😀

202506-电子信息工程生产实习-day5

1. 关注、发现、附近三个按钮之间的切换

为三个按钮增加对应的槽函数：

```
void Home::slots_showAttention()
void Home::slots_showFind()
void Home::slots_showNear()
```

在Home的构造函数中 手动绑定信号与槽

```
    //动手绑定信号与槽
    if( !connect(ui->pb_attention,&QPushButton::clicked,this,&Home::slots_show
        QMessageBox::critical(this,"ERROR","显示关注按钮信号槽绑定失败");
    }
    if( !connect(ui->pb_find,&QPushButton::clicked,this,&Home::slots_showFind
        QMessageBox::critical(this,"ERROR","显示关注按钮信号槽绑定失败");
    }
    if( !connect(ui->pb_near,&QPushButton::clicked,this,&Home::slots_showNea
        QMessageBox::critical(this,"ERROR","显示关注按钮信号槽绑定失败");
    }
```

- slots_showAttention 槽函数实现：将【关注】按钮加粗，其他的取消加粗。如果连续多次点击同一个按钮，会造成连续浪费的处理，所以需要校验下。

  ```
  void Home::slots_showAttention(){
      //发现如果连续多次点击同一个按钮，会造成连续浪费的处理，所以需要校验
      QFont f = ui->pb_attention->font();
      if( f.bold() ){  //如果关注已经加粗了，说明此时正是关注页面，所以没必要重
          return ;
      }
  ```

```
    setBold(true,false,false);  //关注加粗，其他的取消加粗

    //初始化关注内容
    initAttentionContent();
}
```

其中的函数 initAttentionContent 函数为手动实现：

```
void Home::initAttentionContent(){
    if(m_contentVec.size()<=0){
        QMessageBox::warning(this,"WARNING","暂无内容");
        return ;
    }

    //清空表格tablewidget
    ui->tw_content->clear();

        //放置所有关注的的表格项
    QList<TableItem*> lstAttention;

    //先过滤出来关注的content,，并添加TableItem
    for(Content& content: m_contentVec){
        if(content.detail_info.attention){  //如果是关注的
            TableItem* pitem = new TableItem(content);
            lstAttention.push_back(pitem);
        }
    }

    //计算新的行数
    int attenSize = lstAttention.size();

    //设置行数
    int rowNum = (attenSize+1)/2;
    ui->tw_content->setRowCount(rowNum);

    //设置关注内容和行高
    for(int i=0;i<attenSize;++i){
        ui->tw_content->setCellWidget(i/2,i%2,lstAttention[i]);
```

```
        ui->tw_content->setRowHeight(i/2,250);
    }
}
```

- slots_showFind 槽函数实现：将【发现】按钮加粗，其他的取消加粗。如果连续多次点击同一个按钮，会造成连续浪费的处理，所以需要校验下。

```
void Home::slots_showFind(){
    QFont font = ui->pb_find->font();
    if( font.bold() ){
        return ;
    }

    setBold(false,true,false);

    initFindContent();
}
```

`initFindContent` 函数在之前已经实现过了。

- slots_showNear 槽函数实现：将【附近】按钮加粗，其他的取消加粗。如果连续多次点击同一个按钮，会造成连续浪费的处理，所以需要校验下。

```
void Home::slots_showNear(){
    QFont font = ui->pb_near->font();
    if( font.bold() ){
        return ;
    }
    setBold(false,false,true);
    initNearContent();
}
```

