CONTENTS

1. Introduction and Background.................................................................1

2. Narrative.............................................................................................4

3. Project Environment............................................................................18

4. System Design....................................................................................19
   Forms.................................................................................................24
   Modules.............................................................................................39
   Screen Layouts...................................................................................41

5. Summary and Future Work.................................................................44

6. References.........................................................................................46

7. Appendix A – Program Listing............................................................47

8. Appendix B – Sample Batch Report....................................................83

9. Appendix C – Help Screen Printouts....................................................97
LIST OF FIGURES

Figure 1 - Data Flow Diagram.................................................................20
Figure 2 - Table Relationships...............................................................21
Figure 3 - Database Tables.................................................................22
Figure 4 - Forms vs. Tables.................................................................23
Figure 5 - Main Screen Functions..........................................................41
Figure 6 - Administrator Form Functions.............................................42
ABSTRACT

This project is an automated receipt-and-reporting system for the Business Office at Texas A&M University - Corpus Christi. The system provides cashiers and supervisors a convenient means of keeping records of receipts issued to vendors, students, University employees, and University offices and departments. In addition to producing receipts, the system generates daily and monthly reports, retrieves receipts or reports for specific dates or criteria, and stores all information pertaining to receipts. The project is a networked-database system that presents a friendly graphical user interface to enter and retrieve information in a convenient manner.
INTRODUCTION AND BACKGROUND

The Texas A&M - Corpus Christi Business Office maintained records of general receipts issued to vendors, students, University employees, and University offices and departments as well as others with a manual system. A cashier handed a receipt to the customer and kept a carbon copy for University records. At a period during the day, usually the end of the working day, each cashier "batched" his or her receipts. Simply, a batch consisted of a set of receipts and their information for a particular time period of the day. A batch period is usually the entire working day, but several batches can be produced in a day. Batches contained a monetary total of the cashier's receipts for the day, plus calculator-tape totals for various reports. Cashiers also manually prepared a batch/cash reconciliation form for the receipts. Next, the cashiers submitted their batch receipts and totals to the University Accounting Clerk who then entered each receipt into the accounting system. Supervisors then had the tedious task of dissecting the receipts' information to create reports for themselves. Reports spanned periods of time such as months; therefore, receipts were retrieved manually in large booklets where they are kept to complete the reports. Receipts also overlapped into several reports that can make the report-creation process confusing and lead to errors.

Many person-hours were spent creating reports and batching receipts for the day. Often, all types of errors occurred after a batch was made. Typically, errors included mistakes in
arithmetic, overlooking key information for reports, and misplacing receipts. More valuable person-hours were spent determining where errors took place and correcting the mistakes. The Business Office issued 7166 receipts for a total of 23.9 million dollars according to the Office of Comptroller in 1998; therefore, reducing mistakes was a top priority. The Business Office was searching for a solution to eliminate the majority of mistakes and make the receipt tasks of cashiers and supervisors an easier process.

With a database system, errors would occur less frequently and would be quicker to locate. Individual reports, which took an entire business day to create manually, are generated in seconds. New receipts are formatted to print, and searching for old receipts to reprint is not a tedious task. Supervisors know which cashier performs which duty at all times because they too have access to the cashiers’ records. Less data entry takes place because each individual receipt does not have to be entered into the accounting system for record keeping. Finally, the new reports created consolidate daily receipts to enter only totals.

In addition to the several person-hours that are saved, the database system strengthens controls over the University’s assets. The old system could allow for the possibility of undetected recordings of receipts or for the misplacing of receipts. Another feature is that the data is not scattered. The system is on a network site to maintain accuracy among data. Finally, the new system does not allow any prior day’s work to be left open and unrecorded. Cashiers do not have the excuse of lack of time because the database system does practically
all the receipt tasks for them. The system prompts the cashiers to batch their receipts, print all reports belonging to a batch, and submit the information before issuing more receipts.
The database-system users are divided into four contexts: administrators, cashiers, guests, and ex-users. (Note: Ex-users are those former users who no longer have permission to enter the system. Since ex-users’ records are always needed, the ex-user’s information cannot be deleted from the system. The term “user” does not apply to ex-users.) Before a user can use the system, an administrator must add the user to the system. The database system is on a network site, and users of the database system are given permission to the site. Administrators have all the permissions of a cashier, and cashiers have all the permissions of a guest. Administrators have access to all features of the database system that includes: adding or editing users, adding or editing accounts, adding or editing payment types, editing receipts, making design changes to the system, and printing an administrator report. Cashiers are only able to create receipts, create batches that generate receipt reports, or retrieve receipts and batches. Guests can only retrieve receipts or batches. Cashiers and guests have no method of affecting the schema of the database system.

Once the system is running relatively smoothly for a couple of months the supervisors will replace the system developer as the database system administrators. The supervisors know enough about Access97 to use it for its basic features, but cannot handle Access97’s complex security features if a major problem should occur. Therefore, security features have been
implemented that they can more easily understand and implement themselves through a graphical-user interface. Although not as secure as Access97's security features, the database system's security features are adequate for the users of the system.

Security is implemented by hiding all toolbars, menus, the database itself, and certain features of the system to the user. Only the administrators have the proper methods of resetting the system for design editing. Depending on the user's context, the user cannot access all features of the database. For example, a cashier cannot create or edit users. Only administrators are able to create users. After user login, the database system checks the user's context, a field in the Users Table, and disables those features that he or she does not have access.

