PC Cash – a Cash Settlement System

By

Jeffrey S. Evers

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

May 2002

PC Cash – a Cash Settlement System

By

Jeffrey S. Evers

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

© Copyright 2002 Jeffrey S. Evers

The author grants to the Information Engineering Technology Program permission to reproduce and distribute copies of this document in whole or in part.

Jeffrey S. Evers	Date
Robert Schlemmer, Faculty Project Advisor	Date
Lawrence G. Gilligan, Department Head	Date

Acknowledgements

As is always the case, there are people to thank.

Professor Bob Schlemmer, for coordinating the IET program and also his advice with this project;

Dr. Sam Geonetta, for his valuable insight and also his dedication to the IET program;

Gar Seigla, for his friendship;

And quite a few others who made this degree more than just a piece of paper.

Dedication

This project is dedicated to my wife Barb and my daughter Kelly. Obtaining this degree would not have been possible without their constant support and encouragement during the past four years. Thank you.

Table of Contents

Abstract

PC Cash was designed for Paramount's Kings Island amusement park. The cash control department at Kings Island is responsible for counting and verifying the sales amounts from each cash register in the park. Prior to *PC Cash*, sales information was hand-written on a report sheet. A report sheet was completed for every cash register that was in use for the day. The information from the report sheets were then manually entered into a mainframe computer. The inefficiency of manually calculating and recording sales information prompted the development of *PC Cash*.

This application provides an efficient method to reconcile items from multiple cash registers. *PC Cash* allows for fast, accurate entry of all types of revenue including currency, coin, checks, coupons, gift certificates and much more. Built-in reports are created with a few clicks of the mouse, including check listings, variance reports, and cash transfer reports. Other sources of data can be imported from point of sale systems or exported to mainframe computers.

PC Cash was developed using Microsoft technologies, including Visual Basic 6.0, SQL Server 2000, Active X Data Objects (ADO), and HTML Help Workshop.

PC Cash – a Cash Settlement System

1. Description and Intended Use

PC Cash is a cash settlement application that helps users count and verify money from multiple cash registers. Cash settlement applications are used to consolidate several steps used in verifying daily sales revenue. Several items from the cash registers need to be counted and verified. Some of these items include currency, coin, checks, bankcards, and other miscellaneous items.

PC Cash is an easy-to-use system that simplifies and speeds up cash office functions to help reduce labor costs, ensures counting accuracy, and verifies register totals.

1.1 Statement of the Problem

The cash control department at Kings Island is responsible for counting and verifying the sales amounts from each cash register in the park. A report is completed for every cash register that is in use for the day. Items counted, verified and entered into the report include: currency, rolled coin, loose coin, checks, bankcard receipts, register pickups and other miscellaneous items. Currency is also categorized by denomination to maintain an accurate audit trail for other accounting purposes.

Register pickups take place throughout the day. Surplus amounts of money removed from the cash register are returned to the money room. An Access database verifies and tracks each pickup. A printed report shows a total of each register's pickup amount.

Several tellers count and verify the cash items from each register. All data is handwritten on a report sheet. Totals for each report sheet are computed with a

calculator. After completing the cash verification of all assigned registers, the teller sub-totals all the register reports, using a calculator. All the information from each register report is then entered into the park's mainframe computer. Sales reports are generated from this data.

Various problems result from these procedures. One problem is the time needed to process the daily cash register reports. Sales information is recorded on the register sheet, then entered into a calculator to obtain totals, then entered into a computer. Entering redundant information increases the processing time. The possibility of inconsistencies between the handwritten information and the information entered into the calculators and mainframe computer also exists. This entire process greatly increases the effort of auditing the daily sales.

1.2 Overview of Cash Settlement

The purpose of a cash settlement system (CSS) is to count and verify a reported amount of money. Cash settlement systems are used in retail stores and banks. The concept of a computer based CSS is not new. DeLaRue (formally known as Brandt Inc.) developed a CSS that used a dedicated piece of hardware. The original CSS had a keyboard, display, printer, and a CPU in one unit. These units were elaborate adding machines that accumulated totals. The cost for a hardware-based CSS was approximately $4,500.

DeLaRue abandoned hardware-based cash settlement systems. According to Greg Griffin of DeLaRue, "consumers are demanding pc-based cash settlement systems. Banks can purchase personal computers for less than half the price of hardware-based cash settlement system". DeLaRue could no longer justify the price of a hardware-based

cash settlement system. DeLaRue developed a pc-based cash settlement system. The $1,200 software application is installed on a computer of the customer's choice. The application only operates in a DOS environment. A Windows-based system is not available.

DeLaRue and other software developers have concentrated their efforts to design cash settlement systems for the banking sector. Financial institutions use multiple cash settlement systems in one location and purchase multiple copies of the CSS software. Developers have designed software applications to collect data from their cash settlement systems. The data is stored in a proprietary format. Customers may purchase report templates designed solely for the banking sector.

A typical retail store needs one CSS. Because of low demand, software developers have not developed any innovative PC-based cash settlement systems for retail operations. As a result, PC-based CSS for retail works the same as the original hardware-based CSS. Totals from each cash register are added to obtain a grand total for the day's sales. The only permanent record of the data is a printed report. The information from each register is stored in a temporary data file on the PC's hard disk. The data files are erased whenever a new day's information is processed.

Retail cash settlement systems have various drawbacks. No long-term data storage is available for the retail CSS. This makes it difficult to compile sales information for any given period greater than one day. Another drawback is that a software application is not available to collect data from multiple retail cash settlement systems. When multiple systems are used, the totals from each system are manually

added. Mr. Griffin confirmed that there is no current plan to upgrade cash settlement software for retail operations.

1.3 Solution

The proposed solution to make Kings Island's cash office procedures more efficient is to develop a PC-based cash settlement system, *PC Cash*. This application consolidates several steps used to reconcile cash register totals. The core of this application is a relational database. The database can permanently retain all of the sales data. The archived data can be recalled for viewing and printing various reports. *PC Cash* can operate in a single-user environment as well as in a multi-user network environment.

There are two main elements to this application, the database and the user interface. The user interface is the link to the database. Cash office staff enters all daily sales information into the database through the user interface. The database calculates the totals from the sales information and stores a permanent record. Other software applications can access the database and retrieve these records.

2. User Profile

There are two categories of application users. The two levels include tellers and administrators.

2.1 Tellers

These employees are the primary users of *PC Cash*. Tellers are the people who are responsible for counting and verifying register pick-ups and also the revenue items

from the cash registers. The tellers are responsible for entering all the sales data into the database.

The tellers are seasonal employees. Approximately 75 percent of the tellers return every season to work in the cash office. These employees include college students and school teachers. All of the tellers are experienced computer users. The tellers are expected to know how to use Microsoft Excel. Tellers also use two other programs that are Access Database applications.

2.2 Administrators

Administrators are the second level users of *PC Cash*. Administrators are responsible for maintaining the basic information in the *PC Cash* database. These items include teller names and identification numbers. Other items include cash register descriptions and other information associated with the cash registers.

Administrators consist of seasonal and yearly employees. All of the administrators can be considered advanced computer users. The administrators have been trained in advanced uses of Microsoft Excel. They are also capable of editing Microsoft Access databases.

3. Design Protocols

PC Cash includes three areas of focus from the Information Engineering Technology program. The first area of focus is database development. A SQL Server database is the core of the project. The database provides long term data storage for reports required by the end user. The second area of focus is programming. A Visual

Basic application is used to interface with the database application. The third area of focus includes graphic design elements.

