山东大学

硕士学位论文

基于.net平台的股票分析系统的设计和实现

姓名：克远

申请学位级别：硕士

专业：软件工程

指导教师：赵合计

20090410

摘　要

当今世界经济发达国家的股票市场总市值几乎相当于国家的GDP值，甚至超过国家的GDP值，股票市场发展十分迅猛。在我国，短短的十几年里，股票市场融资量大幅增长，股市总市值已达到五万亿元，成为国民经济的重要推动力量[1]。如今，越来越多的人选择了炒股这种投资理财方式，然而，由于对国内股票市场没有清晰的认识、缺乏股票数据分析相关知识等一系列原因，很多人买卖股票是相当盲目的，甚至是仅凭感觉操作，从而造成了很大的损失。出于以上现象考虑，本次将开发一个股票分析系统。

通过查阅股票分析方面的相关知识，我们了解到股票的基本指标有多空指数(BBI)、意愿指标(BR)、动向指数(DMI)、随机指数(KDJ)、指数平滑移动平均线(MACD)、威廉变异离散量(WVAD)、价格推动量(Power)、相对强弱指数(RSI)、人气指标(AR)、乖离率(BIAS)、顺势指标(CCI)、指数平均数(EXPMA)、K线图(K)、动量指标(MTM)、能量潮(OBV)、心理线(PSY)、停损点转向(SAR)、容量比率(VR)、均价线(AVL)、布林线(BOLL)、CR指标(CR)、换手率线(HSL)、移动平均线(MA)、成交量柱体及其均线(MV)、震荡量指标(OSC)、变动率指标(ROC)、宝塔线(Tower)、威廉指数(WMS)等[2]。其中比较简单、直观，同时也是比较重要的指标是K线图和成交量柱状图。

首先，本文在对股票基本数据、图表、曲线图研究了解的基础上分析了系统的功能性需求和非功能性需求，并对系统需求以用例图的形式来详细说明。

在需求分析基础上，本文进行了股票分析系统架构设计。先根据系统需求提出系统设计目标和原则，然后分别对系统的技术架构和功能架构进行了设计。在功能架构设计中，着重对股票数据管理功能组成进行了设计。

接着进一步进行股票分析系统的详细设计。该部分对系统的各主要模块进行建模，主要使用uml建模语言。另外设计了系统数据库，并根据数据库设计分析建表。

然后，在详细设计的基础上，对各个模块的实现进行了简单介绍，给出了系统的整体效果图和各个部分的实现。然后结合系统截图和代码具体阐述了各模块

实现细节。在图形显示模块部分，首先讨论了GDI+技术。微软公司的Windows GDI+是Windows XP操作系统或Windows Server 2003操作系统的一部分，它可以生成二维空间向量图形、图像[3]。GDI+技术是GDI（图形设备接口）的升级版本，它主要添加了一些新的功能，也优化了一些现有功能。接着在此基础上，实现了绘制股票分析系统曲线图的模块。

最后，本文对股票分析系统的应用情况作了简单介绍，并对系统进一步改进提出了建议。

关键字：股票分析；GDI+；XML；序列/反序列

ABSTRACT

The market capitalizations of developed countries nearly equal their GDP, or even more than that. Stock market in those countries is developing rapidly. In China, the financing of stock market grows a lot during the past 10 years. Stock has become an important part in national economy. Now, an increasing number of people chose stock as an investment and financial management. However, due to having few knowledge about stock technology, they buy or sell stock very blindly. In order to give them some instruction, we develop this Stock Analysis System.

By the means of referring to relevant knowledge, we know that the basic indicator includes BBI, BR, DMI, KDJ, MACD, WVAD, Power, RSI, AR, BIAS, CCI, EXPMA, K, MTM, OBV, PSY, SAR, VR, AVL, BOLL, CR, HSL, MA, MV, OSC, ROC, Tower, and WMS. Among these basic indicators, K and MV are simple and important.

Firstly, this paper bases on the analysis of stock data, chart. Then, it analyses the functional and non-functional requirements, and describes particularly the system requirement by the use case diagram.

According to the requirement analyzing, this paper gives the system architecture design . Based on the system requirements , this paper puts up the system design goals and principles, and then separately discusses the technology and functional structures .

Following the architecture design, this paper particularly designs this Stock Analysis System. This paper describes every module's design mainly using UML. Besides, in this part, we design system database. We also create table for this system.

Then, based on the detailed design, this paper introduces each part's implementation, and puts forward the system result pictures. This part emphasizes the GDI+ technology. In this part, we discuss GDI+ technology. Microsoft Windows GDI+ is the portion of the Windows XP operating system or Windows Server 2003 operating system that provides two-dimensional vector graphics, imaging, and typography. GDI+ improves on Windows Graphics Device Interface (GDI) (the graphics device interface included with earlier

versions of Windows) by adding new features and by optimizing existing features. Take this as fundamental, we realize drawing analysis chart for the module.

In the end, this paper introduces the application of the Stock Analysis System, and proposes advices for further improvement.

Keyword: Stock Analysis; GDI+; XML; Serialize/Deserialize

原 创 性 声 明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律责任由本人承担。

论文作者签名：[illegible]　日　期：09.4.10

关于学位论文使用授权的声明

本人完全了解山东大学有关保留、使用学位论文的规定，同意学校保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权山东大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

(保密论文在解密后应遵守此规定)

论文作者签名：[illegible]　导师签名：[illegible]　日　期：4.20

第1章 绪 论

1.1 股票分析系统开发背景

当今世界经济发达国家的股票市场总市值几乎相当于国家的GDP值，甚至超过国家的GDP值，股票市场发展十分迅猛。在我国，短短的十几年里，股票市场融资量大幅增长，股市总市值已达到五万亿元，成为国民经济的重要推动力量。如今，越来越多的人选择了炒股这种投资理财方式，然而有相当大一部人并没用通过股票这种投资方式获益，反而造成了很大的损失，之所以会出现这种局面，主要是基于以下几个因素。

1. 在理论知识方面，缺乏对股票基本的常识和对股票一些基本操作的错误理解，同时对国内股票市场没有清晰的认识、缺乏股票数据分析相关知识等一系列原因，很多人买卖股票是相当盲目的，甚至是仅凭感觉操作，从而导致了价值投资错误。

2.在技术指标方面，在股票分析系统软件出现之前，股民只能通过手工作业的方式，通过人工记录每天的数据以及指标，画出K线图，通过固有公式自己运算出各项技术指标，这是相当繁琐而且费时的，往往因为某些数据和指标不能及时的计算得出，从而导致无法正确的去选择合适的投资股票。

3.对于市场结构以及资金运作方式的不了解。毫无疑问，在股市的众多结构中，市场资金结构是其根本性结构，它如同一只无形之手，推动市场的波动，决定市场的方向。缺少了大量的市场信息，以及数据计算后的结果，一般的股民对于整个市场结构的把握几乎处于空白状态，在2000年之前的中国股市，散户资金几乎占到了 60%-70%，然而就是在这样的情况下，众多散户依然成为为庄家赢利的最终买单者，损失惨重[4]。

以上问题的产生，主要是一般的股民难以在短时间内掌握股票的基本理论以及操作技巧，并且根本无法深入去分析一家公司的基本面情况，无法做到理性的去进行价值投资。

而现如今中国的股改已经基本完成，前流通的时代逐渐进入。机构的价值投资将是未来市场的主要投资形式。目前机构资金已经取代散户资金，成为了市场资金的主体，市场的资金结构已经悄然来了一个大转变，市场也出现了全新的赢利模式，

在这种全新的模式下，股票分析系统软件应时而生，以价值投资理念为指导，通过对公司及市场的大量客观数据进行科学的运算，再通过符合普通投资者使用习惯并且对投资真正有参考价值的形式表现出来，旨在帮助投资者能在短时间内找到真正的好公司，并把握股价的估值水平，从而对公司有个更深入的了解，增加成功的概率。

基于上述所说的种种情况，本次毕业设计的主要研究方向在于开发基于新的盈利模式下股票分析软件，利用 GDI+技术的强大图形显示的优势，从而能够让众多客户简单、清晰、直观和全面的了解股市的动态。

1.2 国内外研究现状

本次设计开发的股票分析系统，重点是开发研究其中有关 GDI+图形显示模块的工作。目前此类系统在国内采用 GDI+结合 C++以及.net 进行开发是较为主流的开发方式，而在国外则较多都采用以 WLD（Wealth-Lab Developer）为开发平台结合 C#为模式进行此类系统的开发和研究[5]。

国外的股票分析系统软件的设计理念更加趋向于公式化的计算，通过与大量的历史资料的对比与分析，而后向用户提供合理化的建议。国内的设计理念更加趋向于分析各种图表以及各种曲线的走势，通过分析资料给出一个图表和曲线的形式，很少有给用户更为直观的提示，大部分都是让用户自行通过图表以及各种曲线自行的进行决策，本系统在这方面有了一些突破，加入了股票风险预警机制提示功能，能够帮助客户起到一定风险规避的作用。

1.3 解决的主要问题

本股票分析系统着眼于对股票数据的分析，通过图形化界面向用户直观的展

示，因为涉及版权等一系列问题，所以不用来查看实时数据。

本着简单易操作、清晰直观性、数据准确性原则，股票分析系统使用简洁的操作界面，舍去了以往股票分析系统软件繁杂的操作步骤，让用户能够简单的掌握如何导入、导出股票数据，极大的避免了用户的非法操作和误操作的可能性，从而更加有利的保证了利用本系统分析股票得出的图表和数据的准确性。

在股票分析系统中，GDI+模块，也就是股票数据的图形显示模块是最重要部分之一，如何让用户以最快、最方便、最清晰的方式了解所关注股票的各种数据是GDI+模块设计的核心内容。

股票分析系统中GDI+的实现部分主要运用序列/反序列技术、XML技术、XML Schema自动生成类和GDI+技术，其中GDI+技术是这个模块实现的核心技术。

1.4 论文的组织结构

第一章绪论，主要描述股票分析系统的开发背景、GDI+技术的国内外研究现状，本文解决的主要问题和完成的工作。

第二章股票分析系统需求分析。首先进行了股票分析系统的概述。其次描述了该系统使用的技术和运行环境。最后对需求分析按照功能性需求和非功能需求两个类别进行描述。

第三章股票分析系统架构设计，主要进行系统的架构设计。首先对系统的设计目标和原则进行了阐述。接着分别描述了技术架构和功能架构的设计过程。

第四章股票分析系统详细设计，本章主要进行系统的详细设计。首先在系统建模部分，对主要模块进行了uml建模。然后，分析设计了数据库。

第五章股票分析系统实现。首先描述了系统的整体实现，并对各个模块的实现进行了描述。其次，着重分析了GDI+图形显示模块的实现。

第六章对论文进行了总结，并对系统的进一步提升提出了改进意见。

第 2 章 股票分析系统需求分析

2.1 系统概述

由于涉及到版权等一系列问题，本系统将不用来查看实时数据，只是用于保存股票历史交易数据信息，对股票历史数据进行显示和分析。

本系统在设计之初就以简单易操作、清晰直观、数据准确为目标，希望用户能够通过简单的操作便能够较为熟练的操作该系统，简洁的操作界面，舍去了以往股票分析系统软件繁杂的操作步骤，让用户能够简单的掌握如何导入、导出股票数据，极大的避免了用户的非法操作和误操作的可能性，从而更加有利的保证了利用本系统分析股票得出的图表和数据的准确性。

同时利用移动平均线法，即以道•琼斯的"平均成本概念"为理论基础，采用统计学中"移动平均"的原理，将一段时期内的股票价格平均值连成曲线，用来显示股价的历史波动情况，进而反映股价指数未来发展趋势，并通过 K 线图、股价分时图表、股票成交量图等大量清晰直观的图表的方式，让用户能够轻松的看到股票市场的变化和个股的异动，同时通过大量的股票数据信息计算，根据葛南维移动平均线八大法则来对股票市场进行风险预警。

