



**NEAR EAST UNIVERSITY**

**Faculty of Engineering**

**Department of Computer Engineering**

**ACCOUNTING SOFTWARE USING C#**

**GRADUATION PROJECT**

**COM 491**

**Submitted by : Ender Keskin (20112097)**

**Supervisor : Assist. Prof. Dr. Besime Erin**

**Nicosia – 2014**



## **ACKNOWLEDGEMENTS**

First, I would like to thank my supervisor Asist. Prof. Dr. Besime Erin for his invaluable advice and belief in my work and myself over the course of this Graduation Project.

Second, I would like to express my gratitude to Near East University for the scholarship that made the work possible.

Third, I thank my family and Ahmet Eker for their constant encouragement and support during the preparation of this project.

Finally, I would also like to thank all my friends for their advice and support.

## ABSTRACT

The purpose of this Project is to find solution of accounting and use it by companies around the world. This application has been developed for small and medium-sized businesses.

Accounting software project is implemented C # programming language using Visual Studio 2010. Datas keep in Sql Server 2008. First of all, users must be login to do transactions. All transactions do on own account. So, confusion is avoided. Another features of the software, each modules are in a separate tab. The following are this modules.

- Stock
- Current
- Invoice
- Safe
- Bank
- Cheque and Bond
- Reports
- Users
- Extras
- Support

You can do all your transactions in the project. Here we provide accounting software project abstract, screenshots, work flow diagrams and final document.

# **TABLE OF CONTENT**

## **ACKNOWLEDGEMENT**

## **ABSTRACT**

## **TABLE OF CONTENT**

## **LIST OF FIGURES**

## **INTRODUCTION.....1**

## **CHAPTER ONE : OVERVIEW**

### **1.Overview.....2**

#### **1.1. Accounting Information System.....2**

##### **1.1.1. History.....2**

##### **1.1.2. Software architecture of modern AIS.....2**

##### **1.1.3. Advantages and implication of AIS.....3**

##### **1.1.4. How to effectively implement AIS.....4**

#### **1.2. What is Accounting Software?.....7**

##### **1.2.1. The History of Computerized Accounting.....7**

##### **1.2.2. Advantages and Disadvantages of Computerized Accounting.....9**

## **CHAPTER TWO : TECNICAL OVERVIEW**

### **2. Techical Overview .....10**

#### **2.1. Microsoft Visual Studio.....10**

##### **2.1.1. Architecture.....10**

##### **2.1.2. Features.....12**

##### **2.1.3. Supported products.....16**

##### **2.1.4. History.....17**

#### **2.2. Microsoft SQL Server.....19**

##### **2.2.1 History.....19**

##### **2.2.2. Architecture.....19**

##### **2.2.3. Data storage.....20**



2.2.4. Services.....	22
2.2.5. Programmability.....	23
2.3. DevExpress.....	23
2.3.1. Product List.....	24

### **CHAPTER THREE : SYSTEM DESIGN AND REQUIREMENTS OF ACCOUNTING SOFTWARE**

3. Design.....	25
3.1 Database Design.....	25
3.2 System Design and Requirement.....	27
3.2.1 Data Flow Diagrams.....	27
3.2.2 Object Sequence Diagrams.....	32
3.2.3 Screenshots.....	33

### **CHAPTER FOUR : ACCOUNTING SOFTWARE CODE**

4. Codes.....	44
<b>CONCLUSION.....</b>	<b>146</b>
<b>REFERENCES.....</b>	<b>147</b>

## LIST OF FIGURES

Fig 1 : Microsoft Visual Studio Last Logo .....	10
Fig 2 : Visual Studio Web Designer in code editor view .....	13
Fig 3 : Visual Studio 2005 in Class Designer view .....	13
Fig 4 : History of Microsoft Visual Studio .....	18
Fig 5 : Tables of Sql Server .....	25
Fig 6 : View of Sql Server .....	26
Fig 7 : Login Activity Diagram .....	27
Fig 8 : Registry Activity Diagram .....	28
Fig 9 : Admin Activity Diagram .....	29
Fig 10 : User Activity Diagram .....	30
Fig 11 : Context Data Flow Diagram .....	31
Fig 12 : Login Data Flow Diagram .....	31
Fig 13 : Admin Login Object Sequence Diagram .....	32
Fig 14 : General Accounting Software Sequence Diagram .....	32
Fig 15 : User Panel .....	33
Fig 16 : Main Menu .....	33
Fig 17 : Stock Id Card .....	34
Fig 18 : Stock List .....	35
Fig 19 : Stock Transactions .....	35
Fig 20 : Customer Id Card .....	36
Fig 21 : Customer List .....	36
Fig 22 : Customer Transactions .....	37
Fig 23 : Sales Invoice .....	37
Fig 24 : Sales Invoice List .....	38
Fig 25 : Safe Id Card .....	38
Fig 26 : Safe List .....	39

Fig 27 : Safe Transactions .....	39
Fig 28 : Collecting Entry .....	40
Fig 29 : Payment Entry .....	40
Fig 30 : Bank Menu .....	41
Fig 31 : Cheque / Bond Menu .....	41
Fig 32 : Reports Menu .....	42
Fig 33 : User Menu .....	42
Fig 34 : Extras Menu .....	43
Fig 35 : Support Menu .....	43

## INTRODUCTION

The title of the project is "Accounting Software using C #". Accounting software is an application software that records and processes accounting transactions within functional modules such as accounting payable, accounting receivable, payroll and trial balance. It functions as an accounting information system.

Accounting software consists of computer applications and programs that all organizations, including nonprofits, businesses and government agencies, use to record and report financial information. The software may cover specific areas, such as accounts receivable or payable, or broad operating fields, such as departmental budgets or financial statements. Accounting software play a significant role in corporate decision-making processes. The software helps senior management plan budget work and financial accounting and reporting activities. Accounting managers also use the software to prepare financial statements, such as balance sheets, income statements, cash flow statements and equity statements. Organizations purchase accounting software based on operating activities, regulatory compliance requirements and financial reporting needs, according to Accounting Software Advisor, an accounting and information technology consulting firm. A small business typically purchases a limited-scope accounting software, while a large company generally needs an enterprise resource planning software. Narrow down accounting software choices by making a list of small business accounting features you need to run your business. Most of small business accounting features include:

- Inventory management
- Sales tracking
- Budgeting
- Estimates
- Payroll
- Business tax reporting

Advantages;

- Data Accuracy
- Time
- Inventory
- Reports



# **CHAPTER ONE : OVERVIEW**

## **1. Overview**

### **1.1. Accounting Information System**

An accounting information system (AIS) is a system of collection, storage and processing of financial and accounting data that is used by decision makers. An accounting information system is generally a computer-based method for tracking accounting activity in conjunction with information technology resources. The resulting statistical reports can be used internally by management or externally by other interested parties including investors, creditors and tax authorities. The actual physical devices and systems that allows the AIS to operate and perform its functions.

#### **1.1.1. History**

Initially, accounting information systems were predominantly developed “in-house” as legacy systems. Such solutions were difficult to develop and expensive to maintain. Today, accounting information systems are more commonly sold as prebuilt software packages from vendors such as Microsoft, Sage Group, SAP and Oracle where it is configured and customized to match the organization’s business processes. As the need for connectivity and consolidation between other business systems increased, accounting information systems were merged with larger, more centralized systems known as enterprise resource planning (ERP). Before, with separate applications to manage different business functions, organizations had to develop complex interfaces for the systems to communicate with each other. In ERP, a system such as accounting information system is built as a module integrated into a suite of applications that can include manufacturing, supply chain, human resources. These modules are integrated together and are able to access the same data and execute complex business processes. With the ubiquity of ERP for businesses, the term “accounting information system” has become much less about pure accounting (financial or managerial) and more about tracking processes across all domains of business.

#### **1.1.2. Software architecture of modern AIS**

A modern AIS typically follows a multitier architecture separating the presentation to the user, application processing and data management in distinct layers. The presentation layer manages how the information is displayed to and viewed by functional users of the system (through mobile devices, web browsers or client application). The entire system is backed by a centralized database that stores all of the data. This can include transactional data generated from the core business processes (purchasing, inventory, accounting) or static, master data that is referenced when processing data (employee and customer account records and configuration settings). As



transaction occur, the data is collected from the business events and stored into the system's database where it can be retrieved and processed into information that is useful for making decisions. The application layer retrieves the raw data held in the database layer, processes it based on the configured business logic and passes it onto the presentation layer to display to the users. For example, consider the accounts payable department when processing an invoice. With an accounting information system, an accounts payable clerk enters the invoice, provided by a vendor, into the system where it is then stored in the database. When goods from the vendor are received, a receipt is created and also entered into the AIS. Before the accounts payable department pays the vendor, the system's application processing tier performs a three-way matching where it automatically matches the amounts on the invoice against the amounts on the receipt and the initial purchase order. Once the match is complete, an email is sent to an accounts payable manager for approval. From here avoucher can be created and the vendor can ultimately be paid.

