NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

STOCK MANAGEMENT SYSTEM

GRADUATION PROJECT
COM - 400

Student: Ahmet Kenan KÖŞGER

Supervisor: Mr. Halil ADAHAN

Nicosia 2004

Abstract

Computer industry is always developing. Invention of the Calculus and up to now, people are using computer's technology in everywhere like the space sector. Computer technology is everywhere. People are using these technologies for the comfort that decrease the spending of time when you are making a work in the anywhere.

Computer Information Systems also part of Computer Industry that is related software. C.I.S creates System Solutions, Package Program solutions etc. In my project I was choose package program solutions for CEMENT CO. related that daily transaction of CEMENT Company. This project helps the user about data storage when who need that data to use again.

When I preparing that project I was used Visual Basic program language and Microsoft Access for the input data storage.

List of Figures

List of Tables

Table of Contents

Chapter I: Introduction

1. Introduction

A database management system (DBMS) consists of a collection of interrelated data and a set of programs to access that data. The collection of data, usually refered to as the database, contains information about one particular enterprise. The primary goal of a DBMS is to provide an environment that is both convenient and efficient. To use in retrieving and storing database information.

Database systems are designed to manage large bodies of information. The management of data involves both the definition of structures for the storage of information and the provision of mechanisms for the manipulation of information. In addition, the database system must provide for the safety of the information stored, despite system crashes or attempts at unauthorized access. If the data is to be shared among several users, the system must avoid possible unexpected results.

The importance of information in most organizations, and hence the value of the database, has led to the development of a large body of concepts and techniques for the efficient management of data.

In a typical file processing system, permanent records are stored in various files, and a number of different application programsa re written to extract records from and add records to the appropriate file. This scheme has a number of major disadvantages.

1. **Data Redundancy and Inconsistency:** Since the files and application programs are created by different programmers over a long period of time, the files are likely to have different formats and the programs may be written in several programming languages. Moerover, the same piece of information, may be duplicated in several files.

2. **Difficulty in Accessing Data:** Conventional file-processing environments do not allow data to be retrieved in a convenient and effiicient manner. Beter data retrieval systems must be developed for general use and applications.

3. **Data Isolation:** Since data is scattered in various files, and files may be in different formats, it is difficult to write new application programs to retrieve the appropriate data.

4. **Concurrent Access Anomalies:** In order to improve the overall performance of the system and obtain a faster redponse time, many systems allow multiple users to update the data simultaneously. In such an environment, interaction of concurrent updates may result in consistent data. So a supervision must be maintained in the system. Since the data may be accessed by many different application programs which have noot been previously coordinated, supervision is very difficult to provide.

5. **Security Problems:** Not every user of the database system should be able to access all the data. Since application programs are added to the system in an ad hoc manner, it is difficult to enforce such security constraints.

6. **Integrity Poblems:** Th data values stored in the database must satisfy certain types of consistency constraints. These constraints are enforced in the system by adding appropriate code in the various application programs. However, when new constraints are added, it is difficult to change the programs to enforce them. The problem is compounded when constraints involve several data items from different files.

1.1 Report Project Bisection

The rest of this repot is organized as follows. In Chapter Two, necessary background information needed to cope with the project is introduced. Chapter Three presents program design that were taken into consideration while progressing in the project. The conclusion of the project is given in Chapter Four. All the references used in this project are presented in the References Section. As for the appendices supplementing this project, they will cover the appropriate codes.

Chapter II: Microsoft Database Access

2.1 Microsoft Access Database

2.1.1 Access Mechanism

Microsoft Access is a relational database management system (DBMS). At the most basic level, a DBMS is a program that facilitates the storage and retrieval of structured information on a computer's hard drive. Examples of well-know industrial-strength relational DBMSes include:

1. Oracle.
2. Microsoft SQL Server.

Well- know PC- based ("desktop") relational DBMSes include:

1. Microsoft Access
2. Microsoft FoxPro

2.1.2 Different Faces of Access

Microsoft generally likes to incorporate as many features as possible into its products. For example,

The Access package contains the following elements:

1. A relational database system that supports two industry standard query languages: Structured Query Language (SQL) and Query By Example (QBE).
2. A full-featured procedural programming language-essentially a subset of Visual Basic.
3. A simplified procedural macro language unique to Access.
4. A rapid application development environment complete with visual form and report development tools.
5. A sprinkling of objected-oriented extensions.
6. A various wizards and builders to make development easier for new users, these "multiple personalities" can be a source of enormous frustration. The problem is that each personality is based on a different set of assumptions and a different view of computing.
7. The relational database personality expects you to view your application as sets of data.

8. The procedural programming personality expects you to view your application as commands to be executed sequentially.
9. The object-oriented personality expects you to view your application as objects, which encapsulate state and behavior information.

Microsoft makes no effort to provide an overall logical integration of these personalities (indeed, it is unlikely that such an integration is possible). Instead, it is up to you as a developer to pick and choose the best approach to implementing your application.

Since there are often several vastly different ways to implement a particular feature in Access, recognizing the different personalities and exploiting the best features (and avoiding the pitfalls) of each are important skills for Access developers.

The advantage of these multiple personalities is that it is possible to use Access to learn about an enormous range of information systems concepts without having to interact with a large number of "single-personality" tools, for example:

1. Oracle for relational databases
2. PowerBuilder for rapid applications development,
3. Small Talk for object- oriented programming.

Keep this advantage in mind as we switch back and forth between personalities and different computing paradigms.

We will see in the following figure (Figure 2.1) the main features of the database window including the tabs for viewing the different database objects:

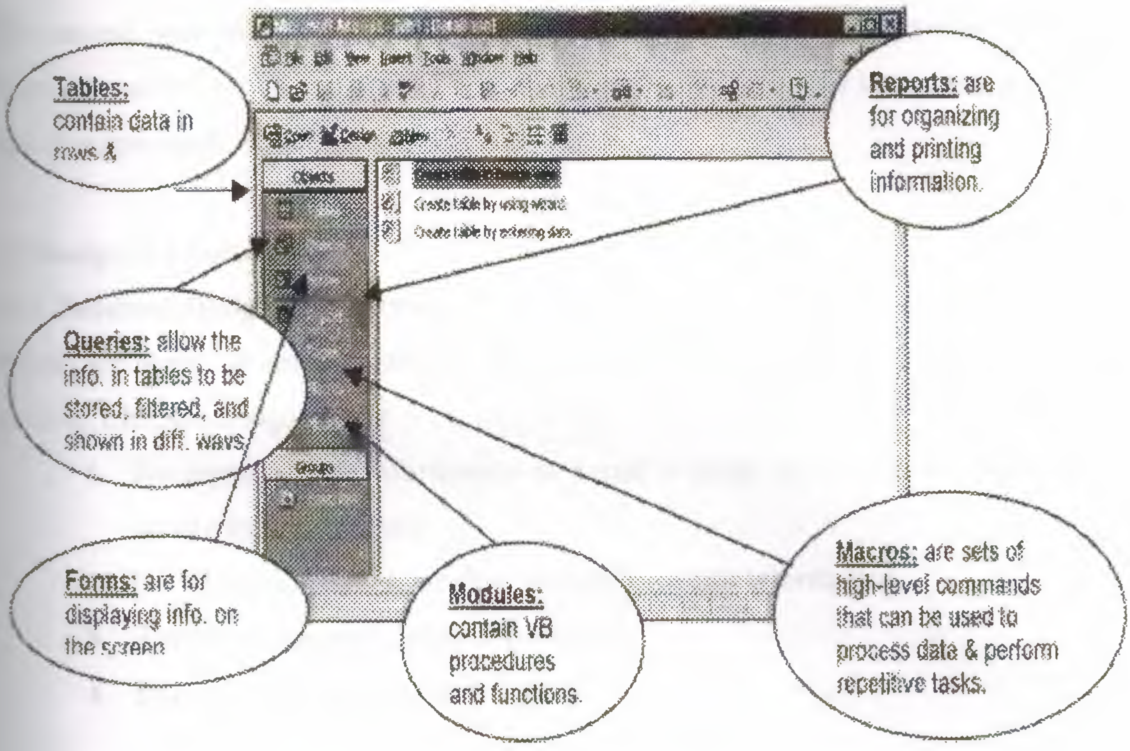

Tables: contain data in rows &
Reports: are for organizing and printing information.
Queries: allow the info. in tables to be stored, filtered, and shown in diff. ways.
Forms: are for displaying info. on the screen
Modules: contain VB procedures and functions.
Macros: are sets of high-level commands that can be used to process data & perform repetitive tasks.

Figure 2.1: Microsoft Access functions and parts in the front view.

2.2 Microsoft Visual Basic

2.2.1 Learning the Basics of Programming

Programming can be an enormously complex and difficult activity. Or it can be quite straightforward. In either case, the basic programming concepts remain the same.

Strictly speaking, the language that is included with Access is not Visual Basic-it is a subset of the full, stand- alone Visual Basic language (which Microsoft sells separately).

In Access version 2.0, the subset is called “Access Basic”. In version 2000, it is slightly enlarged subset called “Visual Basic for Applications” (VBA). However, in the context of the simple programs we are writing here, these terms are interchangeable.

2.2.2 Interacting with the interpreter

Access provides two ways of interacting with the VBA language. The most useful of these is through saved modules that contain VBA procedures. These Procedures (subroutines and functions) can be run to do interesting things like process transactions against master tables, provide sophisticated error checking, and so on.

The second way to interact with VBA is directly through the interpreter. Interpreted languages are easier to experiment with since you can invoke the Interpreter at any time, type in a command, and watch its execution.

2.3 Design of a Database

2.3.1 Database Design Architecture

When we want to make a design for a database, this design must be good and suitable. Efficient design means:

1. No repetition of information-to avoid wasting space and complicating the update for the database.
2. No losing of information-then we will have less information.
3. Ability to represent certain information
4. Easy and fast search ability

In designing such a system, it may become necessary to decompose a relation to a number of smaller relations. Using functional dependencies, we can define several normal forms, which represent good database designs. There are a large number of normal forms, but the most important forms are BCNF and 3NF.

There is something called dependency preservation, since the update is made to the database, the system should be able to check that the update will not create an illegal relation-that is, one that does not satisfy all of the design functional dependencies. In order to check updates efficiently, it's desirable to design relational database schemes that allow update validation without the computation of joins. So we have to check the relations individually to determine what functional dependencies may be tested.

2.3.2 Bote-Codd Normal Form (BCNF)

It is one of the more desirable normal forms. A relation scheme R is in BCNF if for all functional dependencies that hold on R of the form $\alpha \rightarrow \beta$, where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following holds:

1. $\alpha \rightarrow \beta$ is a trivial functional dependency (that is, $\beta \subseteq \alpha$).
2. α is a superkey for scheme R.

A database design is in BCNF if each member of the set of relation schemes comprising the design is in BCNF. So, to understand how this decomposition work we can see the algorithm:

```
    Result: = {R;
    Done: = false;
    Compute $F^+$;

    While (not Done) do
    If (there is a scheme $R_i$ in Result that is not in BCNF)
    Then begin
    Let $\alpha \to \beta$ be a nontrivial functional dependency that holds
on $R_i$ such
    that $\alpha \to R_i$ is not in $F^+$, and $\alpha \cap \beta \equiv \emptyset$;
    Result:= ( Result-$R_i$ ) $\cup$ ( $R_i$ $\beta$ ) $\cup$ ( $\alpha$ , $\beta$ );
    end
    Else Done:= true;
```

Pay attention to the fact, not every BCNF decomposition is dependency-preservation. So that's why we have a weaker normal form called 3NF.

2.3.3 Third Normal Form (3NF)

It is a good normal form, which allow the nontrivial functional dependencies whose left side is not a superkey. A relation scheme R is in 3NF if for all functional dependencies that hold on R of the form $\alpha \to \beta$, where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following holds:

1. $\alpha \to \beta$ is a trivial functional dependency (that is, ft c a).
2. α is a superkey for scheme R.
3. Each attribute A in β is contained in a candidate key for R.

To understand how this decomposition work we can see the algorithm:

```
i:= 0;
For each functional dependency $\alpha \to \beta$ in
F do
If none of the schemes $R_j$, $1 \leq j \leq i$ contains $\alpha$ $\beta$
Then begin
```

```
i:= i + 1;
$R_i$:= α  β
end
If none of the schemes $R_j$, 1 ≤ j ≤ i contains a candidate key for R
Then begin
i:= i + 1;
$R_i$:= any candidate key for R;
end
Return ( $R_1$, $R_2$, ….., $R_i$ )
```

So, we have seen that every BCNF scheme is also in 3NF, and BCNF is more restrictive constraint than 3NF.

2.4 Microsoft SQL Server

Microsoft SQL Server 2000 is fully Web-enabled with end-to-end support for Extensible Markup Language (XML). It also can be considered a new, integrated data-mining engine, allowing customers to benefit from the efficiencies and opportunities offered by business-to-business (B2B) and business-to-consumer (B2C) E-commerce. The Microsoft Windows® DNA platform, including SQL Server already leads in the B2B E-commerce industry with support from original vendors.

SQL Server 2000 offers scalability and reliability for the largest Web and corporate applications through groundbreaking support for software scale-out, high-end hardware platforms and state-of-the-art networking and storage technologies to compete with its competitor Oracle. With the industry's most modern database architecture and deep integration with the Microsoft Windows DNA 2000 platform, SQL Server 2000 offers the fastest time-to-market, time needed to produce a product to the market, for building applications for the Business Internet, helping companies realize the possibilities of the digital economy.

"Microsoft Windows DNA 2000 is the most comprehensive integrated platform for building and deploying next-generation Web applications. SQL Server 2000 anchors this platform by delivering unequaled support for Web technologies; deeply integrating business intelligence features, and making software scale a reality for the most demanding e-commerce scenarios" as Microsoft claims.

2.4.1 Fully Web-Enabled for B2B E-Commerce

SQL Server 2000, in conjunction with the complete Microsoft developing application, like Windows DNA 2000 platform and Microsoft Visual Basic, will make it easier for corporations of any size to conduct business over the Web with customers and suppliers, and to integrate legacy systems with next-generation business solutions. Furthermore, SQL Server 2000 will help companies better understand customer behavior on the Web and quickly discover new business opportunities offered by a digital economy.

SQL Server 2000 delivers fully integrated, standards-based XML support that is flexible, high-performance and easy to use for Web developers and database programmers. Unlike competitive databases, Oracle for example, SQL Server 2000 offers powerful tools for manipulating XML inside the database and enables developers to retrieve XML-based data from the database without writing code.

SQL Server 2000 support an integrated XML infrastructure that allows documents to be managed and routed from database applications through BizTalk Server 2000, across the enterprise and with valued trading partners.

Data mining automatically sifts through large volumes of business information, helping customers to identify untapped opportunities and to predict winning strategies and outcomes. By integrating a data-mining engine into SQL Server 2000 Analysis Services (formerly OLAP Services), Microsoft has vastly reduced the complexity of this sophisticated and powerful technology. Analysis Services includes clustering and decision tree data-mining algorithms and is easily extensible by third parties via OLE DB for Data Mining.

SQL Server 2000 enables high-performance, standards-based, secure access via the Web through HTTP, even through a firewall, which will be helpful in our system since some might register from their countries where there might be firewalls. Web-based clients retain full capabilities whether accessing the relational store or Analysis Services.

SQL Server 2000 Analysis Services includes unique, new features for analyzing Web click-stream data, performing closed-loop analysis, and sharing analysis results across the Web through firewalls.

Simplifying access to strategic data resources, SQL Server 2000 English Query allows end users of all skill levels to pose questions in English to the database via the Web. English Query translates a user's question into a proper database query and returns the desired results from the relational store or Analysis Services. Unfortunately this feature has no importance to our project since all our work is Visual Basic based which leaves nothing for the SQL English Query.

2.4.2 Highly Scalability and Reliability

SQL Server 2000 offers scalability and reliability for the most demanding Web and enterprise applications. As proof of this, SQL Server 2000 has achieved world-record performance for a number of industry benchmarks, including the Transaction Processing Performance Council's TPC-C benchmark, eclipsing competing products running on any hardware or operating system. SQL Server 2000 has also established the performance record on the Windows platform for the SAP R/3 Sales and Distribution (SD) standard application benchmark. All of this study proves that SQL Server 2000 is really considered one best product in this field and it can outperform Oracle 8, its famous competitor.

While SQL Server 2000 is designed to take advantage of recent and future innovations in high-end SMP hardware platforms, SQL Server 2000 will also deliver an initial installment of Microsoft's vision for software scale-out, the ability to distribute database workload across a cluster of commodity servers. Rather than replacing old systems with bigger, more expensive ones, software scale-out achieves scalability increases simply by adding additional commodity servers to a database cluster.

SQL Server 2000 introduces Distributed Partitioned Views, a feature that provides e-commerce customers with unlimited scalability by dividing workload across multiple, independent SQL Server-based servers.

Supporting high-speed analysis of Web-scale datasets, SQL Server 2000 Analysis

Services allows multidimensional queries against dimensions with hundreds of millions of members. SQL Server 2000 also introduces indexed views, a technology enabling high-performance reporting applications against relational databases.

2.4.3 Data Abstraction Mechanism

A DBMS is a collection of interrelated files and a set of programs that allows users to access and modify these files. A major purpose of database system is to provide usres with an abstract view of the data. That is, the system hides certain details of how the data is stored and maintained. However, in order for the system to be usable, data must be retrieved efficiently. This concern has lead to the design of complex data structures for the represantation of data in the database.

Physical Level: The lowest level of abstraction describe show the data are actually tored.
Conceptual Level: The next higher-level of abstraction describes what data are actually stored in database, and the relationships that exists among data. This level is used by database administrators who must decide what information to be kept in the database.
View Level: The highest level of abstraction describes only the part of the entire database. Many usres of the database system will not be concerned with all of this information. Instead, such usres need only a part of the database. To simplify their interaction with the system, the view level of abstraction is defined.

2.4.4 Data Modeling

Underlying the structure of a database is the concept of a data model, a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints. The various data models that have been proposed fall into three groups; object-based logical models, record-based logical models, and physical data models.

1. Object-Based Logical Models: These models are used in describing data at the conceptual and view levels. They are characterized by the fact they provide fairly flexible structuring capabilities and allow data constraints to be specified explicitly. There are many different models and more are likely to come.

2. The Entity-Relationship Model: This model is based on perception of a real world which consists of a collection of basic objects called entities , and relationships among these objects. An entity is an object that is distinguishable from other objects by a specific set of attributes. A relationship is an association among several entities.

3. The Object-Oriented Model: This model also is based on a collection of objects. An object contains values stored in instance variables within the object. Those values are themselves objects. Thus, objects can contain objects to an arbitrarely deep level of nesting. An object also contains bodies of code that operate on the object. These bodies are called methods.

4. Record-Based Logical Models: These are used in describing data at the conceptual and view levels. They are used both to specify the overall logical structure of the database and to provide a higher-level description of the implementation. Record-based models are so named because the database is structured in fixed-format records of several types. Each record type defines a fixed number of fields, or attributes, and each file is usually of a fixed length.

Record-based models do not include a mechanism for the direct representation of code in the database. Instead, there are seperate languages that are associated within the model to express database queries and updates.

The three most widely accepted data models are the relational, network and hierarchical models.

1. **RELATIONAL MODEL:** This represents data and relationships among data by a collection of tables, each of which has a number of columns with unique names.
2. **NETWORK MODEL:** Data in the network model are represented by collections of records and relationships among data are represented by links , which can be viewed as pointers.
3. **HIEARCHICAL MODEL:** This model is similar to the network model in the sense that data and relationships among data are represented by records and links , respectively. It differs from network model in that the records are

organized as collections of trees rather than orbitrary graphs.

5. Physical Data Models : Physical data models are used to describe data at the lowest level. In contrast to logical data models , there are very few physical data models in use. Two of widely known ones are :

2.5 UNIFYING MODEL

2.5.1 FRAME MEMORY

2.5.1.1 INSTANCES AND SCHEMES

Databases change over time as information is inserted and deleted. The collection of information stored in the database at a particular moment in time is called an instance of the database. The overall design of the database is called the database scheme. Schemes are changed infrequently.

The concept of a database scheme correspondes to the programming language notion of type definition. A variable of a given type has a particular value at a given instant in time. Thus this concept of the value of a given variable in programming language correspondes to the concept of an instance of a database scheme.

Database systems have several schemes, partitioned according to the levels of abstraction. At the lowest level is physical scheme; at the intermediate level, the concetual scheme; at the highest level, a subscheme. In general database systems support one physical scheme, one conceptual scheme, and several subschemes.

2.5.1.1.1 Data Dependence

The ability to modify a scheme definitionin one level without affecting a scheme definition in the next higher-level is called data independence. There are two levels:

1. **Pysical Data Independence** : This is the the ability to modify the physical scheme without causing application programs to be rewritten. Modifications at the physical level are occasionally necessary in order to improve performance.

2. **Logical Data Independence** : This is the ability to modify the conceptual

scheme without causing application programs to be rewritten. Modifications at the conceptual level are necessary whenever the logical structure of the database is altered.

Logical data independence is more difficult to achieve than physical data independence since application programs are heavily dependent on the logical structure of the data they access.

