

目录

- 一、前言.....2
- 二、开发过程.....2
 - （一）设计思路.....2
 - （二）实现过程.....2
 - （三）系统功能检测6
- 三、设计方案8
 - （一）功能组成.....8
 - （二）界面组成.....9
 - （三）代码实现.....9
- 四、技术讨论20
 - （一）存在的问题.....20
 - （二）解决方案.....20

一、前言

本次项目是实现一个根据股票的涨幅进行报警的一个系统。用户可以通过设置一个阈值来实现股票的报警。通过这个系统,可以提醒用户所关注股票的情况,以便及时作出调整。

二、开发过程

（一）设计思路

1. 首先先进行软件页面的布局,一个登陆页面还有一个主页面,主页面分为两个页面,一个是股票信息的界面,一个是关注股票信息的界面
2. 登录页面功能的实现
3. 从股票网页获取股票信息并对股票信息进行处理
4. 输入股票代码查询该股票的相关信息
5. 用户可以对感兴趣的股票进行关注
6. 当股票的跌涨幅大于一定的阈值时就弹窗报警

（二）实现过程

1. 界面设计如下:

登录

用户名：

密码：

登录

注册

stockAlarm

股票代码：

查看

还原

	代码	名称	最新价	涨跌幅	涨跌额	振幅	最高	最低	市盈率	市净率	加关注
1	301011	N华立	61.50	333.10%	47.30	70.28%	70.00	60.02	132.28	7.72	<input type="checkbox"/> 关注
2	688597	N燧烽	14.38	144.56%	8.50	88.78%	19.54	14.32	264.23	3.25	<input type="checkbox"/> 关注
3	301009	N可靠	25.85	106.14%	13.31	52.47%	32.28	25.70	43.65	4.97	<input type="checkbox"/> 关注
4	300708	聚灿光电	33.53	20.01%	5.59	19.69%	33.53	28.03	163.94	6.85	<input type="checkbox"/> 关注
5	300458	全志科技	70.43	20.00%	11.74	18.57%	70.43	59.53	67.56	9.59	<input type="checkbox"/> 关注
6	688595	芯海科技	91.26	20.00%	15.21	23.04%	91.26	73.74	-780.45	10.68	<input type="checkbox"/> 关注
7	300887	谱尼测试	87.60	20.00%	14.60	18.60%	87.60	74.02	-70.38	6.74	<input type="checkbox"/> 关注
8	300046	台基股份	28.03	19.99%	4.67	19.91%	28.03	23.38	101.50	6.62	<input type="checkbox"/> 关注
9	300793	佳禾智能	17.60	19.97%	2.93	19.09%	17.60	14.80	-76.40	3.89	<input type="checkbox"/> 关注
10	688396	XD华润微	86.93	19.13%	13.96	18.42%	87.56	74.12	71.74	7.40	<input type="checkbox"/> 关注
11	688012	中微公司	153.68	16.47%	21.73	19.36%	158.34	132.80	149.15	18.02	<input type="checkbox"/> 关注

