**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC HOA SEN**

**KHOA KHOA HỌC VÀ CÔNG NGHỆ**



**BÁO CÁO ĐỀ TÀI**

**CÔNG NGHỆ DỊCH VỤ WEB**

***Tên đề tài:***

**Xây dựng ứng dụng webservice**

**cung cấp thông tin chứng khoán**

**Giảng viên hướng dẫn : Võ Hoàng Hải**

**Nhóm sinh viên thực hiện : Nguyễn Minh Huy – 081667**

**Nguyễn Anh Khoa – 080835**

**Lương Ngọc Lan – 080838**

**Khóa : QL081**

**Tp.HCM, năm 2011**

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

MỤC LỤC

[DANH MỤC BẢNG – HÌNH ẢNH V](#_Toc311676203)

[LỜI MỞ ĐẦU 1](#_Toc311676204)

[NỘI DUNG 2](#_Toc311676205)

[1. Cơ sở lý thuyết 2](#_Toc311676206)

[1.1 Giới thiệu công nghệ 2](#_Toc311676207)

[1.1.1 Web service 2](#_Toc311676208)

[1.1.2 WCF 2](#_Toc311676209)

[1.2 Quy trình xây dựng một web service 3](#_Toc311676210)

[2. Thiết kế chương trình 4](#_Toc311676211)

[2.1 Tổng quan về ứng dụng 4](#_Toc311676212)

[2.2 Các chức năng chính trong ứng dụng 5](#_Toc311676213)

[2.2.1 Nhóm chức năng Financial Analysis Service 5](#_Toc311676214)

[2.2.1.1 FinancialIndexService 5](#_Toc311676215)

[2.2.1.2 RecommendStockService 6](#_Toc311676216)

[2.2.2 Nhóm chức năng Technical Analysis Service 8](#_Toc311676217)

[2.2.2.1 PriceService 8](#_Toc311676218)

[2.2.2.2 NewsService 9](#_Toc311676219)

[2.2.2.3 TechnicalAnalysisService 10](#_Toc311676220)

[2.2.3 Nhóm chức năng Forecast Service 13](#_Toc311676221)

[2.3 Thiết kế cơ sở dữ liệu 13](#_Toc311676222)

[3. Hiện thực 18](#_Toc311676223)

[3.1 Giao diện chính của ứng dụng 18](#_Toc311676224)

[3.2 Name standard 18](#_Toc311676225)

[3.3 Hiện thực các chức năng 20](#_Toc311676226)

[3.3.1 Nhóm chức năng Financial Analysis Service 20](#_Toc311676227)

[3.3.1.1 GetTheBasicIndicators 20](#_Toc311676228)

[3.3.1.2 GetFinanceReport 22](#_Toc311676229)

[3.3.2 Nhóm chức năng Technical Analysis Service 24](#_Toc311676230)

[3.3.2.1 GetPrice 24](#_Toc311676231)

[3.3.2.2 GetMostIncrease, GetMostDecrease 25](#_Toc311676232)

[3.3.2.3 GetMostTraded 27](#_Toc311676233)

[3.3.3 Nhóm chức năng News 28](#_Toc311676234)

[3.3.4 Nhóm chức năng Forecast Service 30](#_Toc311676235)

[3.3.4.1 GetForecast 30](#_Toc311676236)

[KẾT LUẬN 32](#_Toc311676237)

[1. Đánh giá 32](#_Toc311676238)

[2. Hướng phát triển 32](#_Toc311676239)

[TÀI LIỆU THAM KHẢO 33](#_Toc311676240)

DANH MỤC BẢNG – HÌNH ẢNH

[Bảng 1 – Công thức tính toán các chỉ số. 21](#_Toc310841805)

[Bảng 2 – Ví dụ cách trình bày bảng báo cáo tài chính. 22](#_Toc310841806)

[Hình 1 – Cấu trúc cơ sở dữ liệu 12](#_Toc311629393)

[Hình 2 – Giao diện tổng quan 16](#_Toc311629394)

[Hình 3 – Xem chỉ số và báo cáo tài chính 18](#_Toc311629395)

[Hình 4 – Cấu trúc đối tượng List<FinancialRecord> 21](#_Toc311629396)

[Hình 5 – Xem giá cổ phiếu ACB 22](#_Toc311629397)

[Hình 6 – Xem các mã cổ phiếu giảm giá nhiều nhất 23](#_Toc311629398)

[Hình 7 – Xem các mã cổ phiếu được giao dịch nhiều nhất 25](#_Toc311629399)

[Hình 8 – Xem tin tức 26](#_Toc311629400)

[Hình 9 – Xem phân tích của chuyên gia 27](#_Toc311629401)

[Hình 10 – Xem dự báo giá các phiên giao dịch sắp tới 28](#_Toc311629402)

LỜI MỞ ĐẦU

Hiện nay, Web Service đang là một trong những lựa chọn yêu thích của các nhà phát triển vì tính tương thích cao và dễ hiện thực của nó, đặc biệt là với cơ sở hạ tầng internet đang không ngừng được nâng cấp, việc sử dụng Web Service để truyền tải thông tin, trao đổi dữ liệu giữa các nền tảng công nghệ khác nhau (.NET, Java, PHP…) ngày càng trở nên phổ biến. Trong số các nền tảng công nghệ của Web Service, WCF do Microsoft hỗ trợ được xem như là một trong những nền tảng mạnh mẽ nhất bởi sự nhất quán và tự động của nó, nhà phát triển không cần làm những công việc cấu hình lặp lại ở các tầng bên dưới nữa mà chỉ cần tập trung vào phát triển nghiệp vụ cho ứng dụng Web Service.

NỘI DUNG

1. Cơ sở lý thuyết
   1. Giới thiệu công nghệ
      1. Web service

Web Services là một chuẩn để xây dựng và phát triển ứng dụng phân tán, có khả năng làm việc trên mọi hệ điều hành, mở rộng khả năng phối hợp giữa các ứng dụng, có thể tái sử dụng, tăng cường sự giao tiếp giữa Client và Server thông qua môi trường Web.

Web Services được xây dựng dựa trên SOAP (Simple Object Access Protocal). SOAP có thể được gọi thực hiện và trả về kết quả text (theo định dạng XML), do đó, có khả năng hoạt động rộng lớn với các ứng dụng phần mềm khác nhau chạy trên những nền tảng khác nhau.

Tuy nhiên, web service vẫn tồn đọng một số nhược điểm cần khắc phục, đó chính là vấn đề an toàn và bảo mật. Bên cạnh đó, tồn tại khá nhiều chuẩn sẽ khiến người dùng khó khăn trong việc nắm bắt thông tin.

* + 1. WCF

Là công nghệ nền tảng nhằm thống nhất nhiều mô hình lập trình giao tiếp được hỗ trợ trong .NET 2.0 thành một mô hình duy nhất.