The concept of data independence is similar in many repects to the concept of abstract data types in modern programming languages. Both hide implementation details from the users.

2.5.1.1.2 Data Definition Language

A database scheme is specified by a setof definitons which are expressed by a special language called a data definiton language (DDL). The result of compilation of DDL statements is a set of tables which are stored in a special file called dictionary. A data directory is a file that contains metadata ; that is "data about data".

The storage structure and access methods used by the database system are specified by a set of definitions in a special type of DDL called a data storage and definition language. The result of compilation of these definitions is a set of instructions to specify the implementation details of the database schemes which are usually hidden from the users.

Data Manipulation Language

By data manipulation these are meant:

1. The retrieval of information stored in the database .The insertion of information stored in the database.
2. The deletion of information from the database.
3. The modification of data stored in the database.

A data manipulation language (DML) is a language that enables users to access or manipulate data as organizedby the appropriate data model. There are basically two types:

1. Procedural DMLs require a user to specify what data is needed and how to get it.

2. Nonprocedural DMLs require a user to specify what data is needed without specifying how to get it.

Nonprocedural DMLs are usully easier to learn and use than procedural DMLs. However , since a user does not have to specify how to get the data , these languages may generate code which is not as efficient as that produced by procedural languages.

A query is statement requesting the retrieval of information. The portion of a DML that involves information retrieval is called a query language. Although technically incorrect, it is common practise to use the terms query language and data manipulation language synonymously.

2.5.1.1.3 Database Manager

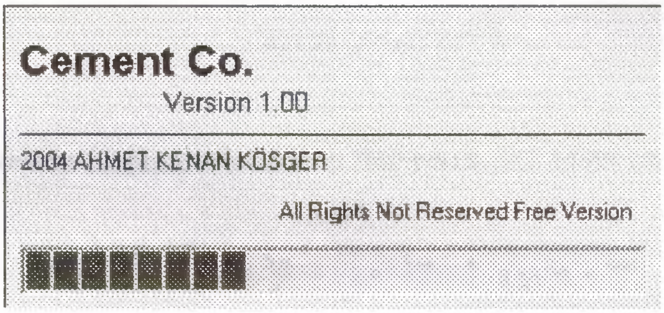
Databases typically require a large amount of storage space, may be terrabytes of data. Since the main memory of computers cannot store this information, it is stored in disks. Data is moved between disk storage and main memry as needed. Since, the movement of data to and from disk is slow relative to the speed of the CPU, it is imperative that the database system structure the data so as to minimize the need to move data between disk and main memory.

The goal of a database system is to simplify and facilitate access to data. A database manager is program module which provides the interface between the low level data stored in the database and the application programs and queries submitted to the system. The database manager is responsible for the following tasks:

Chapter III: Program Design

3.1 SCREEN OUTPUT

Program loading form:



Cement Co.

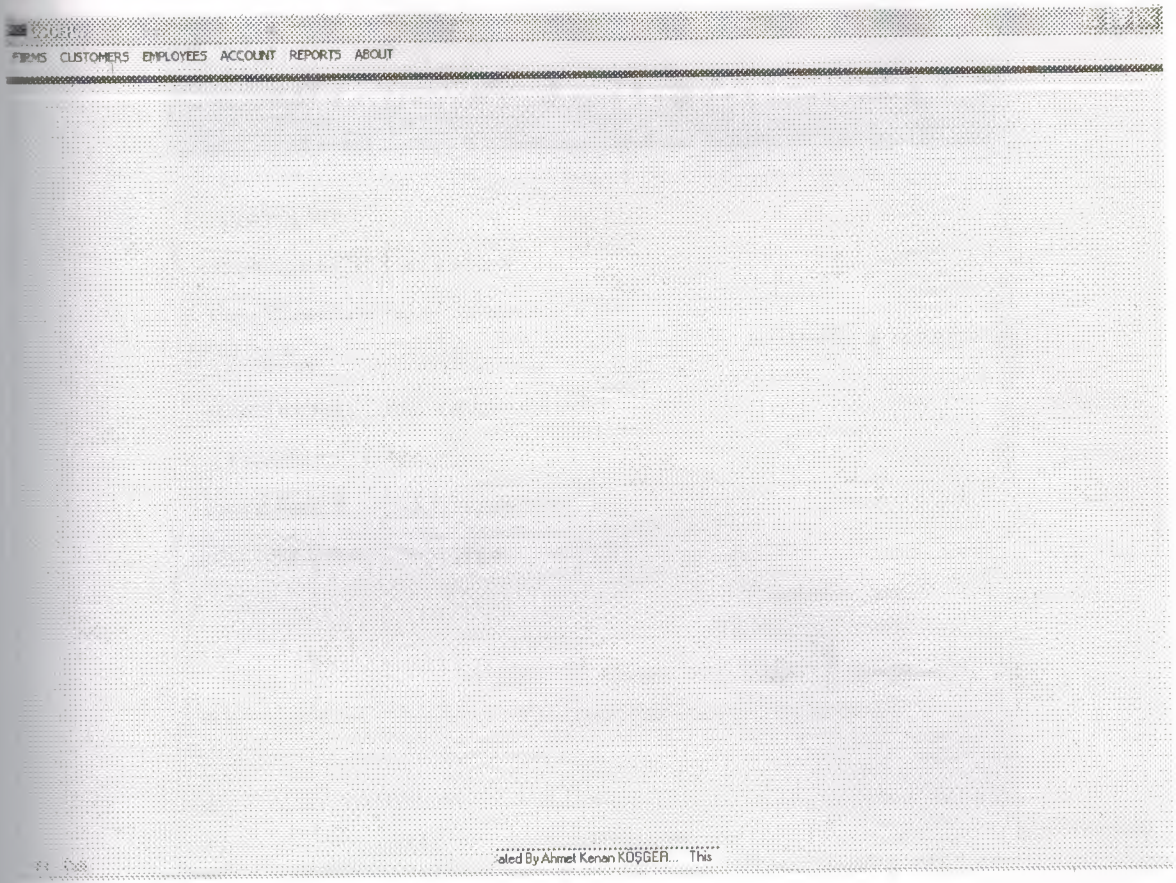
Version 1.00

2004 AHMET KENAN KÖSGER

All Rights Not Reserved Free Version

Figure 3.1: This screen output gives information about program before run the program.

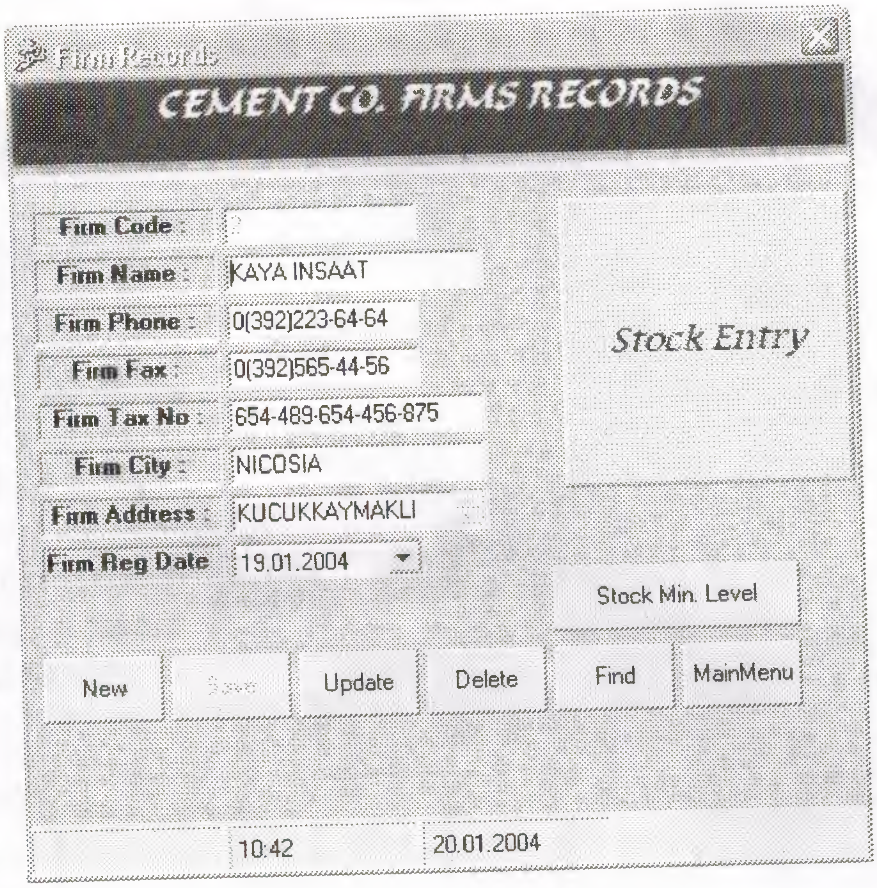
Program MainMenu:



ated By Ahmet Kenan KOŞGER... This

Figure 3. 2: This screen output is MainMenu that you select a process to start transaction entry.

Program Firm Record:



Firm Code : 2

Firm Name : KAYA INSAAT

Firm Phone : 0(392)223-64-64

Firm Fax : 0(392)565-44-56

Firm Tax No : 654-489-654-456-875

Firm City : NICOSIA

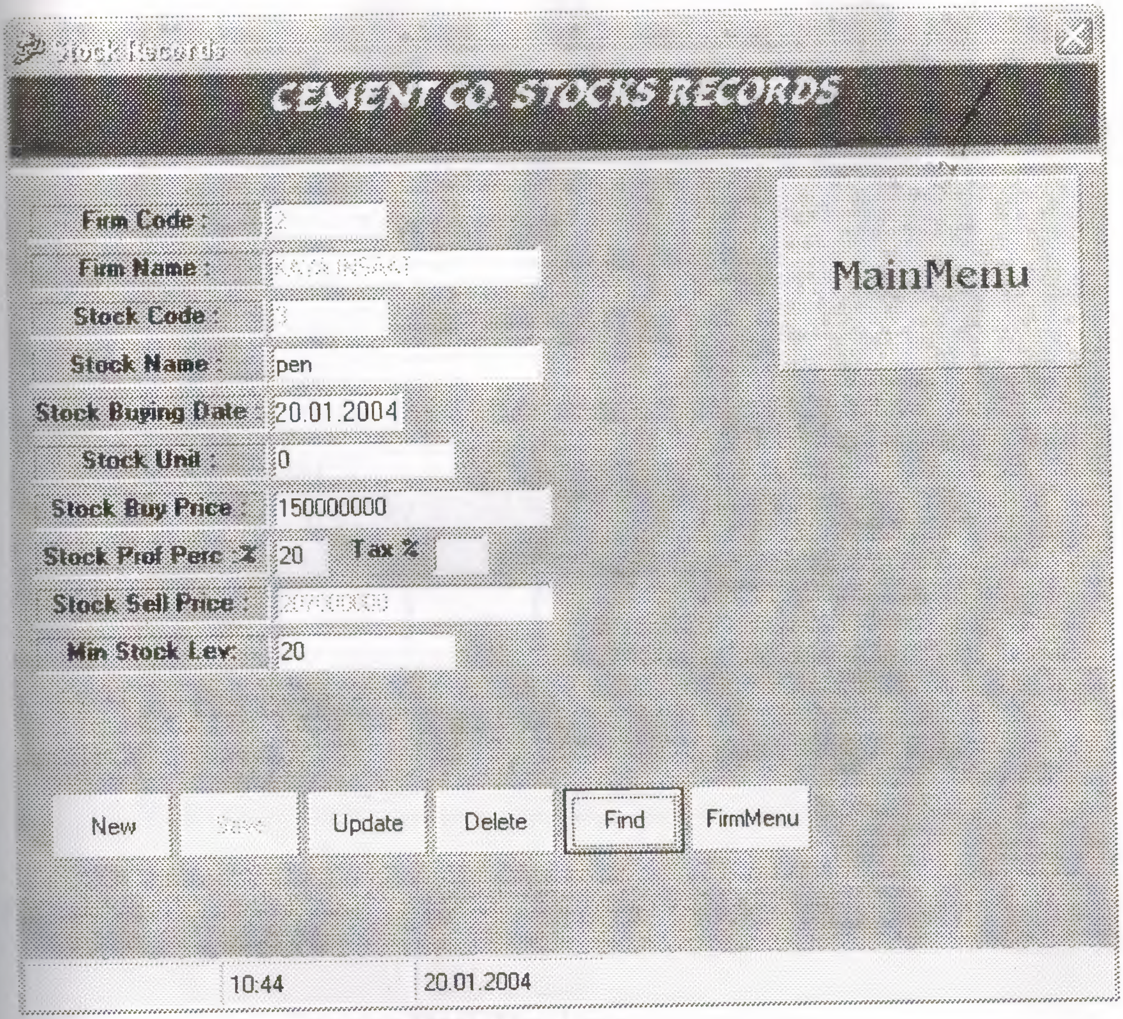
Firm Address : KUCUKKAYMAKLI

Firm Reg Date 19.01.2004

10:42 20.01.2004

Figure 3.3: User enters the information about firm and user will enter stock information.

Program Stock Records:



Stock Records

CEMENT CO. STOCKS RECORDS

Firm Code : 2

Firm Name : KAYA INSAAT

Stock Code : 3

Stock Name : pen

Stock Buying Date : 20.01.2004

Stock Unit : 0

Stock Buy Price : 150000000

Stock Prof Perc :% 20 Tax %

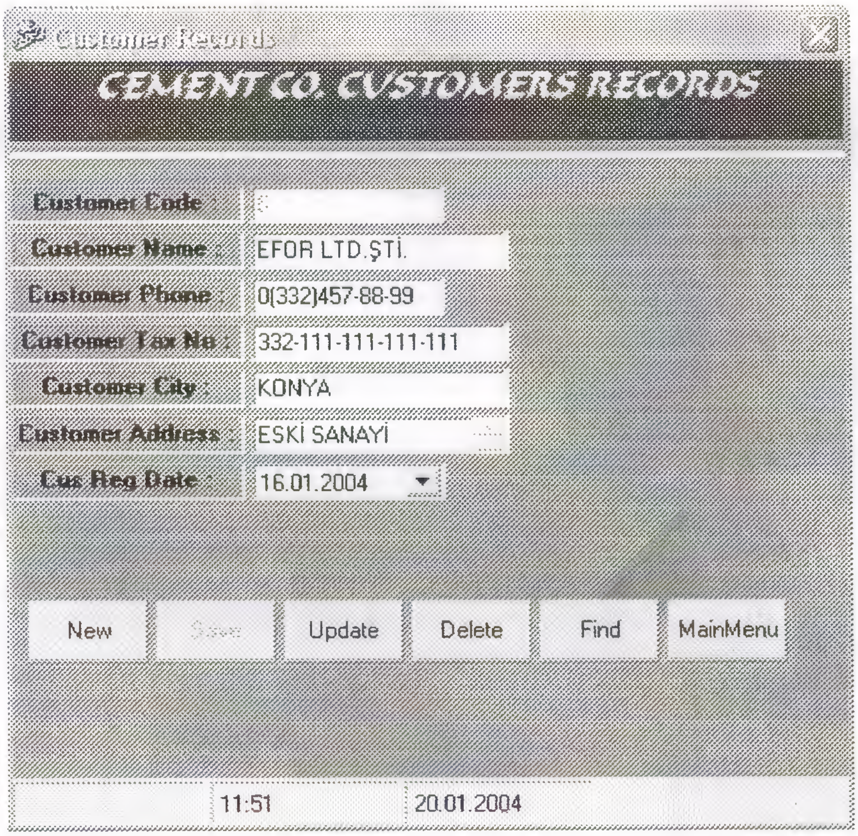
Stock Sell Price : 207000000

Min Stock Lev: 20

10:44 20.01.2004

Figure 3. 4: User entered the stock detail and saved the detail information to stock database.

Program Customer Records:

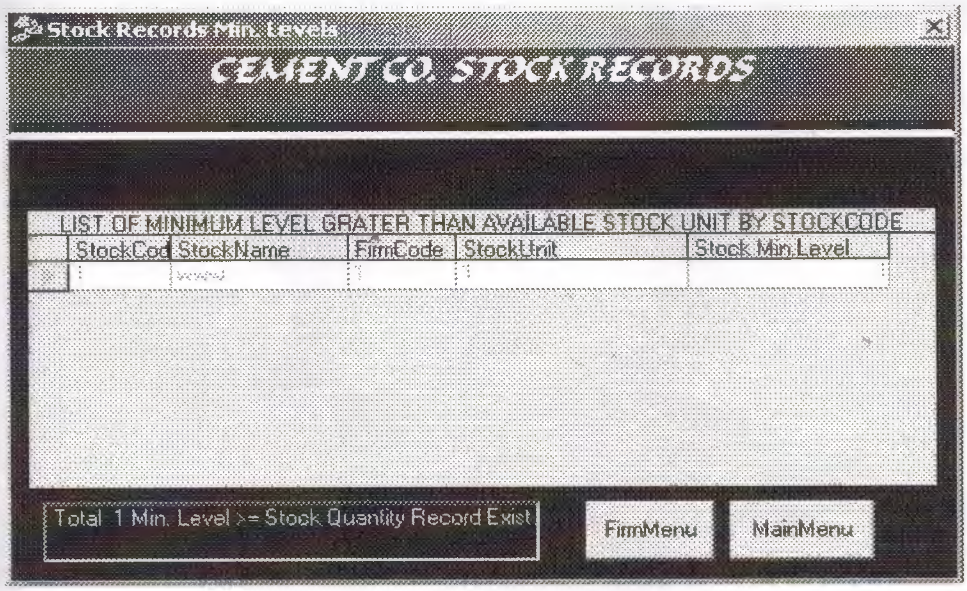


Customer Code :
Customer Name : EFOR LTD.ŞTİ.
Customer Phone : 0(332)457-88-99
Customer Tax No : 332-111-111-111-111
Customer City : KONYA
Customer Address : ESKİ SANAYİ
Cus Reg Date : 16.01.2004

11:51 20.01.2004

Figure 3. 5: User will enter the Customer detail and who will sell a product to customer.

Program Stock Min. Level Screen Output:



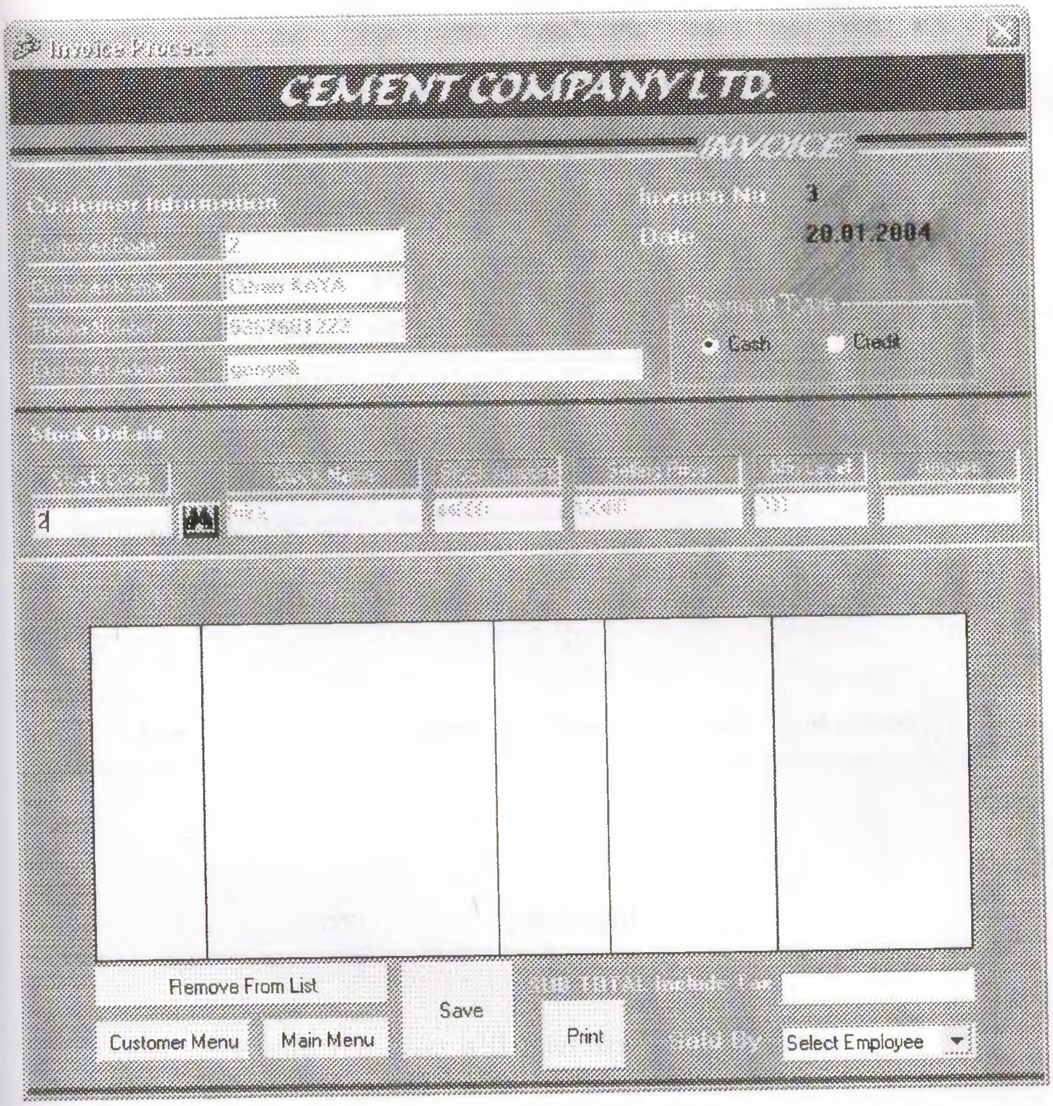
LIST OF MINIMUM LEVEL GRATER THAN AVAILABLE STOCK UNIT BY STOCKCODE

StockCod	StockName	FirmCode	StockUnit	Stock Min Level

Total 1 Min. Level >= Stock Quantity Record Exist

Figure 3. 6: This screen shows Stock Min. Level to new stock Order.