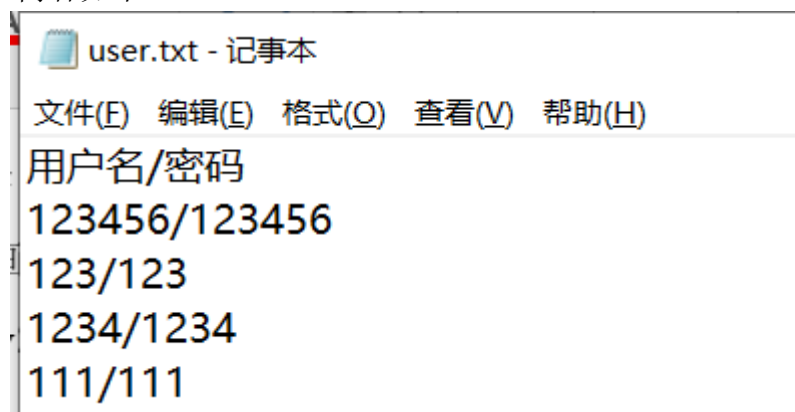
首页

已关注

退出



2. 登录页面有两个功能，一个是用户的登录功能，一个是用户的注册功能，用户的登录信息和注册信息都存在一个文本文件 user.txt 中，其内容如下：

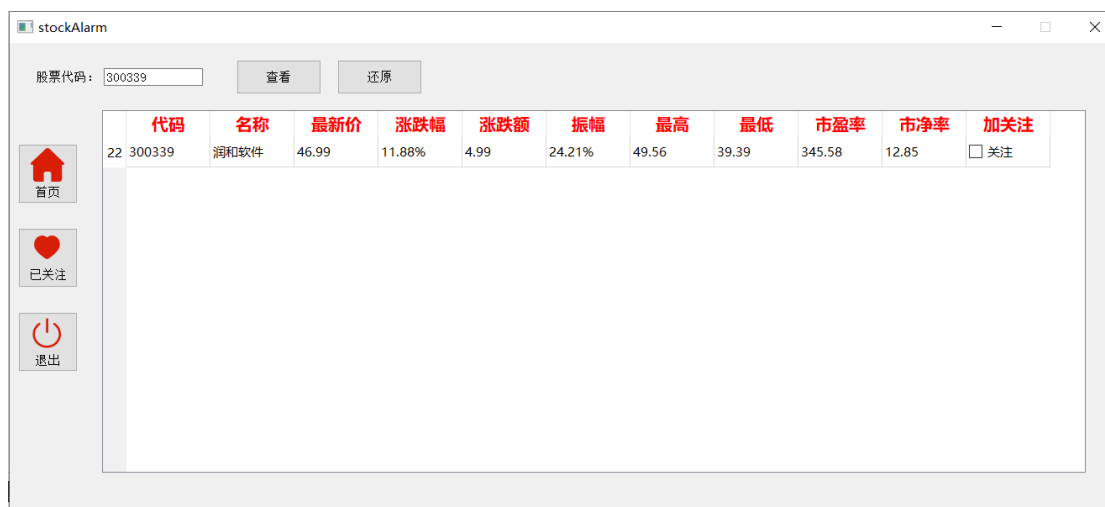


当用户已经注册则可以通过登录进入主窗口界面，没有即进行注册，当注册成功后用户的信息会被记录到 user.txt 文件中。当用户登录时会将用户的用户名写入文件 temp.txt 文件中，以便于找到该用户关注的股票代码并显示。

3. 从东方财富网的行情中心获取深沪 A 股板块的股票信息并对数据进行处理并显示在 QTableWidget 上。该功能的实现用到了 Qt 里面的两个类功能模块。首先用到了 Qt 网络编程模块的类有 QNetworkAccessManager, QNetworkRequest, QNetworkReply。其中，QNetworkRequest 类用于表示网络请求，它也充当一个与请求信息相关的通用容器。QNetworkAccessManager 类用于协调网络操作，一旦创建一个 request，该类用来发出请求并发射信号来报告它的进度。QNetworkReply 类用于网络请求的响应，当一个 request 被发出后，QNetworkAccessManager 就会创建相应的响应，通过 QNetworkReply 提供的信号可以单独监听每个响应，或者使用 manager 的信号。接下用到了 Json 数据处理的模块的类有 QJsonParseError, QJsonDocument, QJsonObject, QJsonArray, QJsonValue。QJsonParseError 类用于在 JSON 解析中报告错误。QJsonDocument 类用于读和写 JSON 文档。QJsonObject 类封装了一个 JSON 对象。QJsonArray 类封装了一个

JSON 数组。QJsonValue 类封装了一个值。

- 根据股票代码查询股票信息的功能是通过隐藏 QTableWidgetItem 的不是该股票代码的其他股票代码的信息，通过还原按钮将其他股票信息还原。查询结果显示如下：



- 用户对股票进行关注，并将关注的股票代码保存在文件 stock.txt 中，以便下次用户登录在关注页面显示关注股票的相关信息

stock.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

用户名/关注的股票代码

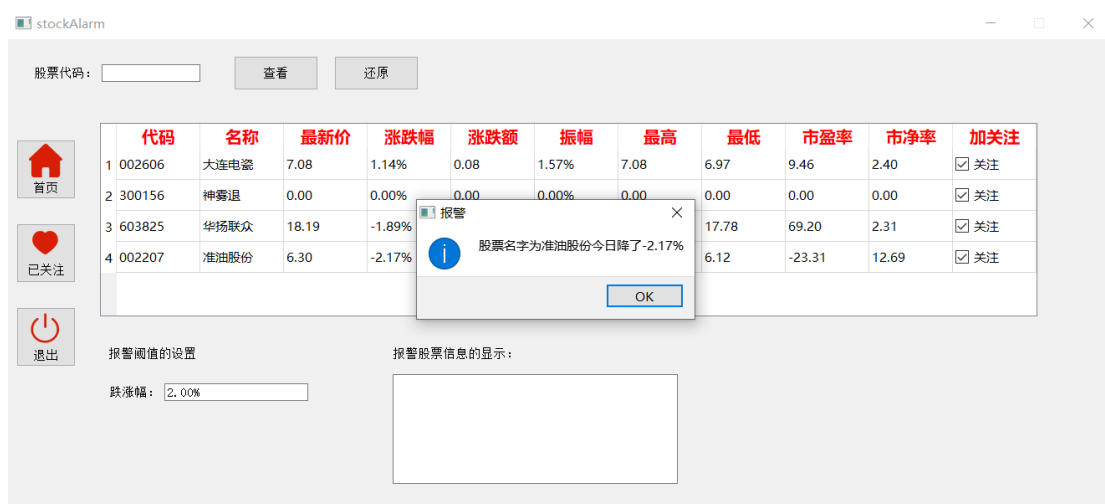
123456/300012/300016/301007/688067/688681