Là một mô hình lập trình cho phép nhà phát triển xây dựng các giải pháp dịch vụ đảm bảo tính ổn định, và bảo mật và thậm chí là đảm bảo giao dịch. Nó làm đơn giản hoá việc phát triển các ứng dụng nối kết và đưa ra cho nhà phát triển những giá trị mà có thể họ chưa nhận ra ngay, đó là cách tiếp cận phát triển hệ thống phân tán thống nhất, đơn giản, và quản lý được.

Kiến trúc chung của WCF, bao gồm:

* Contract: định nghĩa các phương thức, thành phần và cách thức hoạt động của service.
* Service Runtime: chứa các hành xử sẽ xảy ra trong quá trình thực hiện của dịch vụ, như: điều khiển luồng, xử lý lỗi…
* Message: là tập hợp các channel với mỗi channel là một thành phần xử lý mesage theo một cách nào đó. Có 2 channel chính: transport channel và control channel.
* Activation and Hosting.

Mục đích của WCF là cung cấp một mô hình lập trình duy nhất có thể được sử dụng để tạo ra các dịch vụ trên nền tảng .NET cho các tổ chức.

* 1. Quy trình xây dựng một web service

1. Thiết kế chương trình
   1. Tổng quan về ứng dụng

Mua bán chứng khoán được xem là một hoạt động mang tính rủi ro cao. Tuy nhiên, nó vẫn có sức hấp dẫn lớn đối với giới đầu tư vì tỷ suất sinh lợi khổng lồ mà hiếm thấy phương án sử dụng vốn nào có thể đạt được nếu như biết lựa chọn thời điểm ra vào thị trường. Hiện nay, cùng với sự phát triển của thị trường chứng khoán tại Việt Nam, nhu cầu sử dụng công cụ phân tích dự báo thị trường để tìm hiểu giá trị thực sự của cổ phiếu, xu hướng biến động giá… cũng ngày một tăng cao. Trước thực tế đó, nhóm đã quyết định nghiên cứu và phát triển **Hệ thốngcung cấp thông tin phân tích và dự báo thị trường chứng khoán** dựa trên nền tảng công nghệ **WCF**. Mọi dữ liệu về chứng khoán sẽ được tập trung và xử lý. Người dùng chỉ cần tra cứu và xem kết quả trực tuyến.

Hệ thống phân tích và dự báo thị trường chứng khoán được xây dựng như là một **Web Service** để cung cấp những thông tin cần thiết và hữu ích về các loại mã chứng khoán. Dựa trên thông tin đầu vào là báo cáo tài chính, giá cổ phiếu niêm yết, hệ thống sẽ tính toán các thông số dẫn xuất thể hiện sức mạnh tài chính, khả năng sinh lợi nhuận của công ty; xu hướng giá trên thị trường và tín hiệu mua / bán; đưa ra mô hình để dự báo giá cổ phiếu trong các phiên giao dịch sắp tới. Người dùng khi truy cập vào **Web Service** có thể xem các thông tin tài chính về công ty và sử dụng các công cụ được hệ thống hỗ trợ để tìm hiểu và phân tích thông tin về các mã chứng khoán.

Cấu trúc của hệ thống được phân ra làm ba thành phần chính:

**Phân tích cơ bản:** phân tích và tính toán các chỉ số trong báo cáo tài chính.

**Phân tích kỹ thuật:** cung cấp thông tin về giá, khớp lệnh.

**Dự báo:** đưa ra mức giá tham khảo cho các phiên giao dịch sắp tới theo mô hình ARIMA.

* 1. Các chức năng chính trong ứng dụng
     1. Nhóm chức năng Financial Analysis Service
        1. FinancialIndexService
* Chức năng

Cung cấp các chỉ số tài chính của công ty tương ứng với mã cổ phiếu.

* Tình huống sử dụng

Được gọi khi người dùng yêu cầu các thông tin liên quan đến thông số trong báo cáo tài chính và các chỉ số thể hiện tình trạng tài chính của công ty để hỗ trợ việc ra quyết định đầu tư đối với mã cổ phiếu tương ứng.

* Contract

GetTheBasicIndicators:

* Chức năng: lấy những chỉ số cơ bản như ROA, ROE, EPS, đòn bẩy tài chính…
* Input: mã cổ phiếu (string), quý cần xem (string)
* Data Contract

|  |  |  |
| --- | --- | --- |
| **FIItem** | | |
| ***Tên*** | ***Kiểu dữ liệu*** | ***Ví dụ*** |
| Name | string | Đòn bẩy tài chính |
| Value | double | 1.62 |

* Output: danh sách các FIItem *(List<FIItem>).*
* Binding: basicHttpBinding.

GetFinanceReport:

* Chức năng: cung cấp các chỉ số tài chính quan trọng được trích xuất từ báo cáo tài chính gốc của công ty.
* Input: mã cổ phiếu (string), mảng các quý cần hiển thị (List<Period>, với Period là đối tượng bao gồm thứ tự quý và năm.
* Data Contract

|  |  |  |
| --- | --- | --- |
| **FinancialRecord** | | |
| ***Tên*** | ***Kiểu dữ liệu*** | ***Ví dụ*** |
| Title | string | Nợ phải trả |
| Value | List<double> | 841926, 68642, 677558, 34602 |

* Output: trả về danh sách các Financial Record *(List<FinancialRecord>).*
* Binding: basicHttpBinding.
  + - 1. RecommendStockService
* Chức năng

Xếp hạng và đưa ra các mã cổ phiếu khuyến nghị để người dùng có thể xem xét đưa vào danh mục đầu tư.

* Tình huống sử dụng

Được dùng khi có yêu cầu xếp hạng công ty hoặc đưa ra mã có những tính chất tương tự mã cổ phiếu đang xét.

* Contract

GetTopFiveStock:

* Chức năng: xếp hạng các công ty có chỉ số tốt nhất, bao gồm: EPS, P/E, vốn hóa thị trường, ROE, ROA, Quick Ratio, đòn bẩy tài chính.
* Input: mã cổ phiếu (string), biến Boolean cho biết có muốn xem chi tiết cách xếp hạng hay chỉ xem kết quả tổng hợp (total score).
* Data Contract

|  |  |  |
| --- | --- | --- |
| **CompItem** | | |
| ***Tên*** | ***Kiểu dữ liệu*** | ***Ví dụ*** |
| Company | string | Công ty cổ phần sữa Hà Nội |
| Stock | string | HNM |
| Indicators | List<FIItem> | EPS, P/E… |
| Total Score | double | 0.78 |

* Output: danh sách các StockItem *(List<CompItem>)* theo thứ tự giảm dần.
* Binding: basicHttpBinding.