3.1 SQL Server Database

The database for *PC Cash* is a Microsoft SQL Server 2000 database. Microsoft SQL Server 2000 is a relational database system that is scaleable from small departmental networks to enterprise-wide networks. The database was created so that a high level of sophistication was demonstrated. This includes many-to-many relationships, data field constraints, and the use of stored procedures.

An ActiveX component that is a dominant part of *PC Cash* is ADO code. ADO (ActiveX Data Objects) is a Microsoft technology that allows a high-level interface to a variety of data sources. ADO provides consistent, high-performance access to data. *PC Cash* makes use of ADO code to transmit and receive information between the database and the user interface.

3.1.2 Database Layout

See Appendix A.

3.2 Programming

The user interface was developed with Microsoft Visual Basic 6.0. The benefits of using visual basic include the use of integrated database tools and a data environment to create reports. Visual Basic allows the flexibility to expand the project as needed. The user interface contains multiple forms to enter the required data. The interface incorporates elements of graphic design to assist in the navigation of multiple data forms.

3.2.1 Navigation Tree

See Appendix B.

3.2.2 Interface Design and Navigation

Users starting the application are prompted to sign in at a login form. This form is a simple verification of the user and the date. This basic information is required to maintain the integrity of the data, and also for auditing purposes. After a successful login, users see the main application window. It is navigated with standard windows pull-down menus and a toolbar. Users can choose items from a pull-down menu or the toolbar and click on the item they are interested in using.

3.2.3 Graphics and Icons

PC Cash includes the Paramount's Kings Island logo on the login form. Each module has different graphic elements to help distinguish it from other modules. This allows users to build a visual association with where they are in the application.

3.2.4 Color Scheme

Every module in *PC Cash* has a distinct color scheme. For example, the Register Pick-Up module uses shades of blue and the Register Balance module uses shades of green. The control layout in each module is significantly different. The user should be able to recognize where they are in the application. By way of color schemes, graphics, and the control layout, the user should always know where they are in the application. The text in the application uses Tahoma font; this is a true-type font. Ten-point size type is used for the majority of the application.

3.2.5 Help System

PC Cash contains a basic help system. Users are able to access information that describes each modules purpose and functionality. Help is accessed from the main application menu.

4.0 Project Timeframe

The timeline outlines the major milestones to develop *PC Cash*. The development of *PC Cash* has three periods. The first period took place during Senior Design I. This period was used to research the feasibility of developing application. The majority of the application was developed during the second period, Senior Design II. Testing and documenting the application took place during the third period, Senior Design III.

4.1 Timeline

See Appendix C.

5.0 Project Resources

The development of *PC Cash* required a PC to build the application. For convenience and mobility, a Dell laptop computer was used. The computer includes the Windows XP operating system and will support the software required to develop the application.

5.1 Development Cost

The software necessary to develop the application includes Microsoft Visual Basic 6.0 Professional Edition and Microsoft SQL Server 2000 Developer Edition. Cost for all the project's elements can be found in Appendix D.

5.2 System Requirements for PC Cash Application

The only additional program resource that is required for *PC Cash* is SQL Server. Paramount's Kings Island is currently using SQL Server 2000. They also have ten PCs that exceed the minimum requirements for the user interface. Kings Island did not have to disburse any additional funds to implement *PC Cash*. For minimum hardware and software requirements see Appendix E.

6.0 Project Objectives

The goal of *PC Cash* is to deliver a more efficient system of reconciling cash register revenues. Currently the cash office staff transcribes information from several reports, and manually combines information from other sources to provide totals for entry into the mainframe computer. The use of *PC Cash* allows cash office staff to eliminate the multiple steps they currently take to reconcile the cash registers.

This application consolidates several steps used to reconcile cash register totals. There are two main elements to this application, the database and the user interface. The user interface is the link to the database. Cash office staff enters all daily sales information into the database through the user interface. The database calculates the totals from the sales information and stores a permanent record. Other software applications can access the database and retrieve these records.

6.1 User Interface

The user interface was designed to operate on Microsoft Windows NT-based operating systems. NT-based systems now include Windows NT 4.0, Windows 2000,

and Windows XP. The user interface provides cash office staff with the following functions:

- Pickup Module
    - Enter cash register pickup information.
    - Edit pick-up information.
- Reconcile Module
    - Enter currency and coin amounts.
    - Enter checks, bankcard, and other revenue items.
- Report module
    - Provide register pick-up variance report.
    - Provide total cash by teller report.
    - Provide check listing for sales date report.
- Provide database administration to authorized users.
    - Enter & edit teller information.
    - Enter & edit cash register information.
- Help System
    - Provide instructions for the user interface modules.

6.1.2 Database

The second element of the *PC Cash* application is a SQL Server database. The database has several features:

- Provides long-term storage of sales data.
- Calculates totals of the sales information.
- Import and export data from other data sources.

The Kings Island IS department will be developing DTS (Data Transformation Services) packages to do the importing and exporting of the other data sources.

7. Proof of design

The goal of delivering a more efficient system to count and verify cash register revenues was achieved. The cash office staff was able to consolidate several steps in reconciling cash register totals. The user interface consisted of multiple modules to enter the data. Screenshots of the various modules are available in Appendix F.

Prior to *PC Cash* the register pickup information was entered into an Access database. Reports were generated from this database. The tellers would search the report for information, and then record that information on a balance sheet. The register pick-up procedure was incorporated into *PC Cash* application as one of the modules. This module eliminated the time consuming step of re-entering pickup data.

The register entry module provided a dramatic reduction in labor. This module is the point where all of the cash register revenues are entered. The manual method of counting and verifying currency included the use of a currency counting machine. Currency counters provide the number of bills counted of one specific denomination. The count was recorded on a balance sheet. The value of the currency was then calculated with an adding machine. The register entry module permits the operator to enter the piece count of the bills; the value is automatically calculated by the application. Other items that are entered into the register entry module include coin values, checks, bankcards, and other revenue items. Sales data is posted to the database after all of the items are entered. The data can be retrieved for reports at a later time. Stored data from the database is used to generate daily reports. The report module provides three reports.

These reports are used for auditing purposes in the cash office. The first report lists the variances from the register pick-up module. The second report is the total cash counted by an individual teller. The database calculates all of the teller's total cash for the day. This report is included with the cash that the teller transfers to the vault. The third report lists all of the checks that were processed for the day. The check listing is sent to the bank along with the checks.

These different modules accomplished the goal of the application. Consolidating several of the reconciliation methods produced a savings in labor and also increased accuracy of counting.

8. Conclusions

The manual methods that Kings Island used to reconcile cash register revenues created an opportunity that could dramatically increase efficiency. An IT solution seemed to be the obvious solution to help improve their settlement methods.

The most important phase of the development cycle was to understand all of the methods used to balance the cash register revenues. I spent several hours with the Kings Island staff observing and actually counting money from the cash registers. Learning every step of the process provided invaluable insight to creating an IT solution. After researching different technology possibilities, a client-server method was employed. The client side used Visual Basic to develop the user interface. Developing a web application was ruled out due to security issues and the amount of data that needed to be entered. The server side used a SQL Server database. In most applications of this type, SQL Server is unnecessary. A simple Access database could be used. In Kings Island's situation, multiple users access the database at one time. SQL Server was already in

place at Kings Island and the cash office manager requested a system built on SQL Server 2000. SQL Server also permits the IS staff to import and export data to and from the database.

Midway through this project, personnel changes were made in the finance department at Kings Island. Accounting methods were restructured and *PC Cash* did not fit in with the new accounting structure of the cash office. Despite this event, I continued the project as planned and learned a great number of things from this experience.

