ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

Journal of the Franklin Institute ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Journal of The Franklin Institute

www.elsevier.com/locate/jfranklin

M-stock: Efficient stock monitoring for mobile users

F. Aloul[a], A. Sagahyroon[a,*], N. Salem[a], O. Al-Omar[a], S. Shehada[a], B. Al-Rawi[b]

[a]*Department of Computer Science & Engineering, American University of Sharjah, UAE*
[b]*Microsoft Corp., Redmond, WA 98052, USA*

Received 30 May 2009; received in revised form 9 December 2009; accepted 5 February 2010

Abstract

The emergence of sophisticated and powerful mobile devices with impressive computing capabilities and storage capacity paves the way for an era of advanced mobile-based applications. Monitoring of time-critical data, process control, and security are possible candidate applications. In this work we propose a stock monitoring system that utilizes mobile phones. The system was designed to provide an alternative to costly GPRS connections for stock market traders. The system grants real-time monitoring of stock information through a user-friendly and easy to use GUI with plenty of useful features. The system was implemented by combining various Java technologies, specifically J2ME and J2EE. It was tested using the Dubai Financial Market with favorable results.

Keywords: Mobile; Stock-monitoring; GPRS; J2ME

1. Introduction

The rapid spread of mobile phones use inspires new applications that integrate hardware, software, and telecommunication technologies. Recently, these devices were utilized in a number of applications for monitoring and control purposes [1–5]. Mobile phones are becoming more powerful and sophisticated and have evolved to full-scale internet enabled computing devices. Smart phones' emerging capabilities are fueling a rise in the use of mobile phones as input devices to such resources as situated displays, vending machines, and home appliances. Mobile phones' prevalence gives them great potential to be the

*Corresponding author.
E-mail addresses: asagahyroon@aus.edu (A. Sagahyroon), bashar.alrawi@umich.edu (B. Al-Rawi).

0016-0032/$32.00 © 2010 Published by Elsevier Ltd. on behalf of The Franklin Institute.
doi:10.1016/j.jfranklin.2010.02.006

default physical interface for ubiquitous computing applications. This would provide the foundation for new interaction paradigms [6]. In recent years, mobile marketing (M-Marketing) has started to emerge and gain acceptance [7,8]. Yet again, this is made possible because of the explosive growth of mobile devices, coupled with advances in the information, communication fields, and mobile technologies. The advantages of having direct access to a device carried by consumers most of the time such as their cell phones are endless from a marketing point of view. Hence, one can argue that mobile phones are, and will continue to increasingly become an essential component of our daily life, and the ever increasing power of these devices will pave the way for the development of very sophisticated applications in the near future.

In this paper, we report on the design and implementation of a stock monitoring platform using mobile phones. In recent years, there has been a growing interest among individuals in stock trading, especially in emerging markets such as the Dubai Financial Market (DFM) [11]. A common practice in these markets is for customers to call brokerage firms seeking advice, requesting transaction execution, or simply to get the price of certain stock. Sometimes, users of these firms may wait for long period of time before being provided with access to a broker. Some firms allow the users to access stock information via the web. Despite the easiness of this approach, the user is restricted to places where an internet connection and a computer are available. Advanced mobile phones can connect to the internet via general packet radio service (GPRS), and hence mobile users can access the stock information at any time from anywhere (provided there is mobile service coverage). This approach also has several problems. First, not all websites can be displayed on mobile phone browsers. This can restrict the information retrieved by the user. Second, users are forced to download all the website content which is likely to include unused information. Third, stock prices are continuously changing and websites displaying stock information are always updated. Downloading large amount of stock data can be inconvenient for mobile users since GPRS is known to be both, slow and expensive.

Even if we assume that customers can have access to the internet from anywhere at any time using a GPRS connection, since the customer is charged per kbytes, the GPRS connection becomes very costly with every access to a website like Dubai's Financial Market with plenty of data to download. Additionally, with a normal GPRS connection, if the customer is interested in stock information from different stock markets, he/she has no other choice other than browsing through each website. This solution is costly, time consuming, and inconvenient; in addition to this, the customers' stocks of interest might be only few out of all the stocks available at the website. Note that some websites offer a customized *my stocks* page where individual preferences for stock watching and email alerts can be set up. However, these pages are not efficient for mobile phone browsing and will be costly and slow to download on a mobile phone.

Motivated by the aforementioned factors, we propose an optimized and efficient approach to stock monitoring using mobile phones. The primary goal is to provide a practical alternative to costly GPRS connections for users who use their mobiles to monitor the stock market. The proposed system has a user-friendly and easy to use graphical interface that allows users to create portfolios for their preferred stocks. The stocks selected can be from a single/multiple financial market(s). The user can then retrieve the selected portfolio, hence, downloading only few bytes constituting the required information about the selected stock. In turn, this process reduces the connection costs

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006

significantly, and provides a faster screen refresh rate, as the number of bytes downloaded is significantly reduced.

