Janitorial Contract Bidding Application

By

Mathew Philip Barter

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

March 2002

Dedication

This project stands as a milestone in my life. It reflects all the time and effort I have put into becoming only the second member of my family to graduate from college. Recognizing that everything I do in life is rarely done alone; I dedicate this project to my family. My wife for being especially supportive of all my ideas and still loving me when things got difficult. My two kids for being patient while daddy worked on his laptop every night. My mom and dad for their encouragement and financial contributions. Last but not least, my brother for inspiring me to be a graduate like him.

Table of Contents

List of Figures

Abstract

In the janitorial industry, keeping business information organized can increase the speed of transactions and customer service; this will create more profit for the cleaners. Software programs for the janitorial industry are rare and those programs that are available now are mainly off-the-shelf accounting programs. Many of these software programs are also not user-friendly. The janitorial industry needs a software package that is flexible in its implementation and use. There needs to be a program that can enhance the user's ability to bid for an account/contract. To solve these problems, I have created a Software Program for Janitorial Contract Bidding that allows its users to easily navigate between forms guiding them towards the ability to provide clear and coherent bids for potential customers.

1. Introduction

1.1 Statement of the Problem

When potential customers contact a janitorial company and request a bid for their site, they are typically preparing to replace an existing janitorial service company or to start a new service. In each case, the janitorial company will want to respond quickly with a precise and coherent bid. Some problems that many janitorial companies are faced with are the difficult process of collecting relevant site information, determining the cost of the services that will be provided, and combining this information to form a complete bid (3, pp. 1-2).

As the industry grows, many companies believe that if they can deliver their bid quicker than the competition, through accelerated customer service in the bidding process, then they stand to gain an edge in obtaining the contract (5).

1.2 Janitorial Contract Bidding Application

In the janitorial industry, keeping business information organized can increase the speed of transactions and customer service; this will create more profit for the cleaners. Of the few software programs available for the janitorial industry, bidding and estimating programs are rare. (4, p. 7). The programs that are available now are mainly off-the-shelf accounting programs. Many of these software programs are not user-friendly.

The janitorial industry needs a software package that is flexible in its implementation and use. There needs to be a program that can enhance the user's ability to bid for an account/contract.

1.3 Need for Janitorial Contract Bidding Application

My research findings and experience suggest that the janitorial industry has great potential for growth in the future. Evidence shows that customers are searching for a software package that is capable of meeting their needs (2)(1). Evidence also shows that 90 percent of respondents from both the In-House and Contract Cleaning surveys report using a computer at home and/or at work (6, p. 24). This supports the idea that users will be seeking out software to aid them with their bidding process.

2. User Profile

As stated in the previous section, approximately 90 percent of respondents from two different surveys report that they use a computer at home and/or work. This led me to define the user group element of this project for a category of intermediate computer users. These users have previous computer usage knowledge. My software package is designed such that the user's IT literacy can range from a first time user to a highly literate IT professional.

In considering this I have chosen to use control tips (text) to aid the user while navigating through and interacting with the software. There is not any need for administrative programming support for this software because it will be marketed to consumers as a packaged software program.

3. Design Protocols

3.1 Organizational Scheme/Navigation

3.1.1 Front End

The front end of this project utilizes the Graphical User Interface controls (of Microsoft Access 2000) on several forms to allow the user to input and extract information to be stored and manipulated. The following sequence of diagrams, forms and screen shots demonstrate what the user utilizes in the front end.

When the program is started, the user is presented with a "Switchboard" form or Main Menu that they use to navigate to all of the other main forms (Figure 13). This form has command buttons on it to link to other key forms (Add Company, Edit Company, Add Service, Edit Service, All Ordered Services, Reports Menu, Rates) and a Close Form button. Figure 1 shows the flow of the application stemming from this form.

When the user selects the Add Company button, the add Company form opens (Figure 14). This allows the user to input relevant information about a company or client they want to add to the database.

From the Main Menu, the user can also edit an existing company's information, by selecting the Edit Company button. This opens the Company Selector form (Figure 15). This form displays all the company's that have been entered into the database.

Once the user has added a company and wants to add services to the company, they click on the Add Service button from the Main Menu. This opens the Add Service form (Figure 17). This form allows the user to select services that they would like to add to a company.

Note: "service" refers to the many names of the services forms (i.e. Bathroom, Office, etc.)

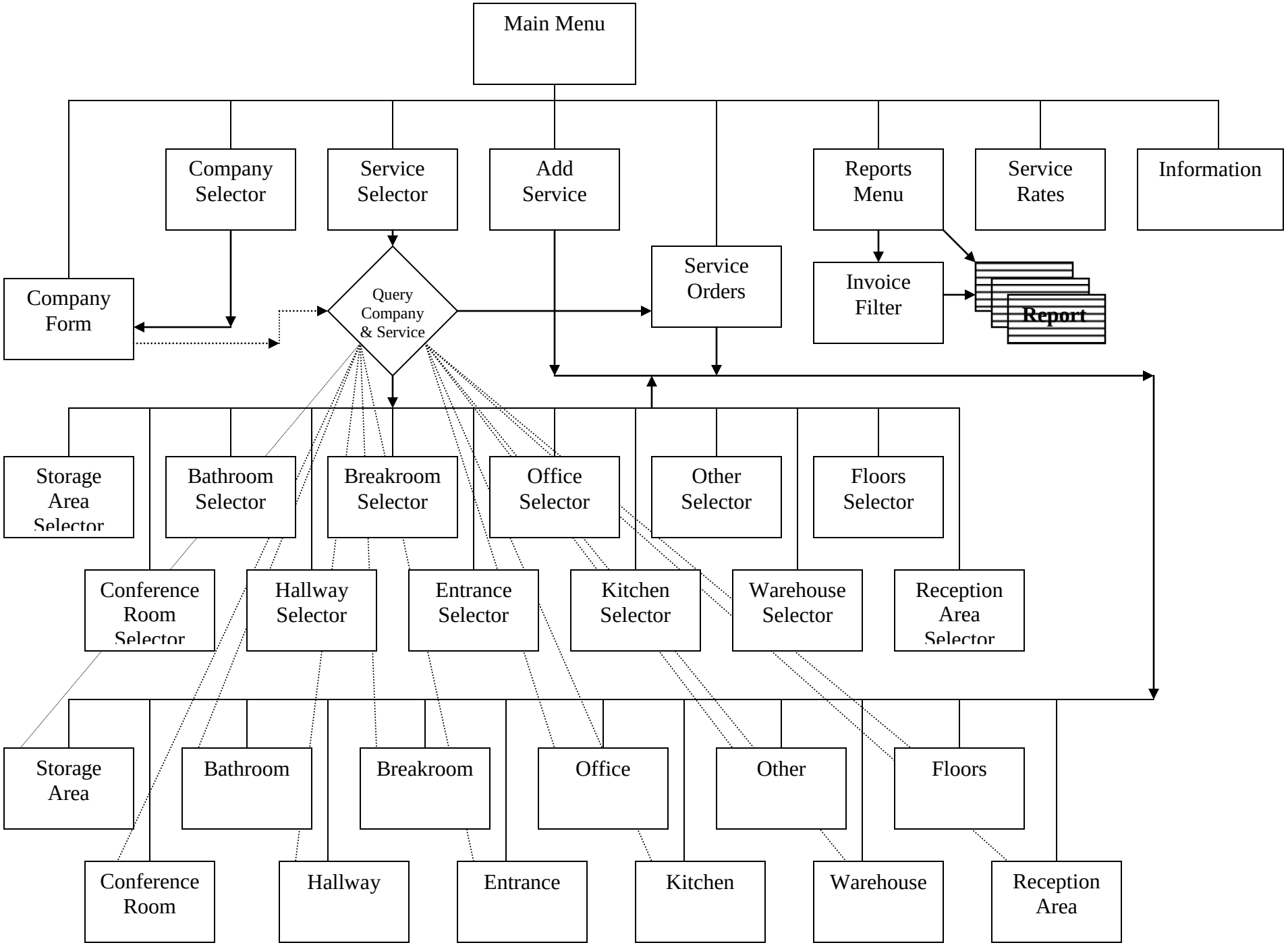

Figure 1. Project Diagram

Upon clicking on a service to add, the user is directed to the corresponding "service" form (Figures 27-38). These forms aid the user in inputting information about the service and calculating the time and total cost for the service.

Since there may come a time that the user would like to edit an existing service that they entered into the database, the user can click on the Edit Service button on the Main Menu to be directed to the Service Selector form (Figure 18). This form is similar to the Add Service form in display. When the user clicks on the service button, a query is executed to populate the corresponding "service" selector form list box.

The "service" selector forms (there are twelve in Figure 1) are used to sort records according to the service name and allow the user to select a service to edit and open up that service form in edit mode. Figure 1 shows how the information travels through the query and into the "service" selector forms and from the "service" selector forms to the "service" forms.

The query that is running in the "service" selector forms is generated by using information gathered by both the Company form and the "service" forms (denoted by the dotted line in Figure 1).

Another way that the user can select a service to edit is to use the Service Orders form (Figure 21). This form is launched by clicking the All Ordered Services button from the Main Menu. The Service Orders form is populated with a similar query used to populate the "service" selector forms. This form allows the user to view all the services ordered in the database. The user utilizes this form to select the specific service that they wish to edit from a list box. The user is then directed to the corresponding "service" form and information as shown in Figure 1.

The "services" forms at the bottom of Figure 1 are used for both inputting data as well as editing existing data. On these forms, there is a command button that can be used to display/access the Service Rates form (Figure 26). This form is used to record the rates that are to be charged for cleaning each of the fixtures. These rates are used to calculate the total cost on each of the forms. This form can also be accessed through the Main Menu by clicking on the Rates command button.

The Reports Menu form (Figure 23) is now called from the Main Menu to create invoices and reports for both the user and customer. It is from here, the user selects the reports that they want to run from a list box and then select to either preview the report or to go directly to printing the report. When the user selects Run Invoices, they are prompted to select a company to run an invoice for (Figure 24). They then run the invoice and a filter is applied to only show those pieces of that invoice.

If the user has any questions regarding the applications intended use or problems running the application, there is an Information form supplied. This form can be accessed from the Main Menu and displays contact and copyright information.

3.1.2 Back End

The database backend was built utilizing Microsoft Access. The databases are relatively simple with some one-to-many relationships. Normalization rules were followed in designing the databases so that information was not duplicated and unnecessary space used. Visual Basic for Applications (VBA) was used to help manage the calculations performed by the forms and databases.

During the decision making process, I originally decided to have a separate table for each of the services. There were twelve different tables to be used just for this process. I

decided, late into my project, to rebuild the tables. I condensed twelve tables into one table therefore reducing redundant information and freeing up space.

I also abandoned my original coding design of using DAO to access the records in the tables. I instead used VBA to access information directly off the forms (Appendix 1). The code shows how the calculations are performed in order to determine the total cost of the service. The code also demonstrates how the time constraint used was be "hard coded" into the code because the time it takes to clean a fixture is set whereas the rate would be variable (Refer to appendix 1).

3.2 Color Scheme

The color scheme for this software is gray, blue and white. I want to keep the color scheme simple and fresh in its appearance to go along with the theme of this software.

3.3 Help

There is a section dedicated to the application Information. This is a section that includes my name, address, phone and email address in the event that I need to be contacted for support. I prefer to give support over the phone rather than have users try to perform a root cause analysis on software that I design. There is also a legal disclaimer that reminds users that this software is proprietary to Mathew Barter.

4. Objectives of the project ("Deliverables)

The overall objective of this project is to develop low-cost, user-friendly software that enables users to build comprehensive and coherent bids.

I plan to use Microsoft Access 2000 to develop user-friendly front-end applications to garner information and back end databases to store the information

. I used Visual Basic for Applications to assist in the design of the front-end applications. The database enables the users (janitorial services) to manipulate the data they collect and to generate reports. The reports can be used to measure progress at individual sites for their customers as well as for the software user (janitorial service) regarding productivity and profitability for each account/contract.

The following were the project deliverables:

- Develop an Access database for this project with front-end entry forms allowing the user to easily add new accounts and services to the database.
- Provide reports for both the customer and the software user (janitorial service).
- Assemble clear and coherent bids/invoices for the customers/user's.
- User-centered design that is easy to use. Users can easily navigate through all the forms.
- Be cost effective.

5. Design and Development

5.1 Budget

Most of the items listed in the budget for this project were already supplied. This aided in keeping the software price to a minimum.