本系统旨在向用户提供一个平台用于存储股票历史数据，随时查阅历史走势和其他一些分析数据，绘制任意时间段内的股票交易数据图表，还具有导入导出数据等功能。实现了多用户账户以及用户账户登录管理流程。

2.2 系统运行环境

本系统采用.Net Framework2.0 框架 和 MS SQL Server2005 数据库。主要运用的技术为 Win Form，GDI+，ADO.NET。

采用 C/S 模式设计，SQL SERVER 2005 数据库，Windows XP 操作系统（需要安装.Net Framework2.0）。小规模运用时，可以考虑选用 SQL SERVER 2005 EXPRESS 以节省成本。

2.3 系统需求问题描述

2.3.1 系统功能需求

根据以上对股票分析系统需求的描述，本系统要包括账号管理模块、用户管理模块、股票数据管理模块和自选股管理模块。

1.账号管理模块

账号管理模块包括登录、登出、修改密码、退出等功能。

1）登录

管理员或普通用户在登录界面输入自己的用户名和密码，提交到服务器，服务器将密码用 MD5 格式加密后，与数据库中的信息进行比较，如果相同，则允许该用户登录。同时在数据库中取出该用户的权限信息，登录后显示不同的菜单项。

2）登出

用户点击登出功能，系统清空当前用户的登录信息，关闭所有当前的子窗口，重新弹出登录框要求用户登录，同时主窗体上当前用户位置显示未登录。

3）修改密码

管理员和普通用户可以修改自己的账户密码，密码修改窗口显示当前用户的用户名，用户输入自己的当前密码，然后两次输入将要修改的新密码。系统判断两次新密码是否一致，如果一致，则提交服务器，服务器从数据库取出原密码比较是否相同，如果相同，允许用户修改密码，将新密码用 MD5 格式加密后存入数据库。

4）退出

用户点击退出功能，系统弹出提示框，询问用户是否确认关闭整个系统，得到确认后，股票分析系统关闭。

账号管理模块的用例图如图 2-1。

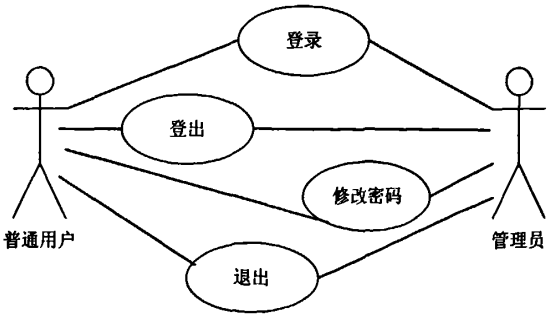

图 2-1 账号管理模块用例图

2. 用户管理模块

用户管理模块是供系统管理员使用的，用来管理系统的用户账户信息，包括添加用户、删除用户、重置用户密码和查询等功能。

1）添加用户

管理员点击添加用户，系统显示添加用户信息窗口，管理员填入用户名，设置该用户权限，点击保存。系统将自动为该用户设定初始密码为 0000，将该用户信息存入数据库，刷新用户列表。

2）删除用户

管理员在用户列表中选中想要删除的用户，点击删除按钮，系统弹出对话框询问是否确定删除，确认后在数据库中删除该用户信息，刷新用户列表。

3）重置用户密码

因为用户忘记密码或其他情况，管理员需要对用户重置密码。点击重置密码，系统将该用户密码重新设置为 0000。

4）查询用户

根据用户名，管理员可以精确或模糊查找该用户，选中后，可以进行后续的操作。

用户管理模块用例图如图 2-2。

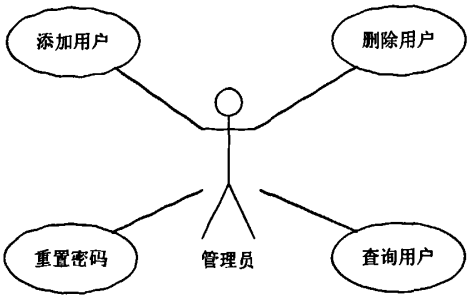

图 2-2 用户管理模块用例图

3.股票数据管理模块

股票数据管理模块包括查看股票详细数据、股票查询、导入数据、导出数据、保存为图片、打印报表等功能。

1）查看股票详细数据

用户选择了要查看的股票之后，系统界面可以显示所选股票的公司名称、开盘价、收盘价、最高价、最低价、成交量、交易日期等信息。

2）股票曲线图显示

用户选择了要查看的股票之后，系统界面可以显示所选定股票的 K 线图、成交量柱状图、均线等信息。这部分是本系统的主要功能部分，也是股票分析系统设计的意义所在,用户可以通过曲线图非常直观快速的了解股票的各种信息数据。

K 线图：反映单股涨跌的最重要的一种图形显示，在一个 K 线图上，可以获得在某一段时间里，股票的最高价，最低价，开盘价，收盘价，并根据以上 4 个价格关系，标志成图形单元，不同的颜色可以表示涨跌情况，分别用红绿来表示，红表示涨，绿表示跌。

收盘价线：是某一段时间里所有收盘价汇聚而成的一条曲线，可以得出这段时间内单股的走势情况。

成交量图：是某一段时间里所有成交量汇聚而成的一条柱状图，可以得出这段时间内单股的交易情况。

移动平均线：是一种趋势追踪的工具，便于识别趋势已经终结或反转，先的趋势正在形成或延续的契机。它不会领先与市场，只是忠实地追随市场，所以它具有滞后的特点，然而却无法造假。股票分析系统中有 5 日、10 日、30 日、60 日、120 日移动平均线。

3）股票查询

查询功能是用户与系统最重要的一个交互方式，用户可以通过这个查询得到所关注的股票的数据。用户在相应文本框中输入股票名，在时间框中输入查询起始时间、终止时间查询股票，在下拉菜单中选择查看形式，系统就可以以每日，每周或每月的形式在主窗口中显示该股票的相关信息。

4）导入导出数据

用户把从网上或其他途径搜集到的股票数据信息保存为 csv 格式，点击导入数据，系统弹出对话框要求用户选择文件路径，选好之后点击确定，系统将该文件中的股票数据信息保存到数据库中。

用户在系统中选择了某个股票，点击导出数据，系统弹出对话框，要求用户选择保存路径，选好后系统将该股票的数据信息导出到一个 excel 文件中。

5）保存图片

有的时候用户希望能够把看到的股票曲线图保存成图片的形式。用户在显示曲线图的界面上点击保存图片，系统弹出对话框要求用户选择保存路径和保存格式，用户选好后，系统将当前图片保存到指定位置。

6）打印报表

如果用户的电脑连接了打印机，可以随时打印自己需要的股票报表信息，以便于随时分析研究。

股票数据管理模块用例图如图 2-3。

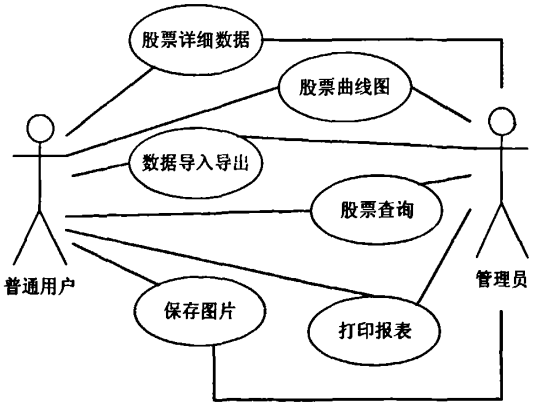

图 2-3 股票数据管理模块用例图

4. 自选股管理模块

用户可以根据股票名称或者股票代码来添加自选股，也可以在列表中删除自选股。

另外，由于相当多的用户对股票不是很了解，所以本系统通过对股票成交量、股价浮动变化进行公式化计算从而能够预知一定的股价变化，从而对股票市场的风险起到一定的预警和规避。

(1)查询

用户输入股票代码，系统可以在数据库中查找相应的股票。用户输入股票名称，系统在数据库中进行模糊查询，返回相应的股票信息。

(2)添加自选股

用户通过查询选择了股票之后，点击添加到自选股，系统将该支股票存入数据库自选股表中。

(3)删除自选股

当用户不再关注某支股票了，可以将其从自选股列表中删除，选中要删除的股票，点击删除，系统弹出对话框询问是否确定删除，确认后，系统从数据库自选股表中删除该支股票，刷新列表。

(4)风险预警

用户点击风险预警查看，系统根据股票的相关数据分析做出简单的预测，使用户可以有一个大致的了解。

自选股管理模块用例图如图 2-4。

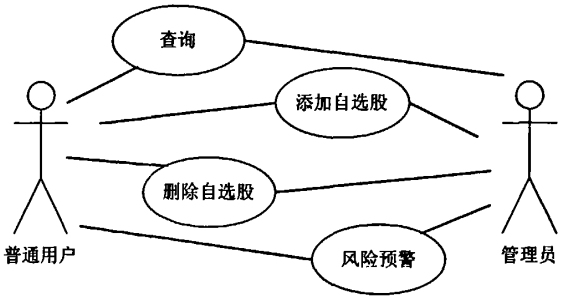

图 2-4 自选股管理模块用例图

综合分析以上各模块，股票分析系统的用例图如图 2-5 所示。

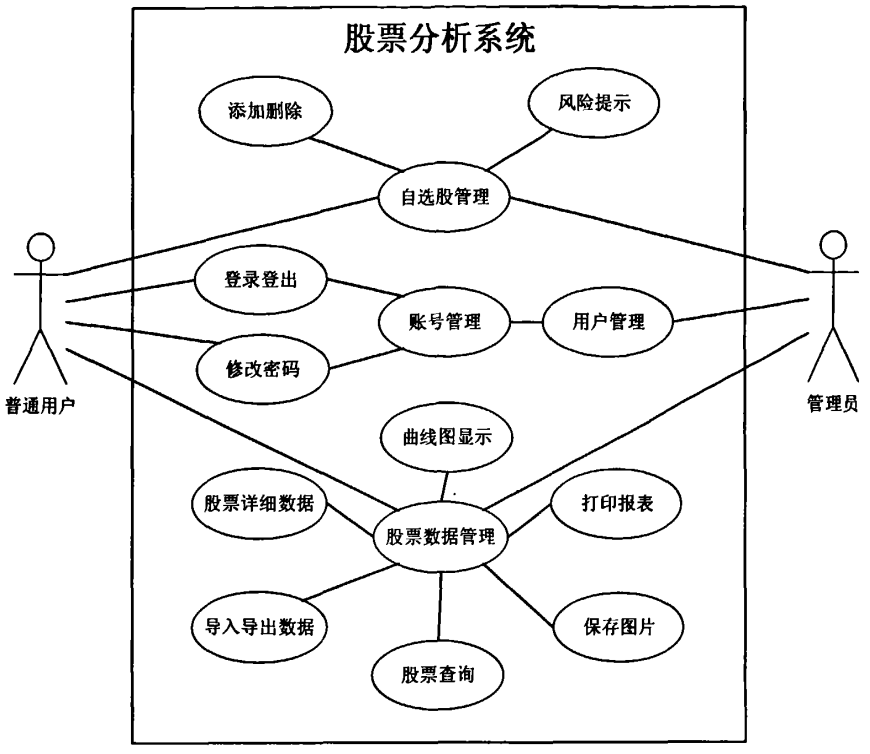

图 2-5 股票分析系统用例图

2.3.2 系统非功能性需求分析

1．运行期质量属性

（1）高性能，要求响应时间：

记录保存时间	小于 3 秒
点击响应时间	小于 3 秒
记录查询响应时间	小于 3 秒
分析响应时间	小于 5 秒
失败事务率	小于百分之三

（2）系统的易操作性，去除繁杂的系统操作界面，让用户能够在最短的时间内，只掌握几个简单关键的操作，便能够较为轻松的使用该系统，通过该分析系统使用自选股票，查询 K 线图、股价、成交量等主要功能，获取自己所需的股票数据信息。

