Client Information System

By

Phil Infante

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

February 28, 2005
Client Information System

By

Phil Infante

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 2005 Phillip Infante

The information in this document is proprietary and may not be reproduced or distributed in whole or in part without the permission of the owner.

Phil Infante
Date 2/28/05

Dr. Hazem Said, Faculty Advisor
Date 2/28/05

James F. Sullivan, Department Head
Date 28 February, 2005
Acknowledgements

I would like to give special thanks to Lisa Veneziano, General Motors Service Parts Operation (GMSPO) warehouse plant manager in Cincinnati, and Cathy Werner, Customer Liaison Supervisor for the Customer Service Department, for nurturing their idea with upper management and believing in my proposal and in my skills to carry the project to its completion. I could not have completed the project without your encouragement and support. I would like to thank John Seigal, Jamil Brown, Jeff Turiak, and Gene Hart, the members of the Customer Service Department, for all their opinions, ideas, and help in order to clarify what the project was all about to the members of the Customer Service Department. I would like to give special thanks to my wife, Nancy Infante, for enduring the last five years of long weekends and weekdays of school work. The patience has finally paid off, and I could not have done it without my wife's emotional support and patience. Finally, I dedicate my project to my daughter Elisabeth. If you stick to completing your goal, only good things can come from your hard work.
# Table of Contents

**Section** | **Page**
--- | ---
Acknowledgements | iii
Table of Contents | iv
List of Illustrations | v
Abstract | vii
1. Description and Intended Use | 1
  1.1. Statement of the Problem | 1
  1.2. Solution | 2
2. Solution | 2
  2.1. User Profile | 3
  2.2. Design Protocols | 3
  2.3. Database Design | 4
  2.4. Programming Coding Standards | 5
    2.4.1. Programming VB Coding Standards | 6
    2.4.2. Programming SQL Coding Standards | 8
3. Deliverables | 8
4. Design and Development | 9
  4.1. Timeline | 9
    4.1.1. Senior Design I Winter 2003 | 9
    4.1.2. Senior Design II Summer 2004 | 10
    4.1.3. Senior Design III Winter 2005 | 10
  4.2. Budget | 11
5. Proof of Design | 11
  5.1. Main Menu Interface | 11
    5.1.1. GM Dealer Form | 12
    5.1.2. Adding a New Call Problem | 14
    5.1.3. Return to Dealer (RTD) | 18
    5.1.4. RTD Contact | 20
    5.1.5. Print Report Interface | 23
    5.1.6. Resned Report | 24
    5.1.7. Problem Category Report | 25
6. Testing | 25
7. Conclusion | 29
8. Recommendations | 30
9 References | 33
## List of Figures

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1. Organizational Scheme Flowchart</td>
<td>4</td>
</tr>
<tr>
<td>Figure 2. Microsoft Access Database Schema</td>
<td>5</td>
</tr>
<tr>
<td>Figure 3. VB Naming Conventions</td>
<td>7</td>
</tr>
<tr>
<td>Figure 4. Main Menu</td>
<td>12</td>
</tr>
<tr>
<td>Figure 5. GM Dealer’s Form</td>
<td>13</td>
</tr>
<tr>
<td>Figure 6. Dealer Contact Form</td>
<td>15</td>
</tr>
<tr>
<td>Figure 7. Choosing Problem Type</td>
<td>16</td>
</tr>
<tr>
<td>Figure 8. Missing Data</td>
<td>17</td>
</tr>
<tr>
<td>Figure 9. Completed Form</td>
<td>18</td>
</tr>
<tr>
<td>Figure 10. Querying the Dealer History</td>
<td>20</td>
</tr>
<tr>
<td>Figure 11. Return to Dealer (RTD) Form</td>
<td>20</td>
</tr>
<tr>
<td>Figure 12. RTD Contacts Form</td>
<td>21</td>
</tr>
<tr>
<td>Figure 13. Parts Adjustment Section</td>
<td>21</td>
</tr>
<tr>
<td>Figure 14. Missing Data from Resned Type</td>
<td>22</td>
</tr>
<tr>
<td>Figure 15. Querying the Dealer RTD History</td>
<td>22</td>
</tr>
<tr>
<td>Figure 16. Reports Menu Form</td>
<td>23</td>
</tr>
<tr>
<td>Figure 17. Resned Report</td>
<td>24</td>
</tr>
<tr>
<td>Figure 18. Problem Category Report</td>
<td>25</td>
</tr>
<tr>
<td>Figure 19. Table of Suggestions from Users</td>
<td>27</td>
</tr>
<tr>
<td>Figure 20. Original Menu Main</td>
<td>28</td>
</tr>
</tbody>
</table>
Abstract