Item	Quantity	Estimated Cost	Actual Cost
PC	1	$999.99	$0.00
Monitor	1	79.99	0
Iomega Zip Disks	4	30	30
Floppy Disks	10	5	5
CD-RW CD's	25	25	25
Microsoft Visual Basic for Applications (Book)	1	49.99	49.99
Microsoft Access	1	459	0
Total Cost		$1,648.97	$109.99

Figure 2. Project Budget

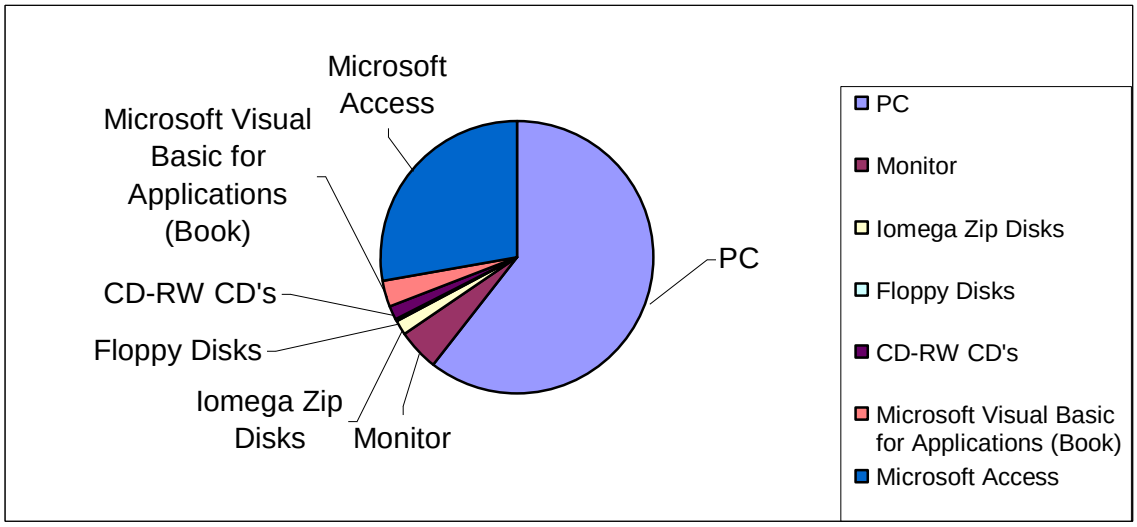

Figure 3. Estimated Costs

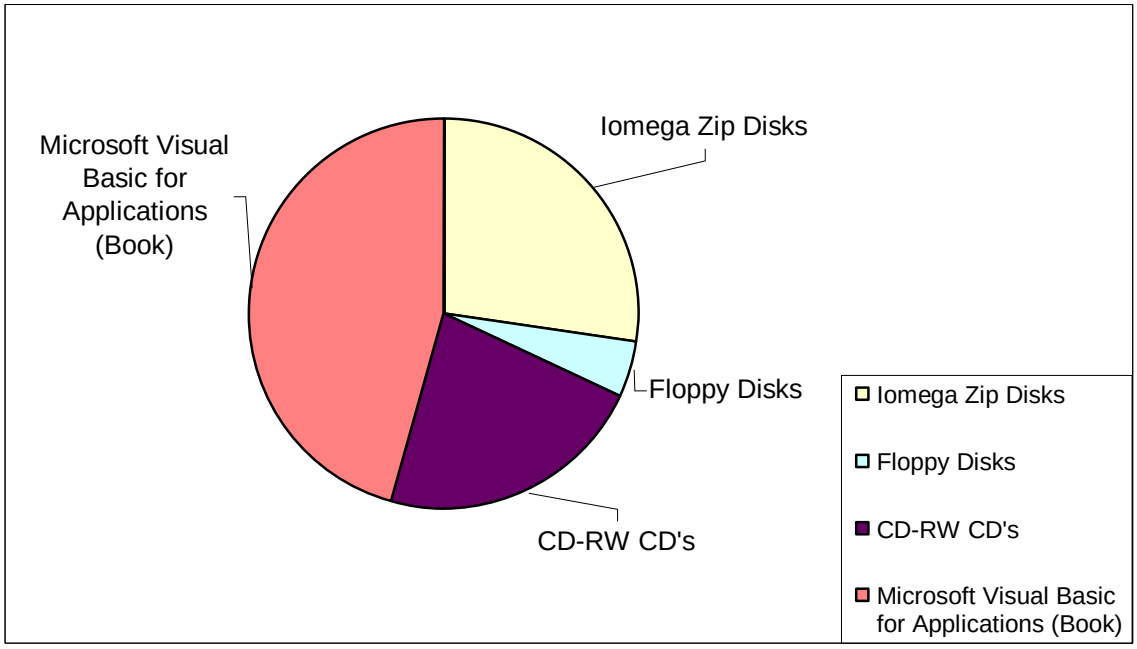

Figure 4. Actual Costs

5.2 Timelines

Task	Start Date	End Date	Duration/Days
Senior Design II	6/20/01	8/29/01	69
-Continue Learning VBA	6/20/01	8/29/01	69
-Develop Prototype	6/20/01	8/22/01	62
-DB (table) design	6/20/01	8/1/01	41
-Design GUI	6/20/01	8/15/01	55
-Develop Flow Charts	6/20/01	8/15/01	55
-Code Testing	8/10/01	8/25/01	15
-Progress Report I	7/18/01	7/18/01	1
-Progress Report II	8/15/01	8/15/01	1
-Reports	6/20/01	8/29/01	69
-Rough Draft	6/20/01	8/8/01	48
-Final Draft	8/8/01	8/29/01	21
-Presentation	7/20/01	8/29/01	39
-Check Prototype	7/20/01	8/15/01	25
-Slide Preparation	8/15/01	8/29/01	14
-Final Presentation	8/29/01	8/29/01	1
Senior Design III	8/30/01	3/13/02	193
-Continue learning VBA	8/30/01	12/15/01	105
-Design Working Product	8/30/01	2/15/02	165
-GUI Forms	8/30/01	12/15/01	105
-VBA Code	8/30/01	1/15/02	135
-Final Product Code Testing	2/15/02	3/3/02	18
-User Beta Testing	3/3/02	3/10/02	7
-Debug Final Code	3/3/02	3/15/02	12
-Compile Executables	3/15/02	3/15/02	1
-Develop User Documentation	3/3/02	3/15/02	12
-Reports	2/20/02	3/13/02	23
-Rough Draft	2/20/02	2/20/02	1
-Final Draft	3/13/02	3/13/02	1
-Progress Report I	2/20/02	2/20/02	1
-Progress Report II	3/1/02	3/1/02	1
-Presentation	3/1/02	3/13/02	12
-Slide Preparation	3/1/02	3/13/02	12
-Final Presentation/Demo	3/13/02	3/13/02	1

Figure 5. Senior Design II and III Timeline

Task	Start Date	End Date	Duration/Days
Senior Design II	**6/20/01**	**8/29/01**	**69**
-Continue Learning VBA	6/20/01	8/29/01	69
-Develop Prototype	6/20/01	8/22/01	62
-DB (table) design	6/20/01	8/1/01	41
-Design GUI	6/20/01	8/15/01	55
-Develop Flow Charts	6/20/01	8/15/01	55
-Code Testing	8/10/01	8/25/01	15
-Progress Report I	7/18/01	7/18/01	1
-Progress Report II	8/15/01	8/15/01	1
-Reports	6/20/01	8/29/01	69
-Rough Draft	6/20/01	8/8/01	48
-Final Draft	8/8/01	8/29/01	21
-Presentation	7/20/01	8/29/01	39
-Check Prototype	7/20/01	8/15/01	25
-Slide Preparation	8/15/01	8/29/01	14
-Final Presentation	8/29/01	8/29/01	1

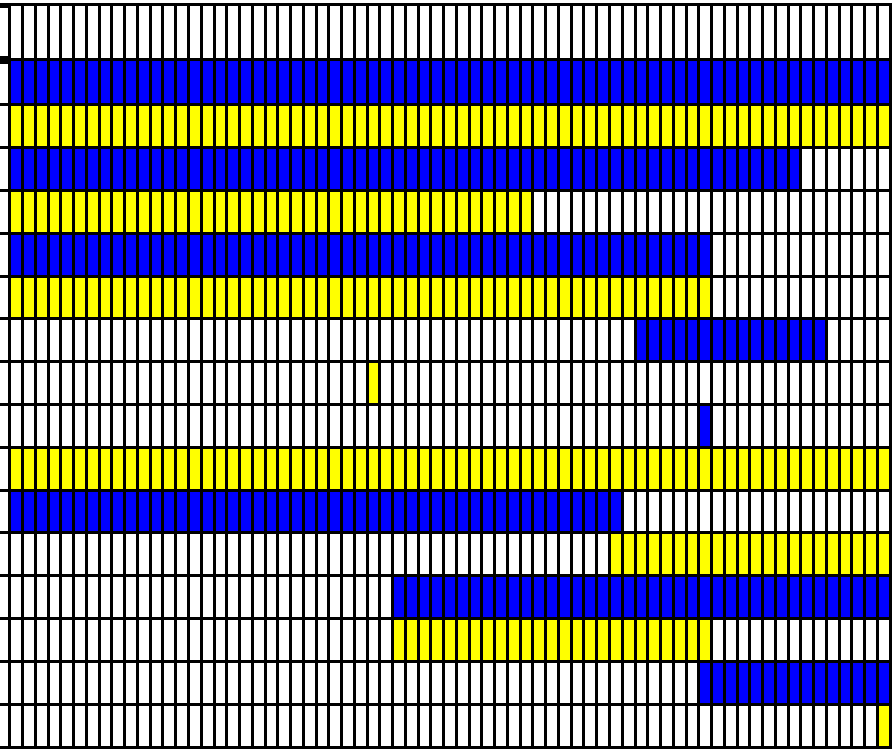

Figure 6. Senior Design II Timeline Graph

Task	Start Date	End Date	Duration/Days
Senior Design III	**8/30/01**	**3/17/02**	**197**
-Continue learning VBA	8/30/01	12/15/01	105
-Design Working Product	8/30/01	2/15/02	165
-GUI Forms	8/30/01	12/15/01	105
-VBA Code	8/30/01	1/15/02	135
-Final Product Code Testing	2/15/02	3/3/02	18
-User Beta Testing	3/3/02	3/10/02	7
-Debug Final Code	3/3/02	3/15/02	12
-Compile Executables	3/15/02	3/15/02	1
-Develop User Documentation	3/3/02	3/15/02	12
-Reports	2/20/02	3/17/02	27
-Rough Draft	2/20/02	2/20/02	1
-Final Draft	3/17/02	3/17/02	1
-Progess Report I	2/20/02	2/20/02	1
-Progress Report II	3/1/02	3/1/02	1
-Presentation	3/1/02	3/17/02	16
-Slide Preparation	3/1/02	3/17/02	16
-Final Presentation/Demo	3/17/02	3/17/02	1

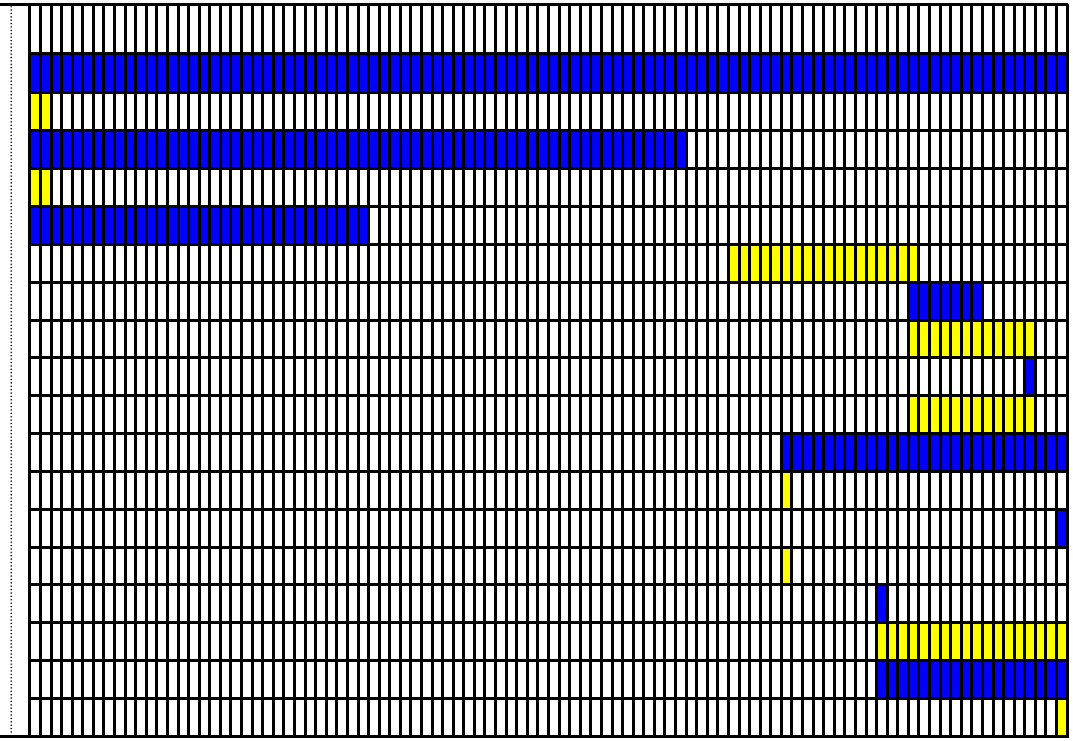

Figure 7. Senior Design III Timeline Graph

5.3 Hardware and Software

The following are the minimum requirements for the software and hardware that are required to complete this project.

- Intel® Pentium® III processors at 866MHz and 933MHz.
- 64MB up to 512MB SDRAM memory for overall system performance.
- Hard drive = 40GB for storage.
- CD-RW (hardware) for distribution.
- Large monitor - up to 19" (17.9" viewable).
- 25 CD-RW for software distribution.
- 4 Iomega Zip disks
- 10 Floppy Disks
- Microsoft Visual Basic for Application (Book)
- Microsoft Access

6. Proof of Design

The basic architecture of the project consists of two components. There is a Visual Basic for Applications (VBA) front end application with a Microsoft Access backend database.

There were three database tables created for this project. The tables are tblCompany, tblAllServices, and tblServiceRates. There is one relationship designated between the table's tblCompany and tblAllservices. It is a one-to-many relationship from the CompanyID in tblCompany to CompanyName in tblAllsServices. The following screen shot (Figure 8) shows the relations between the two tables (tblAllServices and tblCompany).

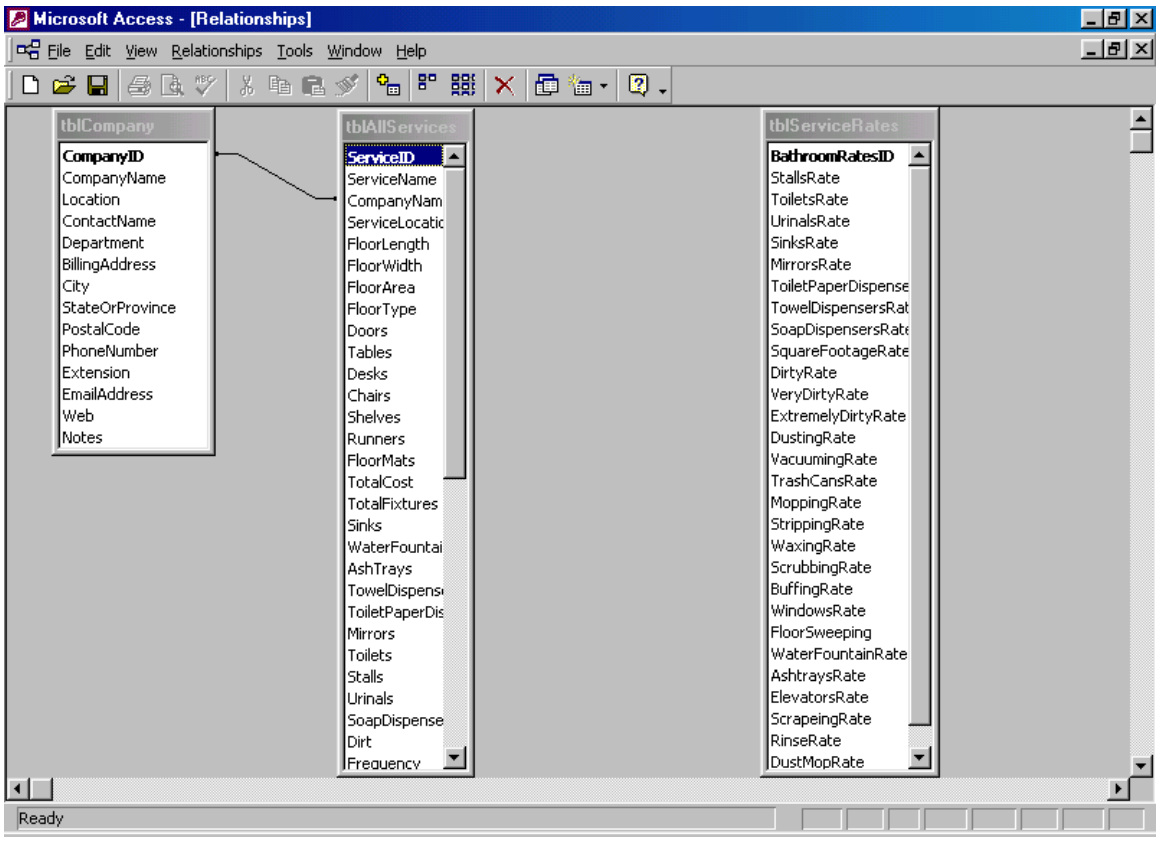

Figure 8. Table Relationships

The table tblAllServices (Figure 9) stores all the information that is put into the "Service" forms (Figures 26-37). The table has a primary key, which is the ServiceID. ServiceID also is a unique identifier with a data type of auto number. This table was redesigned to incorporate twelve other tables into this one. This made developing queries and reports easier to build. The size of the database also was trimmed considerably allowing for a quicker processing speed. I used a lookup wizard to develop choices for the Floor Type field. I used the auto form wizard to help develop initial forms to use for inputting information from this table as well.