（3）良好的数据可靠性和安全性，在用户个人操作方面出现问题（非法操作或者误操作），又或者是突发性的外界干扰因素（例如突然电脑断电等），能够及时并且准确的保存现有数据，不至于造成大量数据丢失，而对用户产生损失。

（4）数据的准确性和完整性，尽量采用相对简单的数据导入方式，避免繁杂的操作导致数据缺失，在系统运行时做到不会因为用户的操作不当，影响分析计算公式产生错误，从而导致数据准确性产生偏差。

2. 开发期质量属性

要求尽量减少连接数据库次数，每次读取的行数，每次写入数据库的行数。数据优化要达到一般的优化水平（即在不减少复杂的动态 SQL 查询的查询优化总量的情况下，能够同时适用于简单事务和查询的环境）。避免全表查询，尽量使用索引，减少更新事务。

第 3 章 股票分析系统架构设计

3.1 系统设计目标和原则

股票分析系统的建设目标是建设一套可视化和易操作性的股票数据分析系统，建立股票市场的众多股票信息数据与股票客户之间直观的、清晰的可视化操作界面，完美的结合包括 K 线分析、股票成交量统计、个股分时走势等几大功能，保证股票数据信息能够准确全面直观的反映给系统操作者即股民，要实现以上的目标需要遵循以下原则：

1．简单直观的系统分析界面：竭尽详细的将股票市场的时时动态能够直观的通过系统反映出来，让用户能够清晰的了解股票市场的资金主要流向、大盘走势以及所关注个股的详细的成交量异动、股价浮动等，能够让用户更加轻松详细的了解每支股票的变化。

2．易操作性：大部分用户对于该系统还是缺乏明确的概念，很难让用户在短时间内掌握一般股票分析系统软件的全部操作，而本次设计的股票分析系统将努力做到操作简单化，只让客户了解几个简单的操作，便能够正确的使用该系统，达到股票分析的目的。

3．股票数据的准确性：在现如今新的股票市场赢利模式下，数据细微的异动和变化，往往就预示或者更甚者影响整个股价的走势，相对于个股来讲，这种数据变化的影响就更加之大，因此本系统着重要致力于如何确保股票数据的准确性，避免错误的信息从而导致用户因此而产生的损失。

4．一定的风险预报和风险规避机制：数据是抽象的，即便是通过本系统能够给予客户一定程度上直观的分析图像，但是股票市场瞬息万变，很多时候通过图像直观反应出来的时候，用户可能已经损失惨重[6]，因此本系统开发了自动预警功能，在准确的股票数据信息基础上，通过系统本身融合的分析功能，从而通过数据的计算得出该个股的时段内的风险性，给出用户一定的参考性意见，从而能够一定程度上达到风险规避的目的。

3.2 系统技术架构

股票分析系统基于.net 平台，采用 C/S 架构实现，分层设计的理念。将数据与逻辑分离，逻辑与界面分离。界面层接收用户输入，并反馈输出结果。逻辑层执行各种实际数据操作，如画图，数据导入等。数据层负责提供逻辑层所需要的原始数据，逻辑层不关心数据来源。

具体如下：

1.表现层

使用 win form 构建客户端界面，主要负责与客户交互、接收用户输入、返回数据结果并展示给用户。

2.逻辑层

主要封装与企业业务相对应的操作、功能，业务逻辑层是本系统的核心部分，接受表现层的数据，调用数据层访问数据库。

3.数据访问层

与数据库服务器交互，为客户端程序提供基础数据、并为业务逻辑对象提供数据访问功能。

系统技术架构设计图如图 3-1 所示。

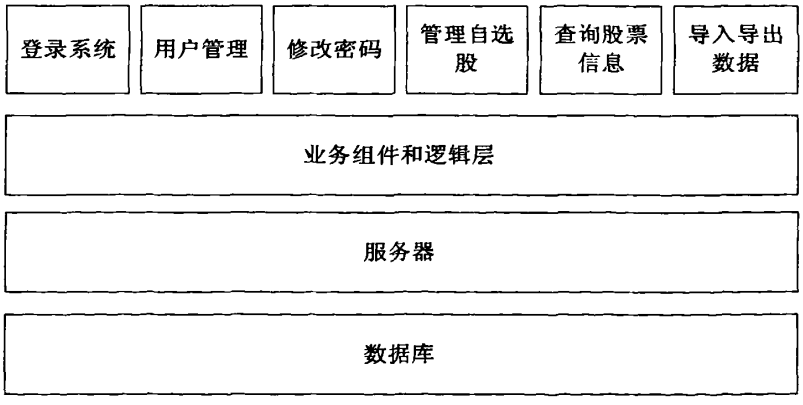

图 3-1 系统技术架构设计图

3.3 系统功能架构

3.3.1 系统功能组成

基于前面章节对股票分析系统的需求分析，设计本系统的功能架构如图 3-2。

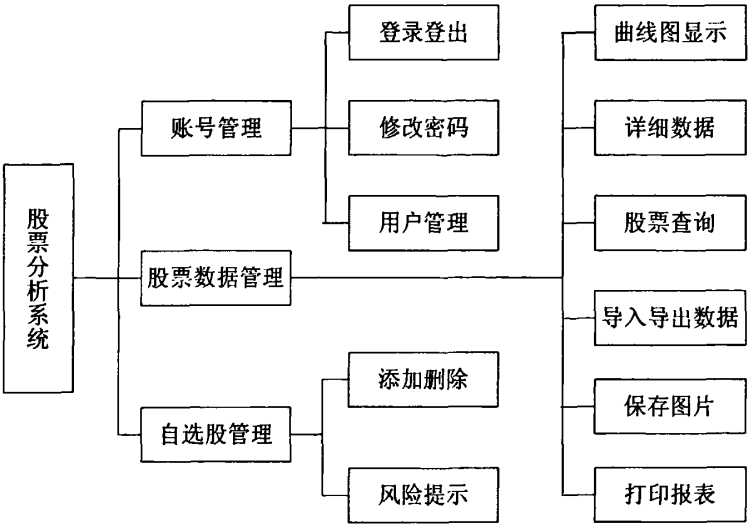

图 3-2 系统功能架构设计图

1. 账号管理：向用户提供登录登出功能，修改密码等基本信息的功能，并向具有管理员权限的用户提供用户管理功能。
2. 股票数据管理：这部分提供了与股票数据有关的各种功能，主要部分是曲线图显示功能。用户根据股票名称或代号查询股票，可以得到该股票的公司信息、开盘价收盘价等基本数据信息，并可以根据需要导出数据、保存图片或打印报表等。
3. 自选股管理：用户根据自己的情况选择关注的股票，系统根据用户的自选股情况提供相应的风险提示。

3.3.2 账号管理模块功能组成

账号管理模块是用户管理自己的账号以及管理员管理用户的功能模块，主要的功能结构如图 3-3。

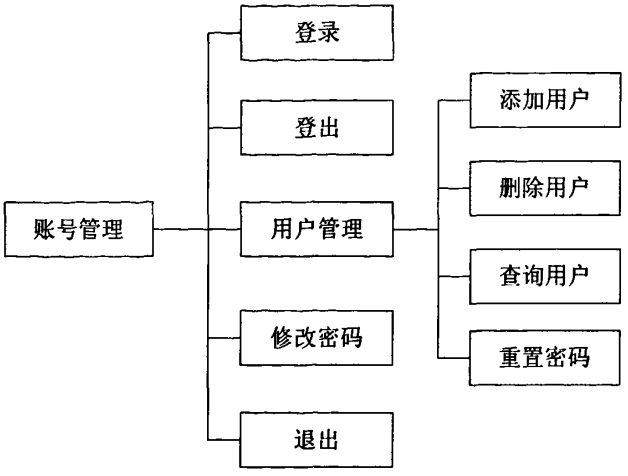

图 3-3 账号管理模块功能结构图

其中用户管理功能只对管理员可见，管理员除了普通用户所具有的登录、登出、修改密码等权限外，还可以向系统添加用户、删除用户、修改用户和重置用户密码。

3.3.3 股票数据管理模块功能组成

股票数据管理功能是股票分析系统的核心部分。股票数据管理功能组成如图 3-4 所示。

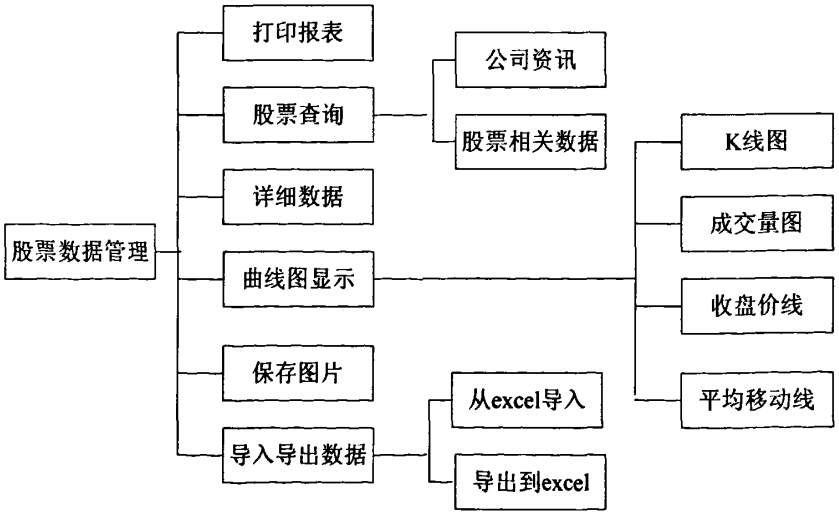

图 3-4 股票数据管理模块功能结构图

在股票分析系统中，股票数据的图形显示模块是最重要部分之一。根据前期调研了解的一些股票和股票的信息，可以以下面的图形显示线来主要表示股票的特征：

1.K 线图

2.历史收盘价曲线图：包括历史每日收盘价曲线图，历史每周平均收盘价曲线图，历史每月平均收盘价曲线图。

3.历史成交量柱状图：历史每日成交量柱状图，历史每周平均成交量柱状图，历史每月平均成交量柱状图。

4.当日实时价位曲线图

5.当日实时成交量柱状图

6.平均移动线：5 日平均移动线，10 日平均移动线，30 日平均移动线，60 日平均移动线，120 日平均移动线。

3.3.4 自选股管理模块功能组成

用户可以通过自选股管理模块来管理自己关注的股票，避免了被大量无关信息所干扰。自选股管理模块功能结构图如图 3-5。

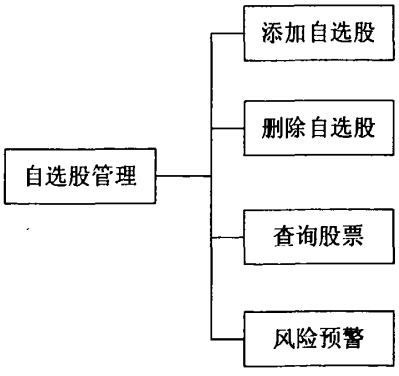

图 3-5 自选股管理模块功能结构图

第 4 章 系统详细设计

经过需求分析和架构设计，了解了股票分析系统的需求和架构流程。本章在此基础上，进一步分析系统的模型结构和数据库结构。

4.1 系统建模

4.1.1 账号管理模块

1. 登录功能

管理员或普通用户输入用户名及密码登录系统。系统将输入的信息和数据库中存储的信息进行比较，如果相符，则允许用户登录，否则，系统提示错误信息，要求用户重新输入。登录的时候，系统根据数据库信息判断用户是管理员还是普通用户，在接下来进入主系统时显示不同的菜单项，以赋予不同的权限。