When a user executes the database program, a login screen prompts the user to enter his or her login-id and password. The database system validates the user and allows access if he or she is indeed a user with a correct password. On the other hand, if the login-id or password is incorrect the user is prompted as to what the error is. If an error does occur, the user is allowed to try to reenter or is given the option to exit the program. When a user enters the system for the first time, the Change-Password Form appears requesting the user to change his or her password. The user's current password is the default password. The form displays the user's current information that is discussed later. The form allows the user to enter his or her password twice to ensure correctness of the password. Once the passwords
pass the validity checks, the user is taken to the Main Screen.

The database system is menu-driven. After the user successfully enters the database system after verification, a menu screen greets the user. This start-up screen, called the "Main Screen" to the user, contains several options available to the user depending on the user's context. Among these options are:

1. Issue Receipt,
2. Retrieve Receipt(s),
3. Make Batch,
4. Retrieve Batch(es),
5. Administrative Tasks,

1. **Issue Receipt** (available only to administrators and cashiers)

The first option of the Main Screen is to issue a receipt. When a cashier or administrator selects this option, a screen that resembles a receipt called the Receipt Form is activated. (Note: At any time, a user may cancel out of this screen by selecting a cancel option.)

Before a user begins typing, several uneditable fields are prefilled for the cashier. The University's name Texas A&M University - Corpus Christi is displayed across the top of the receipt. The date, time, receipt number, and the administrator or cashier's name are shown in different portions of the receipt. A final total field displays the sum of the accounts of the receipt. This total field is linked to the second part of the receipt that is discussed next.
The second part of the receipt contains several areas of input that the administrator or cashier must enter data to complete a receipt. (All fields require input unless otherwise stated.) The first field available for input is the Social Security Number field. This field is optional because the customer may be a student or a non-student. The administrator or cashier then enters the last name, first name, and middle initial of the customer - only the last name is required. Next on the receipt is a group of fields that contains the account information for which the receipt is being issued. This group of fields is the purpose for which the receipt is being issued. (When "account" is mentioned, it refers to the purpose for the issuing of the receipt.) The first field of this group contains a list of frequently used accounts that are available for selection. The user may select an account from the list or enter one if not available on the list. After a user selects an account, the database system automatically retrieves the account number and “plan” for the account. The account number is a unique fifteen-digit number that identifies the account. The third field of the account group is an optional field called Plan. Some accounts have a plan and others do not. A list of plans is provided for those accounts that do have plans. The final field of the account group is the monetary amount owed to the account. A customer may have several accounts for which they are paying; therefore, an administrator or cashier may have to repeat the above process for another account. If the administrator or cashier is done entering account information, he or she enters the last receipt information on the final portion of the receipt.
The final portion of the receipt contains a field for comments, a contact-name field, a contact-extension-number field, a field with a list of payment types, and a grand-total field. The comments field is optional and contains any comments that a user may have in regards to a transaction. Cash, credit cards, Sandollar Cards, and checks are the usual sources of payment, and the payment-type field holds these values. If the payment type is of type "check", then a field appears asking the cashier to enter the check number. The contact-name field is optional and is the name of the department the receipt deals with if the department is not listed as the customer. The contact-extension-number field is optional and is the phone extension of the department. The final field is a grand-total field that is a sum of all the costs related to the account portion of the receipt. This field automatically sums the costs and is locked to the user. After an administrator or cashier finishes entering all information, he or she has two options: Finish and Cancel. If the administrator or cashier selects Cancel, the receipt is deleted after the administrator or cashier is given a second chance to save the receipt. If the user selects Finish, the system prompts the administrator or cashier that the receipt is going to be saved and printed. The user has a chance at this point to delay the saving and printing and edit the receipt for mistakes. If the user chooses to save and print, the receipt is saved to the database and sent to a printer. The administrator or cashier then returns to the Main Screen.

2. Retrieve Receipt(s) (available to administrators, cashiers, and guests)

The second option of the Main Screen is the “Retrieve Receipt(s)” Option. When a user
selects this option, a form called Retrieve-Receipt(s) Form greets the user. This section of the system is divided into two forms: the Retrieve-Receipt(s) Form and the Retrieve-Receipt(s)-More-Options Form. The first form contains two categories and the last form contains one category that each operate independently of each other.

The first category of the Retrieve-Receipt(s) section gives the user the ability to retrieve a receipt based on criteria the user enters. A user may enter information into one or more criteria fields. (If a field is left blank, the database system matches all data for that field.) The criteria fields include the date, a list of user names including ex-users, receipt number, social security number, and name of the customer. This category contains its own "Finish" option. When the Finish option is selected, the screen changes to the Receipt Form of the first option of the Main Screen, but there are differences. Unlike the original receipt screen, there are no editable fields. The user may scroll through several screens which each contain individual receipts previously submitted by cashiers or administrators. Each receipt matches the criteria entered by the user. A user has the option to print the receipt or receipts or return to the Main Screen.

The second category of the Retrieve-Receipt(s) section consists of a single date field. If a user enters a date in this field and selects the category's own Finish option, all receipts for that date are displayed in a report format that lists all receipts one after each other. The user views this report before printing. The user has the option to print the report or return to the
Main Screen.

The Retrieve-Receipt Form has an option called “More Options” that closes the Retrieve-Receipt Form and opens the Retrieve-Receipt(s)-More-Options Form. The final category of the Retrieve-Receipt(s) section is on this form. This category’s purpose is to generate a report that lists receipts based on an account and a range of dates. The user selects an account from the list of available accounts or enters one in the account field then enters a range of dates into two date fields: "Beginning Date" and "Ending Date". The generated report based on the entered criteria is crucial to certain supervisors. The report contains a summary of all receipts that match the account and the range of dates. The user again previews the report before printing. The user may opt to print or return to the Main Screen.