Program Invoice Process:



Invoice No 3

Date 20.01.2004

Customer Code 2

Customer Name Orhan KAYA

Phone Number 9357681222

Customer Address gonyeli

Payment Type ● Cash ○ Credit

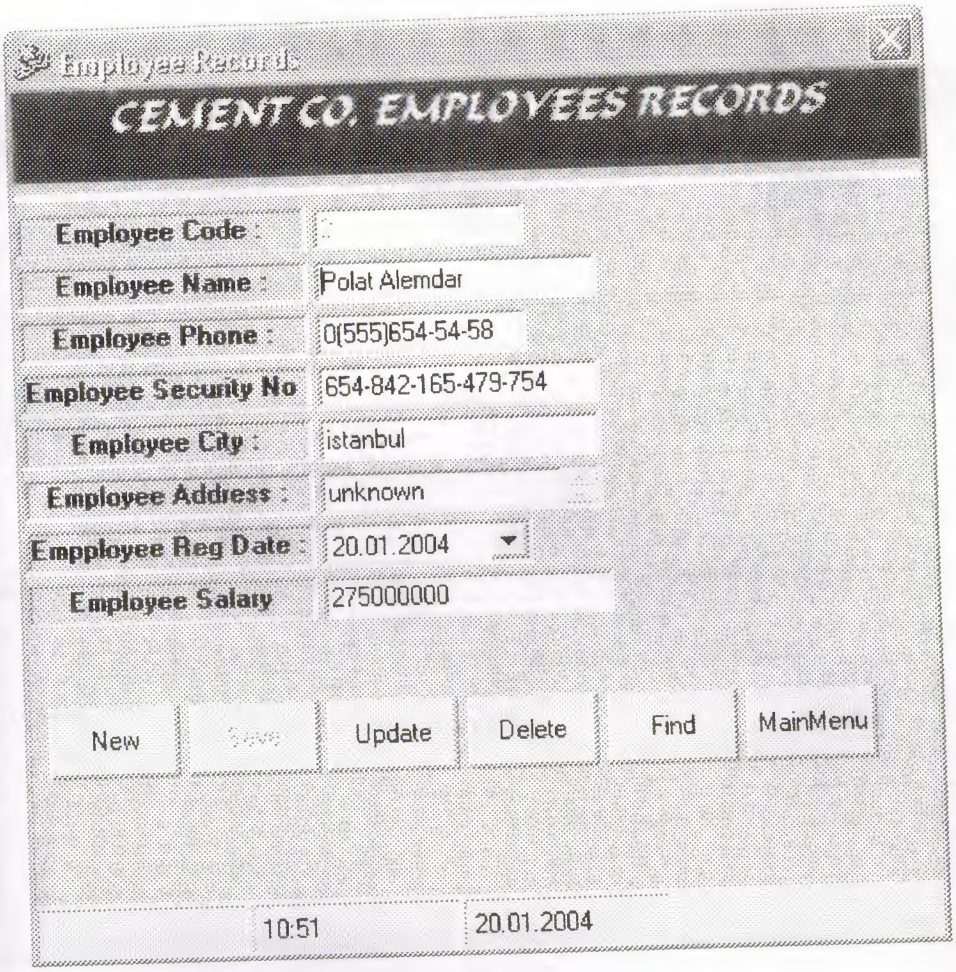
Stock Code 2

SUB TOTAL Include Tax

Sold By Select Employee

Figure 3. 7: User will sell product to customer.

Program Employee Records:

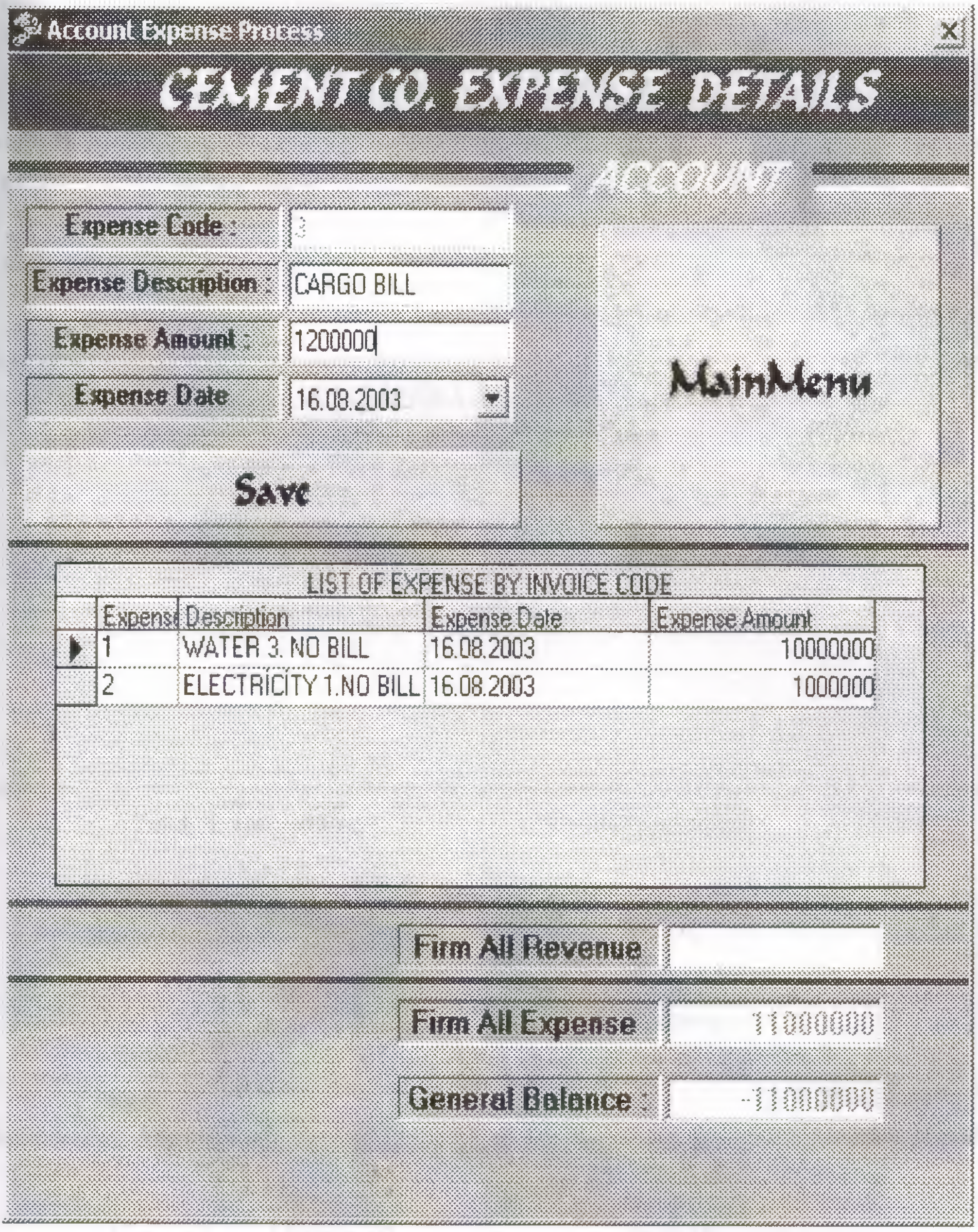


Employee Code :	
Employee Name :	Polat Alemdar
Employee Phone :	0(555)654-54-58
Employee Security No	654-842-165-479-754
Employee City :	istanbul
Employee Address :	unknown
Empployee Reg Date :	20.01.2004
Employee Salary	275000000

10:51 20.01.2004

Figure 3. 8: User will enter the employee detail.

Program Expense Process:



Expense Code : 3

Expense Description : CARGO BILL

Expense Amount : 1200000

Expense Date 16.08.2003

LIST OF EXPENSE BY INVOICE CODE

Expens	Description	Expense Date	Expense Amount
1	WATER 3. NO BILL	16.08.2003	10000000
2	ELECTRICITY 1.NO BILL	16.08.2003	1000000

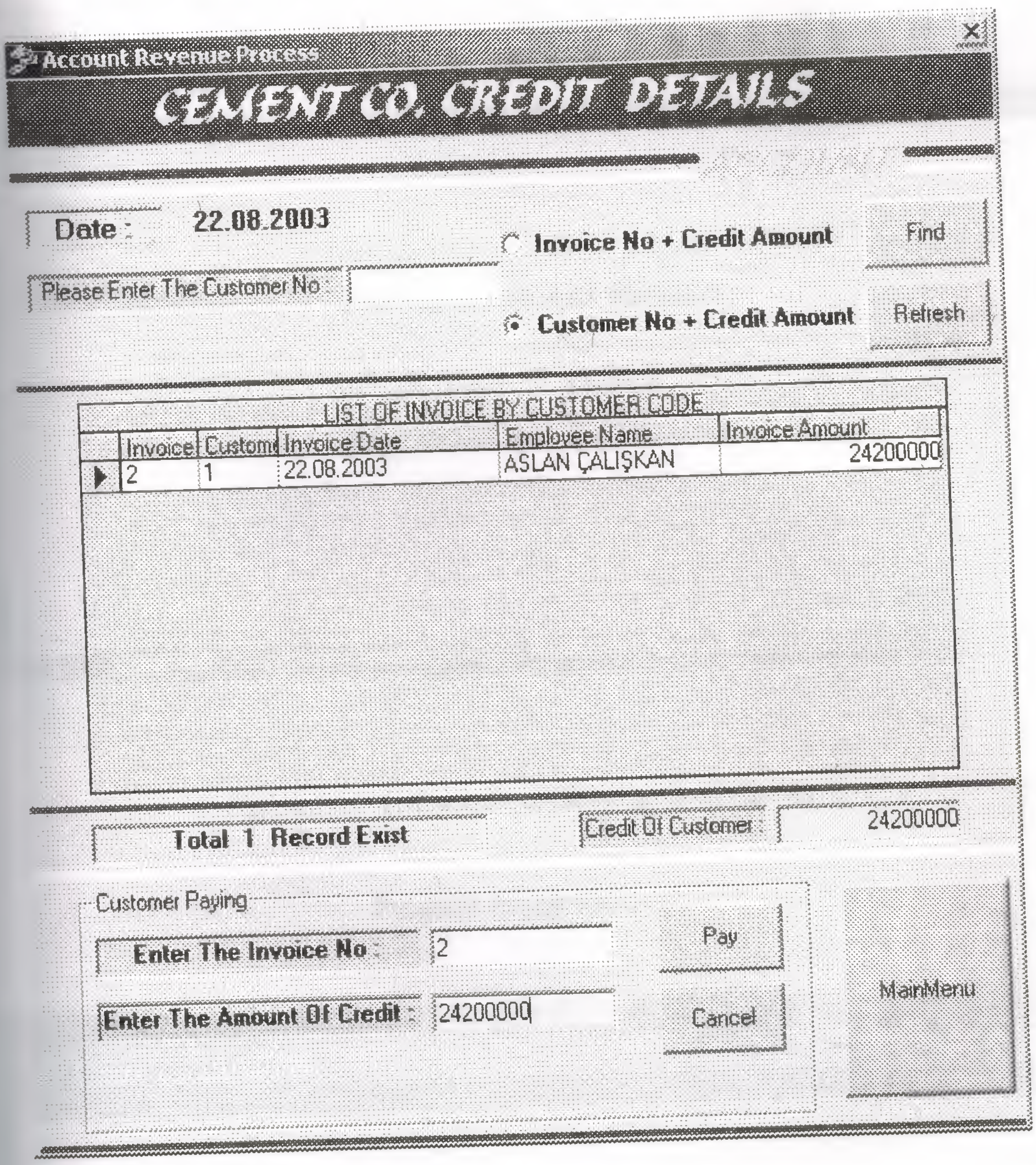
Firm All Revenue

Firm All Expense 11000000

General Balance : -11000000

Figure 3. 9: Users will entry expenses transaction to account database

Program Account Revenue Process:



Date : 22.08.2003

Please Enter The Customer No :

○ Invoice No + Credit Amount

● Customer No + Credit Amount

LIST OF INVOICE BY CUSTOMER CODE

	Invoice	Custome	Invoice Date	Employee Name	Invoice Amount
▶	2	1	22.08.2003	ASLAN ÇALIŞKAN	24200000

Total 1 Record Exist

Credit Of Customer : 24200000

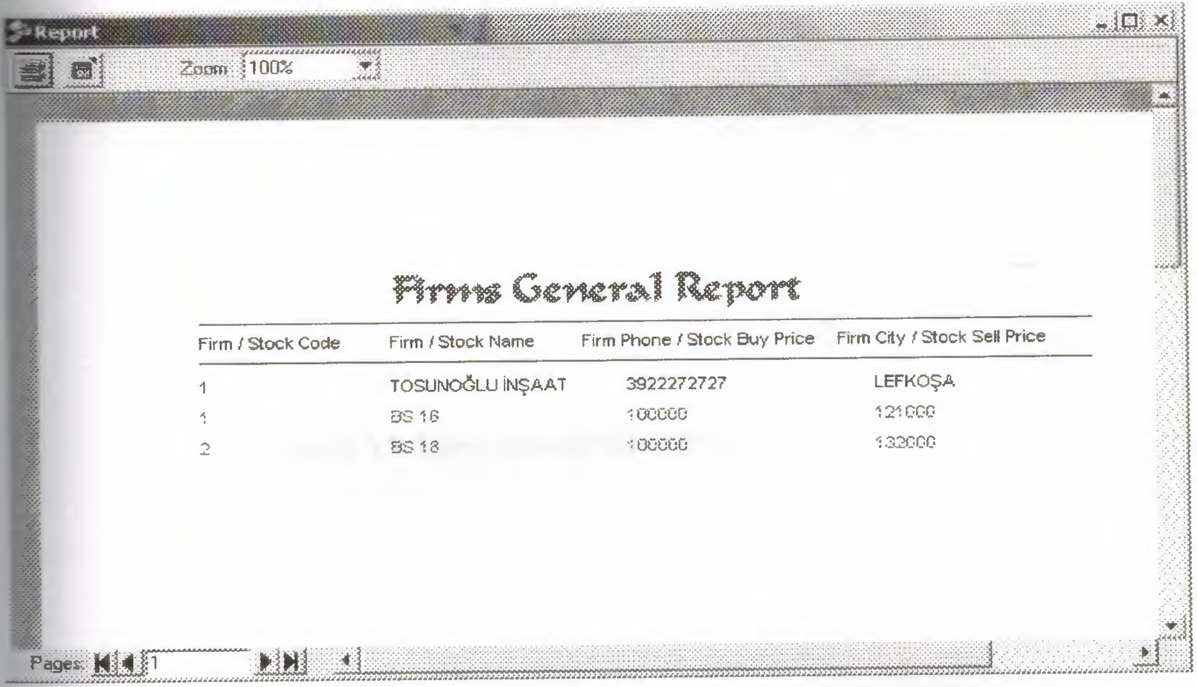
Customer Paying

Enter The Invoice No : 2

Enter The Amount Of Credit : 24200000

Figure 3.10: Users will collect to invoice from the customers.

Program Report Process:

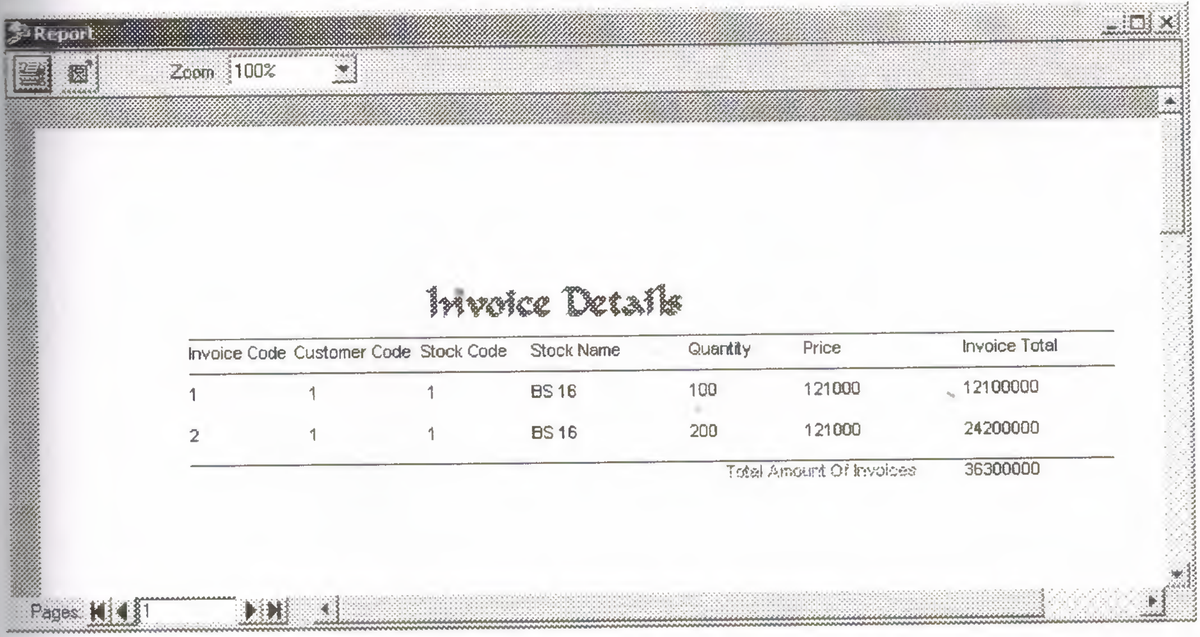


Firms General Report

Firm / Stock Code	Firm / Stock Name	Firm Phone / Stock Buy Price	Firm City / Stock Sell Price
1	TOSUNOĞLU İNŞAAT	3922272727	LEFKOŞA
1	BS 16	100000	121000
2	BS 18	100000	132000

Figure 3.11: Users will see firms general reports.

Program Report Process:

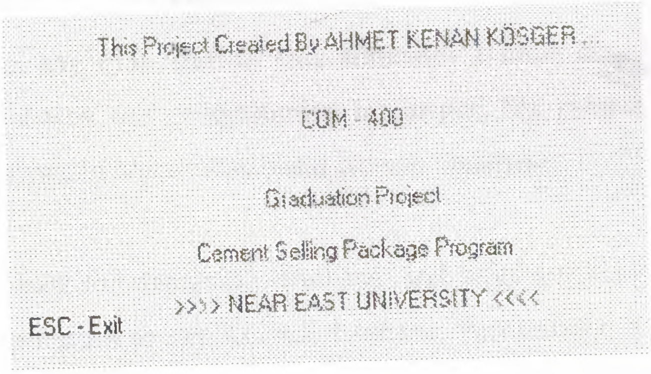


Invoice Details

Invoice Code	Customer Code	Stock Code	Stock Name	Quantity	Price	Invoice Total
1	1	1	BS 16	100	121000	12100000
2	1	1	BS 16	200	121000	24200000
					Total Amount Of Invoices	36300000

Figure 3.12: Users will see customers invoice details.

Program About:



This Project Created By AHMET KENAN KÖSGER ...

COM - 400

Graduation Project

Cement Selling Package Program

>>>> NEAR EAST UNIVERSITY <<<<

ESC - Exit

Figure 3.13: Users will see who prepared project.

Summary and Conclusion

In conclusion, I can say that, during this semester I have acquired a considerable understanding of databases, and programming languages. My project involved particular emphasis on the workings of Visual Basic and Access Database.

The work done involved the designing database, and huge software in Visual Basic as well as linking my program to the Access database. Particularly, I learned that how is the design is more important than the output and after designing, the job become only implementing what we designed, So, paying sometimes for the design at the beginning is useful at the end. In the process I was able to formulate a more seasoned view of the advantages, and disadvantages of these applications.

In fact, I believe that this is not the best program to do such a task but learning the mistakes now will develop me more and more to achieve and become higher and higher.

Finally, I have come to appreciate the value of teamwork and the academic significance of real life situations.

References

DAŞDEMİR, Yaşar, " VERİTABANLARI & SQL", 1. Edition, Türkmen Publishing House, İstanbul, 2002.

KARAGÜLLE, İhsan – PALA, Zeydin, "Microsoft VİSUAL BASİC 6.0 PRO", 2. Edition, Türkmen Publishing House, İstanbul, 2003.

ÇUBUKÇU, Faruk, "SQL SERVER 7.0", 2. Edition, ALFA Publishing House, İstanbul, 2000.