I was able to expand my knowledge of both SQL Server and Visual Basic. I developed several stored procedures. The stored procedures enabled the user interface to interact with the database in a more effective method. I was also able to expand the functionality of the text boxes in Visual Basic. I developed object classes that were used to validate data that was entered into the textboxes. Implementing these classes eliminated 120 lines of code from this project. Code samples can be seen in Appendix G.

One problem I had to overcome was the Help Module. I wanted this module to perform like help modules in other Windows applications. I found a help authoring tool, *HTML Help Workshop*, through the Microsoft web page. It took some time for me to learn the program operation. The help module worked as a stand alone program, but I could not get the help module to function with *PC Cash*. After additional research, I was able to connect the help module with *PC Cash*.

9. Recommendations

This capstone project was a beneficial learning experience. This project provided a means to join the technologies studied in the IET program. Due to time constraints, *PC Cash* has limited functionality. A variety of items could supplement upcoming versions.

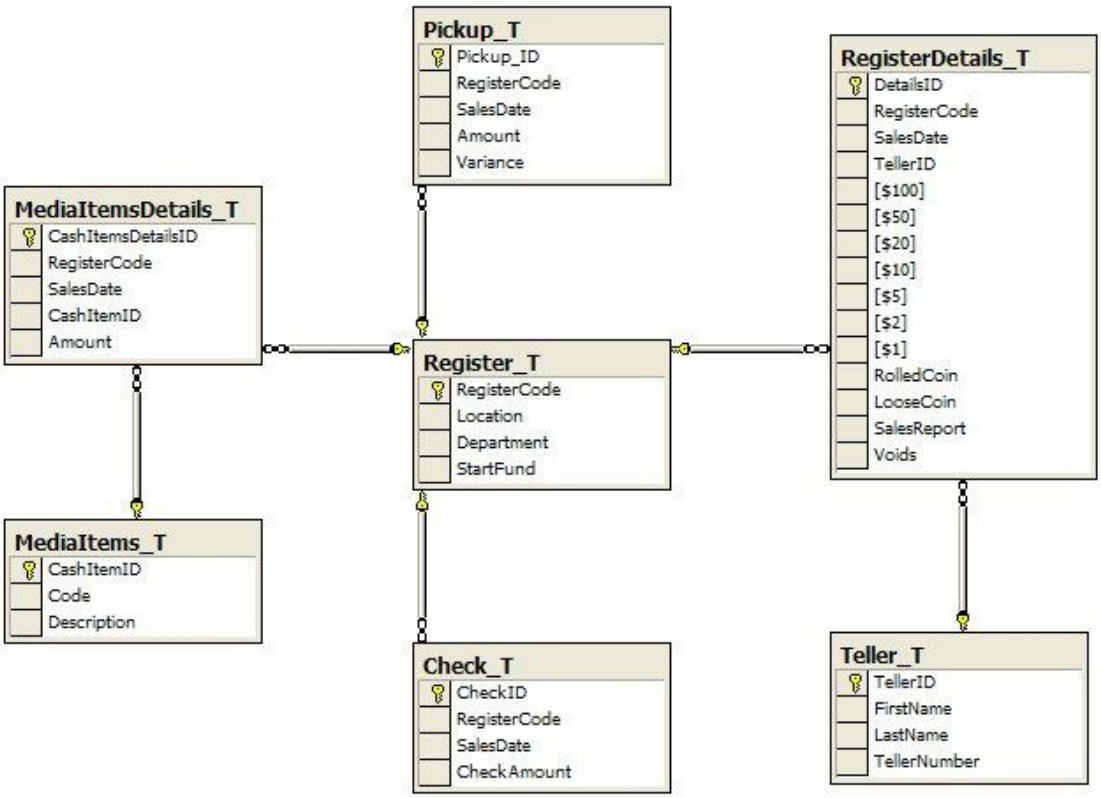
The goal of PC Cash is to increase efficiency in counting money. Interfacing the currency counter with the application is an additional method to increase efficiency. Currency counters are capable of being interfaced with computers. An interface connection would allow the currency counter to transmit the count directly to the application, thus automating the process.

Another item that could be included in the application is a pop-up calculator. The pop-up calculator would let the user perform computations and paste the information into the application. Adding machines are common place in cash offices, but they take up valuable workspace when used beside a cash settlement system. Incorporating a pop-up calculator in the application would eliminate the adding machine.

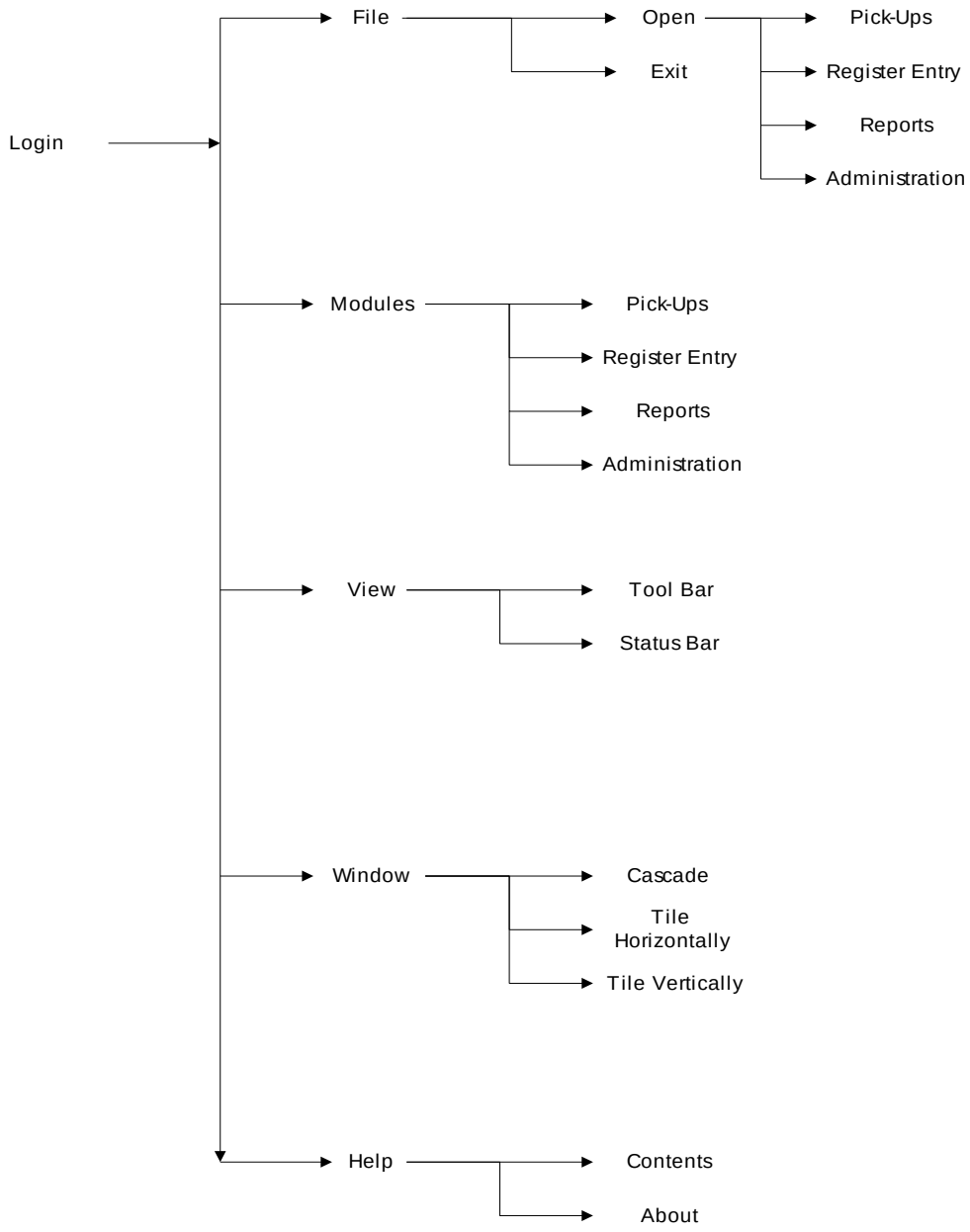
One of the major weaknesses of the current version of *PC Cash* is the report module. The report designer included with Visual Basic 6.0 has limitations. Future versions of *PC Cash* would include Crystal Reports, a more robust report writer. Crystal Reports is a world standard for high-performance reporting. It is used to create interactive content from almost any data source. Reports are easily integrated with applications. Additionally, they can be published to the Web in a variety of formats.

