Inventory Control Using Pocket PC

GRADUATE PROJECT REPORT

Submitted to the Faculty of
the Department of Computing and Mathematical Sciences
Texas A&M University-Corpus Christi
Corpus Christi, Texas

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Krishna B. Vommi
Fall 2002

Committee Members

Dr. R. Stephen Dannelly
Committee Chairperson

Dr. Dulal C. Kar
Committee Member

Dr. David Thomas
Committee Member
Inventory Control Using Pocket PC

GRADUATE PROJECT REPORT

Submitted to the Faculty of
the Department of Computing and Mathematical Sciences
Texas A&M University-Corpus Christi
Corpus Christi, Texas

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Krishna B. Vommi
Fall 2002

Committee Members

Dr. R. Stephen Dannelly
Committee Chairperson

Dr. Dulal C. Kar
Committee Member

Dr. David Thomas
Committee Member
# TABLE OF CONTENTS

1. Requirements Formation ........................................................................................................... 07
2. Narrative ................................................................................................................................... 09
   2.1 User Interface ...................................................................................................................... 09
   2.2 Database Connectivity ....................................................................................................... 12
   2.3 Data Entry .......................................................................................................................... 12
   2.4 Data Transmission .............................................................................................................. 13
   2.5 Significant Changes to the Original Design Plan ................................................................. 13
3. System Design and Implementation ......................................................................................... 14
   3.1 Hardware Components ...................................................................................................... 14
   3.2 Software Components ...................................................................................................... 15
   3.3 Implementation ................................................................................................................. 16
4. Results, Conclusion And Future Work ................................................................................... 19
   4.1 Timing Analysis ............................................................................................................... 19
   4.2 Testing of the ICP Project by the MCS Personnel ............................................................... 21
   4.3 Conclusion ....................................................................................................................... 21
   4.4 Future Work ..................................................................................................................... 22
5. Acknowledgements ................................................................................................................ 23

Bibliography and References .................................................................................................... 24
Appendix A ............................................................................................................................... 26
Appendix B ............................................................................................................................... 29
Appendix C ........................................................................................................31
LIST OF FIGURES

2.1 Login.wml ........................................................................................................10
2.2 Main.wml ........................................................................................................10
2.3 Confirm.wml ..................................................................................................11
2.4 Success.wml ..................................................................................................11
2.5 Error.wml ......................................................................................................12
3.1 Pocket PC ......................................................................................................14
3.2 ICP System ....................................................................................................16
3.3 Data Entry Form As It Appears On Pocket PC ..............................................18
4.1 Time Consumed While Collecting Inventory Data On-site .........................20
4.2 Time Consumed For Data Entry Into The Access 2000 Database ..............20
LIST OF TABLES

4.1 Time taken by the two methods for collecting 100 inventory records on site……19
4.2 Time taken by the two methods for entering 100 inventory records into Access 2000
database on the development workstation………………………………………20
ABSTRACT

The Micro Computer Services located in the Corpus Christi Hall building of Texas A&M University-Corpus Christi has an inventory database of the computer workstations and accessories used by the students and the staff. The inventory database is currently being updated by manual entry of the data into the MCS database tables. This method is time consuming, tedious, error prone and consumes scarce manpower.

The Inventory Control using Pocket PC project has been developed to collect inventory data such as barcode and location of the inventory. The resulting inventory records are stored in the Pocket PC memory. The Pocket PC database is then integrated with the Access 2000 database on the Windows 2000 development workstation. The data in the Access 2000 database is then transferred to the MySQL database residing on the Micro Computer Services remote server. The whole process of updating the inventory database through the Inventory Control using Pocket PC project has resulted in a more efficient inventory management system.
1. REQUIREMENTS FORMATION

Forming The Project Concept

This section describes the development of the Inventory Control using Pocket PC (ICP) project concept, the need for a more efficient inventory management and the main objective of the ICP project.

During the first meeting with the Micro Computer Services (MCS) personnel in the month of February 2002, they expressed their desire and need for a more efficient inventory management system. The MCS was managing their vast inventory through manual data entry. The constraints of this traditional method of inventory control being: time consuming, error prone and requiring manpower.