Websites:

1. www.msdn.microsoft.com.
2. www.vbworld.com.
3. www.microsoft.com.
4. Henry F. KORTH, and Abraham SİLBERSCHATZ. Database System Concepts. Second Edition. McGrew-Hill, Inc.
5. www.vbasic.bilgisi.com\codes.
6. ww.programlama.com.
7. ww.solucan.net\vbasic\downloads

Appendices

Appendix A: DATABASE DESIGN

Firms.Tbl

Field Name	Type	Size
Firmname	Text	50
Firmcode	Integer	2
Firmphone	Text	20
Firmfax	Text	20
Firmtaxno	Text	20
Firmcity	Text	20
Firmadres	Text	50
Firmregdate	Date	8

Stocks.Tbl

Field Name	Type	Size
Firmname	Text	50
Firmcode	Integer	2
Stockcode	Integer	2
Stockname	Text	20
stockminl	Integer	2
Stockbd	Date	8
Stockunit	Double	8
Stockbp	Double	8
Stockpperc	Double	8
Stocksellp	Text	50

Cus.Tbl

Field Name	Type	Size
Cuscode	Integer	2
Cusname	Text	30
Cusphone	Text	20
Custaxno	Text	20
Cuscity	Text	20
Cusadres	Text	50
Cusrd	Date	8

Emp.Tbl

Field Name	Type	Size
Empcode	Integer	2
Empname	Text	20
Empphone	Text	20
Empsc	Text	20
Empcity	Text	20
Empadres	Text	50
Emprd	Date	8
Empsal	Double	8

Invoice1.Tbl

Field Name	Type	Size
Incode	Integer	2
Cuscode	Integer	2
Stockcode	Integer	2
Sname	20	Text
Quan	Integer	2
Uprice	Double	8
Totp	Double	8

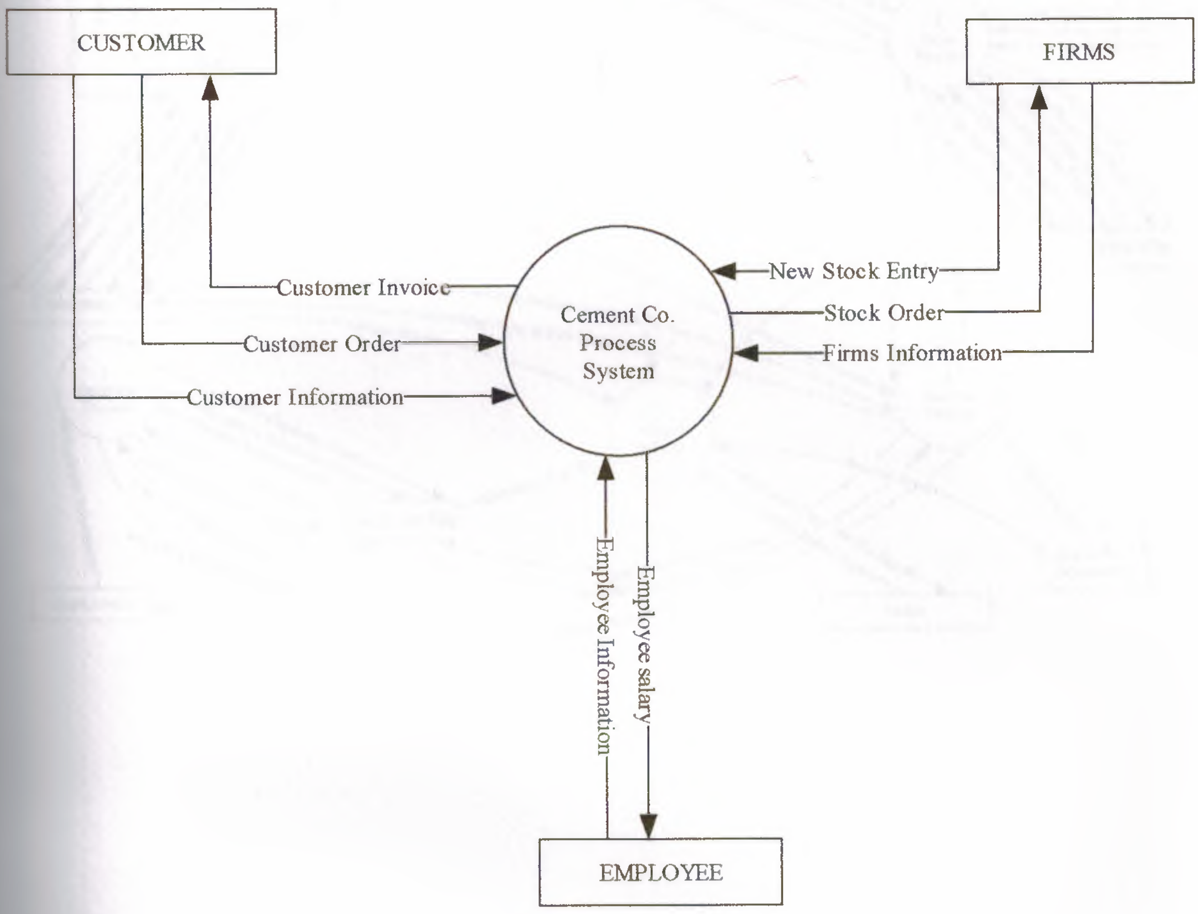
Invoice2.Tbl

Field Name	Type	Size
Incode	Integer	2
Cuscode	Integer	2
Subtot	Double	8
Invdate	Date	8
Empname	Text	20

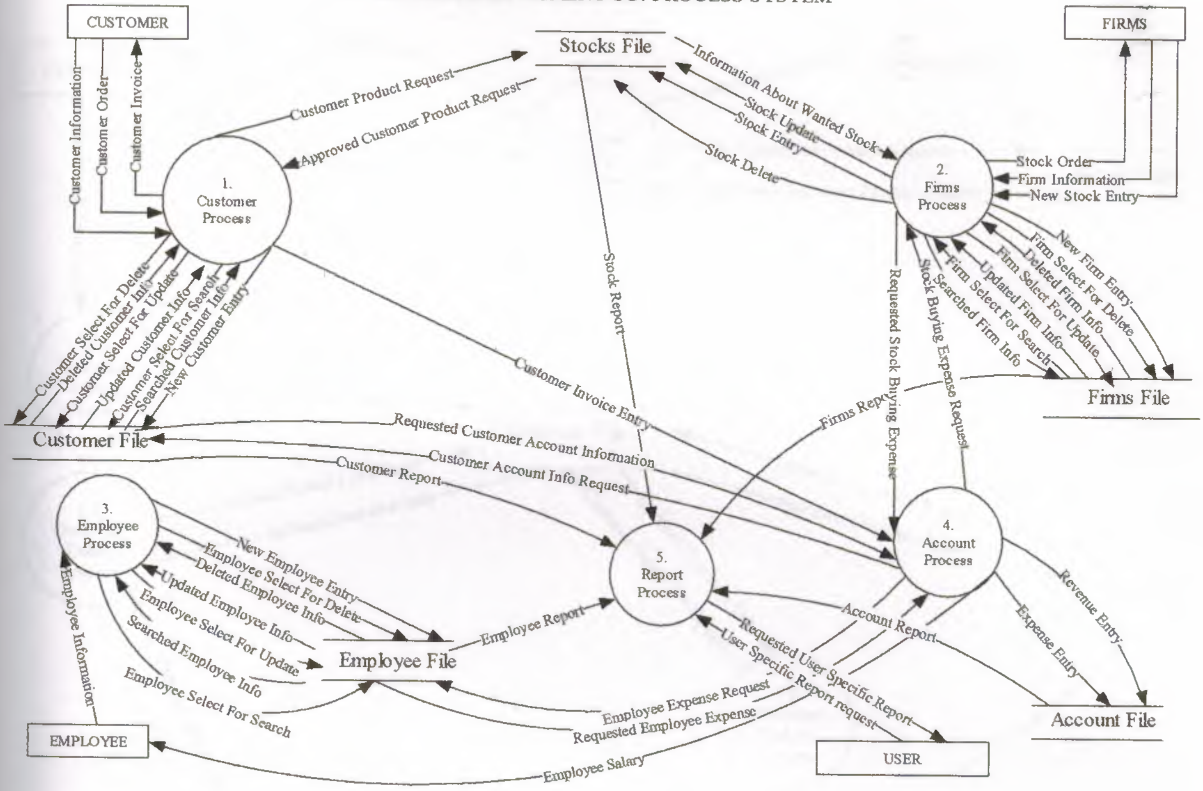
Account.Tbl

Field Name	Type	Size
Accdate	Date	8
Expense	Double	8
Revenue	Double	8
Description	Text	50
Empcode	Integer	2

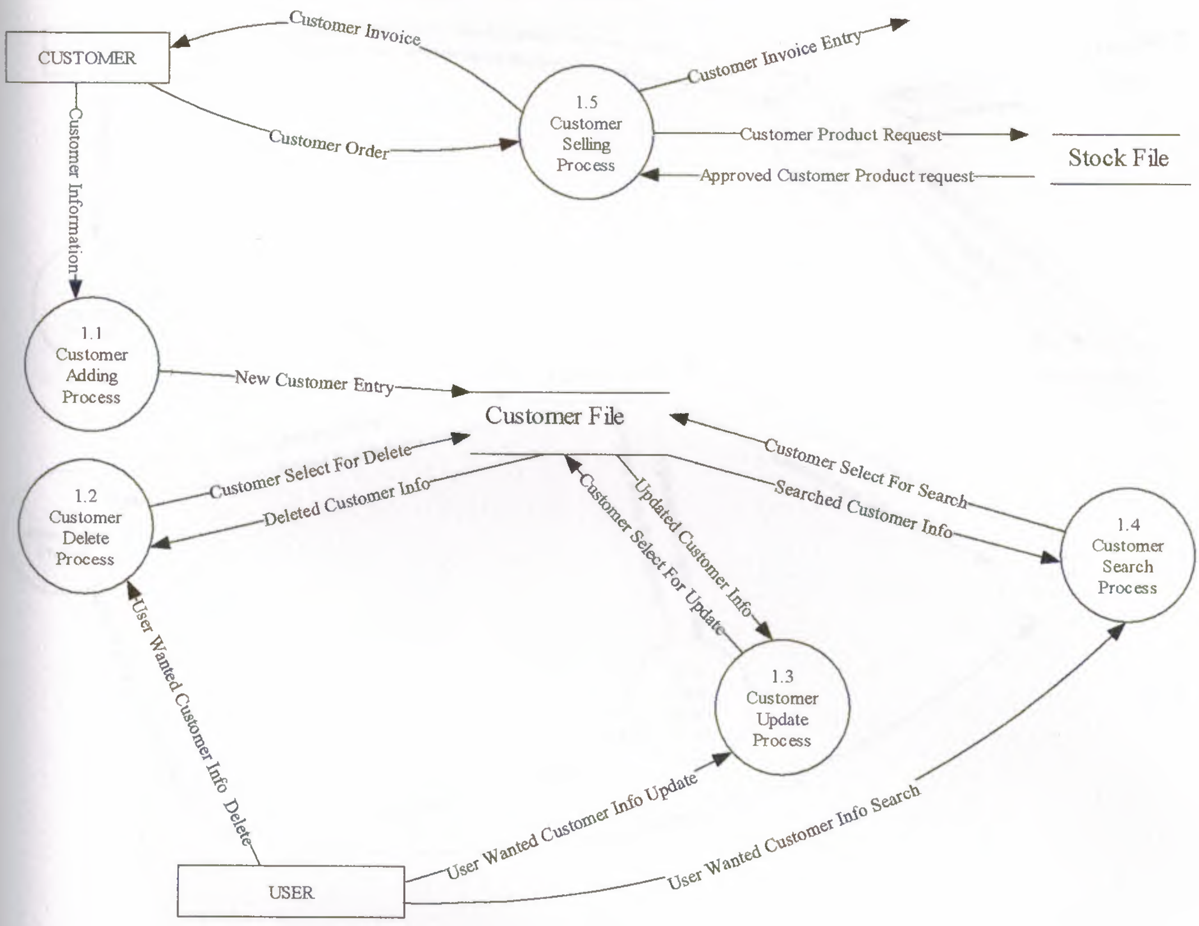
CONTEXT DFD OF CEMENT CO. PROCESS SYSTEM



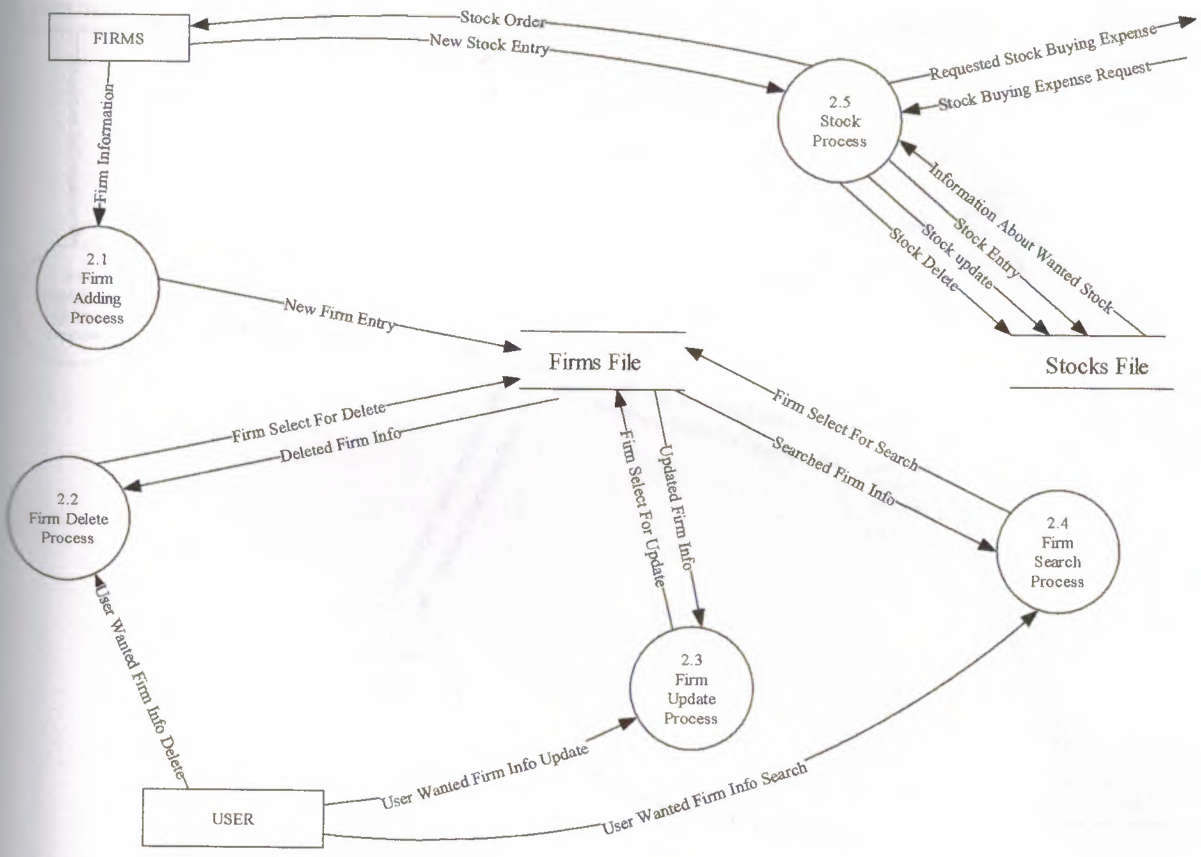
TOP LEVEL DFD OF CEMENT CO. PROCESS SYSTEM



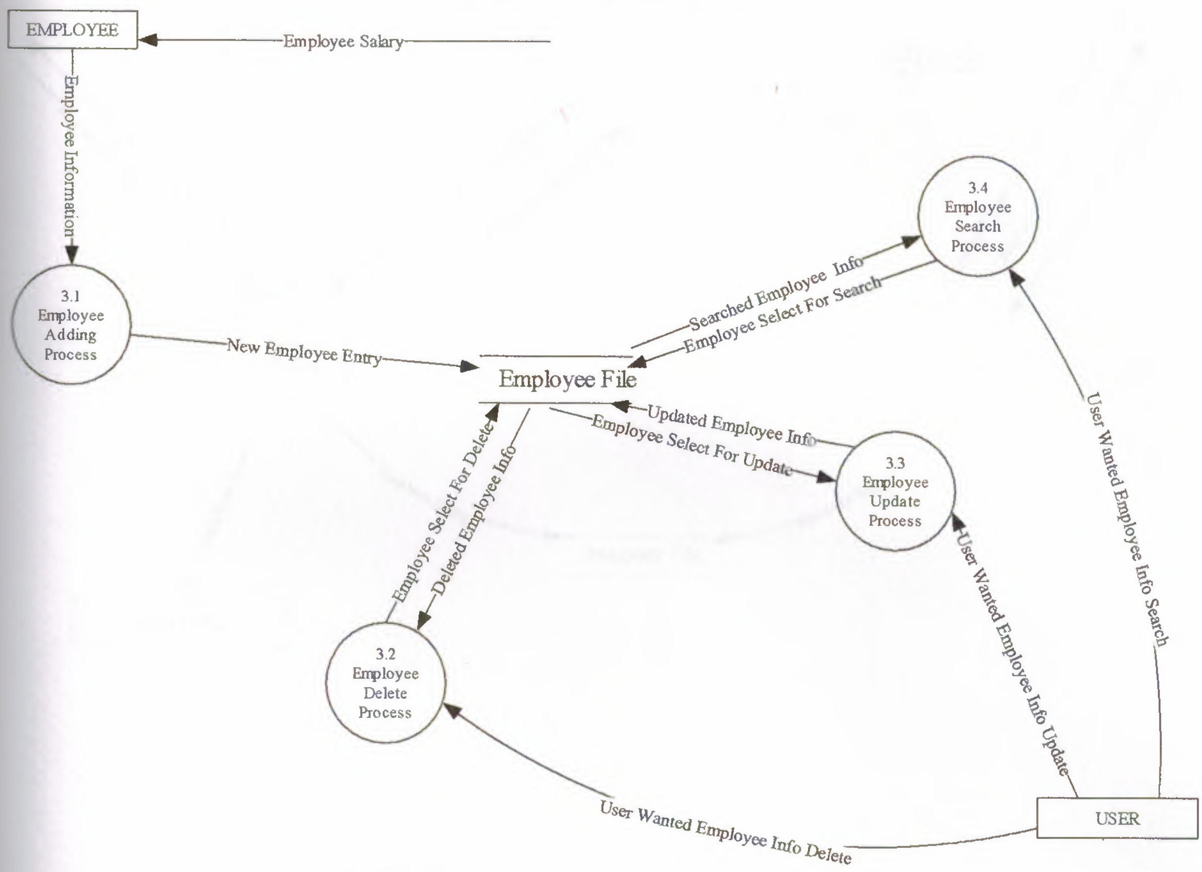
DETAILED DFD OF CUSTOMER PROCESS (1)


CUSTOMER
Customer Invoice
Customer Order
Customer Information
1.5 Customer Selling Process
Customer Invoice Entry
Customer Product Request
Approved Customer Product request
Stock File
1.1 Customer Adding Process
New Customer Entry
Customer File
1.2 Customer Delete Process
Customer Select For Delete
Deleted Customer Info
Customer Select For Update
Updated Customer Info
Customer Select For Search
Searched Customer Info
1.4 Customer Search Process
1.3 Customer Update Process
User Wanted Customer Info Delete
User Wanted Customer Info Update
User Wanted Customer Info Search
USER

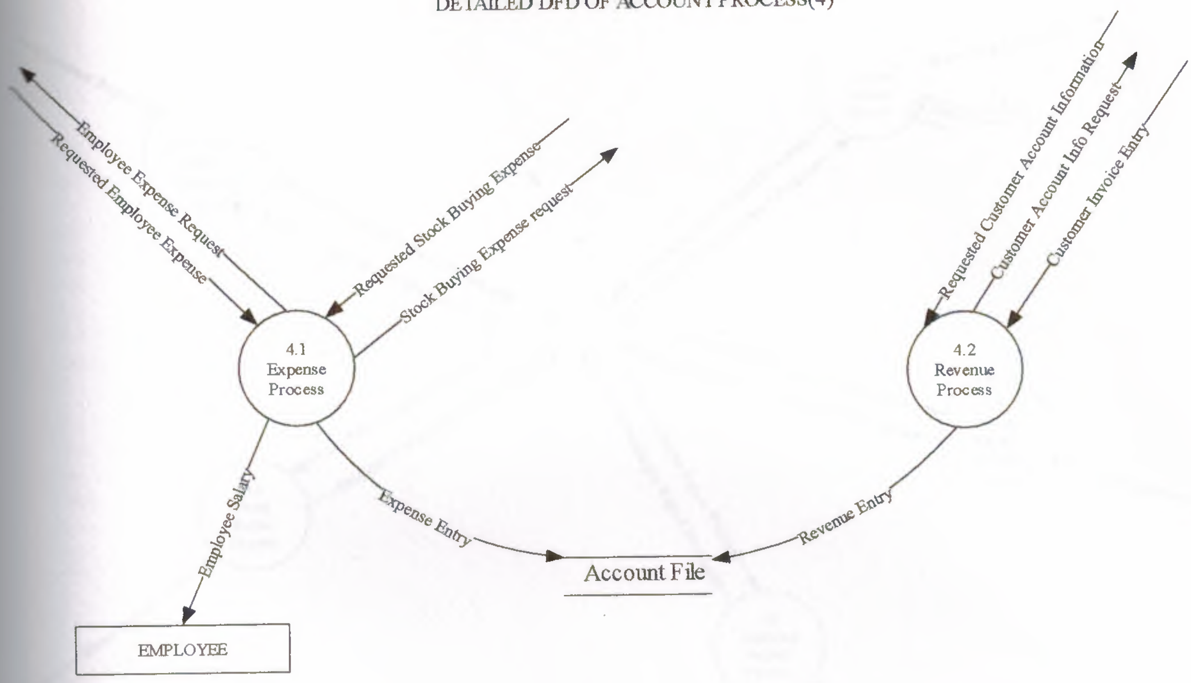
DETAILED DFD OF FIRMS PROCESS (2)



