名空间 `queries` 下，方便调用检查。

Widget

这是一个 `QWidget` 类，作为程序的主界面，包含以下组件:

- 一个侧边栏 `Sidebar`，可用于在不同的页面间切换。
- 数据展示界面 `QStackedWidget` 对象 `dataWidget`，用于显示不同的页面。
  - `widget1` 主要用于数据的编辑
  - `widget2` 专注于数据的可视化
  - `widget3` 尝试用大语言模型给出合理的建议。
- 初始化了 `DatabaseHandler`，以及一个网络处理器 `network`。

MainDataWidget

这个类负责绘制界面1，同时处理用户对数据的增添，修改和删除操作。

界面布局

整个界面分为上下两个区域。

- 上方区域展示 CashNote 的标志，并且根据当月的数据生成收入和支出的数据条。
- 下方区域左侧展示所有数据，右侧提供编辑的窗口。
  - 显示数据按照 “日期+类别+金额” 的方式显示
  - 编辑框会随着选中的数据更新其中的内容，也可以用来查看备注。
  - 编辑框下方提供了两个按钮，包括“新增”和“修改”

设计思路

构造函数 `MainDataWidget::MainDataWidget(DatabaseHandler *dbHandler_, QWidget *parent)`：

- 初始化了各种子部件，包括上方展示部件 `aboveWidget` 和下方操作部件 `belowWidget`。
- 设置了布局和样式，将各部件添加到主布局中。
- 调用了 `refresh()` 方法初始化界面数据，并连接了按钮的点击事件到对应的槽函数。

`setThisMonthWidget()` 方法：

- 根据当前月份从数据库中获取本月的总收入和总支出。
- 根据收入和支出的比例动态调整显示的条形图（使用 `earnedBar` 和 `spentBar`）的宽度。

`addRecord()` 方法：

- 将用户在界面上输入的记录插入到数据库中，使用了 `dbHandler->insertRecord()` 方法。

`drawListWidget()` 方法：

- 从数据库中查询所有记录，并在 `listWidget` 中显示每一条记录。
- 每个记录通过 `DataItemWidget` 自定义部件显示，支持删除操作。

`deleteItem(QListWidgetItem *item)` 方法：

- 删除选定的记录条目，并从数据库中删除对应记录。
- 刷新界面以更新显示和统计数据。

其他槽函数：

- `onSaveButtonClicked()`：处理保存按钮点击事件，调用 `addRecord()` 方法并刷新界面。
- `onUpdateButtonClicked()`：处理更新按钮点击事件，更新选定记录的信息并刷新界面。

- `onItemSelectionChanged()`：处理列表项选择变化事件，更新编辑部件的内容显示。

`refresh()` 方法：

- 刷新界面的方法，调用 `setThisMonthWidget()` 和 `drawListWidget()` 更新界面显示和数据。

DataItemWidget

是创建数据表的重要子部件，用于数据表的一行，显示了日期，类别，金额和一个删除按钮，可以方便地实现删除。

ChartWidget

用于管理主要的数据展示和交互功能，通过柱状图和饼状图两种方式可视化数据。

构造函数 `ChartWidget::ChartWidget(DatabaseHandler *dbHandler_, QWidget *parent)`：

初始化界面布局，包括左侧和右侧两个部分的 QWidget。
左侧部分用于显示柱状图（barChartView），右侧部分包括输入日期、生成按钮和显示饼状图（pieChartView）的组件。
初始化时调用 drawBarChart() 方法，绘制近7日的净开销柱状图。

`drawBarChart(const QList<QPair<QString, int>> &data)` 方法：

根据传入的日期和总支出数据绘制柱状图。
创建 `QBarSeries` 和 `QBarSet` 对象，将日期和对应的总支出数据添加到柱状图中。
设置图表的标题、动画效果，并将图表绑定到 `barChartView` 上显示。

`handleQuery()` 方法：

处理用户点击生成按钮的事件，根据输入的日期格式（年月或年月日）调用相应的数据库查询方法获取数据。
调用 `drawPieChart()` 方法绘制对应日期的消费类别饼状图。

`drawPieChart(const QList<QPair<QString, int>> &data)` 方法：

根据传入的消费类别和对应金额数据绘制饼状图。
创建 `QPieSeries` 对象，并遍历数据集合，为每个消费类别创建一个 `QPieSlice` ，并设置标签显示类别名称和金额。
设置图表的标题、动画效果，并将图表绑定到 `pieChartView` 上显示。

Network

这段代码实现了一个名为 `NetWork` 的类，主要用于与网络进行交互，实现了获取 access token、设置鉴权元素和修改询问语句的功能。以下是代码功能的详细解释：

析构函数 `NetWork::~NetWork()`：

- 在对象销毁时，释放 `reply` 对象和 `manager` 对象的内存。
- 如果 `reply` 不为空，调用 `deleteLater()` 和 `delete` 方法释放内存。

构造函数 `NetWork::NetWork()`：

- 初始化 `NetWork` 类的对象。
- 初始化了 `APIKey`、`APISecret` 和 `accessToken`，这些是与百度接口交互时所需的身份验证信息和访问令牌。
- 创建了 `QNetworkAccessManager` 类的对象 `manager`，用于处理网络请求。

成员函数 `handle_SetEnquireText(const QString& enquiretext)`：

- 修改询问语句，并向百度接口发送请求。
  - 这是因为综合考虑各个大模型厂商，调研发现大多数模型只提供Python接口，无法接入我们的Qt项目。
  - 只有少量模型提供C++接口，百度就是其中之一，故选择这个模型。
- 准备请求数据，构建 JSON 对象，包含 `role` 和 `content`。
  - `role` 为对话中的角色，由于只有单次请求，故只包含了 `user`。
  - `content` 则是请求询问的内容。
- 将 JSON 对象转换为字符串，并设置请求的 URL（包括 `accessToken`）和请求头。
- 使用 `manager` 发送 POST 请求，并通过 `QEventLoop` 等待响应完成。
- 如果没有错误，读取响应的 JSON 数据，并解析出需要的内容，得到一个QString用于后续处理。
- 如果成功解析，返回结果字符串，然后展示在对应的窗口中。
- 如果发生错误，在后台显示错误信息的消息框，并返回空字符串。用户观看到的界面没有信息。

小组成员分工情况

- 李代波：负责页面1及相关功能
- 杨雪巍：负责页面2及相关功能
- 黄柏喻：负责页面3及相关功能

项目总结与反思

总结

- 我们完成了对数据的增删改查，满足了记账软件最基本的要求
- 我们实现了对数据的可视化，使得数据更加清晰明了
- 我们引入了大语言模型，提出了人性化的建议

反思

- 数据编辑还比较粗糙，把所有数据都输出到一张表，不适合数据量较大的情况
- 目前涉及的收支项目品类较少，并且不支持自定义门类，后续可以优化这部分的实现以提供更人性化的服务
- 数据分析方面目前仅仅是展示了简单的图表，后续可以做更好的可视化
- 因为很多组件需要一直存在，所以很多类的析构函数没有释放资源，后续可能造成内存泄露
- 为了实现便捷，大模型调用过程目前是明文写下自己的APIKey，安全性不高，后续应当引入加密措施等保证安全
- 大模型的调用目前依赖网络通信，后续可以考虑在本地部署轻量级模型，在保证交互质量的基础上提高交互速度