### **1.1.3. Advantages and implications of AIS**

A big advantage of computer-based accounting information systems is that they automate and streamline reporting. Reporting is major tool for organizations to accurately see summarized, timely information used for decision-making and financial reporting. The accounting information system pulls data from the centralized database, processes and transforms it and ultimately generates a summary of that data as information that can now be easily consumed and analyzed by business analysts, managers or other decision makers. These systems must ensure that the reports are timely so that decision-makers are not acting on old, irrelevant information and, rather, able to act quickly and effectively based on report results. Consolidation is one of the hallmarks of reporting as people do not have to look through an enormous number of transactions. For instance, at the end of the month, a financial accountant consolidates all the paid vouchers by running a report on the system. The system's application layer provides a report with the total amount paid to its vendors for that particular month. With large corporations that generate large volumes of transactional data, running reports with even an AIS can take days or even weeks.

After the wave of corporate scandals from large companies such as Tyco International, Enron and WorldCom, major emphasis was put on enforcing public companies to implement strong internal controls into their transaction-based systems. This was made into law with the passage of the Sarbanes Oxley Act of 2002 which stipulated that companies must generate an internal control report stating who is responsible for an organization's internal control structure and outlines the overall effectiveness of these controls. Since most of these scandals were rooted in the companies' accounting practices, much of the emphasis of Sarbanes Oxley was put on computer-based accounting information systems. Today, AIS vendors tout their governance, risk management, and compliance features to ensure business

processes are robust and protected and the organization's assets (including data) are secured.

#### **1.1.4. How to effectively implement AIS**

As stated above, accounting information systems are composed of six main components: When an AIS is initially implemented or converted from an existing system, organizations sometimes make the mistake of not considering each of these components and treating them equally in the implementation. This results in a system being "built three times" rather than once because the initial system is not designed to meet the needs of the organization. The organization then tries to get the system to work. Ultimately, the organization begins again, following the appropriate process.

Following a proven process that works, as follows, results in optimal deployment time, the least amount of frustration, and overall success. Most organizations, even larger ones, hire outside consultants, either from the software publisher or consultants who understand the organization and who work to help select and implement the ideal configuration, taking all components into consideration. Certified public accountants (CPAs) with careers dedicated to information systems work with companies to implement accounting information systems that follow a proven process. Many of these CPAs hold a certificate awarded by the American Institute of CPAs—the Certified Information Technology Professional (CITP). CITPs often serve as co-project managers with an organization's project manager representing the information technology (IT) department. In smaller organizations, a co-project manager may be an outsourced IT specialist who manages the implementation of the IT infrastructure.

The steps necessary to implement a successful accounting information system are as follows:

#### **Detailed Requirements Analysis**

Where all individuals involved in the system are interviewed. The current system is thoroughly understood, including problems, and complete documentation of the system transactions, reports, and questions that need to be answered are gathered. User needs that are not in the current system are outlined and documented. Users include everyone, from top management to data entry. The requirements analysis not only provides the developer with the specific needs, it also helps users accept the change. Users who have the opportunity to ask questions and provide input are much more confident and receptive of the change, than those who sit back and don't express their concerns.



## **Systems Design**

The analysis is thoroughly reviewed and a new system is created. The system that surrounds the system is often the most important. What data needs to go into the system and how is this going to be handled? What information needs to come out of the system? how is it going to be formatted? If we know what needs to come out, we know what we need to put into the system. The program we select will need to appropriately handle the process. The system is built with control files, sample master records, and the ability to perform processes on a test basis. The system is designed to include appropriate internal controls and to provide management with the information needed to make decisions. It is a goal of an accounting information system to provide information that is relevant, meaningful, reliable, useful, and current. To achieve this, the system is designed so that transactions are entered as they occur (either manually or electronically) and information is immediately available online for management.

Once the system is designed, an RFP is created detailing the requirements and fundamental design. Vendors are asked to respond to the proposal, to provide demonstrations of the product, and to specifically respond to the needs of the organization. Ideally, the vendor will input control files, sample master records, and be able to show how transactions are processed that result in the information that management needs to make decisions. An RFP for the information technology infrastructure follows the selection of the software product because the software product generally has specific requirements for infrastructure. Sometimes, the software and the infrastructure is selected from the same vendor. If not, the organization must ensure that vendors will work together without "pointing fingers" when there is an issue with either the software or the infrastructure.

## **Documentation**

As the system is being designed, it is documented. The documentation includes vendor documentation of the system and, more importantly, the procedures or detailed instructions that help users handle each process specific to the organization. Most documentation and procedures are online and it is helpful if organizations can add to the help instructions provided by the software vendor. Documentation and procedures tend to be an afterthought but is the insurance policy and the tool used during testing and training before launch. The documentation is tested during the training so that when the system is launched, there is no question that it works and that the users are confident with the change.

## **Testing**

Before launch, all processes are tested from input through output, using the documentation as a tool to ensure that all processes are thoroughly documented and that

users can easily follow the procedures: They know it works and that the procedures will be followed consistently. The reports are reviewed and verified, so that there's no garbage in-garbage out. This is done in a test system not yet fully populated with live data. Unfortunately, most organizations launch systems before thorough testing, adding to end-user frustration when processes don't work. The documentation and procedures may be modified during this process. All identified transactions must be tested during this step. All reports and online information must be verified and traced through the audit trail so that management is ensured that transactions will be handled consistently and that the information can be relied upon to make decisions.

## **Training**

Before launch, all users need to be trained, with procedures. This means a trainer using the procedures to show each end user how to handle a procedures. The procedures often need to be updated during training as users describe their unique circumstances and the "design" is modified with this additional information. The end user then performs the procedure with the trainer and the documentation. The end user then performs the procedure with the documentation alone. The end user is then on his or her own with the support, either in person or by phone, of the trainer or other support person. This is before data conversion.

## **Data Conversion**

Tools are developed to convert the data from the current system (which was documented in the requirements analysis) to the new system. The data is mapped from one system to the other and data files are created that will work with the tools that are developed. The conversion is thoroughly tested and verified before final conversion. There's a backup so it can be restarted, if necessary.

## **Launch**

The system is implemented only after all of the above is completed. The entire organization is aware of the launch date. Ideally, the current system is retained and often run in "parallel" until the new system is in full operation and working properly. With the current mass-market software used by thousands of companies and fundamentally proven to work, the "parallel" run that is mandatory with software tailor-made to a company is generally not done. This is only true, however, when the above process is followed, the system is thoroughly documented and tested, and users are trained before launch.



## **Tools**

Online resources are available to assist with strategic planning of accounting information systems. Information systems and financial forms aid in determining the specific needs of each organization, as well as assigning responsibility to principles involved.

## **Support**

The end users and managers have ongoing support available at all times. System upgrades follow a similar process and all users are thoroughly appraised of changes, upgraded in an efficient manner, and trained.

Many organizations chose to limit the time and money spent on the analysis, design, documentation, and training, and move right into software selection and implementation. If a detailed requirements analysis is performed with adequate time being spent on the analysis, the implementation and ongoing support will be minimal. Organizations that skip the steps to ensure the system meets their needs are often left with frustrated end users, costly support, and information that is not current or correct. Worse yet, these organizations build the system three times instead of once.[1]

## **1.2. What is Accounting Software?**

Computer programs that assist bookkeepers and accountants in recording and reporting on a firm's financial transactions. The functionality of accounting software differs from product to product. Larger firms may choose to implement a customized solution which integrates a vast amount of data from many different departments. Smaller firms often choose an off the shelf product.

### **1.2.1. The History of Computerized Accounting**

Accounting software has changed throughout history. Some computerized systems and software of historical importance include Ada's original computing machine, the IBM 9Pac, Peachtree software, and Quicken software.

### **The First Computing Machine**

The first computing machine ever created was used for accounting. Countess Ada Lovelace worked on a general purpose computing engine. Along with the mathematicians Charles Babbage and Louis Menebria, Ada's writings describe how the



computing engine operates and what it can calculate. As the first description of how to build a computer, this is the basis of computerized accounting.

### **IBM 9Pac**

After World War II, large-scale computerized accounting systems were available in the United States. The IBM 9Pac was one of the first programming systems used by business employees that did not require large amounts of specialized programming knowledge. A 9Pac manual from 1961 explains that this system could be used to add up the amounts of sales recorded by each salesman at a company. Many modern accounting systems are derivatives of IBM's computerized accounting software.

### **SAP**

SAP is a German software company, founded by German computer programmers who left IBM. SAP focuses on large-scale business software. In 1973, the company offered its first software release, called SAP R/1. SAP added additional elements to its software packages to create systems to aid in executive decision making, as well as components that allow automated financial transactions. SAP's original customers were German industrial corporations, and the company is now one of the largest worldwide vendors of computerized accounting software.