Field Name	Data Type		Field Name	Data Type
ServiceID	AutoNumber		SoapDispensers	Number
ServiceName	Text		Dirt	Number
CompanyName	Number		Frequency	Number
ServiceLocation	Text		Windows	Number
FloorLength	Number		VendingMachines	Number
FloorWidth	Number		LunchLine	Number
FloorArea	Number		Refridgerators	Number
FloorType	Text		FileCabinets	Number
Doors	Number		Elevators	Number
Tables	Number		Counters	Number
Desks	Number		Racks	Number
Chairs	Number		Phones	Number
Shelves	Number		Lamps	Number
Runners	Number		Plants	Number
FloorMats	Number		Closets	Number
TotalCost	Currency		Lights	Number
TotalFixtures	Number		TrashCans	Number
Sinks	Number		TaxRate	Number
WaterFountains	Number		ScrubFloors	Yes/No
AshTrays	Number		StripFloors	Yes/No
TowelDispensers	Number		RinseFloors	Yes/No
ToiletPaperDispensers	Number		BuffFloors	Yes/No
Mirrors	Number		DustMopFloors	Yes/No
Toilets	Number		SweepFloors	Yes/No
Stalls	Number		ScrapeFloors	Yes/No
Urinals	Number		WaxFloors	Yes/No

Figure 9. Table tblAllServices

The table tblCompany (Figure 10) stores all the relevant company information from the Company form after the user enters the information. It has a unique identifier (autonumber) for a data type on the CompanyID. It is also the primary key for this table. The email address data type is set to hyperlink in order to open up the default mail client so email can be sent to the contact. The web data type is set to hyperlink to open up the default web browser and the web address placed in the address bar.

Field Name	Data Type
CompanyID	AutoNumber
CompanyName	Text
Location	Text
ContactName	Text
Department	Text
BillingAddress	Text
City	Text
StateOrProvince	Text
PostalCode	Text
PhoneNumber	Text
Extension	Text
Email Address	Hyperlink
Web	Hyperlink
Notes	Memo

Figure 10. Table tblCompany

The last of the tables is tblServiceRates. This table houses all the information that is inputted through the Service Rate form. The following shows the fieldnames and data type used. There is a primary key on the BathroomRatesID with a unique autonumber identifier.

BathroomRatesID	AutoNumber		BathroomRatesID	AutoNumber
StallsRate	Number		TrashCansRate	Number
ToiletsRate	Number		MoppingRate	Number
UrinalsRate	Number		StrippingRate	Number
SinksRate	Number		WaxingRate	Number
MirrorsRate	Number		ScrubbingRate	Number
ToiletPaperDispenserRate	Number		BuffingRate	Number
TowelDispenserRate	Number		WindowsRate	Number
SoapDispensersRate	Number		FloorSweeping	Number
SquareFootageRate	Number		WaterFountainRate	Number
DirtyRate	Number		AshTraysRate	Number
VeryDirtyRate	Number		ElevatorRate	Number
ExtremelyDirtyRate	Number		ScrapeingRate	Number
DustingRate	Number		RinseRate	Number
VacuumingRate	Number		DustMopRate	Number

Figure 11. Table tblServiceRates

When the user starts the application, they are presented with the form frmStart, captioned as Welcome (Figure 12). This form is used to welcome the user to the application they have purchased and inform the user that they are using an application developed by Applied Computer Concepts and Mathew Barter. The form contains a command button labeled Start. The command button executes a macro when it is clicked. The macro opens the form frmSwitchBoard (Figure 13) and closes the form frmStart (Figure 12). The form also contains a command button labeled Exit. The command button executes a macro when it is clicked. The macro closes the form frmStart (Figure 12) and ends the application execution.

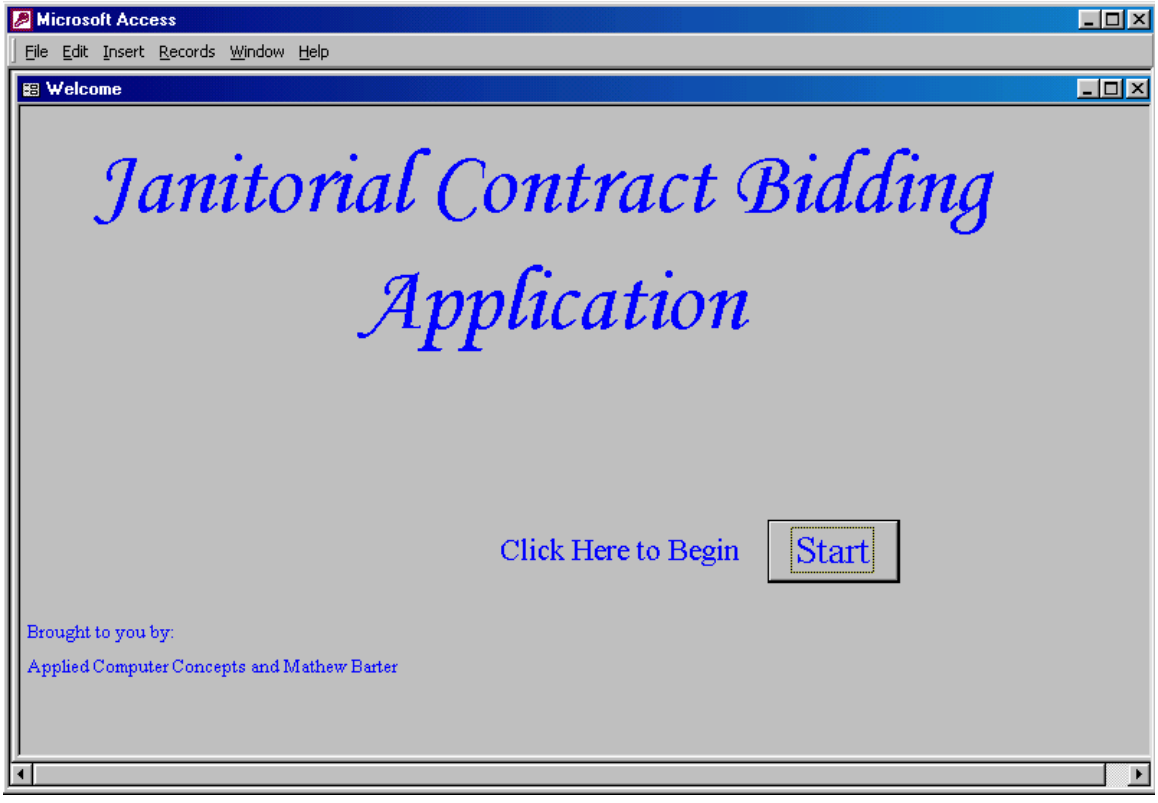

Figure 12. Form frmStart, Welcome

When the user clicks on the start button, they are presented with the frmSwitchBoard captioned as Main Menu (Figure 13). This menu allows the user to easily navigate through the application. The user has the option to either Add a company, Edit and existing company, Add a service, Edit an existing service, View/Edit all existing services ordered, Start the report menu, Open the service rates form or View the information about the application. All of these command buttons are driven by macros that are defined in the command buttons onclick properties. The macros are all set to open forms in view mode with the exception of Add Company which is set to open in add mode.

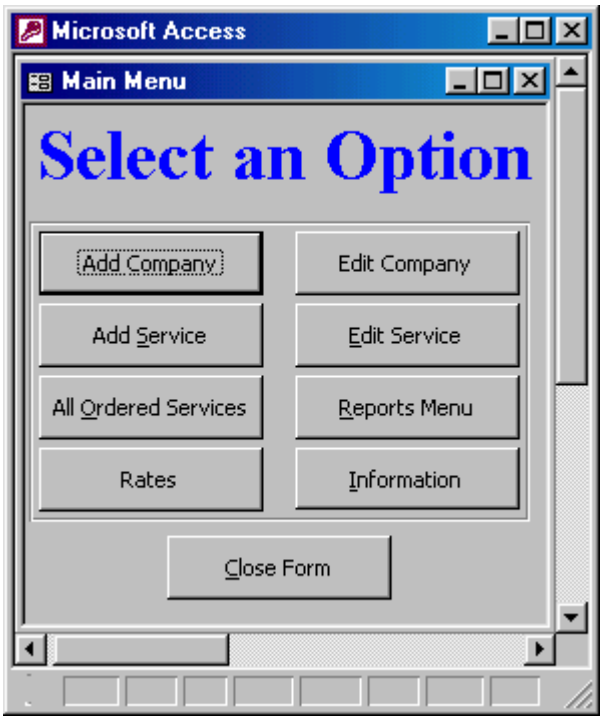

Figure 13. Form frmSwitchBoard, Main Menu

The first thing the user will want to do is enter a company into the database so that they can later attach services to that company. This is accomplished by using the form frmCompany (Figure 14) captioned as Company. The user then enters all relevant information that the form asks the user to supply using the following form.

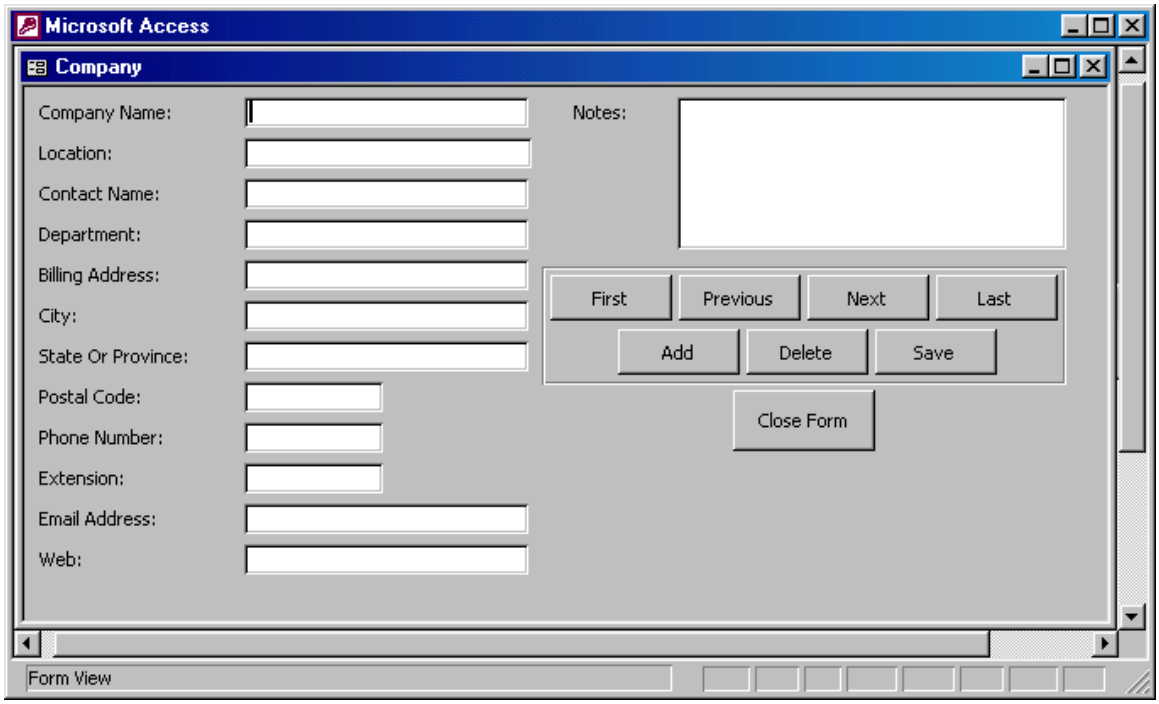

Figure 14. Form frmCompany, Company

After entering in a company to the table tblCompany the user is now able to edit that information easily by using the form frmEditCompany (Figure 15). This form enables the user to view all the current companies added into the database.

By selecting one company in the list box, using the mouse cursor, the user is transported to the corresponding company using the add company form from figure 14. The user has the ability to edit the information here as well as add additional companies to the database. They can also edit more that one company without having to go back to the Company Selector form because the navigation buttons are fully functional. The way the Company Selector is populated with its information is from a query of the tblCompany shown below in Figure 16.