其中 initNearContent() 为手动实现的函数。

1. 存储当前登录用户的地点。在Home类中增加属性

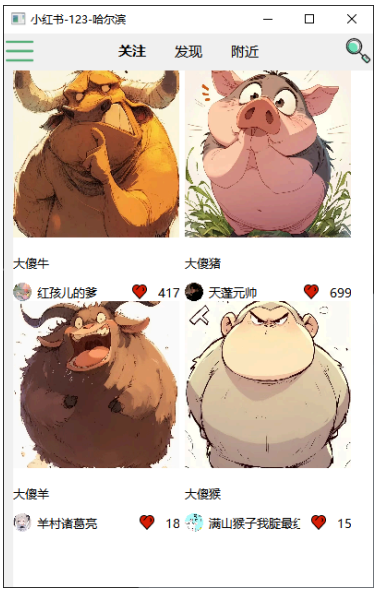
   `QString m_location;  //存储当前的地点`

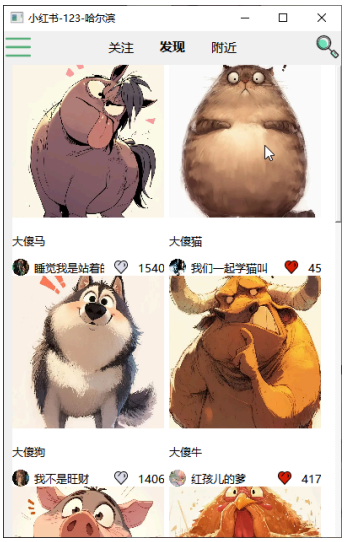
   构造函数中赋值：

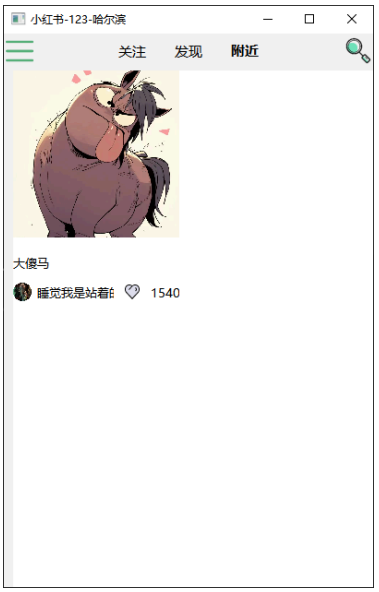
   `m_location = location;`

```
void Home::initNearContent(){
    //1.判空
    if(m_contentVec.size()<=0){
        QMessageBox::warning(this,"WARNING","暂无内容");
        return ;
    }

    //清空表格tablewidget
    ui->tw_content->clear();

    //过滤附近同城的
    QList<TableItem*> lstNearItem;
    for(Content& content: m_contentVec ){
        //判断是否为同一个城市
        if(content.detail_info.location == m_location){
            TableItem* pitem = new TableItem(content);
            lstNearItem.push_back(pitem);
        }
    }

    int itemSize = lstNearItem.size();

    //计算并设置行数
    ui->tw_content->setRowCount((itemSize+1)/2);

    for(int i=0;i<itemSize;++i){
        ui->tw_content->setCellWidget(i/2,i%2,lstNearItem[i]);
        ui->tw_content->setRowHeight(i/2,250);
    }
}
```