Client Information System or CIS is designed to centralize the ability to log, track, and organize credit for parts returned and phone calls from dealership part managers with any array of problems. CIS uses Microsoft Access as a front-end interface and Microsoft Access database for a back-end with capabilities for database. The software program will be built using Microsoft rapid-application-development (RAD) software, Access/Visual Basic. The program is designed for use on Microsoft Windows 2000/XP operating systems. The CIS user interface will consist of Windows GUI with mouse point-and-click functionality.
Client Information System

1. Statement of the Problem

The current system at General Motors Service Parts Warehouse in Cincinnati consists of an informal collection of hard copy forms, spreadsheets, and notes that often vary among the five members of the customer service department. There is no uniform method of obtaining a clear, comprehensive picture of customer care. Phone calls from dealership part managers are written on note pads detailing the time, identification of dealer, problem type, and any information collected. Since there is a lack of uniformity in handling phone calls, poor communication between members of the five member customer service department is a problem. At any given time people in the department will be on vacation, sick, training, or assigned other jobs. Parts managers may need something handled within a specific amount of time. Response time is a concern.

There are several needs that will be addressed in order to solve the problem. There is a need for the ability to log or input credit for parts sent to dealership parts managers. Also there is a need for the ability to track damaged material parts, returned claims, and query reports. In addition there is a need for the ability to input detailed information from part manager phone calls in a uniform manner so that all members of the five member customer service department have access to the information. Finally a need for the ability to create reports including customer performance complaints, wrong part claims, and damage freight claims exists.
2. Solution

Client Information System or CIS is a software application specifically designed for the customer service department in GMSPO. CIS uses Microsoft Access as a front-end interface and Microsoft Access database for a back-end with capabilities for database. The software program will be built using Microsoft rapid-application-development (RAD) software, Access/Visual Basic. The program is designed for use on Microsoft Windows 2000/XP operating systems. The CIS user interface will consist of Windows GUI with mouse point-and-click functionality. CIS will be versatile by including the following capabilities:

- The VBA programming language is used to power the GUI interface and create the GUI as a front-end.
- The use of Microsoft Access linking allows the ability of separating the database into a front-end and back-end.
- General Motors has a license agreement with Microsoft Office.
- Multiple tables for the database allows creating, inserting, or subtracting data for manageable use, such as tables for user groups, dealer information, inventory of part numbers, and parts prices.
- Security settings for users can be programmed in VBA for permission to certain users and creation of groups for software use.
2.1 User Profile

The intended users for CIS are the members of the customer service department of General Motors Service Parts Operation or GMSPO in West Chester, Ohio, and future customer service departments within GMSPO. The customer service department handles about 65 phone calls a day from car dealership part managers. The West Chester GMSPO plant is responsible for servicing a total of 700 car dealership part managers in the Midwest. The user must be able to use a standard Microsoft Office Application, be able to navigate the GUI and enter relating data pertaining to job. The user also must have knowledge and agreement of commonly used practices and guidelines in the customer service department.

2.2 Design Protocols

The flow of CIS consists of one main menu with five choices. Those choices include contacts which allow the customer service user to enter information of businesses related but not part of GM. Examples of contacts are contractors or services used by GM. Another choice is called dealers, which enables users to find dealers by dealer code, route, or dealership name. Return to Dealer or RTD allows users to enter data for additional credit or parts returned. Reports allow users to print reports for additional credit, parts returned, or problem types. The final choice is the exit option to quit the program. (See Figure 1.) for the organizational schema for an example of the main menu screen shot.
2.3 Database Design

The Microsoft Access database scheme will consist of eight tables related to the GM Customer Service Department schema. (See Figure 2,) displays the relational flow of data related to each of the tables in the database.

