M-Stock: Efficient Stock Monitoring for Mobile Users

Fadi Aloul, Assim Sagahyroon, Nancy Salem, Omar Al-Omar, Sahar Shehada

Department of Computer Engineering
American University of Sharjah, UAE

Abstract — *In today's 24x7 business world, there is an increasing reliance on technology to support the core business functions. With this reliance comes a growing acceptance for any technology to maintain continuity of business operations, and even something "more" is always required. M-Stock was designed to provide a real-time mobile scheme to guarantee continuity of business operations, i.e. "on-the-run" access to the stock exchanges.*

This paper guides the readers through the design and implementation stages of M-Stock, a mobile application for stock market users. M-Stock was designed as an enhanced alternative to costly GPRS connections for stock market users. The system grants real-time monitoring of stock information through a user-friendly and easy to use GUI with a plenty of useful features.

I. Introduction

During the early 1990s, stock exchange transactions were done using the phone through brokers. Using phones introduced many inconveniences due to long waiting times and busy lines. Moreover, the availability of the customer monitoring the stocks on the spot was a necessity to perform the transactions. Later on, and as internet use grew widely, stock markets users shifted to performing their transactions live online. Despite the fact that this shift has reduced the inconveniences of using the phone, the customer still had to be locked to his/her desk monitoring the stock market online to perform the transaction.

Nowadays, and with the expansion of mobile technologies, customers can get a better alternative. Customers can have access to the internet anywhere anytime on the run 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 [3] containing a plenty of data to download. Additionally, with a normal GRPS connection, if the customer is interested in stock information from different stock markets, he/she has no other choice but browsing through each website. This solution is costly, time consuming, inconvenient; in addition to this, the customers' stocks of interest might be only few out of all the stocks available at the website.

For these reasons, M-Stock was developed. M-Stock has a friendly and easy to use graphical user interface that allows the users to create portfolios for the 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 the few bytes constituting the required information. In turn, this process (i) reduces the connection costs significantly and (ii) provides a faster screen refresh rate, as the number of bytes downloaded will be reduced.

M-Stock was designed to meet the needs of the stock market customers. The application provides its users with a plenty of valuable features.

- The application has easy to use GUI with the capability of creating/editing/deleting portfolios that store the user preferences of stocks and prices.
- Portfolio information retrieval is done 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 seven days.
- Information retrieval is automatically updated according to the refresh rate minutes set by the user.
- A secured connection using HTTPS protocol to prevent information fraud.

The paper is organized as follows. Section II provides a general overview of the system including its connections and components. Section III describes the application features. The system is tested and analyzed in Section IV. In Section V, previous work in this area is summarized. Finally, the conclusion and future work are presented in Section VI.

II. System Overview

This section covers the details regarding the system connections and system components. The section proceeds with a general layout of the design. Thereafter, the first subsection proceeds with the reader through the protocol used to establish the achieved design objectives. The second subsection will guide the reader through an overview of the complete system and how its different components interoperate and connect.

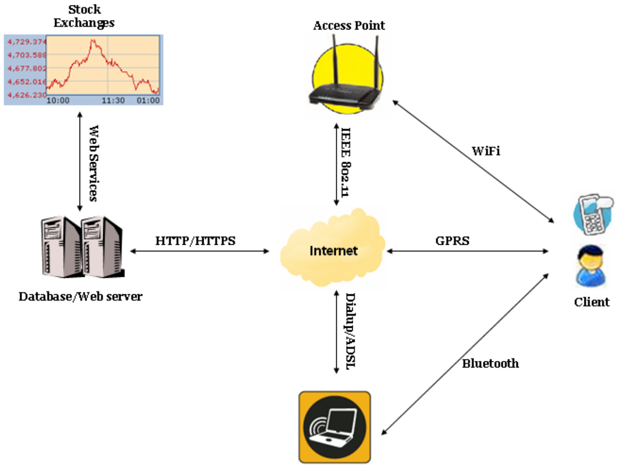

Figure 1: General overview of the M-Stock system.

A. *System Connections*

Figure 1 shows a general overview of the system. The *web server* has to be 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 main tasks:

1. Connect to stock market websites which is done using the webservices provided by the stock markets [8].
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 established protocol explained in Table 1.

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
…	

Table 1: The M-Stock protocol used to efficiently exchange information between the web server and mobile client.

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 the 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 the request to retrieve the request parameters. The server formulates the appropriate SQL query accordingly 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 proper format [15]. 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 into the database data as required. An example is shown in Figure 2.