The subsequent meetings with the MCS personnel lead to the development of the Inventory Control using Pocket PC project concept. The initial idea is to realize the completion of the ICP project using hand-held devices and wireless networking. A hand-held device such as a Pocket PC or a hand-held PC would be used to query the MCS remote server and transmit the data via wireless networking. The MCS manages the inventory of nearly five hundred computers distributed over seven student computer labs in the Corpus Christi Hall building, two student computer labs in the Center for Instruction building, and several computer accessories such as printers, scanners, projectors etc. The Micro Computer Services needs to collect the inventory data such as the barcode of the device and the location of the device frequently, in order to keep track of their vast inventory. The MCS personnel is comprised of skilled professionals as well as student workers.
WML, PHP and Java Servlet technology are the software applications that were initially intended to be used for the ICP project. The modifications from this plan are explained in the section 2.
2. NARRATIVE

This section describes the set of software components and the data transmission mechanisms that were originally mentioned in the ICP Project Proposal and the reasons for adapting a new set of software components for implementing the ICP Project. The concept of the ICP project has been successfully implemented without any significant changes to it as is evident from the later part of the document.

In the Project Proposal, WML, PHP and Java Servlets were the proposed software components to be used for the system design of the ICP project. A substantial amount of work was done on the system design using these components. The WML pages designed using WML/WML script would reside on the MCS remote server. The mini Internet Explorer browser of the Pocket PC would be used to access these WML pages. The inventory data collected on the form fields of the WML pages would be transmitted to the remote server via wireless networking.

2.1 User Interface

This subsection describes the user interface on the hand-held device as was mentioned in the Project Proposal. The user or employee identification and password will be checked for authenticity before access to the database is granted. On authentication by the Web server, the user will have access to the database till he/she ends the session by logging out. When the employee using the hand-held device enters the URL: 
wap://hostname/login.wml in the Web browser, the login.wml page (Figure 2.1), is loaded on the hand-held device. When authentic user identification and password is supplied, the Web server sends the main.wml page (Figure 2.2), to the hand-held device.
The different details of the device to be updated in the database are entered in the text fields provided on the main.wml page. The details include the device identification number, device name, year of manufacturing, device model, location of the device, name of the person using the device, name of the person updating the database etc. When the submit button on the main.wml page is pressed after entering the details of the device to be updated in the inventory, the confirm.wml page (Figure 2.3) is loaded on the hand-held device. The user can make any changes to the data he/she entered by pressing the edit button. The data will be transmitted to the Web server hosting the database on pressing the submit button in the confirm.wml page.
If the request to update the database with the data of the device provided is processed successfully by the Web server, the `success.wml` page (Figure 2.4) is sent to the handheld device. In case an error occurred while processing the request, the Web server sends an error message in the `error.wml` page (Figure 2.5).
2.2 Database Connectivity

Prior to using the hand-held devices for updating the MCS inventory database, ODBC (Open Database Connectivity) is achieved on the mailman.tamu.cc.edu Web server. This enables access to the database through the hand-held devices.

2.3 Data Entry

The data in the different fields of the main.wml page is entered by using the barcode scanner and manually with the keyboard available on the hand-held PC. The barcode scanner, placed in one of the slots available on the hand-held PC, is used to scan the device identification number (available on the property tag attached to the device). The scanned number is then retrieved on to the device_id field on the main.wml page (Figure 2.3). The data in the other fields (except the device_id) on the main.wml page is entered manually.
2.4 Data Transmission

On pressing the submit button on the confirm.wml page (Figure 2.4), the data will be transmitted to the Web server hosting the database.

2.5 Significant Changes to the Original Design Plan

To enhance the efficiency of the ICP application, the user interface (see section 3) has been limited to a single screen on the Pocket PC as requested by the MCS personnel. The above-mentioned set of software components involving WML and the data transmission mechanism requires querying of the remote MCS server for every inventory update via wireless networking. The system thus developed would be of little use in locations without wireless networking. Even in buildings that are wirelessly enabled, the coverage is insufficient in certain locations, thus rendering the system inefficient. Keeping these constraints in view, the system has to be developed using the software components that do not rely on wireless networking. In search of the appropriate tools to be used for the ICP system design, the features of Windows CE operating system of the Pocket PC were studied elaborately. Section 3 discusses these features, the changes to the user interface that were requested by the MCS personnel while testing the ICP project, and also the new set of software components that were used for the ICP project implementation. The results of the ICP project, future work and conclusion are explained in the section 4.
3. SYSTEM DESIGN AND IMPLEMENTATION

This section describes the hardware and software components that are used in the system design and implementation of the ICP project.

3.1 Hardware Components

Desktop PC

A Windows 2000 development workstation used to host the software development kit (SDK) for the Pocket PC applications, Access 2000 database and MySQL ODBC drivers.

Pocket PC

Pocket PC 2002 manufactured by Ipaq is a hand-held computer with Microsoft Windows CE 3.0 operating system. The Pocket PC has 64 MB of RAM and is used to collect the inventory data.

Barcode Scanner

A barcode scanner manufactured by Symbol Technologies as an accessory for the Pocket PC is used to scan the barcodes.