In conclusion, the current version of PC Cash requires knowledge of SQL Server 2000 to maintain the database. The database maintenance should be automated and included into the user interface. These functions would include methods for archiving and backing up data. Again, due to time constraints this functionality was not included in the current version.

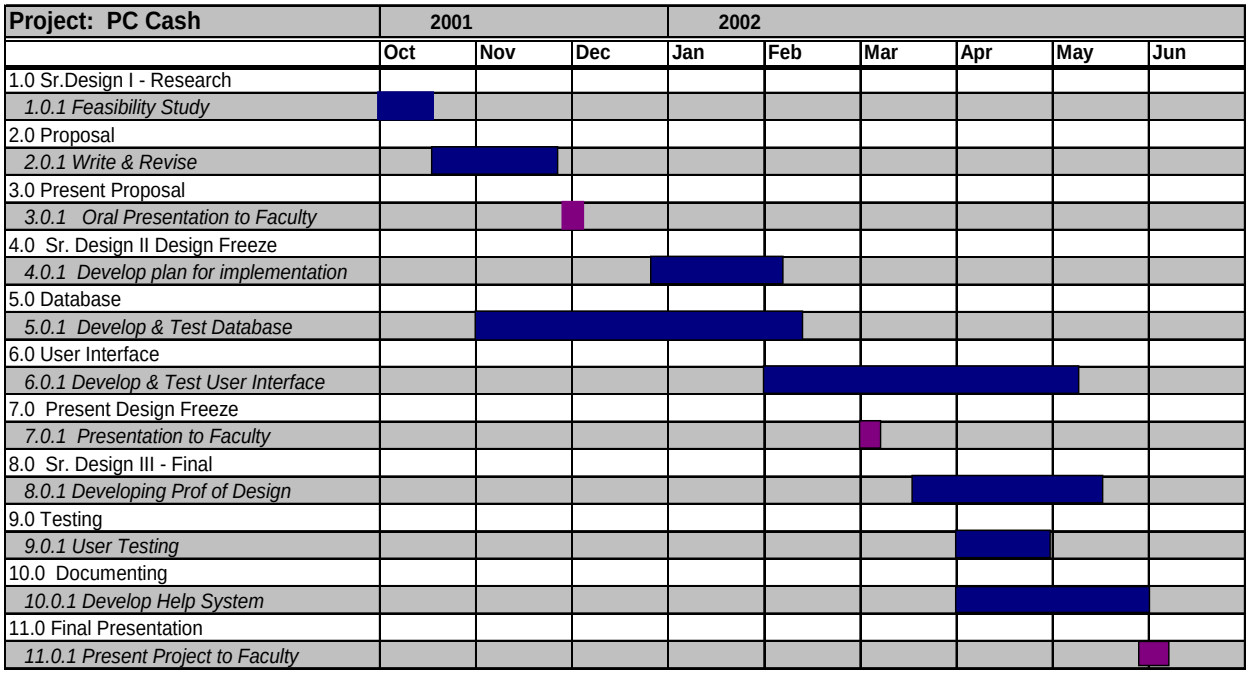
Appendix A.
Entity Relationship Diagram for *PC Cash* Database



Appendix B.
Decision Tree / Program Flow for PC-Cash Interface



Appendix C.
Project Timeline



Appendix D.
Development Cost

Item	Edition	*Price*
Dell Laptop PC	**Inspiron 8100**	$ 2,100.00
Visual Basic 6.0	**Professional Edition**	549.00
SQL Server 2000	**Developer Edition**	499.00
HTML Help Workshop	**4.74**	Freeware
Total Development Cost		$ 3,148.00

Appendix E.
System Requirements

The minimum hardware and software required for running the user interface:

- Microsoft Windows NT 3.51, 4.0, Windows 2000, or XP.
- 80486 or higher microprocessor.
- VGA or higher-resolution screen supported by Microsoft Windows.
- 32MB of Ram, 64MB recommended.

The minimum hardware and software required for SQL Server 2000:

- Microsoft Windows NT Server 4.0 or Microsoft Windows 2000 Server. Windows 2000 Server is required for some SQL Server 2000 features.
- Pentium 166 MHz or higher microprocessor.
- 64MB of RAM, 128MB or more recommended.
- 95MB to 270MB of hard disk space, 250MB typical.
- VGA or higher-resolution monitor, 800x600 or higher resolution required for the SQL Server graphical tools.