### **Peachtree**

Peachtree was one of the first publicly available computerized accounting programs. The company was founded in Atlanta, Georgia, which is famous for its peaches. Peachtree was first sold in 1976 at the Computer System Center, a computer store in Atlanta where the software creators worked. Since it was included with the original IBM PC as well as sold in stores, Peachtree was the first accounting program that was widely known among the general public.

### **Intuit**

Quicken is a computerized accounting program for personal finance. According to its creator, the business software firm Intuit, Quicken was introduced in 1983 and simplified functions such as reconciling checkbooks and tracking budgets. Intuit also introduced Quickbooks, a more comprehensive small business accounting program, and Turbo Tax, an automated program to calculate federal and state income taxes.[2]

### **1.2.2. Advantages & Disadvantages of Computerized Accounting**

Accounting is a crucial function for every business – large or small. There are two types of accounting systems, manual and computerized. Where accounting functions were once performed by hand by trained professionals, today there is a wealth of accounting software available to help anyone keep the necessary accounting records. Whether you chose to use an accountant to perform accounting manually or use a computerized system depends upon your specific accounting needs.

#### **Advantages That Can Save Money**

Computerized accounting systems offer several advantages for small businesses. Systems for small and medium sized businesses can be purchased off the shelf at low cost. These programs allow managers to see the company's financial position in "real time" and make adjustments to the business strategy as needed. Computerized systems can also provide instant reports on stock evaluation, profit and loss, customer accounts and payroll and sales analysis, again, allowing faster adjustments in your business strategy. In addition, transactions need to be input only once, and, with some training, anyone in the company can handle the inputting.

#### **Disadvantages**

Using a computerized accounting system comes with its own set of problems, such as the need to protect against data loss through power failure or viruses, and the danger of hackers stealing data. Computer fraud is also a concern, and you need to instigate a system of controls for who has access to the information, particularly customer information. If there is a security breach and data is stolen, management can be held personally liable for the loss of data. You also need to make sure that the data has been correctly entered into the system, as a mistake in data entry can throw off a whole set of data.[3]

## CHAPTER TWO : TECNICAL OVERVIEW

### 2. Techical Overview

#### 2.1. Microsoft Visual Studio



**Fig 1: Microsoft Visual Studio Last Logo**

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs for Microsoft Windows superfamily of operating system as well as web sites, web application and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger.

Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level including adding support for source-control systems (like Subversion) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle. Visual Studio supports different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language specific service exists. Built-in languages include C, C++ and C++ / CLI, VB.net, C#, F#. Support for other languages such as M, Python and Ruby among others is available via language services installed separately. It also supports XML / XSLT, HTML / XHTML, JavaScript and CSS.

Microsoft provides "Express" editions of its Visual Studio at no cost. Commercial versions of Visual Studio along with select past versions are available for free to students via Microsoft's DreamSpark program.

##### 2.1.1. Architecture

Visual Studio does not support any programming language, solution or tool intrinsically, instead it allows the plugging of functionality coded as a VSPackage.



When installed, the functionality is available as a Service. The IDE provides three services: `SVsSolution`, which provides the ability to enumerate projects and solutions; `SVsUIShell`, which provides windowing and UI functionality and `SVsShell`, which deals with registration of `VSPackages`. In addition, the IDE is also responsible for coordinating and enabling communication between services. All editors, designers, project types and other tools are implemented as `VSPackages`. Visual Studio uses COM to access the `VSPackages`. The Visual Studio SDK also includes the Managed Package Framework (MPF), which is a set of managed wrappers around the COM interfaces that allow the Packages to be written in any CLI compliant language. However, MPF does not provide all the functionality exposed by the Visual Studio COM interfaces. The services can then be consumed for creation of other packages, which add functionality to the Visual Studio IDE. Support for programming languages is added by using a specific `VSPackage` called a Language Service.

A language service defines various interfaces which the `VSPackage` implementation can implement to add support for various functionalities. Functionalities that can be added this way include syntax coloring, statement completion, brace matching, parameter information tooltips, member lists and error markers for background compilation. If the interface is implemented, the functionality will be available for the language. Language services are to be implemented on a per-language basis. The implementations can reuse code from the parser or the compiler for the language. Language services can be implemented either in native code or managed code. For native code, either the native COM interfaces or the Babel Framework (part of Visual Studio SDK) can be used. For managed code, the MPF includes wrappers for writing managed language services. Visual Studio does not include any source control support built in but it defines two alternative ways for source control systems to integrate with the IDE. A Source Control `VSPackage` can provide its own customised user interface. In contrast, a source control plugin using the MSSCCI (Microsoft Source Code Control Interface) provides a set of functions that are used to implement various source control functionality, with a standard Visual Studio user interface. MSSCCI was first used to integrate Visual SourceSafe with Visual Studio 6.0 but was later opened up via the Visual Studio SDK. Visual Studio .NET 2002 used MSSCCI 1.1, and Visual Studio .NET 2003 used MSSCCI 1.2. Visual Studio 2005, 2008 and 2010 use MSSCCI Version 1.3, which adds support for rename and delete propagation as well as asynchronous opening. Visual Studio supports running multiple instances of the environment (each with its own set of `VSPackages`).

The instances use different registry hives (see MSDN's definition of the term "registry hive" in the sense used here) to store their configuration state and are differentiated by their `AppId` (Application ID). The instances are launched by an `AppId`-specific .exe that selects the `AppId`, sets the root hive and launches the IDE. `VSPackages` registered for one `AppId` are integrated with other `VSPackages` for that `AppId`. The various product editions of Visual Studio are created using the different `AppIds`. The Visual Studio Express edition products are installed with their own `AppIds`,

but the Standard, Professional and Team Suite products share the same AppId. Consequently, one can install the Express editions side-by-side with other editions, unlike the other editions which update the same installation. The professional edition includes a superset of the VSPackages in the standard edition and the team suite includes a superset of the VSPackages in both other editions. The AppId system is leveraged by the Visual Studio Shell in Visual Studio 2008.

### **2.1.2. Features**

#### **Code editor**

Like any other IDE, it includes a code editor that supports syntax highlighting and code completion using IntelliSense for not only variables, functions and methods but also language constructs like loops and queries. IntelliSense is supported for the included languages, as well as for XML and for Cascading Style Sheets and JavaScript when developing web sites and web applications. Autocomplete suggestions are popped up in a modeless list box, overlaid on top of the code editor. In Visual Studio 2008 onwards, it can be made temporarily semi-transparent to see the code obstructed by it. The code editor is used for all supported languages.

The Visual Studio code editor also supports setting bookmarks in code for quick navigation. Other navigational aids include collapsing code blocks and incremental search, in addition to normal text search and regex search. The code editor also includes a multi-item clipboard and a task list. The code editor supports code snippets, which are saved templates for repetitive code and can be inserted into code and customized for the project being worked on. A management tool for code snippets is built in as well. These tools are surfaced as floating windows which can be set to automatically hide when unused or docked to the side of the screen. The Visual Studio code editor also supports code refactoring including parameter reordering, variable and method renaming, interface extraction and encapsulation of class members inside properties, among others.

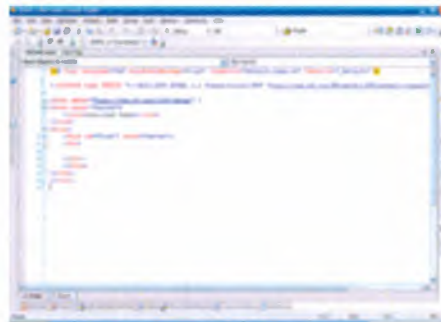
Visual Studio features background compilation. As code is being written, Visual Studio compiles it in the background in order to provide feedback about syntax and compilation errors, which are flagged with a red wavy underline. Warnings are marked with a green underline. Background compilation does not generate executable code, since it requires a different compiler than the one used to generate executable code. Background compilation was initially introduced with Microsoft Visual Basic but has now been expanded for all included languages.



