A sample stock exchange—WAP application

Siddhi Pittayachawan, RMIT University
A Sample Stock Exchange
- WAP Application

by

Siddhi Pittayachawan, M. Eng., B. Eng.

A Minor Thesis submitted in partial
fulfilment of the requirements for
the degree of Master of Engineering
(Telecommunication)

Department of Communication and Electronic Engineering

RMIT

November 2000
# TABLE OF CONTENTS

ABSTRACT .................................................................................................................. 4  
ACKNOWLEDGEMENTS ............................................................................................... 5  
DECLARATION ............................................................................................................. 6  
PERMISSION TO COPY ............................................................................................... 7  
1. INTRODUCTION ..................................................................................................... 8  
2. WIRELESS APPLICATION PROTOCOL ............................................................... 9  
3. ARCHITECTURE OVERVIEW ............................................................................... 10  
   3.1. THE WORLD-WIDE WEB MODEL .................................................................. 10  
   3.2. THE WAP MODEL ......................................................................................... 11  
   3.3. COMPONENT OF WAP ARCHITECTURE .................................................... 12  
      3.3.1. WIRELESS APPLICATION ENVIRONMENT .......................................... 13  
      3.3.2. WIRELESS SESSION PROTOCOL ...................................................... 14  
      3.3.3. WIRELESS TRANSACTION PROTOCOL ............................................ 14  
      3.3.4. WIRELESS TRANSPORT LAYER SECURITY ..................................... 15  
      3.3.5. WIRELESS DATAGRAM PROTOCOL .................................................. 15  
      3.3.6. BEARERS ............................................................................................... 15  
      3.3.7. OTHER SERVICES AND APPLICATIONS .......................................... 15  
   3.4. EXAMPLE OF WAP NETWORK ..................................................................... 16  
4. WAP DEVELOPMENT TOOLKITS ........................................................................... 17  
   4.1. WAPIDE SDK ................................................................................................. 18  
      4.1.1. SYSTEM REQUIREMENTS ................................................................. 18  
      4.1.2. BROWSER ............................................................................................. 19  
      4.1.3. APPLICATION DESIGNER ................................................................. 21  
      4.1.4. SERVER TOOLSET ................................................................................. 22  
      4.1.5. TROUBLESHOOTING ......................................................................... 22  
   4.2. WAP TOOLKIT ............................................................................................... 23  
      4.2.1. SYSTEM REQUIREMENT ...................................................................... 23  
      4.2.2. SIMULATOR AND BROWSER ............................................................... 24  
      4.2.3. TROUBLESHOOTING ......................................................................... 26
ABSTRACT

At the present time, WAP (Wireless Application Protocol) is a new technology that enables transmission of data from the Internet (World Wide Web) into any wireless terminal. Hence, users will obtain additional services that normal wireless devices do not have. Many applications are continuing to be developed for use within WAP wireless terminals, such as a micro browser, calendar, SMS or electronic mail. These applications have been developed for use with wireless devices that have narrow-bandwidth transmission. The chosen application described in this minor thesis concerns a "stock exchange" application. This application retrieves share price information from stock exchange websites and stores this in a local database. Users are able to set their own trigger prices to let them know whenever the share price is at the price the users want to buy or sell. This application is a dynamic system and must retrieve the share information, compare share prices, and send related information to users at regular intervals.
ACKNOWLEDGEMENTS

I would like to thank Professor Richard Harris who gives me an idea to do this application. He helped me to find important information to implement this thesis and corrected this report as well. Moreover, I would like to thank my parents who supported me to study in Australia.
DECLARATION

I certify that this report contains no material which has been accepted for the award of any other degree or diploma in any institute, college or university, and that, to the best of my knowledge and belief, it contains no material previously or written by another person, except where due reference is made in the text of the Literature Review.

Siddhi Pittayachawan
PERMISSION TO COPY

I hereby give permission to the staff of the University Library and to the staff and students of the Department of Communication and Electronic Engineering at RMIT to copy this report in whole or part without reference to me. This permission covers only single copies made for study purposes, subject to the normal conditions of acknowledgement.

Siddhi Pittayachawan
1. INTRODUCTION

WAP (Wireless Application Protocol) is a recently established new technology by a non-profit organisation known as the WAP Forum. WAP is used on wireless terminals such as mobile phones, pager, etc. Unlike computers, these devices have a narrow bandwidth for transmitting data. Therefore, the protocols used on these devices have been optimised to be suitable for this narrow bandwidth. Also, the optimised protocols are based on World Wide Web protocols to enable the connectivity between wireless terminals and the Internet.

The objective of this thesis is to build a prototype application for use in WAP mobile phones. The languages used for building the application are WML (Wireless Markup Language), WMLScript (Wireless Markup Language Script), and others as necessary. WML and WMLScript have been optimised from HTML (HyperText Markup Language) and JavaScript respectively to be suitable for a narrow bandwidth environment.

There are four major chapters in this report as follows:

- **Wireless Application Protocol** chapter that introduces the concept of WAP and WAP Forum requirements.
- **Architecture Overview** chapter that describes the models and layered architectures of WWW and WAP, and also shows a combination of WWW and WAP networks.
- **WAP Development Toolkit** chapter explains the functions of each chosen application for developing the WAP application and compares them together.
- **WAP Application** chapter that explains about the WAP application chosen for this thesis and also describes how to implement it.

---


- 8 -
2. WIRELESS APPLICATION PROTOCOL

Wireless Application Protocol or WAP is the world standard for the presentation and transmission of wireless information and telephony services on mobile phones and other wireless devices. It was developed by WAP Forum to promote industry-wide specification for useful technologies in developing applications and services that operate over wireless communication networks. WAP specifies an application framework and network protocols for wireless devices such as mobile phones, pagers, and personal digital assistants (PDAs). The specifications leverage mobile networking technologies, i.e. digital data networking standards, and Internet technologies, i.e. HTML², XML³, URLs, scripting, and various content formats. The purpose is to enable operators, manufacturers, and content developers to meet the challenges in creating advanced differentiated services and implementations in a fast and flexible manner.

WAP is aimed at the convergence of two rapidly evolving network technologies, wireless data and the Internet. Both of them are growing very fast and continuously gaining new customers. Most of technologies developed for the Internet have been designed for desktop and high-end computers and medium to large bandwidth. This is opposite to wireless devices that have environmental constraints compared to desktop computers, that is:

- Less powerful CPUs
- Less memory (ROM and RAM)
- Restricted power consumption
- Smaller displays
- Different input devices (e.g. a phone keypad)
- Less bandwidth
- More latency
- Less connection stability, and
- Less predictable availability

Therefore, to transfer information from the Internet to wireless devices WAP must be designed properly and the requirements that the WAP Forum needs are: [12]

- Leverage existing standards where possible
- Define a layered, scaleable and extensible architecture
- Support as many wireless networks as possible
- Optimise for narrow-band bearers with potentially high latency
- Optimise for efficient use of device resources (low memory/CPU usage/power consumption)
- Provide support for secure applications and communication
- Enable the creation of Man Machine Interfaces (MMIs) with maximum flexibility and vendor control
- Provide access to local handset functionality, such as logical indication for incoming calls
- Facilitate network-operator and third party service provisioning
- Support multi-vendor interoperability by defining the optional and mandatory components of the specifications
- Provide a programming model for telephony services and integration

3. ARCHITECTURE OVERVIEW

This chapter illustrates architectures from the World-Wide Web model and the WAP model, and shows how they are different. The World-Wide Web model has been developed to maximise the use of bandwidth, while the WAP model has been developed to minimise the use of bandwidth.

3.1. THE WORLD-WIDE WEB MODEL

Figure 1 depicts a model of the Internet. Applications and content are presented in standard data formats, and are browsed by applications called web browsers such as Microsoft Internet Explorer, Netscape Communicator, Opera, etc. These web browsers are networked applications that send requests for wanted data to network servers and the servers respond back with the data encoded using standard formats. In the Internet, the standard naming model is known as URL. While transmitting data between a client and web server, there are various types of data format used, e.g.
HTML, XML, and JavaScript. To allow any web browser to communicate with any web server, networking protocols are needed. There are several protocols being used in the Internet and the most commonly used protocol is the HyperText Transport Protocol (HTTP).

![World-Wide Web Programming Model](image)

**Figure 1: World-Wide Web Programming Model**

The WWW protocols define three classes of servers: [12]

- **Origin server** – The server on which a given content is to be created
- **Proxy** – An intermediary program that acts as both a server and a client for the purpose of making requests on behalf of other clients. The proxy typically locates between servers and clients that have no means of direct communication.
- **Gateway** – A server that acts as an intermediary for some other servers. Unlike a proxy, a gateway receives requests as if it were the origin server for the requested resource.