为了提高系统的安全性，数据库中保存的用户密码是加密后的密码，这样可以增强系统的稳定性。本系统设计采用 MD5 加密格式对密码进行加密。

登录模块的活动图如图 4-1 所示。

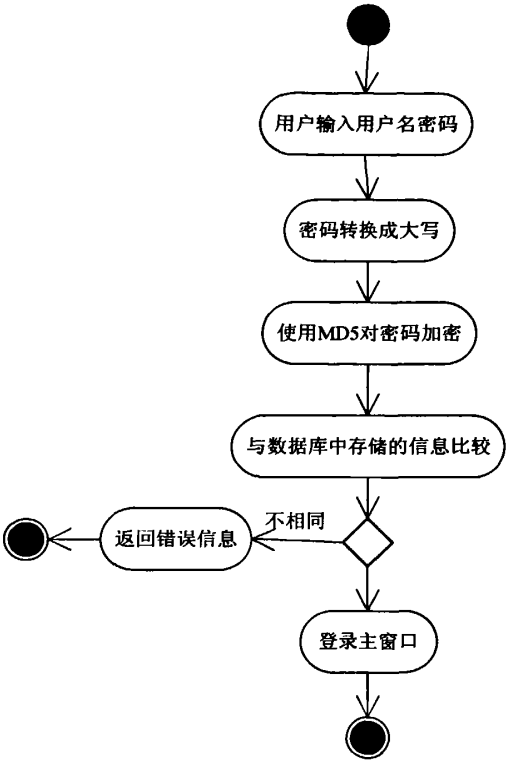

图 4-1 登录模块活动图

2. 登出和退出功能

按照用户一般习惯考虑，本系统将分别设计登出功能和退出功能。登出功能和退出功能的不同之处在于:用户选择登出的时候只是将系统的状态变为未登录，而选择退出的时候，整个程序结束运行。用户点击登出按钮后，系统主窗体显示未登录，清空当前用户自选股显示信息，将用户指针置为空，除登录菜单项和图形按钮外，其余置为灰色不可用状态。

登出模块活动图如图 4-2 所示。

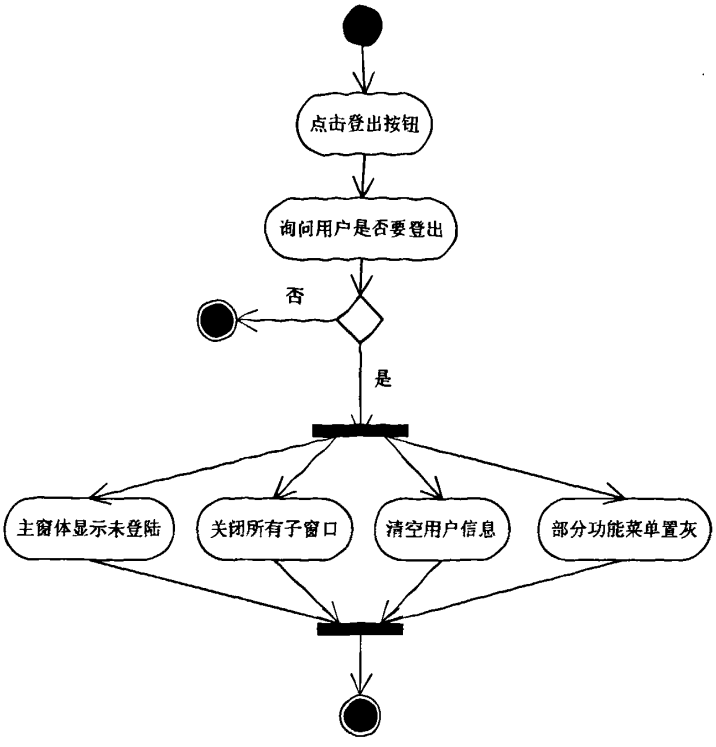

图 4-2 登出模块活动图

3. 用户管理功能

系统管理员有用户管理的权限，主要功能有添加用户，删除用户，重置密码等。在添加用户的窗口中，可以选择将要添加的用户是管理员或是普通用户。

用户管理模块的类图如图 4-3。

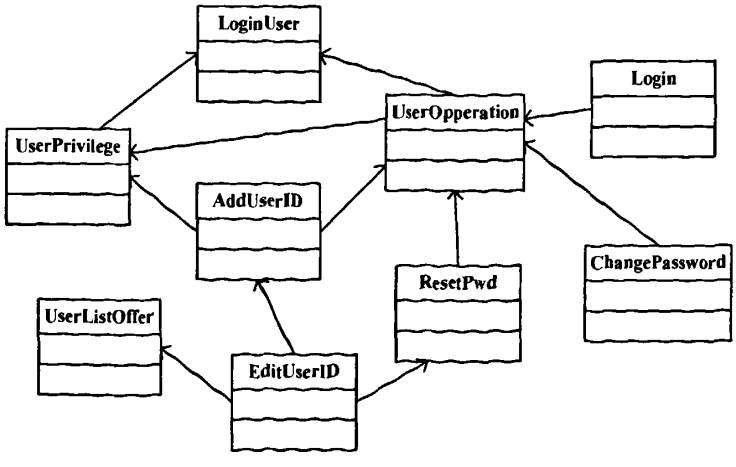

图 4-3 用户管理模块类图

其中添加用户功能的活动图如图 4-4 所示。

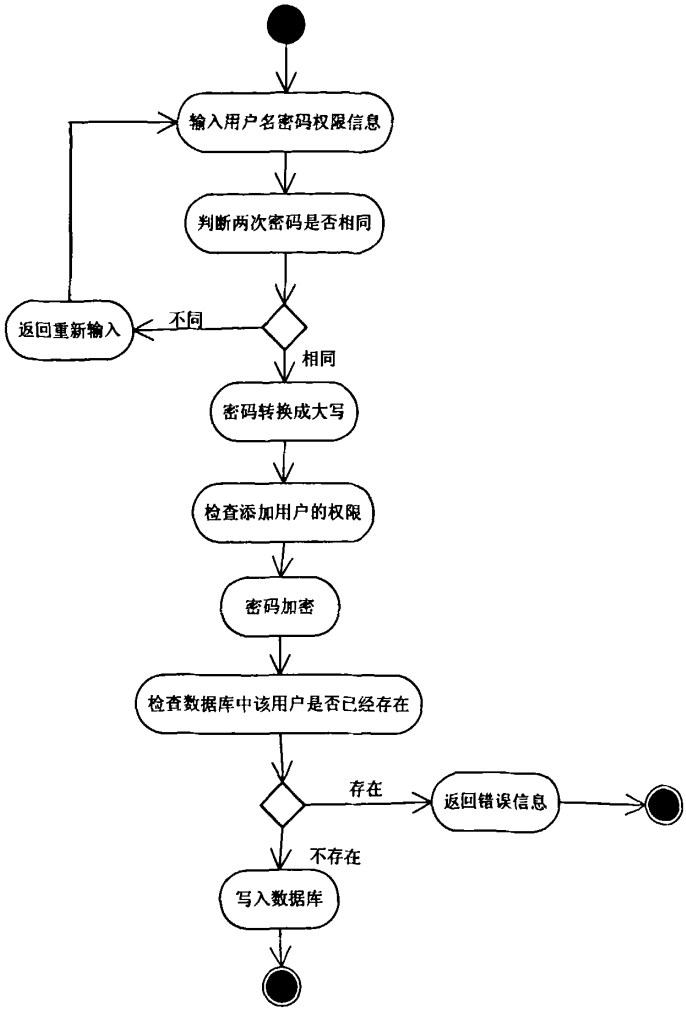

图 4-4 添加用户模块流程图

4.1.2 自选股管理模块

1．添加删除功能

根据用户的习惯考虑，在列表中添加了自选股之后，还要相应的在该自选股显示上添加点击响应功能，用户点击选中的股票后，窗体自动切换到股票数据图表的显示，这样方便了用户的操作。

在执行了添加删除操作后，列表要刷新。采用的办法是先将列表全部清空，

然后再重新加载。

2. 风险管理功能

股票分析系统提供了简单的风险管理功能。通过对股票成交量、股价浮动变化进行公式化计算从而能够预知一定的股价变化，从而对股票市场的风险起到一定的预警和规避。

股票分析系统主要根据以下几点进行风险提示。

(1)如果当前交易日成交量很大，是上一交易日的3倍以上，且股价大幅度收阳（K线图为红色），那么第二天高开的概率将是相当大的，此时若是用户持有该股并有出货的意向，则可以下一个价格相对高的单，能够及时的在高点出货。而如果情况正好相反，那么低开的可能比较大，如果低开的话，那么用户就可以考虑适量买入，从而能够在相对低点买入该股，此时的风险相对较低。

（2）根据换手率判断。在低位成交活跃（成交量高于前日成交量1倍以上）、换手率高（5%以上）、而股价涨幅不大的个股（股价上下浮动不超过5%），通常为庄家吸货，而庄家在吸货之后，在一段时间内会逐步拉高股价以求出货，此时系统将给出提示可以买进，那么用户就可以根据系统提示，能够低价买入，再在合适股价抛出；

（3）委比。某品种当日买卖量差额和总额的比值[7]。是衡量某一时段买卖盘相对强度的指标。它的计算公式为：委比＝（委买手数－委卖手数）/(委买手数＋委卖手数)×100%。委买手数：现在所有个股委托买入下三档的总数量。委卖手数：现在所有个股委托卖出上三档的总数量。委比值的变化范围为－100%到＋100%，当委比值为负时，卖盘比买盘大，此时系统就会提示用户，该股抛盘压力较大，可能在未来一段时间内股价下跌，此时可以考虑将手中该股抛出；而委比值为正时，说明买盘比卖盘大，该股有较大上涨趋势可能，此时系统会给出提示，用户可在此时适量买入该股。

4.1.3 股票数据管理模块

股票数据管理模块是股票分析系统的核心模块，用户通过该模块可以获取所关注股票的相关数据信息以及查看对应的各种股票曲线图。

1. 导入数据功能

导入数据是指用户可以将搜集到的股票信息导入数据库，然后通过股票分析系统来生成获取各种技术曲线，以便更好的分析股票信息。导入数据库的文件为csv文件格式，得到csv文件的方法为：用excel另存->保存为csv[8]。

CSV是指逗号分隔值文件（Comma Separated value），是一种用来存储数据的纯文本文件格式，通常用于电子表格或数据库软件[9]。CSV文件有如下规则。

（1）开头不留空，以行为单位。

（2）可含或不含列名，含列名则居文件第一行。

（3）一行数据不垮行，无空行。

（4）以半角符号作分隔符，列为空也要表达其存在。

（5）列内容如存在，（逗号），则用“”（引号）包含起来。

（6）列内容如存在“”（引号）则用“”包含。

（7）文件读写时引号，逗号操作规则互逆。

（8）内码格式不限，可为ASCII、Unicode或者其他[10]。

一个符合规则要求的导入文件文本如下所示：

Date,Open,High,Low,Close,Volume,Adj Close

2007-05-17,25.98,26.61,25.94,26.06,33579200,26.06

2007-05-16,25.04,25.98,24.99,25.92,31919100,25.92

2007-05-15,25.20,25.50,24.64,24.67,25419300,24.67

将数据导入数据库的模块序列图如图4-5所示。

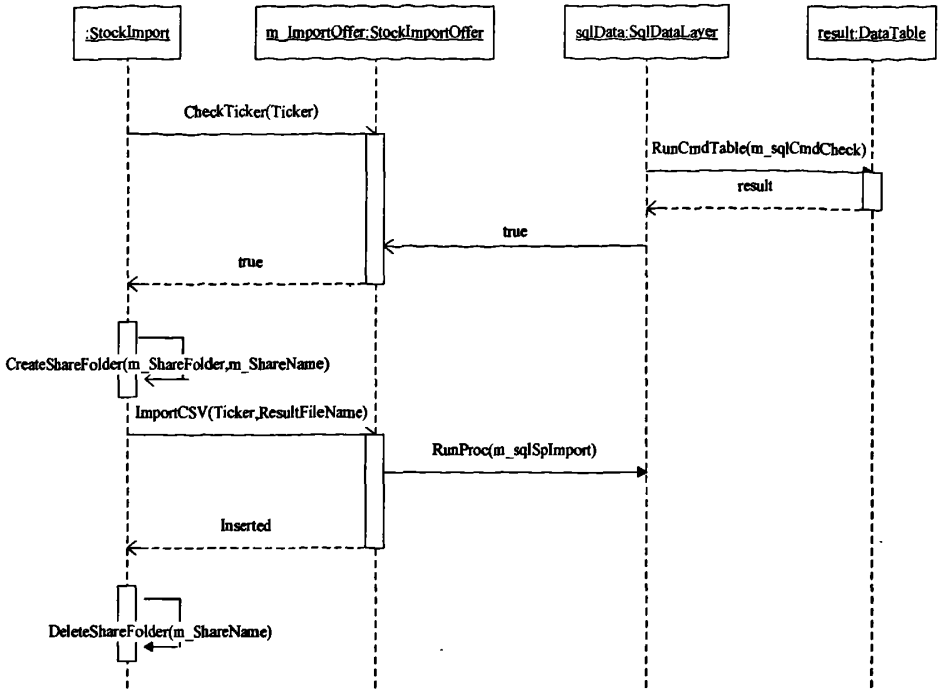

图 4-5 数据导入模块序列图

2．数据导出功能

此模块将数据库中的股票信息导出到一个 excel 表格中，保存为.xls 格式。用户选择好保存位置等信息后，点击确定，系统先获取数据库中股票的具体信息，返回一个结果，然后再将这个结果作为参数传递，导出数据。