123/002207/603825/002606/300156

1234/688067/688681/300287

111/688597/300708/601567

- 股票阈值报警的设计，当关注的股票跌涨幅超过一定阈值即进行报警，报警以弹窗显示，用户可以通过设置阈值进行报警，查探股票跌涨幅的情况，报警弹窗如下：



报警股票信息的显示：

股票名字为准油股份今日降了-2.17%

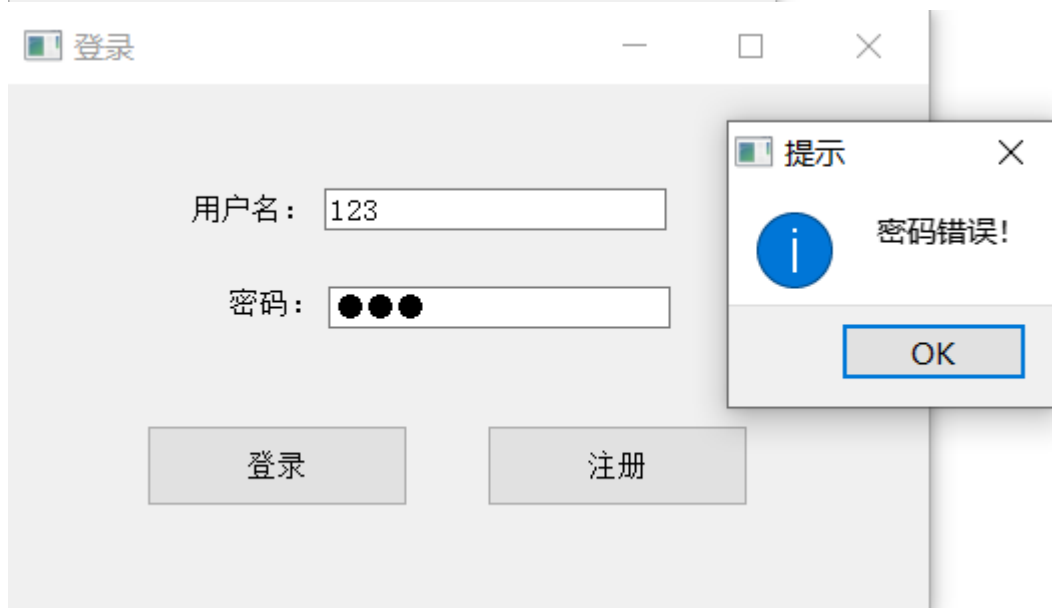
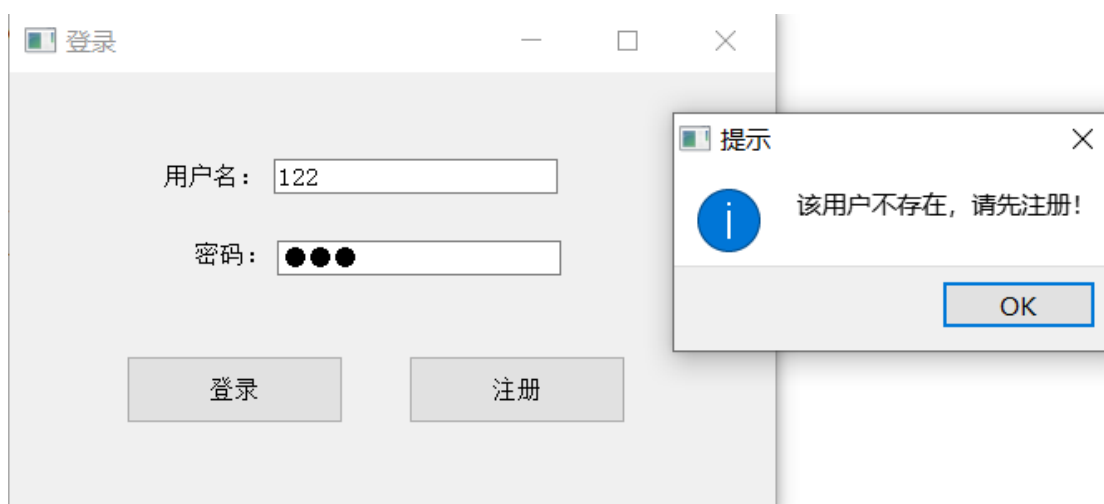
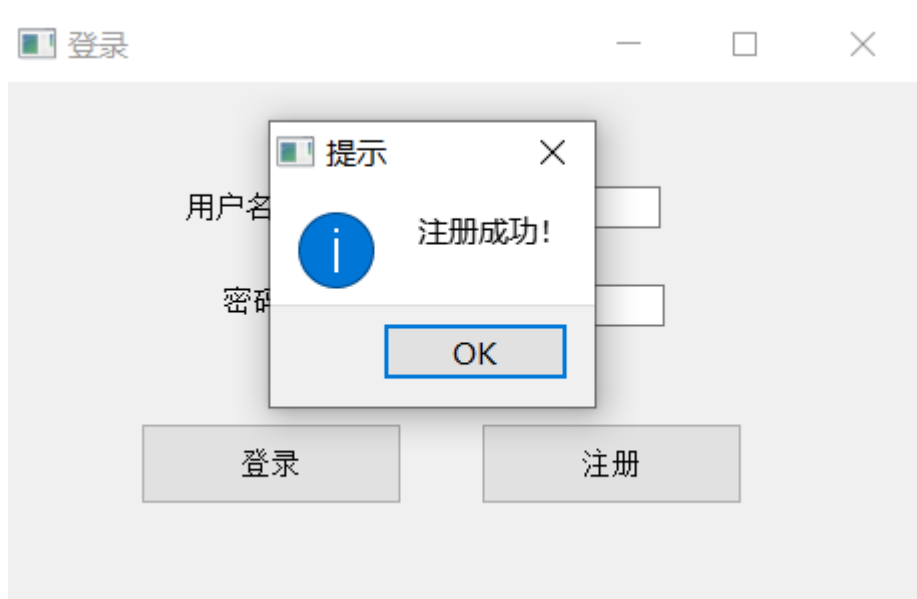
（三）系统功能检测