Within the context of using mobile phones in stock-related activities, Kargupta et al. [9] discussed an experimental mobile data mining system that allows intelligent monitoring of time-critical financial data from a hand held PDA. Their paper described the data mining component of the system that employs a Fourier analysis based approach to represent, visualize, and communicate decision trees over wireless networks. In [10], the authors experimented with the use of non-speech sounds to present dynamic information about share prices without using the visual display space of mobile devices.

Several commercial tools have recently released cost-efficient and easy-to-use stock monitoring product for mobile phones. Examples include Siemens [22], Axmor [20], and Pocket-X [19]. These products are released as black boxes without explanation of the implementation details. They are also restricted to limited specific phones, require a significant time to process, and cost more than our proposed solution. It is worth mentioning that, the Siemens' product was a Java-based application that integrated with only the OnVista Finance Portal and worked for the S55 mobile phone [22]. As far as we know, this paper is the first academic article that clearly describes the process of implementing an efficient stock monitoring system for mobile users.

The rest of this paper is organized as follows: in Section 2, we discuss the system layout and its various components, and the underlying implementation methodology; in the following section, we guide the reader through an example of a typical communication scenario between the system entities; system features are presented in Section 4, while Section 5 discusses the experimental results. The paper is concluded in Section 6.

2. System architecture

This section covers details pertaining to the overall system connections and system components. It provides a view of the general layout of the design; thereafter it introduces the features provided by the application program. The following subsection discusses the different components of the system, their implementation, and how they interoperate and communicate. The protocols used to establish the achieved design objectives are discussed in subsequent sections. The proposed system layout is depicted in Fig. 1.

As shown in Fig. 1, the *web server* is connected to the internet via a fast internet connection (e.g. ADSL). The speed of the internet connection determines the speed of service. The server uses the internet connection for two primary tasks:

1. connect to stock market website, which is accomplished using the web services provided by the stock markets [12]. The server then parses the websites and stores the stock information in a local database;
2. respond to user requests.

The *mobile* application, i.e. client, connects to the server through an agreed upon protocol that grants an efficient, robust, and secure communication. A secure connection is achieved using the HTTPS protocol. Using HTTPS guarantees that information exchange between the two parties is protected against any malicious intrusions. Robustness and efficiency are achieved using an established protocol explained in a later section.

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006

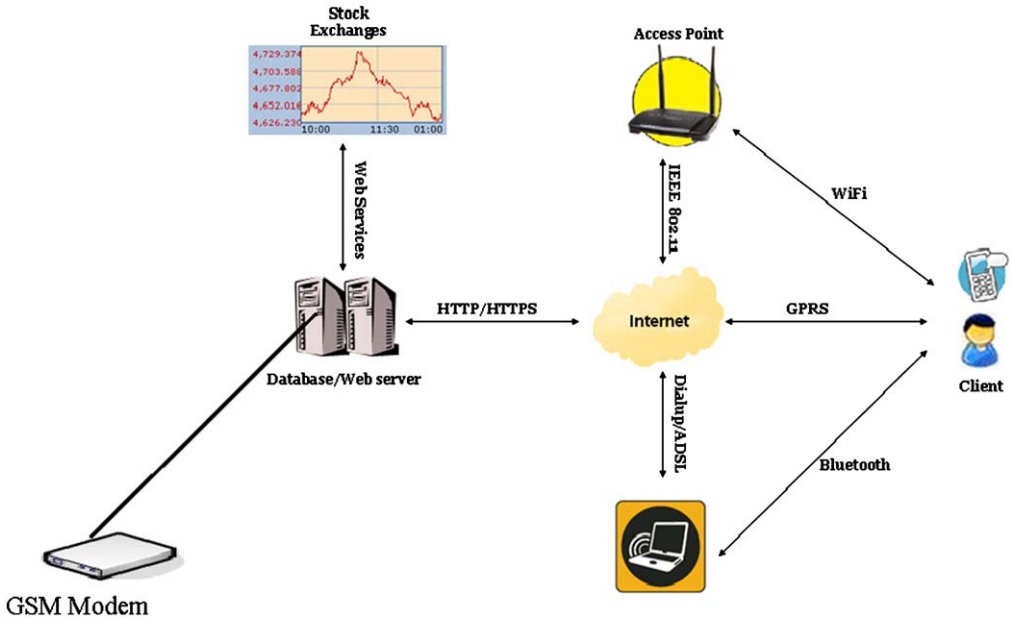

Fig. 1. Architecture or layout of the proposed system.

The design methodology adopted in this work complies with Sun Microsystems recommended three-tier architecture for enterprise computing: (1) mobile phones representing the client tier, (2) a server that represents the middle tier, and (3) services representing the back-end tier.