DETAILED DFD OF EMPLOYEE PROCESS(3)

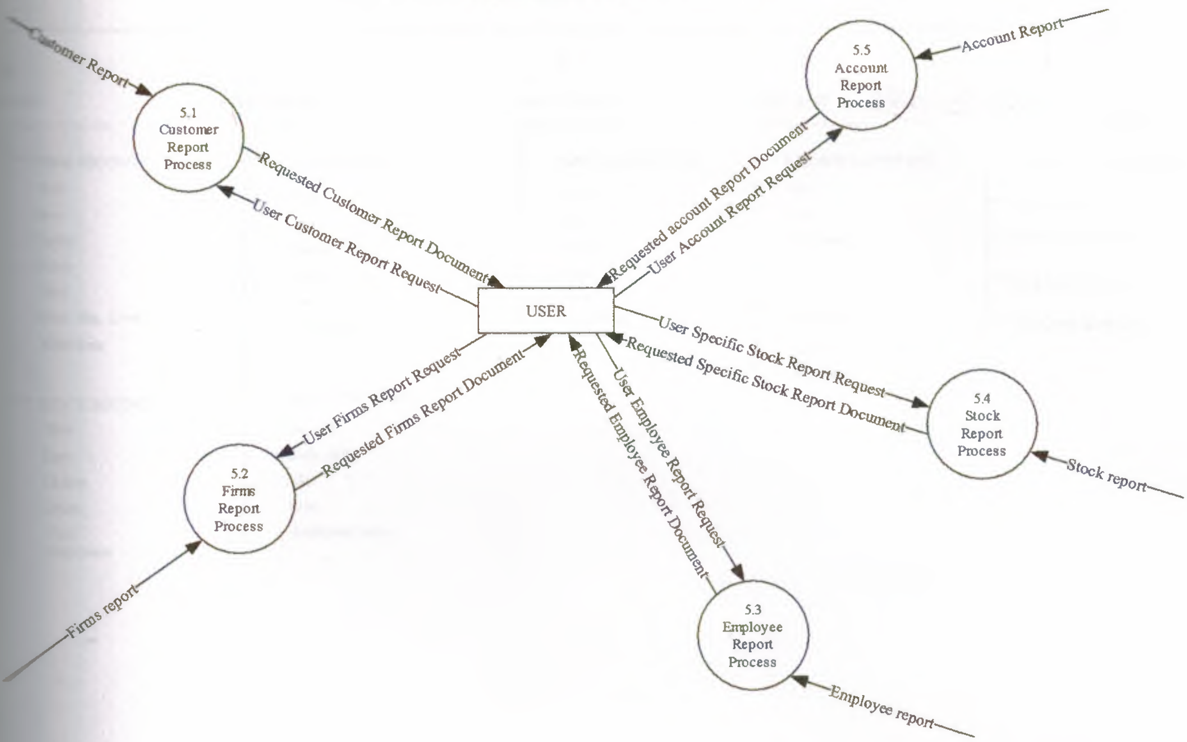

EMPLOYEE
Employee Salary
Employee Information
3.1 Employee Adding Process
New Employee Entry
Employee File
3.4 Employee Search Process
Searched Employee Info
Employee Select For Search
Updated Employee Info
Employee Select For Update
3.3 Employee Update Process
Employee Select For Delete
Deleted Employee Info
3.2 Employee Delete Process
User Wanted Employee Info Search
User Wanted Employee Info Update
User Wanted Employee Info Delete
USER

DETAILED DFD OF ACCOUNT PROCESS(4)

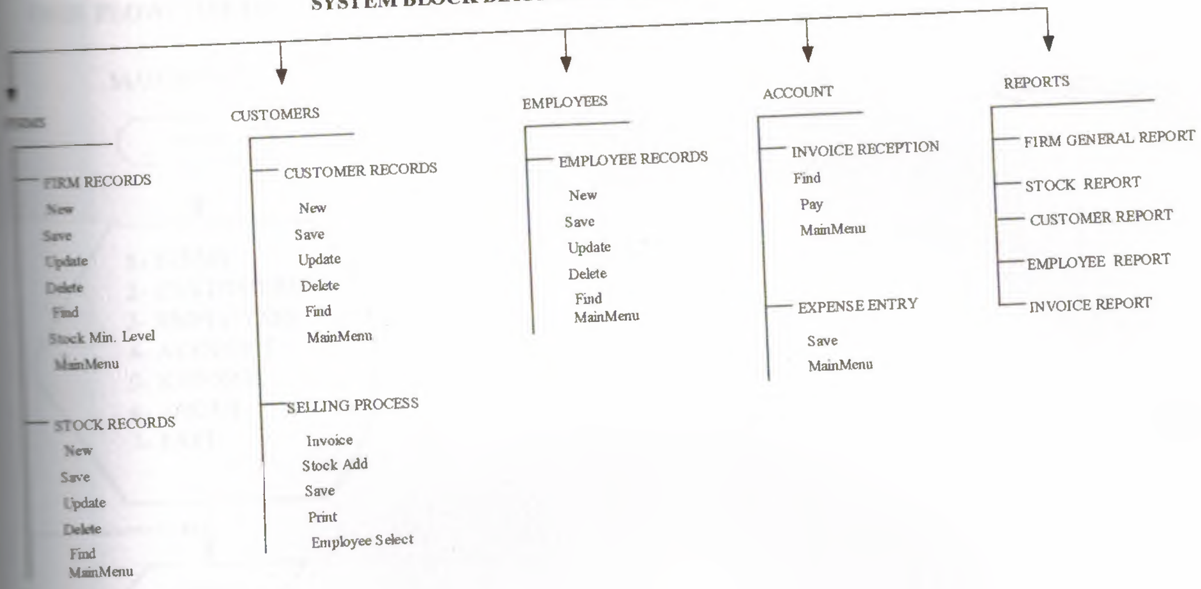


Employee Expense Request
Requested Employee Expense
Requested Stock Buying Expense
Stock Buying Expense request
4.1
Expense
Process
Employee Salary
EMPLOYEE
Expense Entry
Account File
Revenue Entry
4.2
Revenue
Process
Requested Customer Account Information
Customer Account Info Request
Customer Invoice Entry

DETAILED DFD OF REPORT PROCESS(5)

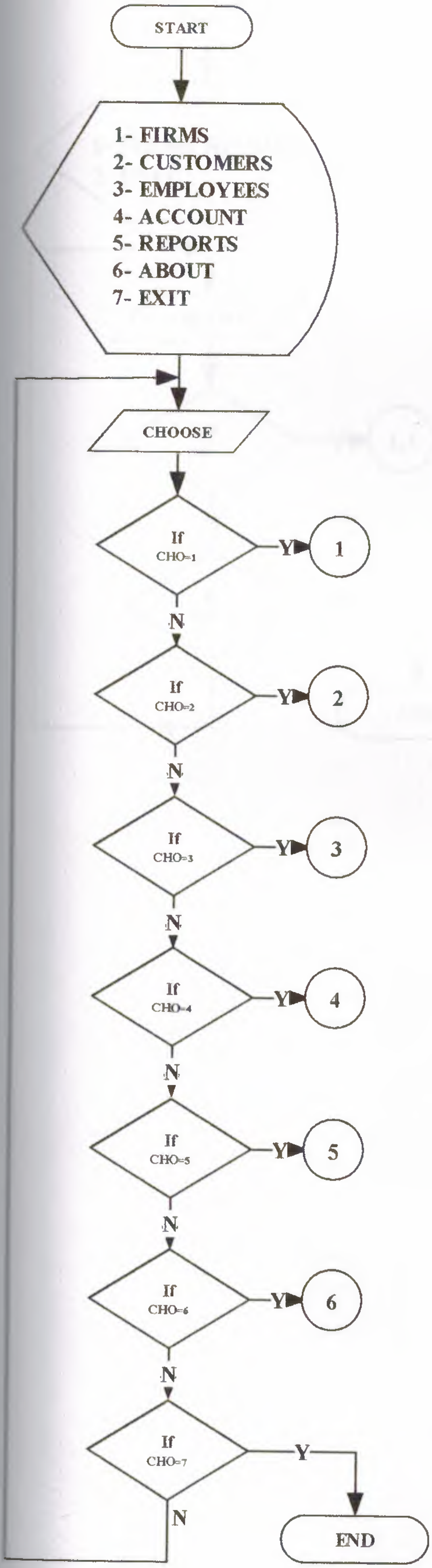

Customer Report
5.1 Customer Report Process
Requested Customer Report Document
User Customer Report Request
5.5 Account Report Process
Account Report
Requested account Report Document
User Account Report Request
USER
User Specific Stock Report Request
Requested Specific Stock Report Document
5.4 Stock Report Process
Stock report
User Firms Report Request
Requested Firms Report Document
5.2 Firms Report Process
Firms report
Requested Employee Report Document
User Employee Report Request
5.3 Employee Report Process
Employee report

SYSTEM BLOCK DIAGRAM OF CEMENT CO.