GetRecommendStock:

* Chức năng: đưa ra các mã cổ phiếu khuyến nghị có chỉ số xấp xỉ với một mã cổ phiếu nhất định.
* Input: mã cổ phiếu (string), biến Boolean cho biết có muốn xem chi tiết so sánh giữa tập chỉ số của các mã cổ phiếu hay chỉ xem kết quả so sánh tổng hợp (similarity).
* Data Contract

|  |  |  |
| --- | --- | --- |
| **StockItem** | | |
| ***Tên*** | ***Kiểu dữ liệu*** | ***Ví dụ*** |
| Stock | string | HNM |
| Indicators | List<FIItem> | EPS, P/E… |
| Similarity | double | 0.78 |

* Output: StockItem chuẩn để đưa ra khuyến nghị và danh sách 4 StockItem *(List<StockItem>)* tương tự nhất.
* Binding: basicHttpBinding.
  + 1. Nhóm chức năng Technical Analysis Service
       1. PriceService
* Chức năng

Cung cấp các thông tin về giá cả và khối lượng giao dịch của cổ phiếu.

* Tình huống sử dụng

Được sử dụng khi người dùng yêu cầu các thông tin liên quan đến giá, khối lượng giao dịch của các cổ phiếu hoặc muốn tìm các cổ phiếu đang được quan tâm nhiều nhất.

* Contract

GetPrice:

* Chức năng: lấy giá và khối lượng giao dịch của một cổ phiếu cụ thể.
* Input: mã cổ phiếu (string), số ngày giao dịch cần lấy giá kể từ ngày hiện tại (int).
* Output: danh sách giá, khối lượng giao dịch của cổ phiếu theo ngày *(List<string[]>).*
* Binding: basicHttpBinding.

GetMostIncrease:

* Chức năng: tìm các cổ phiếu tăng giá nhiều nhất sau khi kết thúc một phiên giao dịch.
* Input: số cổ phiếu cần lấy (int).
* Output: danh sách cổ phiếu được sắp xếp giảm dần theo tỉ lệ thay đổi của giá đóng cửa *(List<string[]>)*.
* Binding: basicHttpBinding.

GetMostDecrease:

* Chức năng: tìm các cổ phiếu giảm giá nhiều nhất sau khi kết thúc một phiên giao dịch.
* Input: số cổ phiếu cần lấy (int).
* Output: danh sách cổ phiếu được sắp xếp tăng dần theo tỉ lệ thay đổi của giá đóng cửa *(List<string[]>).*
* Binding: basicHttpBinding.
* GetMostTraded:
* Chức năng: tìm các cổ phiếu có khối lượng giao dịch nhiều nhất sau khi kết thúc một phiên giao dịch.
* Input: số cổ phiếu cần lấy (int).
* Output: danh sách cổ phiếu được sắp xếp giảm dần theo khối lượng giao dịch *(List<string[]>).*
* Binding: basicHttpBinding.
* GetSupportResistance:
* Chức năng: đưa ra thông tin về ngưỡng hỗ trợ và ngưỡng kháng cự của mã cổ phiếu.
* Input: mã cổ phiếu(string), số tuần (int); được mặc định là 52, tương ứng với 1 năm.
* Output: giá trị ngưỡng hỗ trợ và ngưỡng kháng cự.
* Binding: basicHttpBinding.
  + - 1. NewsService
* Chức năng

Cung cấp tin tức về các loại cổ phiếu, tin tức thị trường và các nhận định phân tích của chuyên gia.

* Tình huống sử dụng

Được sử dụng khi người dùng yêu cầu các thông tin về toàn cảnh thị trường, thông tin về một cổ phiếu nào đó hay muốn xem các bài phân tích của các chuyên gia kinh tế để tham khảo.

Được sử dụng để cung cấp giá và khối lượng giao dịch của một loại cổ phiếu cho TechicalAnalysisService.

* Contract

GetNews:

* Chức năng: cung cấp các tin tức chung của thị trường hoặc tin tức liên quan đến một cổ phiếu cụ thể.
* Input: số tin cần lấy (int).
* Output: danh sách tin tức được sắp xếp giảm dần theo thời gian *(List<string[]>).*
* Binding: basicHttpBinding.

GetAnalysis:

* Chức năng: cung cấp các bài phân tích của các chuyên gia kinh tế.
* Input: số bài phân tích cần lấy (int).
* Output: danh sách các bài phân tích được sắp xếp giảm dần theo thứ tự thời gian *(List<string[]>).*
* Binding: basicHttpBinding.
  + - 1. TechnicalAnalysisService
* Chức năng

Cung cấp hình vẽ biểu đồ thể hiện giá và khối lượng giao dịch của cổ phiếu cho người dùng. Người dùng có thể chọn 1 trong 3 dạng biểu đồ sau để phù hợp với nhu cầu sử dụng:

* Biều đồ đường thể hiện giá đóng cửa của cổ phiếu.
* Biểu đồ OHLC thể hiện giá mở cửa, high, low, giáđóng cửa của cổ phiếu.
* Biểu đồ Candlestick thể hiện giá mở cửa, high, low, giáđóng cửa của cổ phiếu.

Cung cấp các công cụ phân tích kỹ thuật trên biểu đồ: các công cụ này được vẽ chung với biểu đồ giá khi có yêu cầu. Người dùng có thể chọn 1 trong 9 loại công cụ sau đây:

* Trung bình trượt giản đơn, trung bình trượt cấp số nhân: là một công cụ làm trơn chuỗi dữ liệu giúp việc xác định xu hướng dễ dàng hơn.
* Giải biên độ biến động giá Bollinger (Bollinger Bands): dùng để đo độ biến động của giá. Nó gồm 3 đường. Đường trung tâm là đường trung bình trượt. Thêm hoặc bớt độ lệch chuẩn của chuỗi giá trị vào đường trung bình trượt ta được biên trên và biên dưới.
* Tỉ lệ thay đổi (Rate of change): là một chỉ số thể hiện khoảng chênh lệch của giá hiện tại với giá x ngày trước. Nó giúp xác định các điểm bán quá mức và mua quá mức.
* Trung bình trượt phân kỳ và hội tụ (Moving average convergence divergence): là chỉ số dao động giá đơn giản nhất và hiệu quả nhất. Được tính bằng hiệu của 2 đường trung bình trượt nên nó đồng thời mang tính chất của các chỉ số xu hướng giá và các chỉ số dao động giá.
* Chỉ số sức mạnh tương đối (Relative strength index): là chỉ số dao động giá dùng để đo vận tốc và mức độ thay đổi của giá. Nó là một trong những chỉ số dao động giá phổ biến nhất do có nhiều ưu điểm hơn các chỉ số dao động giá khác.
* Chỉ số lưu lượng tiền tệ (Money flow index): chỉ số này có thể xem như tương đương với chỉ số sức mạnh tương đối nhưng có tính thêm khối lượng giao dịch. Nhiều chuyên gia cho rằng khối lượng giao dịch thể hiện trước những biến động của giá do đó chỉ số lưu lượng tiền tệ sẽ đi trước diễn biến của giá.
* Chỉ số stochastic: là một chỉ số dao động giá thể hiện khoảng cách giữa mức giá hiện tại với mức giá cao nhất và thấp nhất trong một khoảng thời gian cho trước. Chỉ số stochastic được chia làm 2 loại: stochastic nhanh và stochastic chậm. Chỉ số stochastic chậm tạo các tín hiệu mua bán đáng tin cậy hơn chỉ số stochastic nhanh.
* Chỉ số Willams %R: chỉ số này là dạng đảo ngược của chỉ số stochastic do đó cách sử dụng cũng như những tín hiệu mua bán cũng tương tự như chỉ số stochastic.

