😀

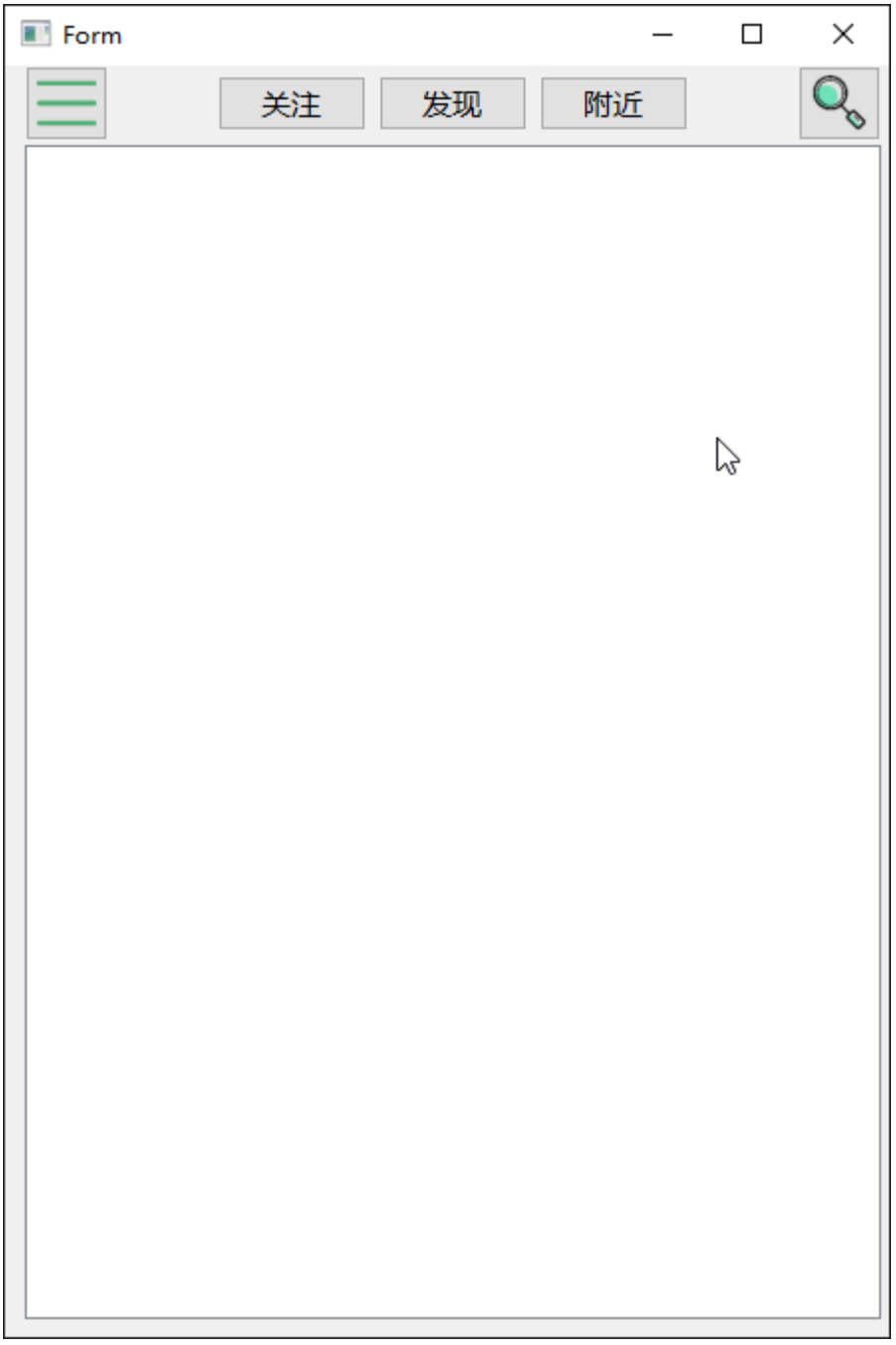
202506-电子信息工程生产实习-day3

1. 主页界面增加样式

在Home类中，修改构造函数参数，增加用户名、位置参数（头文件-源文件，同步运行）。

```
explicit Home(QString user,QString location,QWidget *parent = nullptr);
```