### 3.2. THE WAP MODEL

Figure 2 shows the model of WAP that is similar to the WWW model. This provides many advantages to application developers, including a familiar programming model, and the ability to leverage existing tools. WAP contents and applications are specified in a set of content formats based on the familiar WWW content formats. Content is transferred using standard protocols based on the WWW communication protocols. A micro browser in the wireless terminal coordinates the user interface and is known as
a standard web browser. To identify WAP content, WWW-standard URLs are used on origin servers to identify local resources, e.g. Call control functions, WWW-standard URIs are used.

![WAP Programming Model](image)

**Figure 2: WAP Programming Model**

The WAP content types and protocols have been optimised for hand-held wireless devices. WAP utilises proxy technology to connect between the wireless domain and the WWW. The WAP proxy typically consists of the following functionality: [12]

- **Protocol Gateway** – The protocol gateway translates requests from the WAP protocol stack (WSP, WTP, WTLS, and WDP)\(^4\) to the WWW protocol stack (HTTP and TCP/IP).
- **Content Encoders and Decoders** – The content encoders translate WAP content into compact encoded formats to reduce the size of data to be suitable for the wireless device bandwidth.

### 3.3. COMPONENT OF WAP ARCHITECTURE

The WAP architecture provides a scaleable and extensible environment for application development for mobile communication devices. This is shown in Figure 3. It is a layered design of the entire protocol stack. Each layer of the architecture is accessible by the layers above, as well as by other services and applications.

---

The WAP layered architecture allows other services and applications to utilise the features of the WAP stack through a set of well-defined interfaces. External applications may access the session, transaction, transport, and security layers directly. The following sections provide a description of the various elements of the protocol architecture. [12]

### 3.3.1. WIRELESS APPLICATION ENVIRONMENT

The WAE (Wireless Application Environment) is an application environment based on a combination of mobile telephony and WWW technologies. The objective of the WAE is to establish an interoperable environment that allows operators and service providers to create useful applications and services that can reach a variety of different wireless platforms. WAE includes a micro-browser environment containing the following functionality:

- Wireless Markup Language (WML) – a lightweight markup language, similar to HTML, but optimised for use in hand-held mobile terminals
- WMLScript – a lightweight scripting language, similar to JavaScript™
- Wireless Telephony Application (WTA, WTAI) – telephony services and programming interfaces
- Content Formats – a set of well-defined data formats, including images, phone book records and calendar information

![Figure 3: WAP and WWW Architectures](image)

The WAP layered architecture allows other services and applications to utilise the features of the WAP stack through a set of well-defined interfaces. External applications may access the session, transaction, transport, and security layers directly. The following sections provide a description of the various elements of the protocol architecture. [12]

### 3.3.1. WIRELESS APPLICATION ENVIRONMENT

The WAE (Wireless Application Environment) is an application environment based on a combination of mobile telephony and WWW technologies. The objective of the WAE is to establish an interoperable environment that allows operators and service providers to create useful applications and services that can reach a variety of different wireless platforms. WAE includes a micro-browser environment containing the following functionality:

- Wireless Markup Language (WML) – a lightweight markup language, similar to HTML, but optimised for use in hand-held mobile terminals
- WMLScript – a lightweight scripting language, similar to JavaScript™
- Wireless Telephony Application (WTA, WTAI) – telephony services and programming interfaces
- Content Formats – a set of well-defined data formats, including images, phone book records and calendar information
3.3.2. WIRELESS SESSION PROTOCOL

The WSP (Wireless Session Protocol) provides the application layer of WAP with a consistent interface for two session services. One is a connection-oriented service that operates above the transaction layer protocol WTP, and another is a connectionless service that operates above a secure or non-secure datagram service (WDP).

The WSPs currently consist of services suited for browsing applications (WSP/B). WSP/B provides the following functionality:

- HTTP 1.1 functionality and semantics in a compact over-the-air encoding
- Long-lived session state
- Session suspend and resume with session migration
- A common facility for reliable and unreliable data push
- Protocol feature negotiation

3.3.3. WIRELESS TRANSACTION PROTOCOL

The WTP (Wireless Transaction Protocol) runs on top of a datagram service and provides a lightweight transaction-oriented protocol that is suitable for implementation in mobile stations. WTP operates efficiently over secure or non-secure wireless datagram networks and provides the following features:

- Three classes of transaction service:
  - Unreliable one-way requests
  - Reliable one-way requests
  - Reliable two-way request-reply transactions
- Optional user-to-user reliability – WTP user triggers the confirmation of each received message
- Optional out-of-band data on acknowledgements
- PDU concatenation and delayed acknowledgement to reduce the number of messages sent
- Asynchronous transactions
3.3.4. WIRELESS TRANSPORT LAYER SECURITY

The WTLS (Wireless Transport Layer Security) is a security protocol based on the Transport Layer Security (TLS) protocol, formerly known as Secure Socket Layer (SSL). WTLS provides the following features:

- Data integrity – WTLS contains facilities to ensure that data transmitted between the terminal and application server is unchanged and uncorrupted
- Privacy – WTLS contains facilities to ensure that data transmitted between the terminal and application server is private and cannot be understood by any intermediate parties that may have intercepted the data stream
- Authentication – WTLS contains facilities to establish the authenticity of the terminal and application server
- Denial-of-service protection – WTLS contains facilities for detecting and rejecting data that is replayed or not successfully verified. WTLS makes many typical denial-of-service attacks harder to accomplish and protects the upper protocol layers.

3.3.5. WIRELESS DATAGRAM PROTOCOL

The WDP (Wireless Datagram Protocol) is the transport layer protocol in the WAP architecture. The WDP layer operates above the data capable bearer services supported by the various network types. The WDP provides a consistent service to the upper layer protocols of WAP and communicate transparently over one of the available bearer services.

3.3.6. BEARERS

The WAP protocols are designed to operate over various types of bearer services, including short message, circuit-switched data, and packet data. The bearers offer differing levels of QOS with respect to throughput, error rate, and delays. The WAP protocols are designed to compensate for or tolerate these varying levels of service.

3.3.7. OTHER SERVICES AND APPLICATIONS

The WAP architecture allows other services and applications that are not currently specified by WAP to utilise the features of the WAP stack through a set of well-
defined interfaces. External applications may access the session, transaction, and transport layers directly. These applications, such as electronic mail, calendar, phone book, notepad, and electronic commerce, or services (white and yellow pages), may be developed to use the WAP protocols.

3.4. EXAMPLE OF WAP NETWORK

In Figure 4, the WAP client communicates with two servers in the wireless network. The WAP proxy translates WAP requests to WWW requests. This allows the WAP client to submit requests to the web server in the Internet. Also the proxy translates the responses from the web server into a compact binary format known by the client.

If the web server provides WAP content, such as WML, the WAP proxy can retrieve it directly from the web server. However, if the web server provides WWW content, such as HTML, a filter is used to change the WWW content into WAP content.

![Figure 4: Example WAP Network](image)

The Wireless Telephony Application (WTA) server is an example origin or gateway server that responds to requests directly from the client. The WTA server is used to provide WAP access to features of the wireless network provider’s telecommunications infrastructure. [12]
4. WAP DEVELOPMENT TOOLKITS

At the present time, some telephone providers have been developing WAP development toolkits for use with their own telephone functionality or which are compatible with several telephone providers’ equipment. These toolkits are applications for writing WML and WMLScript, and also connect to some other languages such as Java, Perl, CGI, etc. The content written in WML will be separated using a “deck format” and then encoded into a binary encoded format.

In this thesis, three toolkits were chosen for testing to determine whether their functionality was suitable for development of the special application.

- WapiDE SDK Version 2.1 (Ericsson)
- WAP Toolkit Version 1.3 beta (Nokia)
- UP.SDK Version 4.0 beta 2 (Phone.com)

For each development toolkit, registration is required. After registration, the various companies will provide their development toolkits, user guides, FAQs, etc. The addresses for the development toolkit providers are the following:

- Ericsson at http://ww.ericsson.com/developerszone
- Phone.com at http://developer.phone.com

The computer system used to test the WAP development toolkits was the following:

- Pentium II 400 MHz
- 64 MB RAM
- 6.4 GB hard disk
- Windows 98

---

5 The deck format is a collection of WML cards. The card is a unit contained user interface. The whole deck will be downloaded when it is used. However, only one card will be displayed at a time. Therefore, the deck should not contain too many cards due to limitations of memory in a WAP mobile phone.
4.1. **WapIDE SDK**

Ericsson Company developed this kit. The current version is Version 2.1.