Tình huống sử dụng

* Được sử dụng khi người dùng yêu cầu thể hiện giá cổ phiếu dưới dạng biểu đồ để có cái nhìn trực quan về diễn biến của giá.
* Được sử dụng khi người dùng muốn phân tích giá cổ phiếu theo hướng kỹ thuật để từ đó có thể chọn thời điểm tốt để thực hiện các giao dịch thích hợp.
* Contract

DrawChart:

* Chức năng: vẽ biểu đồ cổ phiếu từ các giá, khối lượng giao dịch cổ phiếu và các công cụ phân tích kỹ thuật được chọn.
* Input: danh sách dữ liệu cần vẽ lên biểu đồ *(List<List<string[]>>*).
* Output: hình vẽ biểu đồ giá cổ phiếu dưới dạng byte *(byte[]).*
* Binding: basicHttpBinding.
  + 1. Nhóm chức năng Forecast Service
* Chức năng

Cung cấp dự báo cho các phiên sắp tới của một mã chứng khoán.

* Tình huống sử dụng

Được gọi khi người dùng yêu cầu dự báo các phiên giao dịch trong tương lai để sử dụng với mục đích tham khảo.

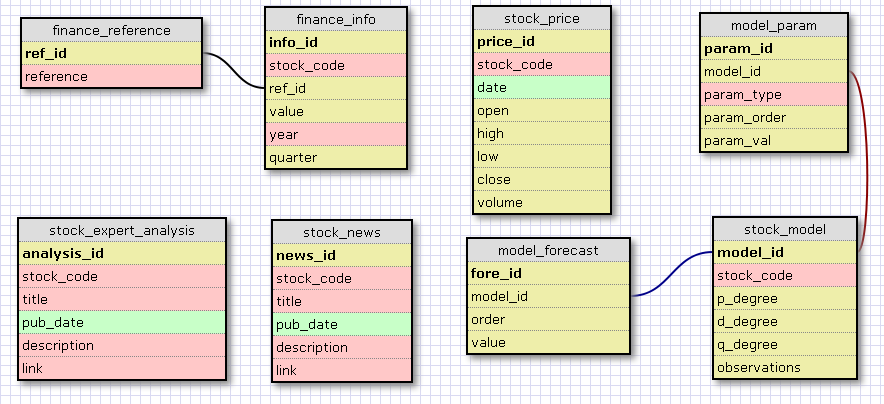
* Contract

GetForecast:

* Chức năng: đưa ra dự báo giá cho tương lai dựa vào cơ sở quy luật biến động của dữ liệu giá trong quá khứ.
* Input: mã cổ phiếu(int), số phiên tương lai cần dự báo(int).
* Data Contract

|  |  |  |
| --- | --- | --- |
| **ForecastItem** | | |
| ***Tên*** | ***Kiểu dữ liệu*** | ***Ví dụ*** |
| Stock | string | VNM |
| Forecast | List<double> | 136.53, 139.12, 142.00 |
| Limit50 | List<double> | 1.23, 2.38, 3.89 |
| Limit50 | List<double> | 3.23, 5.20, 7.09 |

* Output: danh sách dự báo của các phiên và khoảng tin cậy tương ứng
* Binding: Duplex.
  1. Thiết kế cơ sở dữ liệu



Hình – Cấu trúc cơ sở dữ liệu

Tất cả các bảng đều có khóa chính kiểu int tăng dần và có ý nghĩa như sau:

finance\_reference

Đây là bảng lưu trữ tên gọi các thông tin tài chính, bao gồm khóa chính và

|  |  |  |
| --- | --- | --- |
| ***Tên*** | Kiểu dữ liệu | Giải thích |
| reference | varchar (100) | Tên tham chiếu của thông tin tài chính |

finance\_info

Lưu trữ các giá trị tài chính cần thiết, có khóa ngoại liên kết đến bảng finance\_reference, bao gồm khóa chính và

|  |  |  |
| --- | --- | --- |
| ***Tên*** | Kiểu dữ liệu | Giải thích |
| stock\_code | varchar (5) | Mã chứng khoán của công ty |
| ref\_id | integer | Mã tham chiếu thông tin tài chính |
| value | interger | Giá trị |
| year | varchar (4) | Năm tài chính |
| quarter | integer | Quý trong năm |

***Lưu ý:*** bảng 1 và 2 có thể kết hợp lại thành một bảng duy nhất với mỗi cột đại diện cho một giá trị thông tin tài chính. Tuy nhiên cấu trúc như vậy sẽ gặp khó khăn trong việc chỉnh sửa các thông tin tài chính trong cơ sở dữ liệu. Do đó, nhóm thống nhất rằng sẽ lưu dưới dạng cặp name/value để dễ tùy chỉnh và nâng cấp. Thêm vào đó, bảng finance\_reference được thêm vào để giảm dữ liệu lặp.

stock\_news

Các tin tức có liên quan tới chứng khoán và công ty tương ứng sẽ được lưu trữ bằng bảng này, bao gồm khóa chính và

|  |  |  |
| --- | --- | --- |
| ***Tên*** | Kiểu dữ liệu | Giải thích |
| stock\_code | varchar (5) | Mã chứng khoán của công ty |
| pub\_date | date | Ngày đăng tin |
| title | varchar(50) | Tiêu đề bản tin |
| description | interger | Tóm tắt mẩu tin |
| link | varchar (4) | Liên kết tới trang gốc |

stock\_expert\_analysis

Bảng lưu trữ các phân tích và nhận định chung của chuyên gia về thị trường chứng khoán và các yếu tố ảnh hưởng trực tiếp, bao gồm khóa chính và

|  |  |  |
| --- | --- | --- |
| ***Tên*** | Kiểu dữ liệu | Giải thích |
| stock\_code | varchar (5) | Mã chứng khoán của công ty |
| pub\_date | date | Ngày đăng tin |
| title | varchar(50) | Tiêu đề bản tin |
| description | interger | Tóm tắt mẩu tin |
| link | varchar (4) | Liên kết tới trang gốc |

stock\_price

Lưu trữ thông tin giá cổ phiếu trong các phiên giao dịch, bao gồm khóa chính và

|  |  |  |
| --- | --- | --- |
| ***Tên*** | Kiểu dữ liệu | Giải thích |
| stock\_code | varchar (5) | Mã chứng khoán của công ty |
| date | date | Ngày giao dịch |
| open | decimal | Giá mở cửa |
| high | decimal | Giá trần |
| low | decimal | Giá sàn |
| close | decimal | Giá đóng cửa |
| volume | integer | Khối lượng giao dịch |

stock\_model

Lưu trữ thông tin về mô hình dự báo của mã cổ phiếu, bao gồm khóa chính và

|  |  |  |
| --- | --- | --- |
| ***Tên*** | Kiểu dữ liệu | Giải thích |
| stock\_code | varchar (5) | Mã chứng khoán của công ty |
| p\_degree | integer | Bậc p |
| d\_degree | integer | Bậc d |
| q\_degree | integer | Bậc q |
| observations | integer | Số quan sát được dùng để đưa ra mô hình |