System features include the following:

- An easy to use GUI with the capability of creating/storing/editing/deleting portfolios of users' preferred stocks;
- Portfolio information retrieval is possible through:
  - *GPRS:* The connection cost in this case is reduced to a minimum, since only those few bytes requested by the user will be downloaded to the mobile phone;
  - *Bluetooth:* The mobile application connects to the internet through Bluetooth with the aid of special software downloaded to the mobile phone and the Bluetooth server. The connection cost is totally eliminated;
  - *WiFi:* A feature available for smart/3G mobile phones having a WiFi adapter. The application in this case directly routes the connection through WiFi, hence totally eliminating the cost;
- User can set alerts and receive an SMS when a stock price hits a specified target;
- User can retrieve a chart for selected stock containing stock information for the past 7 days;
- Information retrieval is automatically updated according to a refresh rate in minutes set by the user;
- A secured connection using HTTPS protocol to prevent information fraud.

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006

2.1. System components

The proposed system consists of three major components. A server, a client running an application developed for the project at hand, and the database.

2.1.1. The server

The server functions were implemented using PHP scripts. PHP is a server-side scripting language used for the creation of dynamic webpages [13]. When PHP scripts are invoked by an HTTP request, the script will be executed, displayed and then destroyed. When a visitor or user requests a PHP page, the server processes the PHP commands and then sends the results to the visitor's browser, just as with Active Server Pages (ASP). Unlike Windows ASP, however, PHP is an open source and cross-platform. PHP runs on Windows NT and many Unix versions, and it can be built as an Apache module and as a binary that can run as a Common Gateway Interface (CGI). When built as an Apache module, PHP is especially lightweight and speedy. Without any process creation overhead, it can return results quickly. In this implementation, the default PHP libraries were used in addition to the JP-Graph add-on libraries. The combination of the default libraries and the libraries from JP-Graph was sufficient to implement all the functions in the protocol sequences used here.

The main tasks of the PHP server are summarized below:

- retrieve, process, and insert Dubai Financial Market information in the MySQL database;
- listen to requests from mobile devices and engage in commencing the operations requested by the users;
- store user profiles upon arrival of user request to create new profile and creating one;
- send available financial markets upon arrival of user request;
- send available companies within the selected market upon arrival of user request;
- send available stock details (price high, low, bid, offer, etc.) upon arrival of user request;
- pull the portfolio information and send it back to the user upon arrival of request;
- continuously save stock information for past 7 days for all companies;
- plot charts for the selected stock dynamically upon arrival of user request using stored information;
- store alerts information in the database upon arrival of user request.

2.1.2. The client

The application running on the client side was developed and implemented using Java 2 micro-edition (J2ME) [14,15]. Essentially, the application consists of the following classes: a Main Midlet Class, a RecordStore Class, a CreateNewProfile Class, and a Connection Class. These application classes interoperate to provide the user with a friendly, easy to use, and practical tool for performing various stock market related activities. A brief description of the functionality and role of each class is provided below:

- *Main Midlet*
  Java applications that run on Mobile Information Device Profile (MIDP) devices are known as Midlets. The Midlet is the heart of any MIDP application. Any J2ME application typically contains only one main Midlet that in turn contains three basic functions: *StartApp( )* or a start application function, a *PauseApp( )* or pause

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006

application function, and a *DestryApp( )* or destroy application function. The first function, when invoked changes the Midlet status from pause to active; the second function role is to release all the resources held by the application and to save the current state; and the last function releases all the resources of the application and makes sure that all threads running in the background are terminated successfully.

- *RecordStore Class*
  The RecordStore Class makes use of the javax.micro-edition.rms package. Record Management Store (RMS) is a persistence package that provides the application with a mechanism for persistent storage of data. Hence, this class is used for persistent storage of user portfolios and alerts. The application has two record stores; one for the portfolios and the other for alerts. A record store is uniquely identified by its name. The javax.micro-edition.rms package has a set of functions to add/modify/delete records from the record store. We made use of these functions to create/edit/delete portfolios/alerts and store them on the user's device.
- *CreateNewProfile Class*
  The CreateNewProfile Class acts as the interface between the user and the RecordStore Class. The class guides the users through a set of GUIs while capturing the entered information. After this class gathers and formats user's input, it then invokes the appropriate functions to carry or perform the required activities.
- *Connection Class*
  As its name indicates, the Connection Class has all the functions needed to establish the connection with the server. The Connection Class contains functions that receive the formulated URL string from the CreateNewProfile Class to carry out the action specified by the URL. URL format will be further illustrated in the upcoming sections.

2.1.3. The database

The server uses MySQL database. MySQL is an open source relational database management system which uses Structured Query Language (SQL). MySQL was chosen because of its reliability, speed and flexibility. The server, therefore, first receives the URL from the application, tokenizes the user requests, and issues the appropriate SQL statement to perform the required action and store information accordingly in the database.