3. **Make Batch** (available only to administrators and cashiers)

The third option of the Main Screen is the "Make Batch" option that leads to the Batch Form. (A batch is a set of receipts with key reports, based on the receipts, generated by the database system.) The administrator or cashier selects this option when they must batch their receipts during the day, usually the end of the working day. This form's functions automatically generate the reports needed by supervisors of certain departments. The form has four fields related to batch information. For a cashier, the first field contains his or her name because a cashier can only create batches for himself or herself. For an administrator, the first field contains a list of all user names including ex-users because an administrator can
create batches for anyone. After the selection of a name, a group of fields automatically fill that pertain to the previously created batch by the selected name. These "Last Batch Created" fields are Date, Starting Time, and Closing Time. These fields are merely to refresh the administrator or cashier's memory, and this function is useful when several batches are made in a day. Next, the administrator or cashier has two options: Make Batch or Cancel. If the administrator or cashier decides to use the Cancel option, the administrator or cashier returns to the Main Screen. Else if the user selects the Make-Batch option, the administrator or cashier is prompted with a message similar to: "A new batch will be created for user Linda for the date of 1/26/99 between the times of 8:03 AM and 5:00PM." The user or ex-user's name, the date, and times automatically are filled for the administrator or cashier by the system. This prompt has two options: OK or Cancel. If the administrator or cashier selects the Cancel option, the administrator or cashier returns to the previous screen. Else if the administrator or cashier selects the OK option, the administrator or cashier is prompted that the reports for the batch are being printed. The system generates several reports. These reports separate into several categories:

1. Group by Cashier,
2. Group by Account Number,
3. Group by Payment,

The Cashier Report groups by the user. The report displays all batch receipts, the receipts' information, and a grand total of all receipts' totals made by a user or ex-user for a batch.
period. The Account-Number Report is two separate reports. The first account-number report groups the batch receipts by account number for credit card transactions. The second account-number report groups the batch receipts by account number for non-credit-card transactions. The Payment Report follows the same example. The first payment report groups batch receipts by payment type for credit card transactions. The second payment report groups batch receipts by payment type for non-credit-card transactions. The final report groups batch receipts by plan. Not all accounts on receipts have plans since it is an optional field for receipts as previously stated. Those accounts on batch receipts that have a plan are displayed by the report. After the system sends all reports to the printer, the administrator or cashier automatically returns to the Main Screen.

4. **Retrieve Batch(es)** (available to administrators, cashiers, and guests)

The fourth option of the Main Screen is the "Retrieve Batch(es)" option. If the user selects this option, the user has the ability to retrieve any batch ever made by any user and print the batch. The batch, as stated above, contains all receipts for the time period of the batch and the reports based on these batch receipts. The Retrieve-Batch Form retrieves a single batch or several batches based on criteria the user selects or enters. The user selects a user name including ex-users provided by a list in the Cashier-Name field. Next, the user enters a date for the batch in the Batch-Date field. (If a field is left blank, the database system matches all data for that field.) The user then has two options: Cancel or Finish. If the user selects the Cancel option, then the user returns to the Main Screen. If the user selects the Finish
option, the screen changes to the Batch Form of the third Main Screen Option with a few differences. The user is able to scroll through the batch information the system matches. This information is the batch date, the batch start time, and the batch closing time. The user has the ability to print or preview any batch corresponding to the information provided using a "Print Batch" or "Preview Batch" option. When the user is done printing, the user returns to the Main Screen using a "Cancel" option.

5. **Administrative Tasks** (available only to administrators)

The fifth option of the Main Screen is the “Administrative Tasks” option. This option is divided into its own category of options. The Administrator Form options are as follows:

a.) Add User,

b.) Edit User,

c.) Add/Edit Payment Types,

d.) Edit Accounts,

e.) Edit Receipts,

f.) Startup Properties,

g.) Administrator Report,

h.) Close.

5a) **Add User**

This option allows the administrator to add a new user to the system. When this option is
selected, the User Form opens ready for a new user to be entered. The form has the following fields: First Name, Middle Initial, Last Name, User Name, Login ID, Context, and Email Address. The first name, middle initial, and last name of the new user goes into the name fields. The middle initial is optional. The User Name field is automatically filled by the system that does so by combining the name fields. The Login ID is the unique identifier of the user. The Context is the context of the user that may be administrator, cashier, or guest. The Email-Address field is optional. The form has three options: Print All Users, Undo Changes, and Finish. The Print-All-Users option prints a report that lists all users. The Undo-Changes option undoes any changes made to the form. (This option can be used when an administrator wants to clear the form to not add the user.) The final option of the form is the Finish option. This option saves the new user’s record if all required fields have data and closes the form.

5b) Edit User

This option allows the administrator to edit an existing user in the system. When this option is selected, a messagebox appears asking the administrator to choose a user name from the list provided. The administrator may choose to return to the Administrator Form by choosing a Cancel option. If the administrator chooses a name from the list, the User Form appears with the chosen user’s information. The form is in the same format as in 5a above. The administrator may edit any field of the user’s record except the User-Name field that automatically updates itself. The Reset-Password option is the only change to the User
Form. An administrator uses this option to reset the user’s password if forgotten.

5c) Add/Edit Payment Types

This option allows the administrator to add or edit the payment types of the system. This option leads to the Payment-Types Form. This form has two fields: the Payment ID and Payment-Type Name. The form has one checkbox named Credit Card that is a Boolean value. The Payment ID is an integer value that uniquely identifies the payment type. The Payment-Type Name is the name of the payment type. The Credit-Card checkbox labels the payment type as credit card or non-credit card. This form has four options. The Print-All-Payment-Types option prints a report that lists all the payment types of the system. The Add-New-Payment-Type option prepares the form for new entry to the payment types. The Undo-Changes option undoes any changes a user has made to a payment-type. Finally, the Finish option saves any new payment types and closes the form.