***Lưu ý:*** Việc tính toán đưa ra mô hình dự báo cho một mã cổ phiếu cần nhiều phép tính và thời gian, thêm vào đó, đối với những chuỗi thời gian đủ dài thì ta có thể lập mô hình tương đối hiệu quả mà không cần cập nhật liên tục các giá trị mới. Với những lý do trên, việc lập mô hình dự báo sẽ được tính toán từ trước và lưu trữ vào cơ sở dữ liệu để tăng hiệu suất và tính tiện dụng cho hệ thống.

model\_param

Lưu trữ thông tin về tham số mô hình dự báo, bao gồm khóa chính và

|  |  |  |
| --- | --- | --- |
| ***Tên*** | Kiểu dữ liệu | Giải thích |
| model\_id | integer | Tham chiếu đến mô hình tương ứng |
| param\_type | integer | Kiểu tham số |
| param\_order | integer | Bậc tham số |
| param\_val | integer | Giá trị của tham số |

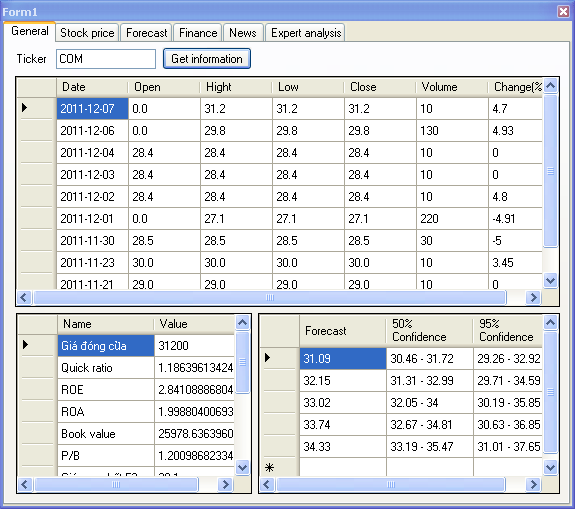
model\_forecast

Lưu trữ thông tin về giá trị dự báo dẫn xuất từ mô hình, bao gồm khóa chính và

|  |  |  |
| --- | --- | --- |
| ***Tên*** | Kiểu dữ liệu | Giải thích |
| model\_id | integer | Tham chiếu đến mô hình tương ứng |
| order | integer | Thứ tự dự báo |
| value | integer | Giá trị của dự báo |

***Lưu ý***: để tính toán sai số dùng trong mô hình dự báo, cần phải lưu lại kết quả dự báo từ những phiên trước. Khi mô hình được lập nên, giá trị dự báo cho 10 phiên sẽ được đưa vào bảng model\_forecast. Với 5 là mốc thời gian, thứ tự từ 1 – 5 là giá trị từ quá khứ cho đến hiện tại. Và từ 6 – 10 là các giá trị dự báo cho tương lai. Mỗi cuối ngày, giá trị dự báo sẽ tự động cập nhật theo phương pháp trượt (moving), cập nhật giá trị từ 2 – 6 cho 1 – 5, sau đó cập nhật dự báo cho giá trị từ 6 – 10. Cơ sở dữ liệu sẽ được giữ sao cho bất cứ lúc nào cũng chỉ có 10 giá trị dự báo tương ứng với một model.

1. Hiện thực
   1. Giao diện chính của ứng dụng

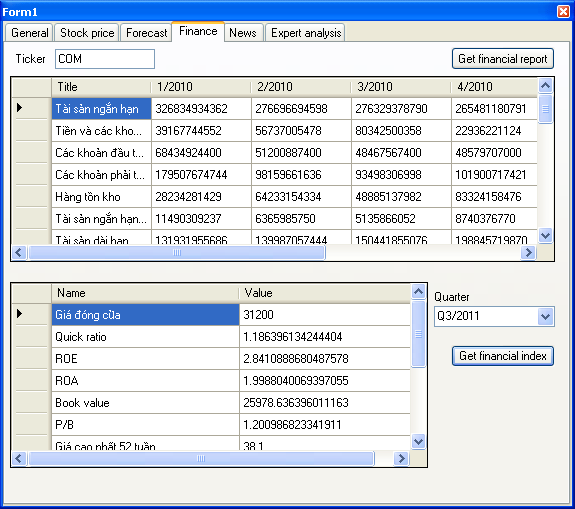


Hình – Giao diện tổng quan

* 1. Name standard

Dưới đây là một số quy tắc khi coding:

* Tất cả tên hàm, thuộc tính, class và interface trong Service đều mang tính miêu tả, tức là mang những từ khóa thể hiện chức năng do Service cung cấp. Tên được đặt bằng tiếng Anh.
* Đối với tên hàm hay tên Contract, viết in hoa chữ cái đầu của mỗi từ. Vd: hàm **GetChart()**.
* Khi có nhiều thuộc tính cùng tên nhưng khác loại có thể dùng kiểu dữ liệu để phân biệt. Vd: **strIndex** và **intIndex.**
* Đối với tên biến, thuộc tính, viết thường chữ cái đầu của từ đầu tiên và viết hoa chữ cái đầu cho các chữ tiếp theo (*camel convention*). Vd: biến **indexValue**.
* Service Contract định nghĩa bằng interface và đánh dấu bằng chữ I ở đầu. Vd: **IStockService**.
* Hàm lấy thông tin, thêm prefix Get ở đầu. Vd: **GetAuthor()**.
* Hàm điều chỉnh giá trị, thêm prefix Set ở đầu. Vd: **SetIndex()**.
* Tên hàm là hoặc bắt đầu với động từ. Vd: **DoSomething()**.
* Nếu viết callback function cho một hàm hoặc interface nào đó, thêm Callback vào phía sau tên hàm hoặc interface tương ứng. Vd: hàm callback cho hàm **GetChart()** sẽ là **GetChartCallback()**.
* Nếu viết trên form thì các đối tượng trong form cần đặt tên theo chức năng và kiểu control. Vd: button Cancel sẽ có tên là **btnCancel**.
  1. Hiện thực các chức năng
     1. Nhóm chức năng Financial Analysis Service
        1. GetTheBasicIndicators



Hình – Xem chỉ số và báo cáo tài chính

Các chỉ số cơ bản cần thiết bao gồm: quick ratio, ROE, ROA, book value, P/B, giá cao nhất trong 52 tuần, giá thấp nhất trong 52 tuần, EPS cơ bản, P/E, beta, đòn bẩy tài chính và hệ số biên lợi nhuận ròng.