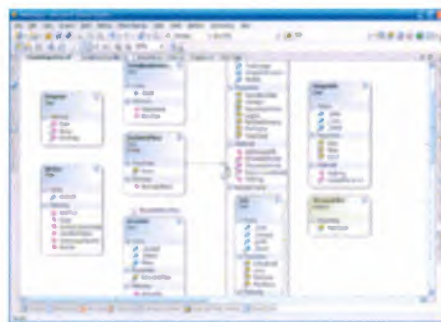
## Debugger

Visual Studio includes a debugger that works both as a source-level debugger and as a machine-level debugger. It works with both managed code as well as native code and can be used for debugging applications written in any language supported by Visual Studio. In addition, it can also attach to running processes and monitor and debug those processes. If source code for the running process is available, it displays the code as it is being run. If source code is not available, it can show the disassembly. The Visual Studio debugger can also create memory dumps as well as load them later for debugging. Multi-threaded programs are also supported. The debugger can be configured to be launched when an application running outside the Visual Studio environment crashes. The debugger allows setting breakpoints (which allow execution to be stopped temporarily at a certain position) and watches (which monitor the values of variables as the execution progresses). Breakpoints can be conditional, meaning they get triggered when the condition is met. Code can be stepped over, i.e., run one line (of source code) at a time. It can either step into functions to debug inside it, or step over it, i.e., the execution of the function body isn't available for manual inspection. The debugger supports Edit and Continue, i.e., it allows code to be edited as it is being debugged (32 bit only; not supported in 64 bit). When debugging, if the mouse pointer hovers over any variable, its current value is displayed in a tooltip ("data tooltips"), where it can also be modified if desired. During coding, the Visual Studio debugger lets certain functions be invoked manually from the Immediate tool window. The parameters to the method are supplied at the Immediate window.

## Designer



**Fig 2: Visual Studio Web Designer in code editor view**



**Fig 3: Visual Studio 2005 in Class Designer view**

Visual Studio includes a host of visual designers to aid in the development of applications. These tools include:

### **Windows Forms Designer**

The Windows Forms designer is used to build GUI applications using Windows Forms. Layout can be controlled by housing the controls inside other containers or locking them to the side of the form. Controls that display data (like textbox, list box, grid view, etc.) can be bound to data sources like databases or queries. Data-bound controls can be created by dragging items from the Data Sources window onto a design surface. The UI is linked with code using an event-driven programming model. The designer generates either C# or VB.NET code for the application.

### **WPF Designer**

The WPF designer, codenamed Cider was introduced with Visual Studio 2008. Like the Windows Forms designer it supports the drag and drop metaphor. It is used to author user interfaces targeting Windows Presentation Foundation.

It supports all WPF functionality including data binding and automatic layout management. It generates XAML code for the UI.

The generated XAML file is compatible with Microsoft Expression Design, the designer oriented product. The XAML code is linked with code using a code-behind model.

### **Web designer/development**

Visual Studio also includes a web-site editor and designer that allows web pages to be authored by dragging and dropping widgets. It is used for developing ASP.NET applications and supports HTML, CSS and JavaScript. It uses a code-behind model to link with ASP.NET code. From Visual Studio 2008 onwards, the layout engine used by the web designer is shared with Microsoft Expression Web. There is also ASP.NET MVC support for MVC technology as a separate download and ASP.NET Dynamic Data project available from Microsoft.

### **Class designer**

The Class Designer is used to author and edit the classes (including its members and their access) using UML modeling.



The Class Designer can generate C# and VB.NET code outlines for the classes and methods. It can also generate class diagrams from hand-written classes.

### **Data designer**

The data designer can be used to graphically edit database schemas, including typed tables, primary and foreign keys and constraints. It can also be used to design queries from the graphical view.

### **Mapping designer**

From Visual Studio 2008 onwards, the mapping designer is used by LINQ to SQL to design the mapping between database schemas and the classes that encapsulate the data. The new solution from ORM approach, ADO.NET Entity Framework, replaces and improves the old technology.

### **Extensibility**

Visual Studio allows developers to write extensions for Visual Studio to extend its capabilities. These extensions "plug into" Visual Studio and extend its functionality. Extensions come in the form of macros, add-ins, and packages. Macros represent repeatable tasks and actions that developers can record programmatically for saving, replaying, and distributing. Macros, however, cannot implement new commands or create tool windows. They are written using Visual Basic and are not compiled. Add-Ins provide access to the Visual Studio object model and can interact with the IDE tools. Add-Ins can be used to implement new functionality and can add new tool windows. Add-Ins are plugged into the IDE via COM and can be created in any COM-compliant languages. Packages are created using the Visual Studio SDK and provide the highest level of extensibility. They can create designers and other tools, as well as integrate other programming languages. The Visual Studio SDK provides unmanaged APIs as well as a managed API to accomplish these tasks. However, the managed API isn't as comprehensive as the unmanaged one. Extensions are supported in the Standard (and higher) versions of Visual Studio 2005. Express Editions do not support hosting extensions. Visual Studio 2008 introduced the Visual Studio Shell that allows for development of a customized version of the IDE. The Visual Studio Shell defines a set of VSPackages that provide the functionality required in any IDE. On top of that, other packages can be added to customize the installation. The Isolated mode of the shell creates a new AppId where the packages are installed. These are to be started with a different executable. It is aimed for development of custom development environments, either for a specific language or a specific scenario. The Integrated mode installs the packages into the AppId of the Professional/Standard/Team System editions, so that the

tools integrate into these editions. The Visual Studio Shell is available as a free download.

After the release of Visual Studio 2008, Microsoft created the Visual Studio Gallery. It serves as the central location for posting information about extensions to Visual Studio. Community developers as well as commercial developers can upload information about their extensions to Visual Studio .NET 2002 through Visual Studio 2010. Users of the site can rate and review the extensions to help assess the quality of extensions being posted. RSS feeds to notify users on updates to the site and tagging features are also planned.

### **2.1.3. Supported products**

#### **Microsoft Visual C++**

Microsoft Visual C++ is Microsoft's implementation of the C and C++ compiler and associated languages-services and specific tools for integration with the Visual Studio IDE. It can compile either in C mode or C++ mode. For C, it follows the ISO C standard with parts of C99 specification along with MS-specific additions in the form of libraries. For C++, it follows the ANSI C++ specification along with a few C++11 features. It also supports the C++/CLI specification to write managed code, as well as mixed-mode code (a mix of native and managed code). Microsoft positions Visual C++ for development in native code or in code that contains both native as well as managed components. Visual C++ supports COM as well as the MFC library. For MFC development, it provides a set of wizards for creating and customizing MFC boilerplate code, and creating GUI applications using MFC. Visual C++ can also use the Visual Studio forms designer to design UI graphically. Visual C++ can also be used with the Windows API. It also supports the use of intrinsic functions, which are functions recognized by the compiler itself and not implemented as a library. Intrinsic functions are used to expose the SSE instruction set of modern CPUs. Visual C++ also includes the OpenMP (version 2.0) specification.

#### **Microsoft Visual C#**

Microsoft Visual C#, Microsoft's implementation of the C# language, targets the .NET Framework, along with the language services that lets the Visual Studio IDE support C# projects. While the language services are a part of Visual Studio, the compiler is available separately as a part of the .NET Framework. The Visual C# 2008, 2010 and 2012 compilers support versions 3.0, 4.0 and 5.0 of the C# language specifications, respectively. Visual C# supports the Visual Studio Class designer, Forms designer, and Data designer among others.



## **Microsoft Visual Basic**

Microsoft Visual Basic is Microsoft's implementation of the VB.NET language and associated tools and language services. It was introduced with Visual Studio .NET (2002). Microsoft has positioned Visual Basic for Rapid Application Development. Visual Basic can be used to author both console applications as well as GUI applications. Like Visual C#, Visual Basic also supports the Visual Studio Class designer, Forms designer, and Data designer among others. Like C#, the VB.NET compiler is also available as a part of .NET Framework, but the language services that let VB.NET projects be developed with Visual Studio, are available as a part of the latter.

## **Microsoft Visual Web Developer**

Microsoft Visual Web Developer is used to create web sites, web applications and web services using ASP.NET. Either C# or VB.NET languages can be used. Visual Web Developer can use the Visual Studio Web Designer to graphically design web page layouts.

## **Team Foundation Server**

Included only with Visual Studio Team System, Team Foundation Server is intended for collaborative software development projects and acts as the server-side backend providing source control, data collection, reporting, and project-tracking functionality. It also includes the Team Explorer, the client tool for TFS services, which is integrated inside Visual Studio Team System.

### **2.1.4. History**

Prior to Visual Studio Version 4.0, there were Visual Basic 3, Visual C++, Visual FoxPro and Visual SourceSafe as separate products.



Product name	Codename	Internal version	Supported .NET Framework versions	Release date
Visual Studio	N/A	4.0	N/A	April 1995
Visual Studio 97	<i>Boston</i>	5.0	N/A	February 1997
Visual Studio 6.0	<i>Aspen</i>	6.0	N/A	June 1998
Visual Studio .NET (2002)	<i>Rainier</i>	7.0	1.0	February 13, 2002
Visual Studio .NET 2003	<i>Everett</i>	7.1	1.1	April 24, 2003
Visual Studio 2005	<i>Whidbey</i>	8.0	2.0, 3.0	November 7, 2005
Visual Studio 2008	<i>Orcas</i>	9.0	2.0, 3.0, 3.5	November 19, 2007
Visual Studio 2010	<i>Dev10/Rosario</i>	10.0	2.0, 3.0, 3.5, 4.0	April 12, 2010
Visual Studio 2012	<i>Dev11</i>	11.0	2.0, 3.0, 3.5, 4.0, 4.5	September 12, 2012
Visual Studio 2013	<i>Dev12</i>	12.0	2.0, 3.0, 3.5, 4.0, 4.5, 4.5.1	October 17, 2013

**Fig 4: History of Microsoft Visual Studio**

### Visual Studio Ultimate 2010

Visual Studio Ultimate 2010 (formerly Team System or Team Suite) is codenamed Rosario. It includes new modeling tools, such as the Architecture Explorer, which graphically displays projects and classes and the relationships between them. It supports UML activity diagram, component diagram, (logical) class diagram, sequence diagram, and use case diagram. Visual Studio Ultimate 2010 also includes Test Impact Analysis which provides hints on which test cases are impacted by modifications to the source code, without actually running the test cases. This speeds up testing by avoiding running unnecessary test cases.