1. 登录窗口的检测

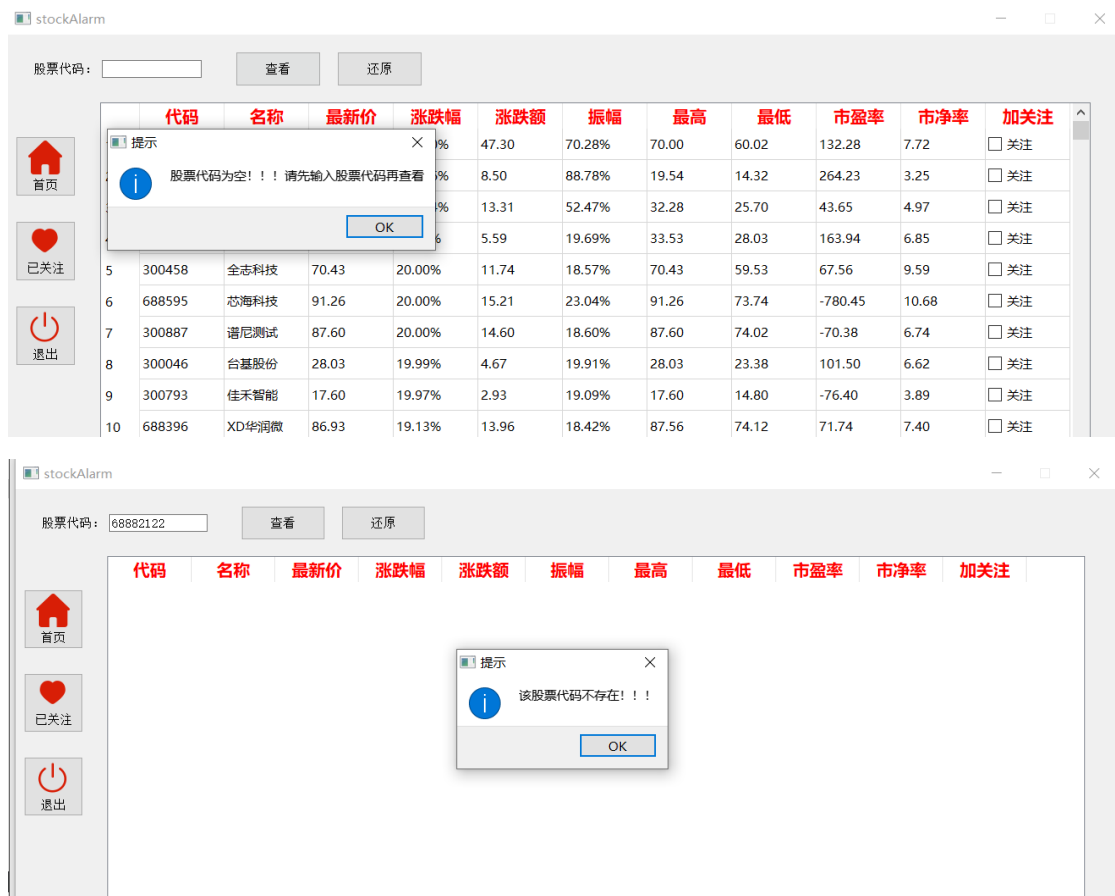
The image displays two screenshots of a login window titled "登录" (Login). The window contains two input fields: "用户名:" (Username) and "密码:" (Password), each followed by a text box. Below the fields are two buttons: "登录" (Login) and "注册" (Register).

Top Screenshot: The "提示" (Prompt) dialog box shows the message "用户名或密码为空!" (Username or password is empty!).

Bottom Screenshot: The "提示" (Prompt) dialog box shows the message "该用户已经注册!" (This user is already registered!).



2. 主页面功能的检测



三、设计方案

(一) 功能组成

1. 用户登录
2. 用户注册
3. 获取股票信息及数据处理
4. 股票代码查询
5. 还原查询前的股票信息
6. 股票的关注
7. 股票报警
8. 退出

(二) 界面组成

1. 登录界面



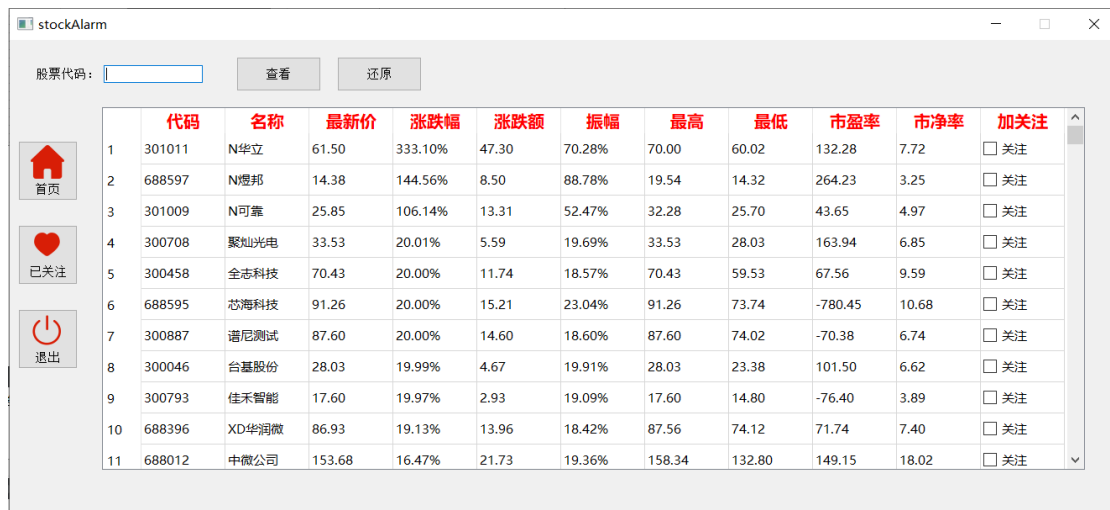
登录

用户名:

密码:

登录 注册

2. 主界面



	代码	名称	最新价	涨跌幅	涨跌额	振幅	最高	最低	市盈率	市净率	加关注
1	301011	N华立	61.50	333.10%	47.30	70.28%	70.00	60.02	132.28	7.72	<input type="checkbox"/> 关注
2	688597	N耀邦	14.38	144.56%	8.50	88.78%	19.54	14.32	264.23	3.25	<input type="checkbox"/> 关注
3	301009	N可靠	25.85	106.14%	13.31	52.47%	32.28	25.70	43.65	4.97	<input type="checkbox"/> 关注
4	300708	聚灿光电	33.53	20.01%	5.59	19.69%	33.53	28.03	163.94	6.85	<input type="checkbox"/> 关注
5	300458	全志科技	70.43	20.00%	11.74	18.57%	70.43	59.53	67.56	9.59	<input type="checkbox"/> 关注
6	688595	芯海科技	91.26	20.00%	15.21	23.04%	91.26	73.74	-780.45	10.68	<input type="checkbox"/> 关注
7	300887	诺尼测试	87.60	20.00%	14.60	18.60%	87.60	74.02	-70.38	6.74	<input type="checkbox"/> 关注
8	300046	台基股份	28.03	19.99%	4.67	19.91%	28.03	23.38	101.50	6.62	<input type="checkbox"/> 关注
9	300793	佳禾智能	17.60	19.97%	2.93	19.09%	17.60	14.80	-76.40	3.89	<input type="checkbox"/> 关注
10	688396	XD华润微	86.93	19.13%	13.96	18.42%	87.56	74.12	71.74	7.40	<input type="checkbox"/> 关注
11	688012	中微公司	153.68	16.47%	21.73	19.36%	158.34	132.80	149.15	18.02	<input type="checkbox"/> 关注

(三) 代码实现

1. 用户登录

```
1. //登录按钮
2. connect(ui->pB_login,&QPushButton::clicked,this,[=]() {
3.     QString user = ui->lE_user->text(); //获取用户名
4.     QString pas = ui->lE_pas->text();   //获取密码
5.     if (user==" " || pas == " ")
6.     {
7.         QMessageBox::information(this, "提示", "用户名或密码为空!");
8.         ui->lE_user->clear();
9.         ui->lE_pas->clear();
10.    return;
```

```

11.     }
12.     QString str = "user.txt";
13.     QFile file1(str);
14.     file1.open(QIODevice::ReadOnly);
15.     QTextStream in(&file1);
16.     in.readLine();    //将首行去掉
17.     while(!in.atEnd())
18.     {
19.         QString line = in.readLine();
20.         QStringList list = line.split("/");
21.         if (list[0] == user)
22.         {
23.             if (list[1] == pas)
24.             {
25.                 file1.close();
26.                 QFile file2("temp.txt");
27.                 file2.open(QIODevice::WriteOnly | QIODevice::Text);
28.                 QTextStream out(&file2);
29.                 out << user;
30.                 file2.close();
31.                 stockAlarm *w1 = new stockAlarm();
32.                 w1 -> show();
33.                 this -> close();
34.                 return;
35.             }
36.             else
37.             {
38.                 file1.close();
39.                 QMessageBox::information(this, "提示", "密码错误! ");
40.                 ui->lE_user->clear();
41.                 ui->lE_pas->clear();
42.                 return;
43.             }
44.         }
45.     }
46.     QMessageBox::information(this, "提示", "该用户不存在, 请先注册! ");
47.     file1.close();
48. });

```

2. 用户注册

```

1. //注册按钮
2. connect(ui->pB_zhuce, &QPushButton::clicked, this, [=]() {
3.     QString user = ui->lE_user->text(); //获取用户名
4.     QString pas = ui->lE_pas->text();   //获取密码

```

```

5.     if (user==" " || pas == " ")
6.     {
7.         QMessageBox::information(this, "提示", "用户名或密码为空!");
8.         ui->lE_user->clear();
9.         ui->lE_pas->clear();
10.        return;
11.    }
12.    QString str = "user.txt";
13.    QFile file1(str);
14.    file1.open(QIODevice::ReadOnly);
15.    QTextStream in(&file1);
16.    in.readLine();    //将首行去掉
17.    while(!in.atEnd())
18.    {
19.        QString line = in.readLine();
20.        QStringList list = line.split("/");
21.        if (list[0] == user)
22.        {
23.            QMessageBox::information(this, "提示", "该用户已经注册!");
24.            file1.close();
25.            ui->lE_user->clear();
26.            ui->lE_pas->clear();
27.            return;
28.        }
29.    }
30.    file1.close();
31.
32.    QFile file2(str);
33.    file2.open(QIODevice::WriteOnly | QIODevice::Append);
34.    QTextStream out(&file2);
35.    out << user << "/" << pas << "\n";
36.    file2.close();
37.    ui->lE_user->clear();
38.    ui->lE_pas->clear();
39.    QMessageBox::information(this, "提示", "注册成功!");
40. });