Figure 3.1 Pocket PC
Docking Cradle

The docking cradle is an accessory used to connect the Pocket PC with the development workstation.

3.2 Software Components

Microsoft SDK for Pocket PC 2002

The SDK available at the Pocket PC developer Web site of Microsoft was downloaded and installed on the development workstation. This SDK constitutes the applications such as embedded Visual Basic (eVB), which can be used to develop different Pocket PC embedded applications. The forms for data entry were designed using eVB. The SDK also presents the developer with an emulator of the hand-held Pocket PC.

Microsoft ActiveSync 3.5

The Microsoft ActiveSync 3.5 installed on the development workstation is used to synchronize the content of the Pocket PC and the development workstation. The databases, files and other applications can be synchronized between the Pocket PC and the workstation using this application.

Barcode Scanner Drivers

The barcode scanner drivers are downloaded and installed from the Web site of Symbol Technologies. The barcode scanner scans the barcode into any text field that has the cursor placed in it.

MS ACCESS 2000

The database to hold the inventory data of the MCS is designed using Access 2000 on the development workstation.
**MYSQL ODBC DRIVERS**

The MySQL ODBC drivers required to export the Access 2000 database to a MySQL database are installed on the development workstation.

**3.3 IMPLEMENTATION**

This section gives a step-by-step description of the ICP system implementation from coding through data transmission, see figure 2.2. The database has been designed in Access 2000 on the development workstation. This database is converted into a Pocket Access database and is transferred to the Pocket PC through Microsoft ActiveSync application.

![Figure 3.2 ICP System](image)
The application which runs on the Pocket PC, see figure 3.3, was written in embedded Visual Basic (eVB) available with the SDK for Pocket PC. The eVB application is then executed on the Pocket PC. A form with different data entry fields appears when the eVB application is run. The cursor by default blinks in the ‘barcode’ field. When the button present on top left side of the Pocket PC is pressed, the barcode scanner reads the barcode into the field with the cursor placed in it. The remaining two fields ‘Building’ and ‘RoomNum’ are persistent which means they hold the previously entered value until it is changed. The data is stored locally in the Pocket PC when the ‘SUBMIT’ button is clicked. The eVB application uses Active Data Objects for CE (ADOCE) technology for data storage in the Pocket PC.

The Access 2000 database is synchronized with the Pocket PC database when the Pocket PC is placed in the docking cradle through the Microsoft ActiveSync application. Any records that cannot be updated due to conflicts are placed in the conflicts table of the Access 2000 database. The conflicts include duplicate barcode or a null barcode since the barcode has been set the primary key of the database. The Access 2000 database is then exported to the MCS remote server hosting the MySQL database using the ODBC drivers.
Figure 3.3 Data Entry Form As It Appears On The Pocket PC
4. RESULTS, CONCLUSION AND FUTURE WORK

4.1 Timing Analysis

The ICP project has resulted in a more efficient Inventory management system. Inventory management with the ICP implementation is easy to use, less time consuming and less error prone. An inventory record containing barcode and location can be collected in four seconds using the ICP system. The ICP system has also eliminated the use of paper and pen completely and lessened the redundancy of work.

The following results are based on one hundred inventory records that were collected using both the Inventory Control Using Pocket PC and the previously used manual data entry methods. Each inventory record contains data such as the barcode and the location of the device (name of the building and the room number). The two steps required for the collection of the inventory, collecting the inventory data on site (see Table 2.1) and entering the data into the Access 2000 database on the development workstation (see Table 2.2), have been timed.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory Control Using Pocket PC</td>
<td>440</td>
</tr>
<tr>
<td>Manual Data Entry</td>
<td>1430</td>
</tr>
</tbody>
</table>

Table 4.1 Time taken by the two methods for collecting 100 inventory records on site
Figure 4.1 Time consumed while collecting Inventory data on site

<table>
<thead>
<tr>
<th>Method</th>
<th>Time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory Control Using Pocket PC</td>
<td>10</td>
</tr>
<tr>
<td>Manual Data Entry</td>
<td>1820</td>
</tr>
</tbody>
</table>

Table 4.2 Time taken by the two methods for entering 100 inventory records into the Access 2000 Database on the development workstation

Figure 4.2 Time consumed for data entry into Access 2000 database
4.2 Testing of the ICP Project by the MCS Personnel

The MCS personnel have tested the ICP project in the month of July 2002 by collecting the inventory data on-site using the Pocket PC. They collected the inventory data of the computers and accessories from two student labs in the CCH building using the ICP application system. The data was transmitted to the Access 2000 database on the development workstation and subsequently to the MCS mySQL remote database. The help file (see Appendix B) written for the ICP project was used for completing the whole process. The MCS personnel verified the data in the database for its correctness and were satisfied with the results yielded by the ICP project. Minor changes were also made to the eVB application aesthetic.