3. Communication scenarios (client/server interaction)

User requests are delivered to the server using either wireless LAN network, Bluetooth (BT), or GPRS. In case of BT/WiFi, the request is routed to the ADSL internet connection where it reaches the server. Basically, these user requests are URLs that are formulated based on a predefined protocol that acts on minimizing the traffic sent back and forth between the mobile and the server; hence, the connection costs are significantly reduced. Upon the arrival of a user request, the server tokenizes it to retrieve the requested parameters. The server formulates the appropriate SQL query that performs the action implied by the request. If the URL requested by the user requires a response, the server streams back the requested information as per retrieved from the database in the form of a dynamically generated PHP page. The mobile device then is responsible for reading the PHP page stream and displaying it in a proper format. For the requests that do not require any response, for instance creating a portfolio, the SQL query is formulated merely to insert the URL processed information

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006

into the database as required. Table 1 shows the protocol or sequence of events that take place depending on the URL request issued, and Fig. 2 illustrates an exchange of packetized information for a user requesting the creation of a portfolio.

Table 1
Description of the exchange protocol.

URL request	Action
act = 1	Retrieve markets
act = 2&mkt = market_name	Retrieve company names for the market specified by market_name string
act = 3	Retrieve prices
act = 4&id = x	Retrieve profile information of profile specified by id
act = 5&id = x&st = stocks&inf = prices	Create a profile: St → stocks selected Inf → prices selected
act = 6&id = x&st = stocks&inf = prices	Edit profile
act = 7&id = x	Delete profile having id x
act = 8&st = DFM_EMMAR&PR = 5.55&noti = "0 or 1"&TEL = 050xxxxxxx	Set SMS alert
act = 9&id = x	Delete alert
act = 10&st = DFM_ACICO+	Get chart

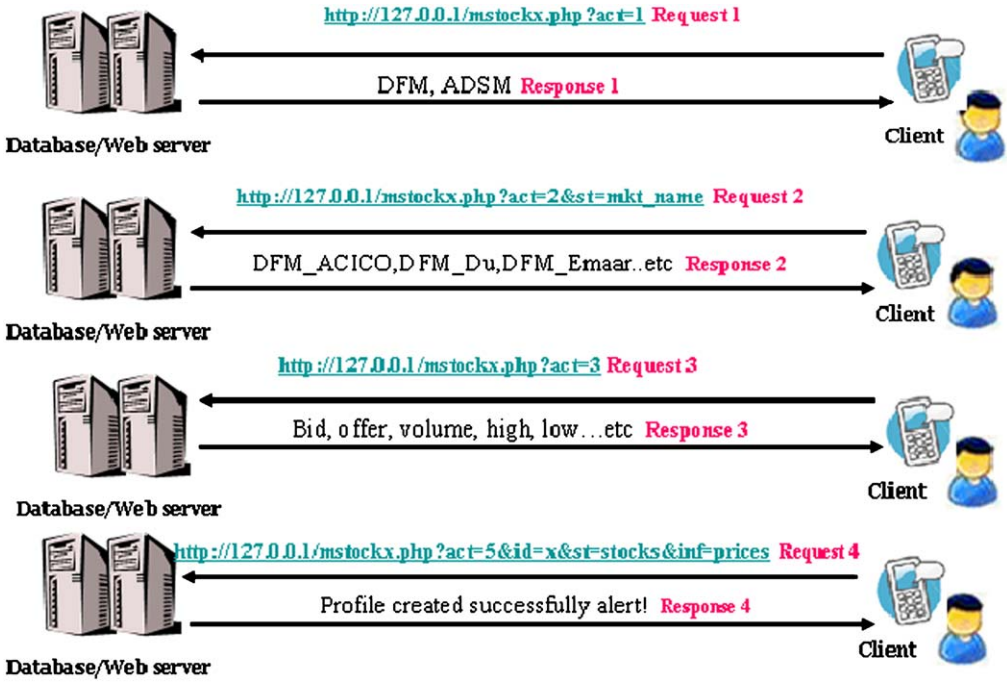

Fig. 2. An example illustrating server/client packetized data exchange for a portfolio creation request.

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006

4. User's interface

The application features hierarchy of our system is shown in Fig. 3.

For the sake of brevity only two features will be discussed in detail. Also the function of some features is self-explanatory from the features' names listed in Fig. 3.

4.1. The alert feature

A Siemens tc35i GSM modem is used to implement the SMS Alerts Feature. The modem is connected to the web server. The GSM modem was programmed to send the SMS alerts using a C# application [16,17]. We chose C# language as it contains many useful libraries like:

- **System.Timers:** used for scheduling the tasks and running the application in the background 24/7.
- **System.IO.Ports:** for creating serial ports and listening to the ports from any incoming or received information coming through the serial ports (GSM modem is connected to port com1).
- **MySql.Data.MySqlClient:** used to open a connection to the MySQL database (server database), and retrieve the needed information by executing SQL queries.