Appendix F.
Screenshots from *PC Cash*

1. Login screen

2. Menus screen

3. Pick-up entry screen

4. Register entry screen; currency & coin

5. Register entry screen; media items

6. Help module

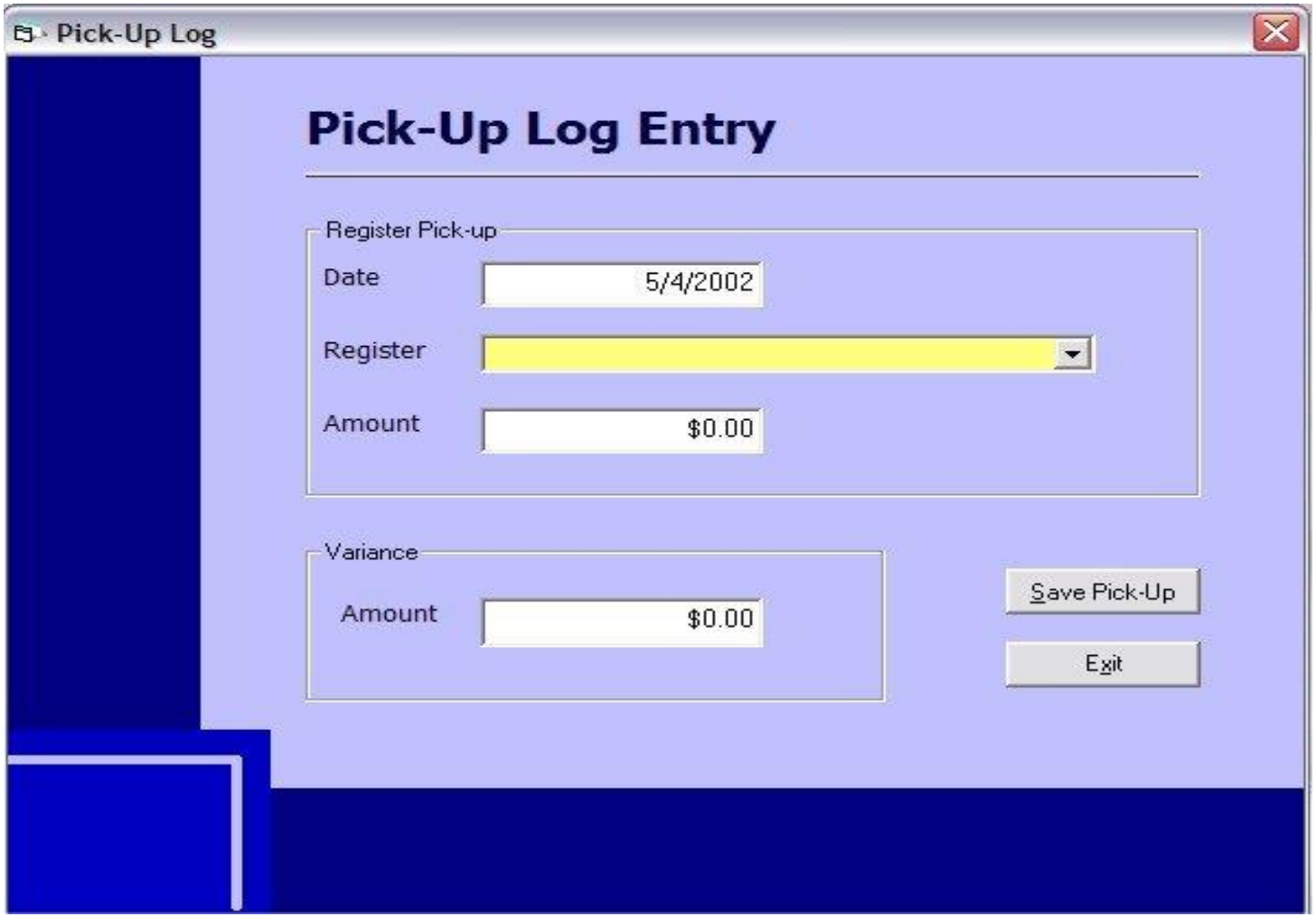