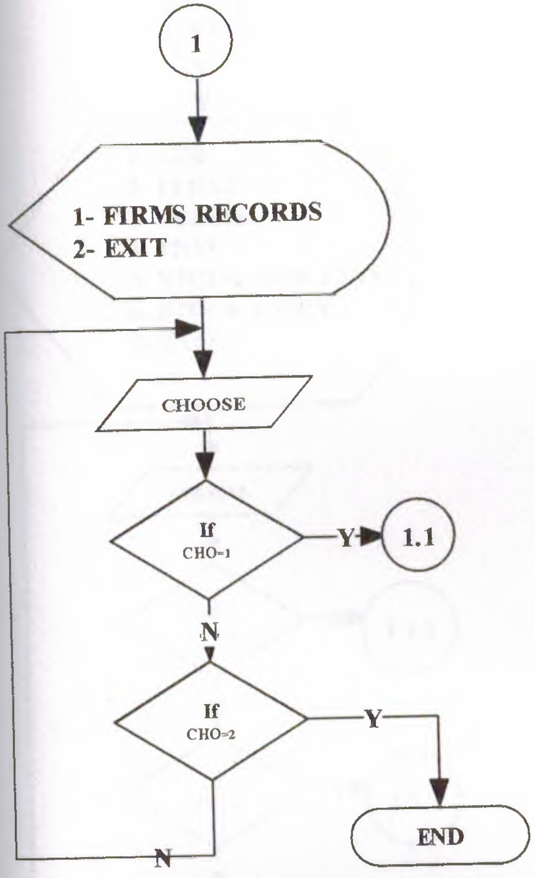


USER FLOWCHARTS

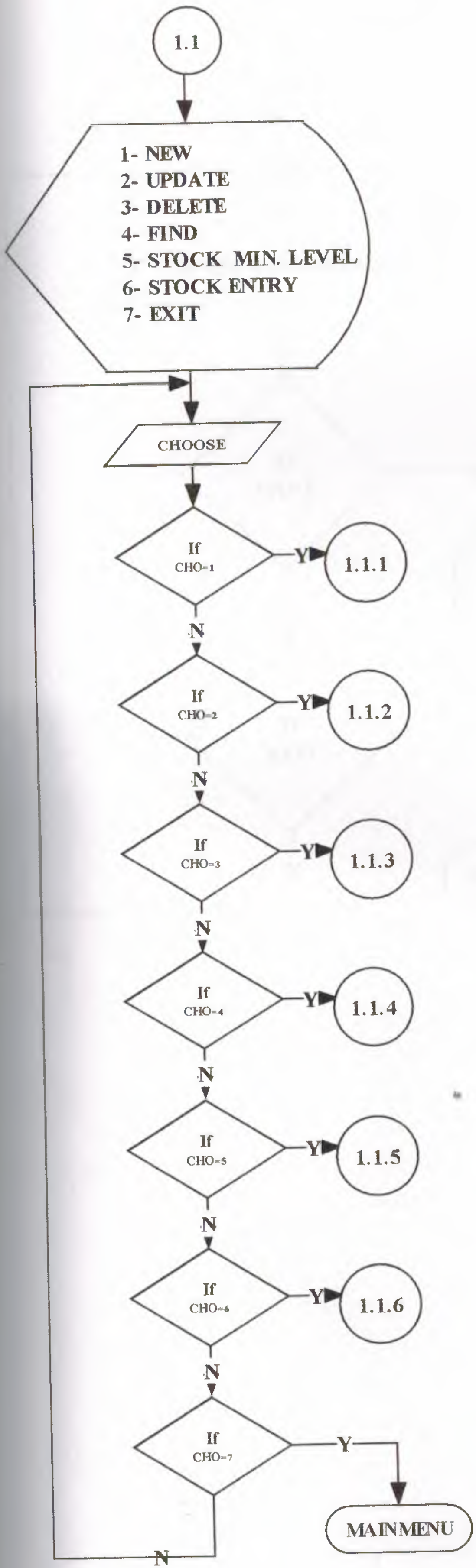
MAINMENU



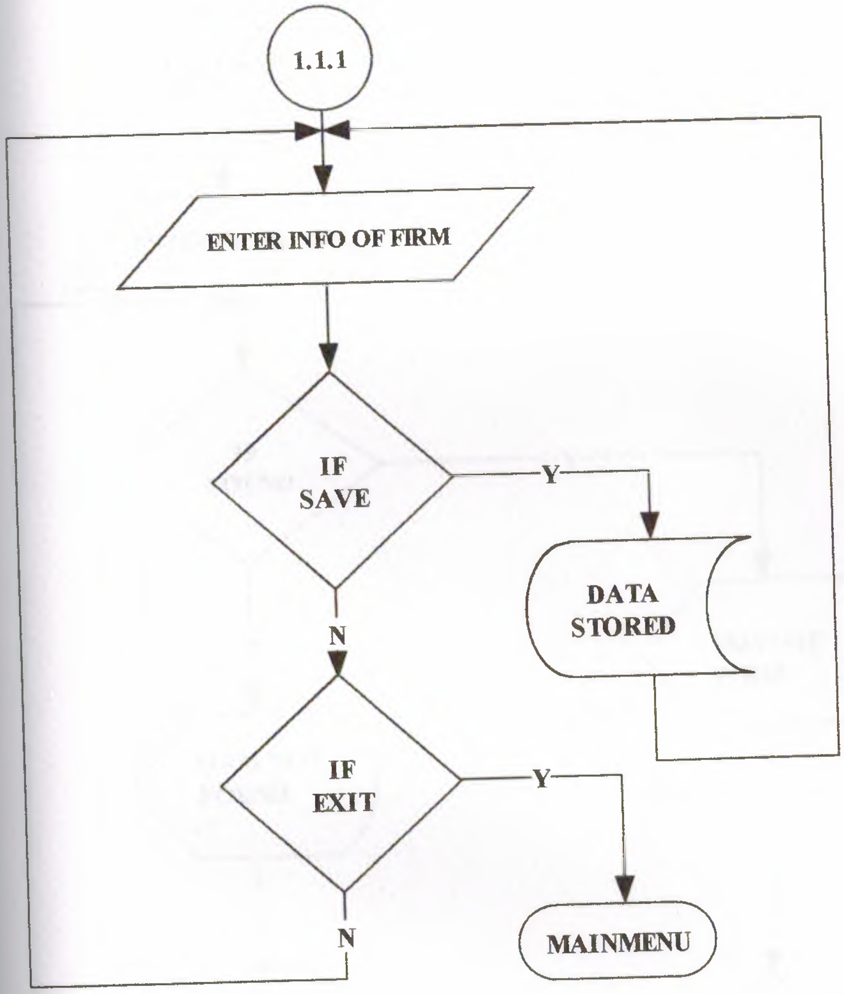
FIRMS



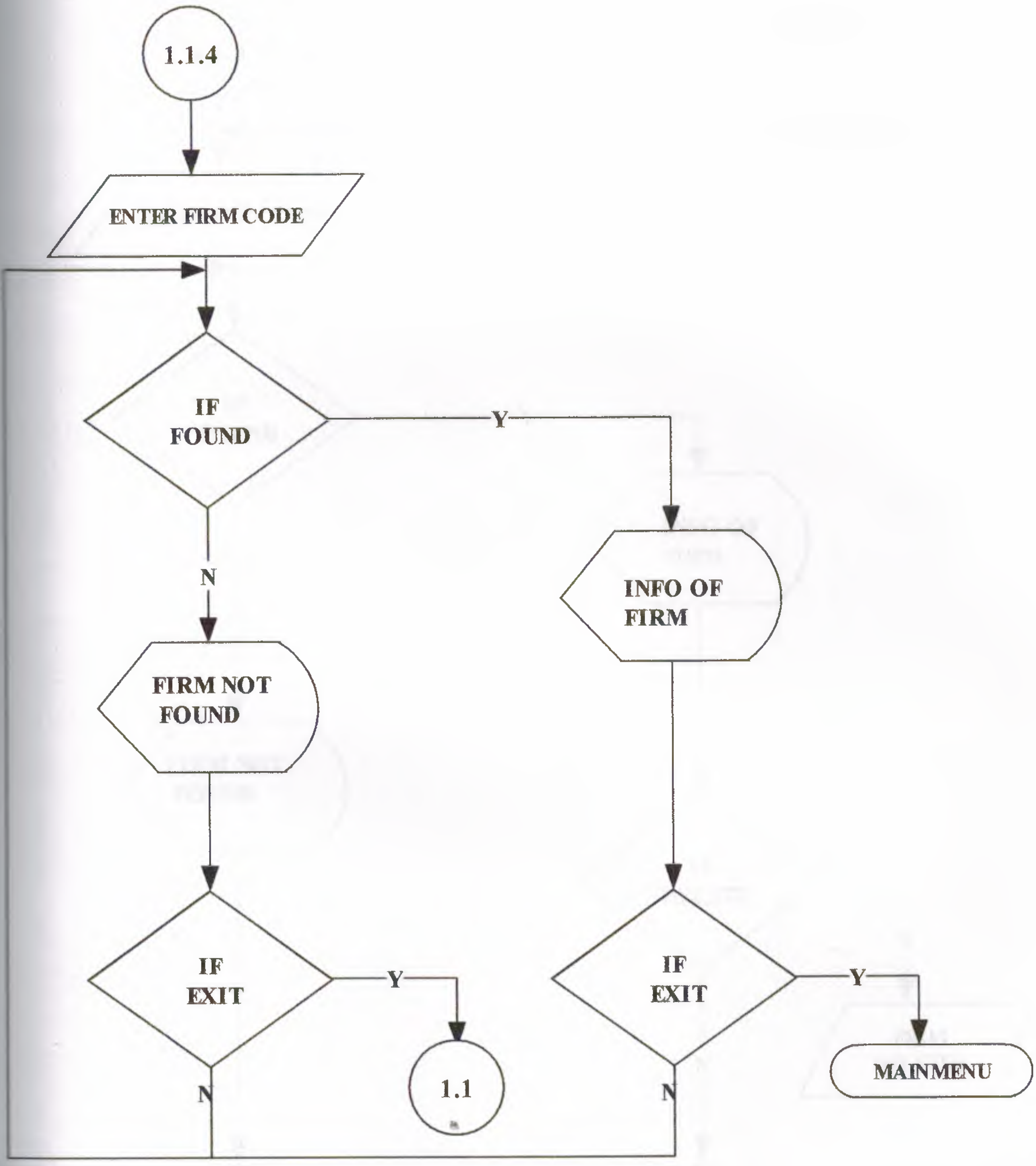
FIRM RECORDS



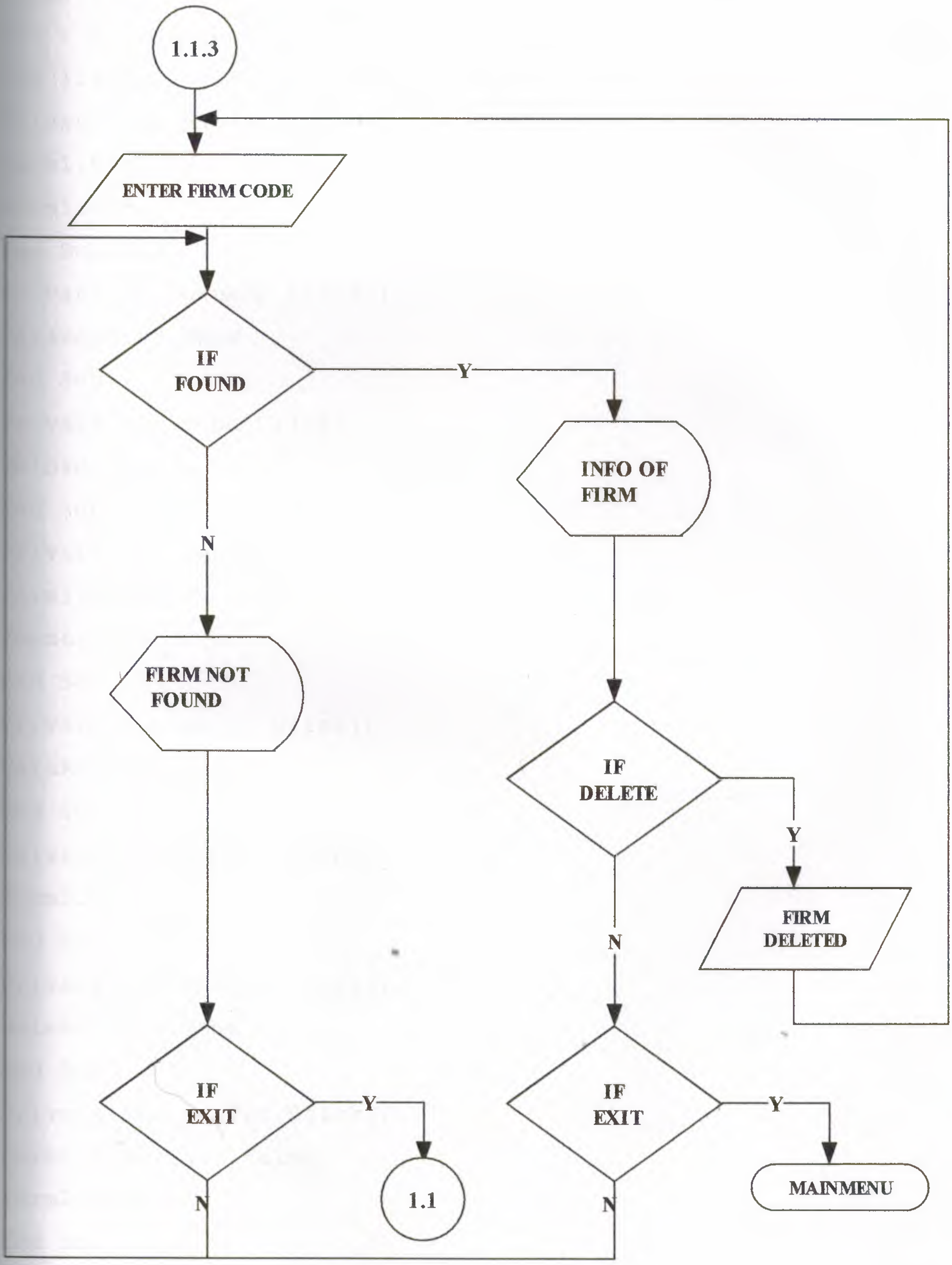
NEW FIRM ENTRY



FIRM FIND


1.1.4
ENTER FIRM CODE
IF FOUND
Y
N
INFO OF FIRM
FIRM NOT FOUND
IF EXIT
Y
N
1.1
IF EXIT
Y
N
MAINMENU

FIRM DELETE


1.1.3
ENTER FIRM CODE
IF FOUND
Y
N
INFO OF FIRM
FIRM NOT FOUND
IF DELETE
Y
N
FIRM DELETED
IF EXIT
Y
N
1.1
IF EXIT
Y
N
MAINMENU

Appendix B: Source Program

```
Form1.Frm
Dim i As Integer
Dim ilkharf, yazi As String
Private Sub mnucr_Click()
form1.Enabled = False
Form5.Show
End Sub
Private Sub mnucrp_Click()
DataReport3.Show
End Sub
Private Sub mnue_Click()
Unload Me
End Sub
Private Sub mnuemp_Click()
form1.Enabled = False
Form6.Show
End Sub
Private Sub mnuer_Click()
DataReport4.Show
End Sub
Private Sub mnuexp_Click()
Form9.Show
End Sub
Private Sub mnufr1_Click()
DataReport1.Show
End Sub
Private Sub mnufrr_Click()
form1.Enabled = False
Form2.Show
End Sub
Private Sub mnuid_Click()
frmcus.Show
End Sub
```

```
Private Sub mnuir_Click()
DataReport5.Show
End Sub
Private Sub mnupr_Click()
Form3.Show
End Sub
Private Sub mnurev_Click()
Form8.Show
End Sub
Private Sub mnusp_Click()
DataReport2.Show
End Sub
Private Sub mnuv_Click()
Form11.Show
End Sub
Private Sub Timer1_Timer()
ilkharf = Left(Label1.Caption, 1)
    yazi = Right(Label1.Caption, Len(Label1.Caption) - 1)
    Label1.Caption = yazi + ilkharf
    Dim sString As String
        sString = "This Program Created By Ezgi Ersoy..."
    If Timer1.Tag = 0 Then
        Me.Caption = sString
        Timer1.Tag = 1
    ElseIf Timer1.Tag < Len(sString) Then
        Me.Caption = Right(sString, Len(sString) - Timer1.Tag)
        Timer1.Tag = Timer1.Tag + 1
    ElseIf Timer1.Tag = Len(sString) Then
        Me.Caption = sString
        Timer1.Tag = 0
    End If
End Sub
```

Form2.Frm

```
Option Explicit
Private database As ADODB.Connection
Private ezgi As ADODB.Recordset
Private ersoy As String
Dim p As Panel
Private Sub Command1_Click()
Command7.Visible = False
clear
coun
Command2.Enabled = True
Command3.Enabled = False
Command4.Enabled = False
Text2.SetFocus
End Sub
Private Sub Command2_Click()
Dim sql, ask, ask1
ask = MsgBox("Do You Want To Save That ?", vbInformation +
vbYesNo, "Save")
If ask = vbYes Then
If Text2.text <> "" And MaskEdBox1.text <> "" And
MaskEdBox2.text <> "" And MaskEdBox3.text <> "" And Text3.text
<> "" And Text4.text <> "" Then
sql = "insert into firms(firmname, firmcode, firmphone,
firmfax, firmtaxno, firmcity, firmadres, firmregdate) values("
sql = sql & "'" & Text2.text & "',"
sql = sql & "" & Text1.text & ","
sql = sql & "'" & MaskEdBox1.text & "',"
sql = sql & "'" & MaskEdBox2.text & "',"
sql = sql & "'" & MaskEdBox3.text & "',"
sql = sql & "'" & Text3 & "',"
sql = sql & "'" & Text4 & "',"
sql = sql & "'" & DTPicker1.Value & "')"
database.Execute (sql)
```

```
Dim i As Integer
ProgressBar1.Min = 0
ProgressBar1.Max = 1000
For i = ProgressBar1.Min To ProgressBar1.Max
ProgressBar1.Visible = True
ProgressBar1.Value = i
Next
ProgressBar1.Visible = False
ask1 = MsgBox("Firm Information Save Successful! ", , "Saved")
Command2.Enabled = False
Command7.Visible = True
Command3.Enabled = True
Command4.Enabled = True
Command6.Enabled = True
Else
ask1 = MsgBox("Please Fill The Other Texts!")
Command7.Visible = False
Command3.Enabled = False
Command4.Enabled = False
Command6.Enabled = True
Text2.SetFocus
End If
End If
End Sub
Private Sub Command3_Click()
Dim ask As String
If Text1.text <> "" And Text2.text <> "" And MaskEdBox1.text
<> "" Then
ask = MsgBox("Do You Want To Update Firm Information?",
vbCritical + vbYesNo, "Update")
If ask = vbYes Then
conn
ersoy = "update firms set firmname='" & Text2.text & "',
firmphone='" & MaskEdBox1.text & "', firmfax='" &
```

```
MaskEdBox2.text & "', firmtaxno='" & MaskEdBox3.text & "',
firmcity='" & Text3.text & "', firmadres='" & Text4.text & "',
firmregdate='" & DTPicker1.Value & "' where firmcode=" &
Text1.text & " "
database.Execute (ersoy)
MsgBox ("Firm Information Updated!")
End If
Else
MsgBox ("Please Find Any Firm!")
End If
Command7.Visible = False
Command3.Enabled = False
Command4.Enabled = False
clear
End Sub
Private Sub Command4_Click()
Dim ask
If Text1.text <> "" Then
ask = MsgBox("Do You Want To Delete This Firm Detail?",
vbExclamation + vbYesNo, "Delete")
If ask = vbYes Then
conn
ersoy = "delete * from firms where firmcode=" & Text1.text &
""
database.Execute (ersoy)
ersoy = "delete * from stocks where firmcode=" & Text1.text &
""
database.Execute (ersoy)
MsgBox ("Firm Information Deleted!")
End If
Else
MsgBox ("Please Find Any Firm!")
End If
Command7.Visible = False
```

```
Command3.Enabled = False
Command4.Enabled = False
clear
coun
End Sub
Private Sub Command5_Click()
database.Close
Unload Me
form1.Show
form1.Enabled = True
End Sub
Private Sub Command6_Click()
Dim find As Integer
conn
find = Val(InputBox("Please Insert The Wanted Firm Code!"))
ersoy = "select * from firms where firmcode=" & find & ""
    Set ezgi = database.Execute(ersoy)
    If ezgi.EOF Then
    MsgBox ("The Wanted Firm is Not Available!")
    Else
    Dim i As Integer
ProgressBar1.Min = 0
ProgressBar1.Max = 1000
For i = ProgressBar1.Min To ProgressBar1.Max
ProgressBar1.Visible = True
ProgressBar1.Value = i
Next
ProgressBar1.Visible = False
    Text1.text = ezgi![firmcode]
    Text2.text = ezgi![firmname]
    MaskEdBox1.text = ezgi![firmphone]
    MaskEdBox2.text = ezgi![firmfax]
    MaskEdBox3.text = ezgi![firmtaxno]
    Text3.text = ezgi![firmcity]
    Text4.text = ezgi![firmadres]
    DTPicker1.Value = ezgi![firmregdate]
    Text2.SetFocus
    Command2.Enabled = False
    Command3.Enabled = True
    Command4.Enabled = True
    Command7.Visible = True
    End If
    ezgi.Close
End Sub
Private Sub Command7_Click()
Form4.Text1.text = Text1.text
Form4.Text2.text = Text2.text
Form4.Show
Unload Form2
End Sub
Private Sub Form_Load()
ProgressBar1.Align = vbAlignBottom
ProgressBar1.Visible = False
coun
With StatusBar1.Panels
    Set p = .Add(, , , sbrTime)
    Set p = .Add(, , , sbrDate)
End With
DTPicker1.Value = Date
End Sub
Private Sub clear()
Text1.text = ""
Text2.text = ""
Text3.text = ""
Text4.text = ""
MaskEdBox1.Mask = ""
MaskEdBox1.text = ""
MaskEdBox1.Mask = "0(999)999-99-99"
```

```
MaskEdBox2.Mask = ""
MaskEdBox2.text = ""
MaskEdBox2.Mask = "0(999)999-99-99"
MaskEdBox3.Mask = ""
MaskEdBox3.text = ""
MaskEdBox3.Mask = "999-999-999-999-999"
DTPicker1.Value = Date
End Sub
Private Sub coun()
Dim Count, Count1
conn
Set ezgi = New ADODB.Recordset
Count = "select * from Firms"
Set ezgi = database.Execute(Count)
    If ezgi.EOF Then
     Command6.Enabled = False
     Text1.text = 1
    Else
    Count1 = "select max(firmcode) as cis from firms"
    Set ezgi = database.Execute(Count1)
    Text1.text = ezgi![cis] + 1
    End If
    ezgi.Close
End Sub
Public Sub conn()
Set database = New ADODB.Connection
    database.CursorLocation = adUseServer
    ersoy = "provider=Microsoft.jet.oledb.3.51; Data Source="
& App.Path & "\ezgi.mdb"
    database.Open ersoy
End Sub
Private Sub Form_Unload(Cancel As Integer)
Unload Me
form1.Enabled = True
```

```
End Sub
```

Form3.Frm

```
Option Explicit
Private Declare Function DrawText Lib "user32" Alias
"DrawTextA" (ByVal hdc As Long, ByVal lpStr As String, ByVal
nCount As Long, lpRect As RECT, ByVal wFormat As Long) As Long
Private Declare Function GetTickCount Lib "kernel32" () As
Long
Const DT_CENTER As Long = &H1
Const DT_LEFT As Long = &H0
Const DT_RIGHT As Long = &H2
Private Type RECT
        Left As Long
        Top As Long
        Right As Long
        Bottom As Long
End Type
Dim kare As RECT
Const yazı As String = "This Project Created By EZGI ERSOY..."
& vbCrLf & _
                       vbCrLf & vbCrLf & _
                       "CIS - 400" & vbCrLf & _
                       vbCrLf & vbCrLf & _
                       "Graduation Project" & _
                       vbCrLf & vbCrLf & _
                       "Cement Selling Package Program" & _
                       vbCrLf & vbCrLf & _
                       ">>>> NEAR EAST UNIVERSITY <<<<"
Private Sub Form_Load()
kare.Left = 0
kare.Top = pano.ScaleHeight
kare.Right = pano.ScaleWidth
kare.Bottom = pano.ScaleHeight
End Sub
Private Sub Timer1_Timer()
kare.Top = kare.Top - 1
kare.Bottom = kare.Bottom
If kare.Top = -170 Then
   kare.Top = pano.ScaleHeight
End If
pano.Cls
DrawText pano.hdc, yazı, -1, kare, DT_CENTER
pano.Refresh
End Sub
Private Sub Form_KeyPress(KeyAscii As Integer)
    Unload Me
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Unload Me
    form1.Enabled = True
    form1.Show
End Sub
Private Sub Frame1_Click()
    Unload Me
End Sub
Private Sub Form_Activate()
form1.Enabled = False
End Sub
```

Form4.Frm

```
Option Explicit
Private database As ADODB.Connection
Private ezgi As ADODB.Recordset
Private ersoy As String
Dim p As Panel
Dim ah As Boolean
Dim a, b, d, e, g, h, f As Double
Private Sub Command1_Click()
clear
coun
Command2.Enabled = True
Command3.Enabled = False
Command4.Enabled = False
Text4.SetFocus
End Sub
Private Sub Command2_Click()
care
If ah = True Then
Dim sql, ask, ask1
ask = MsgBox("Do You Want To Save That ?", vbInformation +
vbYesNo, "Save")
If ask = vbYes Then
If Text4.text <> "" And Text5.text <> "" And Text6.text <> ""
And Text8.text <> "" And Text10.text <> "" And Text11.text <>
"" Then
sql = "insert into stocks(firmname, firmcode, stockcode,
stockname, stockminl, stockbd, stockunit, stockbp, stockpperc,
stocksellp) values("
sql = sql & "'" & Text2.text & "',"
sql = sql & "" & Text1.text & ","
sql = sql & "" & Text3.text & ","
sql = sql & "'" & Text4.text & "',"
sql = sql & "" & Text11 & ","
sql = sql & "'" & Label11.Caption & "',"
sql = sql & "" & a & ","
sql = sql & "" & b & ","
sql = sql & "" & h & ","
sql = sql & "'" & Text9 & "')"
database.Execute (sql)
Dim i As Integer
ProgressBar1.Min = 0
ProgressBar1.Max = 1000
For i = ProgressBar1.Min To ProgressBar1.Max
ProgressBar1.Visible = True
ProgressBar1.Value = i
Next
ProgressBar1.Visible = False
ask1 = MsgBox("stock Information Save Successful! ", ,
"Saved")
Command2.Enabled = False
Command3.Enabled = True
Command4.Enabled = True
Else
ask1 = MsgBox("Please Fill The Other Texts!")
Command3.Enabled = False
Command4.Enabled = False
Text4.SetFocus
End If
End If
End If
End Sub
Private Sub Command3_Click()
Frame1.Visible = True
Command3.Enabled = False
Text4.Enabled = False
Text5.Enabled = False
Text6.Enabled = False
Text8.Enabled = False
Text10.Enabled = False
Text11.Enabled = False
End Sub
```

```
Private Sub Command4_Click()
Dim ask
If Text3.text <> "" Then
ask = MsgBox("Do You Want To Delete This Stock Detail?",
vbExclamation + vbYesNo, "Delete")
If ask = vbYes Then
conn
ersoy = "delete * from stocks where stockcode=" & Text3.text &
""
database.Execute (ersoy)
MsgBox ("Stock Information Deleted!")
End If
Else
MsgBox ("Please Find Any Stock!")
End If
Command3.Enabled = False
Command4.Enabled = False
clear
coun
End Sub
Private Sub Command5_Click()
database.Close
Unload Me
Form2.Show
End Sub
Private Sub Command6_Click()
Dim find As Integer
conn
find = Val(InputBox("Please Insert The Wanted Stock Code!"))
ersoy = "select * from stocks where stockcode=" & find & ""
    Set ezgi = database.Execute(ersoy)
    If ezgi.EOF Then
    MsgBox ("The Wanted Stock is Not Available!")
    Else
    Dim i As Integer
ProgressBar1.Min = 0
ProgressBar1.Max = 1000
For i = ProgressBar1.Min To ProgressBar1.Max
ProgressBar1.Visible = True
ProgressBar1.Value = i
Next
ProgressBar1.Visible = False
    Text1.text = ezgi![firmcode]
    Text2.text = ezgi![firmname]
    Text3.text = ezgi![stockcode]
    Text4.text = ezgi![stockname]
    Text5.text = ezgi![stockunit]
    Text6.text = ezgi![stockbp]
    Text8.text = ezgi![stockpperc]
    Text9.text = ezgi![stocksellp]
    Label11.Caption = ezgi![stockbd]
    Text11.text = ezgi![stockminl]
    Command2.Enabled = False
    Command3.Enabled = True
    Command4.Enabled = True
    End If
    ezgi.Close
End Sub
Private Sub Command7_Click()
database.Close
Unload Me
Unload Form2
form1.Show
form1.Enabled = True
End Sub
Private Sub Command8_Click()
Dim ask As String
If Text12.text <> "" And Text13.text <> "" And Text14.text <>
```

```
"" And Text15.text <> "" And Text16.text <> "" Then
ask = MsgBox("Do You Want To Update Stock Information?",
vbCritical + vbYesNo, "Update")
If ask = vbYes Then
conn
ersoy = "update stocks set stockunit=" & Text5.text & ",
stockbp=" & Text6.text & ", stockpperc=" & Text8.text & ",
stocksellp='" & Text9.text & "', stockminl=" & Text11.text & "
where stockcode=" & Text3.text & " "
database.Execute (ersoy)
Dim i As Integer
ProgressBar1.Min = 0
ProgressBar1.Max = 1000
For i = ProgressBar1.Min To ProgressBar1.Max
ProgressBar1.Visible = True
ProgressBar1.Value = i
Next
ProgressBar1.Visible = False
MsgBox ("Stock Information Updated!")
End If
Else
MsgBox ("Please Enter The New Stock Information!")
Command3.Enabled = False
End If
Text12.text = ""
Text13.text = ""
Text14.text = ""
Text15.text = ""
Text16.text = ""
Command3.Enabled = True
Command4.Enabled = False
Command6.Enabled = True
Frame1.Visible = False
Text4.Enabled = True
```

```
Text5.Enabled = True
Text6.Enabled = True
Text8.Enabled = True
Text10.Enabled = True
Text11.Enabled = True
End Sub
Private Sub Command9_Click()
Frame1.Visible = False
Text12.text = ""
Text13.text = ""
Text14.text = ""
Text15.text = ""
Text16.text = ""
clear
End Sub
Private Sub Form_Load()
ProgressBar1.Align = vbAlignBottom
ProgressBar1.Visible = False
coun
With StatusBar1.Panels
    Set p = .Add(, , , sbrTime)
    Set p = .Add(, , , sbrDate)
End With
Label11.Caption = Date
End Sub
Private Sub Form_Unload(Cancel As Integer)
Unload Me
Form2.Show
End Sub
Private Sub Text10_Change()
On Error Resume Next
a = Text5.text
b = Text6.text
h = Text8.text
```

```
d = Val(a) * Val(b)
e = ((d * Val(h)) / 100) + d
f = e / a
g = ((f * Val(Text10.text)) / 100) + f
Text9.text = g
End Sub
Private Sub coun()
Dim Count, Count1
conn
Set ezgi = New ADODB.Recordset
Count = "select * from stocks"
Set ezgi = database.Execute(Count)
    If ezgi.EOF Then
     Command6.Enabled = False
     Text3.text = 1
    Else
    Count1 = "select max(stockcode) as cis from stocks"
    Set ezgi = database.Execute(Count1)
    Text3.text = ezgi![cis] + 1
    End If
    ezgi.Close
End Sub
Public Sub conn()
Set database = New ADODB.Connection
    database.CursorLocation = adUseClient
    ersoy = "provider=Microsoft.jet.oledb.3.51; Data Source="
& App.Path & "\ezgi.mdb"
    database.Open ersoy
End Sub
Private Sub clear()
Text4.text = ""
Text5.text = ""
Text6.text = ""
Text8.text = ""
```

```
Text9.text = ""
Text10.text = ""
Text11.text = ""
End Sub
Private Sub Text10_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    KeyAscii = 0
    SendKeys "{Tab}"
ElseIf InStr(("1234567890" & vbBack & ""), Chr(KeyAscii)) = 0
Then
    KeyAscii = 0
End If
End Sub
Private Sub Text12_LostFocus()
a = Val(Text12.text)
b = a + Val(Text5.text)
Text5.text = b
End Sub
Private Sub Text13_LostFocus()
a = Val(Text13.text)
Text6.text = a
End Sub
Private Sub Text14_LostFocus()
a = Val(Text14.text)
Text8.text = a
End Sub
Private Sub Text15_LostFocus()
a = Val(Text15.text)
Text10.text = a
End Sub
Private Sub Text16_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
a = Val(Text14.text)
Text8.text = a
```

```
End If
End Sub
Private Sub Text16_LostFocus()
If Val(Text5.text) <= Val(Text16.text) Then
MsgBox ("Please Enter Amount Smaller Than Unit!")
Text16.SetFocus
Else
a = Val(Text16.text)
Text11.text = a
End If
End Sub

Private Sub Text5_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    KeyAscii = 0
    SendKeys "{Tab}"
ElseIf InStr(("1234567890" & vbBack & ""), Chr(KeyAscii)) = 0
Then
    KeyAscii = 0
End If
End Sub
Private Sub Text6_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    KeyAscii = 0
    SendKeys "{Tab}"
ElseIf InStr(("1234567890" & vbBack & ""), Chr(KeyAscii)) = 0
Then
    KeyAscii = 0
End If
End Sub
Private Sub Text7_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    KeyAscii = 0
    SendKeys "{Tab}"
```

```
ElseIf InStr(("1234567890" & vbBack & ""), Chr(KeyAscii)) = 0
Then
    KeyAscii = 0
End If
End Sub
Private Sub Text8_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    KeyAscii = 0
    SendKeys "{Tab}"
ElseIf InStr(("1234567890" & vbBack & ""), Chr(KeyAscii)) = 0
Then
    KeyAscii = 0
End If
End Sub
Private Sub care()
If Len(Text4.text) = 0 Then
MsgBox ("Please Fill The Texts!")
Text4.SetFocus
Else
If Val(Text5.text) <= Val(Text11.text) Then
ah = False
MsgBox ("Please Enter Amount Smaller Than Unit!")
Text11.SetFocus
Else
ah = True
End If
End If
End Sub
```