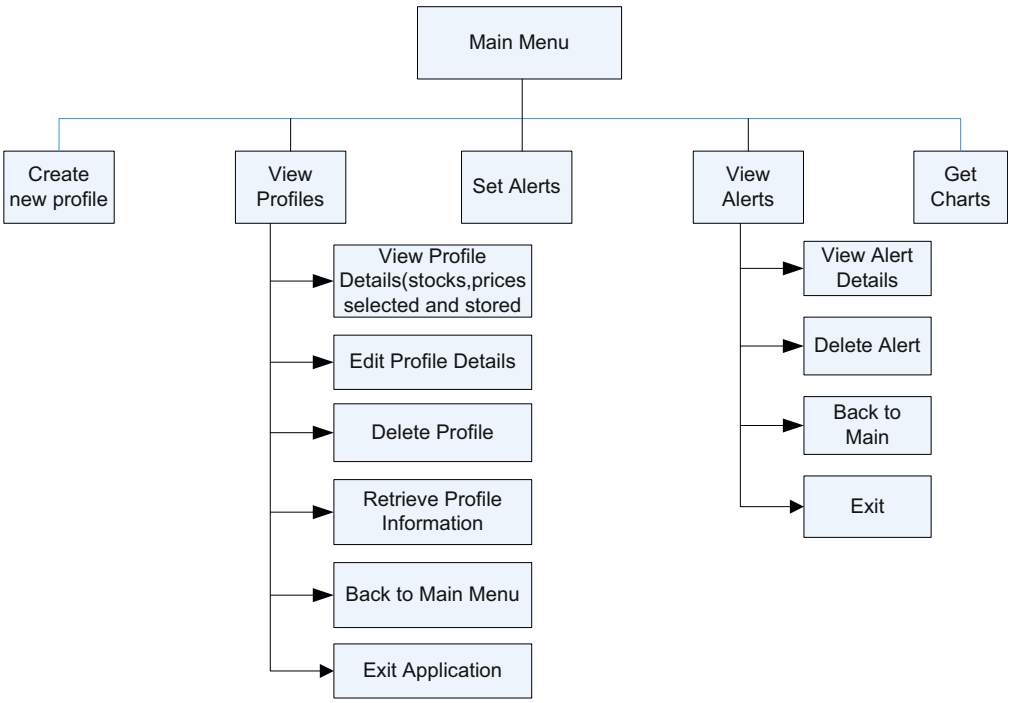

Fig. 3. Features menu.

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006

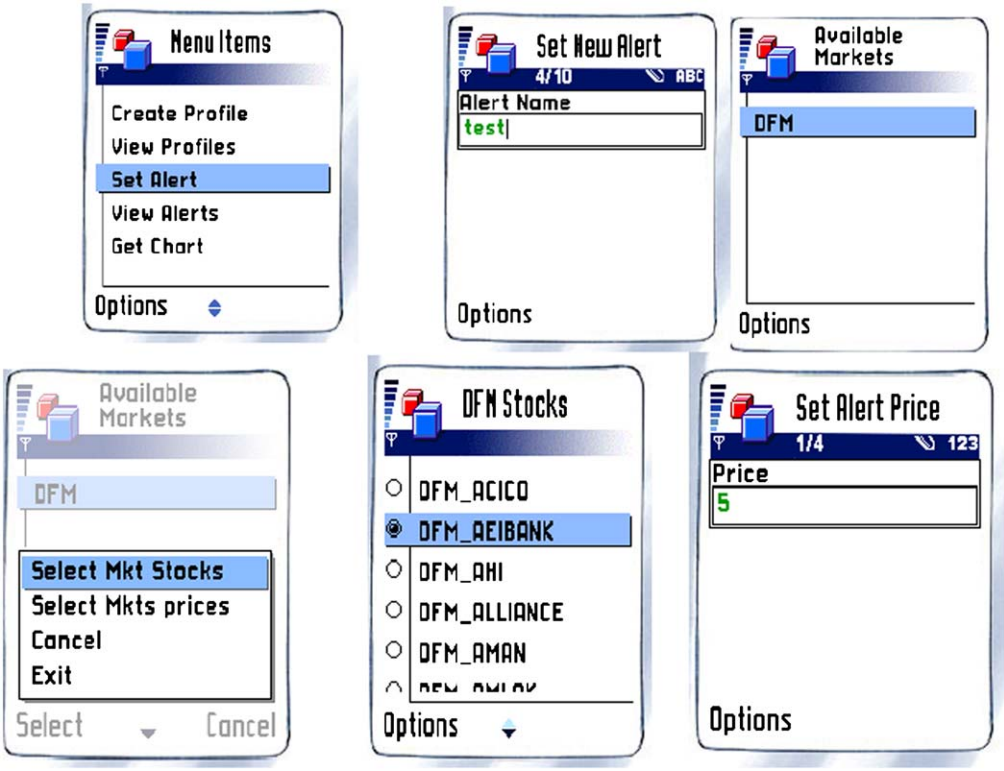

Fig. 4. User's menu guide for alert setting.