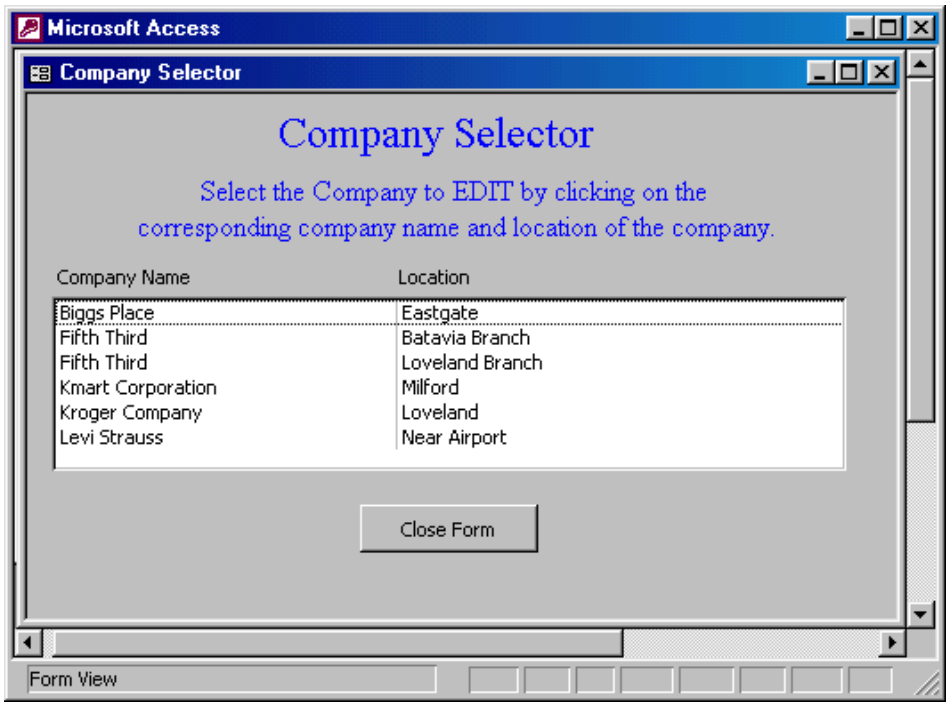

Figure 15. Form frmEditCompany, Company Selector

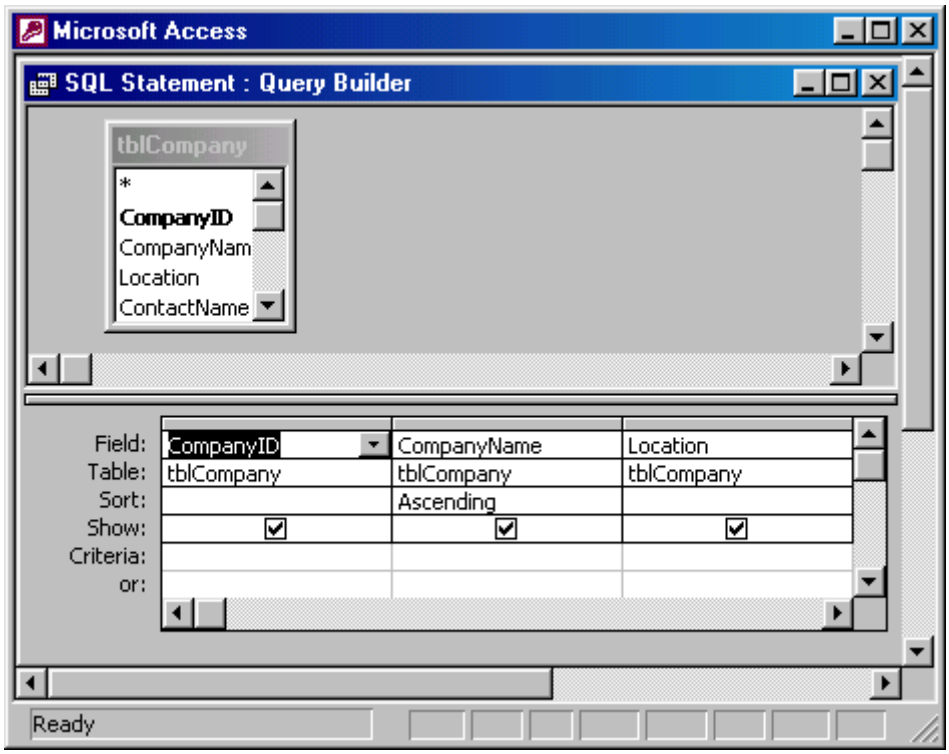

Figure 16. Company Selector Query

After the user has entered a company into the database and has certified that the information is correct, they can now begin to attach services to the company. This is accomplished by using the form frmSelectAddOn (Figure 17) captioned as Add Service. This form has a command button for each of the services that can be added. These command buttons are: Bathroom, Office, Conference Room, Hallway, Reception Area, Other, Breakroom, Storage Area, Warehouse, Kitchen, Entrance, Floors and Close. Depending on which button is clicked results in the corresponding "service" form (Figures 27-38) for that service to open and allow the user to enter information about the service. The command buttons open the "service" form by using a macro defined in the command button properties. The macro is set to open in add mode so all fields are be cleared when the "service" forms (Figure 27-38) open.

Once the user has entered a service into the database, the user may want to edit the information later. This is accomplished by using the form frmSelectServiceEdit (Figure 18). This form looks identical to the Select Add On and also has a command button for each of the services that can be added. These command buttons are: Bathroom, Office, Conference Room, Hallway, Reception Area, Other, Breakroom, Storage Area, Warehouse, Kitchen, Entrance, Floors and Close. These command buttons open the corresponding "service" selector by using a macro defined in the command button properties.

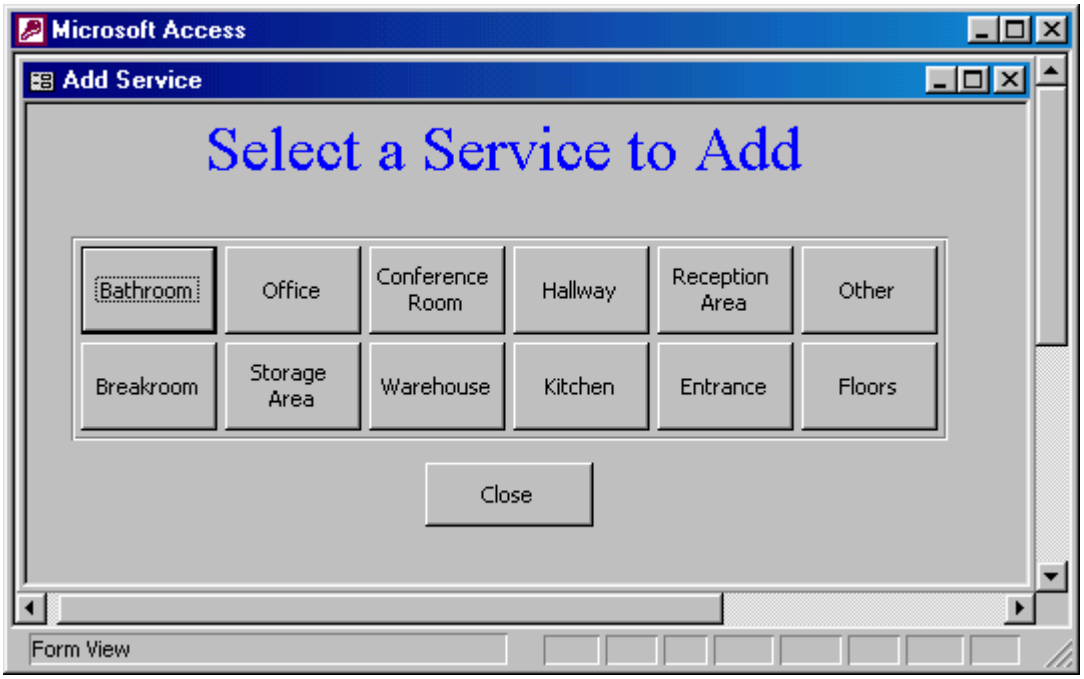

Figure 17. Form frmSelectAddOn, Add Service

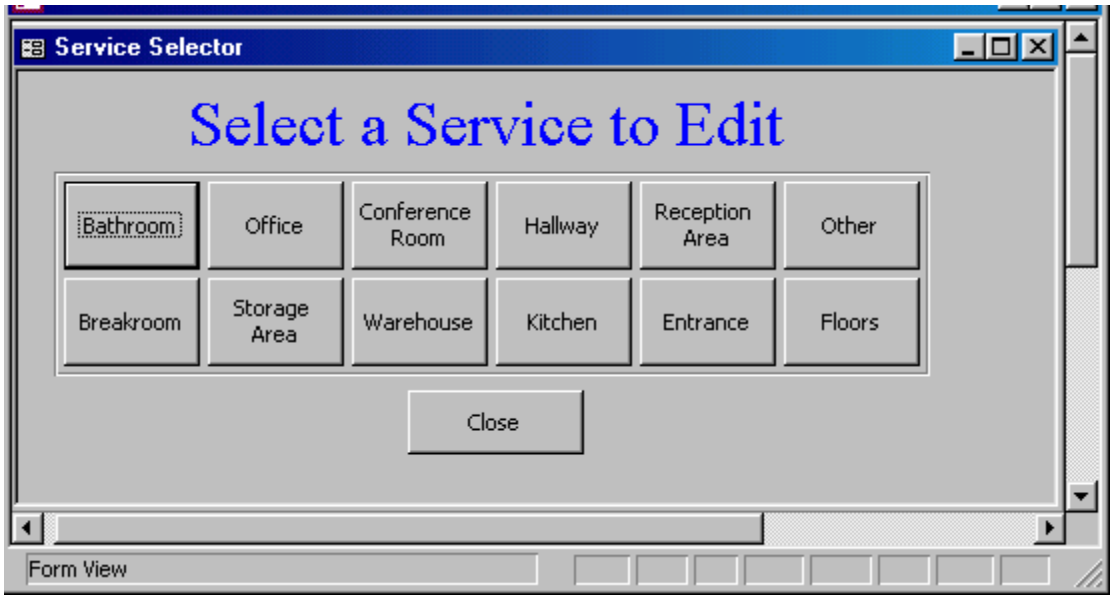

Figure 18. Form frmSelectServiceEdit, Service Selector

When the user clicks on a button of the service that they want to edit, it opens up the "service" Selector form (Figure 19). These forms are nearly identical to each other and there are twelve in total (one for each service). When these forms are opened, it filters all the records so as to only display those that match the command button caption that was clicked, in the list box. When the user makes a selection in the list box, it triggers the BeforeUpdate event of the list box. The code in this event triggers the corresponding "service" form to open then takes the ServiceID from the list box and matches it against the corresponding recordset in the table tblAllServices. That recordset is then displayed in the open form. The following (Figure 19) is an example of what these selector forms look like.

Figure 19. Form frmEditBathroom, Bathroom Selector

The query that is used to populate this list box in Figure 19 is shown below (Figure 20). This is a one to many relationship where the left table is the tblCompany table using the Company ID field and the right table is the tblAllServices using the Company Name field. This query incorporates the following fields from the tblAllServices table: Service ID, Company Name, Service Location and Service Name. The query also incorporates the following fields from the tblCompany table: Company ID and Location. The Service Name field has the criteria specified in accordance to the service selected so that the user only views those services that match the criteria. The query is also sorted ascending on the Company Name from the table tblCompany. Each query for the different "service" selector forms has their own criteria defined to only show the records that match the title of the "service" selector.

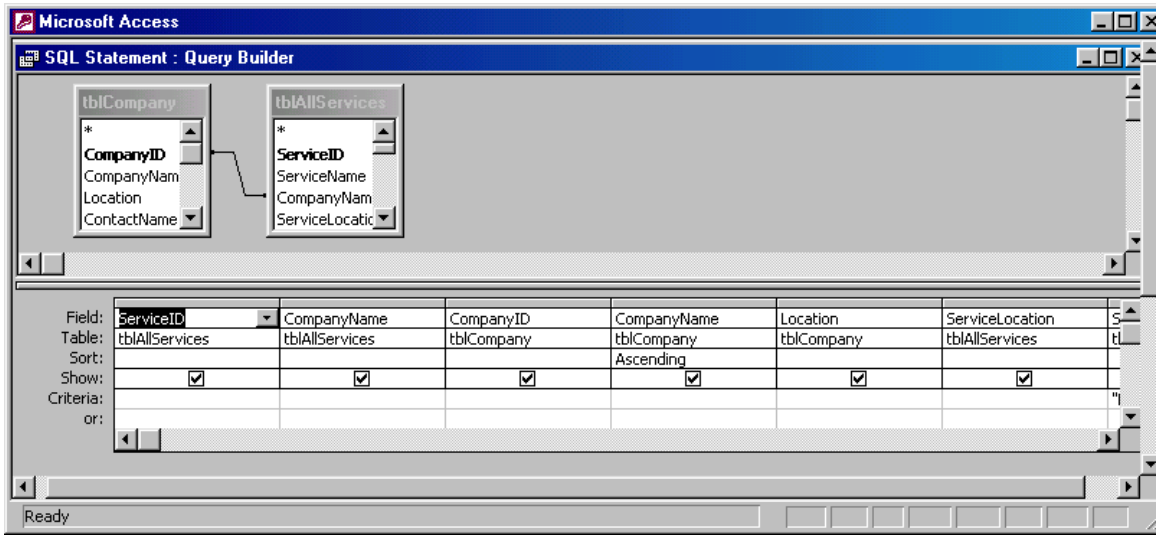

Figure 20. Bathroom Selector Query

If the user wants to instead sort through all the orders that have been placed rather than selecting a group of services, they can use the form frmServiceOrders (Figure 21) captioned as Service Orders. This form lists all of the orders sorted by Company Name and then Company Location. This helps with cutting the time it takes to locate a

company's service. When the user makes a selection in the list box, it triggers the BeforeUpdate event of the list box. The code in this event triggers the corresponding "service" form to open then takes the ServiceID from the list box and matches it against the corresponding recordset in the table tblAllServices. That recordset is then displayed in the open form.

Figure 21. Form frmService Orders, Service Orders

The query that populates the list box in Figure 21 is shown below (Figure 22). This query is very similar to the one in figure 20 however; there is one additional field that is being displayed (Service Name), it sorts the records by ascending on both Company Name and Location and the criteria for the query has been remove. The sorting on the Company Name and Location helps with being able to acquire a record quickly.