4.1.1. **SYSTEM REQUIREMENTS**

The system requirements for this application include the following:

- Windows NT version 4 or Windows 95/98
- Minimum 10 MB of free hard disk space
- 32 MB RAM

The software products included are

- WapIDE
- Perl5
- Xitami Web Server
- TCL/TK

![WapIDE SDK Interface](image)

**Figure 5: WapIDE SDK Interface**

The browser uses the Ericsson mobile phone, model R320, to implement an application. The total hard disk space required for the installation program is 22.7 MB. WapIDE SDK requires an external web server program called Xitami Web Server. This server makes a replica Internet address for developing an application in a stand-alone computer (no connection to the Internet or network is required). WapIDE SDK can also connect to an Ericsson proxy server for seeing some examples from the Internet.
The main functions in the WapIDE SDK can be separated into the following:

- Browser
- Application Designer
- Server Toolset

The browser is used to view applications, i.e. to browse WML cards. Also it interprets WMLScript. Instead of using it as a WAP device to access WAP applications, the tool can also be used as a WML browser. It reads both binary WML bytecode and textual WML files.

4.1.2. BROWSER

There are three ways to load content to the browser; from a:

- WAP gateway using WAP client stack.
- Web server using HTTP client
- File containing content saved on the hard disk

Figure 6: Browser when a R320 device is loaded

To use the browser, the process is the same as when the WAP mobile phone is used. The device must be turned on first by pressing and holding the power button (for R320 model the power button is the “NO” key). After the device is activated, the following generic actions apply for Ericsson devices:

Figure 7: Browser at start
Table 1: Key functions of Ericsson R320 mobile phone

<table>
<thead>
<tr>
<th>Button</th>
<th>User action</th>
<th>Device action</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>Short click</td>
<td>Sends an ACCEPT event</td>
</tr>
<tr>
<td></td>
<td>Hold-down click</td>
<td>Shows a list of options, including soft options</td>
</tr>
<tr>
<td>NO</td>
<td>Short click</td>
<td>Goes BACK in history</td>
</tr>
<tr>
<td></td>
<td>Hold-down click</td>
<td>Turn device OFF</td>
</tr>
<tr>
<td></td>
<td>Up/Left</td>
<td>Go up in links or selection list</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Go down in links or selection list</td>
</tr>
</tbody>
</table>

In the browser, there is the browser log used for tracing what data is being sent between the client and the server. There are various log actions that can be chosen for specific tracing. Table 2 is the list of Log actions available.

Table 2: Log actions [2]

<table>
<thead>
<tr>
<th>Log actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clearing a Log</td>
<td>To clear the log window of all data.</td>
</tr>
<tr>
<td>Saving a Log</td>
<td>To save a log.</td>
</tr>
<tr>
<td>Cache Logging</td>
<td>To cache the log. The trace indicates cache hits, misses, and clearance.</td>
</tr>
<tr>
<td>Request Logging</td>
<td>To trace WML replies.</td>
</tr>
<tr>
<td>Reply Logging</td>
<td>To trace WML requests.</td>
</tr>
<tr>
<td>Source Logging</td>
<td>To trace your WML source. The source will be logged when a new URL is retrieved, or a card in the deck is switched.</td>
</tr>
<tr>
<td>WML/WAP Logging</td>
<td>To trace WML and WAP events.</td>
</tr>
<tr>
<td>Timer Logging</td>
<td>To view the browser related timer events.</td>
</tr>
<tr>
<td>Device Logging</td>
<td>To view the browser related device events.</td>
</tr>
</tbody>
</table>
Moreover, the browser supports UTF-8 (Unicode)\(^6\) thereby it can display many languages. By entering the Unicode, various languages such as Greek, Chinese, Japanese, or Korean can be displayed.

### 4.1.3. APPLICATION DESIGNER

This tool can be used to write and test WML and WMLScript code and see the results of this coding. The application designer only supports the creation of static pages. To make dynamic pages, wmllib (see section about Server Tools) or JAFFA (Java Servlet Library) can be used.

There are three windows in the application designer:

- Device
- Source
- Output

![Application Designer Interface](image_url)

**Figure 8: Application Designer Interface**

The WML and WMLScript can be entered in the source window. The output window shows compilation results and the device windows can be used to test an application.

---

The application designer can test the application without compiling the code, but to be more informative about any error, the code should be compiled first, because while using the device window to test the application, the log window can be used.

Moreover, in the application designer, it is possible to debug some other codes that are already compiled by using the Disassembly Tool. Figure 9 is the result of a disassembled bytecode file.

![Disassembled Bytecode File](image)

Figure 9: A disassembled bytecode file

4.1.4. SERVER TOOLSET

The server toolset can be used to create dynamic WAP applications. It contains various tools and library functions that support development of WAP applications. This includes WML and WMLScript compilers, Perl library with functions for creating WML source, and a WML verifier.

4.1.5. TROUBLESHOOTING

There is another problem in using the WapiDE, which is not listed in the “WapiDE User Guide.” If the set path in “autoexec.bat” is very long, then WapiDE cannot run. The solution is to reduce the path length in this file to be at most a couple of paths.
4.2. WAP TOOLKIT

4.2.1. SYSTEM REQUIREMENT

WAP Toolkit belongs to the Nokia Company. The current version is 1.3 beta. The system requirements include the following list:

- Pentium processor 266 MHz or faster
- 64 MB RAM (128 MB recommended)
- Windows NT 4.0 with Service Pack 3 or Windows 98
- 16-bit colour
- 1024*768 resolution
- 20 MB of hard disk space
- Java™ Run Time Environment 1.2.2 (or higher)
- Nokia 7110 simulator requires installation of the Nokia WAP Server

WAP Toolkit contains the following components:

- WML and WMLScript Editor
- Device Simulators
- WML Browser
- Script Interpreter and run-time libraries
- Debugging tools
- Sample applications
- Java Servlets
- Wireless Bitmap Editor

The system requirements for WAP Server are the following:

- Pentium Processor 266 MHz or faster
- 128 MB RAM (256 MB RAM recommended)
- Minimum 100 MB of free hard disk space
- Windows NT™ 4.0 with Service Pack 5
- Java™ Runtime Environment 1.2.2
- Java™ HotSpot 1.0.1
For browsers, there are three Nokia mobile phone models 6110, 6150, and 7110 that can be displayed. To change the phone model, simply select the model of the mobile phone in the Preference window. WAP Server was not installed for testing because there was no Windows NT™ with Service Pack 5 installed on the computer. The total of hard disk space required for installation is 18.3 MB (excluding the WAP Server).

4.2.2. SIMULATOR AND BROWSER

![WAP Toolkit Interface](image)

Figure 10: WAP Toolkit Interface

The functions of WAP Toolkit simulator are separated into the following stacks:

- Messages
- Variables
- History
- Bookmarks
- WML Deck
- Session
Clicking on the nametags at the bottom of the simulator window can change these functions. This allows users to know what functions in the toolkit are available.

The history stack shows:

- The last card visited will appear at the top of the stack
- The history of the current browser context only. Each time the browser context is re-initialised, the history stack is cleared.

![Image of Nokia WAP mobile phones]

**Figure 11: Nokia WAP mobile phones**

The size of the history stack can be changed under the General tab in the Preferences window.

In WML Deck stack, WML code can be opened into three types:

- Decoded WML
- Undecoded Bytecode
- Original Source

Decoded WML and original source are nearly the same. Undecoded bytecode is to read WML deck in a form of hexadecimal and ASCII code. The WML Deck stack is used for writing WML code to build an application.
4.2.3. TROUBLESHOOTING

There is only one problem with this application. It needs a lot of resources. Even if the computer systems for testing exceed the specified standard, operation is still found to be slow while using the Nokia 7110 mobile phone.7

4.3. UP.SDK

Phone.com Company has developed this application. The current version is Version 4.0 beta, and this was used in the present study.

4.3.1. SYSTEM REQUIREMENTS

The system and network requirements depend on the functionality used. However, most of the models that belong to Phone.com's carriers are compatible with UP.SDK version 3.0. Appendix A shows the list of Phone.com's carriers, mobile phone models, and some information for each model. The UP.SDK uses 10.2 MB of hard disk space.

The following basic requirements apply to most developers:

- To run UP.SDK simulator, a computer with an Intel (or compatible) processor running Windows 95 or Windows NT 4.0 (or later) is needed.
- To test WML services available on the Internet, such as example services provided by Phone.com, an Internet connection is needed.
- To test WML services on a stand-alone computer, an HTTP-compliant web server is needed.