5d) Edit Accounts

This option allows the administrator to add or edit the accounts of the system. This option leads to the Accounts Form. This form has three fields: the Account Number, Account Name, and the Plan. The Account Number is a fifteen-digit integer value that uniquely identifies the payment type. The Account number must pass certain validity checks. If the number fails one of the checks, the administrator is prompted as to the failure. The Account Name is the name of the account. This form has four options. The Print-All-Accounts option
prints a report that lists all the accounts of the system. The Add-New-Account option prepares the form for new entry to the accounts. The Undo-Changes option undoes any changes a user has made to an account. Finally, the Finish option saves any new accounts and closes the form.

5e) **Edit Receipts**

This option allows the administrator to edit receipts. When this option is chosen, the administrator is taken to the Retrieve-Receipts Form of Main-Screen Option Two. There is only one difference. The Allow-Edits flag is set to true to allow editing of receipts. When a user enters the Retrieve-Receipts Form though the Main Screen, the Allow-Edits flag is set to false for any user.

5f) **Startup Properties**

This option allows the administrator to reset the database to allow changes or updates to the design schema. When selected, the administrator is prompted first with a message that warns the administrator to make sure all users are off the system before making updates. Next, the Access97 Startup Dialog Box from the Tools Menu of the default Access97 menu bar appears. This dialog box allows the administrator to set the startup properties of the database system. Examples of startup properties are: Display Form, Display Database Window, Menu Bar, and Allow Viewing Code After Error. The administrator makes the necessary changes to the dialog box before and after updates.
5g) **Administrator Report**

This option allows the administrator to view an administrator report that verifies the cashier’s batch reports. When the administrator chooses this option, a message box appears that prompts for the dates to review – the Beginning Date and the Ending Date. The administrator has two options: Cancel or Finish. If the administrator chooses Cancel, the administrator is returned to the Main Screen. The Finish option displays the Administrator Report in print-preview mode. The report shows credit card and non-credit card totals for each cashier for the dates requested. The report also displays any blank receipts a cashier may have created. Blank receipts are receipts with no accounts. The administrator can close the report and return to the Main Screen or print and close the report and return to the Main Screen.

6. **Exit** (available to administrators, cashiers, and guests)

The final option of the Main Screen will be the "Exit" option. When this option is selected, the user exits the database system.
The database system was created using the Microsoft Access97 application. The master file resides in a network folder on Hummer where access to the network folder must be granted to all users of the database system. Access to the network folder is granted by Computer Services at Texas A&M University - Corpus Christi, and the database system user has access to the folder after login through Novell. The users of the system generally are located at the Business Office at Texas A&M University - Corpus Christi, but other users and guests of the system may be located throughout the campus as long as their computers are among the university network computers. The PC must have The Microsoft Access97 application to run the database system. The PC at a minimum must have a 486 or higher processor, the Microsoft Windows 95 operating system, and 12 MB of memory. Access97 uses the Visual Basic for Applications language for methods, procedures, and functions. The Visual Basic for Applications language provided by Access97 is used extensively throughout the system.
SYSTEM DESIGN
Figure 1 - DATA FLOW DIAGRAM
Figure 2 - TABLE RELATIONSHIPS
<table>
<thead>
<tr>
<th>USERS</th>
<th>BATCHES</th>
<th>RECEIPTS</th>
<th>RECEIPTS ACCOUNTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Login ID</em></td>
<td><em>Batch ID</em></td>
<td><em>Login ID</em></td>
<td><em>Receipt Account ID</em></td>
</tr>
<tr>
<td>Password</td>
<td>Login ID</td>
<td><em>Receipt Number</em></td>
<td>Login ID</td>
</tr>
<tr>
<td>Last Name</td>
<td>Batch Date</td>
<td>Date</td>
<td>Receipt Number</td>
</tr>
<tr>
<td>First Name</td>
<td>Start Time</td>
<td>Time</td>
<td>Account ID</td>
</tr>
<tr>
<td>Middle Initial</td>
<td>Closing Time</td>
<td>SSN</td>
<td>Account Name</td>
</tr>
<tr>
<td>User Name</td>
<td></td>
<td>Last Name</td>
<td>Plan</td>
</tr>
<tr>
<td>Email Address</td>
<td></td>
<td>First Name</td>
<td>Amount</td>
</tr>
<tr>
<td>Context</td>
<td></td>
<td>Middle Initial</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Form of Payment</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Check Number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contact Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contact Ext#</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not Batched</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACCOUNTS</th>
<th>PAYMENT TYPES</th>
<th>STUDENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Account ID</em></td>
<td><em>PaymentID</em></td>
<td><em>SSN</em></td>
</tr>
<tr>
<td>Account Name</td>
<td>Payment Type Name</td>
<td>Last Name</td>
</tr>
<tr>
<td>Plan</td>
<td>Credit Card</td>
<td>First Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Middle Initial</td>
</tr>
</tbody>
</table>

*Denotes Primary Key

Figure 3 - TABLES
<table>
<thead>
<tr>
<th></th>
<th>USERS</th>
<th>BATCH</th>
<th>RECEIPTS</th>
<th>RECEIPTS ACCOUNTS</th>
<th>ACCOUNTS</th>
<th>PAYMENT TYPES</th>
<th>STUDENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Form</td>
<td>S</td>
<td></td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main Form</td>
<td></td>
<td></td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change-Password Form</td>
<td>S,U</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Form</td>
<td>S,I,U</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipt Form</td>
<td>S,I,U,D</td>
<td>S,I,U,D</td>
<td>S,I,U,D</td>
<td>S</td>
<td>S</td>
<td>S,I,U</td>
<td></td>
</tr>
<tr>
<td>Receipt-Accounts Subform</td>
<td>S,I,U,D</td>
<td>S</td>
<td>S,I,U,D</td>
<td>S</td>
<td>S</td>
<td>S,I,U</td>
<td></td>
</tr>
<tr>
<td>Batch Form</td>
<td>S</td>
<td>S,I</td>
<td>U</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payment-Types Form</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S,I,U</td>
<td></td>
</tr>
<tr>
<td>Accounts Form</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S,I,U,D</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S = SELECT (read)  
I = INSERT  
U = UPDATE  
D = DELETE