7. Variance report

8. Check Listing report

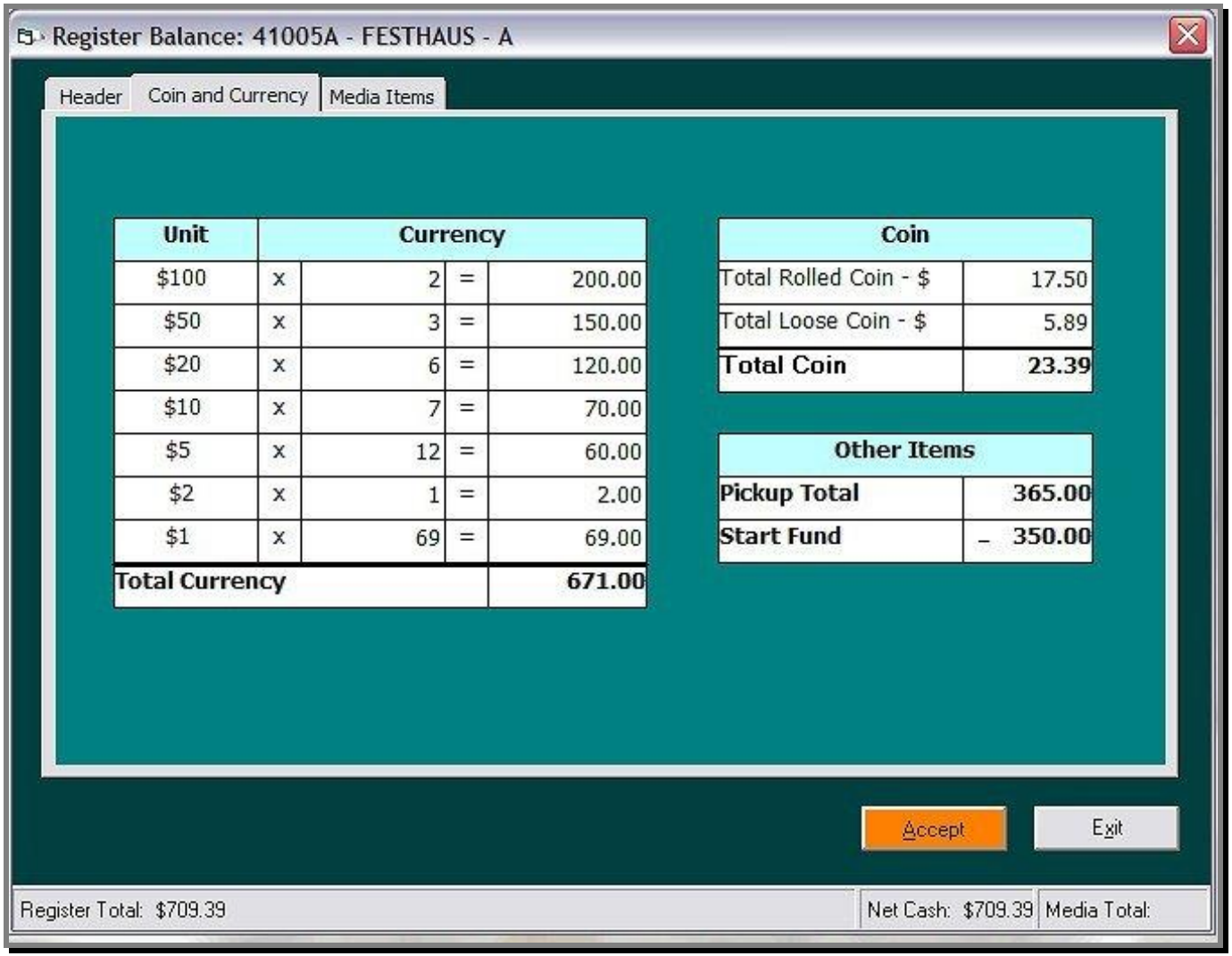
9. Teller's Transfer report

Screen Shot 1. Login screen

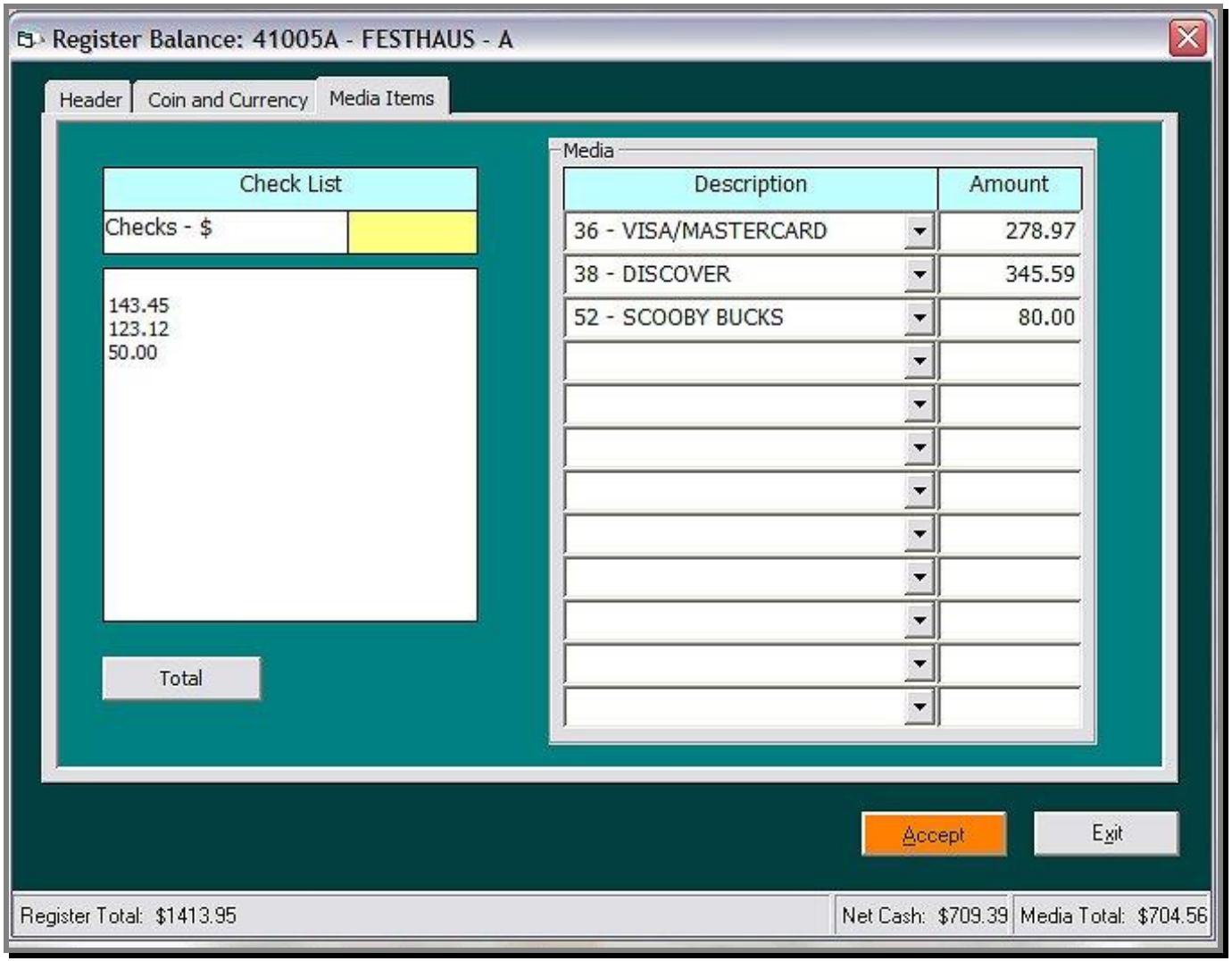
Screen Shot 2. Menu screen



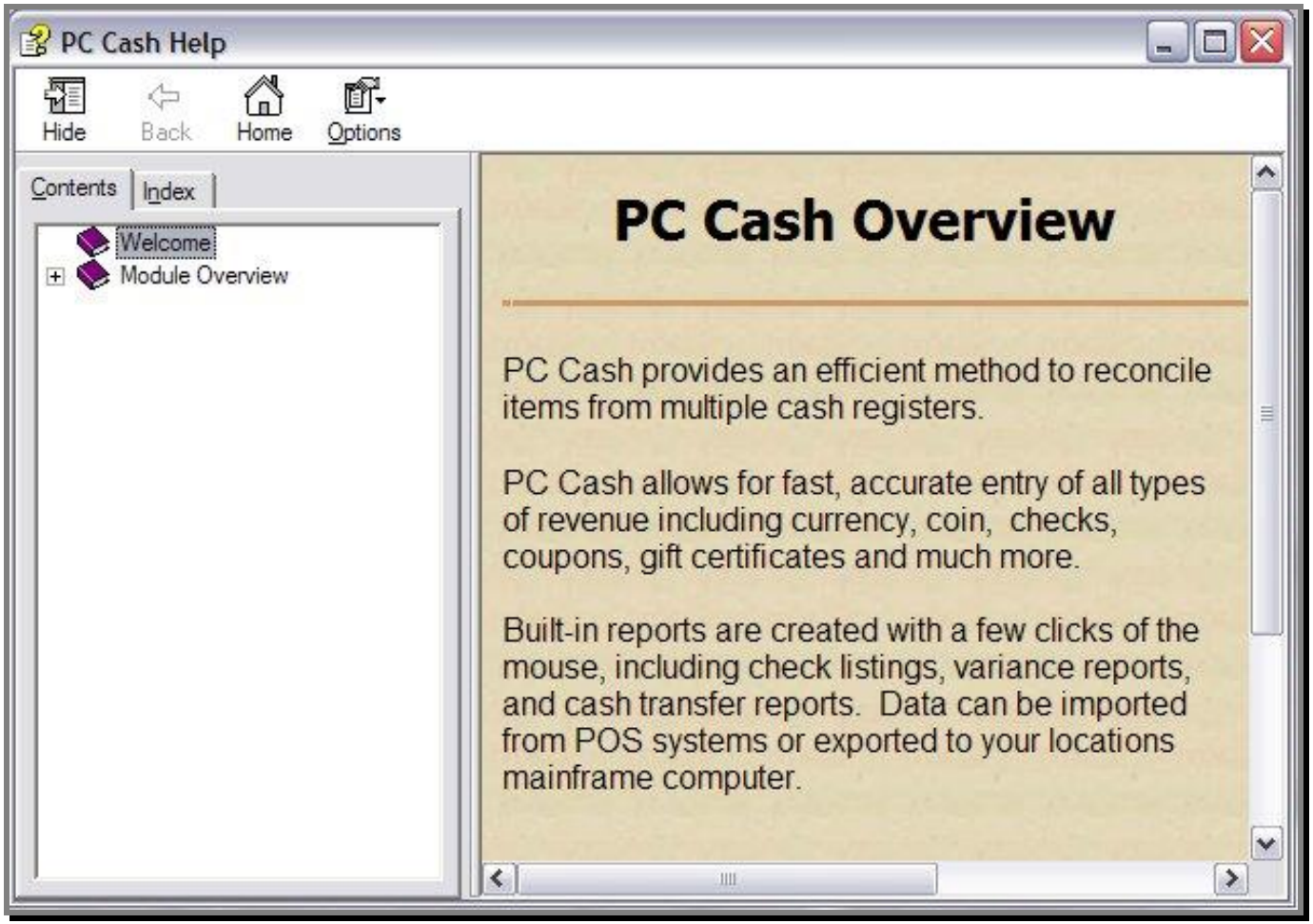