Để có thể thực hiện tính được các chỉ số cơ bản trên, chúng ta cần phải xác định được các chỉ số liên quan được dẫn xuất từ báo cáo tài chính, dưới đây là bảng công thức (dòng thứ hai trong cột công thức tính thể hiện những ký hiệu/ hàm tương ứng trong cơ sở dữ liệu và code):

|  |  |
| --- | --- |
| **Chỉ số cơ bản** | **Công thức tính** |
| Quick ratio | = (Tổng tài sản lưu động và đầu tư ngắn hạn - Tổng hàng tồn kho)/ Tổng nợ ngắn hạn  = (Fi\_100 – Fi\_140) / Fi\_310 |
| ROE | = Tổng lợi nhuận sau thuế / Vốn chủ sở hữu  = Fi\_60 / Fi\_400 |
| ROA | = Tổng lợi nhuận sau thuế / Tổng tài sản  = Fi\_60 / (Fi\_100 + Fi\_200) |
| Book value | = (Vốn chủ sở hữu – Tài sản cố định vô hình) / KLCPLH hiện tại  = (Fi\_400 – Fi\_227) / Fi\_TOS |
| P/B | =Giá đóng cửa / Book value  = ClosePrice / BookValue |
| Giá cao nhất 52 tuần | = Tính bằng hàm MAX của dữ liệu giá thấp nhất của 52 tuần qua  = GetTheHighestClosePrice\_52W(stockCode) |
| Giá thấp nhất 52 tuần | = Tính bằng hàm MIN của dữ liệu giá thấp nhất của 52 tuần qua  = GetTheLowestClosePrice\_52W(stockCode) |
| EPS cơ bản | = Tổng lợi nhuận sau thuế / Tổng số cổ phiếu lưu hành  = GetSumProfitAfterTax(stockCode, currentPeriod) / Fi\_TOS |
| P/E | = Giá đóng cửa / EPS cơ bản  = ClosePrice / EPSBasic |
| Beta | = Hiệp phương sai của tỷ suất sinh lời của chứng khoán và tỷ suất sinh lời của thị trường / Phương sai của tỷ suất sinh lời của thị trường  = CalculateBetaValue(stockCode) |
| Đòn bẩy tài chính | = 1 + (Tổng nợ / Vốn chủ sở hữu)  = 1 + (Fi\_300 / Fi\_400) |
| Hệ số biên lợi nhuận ròng | = Lợi nhuận sau thuế / Doanh thu  = Fi\_60 / Fi\_1 |

Bảng – Công thức tính toán các chỉ số.

Những chỉ số cơ bản trên, sau khi được tính sẽ được lưu trữ trong đối tượng *List<FIItem>*, với mỗi chỉ số là một đối tượng bao gồm mã số (ID), tên (Name) và giá trị (Value).

Các phương thức khác liên quan (được hiện thực đầy đủ trong Methods.cs):

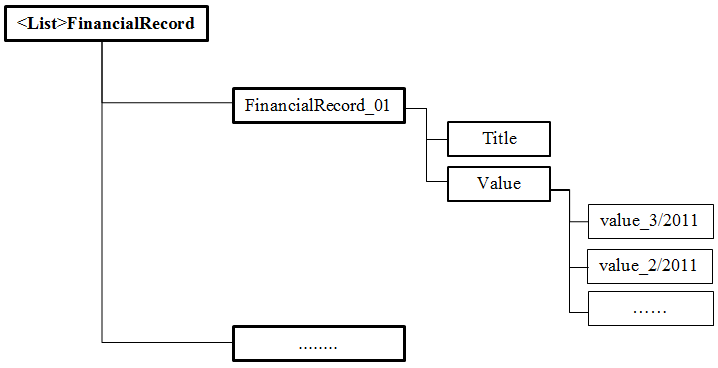
* GetBasicIndicator(stockCode, currentPeriod): lấy những chỉ số cơ bản của mã chứng khoán (stockCode) tại thời điểm nào đó (currentPeriod), để phục vụ cho việc tính toán những chỉ số cơ bản khác.
* DefineValue(**List<string[]> indicatorList, string fiID)**: xác định giá trị (Value) của một chỉ số được dẫn xuất từ báo cáo tài chính, với *indicatorList* là kết quả trả về của hàm *GetBasicIndicator*, *fiID* là mã số (ID) của chỉ số đó.
  + - 1. GetFinanceReport

Để có thể trình bày bảng chỉ số báo cáo tài chính có dạng dưới đây, chúng ta cần phải sử dụng đối tượng *List<FinancialRecord>* để quản lý các chỉ số báo cáo tài chính. Mỗi chỉ số là một đối tượng gồm tiêu đề(Title) và giá trị (Value), thuộc tính Value sẽ bao gồm nhiều value tương ứng với từng quý.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 3/2010 | 4/2010 | ….. |
| Tài sản ngắn hạn | 132950767955 | 110973965588 | ….. |
| ….. | ….. | ….. | ….. |

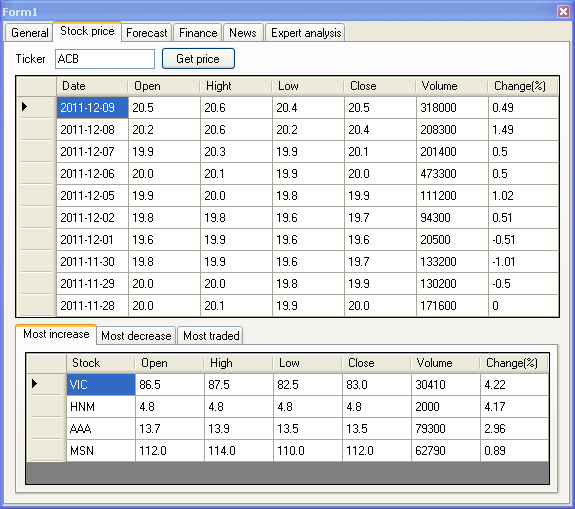
Bảng – Ví dụ cách trình bày bảng báo cáo tài chính.

Khi hiện thực, chúng ta sẽ kiểm tra tiêu đề (Title) của mỗi chỉ số đã được ghi vào *List<FinancialRecord>* hay chưa. Nếu chưa, title sẽ được ghi vào thuộc tính Title, và value sẽ được ghi vào vị trí thứ nhất của thuộc tính Value. Nếu Title đã được ghi nhận, chỉ tiến hành ghi các value vào vị trí thứ hai trở đi của thuộc tính Value. Mô hình dưới đây sẽ thể hiện cấu trúc phân cấp của *List<FinancialRecord>*.