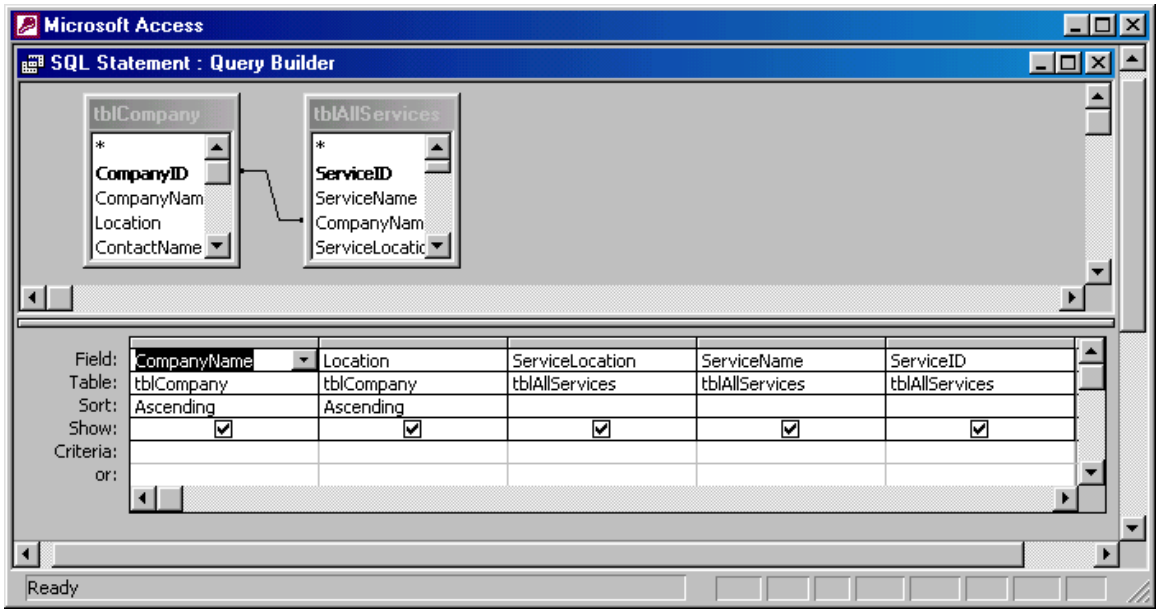

Figure 22. Service Orders Query

After the user has entered all of the information that pertains to an account, they may want to run reports to extract some meaningful information that they can use to their advantage. The reports menu (Figure 23) does just that. The reports menu lists all the current reports, that have been created, in a list box and can be either previewed or printed simply by selecting a report and choosing between the two available command buttons. There is an IF statement that is run when the command button Preview or Print is clicked that assesses the selection made in the list box to see if it is Run Invoices.

If the user selects Run Invoices from the list box then they have a form frmInvoiceFilter (Figure 24) appear so that they can choose which company they want to create an invoice for. The combination box contains all of the available company names. After a company name is selected, the user clicks on the button Run Invoice to create the invoice. The user can also select the clear button to clear an invoice once they have created one.

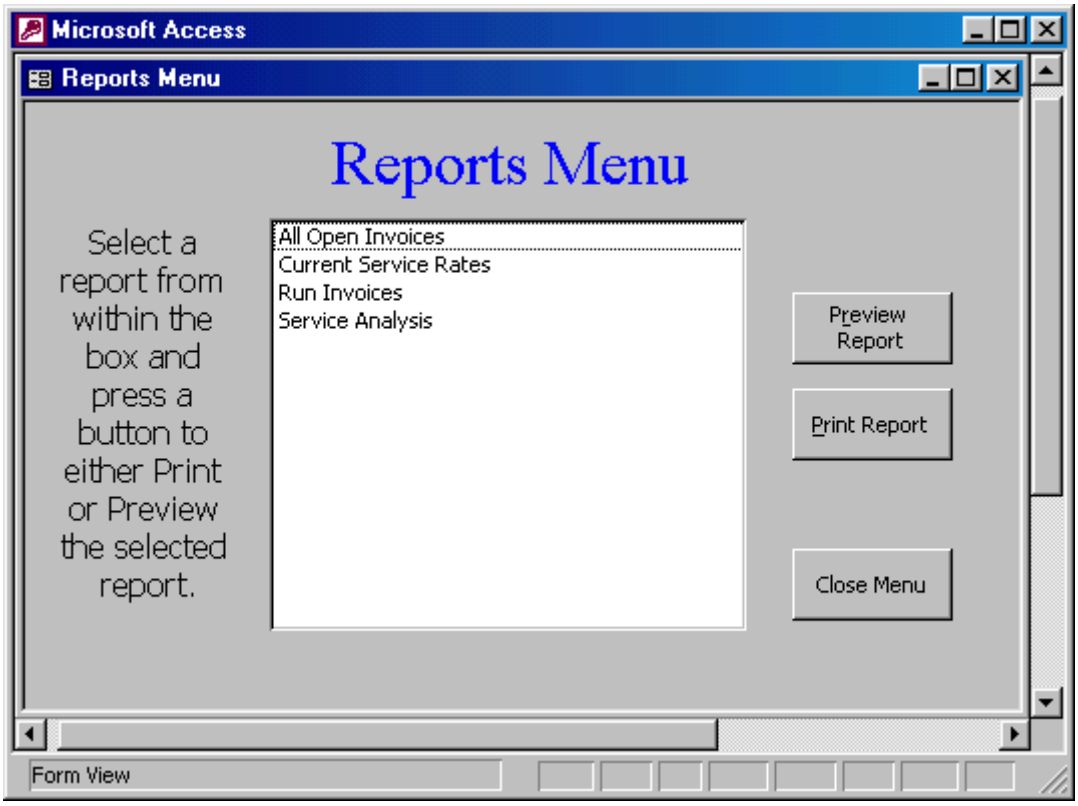

Figure 23. Form frmReports, Reports Menu

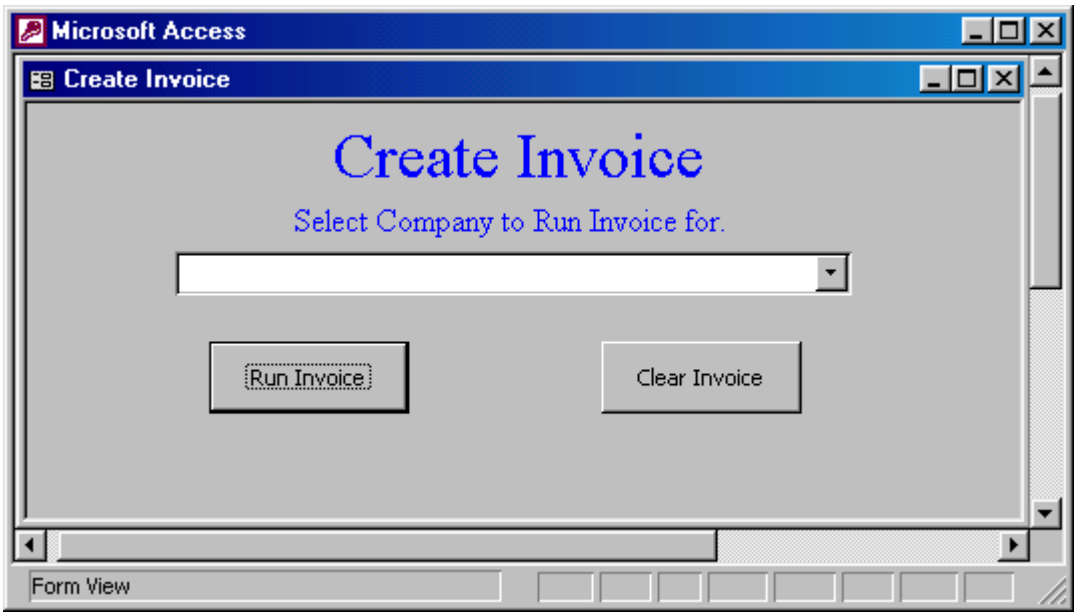

Figure 24. Form frmInvoiceFilter, Create Invoice

Once a company is selected and the run invoice button is clicked, the following report (Figure 25) is an example of what the invoice looks like after it has been filtered. The invoice is grouped first by Company Name then Service Name. This keeps all the services together for easier viewing of what is being charged.

Figure 25. Run Invoice Report

The Service Rates form (Figure 26) is where the user enters the rates that they are using to calculate the total cost of the services. These rates are based on hourly rates that it takes to clean a particular fixture. There are command buttons on the form for saving the rates and closing the forms. This form can be accessed from any of the "service" forms so that the rates can be updated for each company or service if needed.

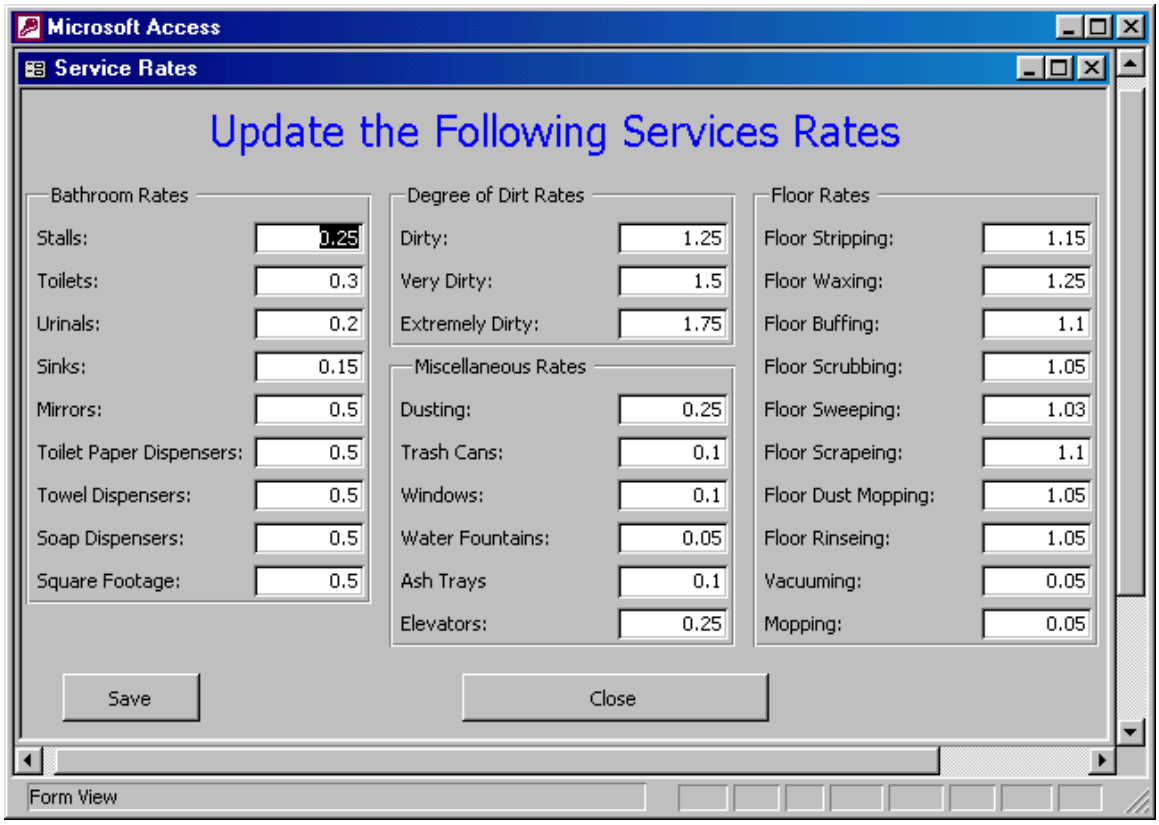

Figure 26. Form frmServiceRates, Service Rates

In order for the user to be able to garner any information, they must use the following "service" forms (Figures 27-38) to input information about the services that are ordered. The total cost of all these forms is used to calculate and produce the reports. These forms use a filter to only show those forms that match specific filter criteria. In the top right hand corner of the form, there is a text box that is used to filter through the database and only show information for that service. This filter is activated with the Form_Open event by running the code Me.FilterOn = True. This means that anytime the form is opened, the filter is applied to that form. The default value for the filter text box is set to be the service name so that future uses of the information is displayed properly in the correct corresponding form. The command button Configure Time and Cost runs a piece of code

(Appendix 1) that is the back bone of the application and the reason for doing this project. It reads in the values of the rates from the Service Rates form (Figure 26) and then calculates the total cost using hard-coded time specifications.

On the form frmBathroom (Figure 27), there are several text boxes on the form that the user utilizes to enter the number of fixtures into. They are as follows: Location, Stalls, Toilets, Urinals, Sinks, Mirrors, Trash Cans, Total Fixtures, Toilet Paper Dispensers, Towel Dispensers, Soap Dispensers, Floor Length, Floor Width, Square Footage and Total Cost. There are two combo box's used to select information. They are: Select Company and Tax Rate. There are two frames on the forms. The first frame is Dirt and it contains three option boxes labeled Dirty, Very Dirty, and Extremely Dirty. The second frame is Frequency and it contains six option boxes labeled Daily, Twice Daily, Three Times Daily, Weekly, Twice Weekly, and Three Times Weekly. There are command buttons for navigating through the different records. They are First, Previous, Next, Last, Add, Save, Delete and Close. There are two other command buttons, Configure Time and Cost and Change Rates. A SQL select statement that takes the company names from the table tblCompany in the database populates the Select Company combo box.

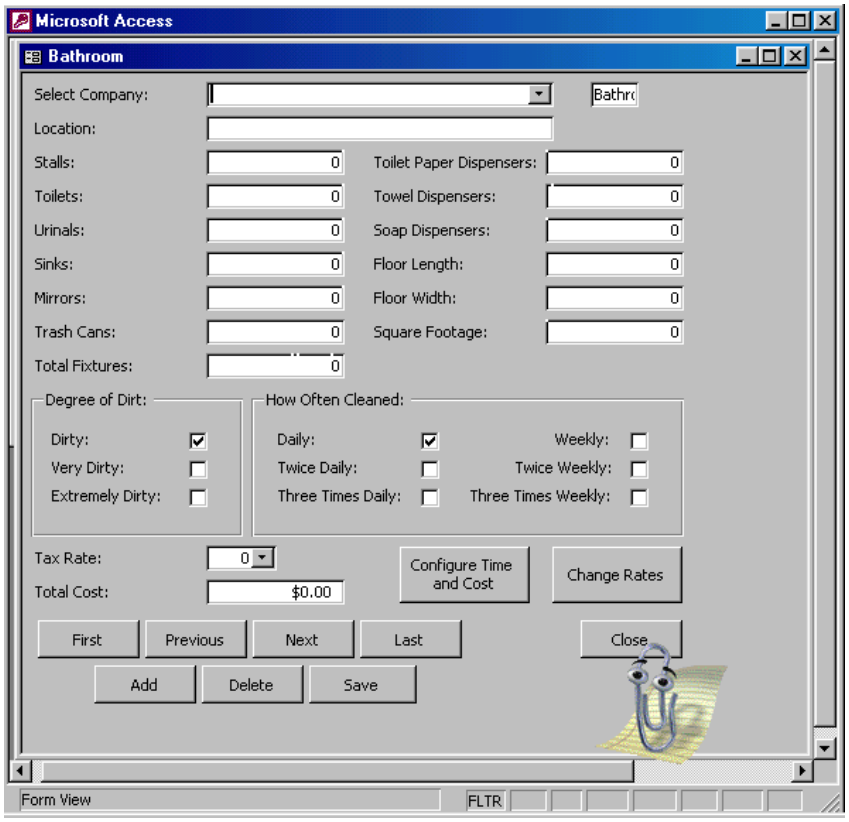

Figure 27. Form frmBathroom, Bathroom

On the form frmOffice (Figure 28), there are several text boxes on the form that the user utilizes to enter the number of fixtures into. They are as follows: Office Location, Desks, Chairs, Windows, Total Fixtures, Trash Cans, File Cabinets, Doors, Office Length, Office Width, Office Area and Total Cost. There are three combo box's used to select information. They are: Floor Type, Select Company and Tax Rate. There are two frames on the forms. The first frame is Dirt and it contains three option boxes' labeled Dirty, Very Dirty, and Extremely Dirty. The second frame is Frequency and it contains six option boxes labeled Daily, Twice Daily, Three Times Daily, Weekly, Twice Weekly, and Three Times Weekly. There are command buttons for navigating through the different records. They are First, Previous, Next, Last, Add, Save, Delete and Close.