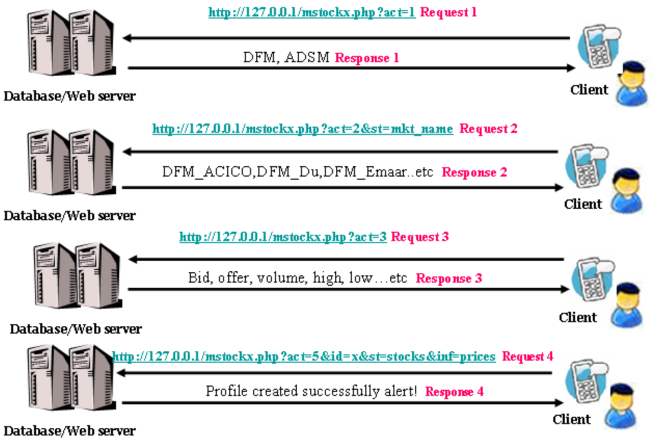
http://127.0.0.1/mstockx.php?act=1 Request 1
DFM, ADSM Response 1
Database/Web server
Client
http://127.0.0.1/mstockx.php?act=2&st=mkt_name Request 2
DFM_ACICO,DFM_Du,DFM_Emaar..etc Response 2
Database/Web server
Client
http://127.0.0.1/mstockx.php?act=3 Request 3
Bid, offer, volume, high, low…etc Response 3
Database/Web server
Client
http://127.0.0.1/mstockx.php?act=5&id=x&st=stocks&inf=prices Request 4
Profile created successfully alert! Response 4
Database/Web server
Client

Figure 2: An example showing the exchanged packets between the web server and the mobile client.

B. *System Components*

The system consists of 3 components which are described in the following sections.

B.1 Server

The server functions were implemented using PHP scripts [15]. 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 were sufficient to implement all of the functions in the defined protocol.

The server is the hardcore of the system. The main tasks of PHP server are:

- Retrieve, process, and insert Dubai Financial Market information in the MySQL database [3].
- Retrieve, process, and insert Abu Dhabi Securities Market information in the MySQL database [1].
- Store user profiles upon arrival of user request to create new profile.
- Send available financial markets available upon arrival of user request
- Send available companies within the selected market upon arrival of user request
- Send available stock prices (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.

B.2 Application

On the mobile side, the application was developed and implemented using Java 2 micro edition (J2ME) [9, 10, 11, 12, 17, 18]. The application classes interoperate to provide the user with a friendly, easy to use and a handy tool for performing various stock market transactions.

The application made use of the javax.microedition.rms package. Record Management Store (RMS) is a persistence package that provides the application with a mechanism for persistent storage of data. Hence, the application used this package for persistent storage of user portfolios and alerts. The 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. Shown below is the application features hierarchy.

1.0 Mobile Stock_X
    1.1 Main Menu
        1.1.1 Create new profile
        1.1.2 View profiles
            1.1.2.1 View profile details (stocks, prices selected and stored)
            1.1.2.2 Edit profile details
            1.1.2.3 Delete profile
            1.1.2.4 Retrieve profile information
            1.1.2.5 Back to main menu
            1.1.2.6 Exit application
        1.1.3 Set alerts
        1.1.4 View alerts
            1.1.4.1 View alert details
            1.1.4.2 Delete alert
            1.1.4.3 Back to main
            1.1.4.4 Exit
        1.1.5 Get charts

B.3 Database

The server uses a 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 hence 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.

III. APPLICATION FEATURES

A. 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 [2, 6, 7, 14]. 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 the Mysql database (server database), and retrieve the needed information by executing SQL queries.

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 are 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.

B. Charts Feature

One of the proposed features of the proposed design was 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 shall be then sent back as an MMS message to the user.

Sending the requested information through MMS is a quite involved process. Unlike SMS, sending an MMS doesn't merely depend on the GSM modem for delivering the MMS message content; 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 PPP (point to point protocol) 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.

This approach, however, was not followed for several reasons. First, the GSM/GPRS modem was more expensive than an GSM modem. Second, programming the complication is complicated. Third, the GPRS connection cost is usually expensive for this service. Therefore, we had to implement a better, faster and a more efficient approach.

The alternative was 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 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. Below is a sample screen shot of a requested chart.

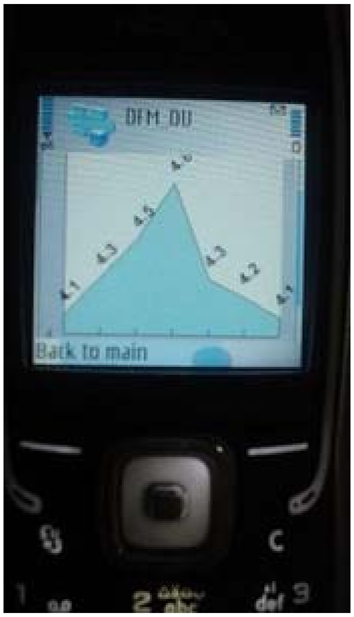

Figure 3: Example of chart displayed on M-Stock.