Form5.Frm

```
Option Explicit
Dim p As Panel
```

```
Private database As ADODB.Connection
Private ezgi As ADODB.Recordset
Private ersoy As String
Private Sub Command1_Click()
clear
coun
Command2.Enabled = True
Command3.Enabled = False
Command4.Enabled = False
Command7.Visible = False
Text2.SetFocus
End Sub
Private Sub Command2_Click()
Dim sql, ask, ask1
ask = MsgBox("Do You Want To Save That ?", vbInformation +
vbYesNo, "Save")
If ask = vbYes Then
If Text2.text <> "" And MaskEdBox1.text <> "" And Text3.text
<> "" And MaskEdBox3.text <> "" And Text3.text <> "" And
Text4.text <> "" Then
sql = "insert into cus(cusname, cuscode, cusphone, custaxno,
cuscity, cusadres, cusrd) values("
sql = sql & "'" & Text2.text & "',"
sql = sql & "" & Text1.text & ","
sql = sql & "'" & MaskEdBox1.text & "',"
sql = sql & "'" & MaskEdBox3.text & "',"
sql = sql & "'" & Text3 & "',"
sql = sql & "'" & Text4 & "',"
sql = sql & "'" & DTPicker1.Value & "')"
database.Execute (sql)
Dim i As Integer
ProgressBar1.Min = 0
ProgressBar1.Max = 1000
For i = ProgressBar1.Min To ProgressBar1.Max
```

```
ProgressBar1.Visible = True
ProgressBar1.Value = i
Next
ProgressBar1.Visible = False
ask1 = MsgBox("Customer Information Save Successful! ", ,
"Saved")
Command2.Enabled = False
Command3.Enabled = True
Command4.Enabled = True
Command7.Visible = True
Command6.Enabled = True
Else
ask1 = MsgBox("Please Fill The Other Texts!", vbCritical,
"Customer")
Command6.Enabled = True
Command3.Enabled = False
Command4.Enabled = False
Text2.SetFocus
End If
End If
End Sub
Private Sub Command3_Click()
Dim ask As String
If Text1.text <> "" And Text2.text <> "" And MaskEdBox1.text
<> "" Then
ask = MsgBox("Do You Want To Update Customer Information?",
vbCritical + vbYesNo, "Update")
If ask = vbYes Then
conn
ersoy = "update cus set cusname='" & Text2.text & "',
cusphone='" & MaskEdBox1.text & "', custaxno='" &
MaskEdBox3.text & "', cuscity='" & Text3.text & "',
cusadres='" & Text4.text & "', cusrd='" & DTPicker1.Value & "'
where cuscode=" & Text1.text & " "
```

```
database.Execute (ersoy)
MsgBox ("Customer Information Updated!")
End If
Else
MsgBox ("Please Find Any Customer!")
End If
Command3.Enabled = False
Command4.Enabled = False
clear
End Sub
Private Sub Command4_Click()
Dim ask
If Text1.text <> "" Then
ask = MsgBox("Do You Want To Delete This Customer Detail?",
vbExclamation + vbYesNo, "Delete")
If ask = vbYes Then
conn
ersoy = "delete * from cus where cuscode=" & Text1.text & ""
database.Execute (ersoy)
MsgBox ("Customer Information Deleted!")
End If
Else
MsgBox ("Please Find Any Customer!")
End If
Command3.Enabled = False
Command4.Enabled = False
clear
coun
End Sub
Private Sub Command5_Click()
Unload Me
form1.Show
form1.Enabled = True
End Sub
```

```
Private Sub Command6_Click()
Dim find As Integer
conn
find = Val(InputBox("Please Insert The Wanted Customer
Code!"))
ersoy = "select * from cus where cuscode=" & find & ""
    Set ezgi = database.Execute(ersoy)
    If ezgi.EOF Then
    MsgBox ("The Wanted Customer is Not Available!")
    Else
    Dim i As Integer
ProgressBar1.Min = 0
ProgressBar1.Max = 1000
For i = ProgressBar1.Min To ProgressBar1.Max
ProgressBar1.Visible = True
ProgressBar1.Value = i
Next
ProgressBar1.Visible = False
    Text1.text = ezgi![cuscode]
    Text2.text = ezgi![cusname]
    MaskEdBox1.text = ezgi![cusphone]
    MaskEdBox3.text = ezgi![custaxno]
    Text3.text = ezgi![cuscity]
    Text4.text = ezgi![cusadres]
    DTPicker1.Value = ezgi![cusrd]
    Text2.SetFocus
    Command7.Visible = True
    Command2.Enabled = False
    Command3.Enabled = True
    Command4.Enabled = True
    End If
    ezgi.Close
End Sub
Private Sub Command7_Click()
```

```
Form7.Text4.text = Text1.text
Form7.Text5.text = Text2.text
Form7.Text7.text = MaskEdBox1.text
Form7.Text6.text = Text4.text
Form7.Show
End Sub
Private Sub Form_Load()
ProgressBar1.Align = vbAlignBottom
ProgressBar1.Visible = False
coun
With StatusBar1.Panels
    Set p = .Add(, , , sbrTime)
    Set p = .Add(, , , sbrDate)
End With
DTPicker1.Value = Date
End Sub
Private Sub clear()
Text1.text = ""
Text2.text = ""
Text3.text = ""
Text4.text = ""
MaskEdBox1.Mask = ""
MaskEdBox1.text = ""
MaskEdBox1.Mask = "0(999)999-99-99"
MaskEdBox3.Mask = ""
MaskEdBox3.text = ""
MaskEdBox3.Mask = "999-999-999-999-999"
DTPicker1.Value = Date
End Sub
Private Sub coun()
Dim Count, Count1
conn
Set ezgi = New ADODB.Recordset
Count = "select * from cus"
```

```
Set ezgi = database.Execute(Count)
    If ezgi.EOF Then
     Command6.Enabled = False
     Text1.text = 1
    Else
    Count1 = "select max(cuscode) as cis from cus"
    Set ezgi = database.Execute(Count1)
    Text1.text = ezgi![cis] + 1
    End If
    ezgi.Close
End Sub
Public Sub conn()
Set database = New ADODB.Connection
    database.CursorLocation = adUseClient
    ersoy = "provider=Microsoft.jet.oledb.3.51; Data Source="
& App.Path & "\ezgi.mdb"
    database.Open ersoy
End Sub
Private Sub Form_Unload(Cancel As Integer)
Unload Me
form1.Show
form1.Enabled = True
End Sub
```

Form6.Frm

```
Option Explicit
Dim p As Panel
Private database As ADODB.Connection
Private ezgi As ADODB.Recordset
Private ersoy As String
Private Sub Command1_Click()
clear
```

```
coun
Command2.Enabled = True
Command3.Enabled = False
Command4.Enabled = False
Text2.SetFocus
End Sub
Private Sub Command2_Click()
Dim sql, ask, ask1
ask = MsgBox("Do You Want To Save That ?", vbInformation +
vbYesNo, "Save")
If ask = vbYes Then
If Text2.text <> "" And MaskEdBox1.text <> "" And Text3.text
<> "" And MaskEdBox3.text <> "" And Text3.text <> "" And
Text4.text <> "" Then
sql = "insert into emp(empname, empcode, empphone, empsc,
empcity, empadres,empsal, emprd) values("
sql = sql & "'" & Text2.text & "',"
sql = sql & "" & Text1.text & ","
sql = sql & "'" & MaskEdBox1.text & "',"
sql = sql & "'" & MaskEdBox3.text & "',"
sql = sql & "'" & Text3 & "',"
sql = sql & "'" & Text4 & "',"
sql = sql & "" & Text5.text & ","
sql = sql & "'" & DTPicker1.Value & "')"
database.Execute (sql)
Dim i As Integer
ProgressBar1.Min = 0
ProgressBar1.Max = 1000
For i = ProgressBar1.Min To ProgressBar1.Max
ProgressBar1.Visible = True
ProgressBar1.Value = i
Next
ProgressBar1.Visible = False
ask1 = MsgBox("Employee Information Save Successful! ", ,
"Saved")
Command2.Enabled = False
Command3.Enabled = True
Command4.Enabled = True
Command6.Enabled = True
Else
ask1 = MsgBox("Please Fill The Other Texts!")
Command3.Enabled = False
Command4.Enabled = False
Command6.Enabled = True
Text2.SetFocus
End If
End If
End Sub
Private Sub Command3_Click()
Dim ask As String
If Text1.text <> "" And Text2.text <> "" And MaskEdBox1.text
<> "" Then
ask = MsgBox("Do You Want To Update Employee Information?",
vbCritical + vbYesNo, "Update")
If ask = vbYes Then
conn
ersoy = "update emp set empname='" & Text2.text & "',
empphone='" & MaskEdBox1.text & "', empsc='" & MaskEdBox3.text
& "', empcity='" & Text3.text & "', empadres='" & Text4.text &
"',empsal=" & Text5.text & ", emprd='" & DTPicker1.Value & "'
where empcode=" & Text1.text & " "
database.Execute (ersoy)
MsgBox ("Employee Information Updated!")
End If
Else
MsgBox ("Please Find Any Employee!")
End If
Command3.Enabled = False
```

```
Command4.Enabled = False
clear
End Sub
Private Sub Command4_Click()
Dim ask
If Text1.text <> "" Then
ask = MsgBox("Do You Want To Delete This Employee Detail?",
vbExclamation + vbYesNo, "Delete")
If ask = vbYes Then
conn
ersoy = "delete * from emp where empcode=" & Text1.text & ""
database.Execute (ersoy)
MsgBox ("Employee Information Deleted!")
End If
Else
MsgBox ("Please Find Any Employee!")
End If
Command3.Enabled = False
Command4.Enabled = False
clear
coun
End Sub
Private Sub Command5_Click()
database.Close
Unload Me
Form1.Show
Form1.Enabled = True
End Sub
Private Sub Command6_Click()
Dim find As Integer
conn
find = Val(InputBox("Please Insert The Wanted Employee
Code!"))
ersoy = "select * from emp where empcode=" & find & ""
```

```
    Set ezgi = database.Execute(ersoy)
    If ezgi.EOF Then
    MsgBox ("The Wanted Employee is Not Available!")
    Else
    Dim i As Integer
ProgressBar1.Min = 0
ProgressBar1.Max = 1000
For i = ProgressBar1.Min To ProgressBar1.Max
ProgressBar1.Visible = True
ProgressBar1.Value = i
Next
ProgressBar1.Visible = False
    Text1.text = ezgi![empcode]
    Text2.text = ezgi![empname]
    MaskEdBox1.text = ezgi![empphone]
    MaskEdBox3.text = ezgi![empsc]
    Text3.text = ezgi![empcity]
    Text4.text = ezgi![empadres]
    Text5.text = ezgi![empsal]
    DTPicker1.Value = ezgi![emprd]
    Text2.SetFocus
    Command2.Enabled = False
    Command3.Enabled = True
    Command4.Enabled = True
    End If
    ezgi.Close
End Sub

Private Sub Form_Load()
ProgressBar1.Align = vbAlignBottom
ProgressBar1.Visible = False
coun
With StatusBar1.Panels
    Set p = .Add(, , , sbrTime)
    Set p = .Add(, , , sbrDate)
End With
DTPicker1.Value = Date
End Sub
Private Sub clear()
Text1.text = ""
Text2.text = ""
Text3.text = ""
Text4.text = ""
Text5.text = ""
MaskEdBox1.Mask = ""
MaskEdBox1.text = ""
MaskEdBox1.Mask = "0(999)999-99-99"
MaskEdBox3.Mask = ""
MaskEdBox3.text = ""
MaskEdBox3.Mask = "999-999-999-999-999"
DTPicker1.Value = Date
End Sub
Private Sub coun()
Dim Count, Count1
conn
Set ezgi = New ADODB.Recordset
Count = "select * from emp"
Set ezgi = database.Execute(Count)
    If ezgi.EOF Then
     Command6.Enabled = False
     Text1.text = 1
    Else
    Count1 = "select max(empcode) as cis from emp"
    Set ezgi = database.Execute(Count1)
    Text1.text = ezgi![cis] + 1
    End If
    ezgi.Close
End Sub
```

```
Public Sub conn()
Set database = New ADODB.Connection
    database.CursorLocation = adUseClient
    ersoy = "provider=Microsoft.jet.oledb.3.51; Data Source="
& App.Path & "\ezgi.mdb"
    database.Open ersoy
End Sub
Private Sub Form_Unload(Cancel As Integer)
Unload Me
form1.Show
form1.Enabled = True
End Sub
Private Sub Text5_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    KeyAscii = 0
    SendKeys "{Tab}"
ElseIf InStr(("1234567890" & vbBack & ""), Chr(KeyAscii)) = 0
Then
    KeyAscii = 0
End If
End Sub
```