A SQL select statement that takes the company names from the table tblCompany in the database populates the Select Company combo box.

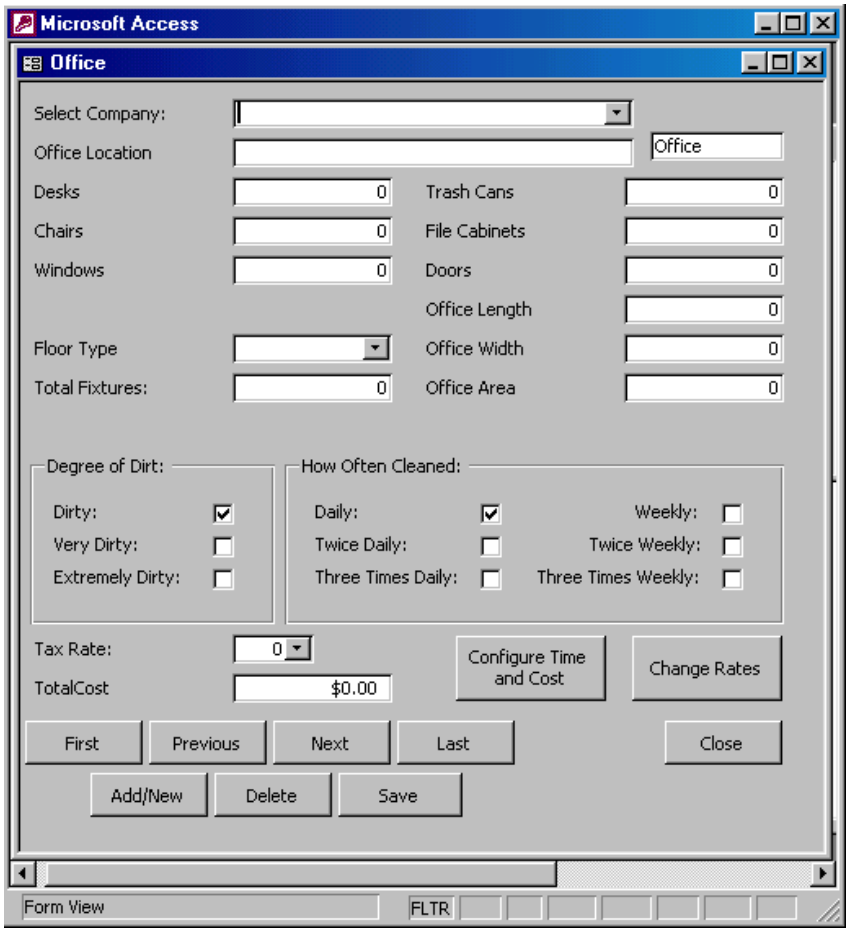

Figure 28. Form frmOffice, Office

On the form frmConferenceRoom (Figure 29), there are several text boxes on the form that the user utilizes to enter the number of fixtures into. They are as follows: Conference Location, Desks/Tables, Chairs, Windows, Total Fixtures, Trash Cans, File Cabinets, Doors, Floor Length, Floor Width, Floor Area and Total Cost. There are three combo box's used to select information. They are: Floor Type, Select Company and Tax

Rate. There are two frames on the forms. The first frame is Dirt and it contains three option boxes' labeled Dirty, Very Dirty, and Extremely Dirty. The second frame is Frequency and it contains six option boxes labeled Daily, Twice Daily, Three Times Daily, Weekly, Twice Weekly, and Three Times Weekly. There are command buttons for navigating through the different records. They are First, Previous, Next, Last, Add, Save, Delete and Close. A SQL select statement that takes the company names from the table tblCompany in the database populates the Select Company combo box.

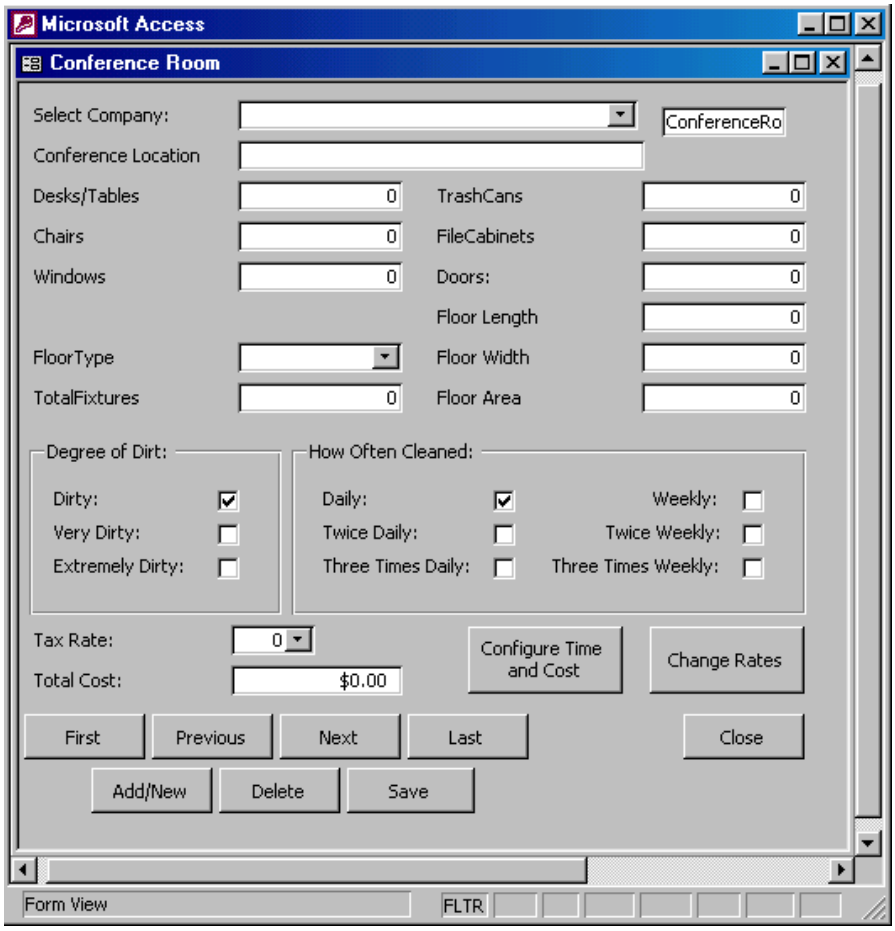

Figure 29. Form frmConferenceRoom, Conference Room

On the form frmHallway (Figure 30), there are several text boxes on the form that the user utilizes to enter the number of fixtures into. They are as follows: Location, Tables,

Chairs, Doors, Water Fountains, Total Fixtures, Trash Cans, Runners, Windows, Floor Length, Floor Width, Floor Area and Total Cost. There are three combo box's used to select information. They are: Floor Type, Select Company and Tax Rate. There are two frames on the forms. The first frame is Dirt and it contains three option boxes' labeled Dirty, Very Dirty, and Extremely Dirty. The second frame is Frequency and it contains six option boxes labeled Daily, Twice Daily, Three Times Daily, Weekly, Twice Weekly, and Three Times Weekly. There are command buttons for navigating through the different records. They are First, Previous, Next, Last, Add, Save, Delete and Close. A SQL select statement that takes the company names from the table tblCompany in the database populates the Select Company combo box.

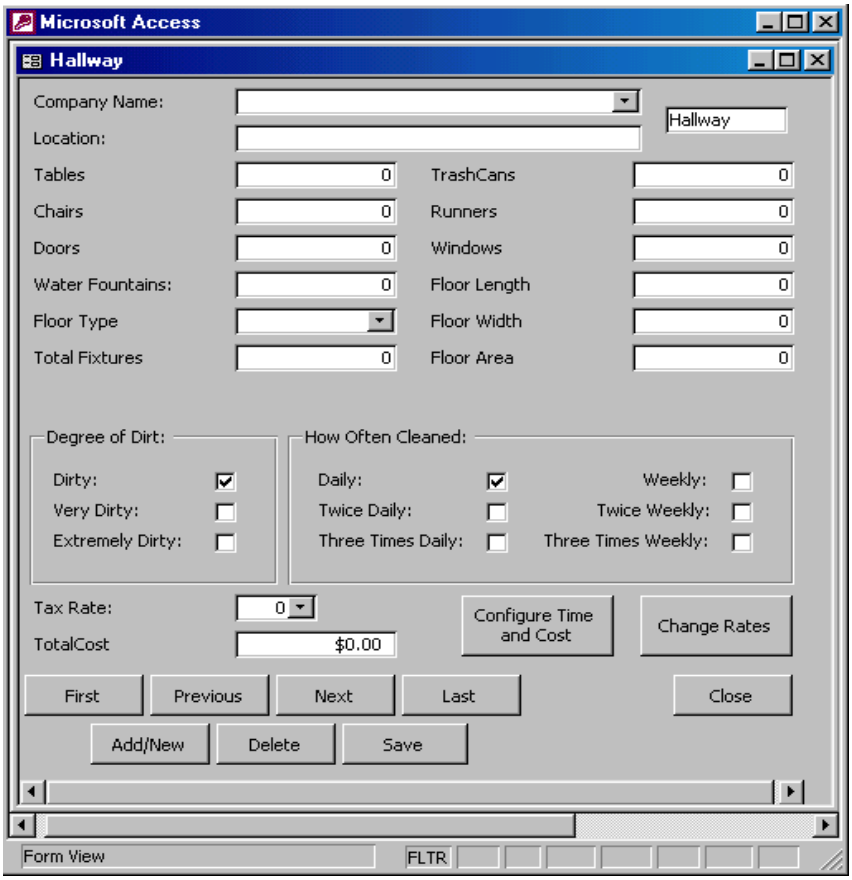

Figure 30. Form frmHallway, Hallway

On the form frmReceptionArea (Figure 31), there are several text boxes on the form that the user utilizes to enter the number of fixtures into. They are as follows: Reception Area Location, Desks, Chairs, Windows, Runners, Phones, Lamps, Plants, Total Fixtures, Trash Cans, File Cabinets, Doors, Water Fountains, Floor Length, Floor Width, Floor Area and Total Cost. There are three combo box's used to select information. They are: Floor Type, Select Company and Tax Rate. There are two frames on the forms. The first frame is Dirt and it contains three option boxes' labeled Dirty, Very Dirty, and Extremely Dirty. The second frame is Frequency and it contains six option boxes labeled Daily, Twice Daily, Three Times Daily, Weekly, Twice Weekly, and Three Times Weekly. There are command buttons for navigating through the different records. They are First, Previous, Next, Last, Add, Save, Delete and Close. A SQL select statement that takes the company names from the table tblCompany in the database populates the Select Company combo box.

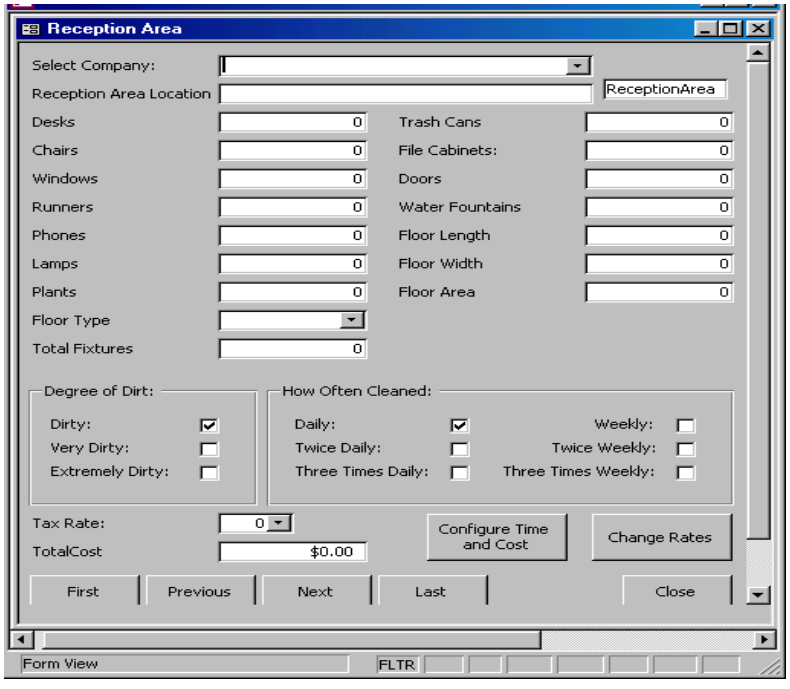

Figure 31. Form frmReceptionArea, Reception Area

On the form frmOther (Figure 32), there are several text boxes on the form that the user utilizes to enter the number of fixtures into. They are as follows: Location, Table/Desks, Chairs, Doors, Shelves, Ash Trays, Towel Dispensers, Mirrors, Toilets, Stalls, Total Fixtures, Trash Cans, Runners, Floor Mats, Water Fountains, Sinks, Toilet Paper Dispensers, Soap Dispensers, Urinals, Windows, Floor Length, Floor Width, Floor Area and Total Cost. There are three combo box's used to select information. They are: Floor Type, Select Company and Tax Rate. There are two frames on the forms. The first frame is Dirt and it contains three option boxes' labeled Dirty, Very Dirty, and Extremely Dirty. The second frame is Frequency and it contains six option boxes labeled Daily, Twice Daily, Three Times Daily, Weekly, Twice Weekly, and Three Times Weekly. There are command buttons for navigating through the different records. They are First, Previous, Next, Last, Add, Save, Delete and Close. A SQL select statement that takes the company names from the table tblCompany in the database populates the Select Company combo box.