```

3. 获取股票信息及数据处理

```

1. //从东方财富网获取深沪个股科创版的股票信息
2. manger = new QNetworkAccessManager();
3. QString url = "http://32.push2.eastmoney.com/api/qt/clist/get?cb=jQuery11240
9803814194851075_1623642572055&pn=1&pz=5000&po=1&np=1&ut=bd1d9ddb04089700cf9
c27f6f7426281&fltt=2&invt=2&fid=f3&fs=m:0+t:6,m:0+t:80,m:1+t:2,m:1+t:23&fiel

```

```

        ds=f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f12,f13,f14,f15,f16,f17,f18,f20,f21,f23,f2
        4,f25,f22,f11,f62,f128,f136,f115,f152&_=1623642572056";
4.  manger->get(QNetworkRequest(QUrl(url)));
5.  connect(manger,&QNetworkAccessManager::finished,this,&stockAlarm::dealData);

1.  //将获取的股票信息进行处理
2.  void stockAlarm::dealData(QNetworkReply *reply)//获取网页数据并将数据显示在表
    格
3.  {
4.      QByteArray data1 = reply->readAll();
5.      QString contentType = reply->header(QNetworkRequest::ContentTypeHeader).
        toString();
6.      //QDebug() << "contentType - " << contentType;
7.      int charsetIndex = contentType.indexOf("charset=");
8.      if(charsetIndex > 0)
9.      {
10.         charsetIndex += 8;
11.         QString charset = contentType.mid(charsetIndex).trimmed().toLowerCase();
12.         if(charset.startsWith("gbk") || charset.startsWith("gb2312"))
13.         {
14.             QTextCodec *codec = QTextCodec::codecForName("GBK");
15.             if(codec)
16.             {
17.                 data1 = codec->toUnicode(data1).toUtf8();
18.             }
19.         }
20.     }
21.     int parenthesisLeft = data1.indexOf('(');
22.     int parenthesisRight = data1.lastIndexOf(')');
23.     if(parenthesisLeft >=0 && parenthesisRight >=0)
24.     {
25.         parenthesisLeft++;
26.         data1 = data1.mid(parenthesisLeft, parenthesisRight - parenthesisLeft);
27.     }
28.     QJsonParseError jsonError;
29.     QJsonDocument doucment = QJsonDocument::fromJson(data1, &jsonError);
        // 转化为 JSON 文档
30.     if (!doucment.isNull() && (jsonError.error == QJsonParseError::NoError))
        { // 解析未发生错误
31.         if (doucment.isObject()) { // JSON 文档为对象
32.             QJsonObject object = doucment.object(); // 转化为对象

```

```

33.         if (object.contains("data")) { // 包含指定的 key
34.             QJsonValue value1 = object.value("data"); // 获取指
                定 key 对应的 value
35.             if (value1.isObject()) { // 判断 value1 是否为对象
36.                 QJsonObject object1 = value1.toObject(); // 转化为对象
37.                 if (object1.contains("diff")) {
38.                     QJsonValue value2 = object1.value("diff");
39.                     if (value2.isArray()) {
40.                         QJsonArray array1 = value2.toArray();
41.                         int nSize = array1.size();
42.                         double yuZhi = ui->lineEdit_2->text().remove(-
                            1, 1).toDouble();
43.                         for (int i = 0; i < nSize; ++i) {
44.                             QJsonValue value3 = array1.at(i);
45.                             if (value3.isObject()) {
46.                                 QJsonObject object2 = value3.toObject();
                                    // 转化为对象
47.                                 QString f[10];
48.
49.                                 f[0] = object2.value("f12").toString();
                                    //股票代码
50.                                 f[1] = object2.value("f14").toString();
                                    //股票名称
51.                                 f[2] = QString::number(object2.value("f2
                                    ").toDouble(), '.', 2); //最新价
52.                                 f[3] = QString::number(object2.value("f3
                                    ").toDouble(), '.', 2) + "%"; //涨跌幅
53.                                 double price = object2.value("f3").toDou
                                    ble();
54.                                 f[4] = QString::number(object2.value("f4
                                    ").toDouble(), '.', 2); //涨跌额
55.                                 f[5] = QString::number(object2.value("f7
                                    ").toDouble(), '.', 2) + "%"; //振幅
56.                                 f[6] = QString::number(object2.value("f1
                                    5").toDouble(), '.', 2); //最高价
57.                                 f[7] = QString::number(object2.value("f1
                                    6").toDouble(), '.', 2); //最低价
58.                                 f[8] = QString::number(object2.value("f9
                                    ").toDouble(), '.', 2); //市盈率
59.                                 f[9] = QString::number(object2.value("f2
                                    3").toDouble(), '.', 2); //市净率
60.                                 // 将数据放入 tableWidget_1
61.                                 ui->tableWidget_1->insertRow(i); //在表格
                                    尾部添加一行