IV. SYSTEM TESTING

A. Server Testing

The testing for the server 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 PC's 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.

B. Application Testing

The application testing was conducted to check the ease of use for any new customer. It was found that M-Stock is a handy and easy to use device even without any introduction to 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 and closing price information to be retrieved. From time point of view, the connection was fairly fast. It just takes few seconds to retrieve the data. Money wise, 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 1 fils as opposed to downloading the whole page which costs around ~2 dirhams. Retrieving charts costs ~7 fils; and setting an SMS alert costs only 3 fils.

C. M-Stock vs. Normal GPRS

This section presents a comparison for real time data retrieved using M-Stock and a normal GPRS connection. We first access the DFM website through a normal 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, which is a software used for sniffing packets transferred [5].
- Price - approximately calculated by multiplying the bytes monitored by the average cost/byte.
- Time in seconds

	M-Stock	Normal GPRS
#Bytes downloaded	~10 Kbytes	~200Kbytes
Price	~ 10-30 Fils	~2 Dirhams
Time	~2 seconds	~300 seconds

Table 2: Comparison between the M-Stock and normal GRPS connections.

V. PREVIOUS WORK

Pocket-X Stock Monitor [16] is one of the commercial products available in the market that is much similar to the proposed design. However, there are some differences in implementation and features. These are explained in Table 3.

VI. CONCLUSIONS

Current mobile phones can only view mobile-accessible websites that offer stock information. Mobile users are forced to download the complete content of the website which is likely to include unused information. Given that most stock prices are continuously changing, downloading the stock information can be inconvenient, since GPRS is known to be expensive and slow. Furthermore, many brokerage and financial markets do not offer mobile-accessible websites.

We propose a mobile application to efficiently monitor and trade stocks. The application consists of a client, placed on the mobile phone, that connects to a server via GPRS or Bluetooth or WiFi. The server connects to the financial markets via the web, collects the latest stock information, filters out unnecessary information, and stores it in a local database. Stock information is sent by the server to the client whenever the client requests it. The sent information is efficiently encoded to reduce the packet size, speed up the transmission, and reduce the telecommunication costs. This approach allows the client to collect stock information from any website. The application is tested using the UAE financial market. Experimental results show the significant reduction achieved in exchanged packet size, transmission delay, and costs.

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.

REFERENCES

[1] Abu Dhabi Securities Market (ADSM). Available at http://www.adsm.ae

[2] B. Yuksekkaya, A. Kayalar, M. Tosun, M. Ozcan, A. Alkar, “A GSM, Internet and Speech Controlled Wireless Interactive Home Automation System,” *IEEE Transactions on Consumer Electronics*, Vol. 52, No. 3, August 2006.

[3] Dubai Financial Market (DFM). Available at http://www.dfm.ae.

[4] Eclipse. Available at http://www.eclipse.org.

[5] Ethereal Network Analyzer v0.99. Available at http://www.ethereal.com/

[6] HSL Mobile Messaging, “Advanced Services SMS Gateway: HTTP Interface For Sending & Receiving SMS,” Revision 4, October 2004.

[7] Huawei VAS SDK User Manual. Huawei Technologies Co. Ltd. 2003.

[8] J. Ellis and M. Young, “JSR172: J2ME Web Services Specification,” white paper, version 1.0, 2004.

[9] JAVA Tutorial. Available at http://java.sun.com.

[10] Java Community Process, “JSR-000118: Mobile Information Device Profile 2.1 Maintenance Release,” May 2006.

[11] J2ME Polish v1.2.4. Available at http://www.j2mepolish.org.

[12] M. Jode, “Programming Java 2 Micro Edition on Symbian OS,” *John Wiley and Sons*, July 2004.

[13] Nokia Prototype SDK for Java™ Platform, Micro Edition. Available at http://forum.nokia.com.

[14] E. Ortiz, “The Wireless Messaging API,” 2002. Available at http://developers.sun.com/techtopics/mobility/midp/ articles/wma/.

[15] PHP Support. Available at http://www.php.net.

[16] Pocket-X Stock Monitor Software. Available at http://www.pocketx.net.

[17] Sun Microsystems, “Simplified Guide to the Java 2 Platform, Enterprise Edition,” *white paper*, September 1999.

[18] Sun Java Wireless Toolkit 2.5 Beta. Available at http://java.sun.com/products/sjwtoolkit/.

	M-Stock	**Pocket-X Stock Monitor**
Implementation Language	J2ME	VB.Net
Compatibility	All devices supporting Java	Only Pocket PCs & 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 minutes delay
Portfolio Manager	Create/Edit/Delete	Create/Edit/Delete
Graphs	Dynamically created as per user request (display past seven 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 at a selected interval
Price	Free	$25/license

Table 3: Comparison of features between M-Stock and the commercial Pocket-X Stock Monitor.