Examples of these requirements are:

- A customer service user enters data relating to job from communication with dealership.
- Parts Returned to Dealer or RTD is specifically logged for any parts returned or credits issued.
• Problem types identified between communication with dealership and customer service user is entered and logged as a call.

Figure 2. Microsoft Access Database Schema

2.4 Programming Coding Standards

The CIS Project is written to a set of coding standards which makes the project more readable and maintainable.
2.4.1 Programming VB Coding Standards

1. Complete detailed comments throughout the project within the programming environment.

2. Use compound names for objects, variables, functions, and sub-routines, when applicable. Use of proper case (i.e. capitalize the first letter of each word). For example txtDealerCode.

3. All variable names must be meaningful/descriptive.

4. Do not use global variables.

5. Use indentation to show scope. Indent 3 or 4 spaces. Indent inside functions, sub-routines, loops, if and case statements.

6. Be consistent.

7. Design and implement Error Handling Routines.
<table>
<thead>
<tr>
<th>Example</th>
<th>Prefix</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>chk</td>
<td>chk</td>
<td>Check Box</td>
</tr>
<tr>
<td>opt</td>
<td>opt</td>
<td>Option Box</td>
</tr>
<tr>
<td>cmd</td>
<td>cmd</td>
<td>Command</td>
</tr>
<tr>
<td>lst</td>
<td>lst</td>
<td>List Box</td>
</tr>
<tr>
<td>txt</td>
<td>txt</td>
<td>Text</td>
</tr>
<tr>
<td>lbl</td>
<td>lbl</td>
<td>Label</td>
</tr>
<tr>
<td>cls</td>
<td>cls</td>
<td>Class Modules</td>
</tr>
<tr>
<td>mod</td>
<td>mod</td>
<td>Modules</td>
</tr>
<tr>
<td>frm</td>
<td>frm</td>
<td>Forms</td>
</tr>
<tr>
<td>slw</td>
<td>slw</td>
<td>String</td>
</tr>
<tr>
<td>dlf</td>
<td>dlf</td>
<td>Double Float</td>
</tr>
<tr>
<td>sgl</td>
<td>sgl</td>
<td>Single Float</td>
</tr>
<tr>
<td>shn</td>
<td>shn</td>
<td>Short</td>
</tr>
<tr>
<td>hgf</td>
<td>hgf</td>
<td>Long</td>
</tr>
<tr>
<td>int</td>
<td>int</td>
<td>Integer</td>
</tr>
<tr>
<td>chr</td>
<td>chr</td>
<td>Character</td>
</tr>
<tr>
<td>bshld</td>
<td>bshld</td>
<td>Boolean/Bit</td>
</tr>
<tr>
<td>simd</td>
<td>simd</td>
<td>Array</td>
</tr>
</tbody>
</table>
Create and edit data

5. Users will be able to complete the following tasks:

4. A login that is authenticated by the database.

3. The Client Information System (CIS) uses ODBC to communicate between the

2. A user interface written in VBA which allows case of navigation.

1. A Microsoft Access based software that manages and tracks customer calls alone

A Software Program Tracking System

Database Management and Programming

3. Deliberables

INNER JOIN publishers ON publishers.pubID = publica.pubID

FROM publishers

SELECT publishers.*, publica.Logo, publica.Phrase

Examples:

5. Use indentation to show scope.

4. Be consistent.

3. All columns must not contain NULL.

2. All variable, table, view, and constraint names must be meaningful/descriptive.

1. Capitalize reserved words in any SQL command or query.

2.4.2 Programming SQL Coding Standards


Weeks 6-10: (February 10, 2003 to June 2, 2004)

Problem/area of inquiry:

Progress Report:

Chart Information:

Weeks 1-5: (January 3, 2003 to February 7, 2003)

4.1.1 Senior Design I Winter 2003

Detailed estimated timetable schedule for project from proposal, design.

4.1 Timeline

4. Design and Development

Search for existing problems by dealing with problem type.

Claims, and damage/identity claims.

Create reports including customer performance complaints, wrong part.

Create, edit, or add RTD credit for parts sent to dealership part managers.