三个界面直接切换如图：


小红书-123-哈尔滨
关注
发现
附近
大傻牛
大傻猪
红孩儿的爹
417
天蓬元帅
699
大傻羊
大傻猴
羊村诸葛亮
18
15


小红书-123-哈尔滨
关注
发现
附近
大傻马
大傻猫
1540
我们一起学猫叫
45
大傻狗
大傻牛
我不是旺财
1406
红孩儿的爹
417

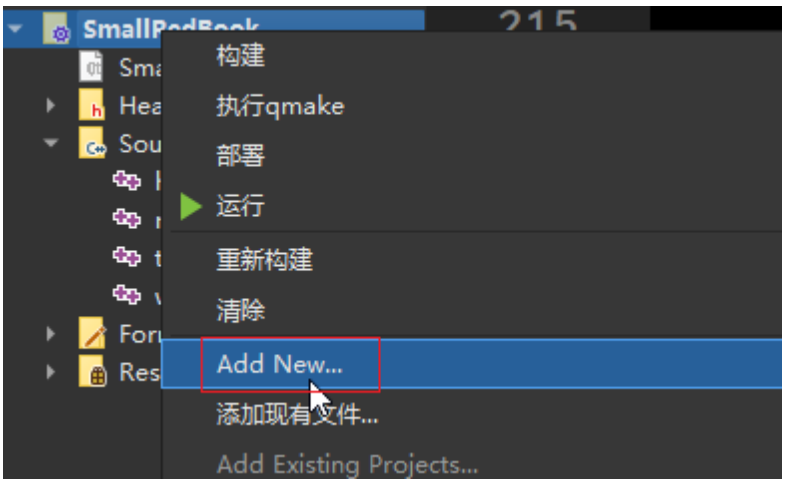

小红书-123-哈尔滨
关注
发现
附近
大傻马
1540

2.详细页

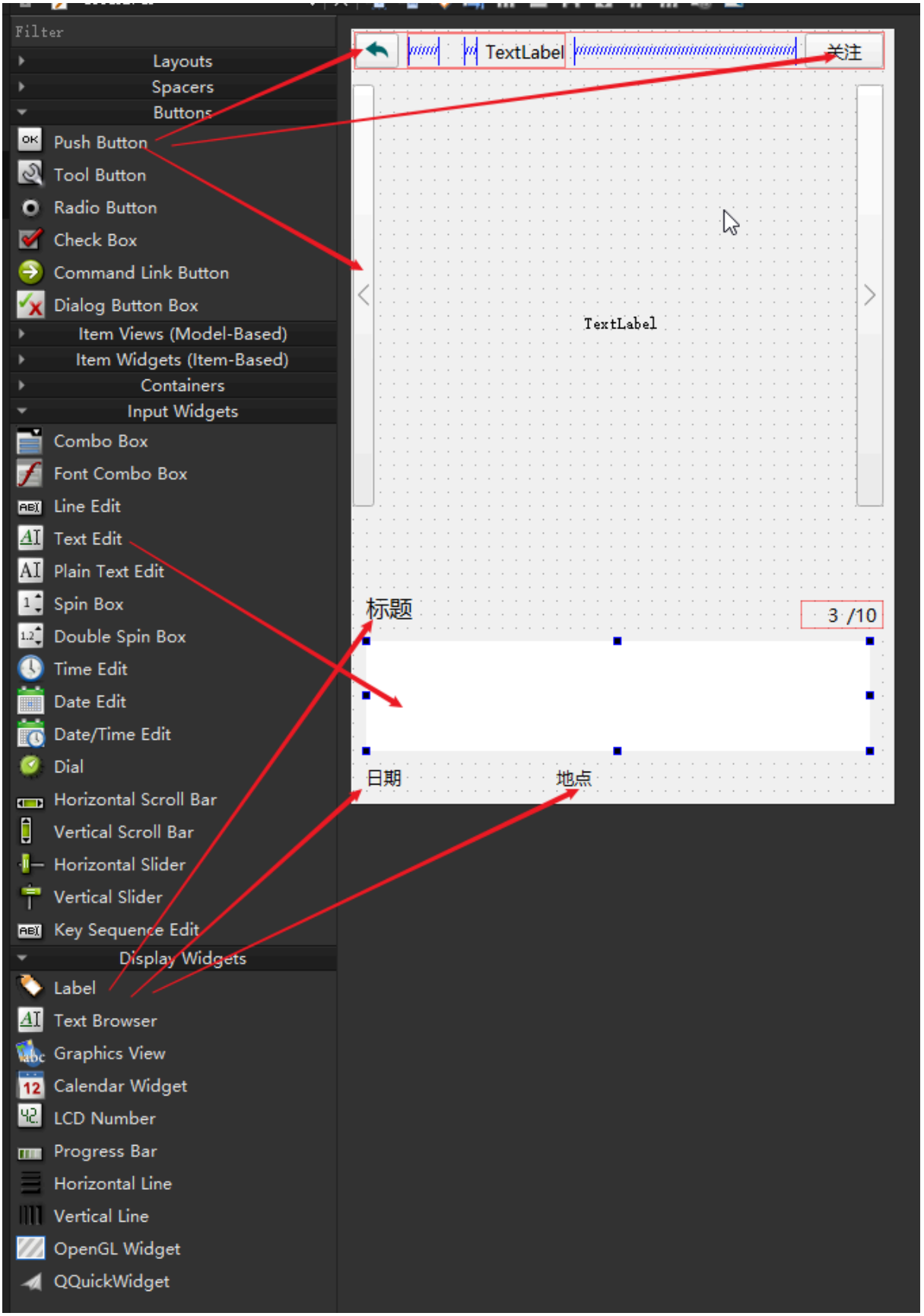
页面展示如图：


睡觉我是站着的
关注
大傻马
1 /10
什么时候我能不被人骑啊？？
#傻马 #沙雕动画 #搞笑动物
2024/01/12
哈尔滨

新建详细窗口


构建
执行qmake
部署
运行
重新构建
清除
Add New...
添加现有文件...
Add Existing Projects...