4.3 Conclusion

The ICP system has resulted in 70% more efficient time consumption system for on-site inventory data collection than the conventional manual data collection (see Table 2.2 and Figure 2.2). The Inventory Control using Pocket PC system relies on Microsoft Activesync application to keep the Access 2000 database on the development workstation and the Pocket PC database synchronized. When the Pocket PC is placed in the docking cradle the Activesync application synchronizes the Pocket PC database with the Access 2000 database. This process takes 10 seconds and would remain more or less the same for even large amounts of data transfer, which is not the same case with manual entry of data. The time consumed is directly proportional to the amount of data to be transferred in the case of the manual entry of data into the Access 2000 database (see Table 2.3 and Figure 2.3). The ICP system is less prone to human errors when compared to the conventional manual data entry method.
4.4 Future Work

The ICP system was designed to collect the inventory data such as barcode and location of the device and the data is exported to the remote MySQL database of the Micro Computer Services. Server side scripts can be written to perform extraction of relevant information, update other related fields of the database etc. This updated database can then be exported to an Access 2000 database residing on the Pocket PC development workstation, which would be used as a reference or a mirror database. Pocket PC application can be expanded to enable it to query the reference database in situations where other details of the device such as name of the user, vendor details etc. are required on-site.
5. ACKNOWLEDGEMENTS

I wish to express my profound gratitude to Dr. R. Stephen Dannelly, who has been a great inspiration for me during my graduate studies, for his invaluable support, encouragement and technical expertise rendered throughout this Project. It is of immense pleasure and privilege to work under his guidance. I express my sincere thanks to Dr. Dulal C. Kar, for his precious time, valuable advice and suggestions. I am grateful to Dr. David Thomas, for his counsel and patient reviewing of my Project Proposal. My special heartfelt thanks to MicroComputer Services in general and Mr. Charles Irby, Mr. Ben Soto and Mr. Michael Williamson in particular for their cooperation, sharing the resources and valuable time. And finally, I would like to thank my beloved parents and sisters for their unending love, affection and support throughout my career.
Books:


Manuals:

Web Sites:

[Devbuzz 2002] An Information Hub for Pocket PC Development. Available from
www.devbuzz.com/content/pocketpc_programming.asp (visited April 08, 2002).


[Symbol 2002] Symbol Technologies. Available from


APPENDIX A – PROGRAM LISTING

Option Explicit

' variable to hold the database location in the Pocket PC memory
Const gDBFileSpec = "\My Documents\mcs_inv.cdb"

' variable declarations for the database operations
' these variables are used by the ActiveX Data Objects to perform various
' database operations
Const adOpenForwardOnly = 0
Const adOpenKeyset = 1
Const adOpenDynamic = 2
Const adOpenStatic = 3
Const adLockReadOnly = 1
Const adLockPessimistic = 2
Const adLockOptimistic = 3

' variable to open connection to the database
Public conn As ADOCE.Connection

' this function opens a connection to the database location. If there are any errors
' in connecting to the database, an error message is displayed.
Function connOpen() As Boolean
On Error Resume Next
 connOpen = True

' create an ADOCE object and use the same object whenever a database operation
' needs to be performed
Set conn = CreateObject("ADOCE.Connection.3.0")

' open the connection to the database with the object
conn.Open gDBFileSpec

' if errors in opening the connection to the database, display a message and close the
' application
If conn.Errors.Count > 0 Then
 MsgBox "errors in connOpen", vbOKOnly
 'connClose
 connOpen = False
End If

On Error GoTo 0
End Function

' this subroutine is required to close the connection to the database
Sub connClose()
'close the database connection
On Error Resume Next

'close the connection to the database object
conn.Close
' set the ActiveX Data Object to nothing
Set conn = Nothing
On Error GoTo 0
End Sub

' this sub routine is used to display various errors that may occur
' while processing different user requests
Sub DispErrors()

' variable declaration
Dim dispErr As String
Dim arb1 As Integer
Dim arb2 As Integer
Dim ADOErr As ADOCE.Error

' show connections errors to the ADOCE database
' the for loop is used to display the errors and their source
For arb1 = 0 To conn.Errors.Count - 1
    Set ADOErr = conn.Errors(arb1)
    dispErr = "desc = " & ADOErr.Description & vbCrLf
    dispErr = dispErr & "number = " & Hex(ADOErr.Number) & vbCrLf
    dispErr = dispErr & "nativeerror = " & ADOErr.NativeError & vbCrLf
    dispErr = dispErr & "source = " & ADOErr.Source
    MsgBox dispErr, vbCritical, strTitleBar
Next arb1
End Sub