```

```

62.         for(int j = 0; j < 10; j++){
63.             ui->tableWidget_1->setItem(i, j, new
                QTableWidgetItem(f[j]));
64.         }
65.         if (!stockList.isEmpty())
66.         {
67.             if(stockList.contains(f[0]))
68.             {
69.                 // 将已关注的股票数据放入
                tableWidget_2
70.                 int rowCou = ui->tableWidget_2->
                    rowCount();
71.                 ui->tableWidget_2->insertRow(row
                    Cou);//在表格尾部添加一行
72.                 for(int j = 0; j < 10; j++){
73.                     ui->tableWidget_2->setItem(r
                        owCou, j, new QTableWidgetItem(f[j]));
74.                 }
75.                 QTableWidgetItem *item1 = new QT
                    ableWidgetItem("关注");
76.                 item1->setCheckState(Qt::Checked
                    );
77.                 ui->tableWidget_2->setItem(rowCo
                    u, 10, item1);
78.                 QTableWidgetItem *item2 = new QT
                    ableWidgetItem("关注");
79.                 item2->setCheckState(Qt::Checked
                    );
80.                 ui->tableWidget_1->setItem(i, 10
                    , item2);
81.                 if ( price > 0)
82.                 {
83.                     if (price > yuZhi)
84.                     {
85.                         QMessageBox::information
                            (this, "报警", "股票名字为"+f[1]+"今日涨了"+f[3]);
86.                     }
87.                 }
88.                 else
89.                 {
90.                     price = -price;
91.                     if ( price > yuZhi)
92.                     {

```

```

93.                                     QMessageBox::information
    (this, "报警", "股票名字为"+f[1]+"今日降了"+f[3]);
94.                                     }
95.                                     }
96.                                     }
97.
98.                                     else
99.                                     {
100.                                     QTableWidgetItem *item = new QT
    ableWidgetItem("关注");
101.                                     item->setCheckState(Qt::Uncheck
    ed);
102.                                     ui->tableWidget_1->setItem(i, 1
    0, item);
103.                                     }
104.                                     }
105.                                     else
106.                                     {
107.                                     QTableWidgetItem *item = new QTable
    WidgetItem("关注");
108.                                     item->setCheckState(Qt::Unchecked);
109.                                     ui->tableWidget_1->setItem(i, 10, i
    tem);
110.                                     }
111.                                     }
112.                                     }
113.                                     }
114.                                     }
115.                                     }
116.                                     }
117.                                     }
118.                                     }
119.         reply->deleteLater();
120. }

```

4. 股票代码查询

```

1. // 查询股票代码并显示该股票信息，没有相关股票代码则弹窗提醒
2. connect(ui->pB_search,&QPushButton::clicked,this,[=]() {
3.     QString textNum = ui->lineEdit->text();
4.     if(textNum == "")
5.         QMessageBox::information(this,"提示","股票代码为空!!! 请先输入股票代码
    再查看");
6.

```

```

7.     if (ui->stackedWidget->currentIndex() == 0)
8.     {
9.         //获取符合条件的索引
10.        QList <QTableWidgetItem *> item = ui->tableWidget_1->findItems(textN
            um, Qt::MatchContains);
11.
12.        for (int j = 0; j < ui->tableWidget_1->rowCount(); j++) {
13.            ui->tableWidget_1->setRowHidden(j, true);           //隐藏所有行
14.        }
15.        if (!item.isEmpty()) { //不为空
16.            for (int j = 0; j < item.count(); j++) {
17.                ui->tableWidget_1->setRowHidden(item.at(j)->row(), false);
                //输出查看的代码股票的相关内容
18.            }
19.        }
20.        else{
21.            QMessageBox::information(this, "提示", "该股票代码不存在!!!");
22.        }
23.    }
24.    else
25.    {
26.        //获取符合条件的索引
27.        QList <QTableWidgetItem *> item = ui->tableWidget_2->findItems(textN
            um, Qt::MatchContains);
28.
29.        for (int j = 0; j < ui->tableWidget_2->rowCount(); j++) {
30.            ui->tableWidget_2->setRowHidden(j, true);           //隐藏所有行
31.        }
32.        if (!item.isEmpty()) { //不为空
33.            for (int j = 0; j < item.count(); j++) {
34.                ui->tableWidget_2->setRowHidden(item.at(j)->row(), false);
                //输出查看的代码股票的相关内容
35.            }
36.        }
37.        else{
38.            QMessageBox::information(this, "提示", "该股票代码不在已关注列表
                中!!!");
39.        }
40.    }
41.
42. });

```

5. 还原查询前的股票信息

```

1. // 恢复表格数据内容

```



```

2. connect(ui->pB_huifu,&QPushButton::clicked,this,[=](){
3.     ui->lineEdit->clear();
4.     if (ui->stackedWidget->currentIndex() == 0)
5.     {
6.         for (int j = 0; j < ui->tableWidget_1->rowCount(); j++) {
7.             ui->tableWidget_1->setRowHidden(j, false);    //恢复所有行
8.         }
9.     }
10.    else
11.    {
12.        for (int j = 0; j < ui->tableWidget_2->rowCount(); j++) {
13.            ui->tableWidget_2->setRowHidden(j, false);    //恢复所有行
14.        }
15.    }
16. });