Visual Studio Ultimate 2010 also includes a Historical Debugger for managed code called IntelliTrace. Unlike a traditional debugger, that records only the currently active stack, IntelliTrace records all events, such as prior function calls, method parameters, events and exceptions. This allows the code execution to be rewound in case a breakpoint wasn't set where the error occurred. Debugging with IntelliTrace will cause the application to run more slowly than debugging without it, and will use more memory as additional data needs to be recorded. Microsoft allows configuration of how much data should be recorded, in effect allowing developers to balance speed of execution and resource usage. TheLab Management component of Visual Studio Ultimate 2010 uses virtualization to create a similar execution environment for testers and developers. The virtual machines are tagged with checkpoints which can later be investigated for issues, as well as to reproduce the issue. Visual Studio Ultimate 2010 also includes the capability to record test runs that capture the specific state of the operating environment as well as the precise steps used to run the test. [4]

## **2.2. Microsoft SQL Server**

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. Its primary query languages are T-SQL and ANSI SQL.

### **2.2.1 History**

#### **SQL Server 2008 R2**

SQL Server 2008 R2 (10.50.1600.1, formerly codenamed "Kilimanjaro") was announced at TechEd 2009, and was released to manufacturing on April 21, 2010. SQL Server 2008 R2 adds certain features to SQL Server 2008 including a master data management system branded as Master Data Services, a central management of master data entities and hierarchies. Also Multi Server Management, a centralized console to manage multiple SQL Server 2008 instances and services including relational databases, Reporting Services, Analysis Services & Integration Services. SQL Server 2008 R2 includes a number of new services including PowerPivot for Excel and SharePoint, MasterDataServices, StreamInsight, Report Builder 3.0, Reporting Services Add-in for SharePoint, a Data-tier function in Visual Studio that enables packaging of tiered databases as part of an application, and a SQL Server Utility named UC (Utility Control Point), part of AMSM (Application and Multi-Server Management) that is used to manage multiple SQL Servers. The first SQL Server 2008 R2 service pack (10.50.2500, Service Pack 1) was released on July 11, 2011. The second SQL Server 2008 R2 service pack (10.50.4000, Service Pack 2) was released on July 26, 2012.

#### **2.2.2. Architecture**

The protocol layer implements the external interface to SQL Server. All operations that can be invoked on SQL Server are communicated to it via a Microsoft-defined format, called Tabular Data Stream (TDS). TDS is an application layer protocol, used to transfer data between a database server and a client. Initially designed and developed by Sybase Inc. for their Sybase SQL Server relational database engine in 1984, and later by Microsoft in Microsoft SQL Server, TDS packets can be encased in other physical transport dependent protocols, including TCP/IP, Named pipes, and Shared memory. Consequently, access to SQL Server is available over these protocols. In addition, the SQL Server API is also exposed over web services.



### 2.2.3. Data storage

Data storage is a database, which is a collection of tables with typed columns.

SQL Server supports different data types, including primary types such as Integer, Float, Decimal, Char, Varchar, binary, Text among others. The rounding of floats to integers uses either Symmetric Arithmetic Rounding or Symmetric Round Down (Fix) depending on arguments: `SELECT Round (2.5, 0)` gives 3. Microsoft SQL Server also allows user-defined composite types (UDTs) to be defined and used. It also makes server statistics available as virtual tables and views (called Dynamic Management Views or DMVs). In addition to tables, a database can also contain other objects including views, stored procedures, indexes and constraints, along with a transaction log. A SQL Server database can contain a maximum of 231 objects, and can span multiple OS-level files with a maximum file size of 260 bytes (1 exabyte). The data in the database are stored in primary data files with an extension .mdf. Secondary data files, identified with a .ndf extension, are used to store optional metadata. Log files are identified with the .ldf extension.

Storage space allocated to a database is divided into sequentially numbered pages, each 8 KB in size. A page is the basic unit of I/O for SQL Server operations. A page is marked with a 96-byte header which stores metadata about the page including the page number, page type, free space on the page and the ID of the object that owns it. Page type defines the data contained in the page - data stored in the database, index, allocation map which holds information about how pages are allocated to tables and indexes, change map which holds information about the changes made to other pages since last backup or logging, or contain large data types such as image or text. While page is the basic unit of an I/O operation, space is actually managed in terms of an extent which consists of 8 pages. A database object can either span all 8 pages in an extent ("uniform extent") or share an extent with up to 7 more objects ("mixed extent"). A row in a database table cannot span more than one page, so is limited to 8 KB in size. However, if the data exceeds 8 KB and the row contains Varchar or Varbinary data, the data in those columns are moved to a new page (or possibly a sequence of pages, called an Allocation unit) and replaced with a pointer to the data.

For physical storage of a table, its rows are divided into a series of partitions (numbered 1 to n). The partition size is user defined; by default all rows are in a single partition. A table is split into multiple partitions in order to spread a database over a cluster. Rows in each partition are stored in either B-tree or heap structure. If the table has an associated index to allow fast retrieval of rows, the rows are stored in-order according to their index values, with a B-tree providing the index. The data is in the leaf node of the leaves, and other nodes storing the index values for the leaf data reachable from the respective nodes. If the index is non-clustered, the rows are not sorted according to the index keys. An indexed view has the same storage structure as an indexed table. A table without an index is stored in an unordered heap structure. Both heaps and B-trees can span multiple allocation units.



## **Buffer management**

SQL Server buffers pages in RAM to minimize disc I/O. Any 8 KB page can be buffered in-memory, and the set of all pages currently buffered is called the buffer cache. The amount of memory available to SQL Server decides how many pages will be cached in memory. The buffer cache is managed by the Buffer Manager. Either reading from or writing to any page copies it to the buffer cache. Subsequent reads or writes are redirected to the in-memory copy, rather than the on-disc version. The page is updated on the disc by the Buffer Manager only if the in-memory cache has not been referenced for some time. While writing pages back to disc, asynchronous I/O is used whereby the I/O operation is done in a background thread so that other operations do not have to wait for the I/O operation to complete. Each page is written along with its checksum when it is written. When reading the page back, its checksum is computed again and matched with the stored version to ensure the page has not been damaged or tampered with in the meantime.

## **Concurrency and locking**

SQL Server allows multiple clients to use the same database concurrently. As such, it needs to control concurrent access to shared data, to ensure data integrity when multiple clients update the same data, or clients attempt to read data that is in the process of being changed by another client. SQL Server provides two modes of concurrency control: pessimistic concurrency and optimistic concurrency. When pessimistic concurrency control is being used, SQL Server controls concurrent access by using locks. Locks can be either shared or exclusive. Exclusive lock grants the user exclusive access to the data no other user can access the data as long as the lock is held. Shared locks are used when some data is being read multiple users can read from data locked with a shared lock, but not acquire an exclusive lock. The latter would have to wait for all shared locks to be released. Locks can be applied on different levels of granularity on entire tables, pages, or even on a per-row basis on tables. For indexes, it can either be on the entire index or on index leaves. The level of granularity to be used is defined on a per-database basis by the database administrator. While a fine grained locking system allows more users to use the table or index simultaneously, it requires more resources. So it does not automatically turn into higher performing solution. SQL Server also includes two more lightweight mutual exclusion solutions latches and spinlocks which are less robust than locks but are less resource intensive. SQL Server uses them for DMVs and other resources that are usually not busy. SQL Server also monitors all worker threads that acquire locks to ensure that they do not end up in deadlocks in case they do, SQL Server takes remedial measures, which in many cases is to kill one of the threads entangled in a deadlock and rollback the transaction it started. To implement locking, SQL Server contains the Lock Manager. The Lock Manager maintains an in-memory table that manages the database objects and locks, if any, on them along with other metadata

about the lock. Access to any shared object is mediated by the lock manager, which either grants access to the resource or blocks it.