Screen Shot 3. Pick-up entry screen



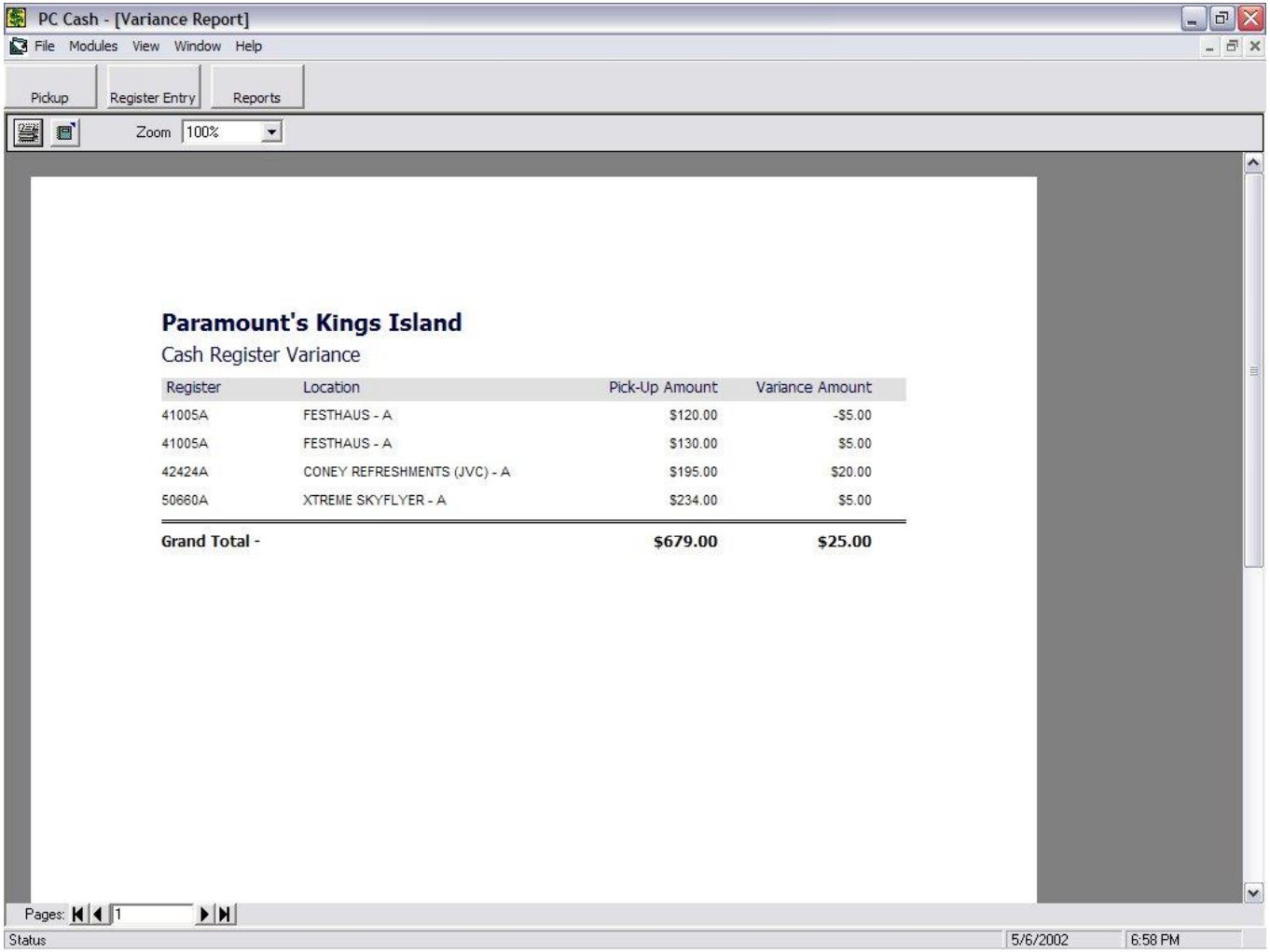
Screen Shot 4. Register entry screen; currency and coin



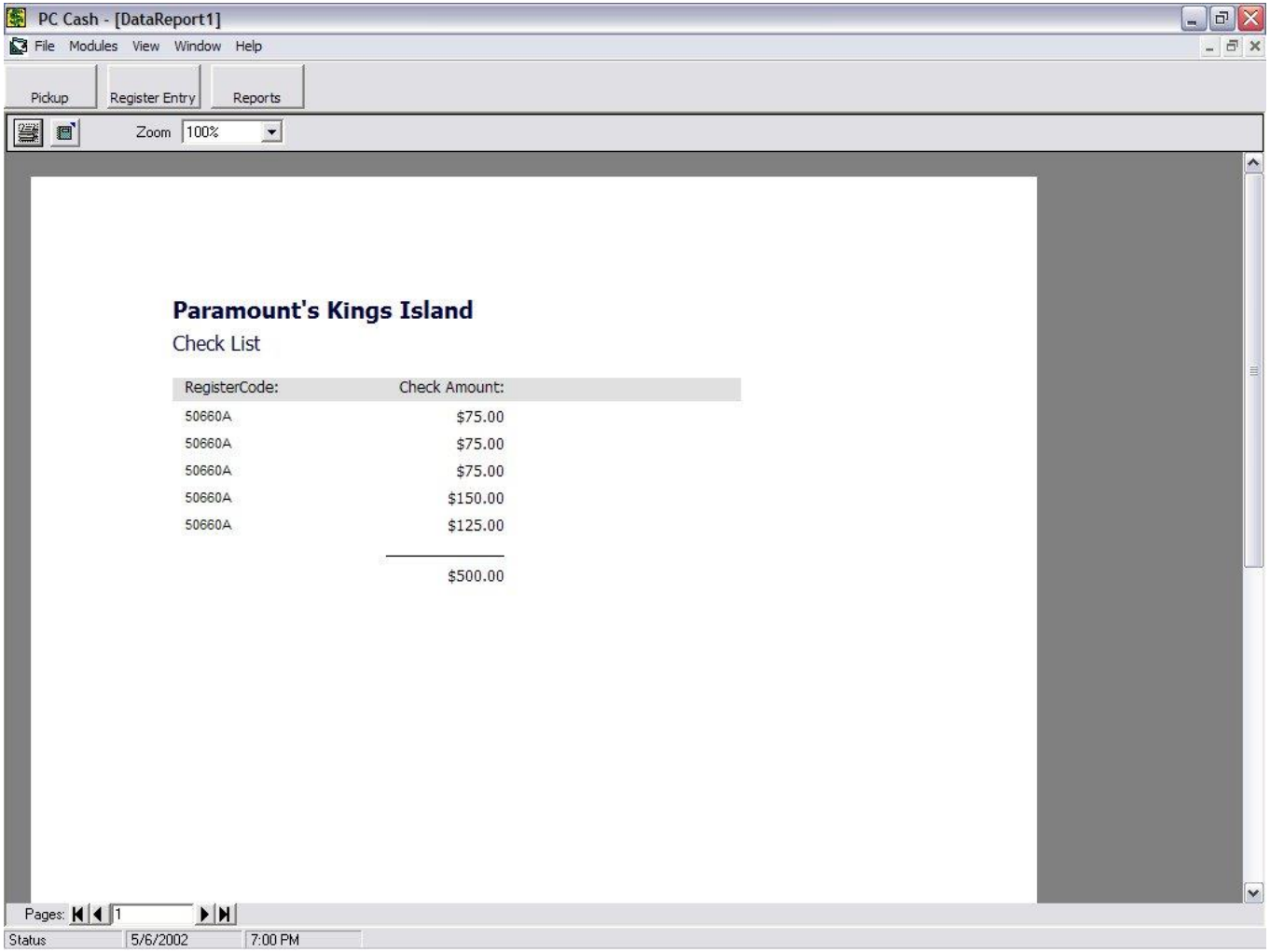
Screen Shot 5. Register entry screen; media items