Figure 4 - FORMS VS. TABLES
FORMS

Access97 Forms have class modules that are used extensively by the Receipts-database system. Some methods that are used frequently are part of a database module that is global and is able to be used by any form module or procedure throughout the system. (These modules are the Batch, Receipt, and Security Module.) One feature that every form of the system contains is access to Help. By pressing the F1 key of the keyboard, the user can obtain help for any form. The common format of the help system is a description of the form and common errors the form produces.

LOGIN FORM

Input: unique login-id, password, Users Table, Receipts Table.

Methods: Exit_Button_Click(), Finish_Button_Click(), Set_Context(), Any_Unbatched_Receipts(), Apply_Main_Form_Security().

Event Buttons: After entering their login-id and password, the system searches the Users Table for the login-id after the user presses the Finish Button that executes the Finish_Button_Click() method. If the login-id is found, then the password found in the table is compared to the password on the form. If the passwords match, access is granted; else, a warning is issued. The warning occurs because of an incorrect login-id or password. During login, the Set_Context() method executes which displays the name of the user and the context of the user on the Main Form. Also during login, the
Apply_Main_Form_Security() method executes that disables those features not allowed to non-administrators. A final method that executes during login is Any_Unbatched_Receipts(). This method searches the Receipts Table for receipt records that are unbatched for dates prior to today. If the receipt record is unbatched, the Not-Batched field of the record is checked true. At any time the user can exit the system by pressing the Exit Button that executes the Exit_Button_Click().

Output: If login is successful, the user is allowed to enter the Main Form and can proceed with its procedures and functions. The user's name and context are displayed across the top of the Main Form. Features in the Main Form are disabled if a user does not have permission. If the user has any unbatched receipts, the Batch Form opens to batch. The Batch Form is discussed later.

MAIN FORM

Input: Receipts Table.

Methods: Issue_Recipient_Button_Click(), Initialize_Receipt(), Retrieve_Recipient_Button_Click(), Make_Batch_Button_Click(), Retrieve_Batch_Button_Click(), Administrative_Tasks_Button_Click(), Exit_Button_Click()

Form Functions: The form stays open at all times because it contains three global variables that are necessary throughout the system. They are CurrentUser, nextrerreur, and Context. The CurrentUser variable contains the login-id of the current user. The nextrerreur variable
holds the value of the next receipt to be issued by the current user. The Context variable holds the context of the current user that is needed for security measures. The form has five buttons linked to forms. When the Issue-Receipt Button is pressed, the \textit{Issue\_Receipt\_Button\_Click()} method executes that opens the Receipt Form. Also during this method, the \textit{Initialize\_Receipt()} method, located in the Receipt Module, executes that initializes a new receipt when the Receipt Form opens. When the Retrieve-Receipt Button is pressed, the \textit{Retrieve\_Receipt\_Button\_Click()} method executes that opens the Retrieve-Receipt Form. When the Make-Batch Button is pressed, the \textit{Make\_Batch\_Button\_Click()} method executes that opens the Batch Form. When the Retrieve-Batch Button is pressed the \textit{Retrieve\_Batch\_Button\_Click()} method executes that opens the Retrieve-Batch Form. When the Administrative-Tasks Button is pressed the \textit{Administrative\_Tasks\_Button\_Click()} method executes that opens the Administrator Form. The Exit Button executes the \textit{Exit\_Button\_Click()} that exits both the database system and Access97.

**RECEIPT FORM & RECEIPTS-ACCOUNTS SUBFORM**

**Input:** date, time, receipt number, cashier's name, social security number, last name, first name, middle initial, account name, account number, account plan, total field, comments, payment type, grand-total, check number, contact name, contact ext#, Students Table, Users Table, Receipts Table, Receipts-Accounts Table, and Accounts Table.

**Methods:** \textit{Finish\_Button\_Click()}, \textit{print()}, \textit{date()}, \textit{time()}, \textit{SSN\_AfterUpdate()}, \textit{Update\_Student\_Info()}, \textit{Delete\_Record\_Click()}, \textit{Print\_Report\_Click()},

26
Close_Button_Click()

**Procedures:** The Receipts-Accounts Subform is located within the Receipt Form and is linked to the Receipt Form through the receipt number. The receipt number and cashier's name are automatically entered through the Initialize_Receipt() method that executes previously through the Main Form as discussed previously. The date and time are system parameters and are retrieved through date() and time() system calls. If a cashier enters a social security number, the field executes the SSN_AfterUpdate() method which searches the Students Table for a match with the social security number field. After the user selects an account name, the field executes the Account_Name_AfterUpdate() which sets the account id and plan fields. The account-name field will have an SQL statement as its record source that selects all the accounts information from the Account Table.