SQL Server also provides the optimistic concurrency control mechanism, which is similar to the multiversion concurrency control used in other databases. The mechanism allows a new version of a row to be created whenever the row is updated, as opposed to overwriting the row, i.e., a row is additionally identified by the ID of the transaction that created the version of the row. Both the old as well as the new versions of the row are stored and maintained, though the old versions are moved out of the database into a system database identified as Tempdb. When a row is in the process of being updated, any other requests are not blocked but are executed on the older version of the row. If the other request is an update statement, it will result in two different versions of the rows both of them will be stored by the database, identified by their respective transaction IDs.

#### **2.2.4. Services**

SQL Server also includes an assortment of add-on services. While these are not essential for the operation of the database system, they provide value added services on top of the core database management system. These services either run as a part of some SQL Server component or out-of-process as Windows Service and presents their own API to control and interact with them.

#### **Visual Studio**

Microsoft Visual Studio includes native support for data programming with Microsoft SQL Server. It can be used to write and debug code to be executed by SQL CLR. It also includes a data designer that can be used to graphically create, view or edit database schemas. Queries can be created either visually or using code. SSMS 2008 onwards, provides intellisense for SQL queries as well.

#### **SQL Server Management Studio**

SQL Server Management Studio is a GUI tool included with SQL Server 2005 and later for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. SQL Server Management Studio replaces Enterprise Manager as the primary management interface for Microsoft SQL Server since SQL Server 2005. A version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as SQL Server Management Studio Express (SSMSE).



A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

### **2.2.5. Programmability**

#### **T-SQL**

T-SQL (Transact-SQL) is the Secondary means of programming and managing SQL Server. It exposes keywords for the operations that can be performed on SQL Server, including creating and altering database schemas, entering and editing data in the database as well as monitoring and managing the server itself. Client applications that consume data or manage the server will leverage SQL Server functionality by sending T-SQL queries and statements which are then processed by the server and results (or errors) returned to the client application. SQL Server allows it to be managed using T-SQL. For this it exposes read-only tables from which server statistics can be read. Management functionality is exposed via system-defined stored procedures which can be invoked from T-SQL queries to perform the management operation. It is also possible to create linked Server using T-SQL. Linked server allows operation to multiple server as one query.

#### **SQL Native Client**

SQL Native Client is the native client side data access library for Microsoft SQL Server, version 2005 onwards. It natively implements support for the SQL Server features including the Tabular Data Stream implementation, support for mirrored SQL Server databases, full support for all data types supported by SQL Server, asynchronous operations, query notifications, encryption support, as well as receiving multiple result sets in a single database session. SQL Native Client is used under the hood by SQL Server plug-ins for other data access technologies, including ADO or OLE DB. The SQL Native Client can also be directly used, bypassing the generic data access layers. On 28 Nov 2011 a preview release of the SQL Server ODBC driver for Linux was released.[5]

### **2.3. DevExpress**

DevExpress (Developer Express Inc) is a software development company founded in 1998 with headquarters in Glendale, California. DevExpress initially started producing UI



Controls for Borland Delphi/C++Builder and ActiveX Controls for Microsoft Visual Studio. Presently, DevExpress has products targeting developers that use Delphi/C++Builder, Visual Studio and HTML5/JavaScript technologies.

### **2.3.1. Product List**

- The DevExpress product line includes the following:
- WinForms Subscription - UI Controls for Windows Forms.
- ASP.NET Subscription - UI Controls for ASP.NET and ASP.NET MVC.
- WPF Subscription - UI Controls for Windows Presentation Foundation (WPF).
- Silverlight Subscription - UI Controls for Microsoft Silverlight.
- Windows 8 XAML Subscription - UI Controls for Windows RT.
- Reporting Subscription - Banded Reports and Text Processor Based Reporting Software for Microsoft Visual Studio Developers.
- Document Server - .NET Framework Library for processing files in XLS, DOC, RTF, ZIP formats and generating Barcodes.
- DevExtreme Mobile - HTML5 and JavaScript based Application Framework that can generate native applications for smartphones and tablets (Windows Phone, iOS and Android).
- CodeRush - Microsoft Visual Studio Add-ins.
- Universal Subscription - a package of all of the above, plus Dashboard software, Microsoft Coded UI Test functional testing support for Windows Forms Controls, Application Frameworks producing WinForms and ASP.NET front-ends
- Report Server - Server-based report generation software system.
- VCL Subscription - UI Controls for the VCL Framework used by Delphi and C++Builder developers.
- TestCafé - Functional Testing Tool for Web Developers. [6]

## CHAPTER THREE : SYSTEM DESIGN AND REQUIREMENTS OF ACCOUNTING SOFTWARE

### 3. Design

#### 3.1 Database Design

We have 22 Tables of all project.

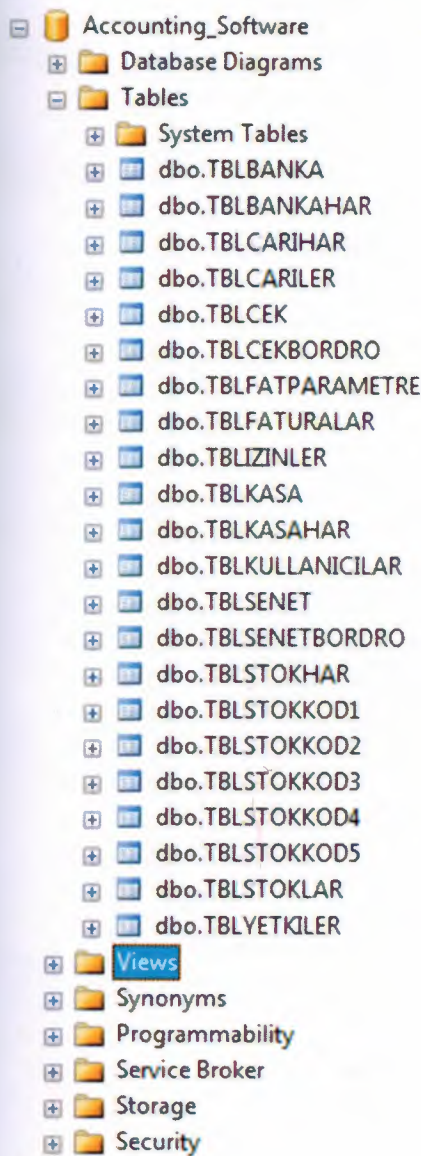
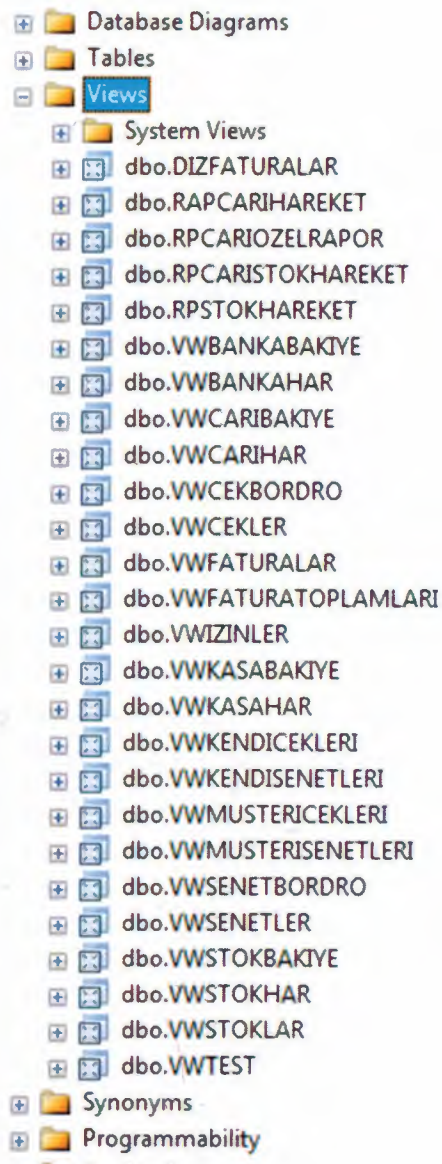


Fig 5 : Tables of Sql Server

We have 26 Views of all project.