```

6. 股票的关注

```

1. // 勾选关注即加入已关注
2. connect(ui->tableWidget_1,&QTableWidget::cellClicked,this,[=](int row){
3.     if(ui->tableWidget_1->item(row, 10)->checkState() == Qt::Checked)
4.     {
5.         int curRow = ui->tableWidget_2->rowCount();//当前行号
6.         ui->tableWidget_2->insertRow(curRow);//在表格尾部添加一行
7.         for(int j = 0; j < 10; j++){
8.             ui->tableWidget_2->setItem(curRow, j, new QTableWidgetItem(ui->tableWidget_1->item(row, j)->text()));
9.         }
10.        QTableWidgetItem *item = new QTableWidgetItem("关注");
11.        item->setCheckState(Qt::Checked);
12.        ui->tableWidget_2->setItem(curRow, 10, item);
13.        stockList.append(ui->tableWidget_1->item(row, 0)->text());
14.    }
15.    else
16.    {
17.        QString text1 = ui->tableWidget_1->item(row, 0)->text();
18.        for(int j = 0; j < ui->tableWidget_2->rowCount(); j++){
19.            if(text1 == ui->tableWidget_2->item(j, 0)->text()){
20.                ui->tableWidget_2->removeRow(j);
21.                stockList.removeAll(text1);
22.                break;
23.            }
24.        }
25.    }
26. });

```

```

27.
28. // 当将关注列表的股票的关注复选框取消时则从关注列表中删除
29. connect(ui->tableWidget_2,&QTableWidget::cellClicked,this,[=](int row){
30.     if(ui->tableWidget_2->item(row, 10)->checkState() == Qt::Unchecked)
31.     {
32.         QString text1 = ui->tableWidget_2->item(row, 0)->text();
33.         for(int j = 0; j < ui->tableWidget_1->rowCount(); j++){
34.             if(text1 == ui->tableWidget_1->item(j, 0)->text()){
35.                 ui->tableWidget_1->item(j, 10)->setCheckState(Qt::Unchecked)
36.                 ;
37.                 stockList.removeAll(text1);
38.                 break;
39.             }
40.         }
41.         ui->tableWidget_2->removeRow(row);
42.     });

```

7. 股票报警

```

1. //报警阈值设置
2. connect(ui->lineEdit_2,&QLineEdit::returnPressed,this,[=]() {
3.     ui->textBrowser->clear();
4.     int count = ui->tableWidget_2->rowCount();
5.     double yuZhi = ui->lineEdit_2->text().remove(-1, 1).toDouble();
6.     for(int i = 0; i < count; i++)
7.     {
8.         double price = ui->tableWidget_2->item(i, 3)->text().remove(-
9.             1, 1).toDouble();
10.        QString name = ui->tableWidget_2->item(i, 1)->text();
11.        QString temp = ui->tableWidget_2->item(i, 3)->text();
12.        if ( price > 0)
13.        {
14.            if (price > yuZhi)
15.                QMessageBox::information(this, "报警", "股票名字为"+name+"今日
16.                    涨了"+temp);
17.            ui->textBrowser->append("股票名字为"+name+"今日涨了
18.                "+temp+"\n");
19.        }
20.        else
21.        {
22.            price = -price;
23.            if ( price > yuZhi)

```

```

23.         {
24.             QMessageBox::information(this, "报警", "股票名字为"+name+"今日
           降了"+temp);
25.             ui->textBrowser->append("股票名字为"+name+"今日降了
           "+temp+"\n");
26.         }
27.     }
28. }
29. });

```

8. 退出, 对 closeEvent 方法进行重载, 使用户关注的股票代码写入文件

```

1. //退出应用
2. connect(ui->tB_exit,&QPushButton::clicked,this,&stockAlarm::close);

1. void stockAlarm::closeEvent(QCloseEvent *) //重载退出函数
2. {
3.     bool flag = false;
4.     QFile file3("stock.txt");
5.     file3.open(QIODevice::ReadOnly);
6.     QTextStream in(&file3);
7.     QStringList tempList;
8.     in.readLine(); //去掉首行
9.     while(!in.atEnd())
10.    {
11.        QString line = in.readLine();
12.        QStringList list = line.split("/");
13.        if (list[0] == user)
14.        {
15.            QString templine = user;
16.            for (int i = 0; i<stockList.length(); i++)
17.            {
18.                templine = templine + "/" + stockList[i];
19.            }
20.            tempList.append(templine);
21.            flag = true;
22.        }
23.        else
24.        {
25.            tempList.append(line);
26.        }
27.    }
28.    file3.close();
29.    if (flag == false)
30.    {

```

```

31.         QString templine = user;
32.         for (int i = 0; i<stockList.length(); i++)
33.         {
34.             templine = templine + "/" + stockList[i];
35.         }
36.         tempList.append(templine);
37.     }
38.     QString str = "用户名/关注的股票代码";
39.     QFile file4("stock.txt");
40.     file4.open(QIODevice::WriteOnly);
41.     QTextStream out(&file4);
42.     out << str << "\n";
43.     for(int i=0; i<tempList.length(); i++)
44.     {
45.         out << tempList[i] << "\n";
46.     }
47.     file4.close();
48.     qApp->quit();
49. }

```

四、技术讨论

（一）存在的问题

报警阈值的初始值是在程序开始前就设置好的，用户登录进去是用系统已经初始化好的阈值进行报警，不能修改系统的初始化阈值。用户只能在软件登录进入后在阈值修改的位置进行修改，修改后的值在下次登录时并不会出现这个阈值而是那个初始化值

（二）解决方案

可以用一个文件来记录每个用户在退出系统前更改的那个阈值，下次登录将那个阈值设置为初始阈值。