**Event Buttons:** The Finish Button will execute the Finish_Button_Click() method which prompts the cashier that the receipt is going to be saved and printed. If the cashier chooses to continue, the Finish_Button_Click() method saves the receipt and executes a print method, Print_Report_Click(), that prints the receipt. The receipt is saved in the Receipts Table and the Receipts-Accounts Table. Next, the Finish_Button_Click() method updates the customer information if a social security number is entered by executing the Update_Student_Info() method, located in the Receipts Module, that updates changed information in the Students Table. The Cancel Button executes the Delete_Record_Click() method that checks the Receipts Table and the Receipts-Accounts Table and deletes any information that may have been entered by the Receipt Form. The form contains two more buttons that are viewed only
when the form is in read-only mode during retrieval of receipts. This mode is entered through the Retrieve-Receipt Form that is discussed later. The first button is the Close Button that executes the Close_Button_Click() that closes the form. The second button is the Print Button that executes the Print_Report_Click() that prints the receipt.

Output: The receipt is sent to the printer for printing, the receipt is added to the Receipts Table, the accounts' information from the receipt is added to Receipts-Accounts Table, the student information is updated or inserted in Students Table.

RETRIEVE-RECEIPT FORM (SECTION I)

Input: date, cashier names, receipt number, last name of customer, Users Table

Methods: Finish_Button_Click(), Apply_Receipt_Form_When_Retrieving_Security(int), More_Options_Click(), Cancel_Button_Click()

Procedure: The cashier name field will have an SQL statement as its record source that selects all user names from the Users Table. When the Retrieve-Receipt Form first opens, Access will execute the SQL statement.

Event Buttons: The form has a Cancel Button that executes the Cancel_Button_Click() method that closes the form, and the More-Options Button that executes the More_Options_Click() that opens the Retrieve-Receipts2 Form. After the user is done entering criteria information for the above input fields (if a field is left blank, the Finish_Button_Click() method of the Finish Button matches all data just like an asterisk(*) would in environments such as Unix), a query searches the Receipts Table for receipts
matching the criteria. For those receipts matched correctly, the query also retrieves the accounts from the Receipts-Accounts Table which match the receipt number. The receipt data is displayed on the Receipt Form with differences. The `Finish_Button_Click()` method also executes the `Apply_Receipt_Form_When_Retrieving_Security(int)` method that is located in the Security Module. This method opens the Receipt Form in edit mode for administrators and read-only mode for non-administrators. (The query that matches the criteria is the record source for the Receipt Form; therefore, the form actually executes the query when the Receipt Form opens. The form has a button that prints instead of the Finish Button. The Print Button prints the current receipt on the screen. Only one receipt is shown at a time, but the user is able to scroll through all the receipts that matched the criteria. The scrolling ability is a built-in feature provided by Access97 as a set of navigation arrows at the bottom of the screen.)

**RETRIEVE-RECEIPT FORM (SECTION II)**

**Input:** date

**Methods:** `All_Receipts_Click()`

**Event Buttons:** After the user is done entering a date for the date field, a query searches the Receipts Table for receipts matching the date. This query is executed through the `All_Receipts_Click()` method of the second Finish Button. For those receipts matched correctly, the query also retrieves the accounts from the Receipts-Accounts Table that matches the receipt number. The receipts are presented in a report that sequentially lists the
receipts for the date. The query that matches the date is the record source for the report; therefore, the report actually executes the query when the report is opened. The report is viewed in a print-preview state similar to any print preview state in a Windows application. The user then has the option of printing the report when done viewing.

**Output:** The form creates the report of receipts.

**RETRIEVE-RECEIPT2 FORM (more options)**

**Input:** account id, beginning date, ending date, Receipts Table, Receipts-Accounts Table, and Accounts Table.

**Methods:** Finish_Button_Click(), Cancel_Button_Click()

**Procedure:** This form is the continuation of the Retrieve-Receipt Form. The account name field has a SQL statement as its record source that executes when the form first opens. The statement selects all account ids and account names from the Accounts Table.

**Event Buttons:** The purpose of the form is to generate a report with those receipts that match the input criteria. The criteria is an individual account and a range of dates. After the user is done entering criteria in the fields, the user selects the Finish Button that executes the Finish_Button_Click() method. The method executes a query that searches the Receipts Table for receipts matching the account and the range of dates. For those receipts matched correctly, the query also retrieves the accounts from the Receipts-Accounts Table that matches the receipt number. The receipts are presented in a report that sequentially lists the receipts. Not all the receipt information is printed; therefore a different format is used. The
query that matches the criteria is the record source for the report; therefore, the report actually executes the query when the report is opened. The report is viewed in a print-preview state similar to any print preview state in a Window's application. The user then has the option of printing the report when done viewing. The form also has a Cancel Button that executes the *Cancel_Button_Click()* method that closes the form.

**Output:** The form produces the receipt of reports.

**BATCH FORM**

**Input:** cashier name, Users Table, Batch Table, Receipts Table.

**Methods:** *Login_ID_AfterUpdate(), Preview_Button_Click(), Print_Batch_Click(), Cancel_Button_Click(), Make_Batch_Button_Click(), Print_Batch_Click(), Add_Batch(string,string,string,string)*

**Procedures:** The login-id/cashier-name field has an SQL statement, which selects all cashier names form the Users Table, as its record source. (Note: cashiers can only create batches for themselves - administrators can make batches for all.) After a cashier or administrator selects a name from the drop-down list field, the *Login_ID_AfterUpdate()* method executes. This method searches the Batch Table for the last batch created by the cashier selected. The query extracts the following information: date, starting time, and closing time and places them in their corresponding fields.

**Event Buttons:** The form contains four buttons: Make Batch, Cancel, Preview Batch, and Print Batch. The Preview-Batch Button and the Print-Batch Button are viewed only in
retrieval mode. This mode is entered though the Retrieve-Batch Form that is discussed next.