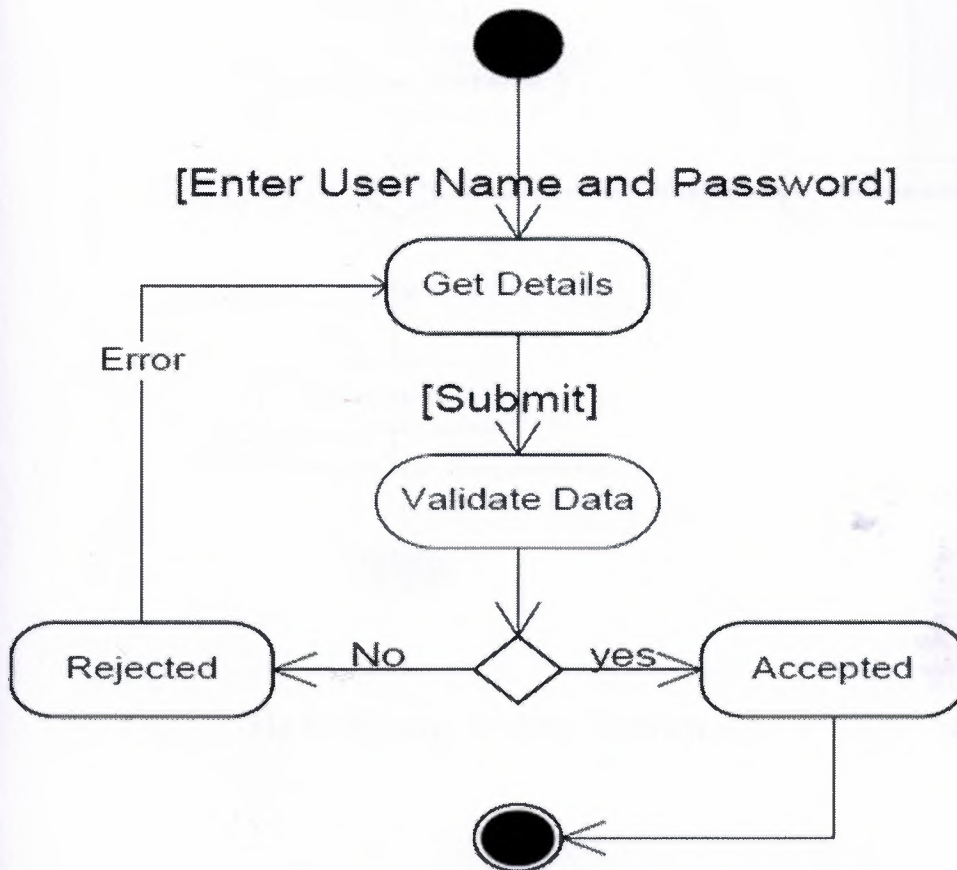


**Fig 6 : View of Sql Server**

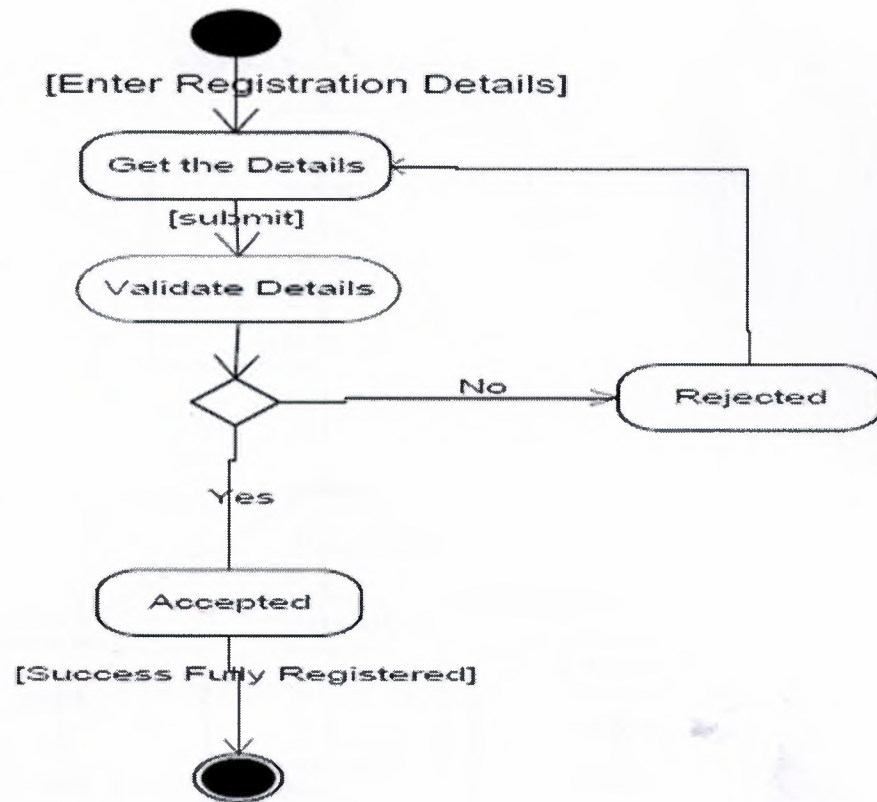


## 3.2 System Design and Requirement

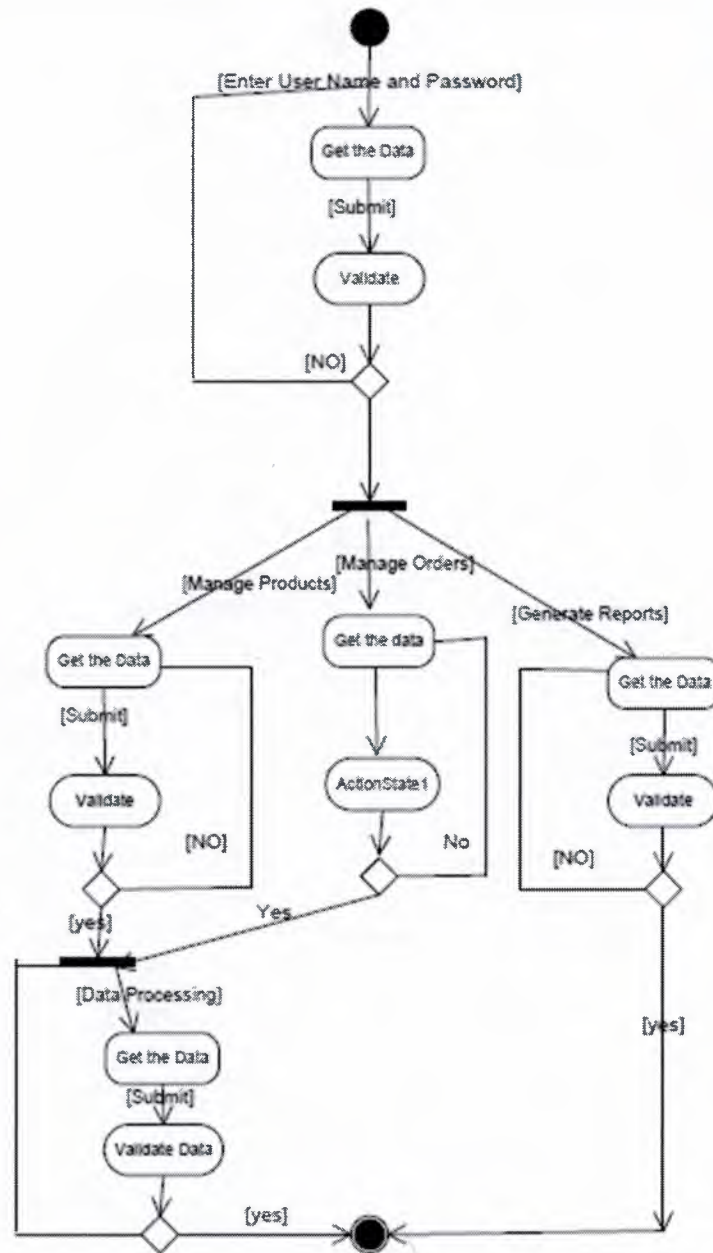
### 3.2.1 Data Flow Diagrams



**Fig 7 : Login Activity Diagram**

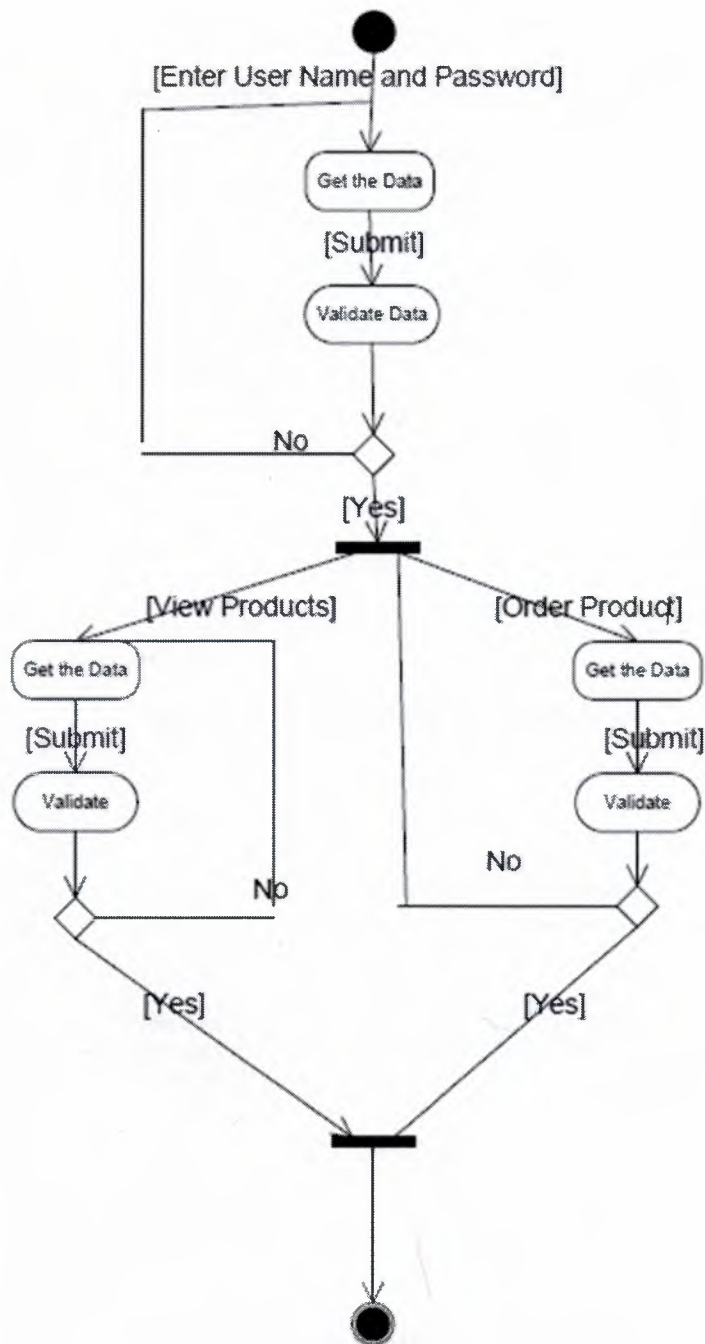


**Fig 8 : Register Activity Diagram**

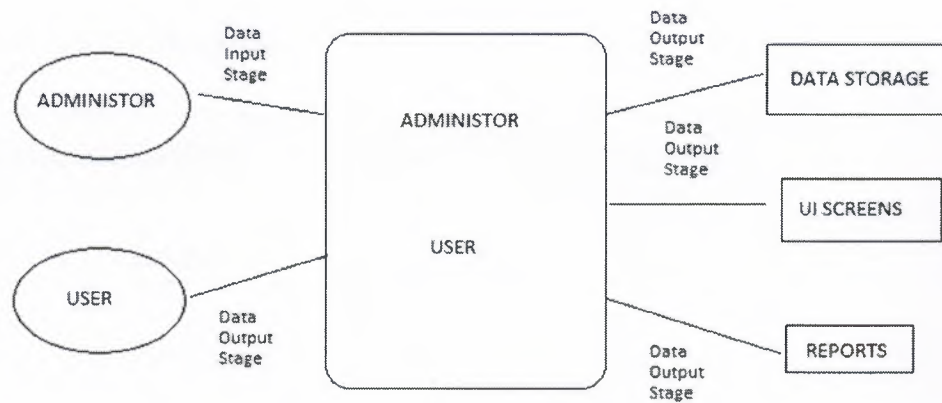