Reports:

Manage and track damaged material parts, returned claims, and query.
4.1.2 Senior Design II Summer 2004

Weeks 1-5 (June 28, 2004 to August 2, 2004)
- Design Organizational Scheme Flowchart
- Design User Interface
- Draft of Project Description and Intended Use and User Profile

Weeks 6-10 (August 9, 2004 to August 23, 2004)
- Design Database
- Program Code to Microsoft Access Database for Presentation
- Documentation
- Demonstration of Prototype submitted and presented on August 23, 2004

4.1.3 Senior Design III Winter 2005

Weeks 1-5 (January 3, 2005 to January 31, 2005)
- Complete Project Functionality
- Complete Testing and Troubleshooting

Weeks 6-10 (February 7, 2005 to February 28, 2005)
- Submit Preliminary Draft of Final Report/ Documentation
- Submit Abstract of Project
- Submit Final Documentation and Demonstrate Project on February 28, 2005
4.2 Budget

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development Budget</td>
<td></td>
</tr>
<tr>
<td>Programming fee $35 per hour for 310 hours</td>
<td>$10,850.00</td>
</tr>
<tr>
<td>Developmental Total</td>
<td>$10,850.00</td>
</tr>
<tr>
<td>Production Budget</td>
<td></td>
</tr>
<tr>
<td>Microsoft Office 2000 Bulk Licensing per 250 computers</td>
<td>Free</td>
</tr>
<tr>
<td>Production Total</td>
<td>Free</td>
</tr>
<tr>
<td>Developmental Total</td>
<td>$10,850.00</td>
</tr>
<tr>
<td>Project Total</td>
<td>$10,850.00</td>
</tr>
</tbody>
</table>

5. Proof of Design

The next section demonstrates in detail how the deliverables of the project were fulfilled and what challenges were encountered.

5.1 Main Menu Interface

The customer service user role is to view, enter, and print reports with CIS. The main menu becomes the main point of entry to the specific uses related to knowledge of service department. The first screen shot organizes the categories, which relates to the Customer Service Department (See Figure 4.). The first button takes you to the GM Dealers form, the second button takes you to the RTD form, the third button takes you to the reports form, and the final button lets you exit the program.
5.1.1 GM Dealers Form

The customer service personal clicks the button for Dealers from the Main Menu. The GM Dealers form is display with information pertaining to the Dealerships, along with the history of problems entered by the customer service person to that dealer (See Figure 5.). There are three text boxes categorized under Lookup by, which allows the person to find a dealer by there unique dealer code. The remaining two boxes lets the user search the dealer by there route or company name. These text boxes are the main focal point because they allow the user to query the dealer by there unique dealer code, assigned route number, and the name of the dealership. When the customer service user
needs to find the dealer in order to view the history or add a phone call problem he/she uses the query to find that particular dealer.

The upper right hand corners of the form are the buttons that the customer service person will use to enter information. The Add Call button allows the user to enter future problems discussed for tracking. The Add Dealer button allows the user to add a new dealer which the GMSPO will be the servicing administrator. The exit button allows the user to exit the GM Dealers form and return the user to the Main Menu form.

![Figure 5. GM Dealer’s Form](image-url)
5.1.2 Adding a New Call Problem

The customer service personal clicks the button for Add Call from GM Dealer form. The Dealer Contact form is displayed with text boxes for information pertaining to the Dealers problems (See Figure 6.). The dealer code is automatically inherited from the GM Dealer form the person is viewing. A date and time stamped is automatically filled along with caller name. The upper right hand corner of the form has the buttons that the customer service person will use to update the information entered, a report button to view reports, and a cancel button to exit the form.

The user will ask a series of questions from the dealer. The user will ask the dealer what ship by carrier did the problem occurred. There choices are none, Schneider, CSO-3, Fedex, UPS, Bax, or other. Theses choices are identified by the unique Ship ID.
The Caller Name, Shipment No., and Problem text boxes are the next series of boxes to be entered by the user (See Figure 7.). The Problem text box has choices which are PC-31 claim, Financial Adjustment, Transportation Issue, Ordering Issue, Lost/Missing Item, Freight, or None. These choices are identified by the unique Problem ID. The Problem text box is important because it allows the customer service department to track the type of problem a dealer is calling about. The Problem text box must be chosen in order to save the information entered, otherwise a dialog box appears telling the user they are missing a category and they must choose a category or click the cancel and close button (See Figure 8.).
The final series of text boxes are the Work Assgn and Comments. The Work Assgn text box is a unique number that identifies the person who picked the parts originally for the dealer. This information is found on the dealer paperwork when the order is received. The Comments text box allows the user to explain the problem to the dealer (See Figure 9.). The user has three buttons in the upper right-hand corner. They are the
Add Update, Report, or Cancel and Close buttons. When the user has entered all the information they click on the Add Update button to save the information for that dealer. If the user chooses not to save the information, he/she would choose Cancel and Close. The information will not be saved, and he/she exits out of the form back to the GM Dealers form.