UP.SDK consists of the following development tools:

- UP.Simulator
- Perl and C library functions
- C++ (Solaris) and COM (Windows) notification, digest, and fax libraries and tools
- Tools for requesting and installing SSL certificates
- Sample WML and WMLScript files and application source code
- Developer documents

7 There is a new version of this application. Its version is Version 2.0. The Nokia Company cut off other phones except 7110. Moreover, the problem mentioned above has been solved.
4.3.2. **UP.SIMULATOR**

UP.Simulator has two modes:

- **HTTP direct** – in this mode (default), the UP.Simulator loads WML directly from a web server, bypassing the UP.Link server. This mode is useful for initial testing of WML code because it does not need any special setup.

- **UP.Link** – in this mode, the UP.Simulator interacts with an UP.Link server just like a real UP.Phone. This mode requires access to an UP.Link server that needs to be registered before accessing it. This mode is used to test UP.Link features run on the server such as bookmarks, fax services, and notifications.

![Figure 12: UP.Simulator](image)

![Figure 13: UP.Simulator navigation key](image)
4.3.3. TROUBLESHOOTING

UP.SDK requires an Internet connection before running, otherwise an application will try to find the Internet connection and enter an infinite loop. To stop the application, the end task command in the Task window must be used.

4.4. OTHER WAP DEVELOPMENT TOOLKITS

At the present time, many companies have been developing WAP development tools. Each of them has their own functionalities and uniqueness. In addition to the WAP development tools described above, there is another tool in a web site called WAP.NET. Unfortunately, to retrieve this tool, payment is required.

4.5. COMPARISON OF WAP DEVELOPMENT TOOLKITS

The table below is a comparison of the WAP development tools considered above.

Table 3: Comparison of WAP development tools

<table>
<thead>
<tr>
<th>System Requirements</th>
<th>WapIDE SDK v2.1</th>
<th>WAP Toolkit v1.3beta</th>
<th>UP.SDK v4.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform</td>
<td>Windows 95/98 or Windows NT 4.0</td>
<td>Windows 98 or Windows NT 4.0 with Service Pack 3</td>
<td>Windows 95/98 or Windows NT 4.0</td>
</tr>
<tr>
<td>Hard disk space (excluding WAP server)</td>
<td>10 MB</td>
<td>20 MB</td>
<td>11 MB</td>
</tr>
<tr>
<td>RAM</td>
<td>32 MB</td>
<td>64 MB (128 MB recommended)</td>
<td>64 MB</td>
</tr>
<tr>
<td>Internet connection</td>
<td>Optional</td>
<td>Optional</td>
<td>Required</td>
</tr>
<tr>
<td>WAP server</td>
<td>Required</td>
<td>Required</td>
<td>None</td>
</tr>
<tr>
<td>Other required application</td>
<td>None</td>
<td>Java Run Time Environment 1.2.2 or higher</td>
<td>None</td>
</tr>
<tr>
<td>Product Components</td>
<td>WML Browser</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Device simulator</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>WML and WMLScript editor</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Java Servlet Library</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Sample applications</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Models</td>
<td>R320</td>
<td>6110, 6150, and 7110</td>
</tr>
<tr>
<td></td>
<td>Other features</td>
<td>Disassembly tool</td>
<td>Wireless Bitmap Editor</td>
</tr>
</tbody>
</table>
From Table 3, WapiDE needs few computer resources while the WAP Toolkit requires a lot of resources. WapiDE can run in a standard computer while WAP Toolkit must run in a high-end computer because WAP server from the WAP Toolkit requires 256 MB of RAM. Even if WapiDE and WAP Toolkit require a WAP server, simulator can run without the WAP server if the WML deck is in the same domain. UP.SDK does not need many resources but it requires an Internet connection to Phone.com’s developer website. In this web site, there is a WAP server and some example applications so that developers do not need to install a WAP server as with the other tools mentioned above.

Each simulator can change the mobile phone models appearance by loading a new device. However, for Ericsson, there is only one available model, and Nokia has three models. UP.SDK is the most compatible with many carriers that have joined with Phone.com. The list below is the current available “skins” for mobile phone views:

- Alcatel One Touch
- Ericsson R280LX
- Mitsubishi MobieAccess
- Mitsubishi T250
- Motorola i1000plus
- Motorola Timeport L7389
- Samsung Duette

To see more details about Phone.com’s carriers, go to Appendix A.

5. WAP APPLICATION

The WAP application to be developed in this project concerns a “Stock Exchange.” Many people who own mobile phones are working in business. They may be working for private companies or government institutions. Many companies rely on the stock exchange to increase their income. The purpose of this application is to stimulate people to use this technology more. This application can show that WAP technology is going to be an important structure of telecommunication in the future.
The basic function of this application is to perform functions that allow the user to know whenever the current price changes to a specific price that a user specifies in their list of preferences. For example, the user can set the maximum price at $10 and this could be the price at which the user is willing to sell. Whenever the price of the shares belonging to the user equals $10 or more, the mobile phone will provide a stimulus to the user in some way, such as an alert message or toning.

There are a lot of mobile users so that a powerful database is needed. In this thesis, an MS Access database is used since the application is only a prototype. In fact that many companies are using SQL or Oracle databases which offer better performance than an MS Access database. The reason that MS Access database has been chosen is it is suitable for a small database, easy to manage, and suitable with Microsoft Windows. Two tables are needed to store the basic information for each customer.

- **Profile table** which stores customer names, phone numbers, and a password
- **Setting table** which stores share names of interest to the customer, and associated trigger prices (maximum and minimum prices)
- **Share Price table** which stores all share names, and share prices (including maximum, minimum, median, and current prices for a specific time period such as a day)

The customer and trigger tables are static databases but the share price table is a dynamic database. Therefore, the Push function is used to send share prices from a stock exchange website to the user terminal at a specific time, such as every 5 minutes. However, many quote websites have protection for their databases from an unauthorised user and it is impossible to retrieve data of share prices directly from their databases. Therefore, dynamic dummy share prices are needed and they will be stored into the same domain as the WAP application and it uses a Push function to send this data to the database.
5.1. AN IMPLEMENTATION OF THIS APPLICATION

Languages used in this application are WML, ASP, SQL, and Java. Each language has its own specific functionality that is listed below:

- **WML** – It is the main language of this application. Without it, a WAP mobile phone cannot understand what it is being sent from a server.
- **ASP** – It is a server-side programming language. It is used to handle an event that a user generates, and connects to a database. There are alternative languages that can be used such as PHP, Cold Fusion, or CGI. However, it is easy to find resources for ASP, which belongs to Microsoft Corporation.
- **SQL** – It is a language to connect to a database. ASP cannot connect to the database directly. It uses an ADO object, which will be changed into SQL automatically. Some parts of this application were written in SQL.
- **Java** – The heart of this application. It retrieves data from a quote website.
- **VBScript** – It is a part of ASP. In fact, ASP can be written with either JavaScript or VBScript. However, VBScript is easy to understand and manage.

This application involved the use of Java, which works as a part of IIS Extension stored in \c:\winnt\java\trustlib, to retrieve data from a quote website. It stores the results of an enquiry in variables of a server, compares share prices with the prices set in the database, and sends a warning message, if necessary. It does not store share prices, which have been retrieved from the quote website into the database. Therefore, a Share Price table is not needed for this application. The quote website chosen to use in this application is Yahoo. The URL is http://au.finance.yahoo.com. It is written in a very clean and easy manner to allow a user to quickly find share prices.

Moreover, Push cannot be used with a simulator because there is no simulator has a device address. To test an application with Push, a real wireless device with WAP-enable must be used.

5.2. SYSTEM REQUIREMENTS

To run ASP smoothly, a Windows NT platform is recommended. The list below described the system used for the final application.
• Pentium II 400 MHz
• 64 MB RAM (128 MB recommended)
• 6.4 GB hard disk
• Windows 2000 Professional
• IIS 5.0, ADO 2.5, and ASP 3.0 installed
• Java 2 SDK Standard Edition v1.3
• Microsoft® Virtual Machine

Windows 2000 Professional comes with IIS 5.0, ADO 2.5, and ASP 3.0 thereby it is a suitable platform to set up an ASP server. To get the latest version of Microsoft® Virtual Machine, please refer to http://www.microsoft.com/java.