设计窗口：使用的组件包含 QLabel、QPushButton、QTextEdit


Filter
Layouts
Spacers
Buttons
Push Button
Tool Button
Radio Button
Check Box
Command Link Button
Dialog Button Box
Item Views (Model-Based)
Item Widgets (Item-Based)
Containers
Input Widgets
Combo Box
Font Combo Box
Line Edit
Text Edit
Plain Text Edit
Spin Box
Double Spin Box
Time Edit
Date Edit
Date/Time Edit
Dial
Horizontal Scroll Bar
Vertical Scroll Bar
Horizontal Slider
Vertical Slider
Key Sequence Edit
Display Widgets
Label
Text Browser
Graphics View
Calendar Widget
LCD Number
Progress Bar
Horizontal Line
Vertical Line
OpenGL Widget
QQuickWidget
TextLabel
关注
TextLabel
标题
3 /10
日期
地点

返回按钮，图标大小设置为20,20，按钮大小设置为25,25大小。关注按钮，宽度设置为50，高度自定。微软雅黑，10号字体。

内容图片，参考高度370，宽度自适应。

标题 微软雅黑14号字体，其他文字组件微软雅黑，10号字体。

- 详情页构造函数：

  增加参数 `const Content& content` ，增加代码如下

```
//设置返回按钮的样式
ui->pb_back->setFlat(true);
ui->pb_back->setStyleSheet(BORDER_FRAME1);

//设置关注按钮的样式
ui->pb_attention->setFlat(true);
ui->pb_attention->setStyleSheet(BORDER_FRAME2);

//判断是否关注
if(content.detail_info.attention){
    ui->pb_attention->setText("已关注");
}else{
    ui->pb_attention->setText("关注");
}

//头像的样式和图片
ui->l_profile->setStyleSheet(PROFILE_STYLE(content.id));

//设置用户名
ui->l_user->setText(content.user);

//设置标题
ui->l_title->setText(content.title);

//设置详细信息组件 QTextEdit ， 详细信息和话题之间有换行
```

```
ui→te_detail→setText(content.detail_info.text+"\n"+content.detail_info.top



//设置时间
ui→l_date→setText(content.detail_info.time);
//设置地点
ui→l_location→setText(content.detail_info.location);
```

在主函数中，定义Detail 对象，并显示，效果如图。

```
Detail detail(Content{0,"睡觉我是站着的","大傻马",false,1540,
                {10,"什么时候我才能不被人骑啊？？","#傻马 #沙雕动画","2024/
detail.show();
```

下面针对于组件 QTextEdit te_detail 进行调整，设置为 无边框、背景透明、文字不可编辑。

```
//设置无边框
ui->te_detail->setFrameStyle(QFrame::NoFrame);
//设置背景透明
ui->te_detail->setStyleSheet("background: transparent;");
//设置不可编辑
ui->te_detail->setDisabled(true);
```

效果如图：

接下来添加显示图片，构造函数中

```
//循环遍历  图片总数
for(int i= 0;i<content.detail_info.total_pic;++i){
        QPixmap pix(QString ("../SmallRedBook/pic/content/%1-%2.jpg").arg(cor
     m_picVec.push_back(pix);
}
//初始显示下标为0的图片
m_currentPicIndex = 0;
showDetailPic();
```

增加 QVector<QPixmap> m_picVec ，int m_currentPicIndex 成员属性，分别装配置所有图片和当前正在显示的图片的下标。

showDetailPic() 为新增的函数，用于控制显示图片和图片下标

```
void Detail::showDetailPic(){
   //获取显示图片组件的大小
```

```
    QSize size = ui->l_detailpic->size();
    //按照宽高比显示图片
    ui->l_detailpic->setPixmap(m_picVec[m_currentPicIndex].scaled(size,Qt::Ke
    //设置图片显示的下标
    ui->l_num->setText(QString::number(m_currentPicIndex+1));
}
```

显示效果如图：

接下设置图片总数数字，无法保证所有的帖子图片总数都是10，在构造函数中增加如下代码。

```
ui->l_total->setText(QString::number(content.detail_info.total_pic));
```