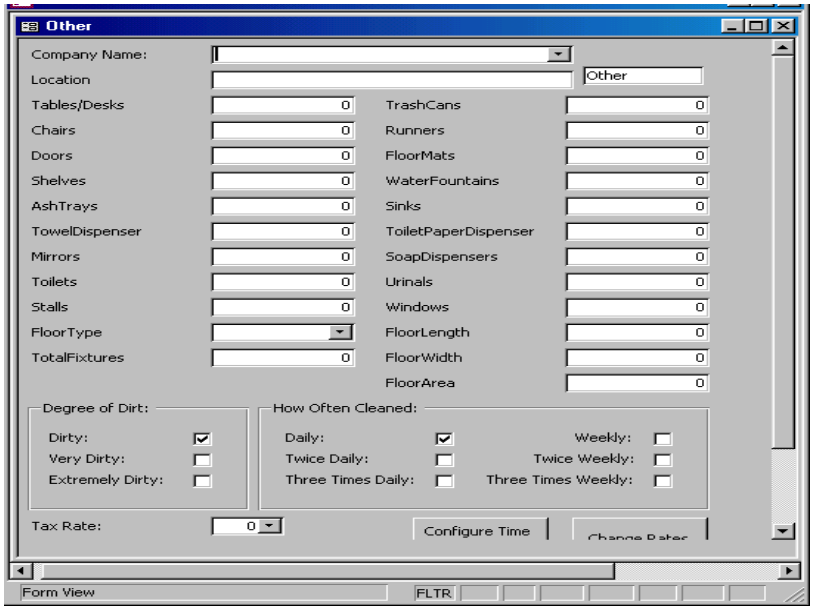

Figure 32. Form frmOther, Other

On the form frmBreakRoom (Figure 33), there are several text boxes on the form that the user utilizes to enter the number of fixtures into. They are as follows: Location, Tables, Chairs, Windows, Refridgerators, Vending Machines, Runners, Total Fixtures, Trash Cans, Sinks, Lunch Line, Doors, Water Fountains, Floor Length, Floor Width, Floor Area and Total Cost. There are three combo box's used to select information. They are: Floor Type, Select Company and Tax Rate. There are two frames on the forms. The first frame is Dirt and it contains three option boxes' labeled Dirty, Very Dirty, and Extremely Dirty. The second frame is Frequency and it contains six option boxes labeled Daily, Twice Daily, Three Times Daily, Weekly, Twice Weekly, and Three Times Weekly. There are command buttons for navigating through the different records. They are First, Previous, Next, Last, Add, Save, Delete and Close. A SQL select statement that takes the company names from the table tblCompany in the database populates the Select Company combo box.

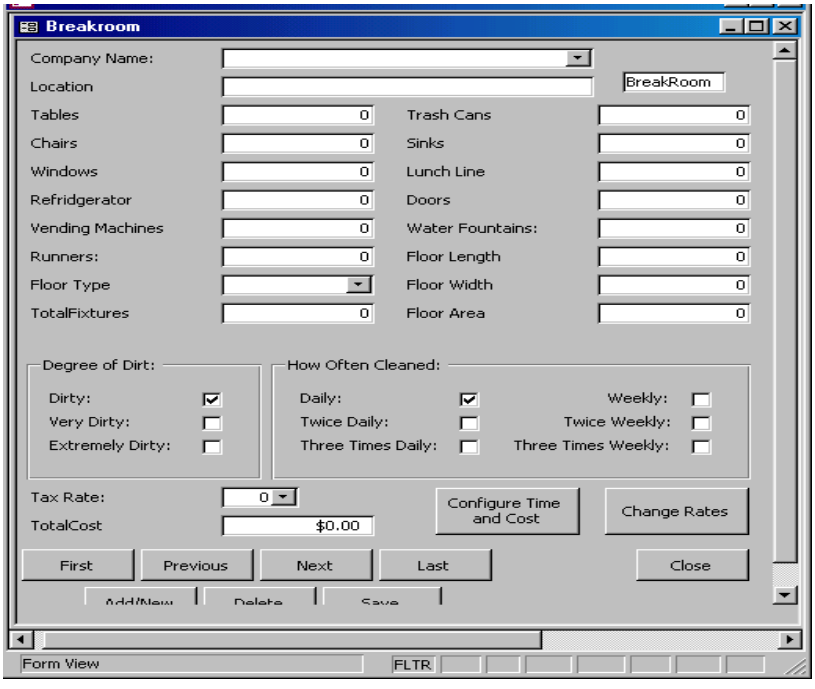

Figure 33. Form frmBreakroom, Break Room

On the form frmStorageArea (Figure 34), there are several text boxes on the form that the user utilizes to enter the number of fixtures into. They are as follows: Location, Windows, Sinks, Closets, Tables, Chairs, Total Fixtures, Trash Cans, Shelves, Lights, Doors, Floor Length, Floor Width, Floor Area and Total Cost. There are three combo box's used to select information. They are: Floor Type, Select Company and Tax Rate. There are two frames on the forms. The first frame is Dirt and it contains three option boxes' labeled Dirty, Very Dirty, and Extremely Dirty. The second frame is Frequency and it contains six option boxes labeled Daily, Twice Daily, Three Times Daily, Weekly, Twice Weekly, and Three Times Weekly. There are command buttons for navigating through the different records. They are First, Previous, Next, Last, Add, Save, Delete and Close. A SQL select statement that takes the company names from the table tblCompany in the database populates the Select Company combo box.

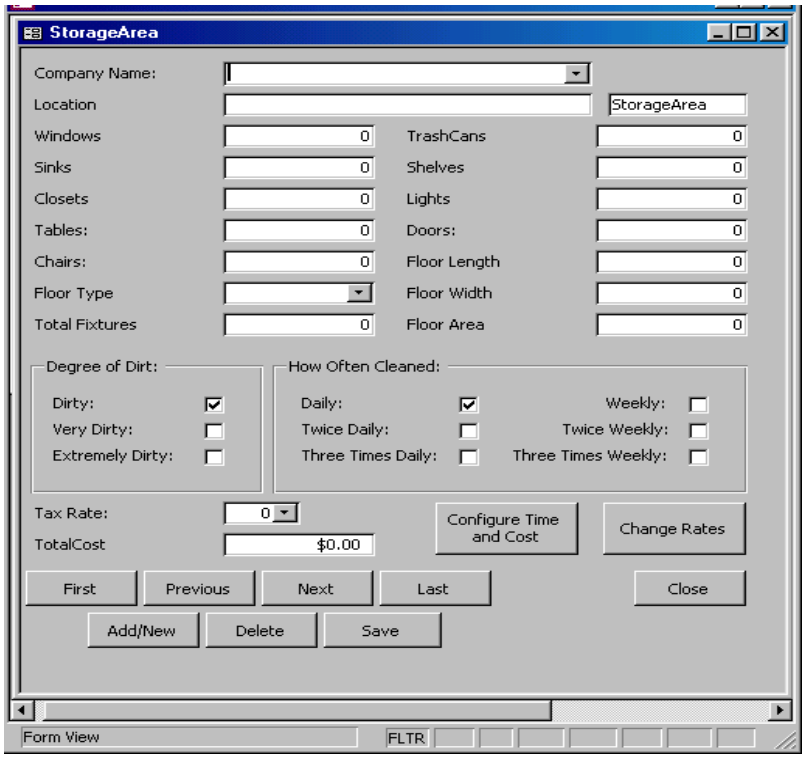

Figure 34. Form frmStorageArea, Storage Area

On the form frmWarehouse (Figure 35), there are several text boxes on the form that the user utilizes to enter the number of fixtures into. They are as follows: Location, Desks, Tables, Chairs, Runners, Closets, Doors, Total Fixtures, Trash Cans, Shelves, Racks, Water Fountains, Windows, Floor Length, Floor Width, Floor Area and Total Cost. There are three combo box's used to select information. They are: Floor Type, Select Company and Tax Rate. There are two frames on the forms. The first frame is Dirt and it contains three option boxes' labeled Dirty, Very Dirty, and Extremely Dirty. The second frame is Frequency and it contains six option boxes labeled Daily, Twice Daily, Three Times Daily, Weekly, Twice Weekly, and Three Times Weekly. There are command buttons for navigating through the different records. They are First, Previous, Next, Last, Add, Save, and Delete. A SQL select statement that takes the company names from the table tblCompany in the database populates the Select Company combo box.

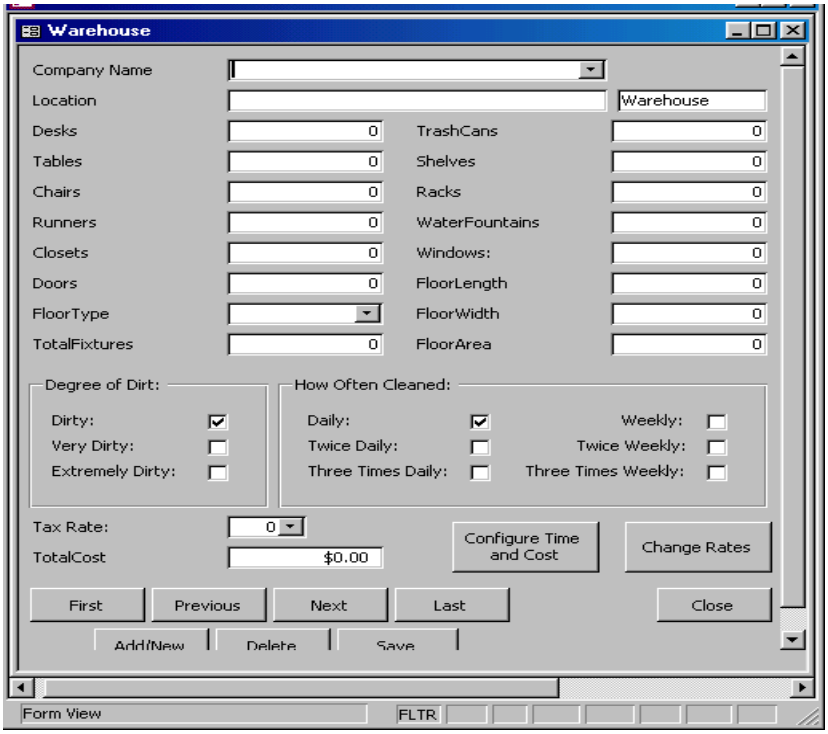

Figure 35. Form frmWarehouse, Warehouse

On the form frmKitchen (Figure 36), there are several text boxes on the form that the user utilizes to enter the number of fixtures into. They are as follows: Location, Doors, Chairs, Sinks, Water Fountains, Floor Mats, Total Fixtures, Trash Cans, Counters, Desks, Tables, Windows, Floor Length, Floor Width, Floor Area and Total Cost. There are three combo box's used to select information. They are: Floor Type, Select Company and Tax Rate. There are two frames on the forms. The first frame is Dirt and it contains three option boxes' labeled Dirty, Very Dirty, and Extremely Dirty. The second frame is Frequency and it contains six option boxes labeled Daily, Twice Daily, Three Times Daily, Weekly, Twice Weekly, and Three Times Weekly. There are command buttons for navigating through the different records. They are First, Previous, Next, Last, Add, Save, Delete and Close. A SQL select statement that takes the company names from the table tblCompany in the database populates the Select Company combo box.

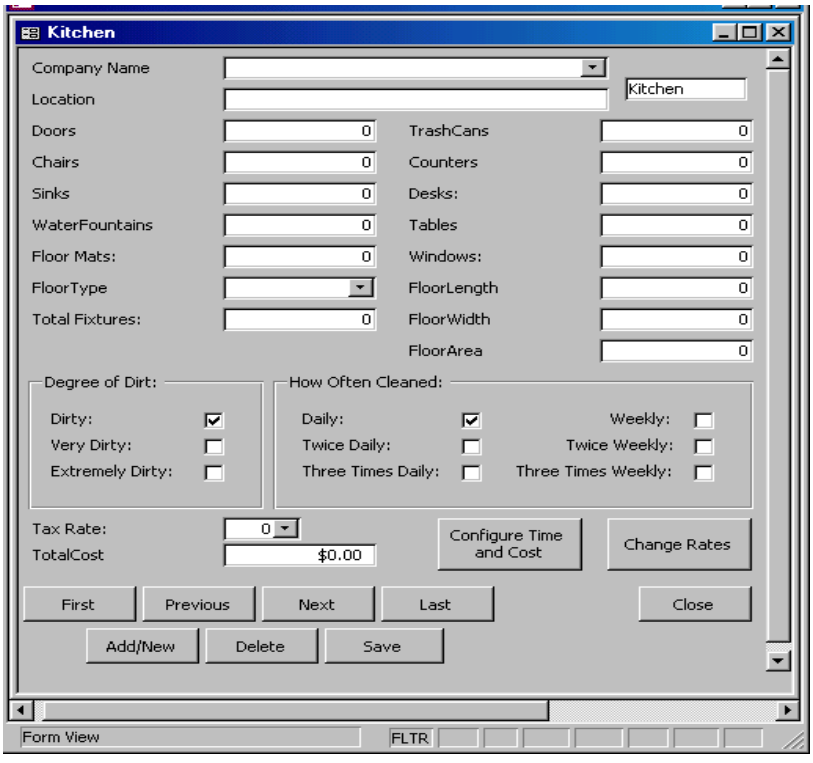

Figure 36. Form frmKitchen, Kitchen

On the form frmEntrance (Figure 37), there are several text boxes on the form that the user utilizes to enter the number of fixtures into. They are as follows: Location, Tables, Chairs, Doors, Windows, Ash Trays, Total Fixtures, Trash Cans, Elevators, Water Fountains, Floor Length, Floor Width, Floor Area and Total Cost. There are three combo box's used to select information. They are: Floor Type, Select Company and Tax Rate. There are two frames on the forms. The first frame is Dirt and it contains three option boxes' labeled Dirty, Very Dirty, and Extremely Dirty. The second frame is Frequency and it contains six option boxes labeled Daily, Twice Daily, Three Times Daily, Weekly, Twice Weekly, and Three Times Weekly. There are command buttons for navigating through the different records. They are First, Previous, Next, Last, Add, Save, Delete and Close. A SQL select statement that takes the company names from the table tblCompany in the database populates the Select Company combo box.