数据导出模块的序列图如图 4-6 所示。

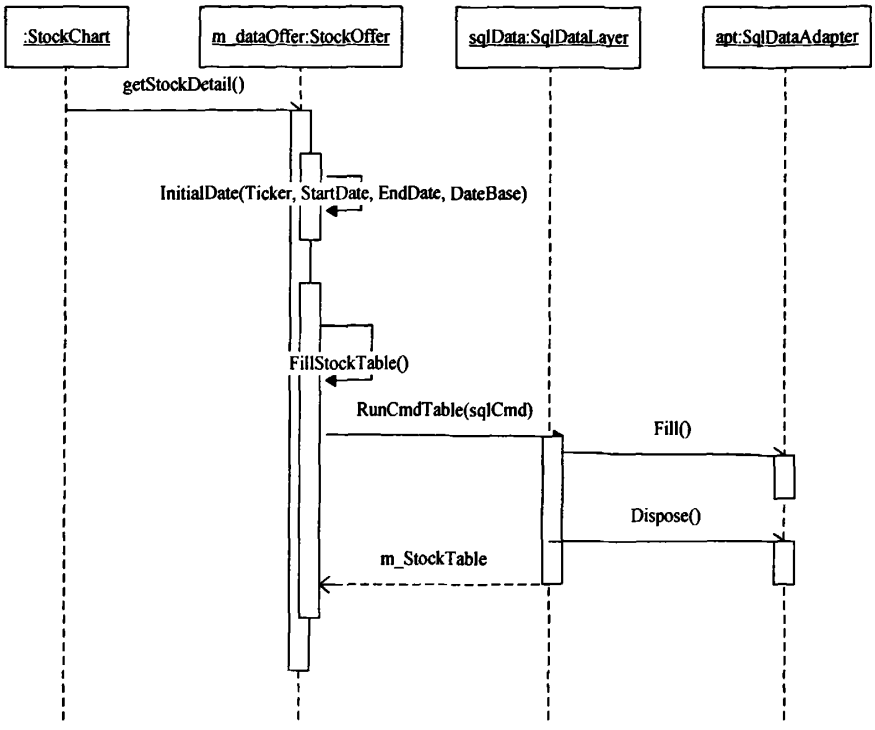

图 4-6 数据导出模块序列图

3．查询功能

查询这个小模块是用户与系统最重要的一个交互方式，用户可以通过这个查询得到想要的数据。通过股票名，查询起始时间、终止时间查询股票，可以以每日，每周或每月在图形显示主窗口中显示。

股票数据管理模块中的图形显示功能和图片显示功能都用到了 GDI+技术。

GDI 是位于应用程序与不同硬件之间的中间层，这种结构让程序员从直接处理不同硬件的工作中解放出来，把硬件间的差异交给了 GDI 处理。GDI 通过将应用程序与不同输出设备特性相隔离，使 Windows 应用程序能够毫无障碍地在 Windows 支持的任何图形输出设备上运行。它把 windows 系统中的图形输出转换成硬件命令然后发送给硬件设备[11]。GDI 是以文件的形式存储在系统中，系统需要输出图形时把它载入内存，如果转换成硬件命令时遇到非 GDI 命令，系统还可能载入硬件驱动程序，驱动程序辅助 GDI 把图形命令转换成硬件命令。在 Visual Studio.NET 中 Microsoft 解决了 GDI 中的许多问题,并让它变得易用,GDI 的.net

版本叫做 GDI+[12]。

图 4-7 是显示股票信息的类图。

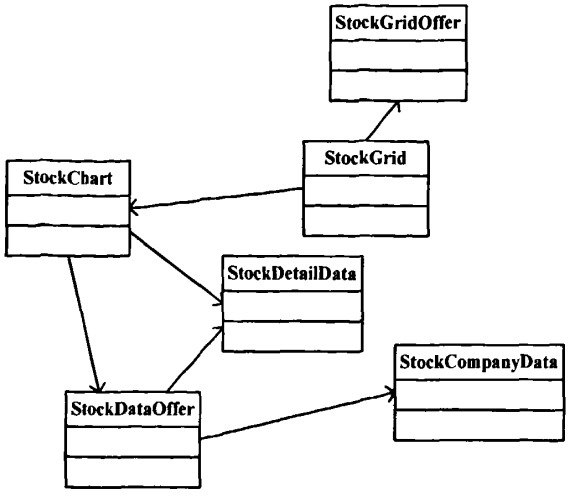

图 4-7 显示股票信息的类图

类 StockGrid 显示整体的股票数据显示窗口，StockGridOffer 类处理其中具体的相关操作，StockChart 类显示股票曲线图。StockChart 类需要 StockDataOffer 类具体执行操作，还要用到 StockDatilData 类中的相关函数。在 StockDataOffer 类中，用到 StockCompanyData 的相关操作信息。

4. 保存图片功能

保存图片功能是图形操作中的重要功能之一。在保存一个图像时，图像相应的类型信息也必须进行保存；也就是说，该图像的扩展名在这一过程中具有重要角色。每一种类型相应于一个特定的格式。实质上，在保存一个图像时，根据该格式输出数据是非常必要的。借助于 GDI+ API 的优势，一个对 Image 类的 Save() 方法的简单调用就可以把相应的写数据操作中所有细节省略掉。这个方法使用两个参数：被保存的图像的名字和待保存图像的格式。该格式能够通过 ImageFormat 类提供的类型来指定[13]。GDI+支持的图片格式有：Bmp、Emf、Exif、 Gif、Guid、Icon、Jpeg、MemoryBmp、Png、Tiff、Wmf。

5. 图形显示功能

GDI+这个模块相对独立于其他的模块，它只负责图形的设计和实现。所以这个模块与其他模块的交互也相对比较简单，外部模块在实例化的时候把要画图的

Panel 传进来就可以了。下面对 DrawPictrue 类进行设计。该类主要通过 Draw 和 DailyDraw 函数来实现图形显示。Draw 函数是主窗口的图形显示，DailyDraw 函数是股票日信息的图形显示。一个 GetStockInfo 函数返回一个 StockDetailData 类型的类，以确定当前鼠标所指时间段的股票详细数据。还有个 StockDetailData 和 StockPerMinItem 的 List 属性，根据这个 List 所包含的数据，画出相应的图形。StockDetailData 是主窗口中的数据类，StockPerMinItem 是股票日信息的数据类。

系统中 GDI+部分都是通过封装在 Grahpics 中的各种方法画成的，考虑到图形的可扩展性，为方便以后添加新的股票图形，把 GDI+主窗口设计成两大块，即画刻度和画各种图形。

4.2 系统数据库设计

4.2.1 数据库设计分析

根据以上对股票分析系统实际需求的分析，设计数据库主要考虑以下需求：

系统管理者或者系统用户通过相应的用户名和密码登录到股票分析系统。若用户名或密码有误，系统不让其登录，并提示错误信息；若用户名和密码都符合要求，管理者或一般用户顺利登录系统，然后可以操作，管理股票分析系统。

系统管理者可以添加、删除系统管理者或系统用户，更改系统管理者或系统用户的登录密码，更改自己的登录密码。而系统用户没用管理其他用户的权限。

系统管理者和系统用户除了在用户管理模块上的权限不同之外，其他模块上都是一样的，所以两者都可以查看所有股票的列表，导入股票数据，查看单只股票的详细信息（包括收盘价线，k 线图，成交量图，平均移动线等等），关注个股，并可以对关注股进行管理操作。

分析以上需求，系统需要管理系统管理者，系统用户信息和股票信息，所以要把这些信息保存在数据库中。

系统管理者和系统用户的属性基本一致，可以将系统管理者和系统用户保存在相同的存储文件中。权限的细节存储于另外的存储文件中。

系统需要给用户提供的股票信息包括每一天的交易信息和实时的当前交易信息。需要分别存在数据库中。

由于系统在给用户提供股票信息的时候需要同时把该股所属公司的信息也提供给用户，所以需要将公司信息也保存在数据库中。

另外用户可以关注个股，则可以把用户与其关注股票的关系存在数据库中。

4.2.2 数据库建表

按照以上对数据库设计的分析，下面为数据库建表。

存储有关上市公司基本信息的表 StockCompany 见表 4-1。

表 4-1 StockCompany

字段名	类型	描述	约束
Ticker	Varchar(20)	分类	Not null
Stock_Name	Varchar(200)	股票名称	唯一，Not null
Sector_Name	Varchar(200)	证券类别	Not null
Industry_Group_Name	Varchar(200)	行业类别	Not null

存储股票日交易信息的表 StockDetail 见表 4-2。

表 4-2 StockDetail

字段名	类型	描述	约束
Trade_Day	Datetime	交易日	Not null
Open_Price	money	开盘价	Not null
High_Price	money	最高价	Not null
Low_Price	money	最低价	Not null
Close_Price	money	收盘价	Not null
Volume	bigint	成交量	Not null
Stock_Name	Varchar(200)	股票名	唯一，Not null

存储股票日交易实时信息表 StockDaily 见表 4-3。

表 4-3 StockDaily

字段名	类型	描述	约束
Trade_Day	Datetime	交易日	Not null
DayTime	Datetime	时间	Not null
Price	money	价格	Not null
Volume	bigint	成交量	Not null
Stock_Name	Varchar(200)	股票名	唯一，Not null

存储用户账户相关信息表 UserInfo 见表 4-4。

表 4-4 UserInfo

字段名	类型	描述	约束
UserId	int	用户 id	唯一，Not null
UserName	Varchar(50)	用户名	Not null
PasswordHash	Varchar(50)	加密密码	Not null
Privilege	int	权限	Not null

存储用户关注股票信息表 UserStock 见表 4-5。

表 4-5 UserStock

字段名	类型	描述	约束
Id	int	自动排序 id	唯一，Not null
UserId	int	用户 id	Not null
Stock_Name	Varchar(200)	股票名	Not null

第 5 章 股票分析系统实现

从股票分析系统的用户群体角度考虑，应将本系统界面设计的简洁、方便使用。

根据系统的主要功能进行划分，将主菜单确定为：账号管理、股票数据、我的股票、窗口和帮助。其中账号管理的子菜单项为登录、登出、更改密码、用户管理和退出，股票数据的子菜单项为股票列表、打印报表、打印预览、导入数据、导出数据，我的股票的子菜单项为管理自选股、风险提示，窗口的子菜单项为排列、层叠、横排和纵排。

另外根据用户的喜好及使用频率情况，在主窗口上设置了图形菜单项：登录、登出、更改密码、用户管理、自选股列表、打印预览、打印报表、导入数据、编辑自选股列表、导出数据、收藏、保存图片。