修改完毕后，运行程序，主页界面如图：



- **主页构造函数**

1. 设置标题：

```
setWindowTitle(QString("小红书-%1-%2").arg(user).arg(location));
```

2. 设置窗口组件样式：

```
void setHomeStyle();
```

设置主页的样式，包括上面一排的按钮和展示帖子的内容（tableWidget）。

```
void Home::setHomeStyle(){
    //设置按钮样式，
    ui->pb_set->setFlat(true);
    ui->pb_set->setStyleSheet(BORDER_FRAME4);

    //搜索按钮样式
    ui->pb_search->setFlat(true);
    ui->pb_search->setStyleSheet(BORDER_FRAME4);

    //关注、发现、附近。设置按钮扁平，加圆角样式表
    ui->pb_attention->setFlat(true);
    ui->pb_attention->setStyleSheet(BORDER_FRAME3);

    ui->pb_find->setFlat(true);
    ui->pb_find->setStyleSheet(BORDER_FRAME3);

    ui->pb_near->setFlat(true);
    ui->pb_near->setStyleSheet(BORDER_FRAME3);
    //默认设置，发现为加粗
    setBold(false,true,false);

    //设置表格tablewidget样式，
    //去掉行、列标题
    ui->tw_content->verticalHeader()->hide();
    ui->tw_content->horizontalHeader()->hide();

    //设置无边框
    ui->tw_content->setFrameStyle(QFrame::NoFrame);

    //设置表格网格不可见
    ui->tw_content->setShowGrid(false);

    //设置表格单元格，不可选中
    ui->tw_content->setSelectionMode(QAbstractItemView::NoSelection);
```

```
    //设置表格为两列，并指定每列的宽度
    ui->tw_content->setColumnCount(2);
    QRect rect = ui->tw_content->geometry();
    ui->tw_content->setColumnWidth(0/*第0列*/,rect.width()/2-9);  //中间预
    ui->tw_content->setColumnWidth(1/*第1列*/,rect.width()/2-9);

}
```

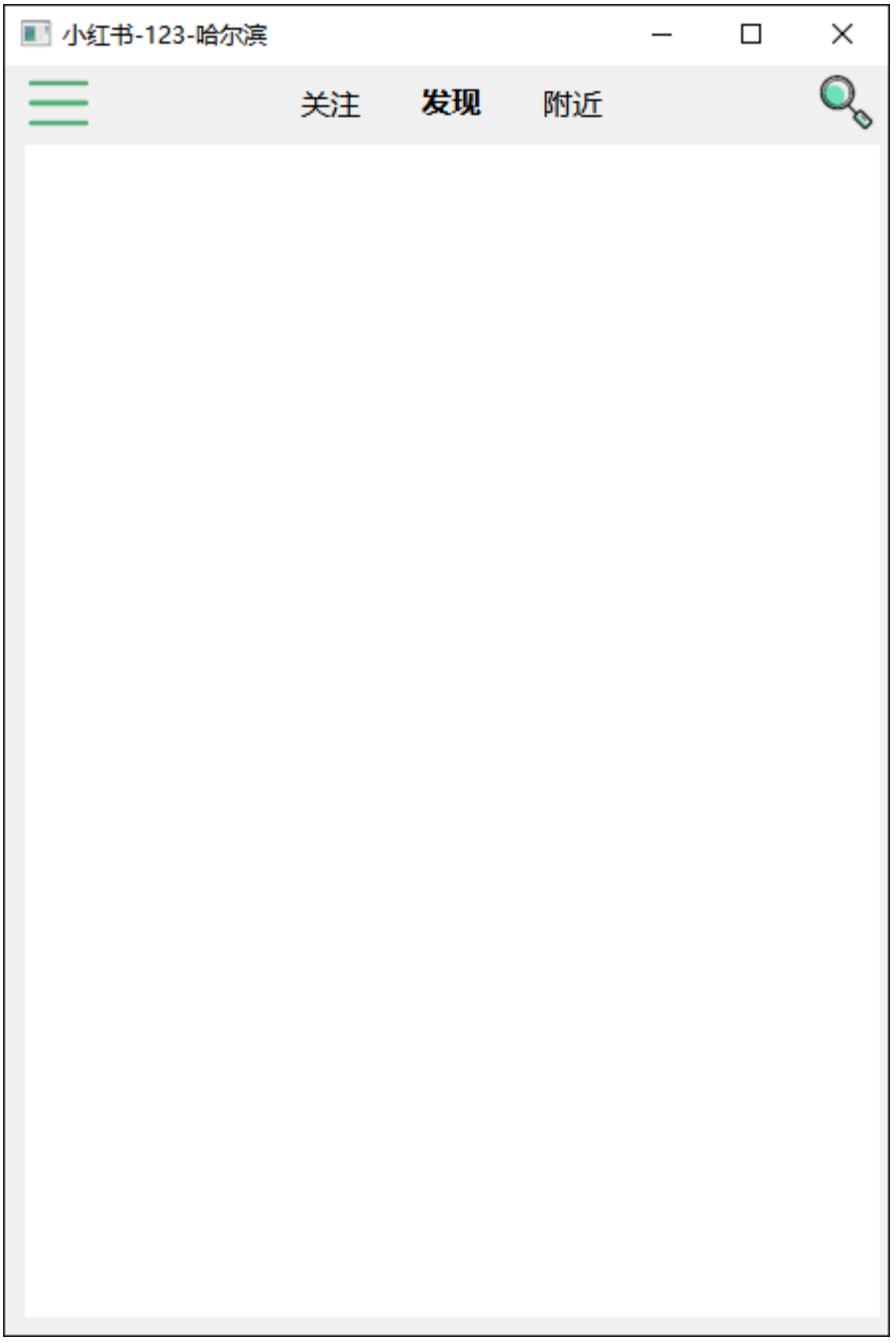
其中的 setBold 为手动添加的函数。

```
void Home::setBold(bool a,bool b,bool c){
    //关注：
    QFont font = ui->pb_attention->font();
    font.setBold(a);
    ui->pb_attention->setFont(font);

    //发现
    font = ui->pb_find->font();
    font.setBold(b);
    ui->pb_find->setFont(font);

    //附近
    font = ui->pb_near->font();
    font.setBold(c);
    ui->pb_near->setFont(font);
}
```