The C# application is running 24/7 and is continuously monitoring the database. The alert parameters set by the user are continuously checked against the real time values retrieved from the stock market that is stored in the database. If a match is found, then an SMS alert will be sent accordingly to the user's mobile phone number stored in the database. Thereafter, the alert information shall be deleted from the database when the SMS is delivered successfully. Fig. 4 illustrates various menu screens displayed by the mobile application during an *alert* selection or setting.

4.2. Charts feature

One of the practical and useful features of the proposed design is to allow the user to retrieve a chart containing stock information monitored through the past 7 days. First, this proposed feature was to be implemented in a similar manner to the SMS feature. Hence, the user selects a stock and requests the chart. Thereafter, the user's request shall be forwarded to the server that in turn pulls the stock information during the past seven days and plots the chart accordingly. The plot is then sent back as a Multimedia Messaging Services (MMS) message to the user.

Sending the requested information through MMS is a quite involved process. Unlike SMS, sending an MMS does not merely depend on the GSM modem for delivering the

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006

MMS message content; the GPRS technology for data transmission should be supported by the GSM modem as well. To send an MMS message, thus, we need a GSM/GPRS modem connected to the PC's serial port. Since MMS technology usually runs over GPRS, a point to point protocol (PPP) connection needs to be initiated between the GSM/GPRS modem and the PC. PPP works on transmitting the datagrams, containing MMS content, from serial point to point links. This connection shall be established through a set of AT commands to get attached to a GPRS connection through which the running application can stream data and transmit in the form of MMS message.

The above alternative, however, was not followed for several reasons. First, the GSM/GPRS modem was more expensive than a GSM modem. Second, it complicates the programming aspects and adds a processing overhead. Third, the GPRS connection cost is usually expensive for this service. Therefore, we opted for a better, faster, and a more efficient approach.

The alternative is to invoke a PHP page, upon requesting the chart, which pulls the required data from the database and creates the chart dynamically. Meanwhile, the server sends a response to the requesting client containing the image streamed. The application then receives the stream, creates the image, and displays it immediately to the user. This approach, hence, is much faster, convenient and more cost efficient. If the MMS approach

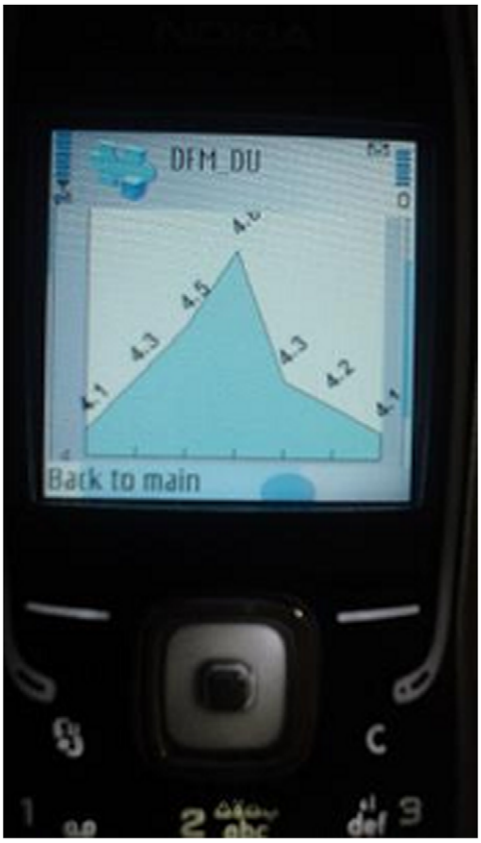

Fig. 5. Screen capture of a chart display.

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006

was used, the user had to subscribe to MMS service. The user had to bear the delay until the MMS message is received. On the server side, continuously running an MMS application that keeps monitoring the database for user requests would overload the system. Last but not least, the system had to bear a higher cost by maintaining the GPRS connection and delivering the MMS messages to the customers. Fig. 5 shows a sample screen shot of a requested chart.

Another suggestion is to have a chart feature to be implemented on the mobile device. In essence, the mobile would retrieve the data from the server and plot the graph locally. This approach has limitations since the server will have to transfer a significant amount of data and later the mobile device, with the limited CPU and memory capabilities, will need to plot the graph which will need a lot of time. However, plotting the graph on the server can be done much faster, since the server has a lot of memory and fast processors. Furthermore, the generated graph size is typically fixed, small, and independent of the data points, since the data is averaged in the graph. Finally, transferring the graph is also fast given the high GPRS speed. Therefore, we opted for plotting the graphs on the server.