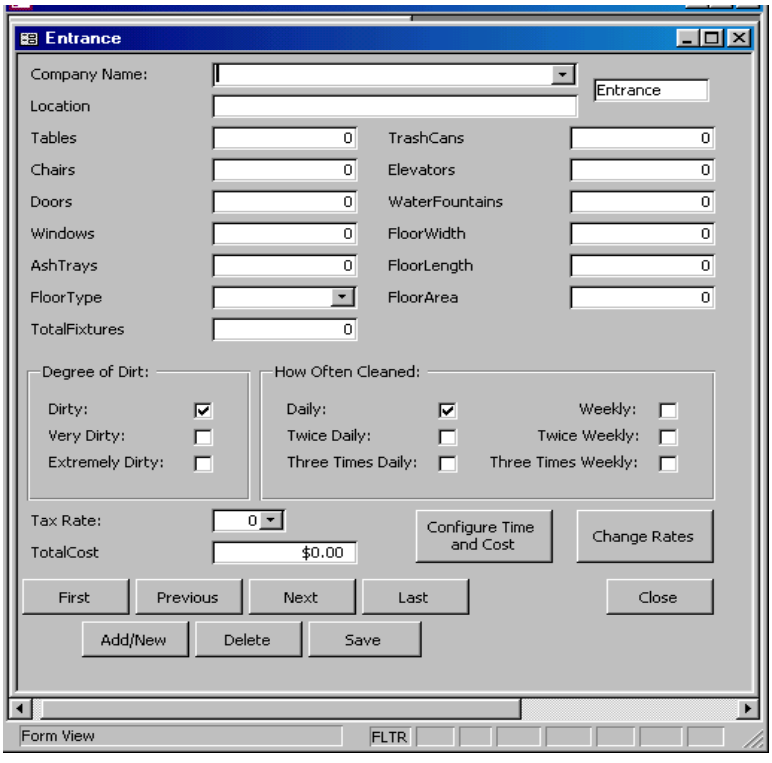

Figure 37. Form frmEntrance, Entrance

On the form frmFloors (Figure 38), there are several text boxes on the form that the user utilizes to enter the number of fixtures into. They are as follows: Location, Floor Length, Floor Width, Floor Area and Total Cost. There are three combo box's used to select information. They are: Select Company, Floor Type and Tax Rate. There are eight option boxes on the form. They are: Scrub Floors, Strip Floors, Rinse Floors, Buff Floors, Dust Mop Floors, Sweep Floors, Scrape Floors and Wax Floors. If the Floor Type selected is Carpet, then the option boxes Strip Floors, Buff Floors, Dust Mop Floors, Scrape Floors and Wax Floors are disabled because they do not apply to carpet. There are command buttons for navigating through the different records. They are First, Previous, Next, Last, Add, Save, Delete and Close. A SQL select statement that takes the company names from the table tblCompany in the database populates the Select Company combo box.

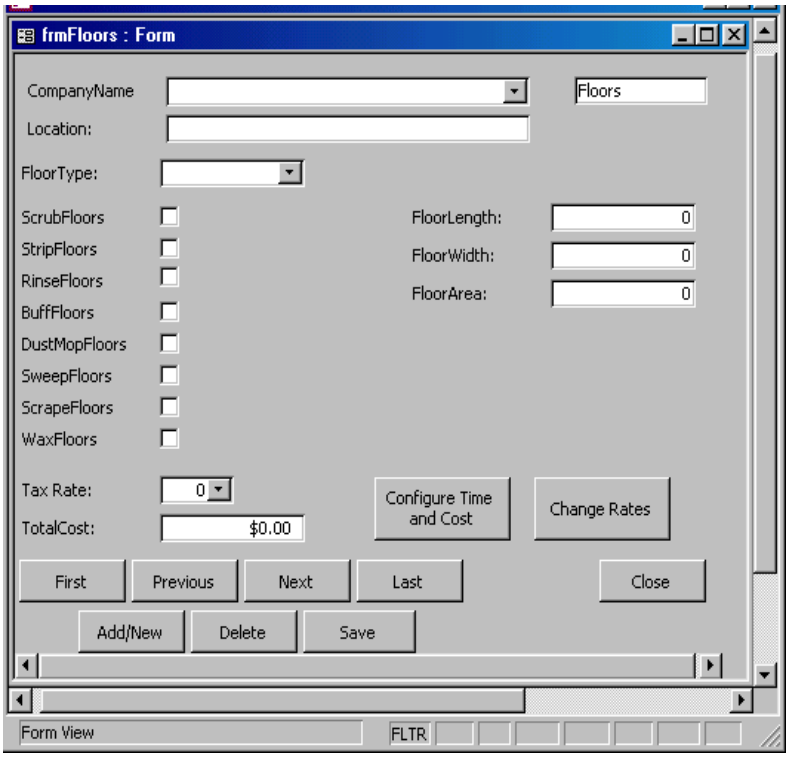

Figure 38. Form frmFloors, Floors

There is also a command button on the Main Menu for access to the application information. This information is for both version description and contains information for support contacts. It also marks this application as proprietary software that is copyright protected.

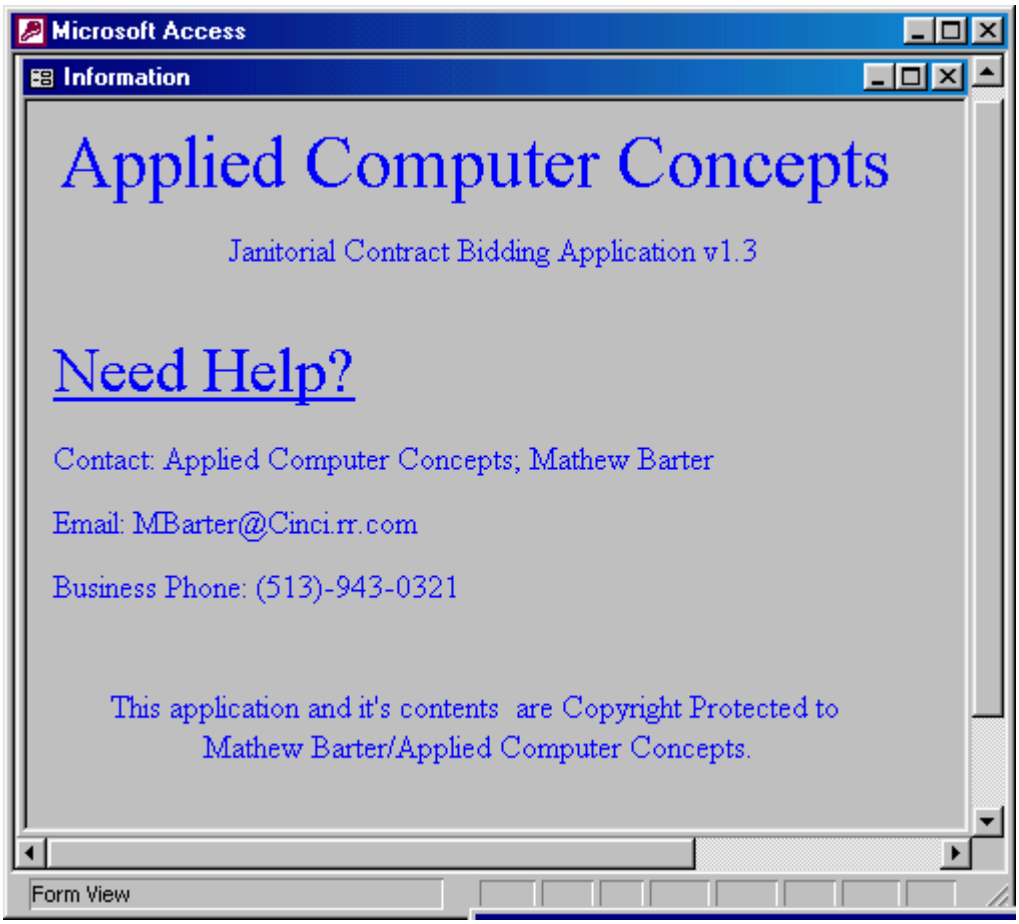

Figure 39. Form frmInformation, Information

7. Conclusions and Recommendations

While working on this project, I came to realize the full cycle that a project can travel. Along that route, I ran into many decisions that had to be made. Some of those decisions were made in haste and I spent a good amount of time fixing those mistakes in the future (ex. Reconstructing all my tables three quarters of the way into my project). Overall I

learned that the key to programming a big program is to lie out a good plan of attack. The foundation of the project makes the rest of the project go much smoother.

I have already decided to make several updates to this software after I get it to market. I want to revamp most of the code to include function statements that I can call into all my forms rather than hard coding all the forms. This will give me the ability to manage my code easier in future releases. I also want to incorporate some work loading programs that can be used to forecast what the future demands may be for a particular location given different factors (labor increases, supply cost increase, etc.). In a different version I will probably go away from having the user select the company name each time they want to add each service. This will be sufficient for use with small companies because of the limited number of service to add but it would be rather cumbersome to use on very large accounts. I would recommend having one location selected and then applying that location name to each of the subsequent services added until the location name changes.

Bibliography

1. Beardslee, Randy. Owner, Make Cents. Personal interview. January 31, 2001

2. Doshier, Curt. Owner, Southwest Commercial Services. Personal interview. January 31, 2001

3. Fulmer, Robert. "Profit from organization with software". *Cleaning and Maintenance Management Magazine*. January 10, 2001. 1-2

4. Graham, John. "Good news, bad news". *Cleaning and Maintenance Management Magazine*. January 6, 2001. 6-10

5. Lusey, Mike. Vice President Marketing, FBG Service Corporation. Personal interview. January 25, 2001

6. Rogers, Jason. "Databank: survey says: cleaning professionals love computers". *Cleaning and Maintenance Management Magazine*. January 6, 2001. 6-10

Appendix 1

Configure Time and Cost Code (Backbone code to the project)

```
Private Sub cmdTimeandCost_Click()
'Calculate the total fixtures
TotalFixtures = Tables + Chairs + Doors + Shelves + AshTrays + TowelDispenser + _ Mirrors +
Toilets + Stalls + TrashCans + Runners + FloorMats + WaterFountains + _ Sinks  +
ToiletPaperDispenser + SoapDispensers + Urinals + Windows

'Calculate the tax rate
Tax = (TaxRate / 100) + 1
'Calculate the Area of the Office
FloorArea = FloorLength * FloorWidth

'Determine the floortime to be used by selection
If FloorType = "Tile" Then
      FloorTime = 1.09
      iFloorRate = Form_frmServiceRates.MoppingRate
      ElseIf FloorType = "Carpet" Then
          FloorTime = 1.12
          iFloorRate = Form_frmServiceRates.VacuumingRate
          ElseIf FloorType = "Concrete" Then
             FloorTime = 1.16
             iFloorRate = Form_frmServiceRates.MoppingRate
             ElseIf FloorType = "Elevated Tile" Then
                FloorTime = 1.09
                iFloorRate = Form_frmServiceRates.MoppingRate
                ElseIf FloorType = "Other" Then
                   FloorTime = 1
                   iFloorRate = Form_frmServiceRates.FloorSweeping
                   Else
                      FloorTime = 1
                      iFloorRate = Form_frmServiceRates.FloorSweeping
End If

'Calculate the Dusting Total
iDustingTotal = (Tables + Doors + Shelves + Windows) * Form_frmServiceRates.DustingRate
'Calculate time and cost for each service
iTrashCansTotal = ((TrashCans * 5) / 60) * Form_frmServiceRates.TrashCansRate
iWindowsTotal = ((Windows * 20) / 60) * Form_frmServiceRates.WindowsRate
iChairsTotal = ((Chairs * 10) / 60) * Form_frmServiceRates.VacuumingRate
iRunnersTotal = (((FloorMats + Runners) * 10) / 60) * Form_frmServiceRates.VacuumingRate
iWaterFountainTotal = ((WaterFountains * 10) / 60) *
Form_frmServiceRates.WaterFountainRate
iSinksTotal = ((Sinks * 20) / 60) * Form_frmServiceRates.SinksRate
iAshTrayTotal = ((AshTrays * 15) / 60) * Form_frmServiceRates.AshTraysRate
```

Appendix 1 Continued

```
iTowelDispensers = ((TowelDispenser * 30) / 60) *
Form_frmServiceRates.TowelDispensersRate
iSoapDispensers = ((SoapDispensers * 30) / 60) * Form_frmServiceRates.SoapDispensersRate
iMirrors = ((Mirrors * 45) / 60) * Form_frmServiceRates.MirrorsRate
iToilets = ((Toilets * 240) / 60) * Form_frmServiceRates.ToiletsRate
iStalls = ((Stalls * 120) / 60) * Form_frmServiceRates.StallsRate
iToiletPaperDispensers = ((ToiletPaperDispenser * 30) / 60) *
Form_frmServiceRates.ToiletPaperDispensersRate
iUrinals = ((Urinals * 120) / 60) * Form_frmServiceRates.UrinalsRate

'Determines how dirty it is and adds a premium
If Dirt = 1 Then
    iDirt = Form_frmServiceRates.DirtyRate
    ElseIf Dirt = 2 Then
       iDirt = Form_frmServiceRates.VeryDirtyRate
       ElseIf Dirt = 3 Then
          iDirt = Form_frmServiceRates.ExtremelyDirtyRate
          Else
             iDirt = 1
End If

'Determines the Frequency of how often it will be completed
If Frequency = 1 Then ' Based on Weekly Rates
    iFrequency = 7 'Days
    ElseIf Frequency = 2 Then
       iFrequency = 14
       ElseIf Frequency = 3 Then
          iFrequency = 21
          ElseIf Frequency = 4 Then
             iFrequency = 1
             ElseIf Frequency = 5 Then
                iFrequency = 2
                ElseIf Frequency = 6 Then
                   iFrequency = 3
                   Else
                      iFrequency = 1
End If
'Calculate the cost of vacuuming office
iFloorTotal = FloorArea * iFloorRate * FloorTime
'Calculate the Total Cost
TotalCost = (((iMirrors + iToilets + iStalls + iToiletPaperDispensers + iUrinals +
iTowelDispensers + iSoapDispensers + iSinksTotal + iWaterFountainTotal + iDustingTotal +
iRunnersTotal + iFloorTotal + iTrashCansTotal + iWindowsTotal + iChairsTotal +
iAshTrayTotal) * iFrequency) * iDirt) * Tax
End Sub
```