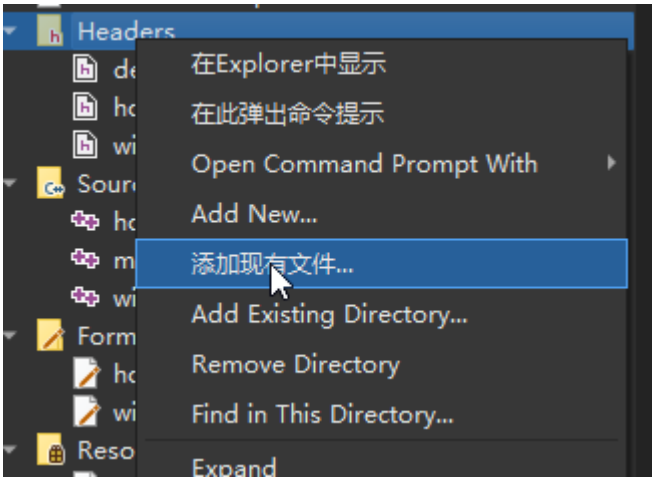
程序界面如图：

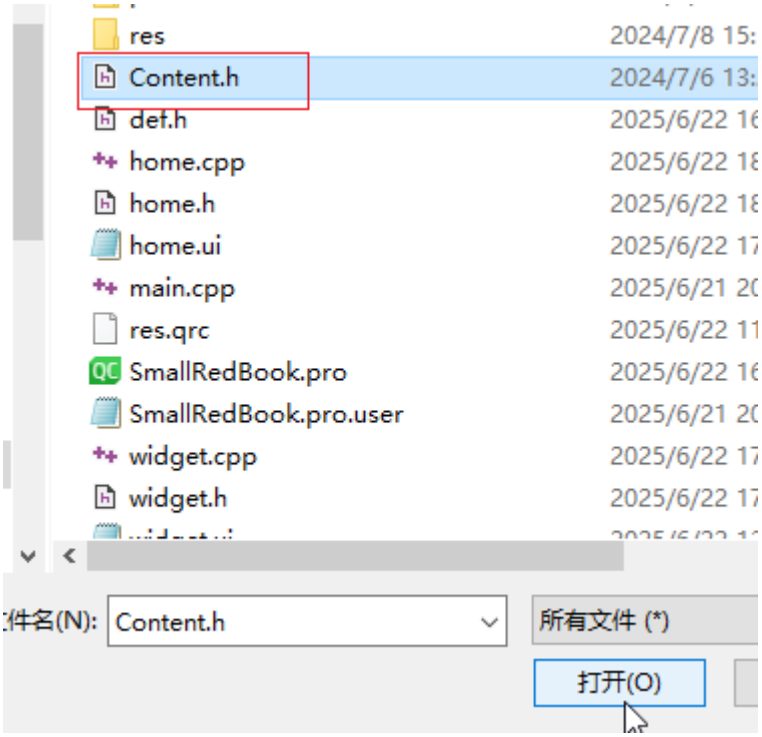


3. 解析json文件内容。

   Content.h 中包含了json内容对应的结构体数据 和 导入需要用的函数，导入主页内容配置文件：Content.h，添加到项目目录中

名称	修改日期	类型	大小
config	2024/7/8 14:22	文件夹	
pic	2024/7/8 14:22	文件夹	
res	2024/7/8 15:59	文件夹	
Content.h	2024/7/6 13:28	C/C++ Header	1 KB
def.h	2025/6/22 16:26	C/C++ Header	2 KB
home.cpp	2025/6/22 18:09	C++ Source	2 KB
home.h	2025/6/22 18:05	C/C++ Header	1 KB
home.ui	2025/6/22 17:10	UI 文件	4 KB
main.cpp	2025/6/21 20:11	C++ Source	1 KB
res.qrc	2025/6/22 11:29	QRC 文件	2 KB
SmallRedBook.pro	2025/6/22 16:36	PRO 文件	1 KB
SmallRedBook.pro.user	2025/6/21 20:44	USER 文件	23 KB
widget.cpp	2025/6/22 17:33	C++ Source	3 KB
widget.h	2025/6/22 17:33	C/C++ Header	1 KB
widget.ui	2025/6/22 13:09	UI 文件	4 KB
小红书资源.rar	2025/6/21 20:27	WinRAR 压缩文件	52,803 KB





分两步，先读取，在解析。

读取：增加函数 `QString readContent (const QString& file)`。

实现如下：使用qt提供的文本流QTextStream，进行读取

```
if( file.isEmpty()){
        QMessageBox::critical(this,"ERROR","解析文件路径为空");
        return "";
}
QFile m_jsonfile(file);
if (!m_jsonfile.open(QFile::ReadWrite | QFile::Text)) {
    QMessageBox::critical(this,"ERROR","打开文件失败");
    return "";
}

// 读取文件的全部内容
QTextStream stream(&m_jsonfile);
stream.setCodec("UTF-8");     // 设置读取编码是UTF8
return stream.readAll();   //读取到内容返回
```

增加解析json内容函数：

QVector<Content> **analyJsonContent** (const QString str)

```
if( str.isEmpty()){
    QMessageBox::critical(nullptr,"ERROR","解析文件内容为空");
    return {};
}

// QJsonParseError类用于在JSON解析期间报告错误。
QJsonParseError jsonError;
// 将json解析为UTF-8编码的json文档，并从中创建一个QJsonDocument。如
QJsonDocument doc = QJsonDocument::fromJson(str.toUtf8(), &jsonErro
// 判断是否解析失败
if (jsonError.error != QJsonParseError::NoError && doc.isNull()) {
    //qDebug() << "Json格式错误！" << jsonError.error;
    QMessageBox::critical(nullptr,"ERROR","Json格式错误！");
    return {};
}

/*
{
  "id":0,
  "nickname": "张三",
  "title": "大傻马",
  "like": true,