' this sub routine is used to clear all the text fields on the
' eVB form on the Pocket PC and place the cursor in the Barcode field
Private Sub ClearCmd_Click()
    'clear all the fields
    BarCode.Text = ""
    Building.Text = ""
    RoomNum.Text = ""
    BarCode.SetFocus
End Sub

' this sub routine is used to update the database on the Pocket PC
' the sub routine checks for the connection to the database, opens the database
' table, captures the data entered in the eVB form fields and updates the database.
' after updating an inventory record, clear the barcode field and place the cursor in it.
Private Sub SubmitCmd_Click()

Dim rs As ADOCE.Recordset
'update the database
If connOpen = True Then
    Set rs = CreateObject("ADOCE.Recordset.3.0")
    On Error Resume Next
    rs.Open "MainTbl", conn, adOpenKeyset, adLockOptimistic
    rs.AddNew
    rs.Fields("BarCode") = CStr(BarCode.Text)
    rs.Fields("Building") = CStr(Building.Text)
    rs.Fields("RoomNum") = CStr(RoomNum.Text)
End Sub
rs.Update

If conn.Errors.Count > 0 Then
    DispErrors
    MsgBox "Error in updating database", vbOKOnly
End If

BarCode.Text = ""
BarCode.SetFocus
'place the cursor in the barcode field
On Error GoTo 0
rs.Close
connClose
End If

End Sub

' the eVB application on the Pocket PC is terminated by clicking the
' OK button provided on the title bar of the application
Private Sub Form_OKClick()
'end the application
   App.End

End Sub
Windows 2000 Development Workstation

1) Log on to the development workstation as 'Administrator'. Make sure the Pocket PC is docked properly on the cradle and is connected to the workstation. The Microsoft Activesync window opens and displays the applications that are synchronized between the workstation and the Pocket PC. The window would look similar to the following screenshot. The important application to look for in this case is 'Pocket Access' with the status being 'Synchronized'.

![Microsoft Activesync Window](image)

**Pocket PC**

1) To start collecting the inventory, tap: Start > mcs_inv. A form titled MCS_Form opens up with three input fields.

2) By default the cursor would be blinking in the Barcode field, if not, tap the field named Barcode with the Stylus before scanning the Barcode. The scanner reads the Barcode into any field with the cursor in it.

3) Enter the Building and Room Number using the soft keyboard at the bottom right corner of the Pocket PC screen. These two fields are persistent, which means it needs to be changed only if there is a change in their values. Tap the 'Submit' button to update the database on the Pocket PC.

4) After collecting the inventory, place the Pocket PC in the docking cradle and wait for the Microsoft Activesync window to display the application status on the
development workstation. Check the conflicts tables present below the MainTbl in the mcs_inv database for any conflicts due to which some records may not be updated in the database.

Windows 2000 Development Workstation

Transferring the Access 2000 database to the MySQL server:

1) Make sure correct privileges have been set in the MySQL database to read/write the tables from the development workstation.

2) Open the 'mcs_inv' Access 2000 database and select the table (MainTbl) that has to be exported to the MySQL database. Click File > Export, select the database name (mcs_inv) and then select ODBC databases in the pop down menu of Save as type.

3) Type in any Table Name and click OK. On the next screen 'Select Data Source', click on the 'Machine Data Source' tab. Select the Access-MySQL under data source name with type as System and click the OK button.

4) Enter the required information on this screen and click OK. Check for the table in the MySQL database.
APPENDIX - C

Access 2000 Database structure

The Access 2000 database designed on the Windows 2000 development workstation has three tables. The “MainTbl” is used to hold the inventory data collected by the Pocket PC, while the “MsysCeCMainTbl” and “MsysCeConflicts” are used to hold the conflicts and the errors that might occur while transferring the database from the Pocket PC database.

The following is a screen shot of the Access 2000 database showing the structure of the different tables used.
### Names of the Files and Locations

<table>
<thead>
<tr>
<th>Name</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access 2000 database</td>
<td>C:\Documents\Administrator\MyDocuments (mcs_inv.mdb)</td>
</tr>
<tr>
<td>Pocket PC SDK</td>
<td>C:\Windows\Start\Menu\mcs_inv.vb (mcs_inv.vb)</td>
</tr>
<tr>
<td>Pocket PC Application</td>
<td>C:\Windows\Start\mcs_inv.evbc (mcs_inv.evbc)</td>
</tr>
</tbody>
</table>