![Completed Form](image)

**Figure 9. Completed Form**

The GM Dealers form has a Call History section to view all the problems a dealer has had in the past from the phone calls logged from the customer service user. This information can be useful for identifying any reoccurring problems or complaints (See Figure 10.).
5.1.3 Return to Dealer (RTD)

The customer service personal clicks the button for RTD from the Main Menu. The Return to Dealer (RTD) form is displayed. The Return to Dealer (RTD) form displays information pertaining to the Dealer’s history of credit issued for parts or parts returned back (See Figure 11.). The form has two subforms within the Return to Dealer form. The first display is a history of parts returned back to dealer and credit issued. The second will consist of parts data related to each particular application field in the first subform.

There are three text boxes categorized under Lookup by, which allows the person to find a dealer by the unique dealer code. The remaining two boxes let the user search the dealer by the route or company name. These text boxes are the main focal point.
because they allow the user to query the dealer by the unique dealer code, assigned route number, and the name of the dealership. When the customer service user needs to find the dealer in order to view the history or add a phone call problem he/she uses the query to find that particular dealer.

In the upper right hand corner of the form are the buttons that the customer service person will use to enter information. The Add button allows the user to enter future parts credit or returned for tracking. The exit button allows the user to exit the Return to Dealer (RTD) form and return the user to the Main Menu form.

![Return to Dealer (RTD) Form](image_url)

Figure 11. Return to Dealer (RTD) Form
5.1.4 RTD Contact

The customer service user clicks the Add button from the Return to Dealer (RTD) form. The RTD Contact form displays text boxes with information pertaining to the Dealers problems. The dealer code is automatically inherited from the Return to Dealer (RTD) form the person is viewing. A date and time stamped is automatically filled along with caller name. The user will ask a series of questions from the dealer. The Caller Name, Application No., and Comments text boxes are to be entered by the user (See Figure 12.). The Parts Adjustment section allows the user to enter the part number information to be creditted or returned.

![RTD Contact Form](image)

Figure 12. RTD Contacts Form
The Parts Adjustment section allows the user to enter the part number information to be credited or returned. The user will ask the dealer what is the quantity number of parts and the credited quantity or returned quantity for the parts. The Resned Code text box has choices for that type of problem that a part will be issued credit or return. The choices are identified by the unique Resned ID. The Resned text box is important because it allows the customer service department to track the types of problems a dealer is having with a part (See Figure 13.). The Resned text box must be chosen in order to save the information entered, otherwise a dialog box appears telling the user they are missing a category and they must choose a category or click the cancel and close button (See Figure 15.).

Figure 13. Parts Adjustment Section
The Return to Dealer (RTD) form has a RTD History and Part History sections to view all credited or returned parts a dealer has had in the past. This information can be useful for identifying any reoccurring credits or returned parts a dealer is currently or has had in the past (See Figure 15.).
5.1.5 Print Report Interface

The customer service personal clicks the button for Reports from the Main Menu. The Reports form is displayed with a Select Report to Preview text, which is populated with reports to be viewed. Currently the Reports form consists of five choices for reports with related data from the GM Dealers and Return to Dealer (RTD) forms (See Figure 16.).

![Figure 16. Reports Menu Form](image)
5.1.6 Resned Report

The Resned Report is shown as a grouping of all the Resnd Type related to the information entered by the user for all credits or returned parts (See Figure 17.). This information can be useful for identifying how often a dealer is requesting to receive credit and returning parts. Any reoccurring credits or returned parts a dealer is currently or has had in the past can be interpreted as abuse of the process.

---

**Figure 17. Resned Report**
5.1.7 Problem Category Report

The Problem Category Report is shown as a grouping of all the Problem Types related to the information entered by the user from phone calls received by the customer service department (See Figure 18.). This information can be useful for identifying what types of problems are reoccurring for the dealer or has had in the past. This information can be passed on to the supervisors involved with the process for corrective action.