The Preview-Batch Button executes the Preview_Button_Click() which allows the user to view the Batch reports before printing them. The Print-Batch Button executes the Print_Batch_Click() method that allows the user to print the Batch reports for the current batch displayed on the screen. The Cancel Button executes the Cancel_Button_Click() method which returns to the Main Screen. The Make-Batch Button executes the Make_Batch_Button_Click() method. First, this method prompts the user with a messagebox with the date of the batch, the starting and closing time of the batch, and the cashier's name. (The starting time is derived by taking the closing time of the previous batch. If the previous batch’s closing time is from the day before, the time of 12:00:00AM is used.) If the cashier or administrator chooses to continue, the user is be prompted that the reports for the batch are printing and the printing is done through a call to the Print_Batch_Click() method. During the Make_Batch_Button_Click() method, the batch period information is added to the Batch Table through the Add_Batch(string,string,string,string) method located in the Batch Module. The date, starting time, and closing time is added along with the login-id of the cashier. The Add_Batch(string,string,string,string) method also searches the Receipts Table and unchecks those receipts for the batch period that are labeled "Not Batched". The "Not Batched" attribute is a field in the Receipts Table.

Output: Several reports are generated. The first report the Cashier report uses a query which displays all the receipts for the batch period for that particular cashier with a grand total. The final categories of reports: Account Number, Payment, and Plan are each
separated into two reports - Credit Card and Non-Credit-Card transactions. The individual queries group by the category with the credit card constraint.

**RETRIEVE-BATCH FORM**

**Input:** cashier name, batch date, Users Table, and Batch Table.

**Methods:** Finish_Button_Click(), Cancel_Button_Click()

**Procedures:** The login-id/cashier-name field has a SQL statement as its record source that selects all users' names from the Users Table. The user then types a date; otherwise, if left blank, all dates are retrieved for the cashier.

**Event Buttons:** The Finish Button executes the Finish_Button_Click() method which opens the Batch Form with differences. (The Batch Form is read-only and has a print and cancel button. The form has the select query as its record source; therefore, only those records that match the criteria are shown one screen at a time. The user is able to scroll through the batches with the built-in navigation buttons. The information provided is the cashier name, login-id, date, starting time, and closing time. If a user sees a particular batch in which he or she wishes to print, the user presses the Print Button as discussed earlier in the Batch Form section.) The user may exit with the Cancel_Button_Click() method of the Cancel Button.

**USER FORM**

**Input:** name, login-id, context, email address, password, Users Table.

**Methods:** Finish_Button_Click(), Undo_Button_Click(), Print_All_Users_Button_Click(),
Reset_Button_Click()

**Event Buttons:** The Finish Button executes the method `Finish_Button_Click()` that excepts the input and creates the appropriate record for the user specified in the Users Table and exits the form. The Undo Button executes the `Undo_Button_Click()` which undoes any changes to the form. The Print-All-Users Button executes the `Print_All_Users_Button_Click()` that creates a report that lists all users of the system. The Reset-Password Button executes the `Reset_Button_Click()` that resets the password of the user to the system default.

**Output:** The user is added to the Users Table. If the Print-All-Users Button is pressed then a report is generated that lists all the users of the system.

**ACCOUNTS FORM**

**Input:** account number, account name, plan, and Accounts Table.

**Methods:** `Finish_Button_Click()`, `Undo_Button_Click()`, `Print_Accounts_Click()`, `New_Account_Button_Click()`

**Procedures:** The form is directly linked to the Accounts Table, and shows each account record one page at a time. The information provided is the account number, the account name, and the plan if it applies. The user may scroll through the accounts using the built-in scroll navigation buttons.

**Event Buttons:** The Finish Button executes the method `Finish_Button_Click()` that excepts the input and creates the appropriate record for the account specified in the Accounts Table and exits the form. The Undo Button executes the `Undo_Button_Click()` which undoes any
changes to the form. The Print-All-Accounts Button executes the Print_Accounts_Click() that creates a report that lists all users of the system. The Add-New-Account Button executes the New_Account_Button_Click() that prepares the form for data entry.

Output: A report is generated for the Print-All-Accounts Button that simply lists all the information for every account record. The Accounts Table is updated whenever a record is changed or a new recorded is added.

PAYMENT-TYPES FORM

Input: payment id, payment-type name, credit card, and Payment-Types Table.

Methods: Finish_Button_Click(), Undo_Button_Click(), Print_PayT_Click(), New_Payment_Type_Button_Click()

Procedures: The form is directly linked to the Payment-Types Table, and shows each payment-type record one page at a time. The user may scroll through the payment type using the built-in scroll navigation buttons.

Event Buttons: The Finish Button executes the method Finish_Button_Click() that excepts the input and creates the appropriate record for the payment type specified in the Payment-Types Table and exits the form. The Undo Button executes the Undo_Button_Click() which undoes any changes to the form. The Print-All-Payment-Types Button executes the Print_PayT_Click() that creates a report that lists all payment types of the system. The Add-New-Payment-Type Button executes the New_Payment_Type_Button_Click() that prepares the form for data entry.
Output: A report is generated for the Print-All-Payment-Types Button that simply lists all the information for every payment-type record. The Payment-Types Table is updated whenever a record is changed or a new recorded is added.