5.3. SETTING UP ASP SERVER

If a platform used to run this application is not Windows 2000, then IIS 5.0, ADO 2.5, and ASP 3.0 need to be installed. For more details, please refer to Chapter 2 in "Beginning ASP Databases." [4]

In addition, after an installation has been finished, configuring MIME types is required. MIME (Multipurpose Internet Mail Extensions) is a specification for the format of data that can be sent over the Internet. When a server sends data in response to a request that it receives, it sends a MIME type with it. This MIME type can also be explicitly set by an application. Normally, the file extension of the requested file is associated with a MIME type and so the server automatically issues the correct MIME type. Then, when a browser receives information from the server, it checks its MIME type to see what to do with it. If, for example, it sees that the data has a MIME type of "image/vnd.wap.wbmp" then it knows to display it as a picture. The MIME types accepted by the browser can also be discovered dynamically by MIME type through determining the supported MIME types of the device. [1] To configure MIME types, please refer to Chapter 2 in "Professional WAP with WML, WMLScript, ASP, JSP, XML, XSLT, WTA, Push and VoiceXML."
5.4. APPLICATION ARCHITECTURE

Figure 14: The Structure of Application

Figure 14 shows the application structure. It is in user interface order. The front-page of an application is contained in the index.wml file. Therefore, the index.wml icon is put at the top of this structure. To see more details, please refer to Appendix C.

In the application developed for this project, there were three types of file used, which are wml, asp, and class files respectively. A WML file contains generic tag programming which is similar to HTML tags. An ASP file contains server-side programming that concerns connection to a database. Finally, the class file contains Java class data, which is compiled using the Java 2 SDK.
To run this application, all Java class files must be placed in c:\winnt\java\trustlib directory and declared as public if necessary. Moreover, Microsoft® Virtual Machine must be installed. To see every file in this application, please refer to Appendix B.

5.5. FUNCTIONS

This section is written in order to describe only the necessary functions required for this application. To learn how to programme, please refer to the Reference section. The functions in this section are listed in order of appearance.

5.5.1. TIMER

There are two decks, which use this function. One is index.wml and the other is today.asp. Timer is a tag function in WML that will do something after a specified amount of time as coded in the file.

![Figure 15: The Front-page of Application](image)

Figure 15 shows two screenshots from this application. The front-page is on the left hand side. If a user does not press the accept button, then it will automatically change to the Menu page after expiry of the timer.

In line 6 of index.wml, the property of the timer is set to 20. This means it will automatically change to the Menu deck after 2 seconds if a user does not press the accept button. This tag is located in line 172 in today.asp as well. Its value is set at 600. This means that the today.asp file will go to redirect.asp after one minute.
5.5.2. POSTING DATA

From the code above, the “go” tag will send two queries to changepasswd.asp. One is the *username* and the other is *passwd*. A method to send these queries is the GET method. There are two ways of posting data through the Internet that are known as GET and POST methods. When the GET method is used, data being sent to a server is appended to the end of a URL. When the POST method is used, this data is passed in the body of the message. The POST method has advantages over the GET method. Since the length of URLs is restricted on some servers and browsers (typically it may be 127 characters although it is dependent on the server, gateway or browser). Including the data in the URL of the request is less secure as it can be examined more easily. Moreover, some characters within a URL have special meaning and would need to be encoded. Data included in the body of a request does not require encoding. Sending data to the server using the POST method requires only a little more work to implement than the GET method, and has the added benefits of security and unlimited data size. [1] However, there is no WAP mobile phone or simulator that currently supports the POST method.

5.5.3. OBTAINING STOCK PRICE

To get a stock price from a quote website, IIS server extensions are used. The reason is that it is faster than using ASP, which is an interpreted language. There are three ways to make IIS server extensions.

1. An ISAPI (Internet Service Application Program Interface) function in C/C++
2. An ActiveX object in C/C++, Visual Basic, or Java
3. A Java class, which is called directly

In this thesis, the third option was selected. To obtain more information, please refer to http://hotwired.lycos.com/webmonkey/programming/asp/. [3]
This function is located in the today.asp file. There is a function called GetStockPriceFromYahoo. To see a flowchart of this function, please refer to Appendix D. When a user calls the today.asp file, it would create a connection between itself and a database and retrieve quote codes belonging to the user. Then, it would call GetStockPriceFromYahoo function and posting the quote code. This function would call StockGrabber (Java class) from IIS server extensions (in c:\winnt\java\trustlib). StockGrabber would generate an Internet connection to the Yahoo website and obtain an HTML file which depends on what quote that the user has selected. If there is no problem with retrieving HTML file, then it will return back values of the relevant share price. GetStockPriceFromYahoo will check whether the share price received is a real number or a fraction. If it is the fraction, then it will extract only the numerator and the denominator and then divide them to produce a real number. Then, it will change the resulting variable to be of type single precision. This is an important part because, in the database, share prices are stored in single precision. If the type of variable is not changed to be single precision, then the results will be wrong.

5.6. LIMITATIONS

This application is essentially a prototype application and consequently has a number of significant limitations.

1. WAP phone simulators have different procedures to input alphabetic characters. Some of them can choose directly whether an uppercase or lowercase letter can be inputted and others cannot. Therefore, this application has been made to change every username and password to lowercase in order to avoid this problem. On the other hand, a user cannot choose to have a name with the first letter as an uppercase letter.

2. In GetStockPriceFromYahoo in the today.asp file, there is no function to search </sup>, <sub> and </sub> tags. It searches only <sup> tag. Therefore, if there are no tags of this type, then there will be an error. Moreover, there is no function to handle errors for this event.
3. Ideally, this application would be done with push technology. In WAP, push technology can send information to a user if a WAP server knows a device address. The most common types of device addresses are:

- **IPv4 addresses**: This is a common 32-bit Internet address, normally expressed as a dotted quad of the form `aaa.bbb.ccc.ddd`.

- **MSISDN addresses**: This is the “Mobile Station International Subscriber Directory Number” or, in other words, a normal mobile phone number of the form +91-98450-00007, which contains the mobile country code, the mobile network code, and the subscriber number in the network. The format of this address can have minor variations depending on whether the network is of type GSM, CDMA, IS-136, AMPS, PDC, PHS, IDEN or any of the other bearer network types defined in the WDP 1.2 specifications.

- **IPv6 addresses**: This evolving addressing format of IP would address the shortage of IPv4 addresses that we are currently facing. IPv6 addresses are 128 bits in length and the addressing space is so large that even at double the current population of the earth, each individual can still have the square of a thousand trillion addresses reserved for themselves. Despite all the interoperability problems with IPv4 that are there, if IPv6 addresses are used sometime in the near future, WAP is ready for them. [1]

However, a simulator does not have any dummy address. Therefore, to accomplish this, a real WAP mobile phone must be used. To solve this problem, a timer function and redirecting page are used. The timer function will go to the redirect.asp file at every a minute and redirect.asp file will go to today.asp file immediately after it is accessed.

4. The quote website chosen in this thesis is Yahoo in Australia (http://au.finance.yahoo.com). If a user puts in a quote code, which is not in Australia, i.e. AOL (USA), then this application cannot find a share price and will return an error. Hence, to support a user that is not in Australia, this application must use other quote websites, e.g. http://quote.yahoo.com (Yahoo in USA).
5. This application was only tested in Windows 2000 Professional. It has never been tested on any other platform. If this application has to be distributed to other platforms, then that platform must have an ASP server. Normally, Microsoft supports ASP. Nowadays, there are two companies who provide ASP as an add-on to web servers other than Microsoft which are Chili!Soft (http://www.ChiliSoft.com) and HalcyonSoft (http://www.halcyonsoft.com). In addition, if the platform is not Windows 2000 or Windows NT, the directory where Java class files must be placed will be different.

6. This application still has no function to handle two particular errors. When a user tries to change or delete a share price, which is not in a database, it will return an error. Another error is that there is no function to detect whether the user has entered all the data required for inputting, changing, or deleting in the database in Change Price, Delete Price, New Price, Register, Login, and Change Password pages. This could be achieved using WMLScript.

7. A database used in this application, which is MS Access database, is not good enough for a big commercial company. SQL Server or Oracle offer better performance for managing large commercial size databases.

8. A user cannot change the length of time for which today.asp will be refreshed.

9. This application cannot be used for real share market because most of the quote websites are delayed about 20 minutes from real prices and this could be significant for a real user.

10. If a real WAP mobile phone were to be tested in this thesis, then there would need to be a device address. With this address, this application can generate a voice call by using the WTAI function. Therefore, a user does not need to look at a screen to check whether a share price reaches the trigger prices. The syntax for WTAI calls is:

   `<anchor title="call">`<br>
   `<go href="wtai://wp/mc;15551234567"/>`<br>
   15551234567<br>
   `</anchor>`
Where 15551234567 is a number of mobile phone the user wants to generate a call. The anchor tag can be replaced by:
<do type="accept"><go href="wtai_url"/>wtai link</do>
or
<option onpick="wtai_url">wtai link</option>
In addition, a timer tag can be used as well.