未登录时，菜单项股票数据和我的股票为灰色、不可用。

股票分析系统主窗口如图 5-1。

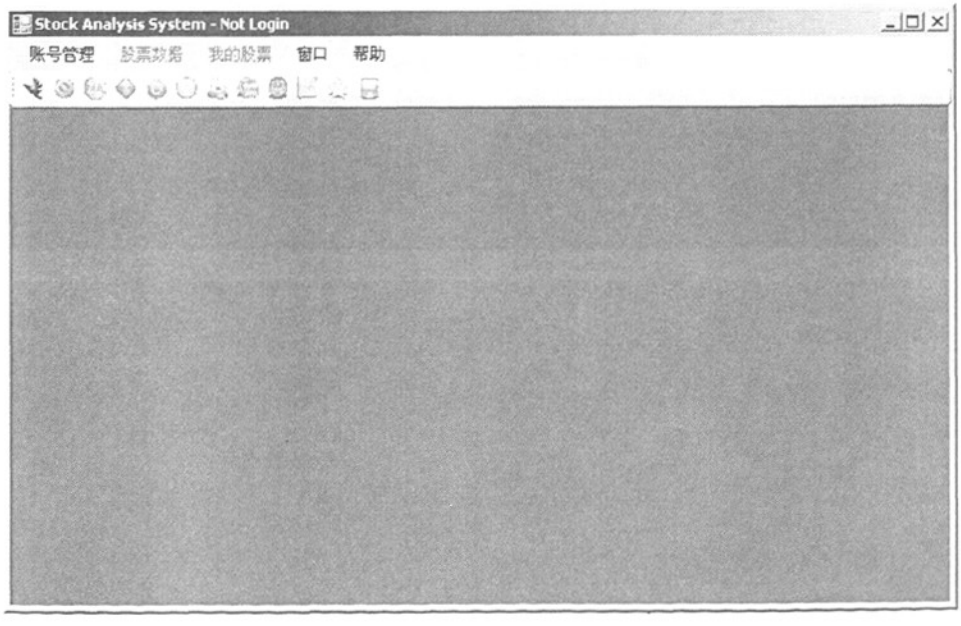

图 5-1 股票分析系统主窗口

5.1 账号管理模块

这个模块的实现比较简单，比较重要的是实现了对密码的加密存储以及登出功能不同于退出功能的操作。

1．密码加密

数据库中保存的用户密码是加密后的密码，这样可以增强系统的安全性。本系统采用 MD5 加密格式对密码进行了加密[14]。将密码加密的代码如下：

```
public static string GetMD5PassWord(string realPassWord)
        {
            if (realPassWord == null)
            {
                throw new Exception("Real password can not been NULL");
            }
            else
            {
                string hashPassword =
System.Web.Security.FormsAuthentication.HashPasswordForStoringInConfigFile(realPassWo
rd, "md5");
                return hashPassword;
            }
        }
```

2．登出功能

用户选择登出的时候，系统主窗体显示未登录，清空当前用户自选股显示信息，将用户指针置为空，除登录菜单项和图形按钮外，其余置为灰色不可用状态。下面代码实现了登出功能。

```
private void 登出ToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if(DialogResult.Yes == MessageBox.Show("Are you sure to logout?",
"Information", MessageBoxButtons.YesNo, MessageBoxIcon.Information))
            {
                loginUser = null;
                this.Text = "Stock Analysis System - Not Login";
                ClearUserStock();
```

```
            loginUser = null;
            登录ToolStripMenuItem.Enabled = true;
            menuManage.Enabled = false;
            menuStockData.Enabled = false;
            menuMyStock.Enabled = false;
            更改密码.Enabled = false;
            股票列表.Enabled = false;
            用户管理.Enabled = false;
            股票管理.Enabled = false;
            导入数据.Enabled = false;
            登出.Enabled = false;
            登入.Enabled = true;
            打印预览.Enabled = false;
            打印报表.Enabled = false;
            foreach (Form form in this.MdiChildren)
            {
                form.Close();
            }
        }
    }
```

5.2 股票数据管理模块

5.2.1GDI+图形显示

股票分析系统GDI+图形显示界面主要有2个panel和2个groupBox，分别的作用为：一个用于GDI+主窗口图形显示的panel，一个用于单股单日信息的panel，一个用于显示某个时间点详细信息的 groupBox，一个提供给用于查询的groupBox。GDI+图形显示界面见图5-2。

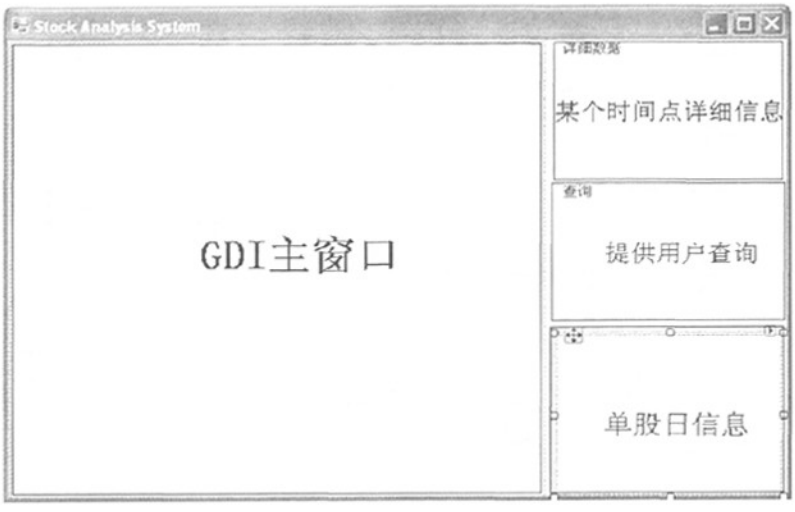

图 5-2 GDI+图形显示界面

图形显示功能的实现用到了 xml 技术、xml schema 技术和序列/反序列技术。

1) Xml 技术

XML 是 Extentsible Markup Language（可扩展标记语言）的缩写，是用来定义其它语言的一种元语言，其前身是 SGML(标准通用标记语言)[15]。

在 Visual C# 中，可以将使用 XML 编写的代码文档化。C# 是 Visual Studio .NET 中唯一具有此项功能的编程语言[16]。

股票分系统中的 XML 文档主要是用来配置 GDI+图形显示界面中股票图形的颜色，通过 XML 文档可以配置用户自己喜欢的股票图形色调[17]。下面是 XML 配置文档。

```
<?xml version="1.0" encoding="utf-8" ?>
<DrawConfiguration xmlns="http://tempuri.org/DrawAttributes.xsd">
  <Configs>
    <GridColor>Navy</GridColor>
    <ClosePriceColor>Brown</ClosePriceColor>
    <VolumeColor>Blue</VolumeColor>
    <VolumeBrush>Blue</VolumeBrush>
    <KLineUpperColor>Red</KLineUpperColor>
    <KLineLowerColor>Green</KLineLowerColor>
    <MA5Color>Gray</MA5Color>
    <MA10Color>Blue</MA10Color>
    <MA30Color>Navy</MA30Color>
```

```
    <MA60Color>Yellow</MA60Color>
    <MA120Color>HotPink</MA120Color>
  </Configs>
</DrawConfiguration>
```

2) Xml schema 技术

XML Schema 描述了 XML 文档的结构。可以用一个指定的 XML Schema（通常以 xsd 扩展名结尾）来验证某个 XML 文档[18]，以检查该 XML 文档是否符合其要求。如果符合的话，那么该 XML 文档被称为是有效的（valid），否则它就是非有效的（invalid）。可以通过 XML Schema 指定一个 XML 文档所允许的结构和内容，并可据此检查一个 XML 文档是否是有效的[19]。

XML 架构定义工具从 XDR、XML 和 XSD 文件或者从运行库程序集中的类生成 XML 架构或公共语言运行库类[20]。使用.NET Framework 中自带的 XML 架构定义工具（Xsd.exe）执行从 XDR 转换到 XSD，XML 转换到 XSD，XSD 到 DataSet，XSD 到类，类到 XSD 的操作；本系统采用了从 XSD 转换到类的操作，图 5-3 是一个 XML Schema 的设计。

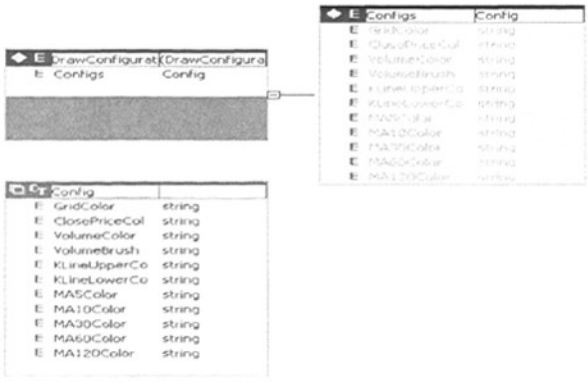

图 5-3 xml schema 设计范例

以上 XML Schema 的设计的代码如下。

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="DrawAttributes" targetNamespace="http://tempuri.org/DrawAttributes.xsd"
elementFormDefault="qualified" xmlns="http://tempuri.org/DrawAttributes.xsd"
```

```
xmlns:mstns="http://tempuri.org/DrawAttributes.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="DrawConfiguration">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Configs" type="Config" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="Config">
    <xs:sequence>
      <xs:element name="GridColor" type="xs:string" />
      <xs:element name="ClosePriceColor" type="xs:string" />
      <xs:element name="VolumeColor" type="xs:string" />
      <xs:element name="VolumeBrush" type="xs:string" />
      <xs:element name="KLineUpperColor" type="xs:string" />
      <xs:element name="KLineLowerColor" type="xs:string" />
      <xs:element name="MA5Color" type="xs:string" />
      <xs:element name="MA10Color" type="xs:string" />
      <xs:element name="MA30Color" type="xs:string" />
      <xs:element name="MA60Color" type="xs:string" />
      <xs:element name="MA120Color" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3) 序列/反序列技术

序列化是将对象状态转换为可保持或传输的格式的过程。与序列化相对的是反序列化，它将流转换为对象。这两个过程结合起来，可以轻松地存储和传输数据[21]。

.NET Framework 提供了两种序列化技术[22]。

二进制序列化保持类型保真度，这对于在应用程序的不同调用之间保留对象的状态很有用。例如，通过将对象序列化到剪贴板，可在不同的应用程序之间共享对象。用户可以将对象序列化到流、磁盘、内存和网络等等。远程处理使用序列化“通过值”在计算机或应用程序域之间传递对象。

XML 序列化仅序列化公共属性和字段，且不保持类型保真度。当用户要提供或使用数据而不限制使用该数据的应用程序时，这一点是很有用的。由于 XML 是

一个开放式标准，因此，对于通过 Web 共享数据而言，这是一个很好的选择。SOAP 同样是一个开放式标准，这使它也成为一个颇具吸引力的选择[23]。

股票分析系统中采用 XML 的反序列，就是把 XML 配置文档中数据流转换成一个相应的对象，以供其它模块使用，以下是反序列的代码。

```
private void GetXMLFile(string XMLFilePath)
        {
            StreamReader sr = null;
            try
            {
                sr=new
            StreamReader(Stock_Analysis_System.Forms.StockMain.ApplicationStartPath
                        + "\\" + XMLFilePath);
            XmlSerializer xmlSerializer = new XmlSerializer(typeof(DrawConfiguration));
            DrawConfig = (DrawConfiguration)xmlSerializer.Deserialize(sr);
            }
            catch
            {
            }
            finally
            {
                sr.Close();
            }
        }
```

下面是图形的具体绘制实现过程。

1. GDI+主窗口

K 线图，收盘价曲线，成交量柱状图，平均移动线都在这个主窗口中显示，如果只有这些线型图的话，主窗口呈现给用户的视觉感觉比较单调、空洞，所以设计 GDI+主窗口还应在相对空旷的地方，加上些单股的说明信息，本系统就在 GDI+主窗口的左上角，加入了单股的总结数据，图 5-3 是股票名为 AAPL 在主窗口的显示状态。