  "like_num": 1540,
  "detail": {
    "total_pic": 10,
    "text": "我是一个大傻马",
    "topic": "#傻人 #沙雕动画 #搞笑动物",
    "time": "2024/01/12",
    "location": "中国-江苏"
  }
},
*/


//解析的内容放到了 doc中，获取根大括号
```

```
QJsonObject rootObj =doc.object();
QJsonValue arr = rootObj.value("arr");  //数组
QVector<Content> contentVec;
if(arr.type()==QJsonValue::Array){   //数组的大小决定了显示内容的数量
    // 转换为QJsonArray类型
    QJsonArray jsonArr = arr.toArray();

    for (int i = 0; i < jsonArr.size(); i++) {
        QJsonObject content = jsonArr.at(i).toObject();
        int id = content.value("id").toInt();
        QString nickname = content.value("nickname").toString();
        QString title = content.value("title").toString();
        bool like = content.value("like").toBool();
        int like_num = content.value("like_num").toInt();

        QJsonObject detail = content.value("detail").toObject();
        int total_pic = detail.value("total_pic").toInt();
        QString text = detail.value("text").toString();
        QString topic = detail.value("topic").toString();
        QString time = detail.value("time").toString();
        QString location = detail.value("location").toString();
        bool attention = detail.value("attention").toBool();

//          qDebug()<<id;
//          qDebug()<<nickname;
//          qDebug()<<title;
//          qDebug()<<like;
//          qDebug()<<like_num;
//          qDebug()<<total_pic;
//          qDebug()<<text;
//          qDebug()<<topic;
//          qDebug()<<time;
//          qDebug()<<location;

        //注意初始化顺序不要写乱
        Content con{id,nickname,title,like,like_num,{total_pic,text,topic,time,l

        contentVec.push_back(con);
```

```
    }
}

return contentVec;
```

将解析后的content存储起来，增加成员属性 `m_contentVec` 。

构造函数中增加

```
//读取json文件并解析数据
m_contentVec = analyJsonData(   readJsonFile("../SmallRedBook/pic/content.
qDebug()<<"size = "<<m_contentVec.size();
```

输出测试结果如下：