11. The database used in this application is stored in a real number format not fractional numbers. Typically, most share prices shown in share markets are fractional, thereby a user will encounter some difficulties in interpreting the numbers.

12. This application does not check whether a new share, which a user creates, is available at a quote website.

6. RESULT

The purpose of this thesis was to develop an understanding of WAP and to demonstrate WAP technologies using a simple application. In this section, the application will be demonstrated using screenshots captured from the application.

The application starts at the front-page as shown in screenshot number 1. This page times out after 2 seconds. Therefore, if a user does not press the accept button, it will automatically go to the Menu page which contains two links. One is a Login link and the other is the Register link. For registration, it makes no difference whether a user enters an uppercase or lowercase alphabet letter because this application will change every alphabetic character into lowercase. Screenshot number 2 is the menu page.
The screenshot number 3 shows the Login page. This page is the same as the Register page, as well. After logging in, if there is no problem with the username or password, then it will display the next page as shown in screenshot number 4. There is a greeting message to let the user know that a correct account name and password have been entered. If the user enters either a wrong username or password, then it will display a warning message and let the user go back and correct them. Now, there are 4 options, which can be chosen:

- Today’s Price – Check current share prices that belongs to a user
- Set Price – Add, change, or delete share prices from a database
- Change Password – Change a password
- Log out – Exit this application

The WAP mobile phone has quite a small screen and so the Log out option cannot be seen in the fourth screenshot. To see this option, scrolling or using the directional button on the mobile phone must be used. Screenshot number 5 and 6 demonstrate a registration. For example, if Professor Richard Harris wants to use this application, he can register as well using this method.

Screenshots 7 and 8 show current share prices and share prices set by the user. There are two triggers for share prices, which the user can set. One is a maximum price and the other is a minimum price. Whenever the current share price reach either the maximum or the minimum prices, then Today’s Price page will display ‘Max’ or ‘Min’ to let the user know. (See for example, screenshot 10 on next page.)
For example, if TLS' price is changed from $6.3 to $6.2 when the current price is $6.296, then, in Today's Price page, then neither ‘Max’ nor ‘Min’ will be displayed. This example is shown in screenshot numbers 9 and 10.

Screenshot number 11 shows a menu in the Setting page. There are four options that can be selected. Screenshot number 12 is the Change Price page. The user must input all data, which includes the Share Code, Max Price and Min Price.
The two screenshots above show the process of changing the share price. After inputting all data shown in screenshot number 12, this application will ask the user to confirm a change of share price. A ‘No’ is placed as the first (default) option as a double check. If the user confirms that this is the correct decision, then Change Result will show the results.

Screenshot number 15 and 16 show how to create a new share price. From the above screenshots, the AMP share option is created and the maximum price to $18.

Screenshot number 19 and 20 show how to delete a share. From the above screenshots, the AMP share has been deleted.
When the user does not want a particular share, e.g. AMP, that share can be deleted from the menu using the ‘Current Setting’ page. Four screenshots above demonstrate how to delete the AMP share option from an account.

Another feature of this application is a user option to change a password. It can be changed from the main page of the application. When the user wants to change the password, three passwords must be entered correctly. The first password is the current password. The second and third one are the new password. The user must enter exactly the same for both new passwords.

Finally, when the user wants to exit this application, the user can simply choose log out from the main menu. Then the application will show the following messages.
There were some difficulties that were encountered while using some simulators and toolkits from various companies. The following list of problems was found whilst testing them with this application.

**WapIDE SDK** *(Ericsson)*

This toolkit is not compatible with IIS. It seems to run only its own WAP server, which is called Xitami. So, this toolkit is not practical for testing and development with ASP.

**WAP Toolkit** *(Nokia)*

This toolkit is also not compatible with IIS. It needs its own WAP server that Nokia offer only on a 30-day trial but this thesis needs more time than that.

**UP.SDK** *(Phone.com)*

This emulator is compatible with IIS. There is no difficulty with installing and using it. The only problem is that it requires an Internet connection. It is based on only WML 1.1 so this application cannot be programmed with the full functionality specified in WML 1.2.

**MADK** *(Motorola)*

This emulator requires a lot of installation and there are several bugs in this application. This program cannot run on the testing computer.

**WAPman** *(EdgeMatrix)*

This emulator is a stand-alone and very small application. It is compatible with IIS. The only problem is that it is not realistic. The GUI is too fancy to be a real mobile phone.

**EzWAP** *(EZOS)*

This emulator is small. However, it lacks a realistic interface. It is compatible with WML 1.2.
**Deck-It WML Previewer** (PyWeb.com)

This emulator seems to be the best for testing the application. It is stand-alone and a small application. It is compatible with IIS and based on WML 1.2. However, the keypad cannot be pressed. Data must be entered directly through a keyboard from a computer.

**7. CONCLUSION**

This application demonstrates the use of WAP technology in a simple way. There are a lot of applications continuously being developed. This application could be a major advantage for helping a businessman if a database in a quote website is updating in real time. ASP is not the only solution to develop this application. There are alternative solutions that could have accomplished the same result. For example, using JSP or PHP, which are more compatible with other platforms than ASP. There are some more features that can be added to this application: For example, if a share market supports this application, then the user can set a trigger from a share price to automatically sell or buy shares with an amount set by the user. In addition, there are a few of WAP mobile phones, which have large screens. If the screen standard of WAP mobile phone improves, then this will have a significant affection on users. For example, most WAP mobile phones could have screens that are comparable to the size of the phone, then this application may be developed to have some graphs or to show records previous prices in the past few hours or even 30 days ago. Furthermore, it could be programmed to predict share prices if it were developed further.
### LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADO</td>
<td>ActiveX Data Objects</td>
</tr>
<tr>
<td>ASP</td>
<td>Active Server Pages</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CGI</td>
<td>Common Gateway Interface</td>
</tr>
<tr>
<td>ECMA</td>
<td>European Computer Manufacturers Association</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transport Protocol</td>
</tr>
<tr>
<td>IIS</td>
<td>Internet Information Server</td>
</tr>
<tr>
<td>JSP</td>
<td>Java Server Pages</td>
</tr>
<tr>
<td>MIME</td>
<td>Multipurpose Internet Mail Extensions</td>
</tr>
<tr>
<td>MMI</td>
<td>Man Machine Interface</td>
</tr>
<tr>
<td>MSISDN</td>
<td>Mobile Station International Subscriber Directory Number</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>PWS</td>
<td>Personal Web Server</td>
</tr>
<tr>
<td>QOS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>ROM</td>
<td>Read-Only Memory</td>
</tr>
<tr>
<td>SGML</td>
<td>Standardised Generalised Markup Language</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Socket Layer</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transmission Control Protocol/Internet Protocol</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>VBA</td>
<td>Visual Basic Application</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>WAE</td>
<td>Wireless Application Environment</td>
</tr>
<tr>
<td>WAP</td>
<td>Wireless Application Protocol</td>
</tr>
<tr>
<td>WDP</td>
<td>Wireless Datagram Protocol</td>
</tr>
<tr>
<td>WML</td>
<td>Wireless Markup Language</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>WMLScript</td>
<td>Wireless Markup Language Script</td>
</tr>
<tr>
<td>WSP</td>
<td>Wireless Session Protocol</td>
</tr>
<tr>
<td>WTA</td>
<td>Wireless Telephony Application</td>
</tr>
<tr>
<td>WTLS</td>
<td>Wireless Transport Layer Security</td>
</tr>
<tr>
<td>WTP</td>
<td>Wireless Transaction Protocol</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
GLOSSARY

Bytecode – content encoding where the content is typically a set of low-level opcodes and operands for a targeted hardware (or virtual) machine.

Card – a single WML unit of navigation and user interface. May contain information to present to the user, instructions for gathering user input, etc.

Client – a device or application that initiates a request for a connection with a server.

Content – subject matter (data) stored or generated at an origin server. Content is typically displayed or interpreted by a user agent in response to a user request.

Deck – a collection of WLM cards. A WML deck is also an XML document. May contain WMLScript.

JavaScript – a standard language that can be used to add dynamic behaviour to HTML documents. JavaScript is one of the originating technologies of ECMAScript.

Man-Machine Interface – a synonym for user interface.

Origin Server – a server on which a given resource resides or is to be created. Often referred to as a web server or an HTTP server.