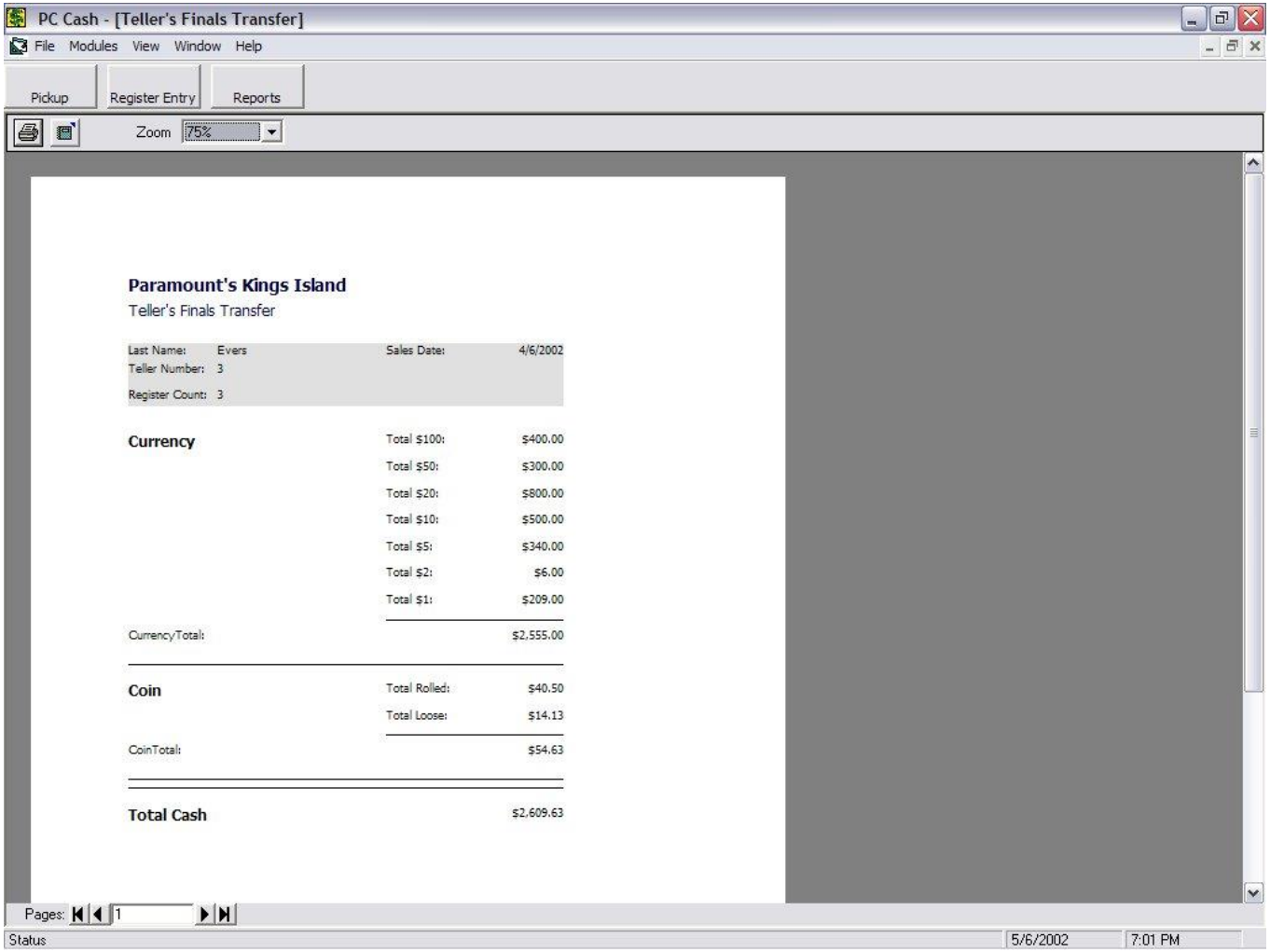
Screen Shot 6. Help module



Screen Shot 7. Variance report



Screen Shot 8. Check Listing report



Screen Shot 9. Teller's Transfer report

Appendix G.
Code Samples

Visual Basic: textbox class object.

```
'This class is an extension of the text box
'Purpose: ONLY INTEGER values can to be entered in a textbox
'This also highlights the textbox that has current focus
Option Explicit
Public WithEvents TextBox As TextBox
```

```
Private Sub TextBox_KeyPress(KeyAscii As Integer)
Select Case KeyAscii
   Case 13
      KeyAscii = 0
      SendKeys "{TAB}" 'Check for enter key
   Case 0 To 31       'Accept control characters
   Case 48 To 57      'Accept digits
   Case Else
      KeyAscii = 0   'Reject everything else
   End Select
End Sub
```

```
Private Sub TextBox_GotFocus()
TextBox.BackColor = YELLOW
TextBox.SelStart = 0
TextBox.SelLength = Len(TextBox.Text)
End Sub
```

```
Private Sub TextBox_LostFocus()
TextBox.BackColor = vbWhite
End Sub
```

Visual Basic code using ADO to access a SQL stored procedure.

```
Private Sub cboRegisterCode_Click()
strRegister = cboRegisterCode.Text
strRegister = Left(strRegister, 6)
'Set the command object properties
Set LookupCMD = New ADODB.Command
 LookupCMD.CommandText = "up_parmsel_PickupLookup"
 LookupCMD.CommandType = adCmdStoredProc
 LookupCMD.ActiveConnection = cnnConnection
'Create and append the parameters to the parameters collection
Set LookupPARAM1 = LookupCMD.CreateParameter("Register", adVarChar, adParamInput, _
   7, strRegister)
LookupCMD.Parameters.Append LookupPARAM1
Set LookupPARAM2 = LookupCMD.CreateParameter("Date", adDate, adParamInput, _
   8, dtSalesDate)
LookupCMD.Parameters.Append LookupPARAM2
Set LookupPARAM3 = LookupCMD.CreateParameter("PickupTotal", adCurrency, adParamOutput, 8)
LookupCMD.Parameters.Append LookupPARAM3
Set LookupPARAM4 = LookupCMD.CreateParameter("Fund", adCurrency, adParamOutput, 8)
LookupCMD.Parameters.Append LookupPARAM4
LookupCMD.Execute
'Load the values to coin and currency page
```

```
        lblPickupTotal.Caption = LookupPARAM3
        lblStartFund.Caption = LookupPARAM4
'Remove references to objects
Set LookupPARAM1 = Nothing
Set LookupPARAM2 = Nothing
Set LookupPARAM3 = Nothing
Set LookupPARAM4 = Nothing
Set LookupCMD = Nothing
'Update Register form with data from selected register
frmRegister.Caption = "Register Balance: " & cboRegisterCode.Text
SSTab1.Tab = 1
curTotal = Val(lblCurrencyTotal.Caption) + Val(lblCoinTotal.Caption) _
    - Val(lblStartFund.Caption) + Val(lblPickupTotal.Caption)
sbRegister.Panels("Cash") = "Net Cash:  $" & curTotal
sbRegister.Panels("Total") = "Register Total:  $" & curTotal
txtHundred.SetFocus
End Sub
```

SQL Server Stored Procedure.

```
--This procedure is used to return values for the start-fund & total sum of pickups from the selected register
CREATE PROCEDURE dbo.up_parmsel_PickupLookup
            @RegisterCode AS NVARCHAR(7),
            @Date AS DATETIME,
            @PickupTotal AS MONEY OUTPUT,
            @Fund AS MONEY OUTPUT
AS
SELECT @PickupTotal = SUM(dbo.Pickup_T.Amount)
FROM Pickup_T
WHERE dbo.Pickup_T.RegisterCode = @RegisterCode AND dbo.Pickup_T.SalesDAte = @Date
IF (@PickupTotal IS NULL) -- if no pickups have been made, then assign the variable to 0
BEGIN
            SET @PickupTotal = 0
END

SELECT @Fund = dbo.Register_T.StartFund
FROM dbo.Register_T
WHERE dbo.Register_T.RegisterCode = @RegisterCode
RETURN
GO
```

References

A. Dell Computer Corporation web page. Http://www.dell.com/us/en/bsd/products/model_inspn_inspn_4100.htm. October 27, 2001.

B. Griffin, Greg. Vice President of Sales, DeLaRue / Money Processing Systems. Telephone interview. October 1, 2001.

C. Macdonald, Rob. Serious ADO: Universal Data Access with Visual Basic. New York: Springler-Verlag, 2000.

D. Microsoft SQL Server web site. Http://www.microsoft.com/sql/howtobuy/pricing. October 17, 2001.

E. Microsoft Visual Basic 6.0 web page. Http://www.microsoft.com/catalog/display.asp?site=729&subid=22&pg=8. October 17, 2001.

F. Otey, Michael and Paul Conte. SQL 2000 Developer's Guide. Berkeley: Osborne/ McGraw Hill, 2001.

G. Powers, Darlene. Cash Control Department Head, Paramount's Kings Island. Personal interview. September 29, 2001.