图 5-3 股票 AAPL 在主窗口的显示状态

2. 主窗口坐标的绘制

坐标的确定也是 GDI+部分的难点，由于是在 Panel 上作图，所以起始坐标是相对于 Panel 的坐标，这样做就方便了很多。图 5-4 是股票名为 MMM 的刻度框架。

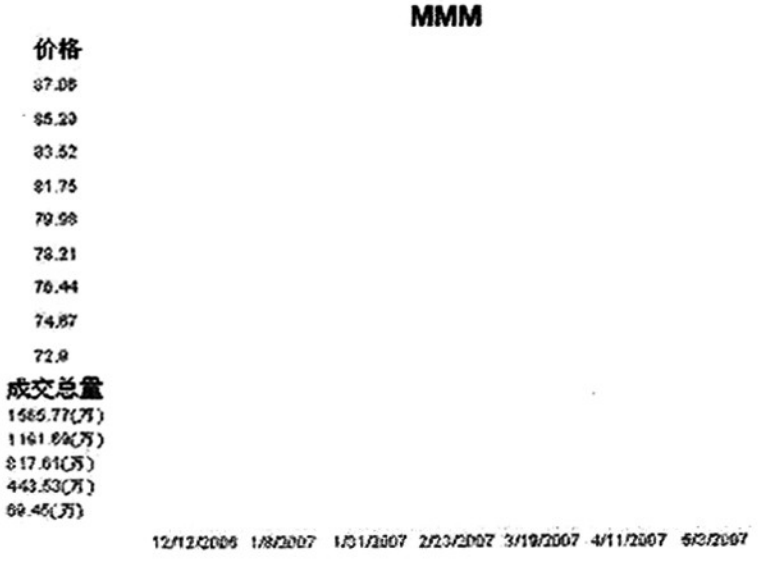

图 5-4 刻度框架

X 轴中的时间刻度也是通过 List 中 StockDetailData 所包含的时间跨度所确定的，固定分 8 个小段。由于刻度坐标都已画好，所以画股票图形只需找到对应的坐标点，然后调用 Graphics 的作图方法即可。X 轴实现的框架代码：

```
private void DrawingXAxisNumber(PointF startPoint, PointF endPoint, int dividePart)
{
    if (StockDataList.Count <= 8)
    {
        //当数据中交易日的天数没有 8 天的情况，即不能分成 8 小段，就显示所有天
数。
    }
    else
    {
        //当数据中交易日的天数不少于 8 天的情况，即可以分成 8 小段，把天数平均
分成 8 段来显示。
    }
}
```

Y 轴中价格的度量和成交量的度量是通过 List 中 StockDetailData 所包含的最大收盘价、成交量和最小收盘价、成交量的跨度所确定的。

Y 轴实现的框架代码：

```
private void DrawingYAxisNumber(PointF startPoint, PointF endPoint, int dividePart, bool
isStockVolume)
{
    if (isStockVolume == false)
    {
        // 标识收盘价的 Y 轴坐标
    }
    else
    {
        // 标识成交量的 Y 轴坐标
    }
}
```

3. K 线图的绘制

K 线图是股票分析软件的最重要、最常用的一种图，它是通过对一段时期内股价变动情况的分析来找出未来股价变动的趋势。

图 5-5 是股票分析系统中 AAPL 股票从 2007-3-5 到 2007-5-17 这段时间的日 K 线图。

图 5-5 K 线图

在已有刻度坐标的情况下，只要知道直线的两个端点以及方形的起始点坐标和长宽的大小就可以画出单个 K 线图，从上文中对 K 线图的了解，K 线图有 4 个变量组成：开盘价、收盘价、最高价和最低价，反映过去一段时间内的股价的波动情况。本系统也同样以直线和方形来画单个 K 线图的，首先画直线，即最高价和最低价的一条连线，然后画方形，即开盘价和收盘价组成的方形。这样顺序画的效果是让直线不贯穿整个方形。而方形的宽度定义为网格时间刻度的 1/2。直线和方形的长度则是由具体的价格对应到坐标上的刻度所决定的。

下面是单个 K 线图的代码：

```
private void DrawStockSingleKLine(double openPrice, double closePrice, double highPrice,
                                  double lowPrice, int xAxisNum)
{
    PointF openPricePoint = GetPointByRealNum(offsetStartPoint, offsetMiddlePoint.Y,
                                              xAxisNum, openPrice, false);
    PointF closePricePoint = GetPointByRealNum(offsetStartPoint, offsetMiddlePoint.Y,
                                               xAxisNum, closePrice, false);
    PointF highPricePoint = GetPointByRealNum(offsetStartPoint, offsetMiddlePoint.Y,
                                              xAxisNum, highPrice, false);
    PointF lowPricePoint = GetPointByRealNum(offsetStartPoint, offsetMiddlePoint.Y,
                                             xAxisNum, lowPrice, false);
    if (openPrice < closePrice)
    {
```

```
            tmpSolidBrush.Color = drawAttributes.KLineUpperColor;
            tmpRect.X = closePricePoint.X - xAxisTradeNum / 4;
            tmpRect.Y = closePricePoint.Y;
            tmpRect.Width = xAxisTradeNum / 2;
            tmpRect.Height = openPricePoint.Y - closePricePoint.Y;
            tmpPen.Color = drawAttributes.KLineUpperColor;
            graphics.DrawLine(tmpPen, highPricePoint, lowPricePoint);
        }
        else
        {
            tmpSolidBrush.Color = drawAttributes.KLineLowerColor;
            tmpRect.X = openPricePoint.X - xAxisTradeNum / 4;
            tmpRect.Y = openPricePoint.Y;
            tmpRect.Width = xAxisTradeNum / 2;
            tmpRect.Height = closePricePoint.Y - openPricePoint.Y;
            tmpPen.Color = drawAttributes.KLineLowerColor;
            graphics.DrawLine(tmpPen, highPricePoint, lowPricePoint);
        }
    if (openPricePoint.Y - closePricePoint.Y > -1 && openPricePoint.Y - closePricePoint.Y < 1)
        {
            if (openPrice <= closePrice)
            {
                tmpPen.Color = drawAttributes.KLineUpperColor;
                graphics.DrawLine(tmpPen, closePricePoint.X - xAxisTradeNum / 4,
                                  closePricePoint.Y, closePricePoint.X + xAxisTradeNum / 4,
                                  closePricePoint.Y);
            }
            else
            {
                tmpPen.Color = drawAttributes.KLineLowerColor;
                graphics.DrawLine(tmpPen, closePricePoint.X - xAxisTradeNum / 4,
                                  closePricePoint.Y, closePricePoint.X + xAxisTradeNum / 4,
                                  closePricePoint.Y);
            }
        }
        graphics.FillRectangle(tmpSolidBrush, tmpRect);
    }
```

上面的代码中，GetPointByRealNum 方法取得实际价格或成交量在图形窗口上的坐标值；drawAttributes 类中定义了很多颜色属性，值是从配置文件读出来的。把单个 K 线图联起来就形成一个真正的 K 线图。

4. 成交量柱状图

成交量柱状图从 2007-1-22 到 2007-5-17，如图 5-6 显示：

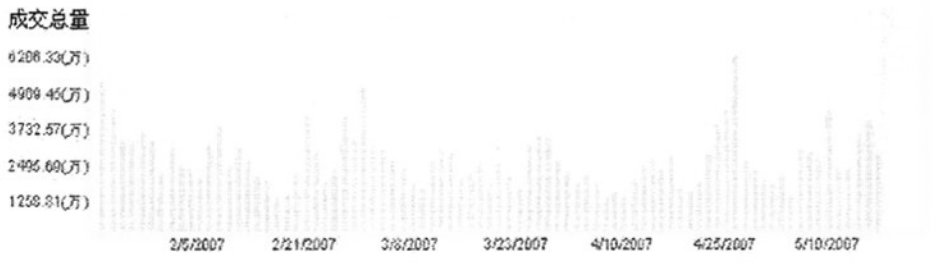

图 5-6 柱状图

根据上面 K 线图的绘制代码，可以完成成交量柱状图的绘制代码。成交量柱状图的方形宽度可以和 K 线中方形宽度一致，长度由实际的成交量而定。

画成交量柱状图程序先画单个柱状图，然后合并起来，放在同一个函数里实现。定义 StockDataList.Count 个矩形，以指向单个柱状图。定义 StockDataList.Count 个坐标点，以指向单个柱状图的成交量坐标点。接着取得单个成交量的值在图形窗口上的坐标值，给定义的每个矩形赋值，对应于每个成交量柱状图。最后给每个已定义的成交量柱状图填充颜色。

5. 收盘价曲线图

经过以上 K 线图和成交量柱状图的分析及实现，收盘价曲线图的绘制就容易了很多，收盘价曲线图只要在坐标图上找到收盘价所以对应的坐标点，然后用直线连接起来就完成了。图 5-7 是 2006-11-20 到 2007-5-17 这个时间段里的日收盘价曲线图。

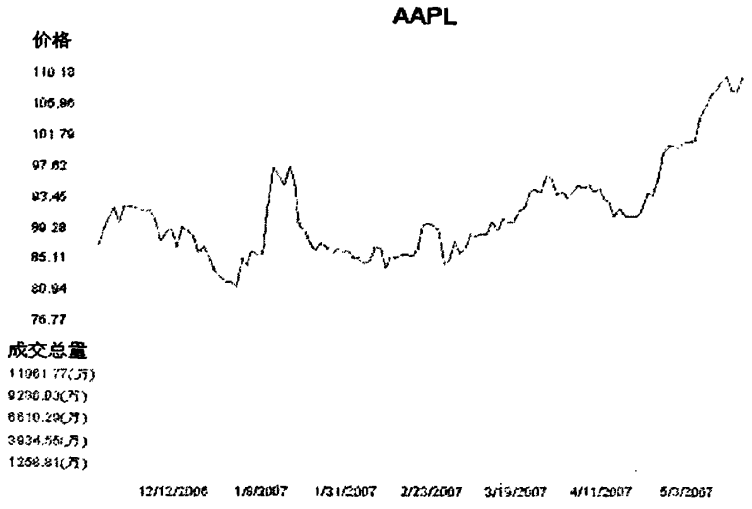

图 5-7 收盘价曲线图

6. 移动平均线

移动平均线分时日可以绘制出很多曲线，系统中有 5 日、10 日、30 日、60 日和 120 日五种[24]。

移动平均线是指一段时间内的算术平均线，通常以收盘价作为计算值。其公式为 MA（N）=（第 1 日收盘价+第 2 日收盘价+……+第 N 日收盘价）/N[25]。

图 5-8 是 MA(5) 、MA(10)、MA(60)在 2006-11-20 到 2007-5-17 时间段的三条曲线（MA(5)苍白色、MA(10)蓝色、MA(60)黄色）。

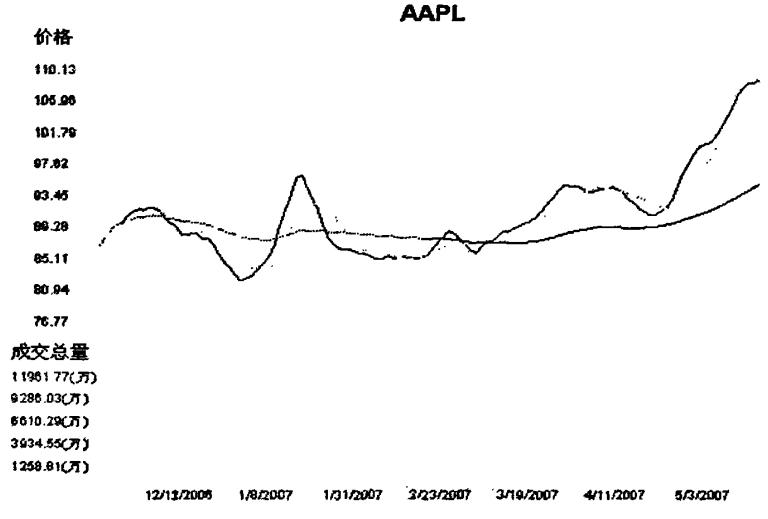

图 5-8 移动平均线

由于查询的数据是从 2006-11-20 开始的，所以要是 5 日均线的话，从 2006-11-20 开始的那 5 天是不能形成数据的，程序中用模拟均线的方法，取前几日的平均值来表示，图中的虚线部分就是模拟均线，所以越是长期的均线，虚线部分就越长。