SGML – the Standardised Generalised Markup Language is a general-purpose language for domain-specific mark-up languages.

Terminal – a device providing the user with user agent capabilities, including the ability to request and receive information. Also called a mobile terminal or mobile station.

User Agent – a user agent is any software or device that interprets content (e.g., WML). This may include textual browsers, voice browsers, search engines, etc.

WMLScript – a scripting language used to program the mobile device. WMLScript is an extended subset of the JavaScript scripting language.

XML – the Extensible Markup Language is a World Wide Web Consortium (W3C) proposed standard for Internet mark-up languages, of which XML is one such language. XML is a restricted subset of SGML.
REFERENCES


BIBLIOGRAPHY

APPENDIX A: UP.BROWSER PHONE REFERENCE

Phone.com’s carrier and OEM licenses have publicly announced the phones listed in the table below.8

<table>
<thead>
<tr>
<th>OEM</th>
<th>Model</th>
<th>Type</th>
<th>Shipping</th>
<th>Location</th>
<th>UP.Browser Version</th>
<th>Device ID</th>
<th>Image Display</th>
<th>How to Access Browser Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alcatel</td>
<td>OneTouch</td>
<td>GSM</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>ALAZ</td>
<td>Yes</td>
<td>Press-hold # key</td>
</tr>
<tr>
<td>Alcatel</td>
<td>OneTouch DB</td>
<td>GSM</td>
<td>Yes</td>
<td>Europe</td>
<td>4.1.8</td>
<td>SH</td>
<td>Yes</td>
<td>Press-hold # key</td>
</tr>
<tr>
<td>Audiovox</td>
<td>CDM-9000</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>TA01</td>
<td>No</td>
<td>Press Roll Key</td>
</tr>
<tr>
<td>Casio</td>
<td>C303CA</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>CA11</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Denso</td>
<td>C202DE</td>
<td>CDMA</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>DN01</td>
<td>Yes</td>
<td>Press Set key</td>
</tr>
<tr>
<td>Denso</td>
<td>Sprint TouchPoint</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>DS12</td>
<td>Yes</td>
<td>Select Help function</td>
</tr>
<tr>
<td>Denso</td>
<td>Sprint TouchPoint</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>DS13</td>
<td>Yes</td>
<td>Select Help function</td>
</tr>
<tr>
<td>Ericsson</td>
<td>R280</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>ERK0</td>
<td>Yes</td>
<td>Press Menu key</td>
</tr>
<tr>
<td>Hanwha</td>
<td>S71</td>
<td>CDMA</td>
<td>Yes</td>
<td>Korea</td>
<td>3.1</td>
<td>HW01</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Hitachi</td>
<td>C201H</td>
<td>CDMA</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>HI01</td>
<td>Yes</td>
<td>Press F key</td>
</tr>
<tr>
<td>Hitachi</td>
<td>C302H</td>
<td>CDMA</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>HI11</td>
<td>Yes</td>
<td>Press EZ key</td>
</tr>
<tr>
<td>Hyundai</td>
<td>D-3</td>
<td>CDMA</td>
<td>Yes</td>
<td>Korea</td>
<td>3.1</td>
<td>HD03</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Hyundai</td>
<td>HGC-R201</td>
<td>CDMA</td>
<td>Yes</td>
<td>Korea</td>
<td>3.1</td>
<td>HD01</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Hyundai</td>
<td>P-21</td>
<td>CDMA</td>
<td>Yes</td>
<td>Korea</td>
<td>3.1</td>
<td>HD02</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Kyocera</td>
<td>C307K</td>
<td>CDMA</td>
<td>Yes</td>
<td>Korea</td>
<td>3.1</td>
<td>KC11</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

8 Phone.com, Inc. UP.Browser Phone Reference, [online], http://developer.phone.com [Accessed 30 October 2000]
<table>
<thead>
<tr>
<th>Brand</th>
<th>Model</th>
<th>POC</th>
<th>Yes/No</th>
<th>Region</th>
<th>Version</th>
<th>CoPO</th>
<th>Yes/No</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kyocera</td>
<td>TK01</td>
<td>PDC</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>KCT2</td>
<td>Yes</td>
<td>Press EZ key</td>
</tr>
<tr>
<td>Kyocera</td>
<td>TK02</td>
<td>PDC</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>KCT1</td>
<td>Yes</td>
<td>Press EZ key</td>
</tr>
<tr>
<td>LG</td>
<td>LGC-840f</td>
<td>CDMA</td>
<td>Yes</td>
<td>Korea</td>
<td>3.1</td>
<td>LG09</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>LG</td>
<td>LGI-2100</td>
<td>CDMA</td>
<td>Yes</td>
<td>Korea</td>
<td>3.1</td>
<td>LG02</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>LG</td>
<td>LGP-7300f</td>
<td>CDMA</td>
<td>Yes</td>
<td>Korea</td>
<td>3.1</td>
<td>LG01</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>LG</td>
<td>SD-500</td>
<td>CDMA</td>
<td>Yes</td>
<td>Korea</td>
<td>3.1</td>
<td>LG10</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Mitsubishi</td>
<td>MA120</td>
<td>CDPD</td>
<td>Yes</td>
<td>North America</td>
<td>2.0</td>
<td>SP01</td>
<td>No</td>
<td>Press Menu key</td>
</tr>
<tr>
<td>Mitsubishi</td>
<td>T250</td>
<td>CDPD</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>T250</td>
<td>Yes</td>
<td>Press Menu key</td>
</tr>
<tr>
<td>Motorola</td>
<td>Accompli 6188</td>
<td>GSM</td>
<td>Yes</td>
<td>Asia Pacific</td>
<td>4.1</td>
<td>MOT-CB</td>
<td>Yes</td>
<td>Tap on the Menu</td>
</tr>
<tr>
<td>Motorola</td>
<td>i500+, i700+,</td>
<td>iDEN</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>MO01</td>
<td>No</td>
<td>Press Menu key</td>
</tr>
<tr>
<td>Motorola</td>
<td>StarTac ST786x,</td>
<td>CDMA</td>
<td>Yes</td>
<td>Korea, Latin</td>
<td>3.1</td>
<td>MC01</td>
<td>No</td>
<td>Press FCN-1</td>
</tr>
<tr>
<td>Motorola</td>
<td>Talkabout V8162</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>MCC9</td>
<td>No</td>
<td>Press Func-Menu</td>
</tr>
<tr>
<td>Motorola</td>
<td>Talkabout V2267</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>MCCC</td>
<td>No</td>
<td>Press Func-Menu</td>
</tr>
<tr>
<td>Motorola</td>
<td>Talkabout V2288</td>
<td>GSM</td>
<td>Yes</td>
<td>Europe, Australia</td>
<td>4.1.8</td>
<td>MOT-C4</td>
<td>Yes</td>
<td>Press-hold Menu</td>
</tr>
<tr>
<td>Motorola</td>
<td>Talkabout V8162</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>MCC9</td>
<td>No</td>
<td>Press Func-Menu</td>
</tr>
<tr>
<td>Motorola</td>
<td>Timeport, P7389</td>
<td>GSM</td>
<td>Yes</td>
<td>Europe, North America, Australia</td>
<td>4.0</td>
<td>MOT-CB</td>
<td>Yes</td>
<td>Press-hold Menu</td>
</tr>
<tr>
<td>NeoPoint</td>
<td>NP1000</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.0</td>
<td>IG01</td>
<td>Icons only</td>
<td>Press-hold M key</td>
</tr>
<tr>
<td>Nokia</td>
<td>6185</td>
<td>CDMA</td>
<td>No</td>
<td>North America</td>
<td>3.1</td>
<td>NK00</td>
<td>Yes</td>
<td>Press Power key</td>
</tr>
<tr>
<td>Panasonic</td>
<td>C308P</td>
<td>CDMA</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>MA11</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Brand</td>
<td>Model</td>
<td>Technology</td>
<td>Feature</td>
<td>Region</td>
<td>Version</td>
<td>Feature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>------------</td>
<td>---------</td>
<td>--------------</td>
<td>---------</td>
<td>----------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Panasonic</td>
<td>D2</td>
<td>PDC</td>
<td>Yes</td>
<td>Japan</td>
<td>4.0</td>
<td>D2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Panasonic</td>
<td>D305P</td>
<td>PDC</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>MAC1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Panasonic</td>
<td>P-PAT</td>
<td>PHS</td>
<td>Yes</td>
<td>Japan</td>
<td>4.0</td>
<td>P-PAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualcomm</td>
<td>QCP-860, CQP-1960</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America, Latin America</td>
<td>3.0</td>
<td>QC31</td>
<td>No</td>
<td>Press i key</td>
</tr>
<tr>
<td>Qualcomm</td>
<td>QCP-1900, QCP2700</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America, Latin America</td>
<td>3.0</td>
<td>QC12</td>
<td>No</td>
<td>Press i key</td>
</tr>
<tr>
<td>Qualcomm</td>
<td>QCP-2760</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America, Latin America</td>
<td>3.0</td>
<td>QC32</td>
<td>No</td>
<td>Press i key</td>
</tr>
<tr>
<td>Samsung</td>
<td>Duette</td>
<td>CDPD</td>
<td>Yes</td>
<td>North America</td>
<td>2.0</td>
<td>SH</td>
<td>No</td>
<td>Press Menu key</td>
</tr>
<tr>
<td>Samsung</td>
<td>SCH-3500</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>SC01</td>
<td>Yes</td>
<td>Press-hold Menu key</td>
</tr>
<tr>
<td>Samsung</td>
<td>SCH-6100</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>SC03</td>
<td>Yes</td>
<td>Press-hold Menu key</td>
</tr>
<tr>
<td>Samsung</td>
<td>SCH-8500</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>SC02</td>
<td>Yes</td>
<td>Press-hold Menu key</td>
</tr>
<tr>
<td>Samsung</td>
<td>SGH-800</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>SG08</td>
<td>Yes</td>
<td>Press-hold # key</td>
</tr>
<tr>
<td>Sanyo</td>
<td>C304A</td>
<td>CDMA</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>SY11</td>
<td>Yes</td>
<td>Press EZ key</td>
</tr>
<tr>
<td>Sanyo</td>
<td>D301SA</td>
<td>PDC</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>SYC1</td>
<td>Yes</td>
<td>Press EZ key</td>
</tr>
<tr>
<td>Sanyo</td>
<td>SCP-4000</td>
<td>CDMA</td>
<td>Yes</td>
<td>North America</td>
<td>3.1</td>
<td>SY01</td>
<td>Yes</td>
<td>Select Help function</td>
</tr>
<tr>
<td>Sanyo</td>
<td>TS01</td>
<td>PDC</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>SYT1</td>
<td>Yes</td>
<td>Press EZ key</td>
</tr>
<tr>
<td>Siemens</td>
<td>C35 M35</td>
<td>GSM</td>
<td>Yes</td>
<td>Europe</td>
<td>4.1.8</td>
<td>SIE-C3I</td>
<td>Yes</td>
<td>Select at menu function</td>
</tr>
<tr>
<td>Siemens</td>
<td>S35</td>
<td>GSM</td>
<td>Yes</td>
<td>Europe</td>
<td>4.1.8</td>
<td>SIE-S35</td>
<td>Yes</td>
<td>Select at menu function</td>
</tr>
<tr>
<td>Sony</td>
<td>C305SN</td>
<td>CDMA</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>SN11</td>
<td>Yes</td>
<td>Press Jog dial</td>
</tr>
</tbody>
</table>