Hình ­– Cấu trúc đối tượng List<FinancialRecord>

* + 1. Nhóm chức năng Technical Analysis Service
       1. GetPrice

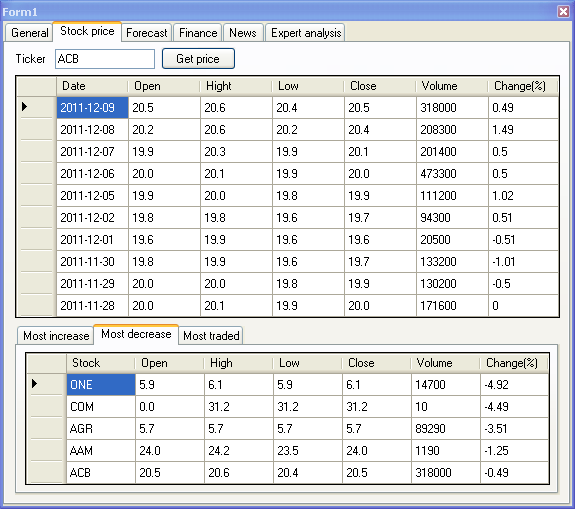


Hình – Xem giá cổ phiếu ACB

Do dữ liệu giá đã được lưu trong cơ sở dữ liệu nên chỉ cần thực hiện gọi query từ lớp kết nối cơ sỡ dữ liệu là ta có được danh sách giá đầy đủ của các cổ phiếu. Ngoài ra để bổ sung thêm thông tin hữu ích cho người dùng, tỉ lệ thay đổi của giá được tính thêm trong code và được bổ sung vào danh sách giá.

Kết quả trả về là một đối tượng *List<string[]>* chứa danh sách giá theo ngày của một cổ phiếu với một phần tử trong *List* tương đương với một dòng trong cơ sở dữ liệu và một phần tử trong *string[]* tương đương với dữ liệu của một cột trong cơ sở dữ liệu.

* + - 1. GetMostIncrease, GetMostDecrease



Hình – Xem các mã cổ phiếu giảm giá nhiều nhất

Để lấy được các cổ phiếu tăng và giảm giá nhiều nhất, ta cần tính tỉ lệ thay đổi của giá mới nhất của tất cả các cổ phiếu. Lọc ra danh sách các cổ phiếu tăng giá hoặc giảm giá. Sau đó sắp xếp danh sách này giảm dần hoặc tăng dần theo tỉ lệ thay đổi ta được các cổ phiếu tăng hoặc giảm giá nhiều nhất.

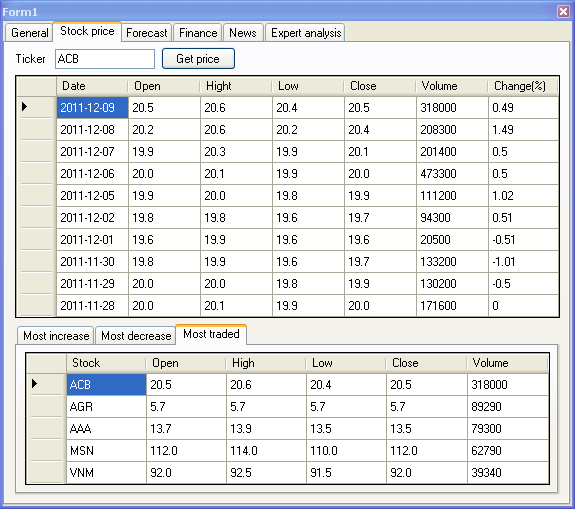
Do chỉ cần tính tỉ lệ thay đổi của giá mới nhất nên chỉ cần lấy 2 dữ liệu giá mới nhất của từng cổ phiếu. Để làm được điều này trước tiên ta phải lấy tất cả mã cổ phiếu bằng cách thực hiện gọi query select với từ khóa distinct từ lớp kết nối cơ sỡ dữ liệu. Khi đã có danh sách mã cổ phiếu ta tiếp tục thực hiện query để lấy 2 dữ liệu giá mới nhất với từng mã cổ phiếu. Sau đó thực hiện tính tỉ lệ thay đổi như bình thường.

Tiếp theo, ta loại các cổ phiếu có tỉ lệ thay đổi âm khi cần lấy các cổ phiếu tăng giá nhiều nhất hoặc loại các cổ phiếu có tỉ lệ thay đổi dương khi lấy các cổ phiếu giảm giá nhiều nhất.

Sau khi đã lọc ra danh sách các cổ phiếu tăng giá hoặc giảm giá, ta sắp xếp lại để được kết quả cuối cùng.

Kết quả trả về là một đối tượng *List<string[]>* chứa danh sách cổ phiếu với một phần tử trong *List* tương đương với một dòng trong cơ sở dữ liệu và một phần tử trong *string[]* tương đương với dữ liệu của một cột trong cơ sở dữ liệu.

* + - 1. GetMostTraded



Hình – Xem các mã cổ phiếu được giao dịch nhiều nhất

Do chỉ cần so sánh khối lượng giao dịch của các cổ phiếu với nhau nên ta chỉ cần thực hiện gọi câu query sau:

*select \* from*

*(select stock\_code, open, high, low, close, volume from stock\_price order by date desc) as temp*

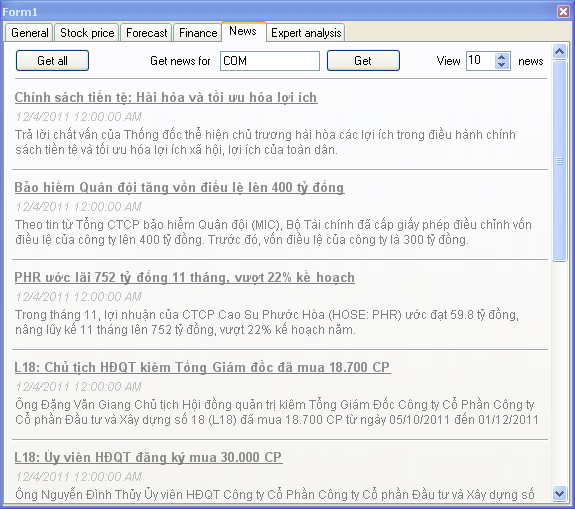
*where stock\_code not in ('^VNINDEX', '^HASTC') group by stock\_code*