ADMINISTRATOR FORM

Methods: Adduser_Button_Click(), Edit_User_Button_Click(),
Edit_Payments_Button_Click(), Edit_Accounts_Button_Click(),
Edit_Receipts_Button_Click(), AdminRep_Button_Click(), Close_Button_Click()

Event Buttons: The Add-User Button executes the Adduser_Button_Click() method that opens the User Form, discussed previously, for data entry only. The Edit-User Button executes the Edit_User_Button_Click() method that prompts the administrator for a user’s name. A list is provided for the administrator. After choosing a name or canceling, the User Form opens with the chosen user’s record. The Add/Edit-Payment-Types Button executes the Edit_Payments_Button_Click() method that opens the Payment-Types Form for editing or adding a new payment type. The Add/Edit-Accounts Button executes the Edit_Accounts_Button_Click() method that opens the Accounts Form for editing or adding new accounts. The Edit-Receipts Button executes the Edit_Receipts_Button_Click() method that opens the Retrieve-Receipts Form. The Retrieve-Receipts Form’s edit-permission flag is set to true to allow the administrator to edit receipts. The Startup-Properties Button executes a macro named “Reset”. When selected, the administrator is prompted first with a message that warns the administrator to make sure all users are off the system before
making updates. Next, the Access97 Startup Dialog Box from the Tools Menu of the default Access97 menu bar appears. This dialog box allows the administrator to set the startup properties of the database system. The Administrator-Report Button executes the AdminRep_Button_Click() method that creates an administrator report. The Close Button executes the Close_Button_Click() method that closes the form.

**Output:** The Administrator Report is produced.

### CHANGE-PASSWORD FORM

**Input:** passwords, Users Table

**Methods:** Form_Open(Integer), Finish_Button_Click(), Cancel_Button_Click()

**Procedure:** When the form first opens, the form executes the Form_Open(Integer) method that automatically opens the form to the user’s record.

**Event Buttons:** The user enters his or her password twice to ensure correctness. The user then presses the Finish Button that executes the Finish_Button_Click() method. This method checks if the password has been changed, that the password is not empty, and that both passwords match. If the passwords meet all the checks, then the password is saved to the Users Table. The form closes and the user is prompted that the password was changed. The Cancel Button executes the Cancel_Button_Click() method that closes the form. This method warns the user that it is highly recommended to change his or her password at this time. The user can choose to change his or her password or close the form.

**Output:** The new password is saved to the Users Table.
MODULES

BATCH MODULE

The batch module contains the \texttt{Add\_Batch\(\text{string, string, string, string} \)} method. This method takes the login ID, date, starting time, and closing time of a batch and adds the variables as a record to the Batch Table. This method also updates the receipt records of the Receipts Table that fall on the batch date between the starting time and closing time. The records contain a field called “Not Batched” and they are checked false to indicate that they have been batched.

RECEIPT MODULE

The Receipt Module contains four methods. The first is the \texttt{Initialize\_Receipt()} method that initializes a new receipt to be issued to a customer. The method searches the Receipts Table to find the last receipt number issued by the user then adds one to it to arrive at the new receipt number. The second method is the \texttt{Any\_Unbatched\_Receipts()} that searches the Receipts Table to find an unbatched receipt for a day prior to the day’s date. The third method is the \texttt{Update\_Student\_Info()} that updates students records within the Receipt Form when a receipt is being issued. The final method is the \texttt{Account\_Check\(\text{String} \)} which verifies account numbers when entered in the Receipt Form or the Accounts Form.
SECURITY MODULE

The Security Module applies security throughout the system depending on the context of the user. Guests have fewer privileges than cashiers, and cashiers have less privileges than administrators. The Apply_Batch_Form_Security() applies security to the Batch Form for certain reasons. Cashiers can only batch receipts for themselves and not for others; therefore, the login-id/cashier-name field contains a list, but only with the current cashier's name. Administrators can batch receipts for any user; therefore, the login-id/cashier-name field contains a list with all users name. The method ensures these properties. The Apply_Main_Form_Security() applies security to the Main Form (Main Screen). Guests can only retrieve receipts and batches. Cashiers can create or retrieve receipts or batches. Administrators can perform all these tasks plus the many administrative tasks. To ensure these constraints, the method enforces them. The Issue-Receipt, Make-Batch, and Administrative-Tasks buttons are hidden to guests, and the Administrative-Tasks Button is hidden to cashiers. The final method is the Apply_Receive_Form_When_Retrieving_Security(Integer). This method sets an edit-permission flag in the Retrieve-Receipts Form that allows or disallows editing of receipts after retrieval. Only administrators have this permission.
SCREEN LAYOUTS
Figure 5 – MAIN SCREEN FUNCTIONS
SUMMARY AND FUTURE WORK

The Receipt-and-Reporting database system is a user-friendly system that guides the user throughout the system with graphical feedback, such as message boxes, to let the user know what is happening at all times. Many of the users, in particular the cashiers, have never used Access97 and many have little experience using graphical-user interfaces. The users have had an easy time using the system. Options per screen are limited to avoid confusing the user, and after a short period of time the user masters the database system’s features. Supervisors are relieved because of the person-hours saved due to the database system’s reports. What would take the administrators an entire working day to construct can be created in seconds by the database system. Now, the supervisors and the university have an effective system of storing and retrieving receipt data. Since the data is stored in a network folder, access by outsiders is forbidden and the data can be accessed throughout the campus. The database system has replaced the manual system that is slow, tedious, and inefficient for a university system.

Future work may include updating the system to the new Office 2000 product. The new Office 2000 product may make updating the system easier and less confusing. More reports may be needed because the supervisors are constantly trying to find methods of double checking their work or making their job easier. Administrators may also find a method of obtaining all available accounts from the accounting system and adding them to the database.
system in an efficient manner. Currently, there is a method of retrieving the accounts, but adding them to the system was too complex for the future administrators. Plus, the accounts were incomplete which made the process even more confusing.

Another issue that needs to be addressed is the number of receipts the system can hold before it needs to replaced or updated. In 1998, there was a little over 7000 receipts issued. If for now on 10000 are issued per year, in ten years that would mean 100,000 receipts in the system. At that point the system administrators might consider an upgrade or replacement of the system because of Access 97 limitations.
REFERENCES