- 53 -
<table>
<thead>
<tr>
<th>Toshiba</th>
<th>Model</th>
<th>Technology</th>
<th>Mode</th>
<th>Country</th>
<th>Version</th>
<th>Module</th>
<th>Yes/No</th>
<th>Press EZ key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toshiba 701G</td>
<td>PDC</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>TS11</td>
<td>Yes</td>
<td>Press EZ key</td>
<td></td>
</tr>
<tr>
<td>Toshiba C301T</td>
<td>CDMA</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>TS11</td>
<td>Yes</td>
<td>Press EZ key</td>
<td></td>
</tr>
<tr>
<td>Toshiba D302T</td>
<td>PDC</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>TOxx</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toshiba TT01</td>
<td>PDC</td>
<td>Yes</td>
<td>Japan</td>
<td>3.1</td>
<td>TST1</td>
<td>Yes</td>
<td>Press EZ key</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX B: THE FILES IN WAP APPLICATION

Before distributing this application, it is necessary to ensure all files are available.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adovbs.inc</td>
<td>ADO file keeps information of ADO objects.</td>
</tr>
<tr>
<td>changeaction.asp</td>
<td>A file that changes maximum and minimum share prices in a database.</td>
</tr>
<tr>
<td>changeconfirm.asp</td>
<td>A file that shows confirmation messages to a user before making changes in a database.</td>
</tr>
<tr>
<td>changepasswd.asp</td>
<td>A file that displays a user interface of changing password. Two new passwords must be exactly the same to change the password.</td>
</tr>
<tr>
<td>changepasswdact.asp</td>
<td>A file that changes user password in a database.</td>
</tr>
<tr>
<td>cursor.asp</td>
<td>A file that is used to check cursor type and lock type parameters, which use to create a connection to a recordset in a database. This file must be displayed in HTML browser.</td>
</tr>
<tr>
<td>deleteaction.asp</td>
<td>A file that deletes a share price which a user requests.</td>
</tr>
<tr>
<td>deleteconfirm.asp</td>
<td>A file that shows confirmation messages to a user before deleting share price that is set by the user in the database.</td>
</tr>
<tr>
<td>index.wml</td>
<td>The first file that displays user interface in this application.</td>
</tr>
<tr>
<td>login.asp</td>
<td>A file that displays login page which a user must input a username and password.</td>
</tr>
<tr>
<td>logout.wml</td>
<td>A file that cleans a username and password, which are cached by a browser.</td>
</tr>
<tr>
<td>main.asp</td>
<td>A file that shows greeting messages if a username and password are correct or shows error messages if there is something wrong.</td>
</tr>
<tr>
<td>newshare.asp</td>
<td>A file that creates a new share price set by a user.</td>
</tr>
<tr>
<td>readme.txt</td>
<td>A file that contains information how to distribute this application.</td>
</tr>
<tr>
<td><strong>File</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>redirect.asp</code></td>
<td>A file that is used to clear the cache in a browser before redirecting to today.asp.</td>
</tr>
<tr>
<td><code>register.asp</code></td>
<td>A file that displays user interface to input a username and password for a new customer.</td>
</tr>
<tr>
<td><code>registeraction.asp</code></td>
<td>A file that creates a new user recordset in profile table in the database.</td>
</tr>
<tr>
<td><code>setprice.asp</code></td>
<td>A file that shows all share prices, which are set by a user. There are options in this file that the user can choose whether the user wants to create, change or delete a share price.</td>
</tr>
<tr>
<td><code>stock.mdb</code></td>
<td>An MS Access database that keeps information of username, password, and share prices set by users.</td>
</tr>
<tr>
<td><code>StockGrabber.class</code></td>
<td>A Java class which is a file that retrieve information from a quote website is needed to place in c:\winnt\java\trustlib.</td>
</tr>
<tr>
<td><code>StockGrabber.java</code></td>
<td>A Java source code file before compilation.</td>
</tr>
<tr>
<td><code>StockGrabberException.class</code></td>
<td>A Java class which is a file that handle error events during trying to retrieve information from a quote website. This file is needed to place in c:\winnt\java\trustlib.</td>
</tr>
<tr>
<td><code>today.asp</code></td>
<td>A file that displays current share prices, which a user concerns. It will show a message 'Max' if the share price reaches a maximum price which the user set and vice versa.</td>
</tr>
<tr>
<td><code>today1.asp</code></td>
<td>A file that is used to check whether a platform, which this application is distributed, is valid.</td>
</tr>
</tbody>
</table>
This flowchart is written in order of its own chart number. The first chart is called chart number 1 and vice versa. A number in outgoing connector shows a letter and a chart number. For example, A1 means the next chart is in chart number 1 at letter A.
Sending query

main.asp

Setting MIME type and variables

Receive and store query

Opening connection to database and creating recordset cursor

Searching matching recordset in the database

Matching record found?

Yes

Checking password

Password correct?

Yes

main.asp#main

Closing all connections

Closing all connections

main.asp#newuser

Closing all connections

main.asp#reject

Closing all connections

main.asp#main

Closing all connections

Chart 2
Setting refreshing time. Refreshing page at regular intervals.

Calling external Java to retrieve data from Internet

Comparing data with data in database

Current price reach max or min price?

Displaying message

No response

Clear cache

End Programme

Chart 3
Chart 4
APPENDIX D: GETSTOCKPRICE FUNCTION

The method to read this flowchart is the same as in Appendix C.
Set price to be 'Single' number

Begin checking price condition

Is a price reach max? No Is a price reach min? No

Yes Yes

Display 'Max' Display 'Min'

End Function

No display

Chart 2