![Problem Category Report](image.png)

Figure 18. Problem Category Report

6. Testing

The testing phase occurred during a four month period in the GMSPO Customer Service Department from August 4, 2004 and completed on January 3, 2005. Since I am a part of the Customer Service Team, I monitored and tested the prototype. The scheduled
meetings with the Customer Service Department was planned by the plant manager and held every three weeks to collect feedback from the users in order to make the prototype more effective. The purpose of testing is to determine ways of making an application expose flaws in either the design or execution of the development process, along with making changes to relevance to the GMSPO business.

The overall testing breakdown consisted of: identifying problems such as syntax errors, logic errors, and miscellaneous user errors. The miscellaneous user errors can be defined as wrong characters entered by the user in textboxes for numerical value, empty textboxes where the user did not enter any data, and any error that is not identified in the early stages of development, such as any network problems. These problems account for fifty percent of the testing; the other fifty percent was the relevance of data to be used by GMSPO with CIS and what fields would be used.

The three related components of the usability in the testing process I focused on: 1) the recording feedback process, 2) the software observation process and 3) management of the data. Samplings of the feedback from the users in the Customer Service Department, along with consideration of the need are listed in a table (See Figure 19.). Many of the ideas were put in a category of feasibility and importance.
<table>
<thead>
<tr>
<th>Suggestion</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.) Whenever I click on CIS it takes a little longer to load. Is there a way to speed up this process?</td>
<td>Since CIS is on a server and uses Microsoft Access for a back end. This will continue to be a problem as the database grows. This is a performance issue and will have to be addressed in the future as CIS continues to grow.</td>
</tr>
<tr>
<td>2.) Minimize the amount of text boxes to enter information. I need to only enter information that is important. I do not want to spend all day entering information because of my other work and phone calls to answer.</td>
<td>I can take into account the user’s time for entering information. It makes sense to enter information that is currently not accessed by other applications. I can keep CIS minimized to a few text boxes for certain types of information.</td>
</tr>
<tr>
<td>3.) It does not make sense to enter a part number, then have CIS store the price of the part, especially when I fill out the information in Dealer Contact form.</td>
<td>It is important for the price of the part, when entering the part number for the Return to Part (RTD) form. The Material Return department needs to know the amount is being effective. I agree the Dealer Contact form does not need that information at the present time.</td>
</tr>
<tr>
<td>4.) The user ID does not need to be recorded because sometimes others who rarely use the program may want access to CIS. There are a few members in the Customer Service department and what is really important is the information entered. This moment in time security is not an issue.</td>
<td>People may want to find out who logged a problem with a dealer. People may have a question or like something elaborate on information entered by that person. True there are not many people in the department, but as the amount of users grow using CIS it would be smart, in case there is confusion. I agree security is not an issue at the present time.</td>
</tr>
<tr>
<td>5.) Can we make CIS do more or can we add more functionality than its intended use?</td>
<td>CIS can grow and expand as the Customer Service Department deems necessary or as the responsibility increases. I can add in more features then the original concept. I have allowed for comments in the code for future alterations by a programmer.</td>
</tr>
</tbody>
</table>

Figure 19. Table of Suggestions from Users
The program was tested everyday during the testing phase by seven individuals with two uses; entering problems from dealer phone calls, adding and tracking RTD, and supplying reports of both the problems from the dealers and RTD credits.

There were issues during the testing phase about basic software performance such as logging in for the user. Initially I created a log screen which prompts for a username and password. Later during one of the meetings it was decided to eliminate the prompt, but let the user software identification to match a table with all the names. The name would be displayed in the Main Menu upper right-hand side (See Figure 20. and Figure 21.). This was to show who the user was and be able to track the person who added or edited any information in the database. GMSPO decided it was not necessary because the person entering the information was not as important as the data itself.

Figure 20. Original Menu Main
Figure 21. Current Menu Main

There are many examples of certain types of information to be entered by the user. One of these examples is the GM Dealers form which has information about a particular dealer including a history of problems that have been entered in the past (See Figure 22.). The plant manager decided that the form had too much dealer information
and some was not relevant for viewing. The decision was made a month into the testing to minimize the amount of information about a dealer and minimize the amount of text boxes on the form or clutter for easy viewing for the user, keeping it simple (See Figure 23.). I made some adjustments including discarding the call details button because the user can view the information currently on the form. This is one example of the aesthetics on a software form where the customer made a decision over the programmer’s conception.