*order by volume desc limit 5.*

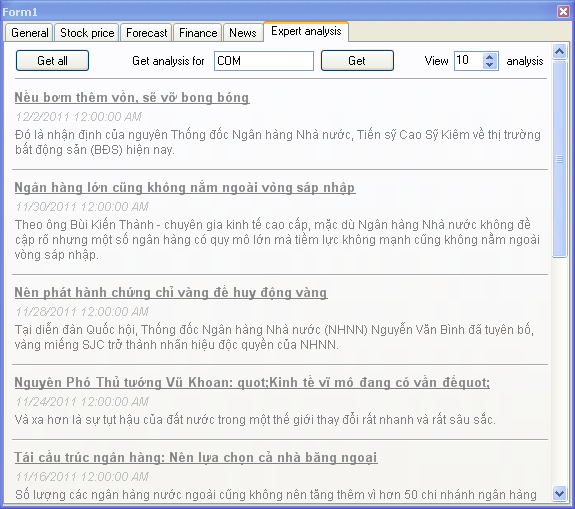
*Giải thích*: trong câu query trên, đầu tiên nó sẽ sắp xếp danh sách giá của toàn bộ các cổ phiếu theo ngày mới nhất. Tiếp theo, nó loại ra các dữ liệu của VNINDEX chung của thị trường. Sau đó khi sử dụng group by sẽ chọn ra được dữ liệu mới nhất của tất cả cổ phiếu. Sắp xếp giảm dần theo khối lượng giao dịch ta được kết quả cuối cùng.

Kết quả trả về là một đối tượng *List<string[]>* chứa danh sách cổ phiếu với một phần tử trong *List* tương đương với một dòng trong cơ sở dữ liệu và một phần tử trong *string[]* tương đương với dữ liệu của một cột trong cơ sở dữ liệu.

* + 1. Nhóm chức năng News



Hình – Xem tin tức

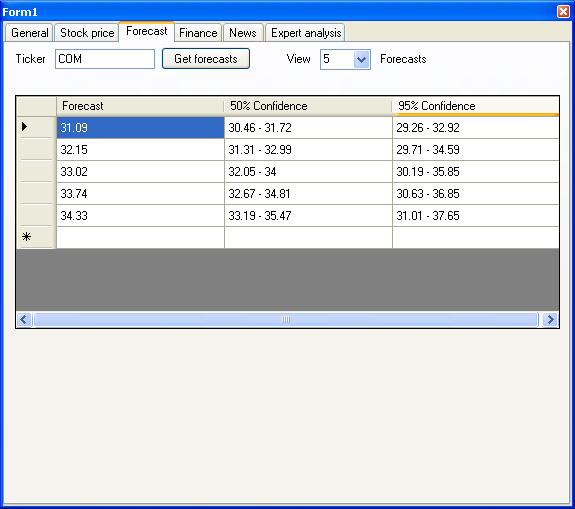


Hình – Xem phân tích của chuyên gia

Do dữ liệu về tin tức và các bài phân tích của chuyên gia đã được lưu trong cơ sở dữ liệu nên chỉ cần thực hiện gọi query từ lớp kết nối cơ sỡ dữ liệu, ta sẽ có được danh sách các tin hoặc các bài phân tích mới nhất.

Kết quả trả về là một đối tượng List<string[]> chứa danh sách tin tức hoặc các bài phân tích của chuyên gia với một phần tử trong List tương đương với một dòng trong cơ sở dữ liệu và một phần tử trong string[] tương đương với dữ liệu của một cột trong cơ sở dữ liệu.

* + 1. Nhóm chức năng Forecast Service
       1. GetForecast



Hình – Xem dự báo giá các phiên giao dịch sắp tới

Dữ liệu về giá cổ phiếu sẽ được truy vấn để lập mô hình dự báo ARIMA. Đây là một dạng mô hình dự báo chuỗi thời gian được ứng dụng rộng

rãi trong các lĩnh vực có liên quan đến kinh tế tài chính. Code lập mô hình và tính kết quả dự báo được viết bằng ngôn ngữ php. Các giá trị dự báo sẽ được lưu trữ trong cơ sở dữ liệu,hàm **GetForecast** chỉ lấy dữ liệu để thể hiện.

Kết quả trả về là một đối tượng List<Forecast> với mỗi dòng là dự báo cho một phiên giao dịch trong tương lai bao gồm giá trị dự báo, khoảng tin cậy 50% và khoảng tin cậy 95%.

Đối với chức năng này, tôi hiện thực theo kiểu duplex. Tức là client và server có thể liên lạc và gửi thông tin cho nhau một cách độc lập.

KẾT LUẬN

1. Đánh giá

Với sự phát triển như vũ bão của công nghệ internet hiện nay, Web Service đang dần chứng tỏ vị thế của mình. Việc sử dụng internet rộng rãi và băng thông ngày càng mở rộng đã khiến cho Web Service trở thành một công cụ giao tiếp lý tưởng đối với các nhà phát triển ứng dụng. Qua đồ án lần này, nhóm chúng tôi đã học được những kinh nghiệm về cách xử lý, phương thức kết nối, tính chất cơ bản của nền tảng WCF do Microsoft hỗ trợ hiện nay đang được ứng dụng rất phổ biến.

Đề tài được thực hiện theo đúng những gì mà nhóm đã đề ra ban đầu. Được thiết kế theo nguyên lý SOA, nhóm đã xác định rằng các service contract trong đề tài chủ yếu chỉ đóng vai trò như một lớp trung gian để gửi và nhận thông tin giữa client và server. Do đó, những nghiệp vụ về xử lý và thu thập thông tin liên quan đã được tách riêng ở một phần khác để giảm độ phụ thuộc giữa logic và dịch vụ cung cấp thông tin.

1. Hướng phát triển

Hiện thực việc gửi các biểu đồ phân tích kỹ thuật.

Phát triển chức năng tìm ra các mã cổ phiếu tương tự như mã người dùng quan tâm để khuyển nghị.

Hỗ trợ tìm kiếm mã cổ phiếu dựa trên các thông tin như, tên mã, tên công ty, giá, lợi nhuận trên một cổ phiếu…

Hiện thực fault contract cho các tình huống như: bộ nhớ quá tải, số lượng người truy cập vượt qua mức cho phép, mất kết nối…

TÀI LIỆU THAM KHẢO

[1] *Pro WCF Practical Microsoft SOA Implementation*

[2] “*Hosting Multiple Service Implementations on the Same Port with WCF*”, truy cập tại địa chỉ <http://blogs.msdn.com/b/dkaufman/archive/2008/06/13/hosting-multiple-service-implementation-on-the-same-port-with-wcf.aspx>

[3] “*Duplex Services*”, truy cập tại địa chỉ <http://msdn.microsoft.com/en-us/library/ms731064.aspx>

[4] “*WCF: Duplex Operations and UI Threads*”, truy cập tại địa chỉ <http://www.codeproject.com/KB/WCF/WCF_Duplex_UI_Threads.aspx>

[5] “*Duplex Contract in WCF and how to implement it?*”, truy cập tại địa chỉ <http://www.dotnetfunda.com/articles/article1195-duplex-contract-in-wcf-and-how-to-implement-it.aspx>

[6] “*Web Service*”, truy cập tại địa chỉ <http://en.wikipedia.org/wiki/Web_service>