定义 dayNumber 个坐标点，用来指向虚拟均线（虚线部分）的值。定义 StockDataList.Count - dayNumber 个坐标点，用来指向移动平均线（实线部分）的值。先取虚拟均线坐标点的平均收盘价(查询天数 <= N)，取得虚线部分的坐标点，然后 MA(N)某天的平均收盘价(查询天数 > N)，取得实线部分的坐标点。接着绘制虚线部分，绘制虚线与实线的交接部分，绘制实线部分。下面是移动平均线的实现方法。

```
private void DrawStockMA(int dayNumber)
{
    int i = 0;
    double sumClosePrice = 0, averageClosePrice;
   tmpPen = new Pen(tmpMAColor, 1);
   PointF[] MAVirtualPointF = new PointF[dayNumber];
    PointF[] MAPointF = new PointF[StockDataList.Count - dayNumber];
    for (i = 0; i < dayNumber; i++)
    {
        sumClosePrice += StockDataList[i].ClosePrice;
        averageClosePrice = sumClosePrice / (i + 1);
        MAVirtualPointF[i] = GetPointByRealNum(offsetStartPoint, offsetMiddlePoint.Y,
                                                i + 1, averageClosePrice, false);
    }
    for (i = dayNumber; i < StockDataList.Count; i++)
    {
        sumClosePrice = sumClosePrice + StockDataList[i].ClosePrice -
                        StockDataList[i - dayNumber].ClosePrice;
        averageClosePrice = sumClosePrice / dayNumber;
        MAPointF[i-dayNumber]=GetPointByRealNum(offsetStartPoint,
                                    offsetMiddlePoint.Y, i+1, averageClosePrice, false);
    }
   tmpPen.DashStyle = System.Drawing.Drawing2D.DashStyle.Dot;
    graphics.DrawLines(tmpPen, MAVirtualPointF);
   tmpPen.DashStyle = System.Drawing.Drawing2D.DashStyle.Solid;
    graphics.DrawLine(tmpPen, MAVirtualPointF[dayNumber - 1], MAPointF[0]);
   if (MAPointF.Length != 1)
        graphics.DrawLines(tmpPen, MAPointF);
```

```
    tmpPen.Dispose();
}
```

5.2.2 导入导出数据

目前来看，最合适用来开发 Office 外接程序的工具其实是 VB，因为 Office 的对象内核依然是以 VB 的模式建立的[26]。VB 中有大量的函数使用可变数量的参数[27]，而这些函数如果在 C#中调用，则不得不用大量的 Type.Missing 或 Missing.Value 来填充，因为 C#不允许可变参数[28]。在本方法中，使用 Missing.Value 来填充未知参数。

导出功能将数据库中的股票信息导出到一个 excel 表格中，保存为.xls 格式。用户选择好保存位置等信息后，点击确定，系统先调用 GetStockDetail 方法获取数据库中股票的具体信息，返回一个 List 类型的结果，然后再将这个结果作为参数传递，调用 ExportStockDetail 方法导出数据。具体代码如下：

```
public static void ExStockDetail(string outFileName, List<StockDetailData>outStockDetail)
        {
            object missing = Missing.Value;
            Microsoft.Office.Interop.Excel.Application app;
            Microsoft.Office.Interop.Excel.Workbook workbook;
            Microsoft.Office.Interop.Excel.Worksheet worksheet;
            Microsoft.Office.Interop.Excel.Range range;
            app = new Microsoft.Office.Interop.Excel.Application();
            app.Visible = false;
            workbook = app.Workbooks.Add(Missing.Value);
            worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.Sheets[1];
            string[,] arr = new string[outStockDetail.Count + 1, 6];
            arr[0, 0] = "TradeDay";
            arr[0, 1] = "OpenPrice";
            arr[0, 2] = "HighPrice";
            arr[0, 3] = "LowPrice";
            arr[0, 4] = "ClosePrice";
            arr[0, 5] = "Volume";
            range = (Microsoft.Office.Interop.Excel.Range)worksheet.Cells[1, 1];
            range = range.get_Resize(1, 6);
            range.Font.Bold = true;
```

```
            for (int i = 0; i < outStockDetail.Count; ++i)
            {
                arr[i + 1, 0] = outStockDetail[i].TradeDay;
                arr[i + 1, 1] = outStockDetail[i].OpenPrice.ToString();
                arr[i + 1, 2] = outStockDetail[i].HighPrice.ToString();
                arr[i + 1, 3] = outStockDetail[i].LowPrice.ToString();
                arr[i + 1, 4] = outStockDetail[i].ClosePrice.ToString();
                arr[i + 1, 5] = outStockDetail[i].StockVolume.ToString();
            }
            range = (Microsoft.Office.Interop.Excel.Range)worksheet.Cells[1, 1];
            range = range.get_Resize(outStockDetail.Count + 1, 6);
            range.Value2 = arr;
            workbook.SaveAs(outFileName, missing, missing, missing, missing, missing,
        Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlExclusive, missing, missing,
                                                missing, missing, missing);
        }
    }
```

5.3 自选股管理模块

1．查询股票

用户可以通过查询功能得到想要的数据。通过股票名，查询起始时间、终止时间查询股票，可以以每日，每周或每月在GDI+主窗口中显示。由于股票的数据量是巨大的而且查询的动作是频繁的，所以本系统在查询的时候做了一个优化。

优化的方法类似网页技术中所使用的缓存技术[29]，首次查询某一只股票的时候，在DataTable中装入的是从查询起始日期前180交易日到截至日期的数据，当第二次查询这个股票的时候，如果查询的日期在这段时间内，数据就直接从DataTable中读取，如果超出范围就重新装入数据。因为装入数据是从数据库中读取数据的，所以时间远远大于从内存中读取数据[30]。图5-9是查询的界面。

图5-9 查询界面

以上是查询的小界面，其中<- 表示查询起始时间向前推移一段时间，这段时间长短与用户查询股票的类型 Daily，weekly，monthly 有关；-> 表示查询终止时间向后推移一段时间，直到数据库中数据的最新更新日期；<< 表示查询的时间段向前平移一段时间；>> 表示查询的时间段向后平移一段时间。

2．显示自选股

显示自选股的操作并不是简单的在列表中显示自选股信息，还要相应的在该自选股显示上添加点击响应功能，用户点击选中的股票后，窗体自动切换到股票数据图表的显示。

向窗口加载自选股信息的代码如下：

```
public void AddUserStock()
    {
        List<UserStockItem> UserStockList=UserStockOperation.GetUserStock(loginUser);
            if (UserStockList == null)
            {
                return;
            }
            int Count = UserStockList.Count;
            ToolStripMenuItem[] menuStocks = new ToolStripMenuItem[Count];
            for (int i = 0; i < Count; ++i)
            {
                menuStocks[i] = new ToolStripMenuItem();
                menuStocks[i].Text = UserStockList[i].Ticker;
```

```
                menuStocks[i].ToolTipText = UserStockList[i].StockName;
                menuStocks[i].Click += new EventHandler(UserStock_Click);
            }
            menuMyStock.DropDownItems.AddRange(menuStocks);
            return;
        }
```

第6章 结 论

股票分析系统在充分了解项目背景及论证系统需求的基础上，已经全部开发完成，并发布给用户使用。现将本文所做的主要工作总结如下：首先，本文对系统结构进行了分析，提出系统开发所使用的平台；随后，对系统的各功能模块进行了详细的介绍，重点介绍了股票数据管理模块这一主要模块；最后，本文结合部分截图和代码对系统实现进行了具体说明。

由于时间仓促等原因，系统还有不完善的地方，比如曲线图实现方面没有囊括股票数据指标的全部种类，而只是选择了其中用户比较常用，比较熟悉了解的种类进行了实现。这部分内容将在以后的系统升级改进中进一步完善。

参考文献

[1]（美）奥利维尔·布兰查德 宏观经济学（第2版 国际版）北京：清华大学出版社，2003年

[2] 程鹏 股市趋势实战技术分析 北京：中国发展出版社，2004年

[3] 袁枫 Windows图形编程 北京：机械工业出版社，2007年

[4] 高上 中国股市十八年 上海：东方出版中心，2008年

[5] 梁爱虎 SOA思想、技术与系统集成应用详解 北京：电子工业出版社，2007年

[6] 丁圣元 股票大作手操盘术 北京：企业管理出版社，2003年

[7] 林俊国 股票投资一点通 北京：清华大学出版社，2008年

[8] Excel Home Excel应用大全 北京：人民邮电出版社，2008年

[9]（美）拉森 Microsoft SQL Server 2005 Reporting Services 北京：清华大学出版社，2008年

[10]（美）马瑞克（Marick,B.） Everyday Scripting with Ruby 北京：电子工业出版社，2008年

[11] 周鸣扬，赵景亮 精通GDI+编程 北京：清华大学出版社，2004年

[12] 周鸣扬 GDI+程序设计实例 北京：中国水利水电出版社，2008年

[13]（美）昌德（Chand,M.） .NET技术大系：GDI+图形程序设计 北京：电子工业出版，2005年03月

[14] 王曦，杨健 网络安全技术与实务 北京：电子工业出版社，2006年

[15] 孙更新，裴红义，杨金龙 XML完全开发指南 上海：科学出版社出版，2008年

[16] Nagel C. C#2005 &.NET 3.0高级编程（第5版）上下卷 北京：清华大学出版社，2007年

[17] 刘晓平 计算机技术与应用进展（2007）（上下册） 北京：中国科学技术大学出版社，2007年

[18] 孙鑫 XML、XML Schema、XSLT 2.0和XQuery开发详解 北京：电子工业

出版社，2009 年

[19] 孙晓非 XML 基础教程与实验指导 北京：清华大学出版社，2008 年

[20] 桂浩，陈刚，范昊 XML 开发技术教程 武汉：武汉大学出版社，2008 年

[21] （美）杰瑞夫（Jeffrey, J.），（法）克里斯托夫（Christophe, N.） Windows 核心编程 北京：清华大学出版社，2008 年

[22] 曾洁玫 GDI+程序设计实例 北京：中国水利水电出版社，2004 年

[23] 毛新生 SOA 原理·方法·实践 北京：电子工业出版社，2007 年

[24] 黎航 股市操练大全：主要技术指标的识别与运用练习专辑 上海：上海三联书店，2007 年

[25] （台湾）黄韦中 主控战略移动平均线:透析平均战法的完全攻略密笈 北京：地震出版社，2006 年

[26] 屠立刚，吴翠凤 Microsoft Office SharePoint Server 2007 管理大全 北京：电子工业出版，2008 年

[27] （美）哈尔弗森 Visual Basic 2008 从入门到精通 北京：清华大学出版社，2008 年

[28] （美）Wei-Meng Lee C#与 VB.NET 网络通信开发实战 北京：人民邮电出版社，2008 年

[29] （美）桑德斯（Sounders，S.） 高性能网站建设指南 北京：电子工业出版社，2008 年

[30]（美）内格尔（Nagel.C） C#高级编程(第 6 版) 北京：清华大学出版社，2008 年

致 谢

感谢我的导师赵合计老师，在我写作的过程中，对我的论文提出了许多宝贵的改进意见，对我的成长起到了很大的帮助。

感谢我的朋友们，一直在身边支持我，鼓励我。

感谢我的母校山东大学，我在这里完成了本科和研究生一共六年的生活。感谢母校给予我的一切。

基于.net平台的股票分析系统的设计和实现

作者：克远
学位授予单位：山东大学