Figure 22. Original GM Dealers

Figure 23. Current GM Dealers

7. Conclusions

The project was created in response to General Motors Service Parts Operation or GMSPO lack of tracking of phone calls, tracking of RTD, and provides reports for supervisors. I have worked for GMSPO in a contract position for five years as a Quality Analyst prior to joining the Customer Service Department last year. GMSPO wanted the software application to be built in Microsoft Access so that employees in the GMSPO North American division will be able to access the program at their locations because of
the familiarity with Microsoft Access. The project fulfilled all Design Freeze
deliverables. Testing was performed to ensure the product’s usability.

8. Recommendations

While working on this project, I encountered quite a few challenges. A solid
understanding of warehousing inventory is recommended along with understanding the
need of the customer, Customer Service Department at the GMSPO. I spent a month
understanding Microsoft Access and VB Programming. I needed to know what I was able
to perform and deliver before accepting the project. I had to present in a meeting the
issues involved and deliverables for the project.

Once CIS was implemented to the specification of General Motors Service Parts
Operation Customer Service Department, I determined there was a better use for CIS as a
multi-user database over Microsoft Access.

The main concern I had with using Microsoft Access as a backend database is
described in the table (See Figure 24.). I assessed a comparison between Microsoft
Access as a stand alone database versus a multi-user database for an understanding the
positive and negative uses of Microsoft Access. The issues of security, user base,
portability, and relocation were the most important to research for the effective use of
CIS as it expands and grows.
<table>
<thead>
<tr>
<th>Security</th>
<th>Microsoft Access</th>
<th>Multi-user Database</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The database can usually be copied locally, which can allow a relatively in-experienced hacker to find out any passwords you’ve protected it with.</td>
<td>The database is never visible to a database user in its entirety. While these are rarely hacked, if they are it tends to be a denial-of-service attack and the data remains secure.</td>
</tr>
<tr>
<td>User Base</td>
<td>Access claims to be a multi-user database and can be set up to work as such, but the way that it works exposes the problems in using it for multi-user work. In Access, processing is done on the client machine which means that that client machine has to take the networked Access database, lock part of it, copy that part of it over the network, work on it, put it back and unlock it again. This relies on the network not to leave the database half written to. As a result of this, the possibility that the data will get corrupted increases.</td>
<td>These databases have an engine on the server on which they are sitting. They work on a Transaction Processing system, where a transaction is defined as a complete unit of work. Unless a transaction can be fully applied it waits for a few minutes for a response from the client before throwing the transaction away. There is no need to recover from this – a typical transaction would be an employee’s address – and in any case that user knows they’ve crashed and can go and check that record when their network connection is returned.</td>
</tr>
<tr>
<td>Portability</td>
<td>Most Access databases that get large enough to justify conversion to a fully scalable database platform have to be re-written. This process is much more difficult than creating them on a scalable platform from the outset. Conversion of the data is an effort, but not impossible. The code and forms however will need re-writing.</td>
<td>While a server based database can be accessed with platform specific tools, applications written against them, especially on the web are on the whole non-platform-specific. Examples of these tools are Perl, PHP, C, and Java. Not only are all these tools available on a range of hardware platforms, they all have plug-ins for many relational databases.</td>
</tr>
<tr>
<td>Relocation</td>
<td>If the server hosting a multi-user Access application changes, all the client machines will need re-pointing to it.</td>
<td>We can set up all users to point to an entry in the Primary Name Server for the university, which means that clients should very rarely need additional reconfiguration.</td>
</tr>
</tbody>
</table>

Figure 24. Listing of Reason for MS Access versus Multi-user Database.
It is my recommendation not to use Microsoft Access if you want to share the database across the network between more than 2 or 3 users. Security, robustness, and scalability become an issue because storing a lot of data and needing good performance becomes a problem if the customer wants to depend on the application for important business processes.
9. References

1. Avaya, Inc. "Call Management System (CMS)". 


5. WonderDesk. "WonderDesk Standard". 