5. System verification

After successfully deploying the application on a mobile and setting up the server on a PC, testing was carried out to assess the performance of our proposed approach.

Server testing—Server testing was done by accessing the server modules by sending HTTP requests using the internet browsers mimicking mobile side requests in a sequence of actions that looks like a regular user [4,13]. The test was done from 3 PCs sending one request per second. The result of that test was successful and the server was able to satisfy all the requests. The server was able to satisfy the test, because Apache creates a mutex that serializes the requests as processes and provides resources for them in sequence.

Application testing—We used the technology acceptance model (TAM) framework [21] to test possibility of adopting the proposed approach by normal users. TAM is an information system theory that models how users can accept, and use a new technology. It is based on two measures: perceived *usefulness* and *ease-of-use*. Random users of various age and backgrounds were invited to participate in the application testing phase. The users included students, staff, and faculty of both genders in the age range of 18–60 years old. The users had different backgrounds such as Engineering, Business, Architecture, and Liberal Arts areas. An important element in the design is the ease of use for potential users. It was found that the system is intuitive, and the device was easy to use even without any formal training in using the application. On the other hand, we checked the cost of creating a new profile, setting an alert, and getting charts. The user created a profile which has only one stock *DU* from Dubai Financial Market (DFM) and closing price information to be retrieved. From a delay point of view, the connection was fairly fast. It just takes few seconds to retrieve the data. Costwise, the connection was extremely cheap. The cost of the connection emphasizes the reduced number of packets transferred and received by the end user and accomplishes the main aim of the design. Hence, the protocol efficiency is validated. Retrieving the data costs only \$0.002 as opposed to downloading the whole page which costs around ~\$0.54. Retrieving charts costs ~\$0.02; and setting an SMS alert costs only \$0.008.

A comparison of real time data retrieved using the proposed approach and a normal GPRS connection is given in Table 2. We first access the DFM website through a normal

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006

Table 2
Comparison between the proposed system and normal GRPS connections.

	Proposed system	Normal GPRS
# Bytes downloaded (kbytes)	~10	~200
Price ($)	~0.03–0.08	~0.54
Time (s)	~2	~300

Table 3
Features comparison with Pocket-X.

	Proposed system	Pocket-X Stock Monitor
Implementation language	J2ME	VB.Net
Compatibility	All devices supporting Java	Only Pocket PCs and PDAs
Connection method	GPRS/BT/WiFi	GPRS/BT/WiFi
Stocks/prices retrieval	As per user selections	All stocks/prices
Alarms	Notification SMS (user offline/online)	Notification popup window (user online)
Delays	On the spot	20 min delay
Portfolio manager	Create/edit/delete	Create/edit/delete
Graphs	Dynamically created as per user request (display past 7 days price fluctuations)	Cumulative, i.e. with every stock quote fetched a point is added to the graph
Refresh rate	Fetch at a selected interval/click refresh to refresh once	Fetch a ta selected interval
Price	Free	$25/license

GPRS connection and monitor the number of bytes downloaded, cost and time. Then, we retrieve the portfolio information, compare, and contrast. As shown below, the comparison is done in terms of the following criteria:

- Number of bytes downloaded—retrieved using Ethereal Network Analyzer, which is a software package, used for sniffing packets transferred [18].
- Price—approximately calculated by multiplying the bytes monitored by the average cost/byte.
- Time in seconds.

It is evident from the results that a significant saving in packet size has been achieved. Also the cost and delay values attest to the fact that the proposed approach is economical and efficient.

Finally, it is worth mentioning that the Java client version was installed and tested on different mobile phones equipped with a GPRS connection and the results were consistent every time.

In Table 3, we compare and highlight the differences between our system, and Pocket-X Stock Monitor [19], a commercial product available in the market. Note that the Pocket-X is sold as a black box without any implementation details. Furthermore, the product has a 20 min delay and is not cost efficient since all stocks from a stock market webpage are

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006

downloaded. Our proposed solution has a much smaller delay and gives the user the option to create a portfolio of selected stocks from various markets. This can help in significantly reducing the packet sizes, telecommunication costs, and time delays. Furthermore, the implementation details are published for free use and the tool will be available online shortly.

6. Conclusions

As the popularity and use of mobile devices continue to rise and as its cost continues to fall, the need for web-related services over mobile will no doubt follow an upward trend.

This paper describes the architecture for a cost-efficient and user-friendly stock monitoring system that works with Java-enabled mobile phones. It proposes to use an intermediate database server that directly retrieves the content from several stock-market webpages, and provides the information using the Java application running on the mobile device. The architecture also includes a GSM modem solution to generate and forward SMS alerts for time-critical information. The proposed design has several advantages including:

1. ability to download selective stocks rather than all stocks from a stock market;
2. optimized packet encoding for faster and cheaper response;
3. customized stock information refresh-rate;
4. easy-to-view portfolios menu;
5. SMS alerts;
6. optimized chart display algorithm;
7. accessible for any Java-enabled phone.