Form7.Frm

```
Option Explicit
Private database As ADODB.Connection
Private ezgi, ezgi1 As ADODB.Recordset
Private ersoy As String
Private sql, sql1, sql2, sql3 As String
Private Sub Command1_Click()
database.Close
Unload Me
Unload Form5
```

```
Form1.Show
Form1.Enabled = True
End Sub
Private Sub Command10_Click()
Form12.Show
End Sub
Private Sub Command2_Click()
database.Close
Unload Me
Form5.Command7.Visible = False
Form5.Show
Form5.Enabled = True
End Sub
Private Sub Command3_Click()
Dim cevap, sec, i
Dim index As Integer
sec = List1(index).ListIndex
If sec < 0 Then
cevap = MsgBox("Select the removed item", vbCritical, "Remove
Item")
Else
cevap = MsgBox("Are you sure to remove this item from the list
?", vbYesNo + vbQuestion, "Remove item")
If cevap = vbYes Then
 For i = 0 To 4
 List1(i).RemoveItem List1(i).ListIndex
 Next
 Dim j, a, b
For j = 0 To List1(4).ListCount
a = Val(List1(4).List(j))
b = b + a
Next
Text2.text = b
End If
End If
End Sub
Private Sub coun()
Dim Count, Count1
conn
Set ezgi = New ADODB.Recordset
Count = "select incode from invoice2"
Set ezgi = database.Execute(Count)
    If ezgi.EOF Then
     Label9.Caption = 1
    Else
    Count1 = "select max(incode) as cis from invoice2"
    Set ezgi = database.Execute(Count1)
    Label9.Caption = ezgi![cis] + 1
    End If
    ezgi.Close
End Sub
Private Sub Command4_Click()
Dim k, i, ask, toplam, stk, stk2, sq, sq1, sq2, decrip
If DBCombo1.text = "Select Employee" Then
ask = MsgBox("Please select the employee name", vbInformation,
"Invoice")
Else
ask = MsgBox("Do You Want to Save / Print The Invoice ?",
vbYesNo + vbQuestion, "Invoice Saving")
If ask = vbYes Then
conn
   For i = 0 To List1(0).ListCount - 1
   stk = Val(List1(0).List(i))
   stk2 = List1(1).List(i)
   sq = Val(List1(2).List(i))
   sq1 = Val(List1(3).List(i))
   sq2 = Val(List1(4).List(i))
sql1 = "insert into invoice1(incode, cuscode, stockcode,
sname, quan, uprice, totp) values ("
sql1 = sql1 & "" & Label9 & ","
sql1 = sql1 & "" & Text4.text & ","
sql1 = sql1 & "" & stk & ","
sql1 = sql1 & "'" & stk2 & "',"
sql1 = sql1 & "" & sq & ","
sql1 = sql1 & "" & sq1 & ","
sql1 = sql1 & "" & sq2 & ")"
database.Execute (sql1)
sql2 = "update stocks set stockunit=stockunit-" & sq & " where
stockcode=" & stk & ""
database.Execute (sql2)
   Next
If Option1 = True Then
toplam = 0
decrip = Label9 + "," + "no" + "Invoice"
sql3 = "insert into account(accdate, expense, revenue,
description, expcode) values('" & Label12 & "', " & 0 & ", " &
Text2.text & " ,'" & decrip & "'," & toplam & "  )"
database.Execute (sql3)
End If
If Option2 = True Then
toplam = Text2.text
End If
sql2 = "insert into invoice2(incode, cuscode, subtot, invdate,
empname) values (" & Label9 & ", " & Text4 & ", " & toplam &
", '" & Label12 & "', '" & DBCombo1.text & "')"
database.Execute (sql2)
ask = MsgBox("Invoice Printed/saved succesfully",
vbInformation, "Invoice")
Command2_Click
End If
End If
End Sub
```

```
Private Sub Command5_Click()
Form10.Show
End Sub
Private Sub Command6_Click()
Dim ask As Integer
On Error Resume Next
ask = MsgBox("Are You Sure Print This Information ?", vbYesNo,
"Account")
'CommonDialog1.Action = 5
If ask = vbYes Then iprint
End Sub
Private Sub Form_Activate()
On Error Resume Next
Text8.SetFocus
End Sub
Private Sub Form_Load()
Option1 = True
coun
Label12 = Date
End Sub
Private Sub Form_Unload(Cancel As Integer)
Unload Me
Form5.Show
Form5.Enabled = True
End Sub
Private Sub List1_Click(index As Integer)
Dim secind, topin, j
On Error Resume Next
secind = List1(index).ListIndex
topin = List1(index).TopIndex
 For j = 0 To 4
 List1(j).ListIndex = secind
 List1(j).TopIndex = topin
 Next
```

```
End Sub
Private Sub Text11_KeyPress(KeyAscii As Integer)
Dim ans
If KeyAscii = 13 Then
KeyAscii = 0
If Val(Text11.text) > Val(Text9.text) Then
ans = MsgBox("Stock not enough to sell this amount / Available
stock is =" & Text9 & " unit", vbCritical, "Invoice")
Else
List1(0).AddItem Text8.text
List1(1).AddItem Text1.text
List1(2).AddItem Text11.text
List1(3).AddItem Text10.text
List1(4).AddItem (Val(Text10.text) * Val(Text11.text))
Text11.Enabled = False
clear
Text8.text = ""
Text8.SetFocus
Dim i, a, b
For i = 0 To List1(4).ListCount
a = Val(List1(4).List(i))
b = b + a
Next
Text2.text = b
End If
ElseIf InStr(("1234567890" & vbBack & ""), Chr(KeyAscii)) = 0
Then
    KeyAscii = 0
End If
End Sub
Private Sub Text8_KeyPress(KeyAscii As Integer)
Dim i, a, b
Dim ans
If KeyAscii = 13 Then
```

```
   If Len(Text8.text) = 0 Then
   ans = MsgBox("Please enter the stock code", vbCritical,
"Invoice")
   Else
   KeyAscii = 0
   b = Text8.text
   For i = 0 To List1(0).ListCount
       If (List1(0).List(i)) = b Then
       ans = MsgBox("You entered this stock before please
enter another stock / for Re-enter remove stock = " & Text8 &
" from the list ", vbCritical, "Invoice")
       Text8.text = ""
       Text8.SetFocus
       Exit Sub
       End If
    Next
   conn
   sql = "select * from  stocks where stockcode=" +
Text8.text + ""
Set ezgi = database.Execute(sql)
If ezgi.EOF Then
ans = MsgBox("Stock Code Not Found", vbCritical, "Search")
clear
Text8.SetFocus
Else
Text1.text = ezgi![stockname]
Text9.text = ezgi![stockunit]
Text10.text = ezgi![stocksellp]
Text3.text = ezgi![stockminl]
ezgi.Close
Text11.Enabled = True
Text11.text = ""
Text11.SetFocus
End If
End If
ElseIf InStr(("1234567890" & vbBack & ""), Chr(KeyAscii)) = 0
Then
    KeyAscii = 0
End If
End Sub
Private Sub conn()
Set database = New ADODB.Connection
    database.CursorLocation = adUseClient
    ersoy = "provider=Microsoft.jet.oledb.3.51; Data Source="
& App.Path & "\ezgi.mdb"
    database.Open ersoy
End Sub
Public Sub clear()
Text8.text = ""
Text1.text = ""
Text9.text = ""
Text10.text = ""
Text11.text = ""
Text3.text = ""
End Sub
Sub iprint()
    Dim X As Printer
    Dim y, x1, x2, i, artim, yb, ys, x6, x5, x3, x4, k, x7
    On Error GoTo ass
    Printer.ScaleMode = 6
    Printer.FontName = "Courier New Tr"
    Printer.FontSize = 10
    y = 10: x1 = 5: x2 = x1 + 50: artim = 7
    Printer.CurrentX = 1
    Printer.CurrentY = 2
    Printer.Print "Cement Co. Account Process Invoice" & "
" & Format(Date, "Short Date")
    Printer.Line (0, 9)-(Printer.ScaleWidth, 9)
```

```
If Not IsNull(Text4.text) Then
    y = y + artim
    Printer.CurrentX = x1
    Printer.CurrentY = y
    Printer.Print "Customer Code :"
    Printer.CurrentX = x2
    Printer.CurrentY = y
    Printer.Print Text4.text
End If
If Not IsNull(Text5.text) Then
    y = y + artim
    Printer.CurrentX = x1
    Printer.CurrentY = y
    Printer.Print "Customer Name :"
    Printer.CurrentX = x2
    Printer.CurrentY = y
    Printer.Print Text5.text
End If
If Not IsNull(Text7.text) Then
    y = y + artim
    Printer.CurrentX = x1
    Printer.CurrentY = y
    Printer.Print "Customer Phone :"
    Printer.CurrentX = x2
    Printer.CurrentY = y
    Printer.Print Text7.text
End If
If Not IsNull(Text6.text) Then
    y = y + artim
    Printer.CurrentX = x1
    Printer.CurrentY = y
    Printer.Print "Customer City :"
    Printer.CurrentX = x2
    Printer.CurrentY = y
    Printer.Print Text6.text
End If
If Not IsNull(Label9.Caption) Then
    y = y + artim
    Printer.CurrentX = x1
    Printer.CurrentY = y
    Printer.Print "Invoice No :"
    Printer.CurrentX = x2
    Printer.CurrentY = y
    Printer.Print Label9.Caption
End If
If Not IsNull(Label12.Caption) Then
    y = y + artim
    Printer.CurrentX = x1
    Printer.CurrentY = y
    Printer.Print "Invoice Date :"
    Printer.CurrentX = x2
    Printer.CurrentY = y
    Printer.Print Label12.Caption
End If
If Not IsNull(DBCombo1.text) Then
    y = y + artim
    Printer.CurrentX = x1
    Printer.CurrentY = y
    Printer.Print "Employee Name :"
    Printer.CurrentX = x2
    Printer.CurrentY = y
    Printer.Print DBCombo1.text
End If
If Not IsNull(Text2.text) Then
    y = y + artim
    Printer.CurrentX = x1
    Printer.CurrentY = y
    Printer.Print "Sub Total :"
```

```
      Printer.CurrentX = x2
      Printer.CurrentY = y
      Printer.Print Text2.text
  End If
      Printer.Print ""
      Printer.Print ""
      Printer.Print ""
      Printer.Print ""
   yb = Printer.CurrentY
   x7 = Printer.TextWidth(Space(70))
   x6 = x7 + Printer.TextWidth(Space(25))
   x5 = x6 + Printer.TextWidth(Space(25))
   x3 = x5 + Printer.TextWidth(Space(25))
   x4 = x3 + Printer.TextWidth(Space(25))
   Printer.Line (30, yb)-(x4 - 30, yb)
   Printer.CurrentX = 30
   Printer.Print "Stock Code";
   Printer.CurrentX = x7 + 30
   Printer.Print "Stock Name";
   Printer.CurrentX = x6 + 30
   Printer.Print "Quantity";
   Printer.CurrentX = x5 + 30
   Printer.Print "Unit Price";
   Printer.CurrentX = x3 + 30
   Printer.Print "Total Amount"
   Printer.Line (30, Printer.CurrentY)-(x4 - 30,
Printer.CurrentY)
   For k = 0 To List1(0).ListCount - 1
   Printer.CurrentX = 30
   Printer.Print List1(0).List(k);
   Printer.CurrentX = x7 + 30
   Printer.Print List1(1).List(k);
   Printer.CurrentX = x6 + 30
   Printer.Print List1(2).List(k);
```

```
    Printer.CurrentX = x5 + 30
    Printer.Print List1(3).List(k);
    Printer.CurrentX = x3 + 30
    Printer.Print List1(4).List(k)
    Printer.Line (30, Printer.CurrentY)-(x4 - 30,
Printer.CurrentY)
Next k
    ys = Printer.CurrentY
Printer.Line (30, yb)-(30, ys)
Printer.Line (x7 + 30, yb)-(x7 + 30, ys)
Printer.Line (x6 + 30, yb)-(x6 + 30, ys)
Printer.Line (x5 + 30, yb)-(x5 + 30, ys)
Printer.Line (x3 + 30, yb)-(x3 + 30, ys)
Printer.Line (x4 + 30, yb)-(x4 + 30, ys)
Printer.EndDoc
Exit Sub
ass:
MsgBox "Error :" & Err.Description, 16, "Account"
End Sub
```

Form8.Frm

```
Option Explicit
Dim rst, rst1, rst2, rst3 As New ADODB.Recordset
Private database As ADODB.Connection
Private ezgi As ADODB.Recordset
Private ersoy As String
Dim text, textq, a, decrip, b, c, ask
Private Sub Command1_Click()
Dim ask
If Option1 = True Then
On Error Resume Next
If Text1.text = "" Then
```

```
ask = MsgBox("Please Enter The Selected Criteria!",
vbCritical, "Account")
Frame1.Visible = False
Text1.SetFocus
Else
conne
text = "select incode, cuscode, invdate, empname, subtot from
invoice2 where incode = " & Text1.text & " and subtot>0 "
Set rst = database.Execute(text)
If rst.EOF Then
ask = MsgBox("Wanted Invoice No Not Exist!", vbCritical,
"Account")
Label2.Visible = False
Label3.Visible = False
Frame1.Visible = False
Else
textq = "select  sum(subtot) as ah  from invoice2 where
incode=" & Text1.text & ""
Set rst1 = database.Execute(textq)
a = rst![cuscode]
Set DataGrid1.DataSource = rst
DataGrid1.Caption = "LIST OF INVOICE BY INVOICE CODE"
DataGrid1.Columns(0).Caption = "Invoice No"
DataGrid1.Columns(1).Caption = "Customer No"
DataGrid1.Columns(2).Caption = "Invoice Date"
DataGrid1.Columns(3).Caption = "Employee Name"
DataGrid1.Columns(4).Caption = "Invoice Amount"
DataGrid1.Columns(4).Alignment = dbgRight
'DataGrid1.Columns(4).NumberFormat = "#,##0"
Label3.Caption = rst1![ah]
Label2.Visible = True
Label3.Visible = True
Frame1.Visible = True
End If
End If
End If
If Option2 = True Then
On Error Resume Next
If Text1.text = "" Then
ask = MsgBox("Please Enter The Selected Criteria!",
vbCritical, "Account")
Frame1.Visible = False
Text1.SetFocus
Else
conne
text = "select incode, cuscode, invdate, empname, subtot from
invoice2 where cuscode = " & Text1.text & " and Subtot>0 "
Set rst = database.Execute(text)
If rst.EOF Then
ask = MsgBox("Wanted Customer No Not Exist!", vbCritical,
"Account")
Label2.Visible = False
Label3.Visible = False
Frame1.Visible = False
Else
textq = "select  sum(subtot) as ah  from invoice2 where
cuscode=" & Text1.text & ""
Set rst1 = database.Execute(textq)
a = rst![cuscode]
Set DataGrid1.DataSource = rst
DataGrid1.Caption = "LIST OF INVOICE BY CUSTOMER CODE"
DataGrid1.Columns(0).Caption = "Invoice No"
DataGrid1.Columns(1).Caption = "Customer No"
DataGrid1.Columns(2).Caption = "Invoice Date"
DataGrid1.Columns(3).Caption = "Employee Name"
DataGrid1.Columns(4).Caption = "Invoice Amount"
DataGrid1.Columns(4).Alignment = dbgRight
'DataGrid1.Columns(4).NumberFormat = "#,##0"
```

```
Label3.Caption = rst1![ah]
Label2.Visible = True
Label3.Visible = True
Frame1.Visible = True
End If
End If
End If
Text1.text = ""
Text1.SetFocus
End Sub
Private Sub Command2_Click()
com
Text1.text = ""
Text1.SetFocus
Label2.Visible = False
Label3.Visible = False
Label4.Visible = False
Label5.Visible = False
Label6.Visible = True
Option1 = False
Option2 = False
Command1.Enabled = False
End Sub
Private Sub Command3_Click()
Dim ask
If Text2.text <> "" And Text3.text <> "" Then
decrip = Text2 + "," + "no" + "Invoice"
b = 0
c = Text3.text
Set rst2 = database.Execute("update invoice2 set
subtot=subtot-" & c & " where incode=" & Text2 & " and
cuscode=" & a & " ")
conne
Set rst3 = database.Execute("insert into account(revenue,
accdate, description, expense ) values(" & Text3.text & ", '"
& Label12 & "', '" & decrip & "'," & b & ")")
Frame1.Visible = False
Text1.text = ""
Text1.SetFocus
ask = MsgBox("Invoice Paid", vbInformation, "Account")
Text2.text = ""
Text3.text = ""
Else
ask = MsgBox("Please Insert The Invoice No / Amount!",
vbCritical, "Account")
Text2.text = ""
Text3.text = ""
Text2.SetFocus
End If
End Sub
Private Sub Command4_Click()
Frame1.Visible = False
Text1.text = ""
Text1.SetFocus
End Sub
Private Sub Command5_Click()
database.Close
Unload Me
form1.Show
End Sub
Private Sub Form_Load()
Label12.Caption = Date
com
Command1.Enabled = False
End Sub
Private Sub com()
On Error Resume Next
conne
```

```
Set rst = database.Execute("Select incode, cuscode, invdate,
empname, subtot from invoice2 where subtot >0")
Set DataGrid1.DataSource = rst
DataGrid1.Caption = "LIST OF INVOICE BY INVOICE CODE"
DataGrid1.Columns(0).Caption = "Invoice No"
DataGrid1.Columns(1).Caption = "Customer No"
DataGrid1.Columns(2).Caption = "Invoice Date"
DataGrid1.Columns(3).Caption = "Employee Name"
DataGrid1.Columns(4).Caption = "Invoice Amount"
DataGrid1.Columns(4).Alignment = dbgRight
'DataGrid1.Columns(4).NumberFormat = "#,##0"
Select Case rst.RecordCount
Case Is > 1
Label8.Caption = "Total  " & Trim(Str(rst.RecordCount)) & "
Record Exist"
Case Is = 1
Label8.Caption = "Total  " & Trim(Str(rst.RecordCount)) & "
Record Exist"
Case Is = 0
Label8.Caption = "Record Not Exist"
End Select
form1.Enabled = False
End Sub
Private Sub Form_Unload(Cancel As Integer)
Unload Me
form1.Show
form1.Enabled = True
End Sub
Private Sub Option1_Click()
Label4.Visible = True
Label5.Visible = False
Label6.Visible = False
Text1.SetFocus
Command1.Enabled = True
```

```
End Sub
Private Sub Option2_Click()
Label5.Visible = True
Label6.Visible = False
Text1.SetFocus
Command1.Enabled = True
End Sub
Private Sub conne()
Set database = New ADODB.Connection
    database.CursorLocation = adUseClient
    ersoy = "provider=Microsoft.jet.oledb.3.51; Data Source="
& App.Path & "\ezgi.mdb"
    database.Open ersoy
End Sub
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    KeyAscii = 0
    SendKeys "{Tab}"
ElseIf InStr(("1234567890" & vbBack & ""), Chr(KeyAscii)) = 0
Then
    KeyAscii = 0
End If
End Sub
Private Sub Text2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    KeyAscii = 0
    SendKeys "{Tab}"
ElseIf InStr(("1234567890" & vbBack & ""), Chr(KeyAscii)) = 0
Then
    KeyAscii = 0
End If
End Sub
Private Sub Text3_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
```

```
   KeyAscii = 0
   SendKeys "{Tab}"
ElseIf InStr(("1234567890" & vbBack & ""), Chr(KeyAscii)) = 0
Then
   KeyAscii = 0
End If
End Sub
```

Form9.Frm

```
Option Explicit
Private database As ADODB.Connection
Private ezgi, rst As ADODB.Recordset
Private ersoy As String
Dim ah, sql, sql1
Private Sub Command1_Click()
database.Close
Unload Me
form1.Show
form1.Enabled = True
End Sub
Private Sub Command2_Click()
Dim sql, ask, ask1
ask = MsgBox("Do You Want To Save That ?", vbInformation +
vbYesNo, "Save")
If ask = vbYes Then
If Text2.text <> "" And Text3.text <> "" And Text3.text <> ""
Then
sql = "insert into account(expcode, description, expense,
accdate) values("
sql = sql & "" & Text1.text & ","
sql = sql & "'" & Text2.text & "',"
sql = sql & "" & Text3 & ","
sql = sql & "'" & DTPicker1.Value & "')"
database.Execute (sql)
Dim i As Integer
ProgressBar1.Min = 0
ProgressBar1.Max = 1000
For i = ProgressBar1.Min To ProgressBar1.Max
ProgressBar1.Visible = True
ProgressBar1.Value = i
Next
ProgressBar1.Visible = False
ask1 = MsgBox("Employee Information Save Successful! ", ,
"Saved")
coun
Text2.text = ""
Text3.text = ""
Text2.SetFocus
Else
ask1 = MsgBox("Please Fill The Other Texts!", vbInformation,
"Account")
End If
End If
End Sub
Private Sub Form_Load()
On Error Resume Next
form1.Enabled = False
coun
Adodc1.RecordSource = "select expcode ,description, accdate,
expense from account where expcode > 0 "
Set DataGrid1.DataSource = Adodc1
DataGrid1.Caption = "LIST OF EXPENSE BY INVOICE CODE"
DataGrid1.Columns(0).Caption = "Expense No"
DataGrid1.Columns(1).Caption = "Description"
DataGrid1.Columns(2).Caption = "Expense Date"
DataGrid1.Columns(3).Caption = "Expense Amount"
```

```
DataGrid1.Columns(3).Alignment = dbgRight
conn
sql = "select  sum(expense) as ah  from account"
Set ezgi = database.Execute(sql)
Text6.text = ezgi![ah]
sql1 = "select  sum(revenue) as ah  from account"
Set ezgi = database.Execute(sql1)
Text4.text = ezgi![ah]
Text5.text = Val(Text4.text) - Val(Text6.text)
ProgressBar1.Align = vbAlignBottom
ProgressBar1.Visible = False
End Sub
Private Sub Form_Unload(Cancel As Integer)
Unload Me
form1.Show
form1.Enabled = True
End Sub
Public Sub conn()
Set database = New ADODB.Connection
    database.CursorLocation = adUseServer
    ersoy = "provider=Microsoft.jet.oledb.3.51; Data Source="
& App.Path & "\ezgi.mdb"
    database.Open ersoy
End Sub
Private Sub coun()
Dim Count, Count1
conn
Set ezgi = New ADODB.Recordset
Count = "select expcode from account"
Set ezgi = database.Execute(Count)
    If ezgi.EOF Then
    Text1.text = 1
    Else
    Count1 = "select max(expcode) as cis from account"
    Set ezgi = database.Execute(Count1)
    Text1.text = ezgi![cis] + 1
    End If
    ezgi.Close
End Sub
Private Sub Text3_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    KeyAscii = 0
    SendKeys "{Tab}"
ElseIf InStr(("1234567890" & vbBack & ""), Chr(KeyAscii)) = 0
Then
    KeyAscii = 0
End If
End Sub
```

Form10.Frm

```
Dim rst, rst1 As New ADODB.Recordset
Private database As ADODB.Connection
Private ersoy As String
Dim ezgi
Private Sub Command1_Click()
If Len(Text1.text) = 0 Then
ask = MsgBox("Please Enter The Stock Name", vbCritical,
"Search")
Text1.SetFocus
Else
conne
ezgi = "select stockcode, stockname, firmname from stocks
where stockname like '" & Text1.text & "%" & "' "
Set rst = database.Execute(ezgi)
If rst.EOF Then
ask = MsgBox("Wanted Product Name Not Exist!", vbCritical,
"Account")
Text1.text = ""
Text1.SetFocus
Else
Set DataGrid1.DataSource = rst
DataGrid1.Caption = "LIST OF PRODUCT BY PRODUCT NAME"
DataGrid1.Columns(0).Caption = "Stock Code"
DataGrid1.Columns(1).Caption = "Stock Name"
DataGrid1.Columns(2).Caption = "Firm Name"
Text1.text = ""
Text1.SetFocus
End If
End If
End Sub
Private Sub conne()
Set database = New ADODB.Connection
    database.CursorLocation = adUseClient
    ersoy = "provider=Microsoft.jet.oledb.3.51; Data Source="
& App.Path & "\ezgi.mdb"
    database.Open ersoy
End Sub
Private Sub Form_Load()
Form7.Enabled = False
End Sub
Private Sub Form_Unload(Cancel As Integer)
Form7.Enabled = True
End Sub


Form11.Frm-Module1


Option Explicit
Private database As ADODB.Connection
Private ezgi As ADODB.Recordset
Private ersoy As String
Public Sub Main()
  Dim Count, Count1, i
    Load form1
    Form11.Show
    DoEvents
    Form11.ProgressBar1.Min = 0
    Form11.ProgressBar1.Max = 13000
    Form11.ProgressBar1.Value = 0
    For i = 0 To 13000 - 1
         Form11.ProgressBar1.Value = Form11.ProgressBar1.Value
+ 1
    Next i
    Unload Form11
     conn
    Set ezgi = New ADODB.Recordset
    Count = "select * from stocks where stockmin1>=Stockunit"
    Set ezgi = database.Execute(Count)
    If ezgi.EOF Then
          form1.Show
    Else
    Form12.Show
    End If
    ezgi.Close
End Sub
Public Sub conn()
Set database = New ADODB.Connection
    database.CursorLocation = adUseClient
    ersoy = "provider=Microsoft.jet.oledb.3.51; Data Source="
& App.Path & "\ezgi.mdb"
    database.Open ersoy
End Sub
```

Form12.Frm

```
Option Explicit
Dim rst As New ADODB.Recordset
Private database As ADODB.Connection
Private ezgi As ADODB.Recordset
Private ersoy As String
Private Sub Form_Load()
com
End Sub
Private Sub com()
On Error Resume Next
conne
Set rst = database.Execute("Select stockcode,
stockname,firmcode, stockunit,stockminl from stocks where
stockminl >= stockunit")
Set DataGrid1.DataSource = rst
DataGrid1.Caption = "LIST OF MINIMUM LEVEL GRATER THAN
AVAİLABLE STOCK UNIT BY STOCKCODE"
DataGrid1.Columns(0).Caption = "StockCode"
DataGrid1.Columns(1).Caption = "StockName"
DataGrid1.Columns(2).Caption = "FirmCode"
DataGrid1.Columns(3).Caption = "StockUnit"
DataGrid1.Columns(4).Caption = "Stock Min.Level"
DataGrid1.Columns(4).Alignment = dbgRight
Select Case rst.RecordCount
Case Is > 1
Label1.Caption = "Total  " & Trim(Str(rst.RecordCount)) & "
Min. Level >= Stock Quantity Record Exist"
Case Is = 1
Label1.Caption = "Total  " & Trim(Str(rst.RecordCount)) & "
Record Exist"
Case Is = 0
Label1.Caption = "Record Not Exist"
```

```
End Select
form1.Enabled = False
End Sub
Private Sub conne()
Set database = New ADODB.Connection
    database.CursorLocation = adUseClient
    ersoy = "provider=Microsoft.jet.oledb.3.51; Data Source="
& App.Path & "\ezgi.mdb"
    database.Open ersoy
End Sub
Private Sub Form_Unload(Cancel As Integer)
Unload Me
form1.Show
form1.Enabled = True
End Sub
```