Additionally, the proposed system disassociate the client from the stock information source, that is, a change in the stock information source (such as website content, address, or port) is managed on the server side only without the need to modify the clients. Another main advantage of the approach is its ability to consolidate content from various sources, i.e. stock market webpages, into a single site access. This approach is very beneficial if no single finance website can provide all the information that is needed by the mobile user.

The technical feasibility of the solution was tested by parsing data from the Dubai Financial Market webpage. The cost advantage was shown in comparison to access the same information through the Dubai Financial Market Mobile Internet webpage. Experimental results indicate that significant reduction in cost, exchanged packet size, and transmission delay is achieved without any inconvenience to the user. We also used the technology acceptance model framework to test if users would accept and use the proposed approach. Users found the approach useful, cost effective, and easy to use.

Despite the success of the proposed solution, it has few limitations. The implementation works only for Java-enabled mobile phones. The phone must be able to communicate via GPRS or WiFi. Furthermore, access to a stock market webpage data needs to be legally authorized, i.e. the service can only be provided in agreement with the content provider.

Future work includes expanding the application to cover international financial markets, testing the system with brokerage systems, and porting the application to run, not only on Symbian OS, but also Windows CE in order to support PDAs in addition to mobile phones.

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006

References

[1] H. Kanma, et al., Home appliance control system over Bluetooth with a cellular phone, IEEE Transactions on Consumer Electronics 49 (4) (2003).
[2] K. Yamauchi, W. Chen, D. Wei, 3G mobile phone applications in telemedicine—a survey, in: Proceedings of the Fifth International Conference on Computer and Information Technology, 2005.
[3] H. Murakami, A. Ito, Y. Watanabe, T. Yabe, Mobile phone based ad hoc network using built in Bluetooth for ubiquitous life, in: Eighth International Symposium on Autonomous Decentralized Systems, ISADS '07, 21–23 March 2007, Sedona, AZ, pp. 137–146.
[4] V. Astarita, M. Florian, The use of mobile phones in traffic management and control, in: Proceedings of the IEEE Intelligent Transportation Systems, Oakland, CA, 2001, pp. 10–15.
[5] Zhang, Pu, et al., A remote patient monitoring system using a Java-enabled 3G mobile phone, in: Proceedings of the 29th IEEE International Engineering in Biology and Medicine Conference, 2007.
[6] R. Ballagas, et al., The smart phone: a ubiquitous input device, IEEE Pervasive Computing 5 (1) (2006).
[7] K. Pousttchi, D. Wiedemann, A contribution to the theory building for mobile marketing: categorizing marketing campaigns through case study research, in: Proceedings of the International Conference on Mobile Business, 2006.
[8] K. Pousttchi, D. Wiedemann, Success factors in mobile viral marketing: a multicase study approach, in: Proceedings of the International Conference on Mobile Business, 2007.
[9] H. Kargupta, et al., MobiMine: monitoring the stock market from a PDA, Journal of ACM SIGKDD Explorations 3 (2002).
[10] S. Brewster, R. Murray, Presenting dynamic information on mobile computers, Journal of Personal Technologies 4 (2000).
[11] Dubai Financial Market (DFM), Available at: ⟨http://www.dfm.ae⟩.
[12] J. Ellis, M. Young, JSR172: J2ME Web Services Specification, White paper, Version 1.0, 2004.
[13] PHP Support, Available at: ⟨http://www.php.net⟩.
[14] Simplified guide to the Java 2 platform—Enterprise Edition, White paper, Sun Microsystems, September 1999.
[15] Sun Java Wireless Toolkit, Available at: ⟨http://java.sun.com/products/sjwtoolkit/⟩.
[16] B. Yuksekkaya, et al., A GSM, internet and speech controlled wireless interactive home automation system, IEEE Transactions on Consumer Electronics 52 (3) (2006).
[17] HSL Mobile Messaging, Advanced Services SMS Gateway: HTTP Interface for Sending & Receiving SMS, Revision 4, October 2004.
[18] The Ethereal Network Analyzer v0.99, Available at: ⟨http://www.ethereal.com/⟩.
[19] Pocket-X Stock Monitor Software, Available at: ⟨http://www.pocketx.net⟩.
[20] Axmor J2ME Applications, Available at: ⟨http://www.axmor.com/j2me-development.aspx⟩.
[21] Fred Davis, Perceived usefulness, perceived ease of use, and user acceptance of information technology, MIS Quarterly 13 (1989) 319–341.
[22] Innovations Report Web Page, Available at: ⟨http://www.innovationsreport.de/html/berichte/innovative_produkte/bericht-15635.html⟩.

Please cite this article as: F. Aloul, et al., M-stock: Efficient stock monitoring for mobile users, J. Franklin Inst. (2010), doi:10.1016/j.jfranklin.2010.02.006