**Fig 9 : Admin Activity Diagram**

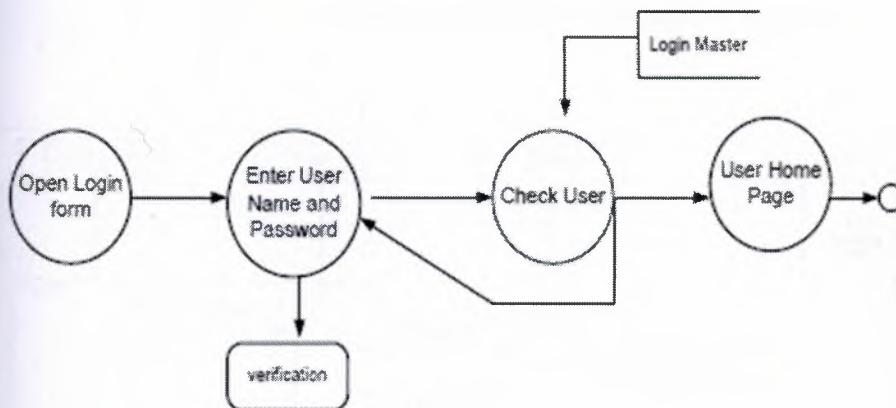




**Fig 10 : User Activity Diagram**



**Fig 11 : Context Data Flow Diagram**



**Fig 12 : Login Data Flow Diagram**

### 3.2.2 Object Sequence Diagrams

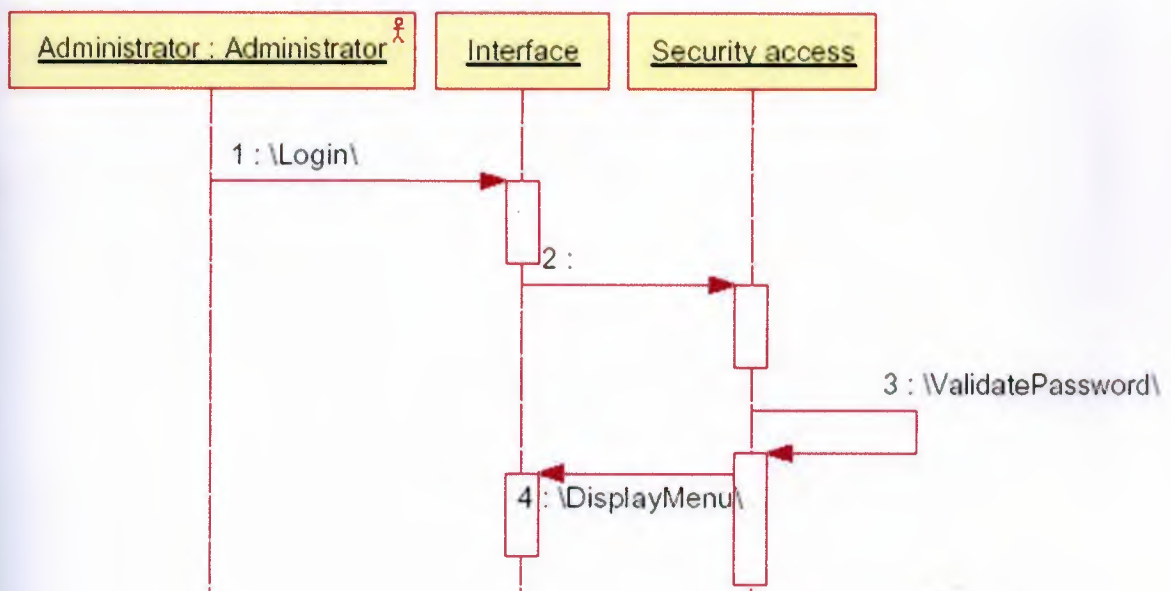


Fig 13 : Admin Login Object sequence Diagram

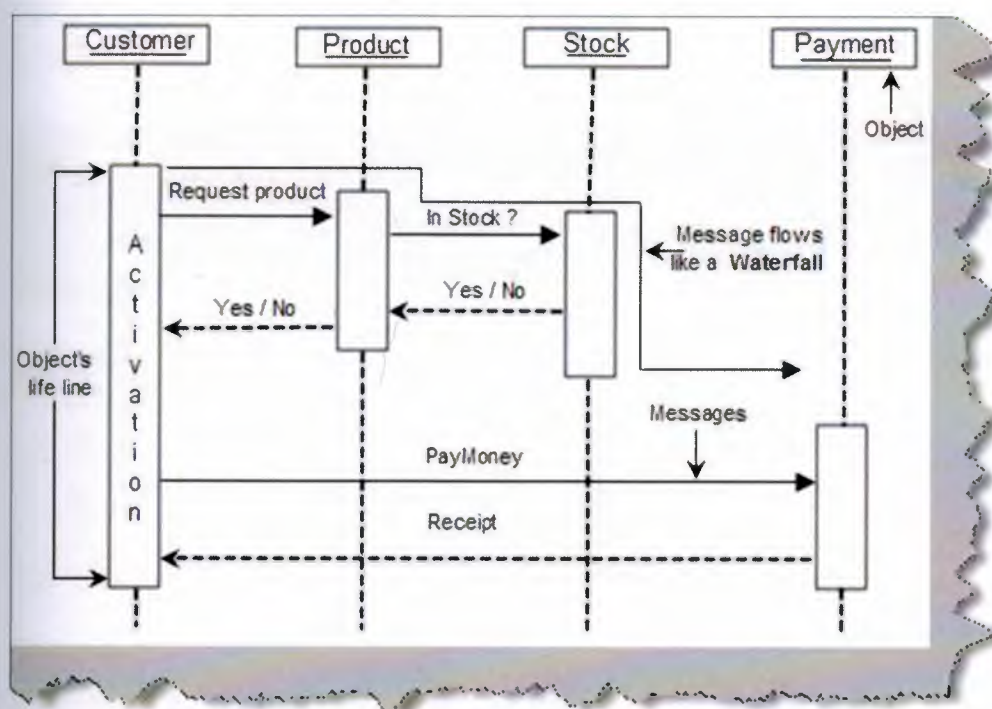


Fig 14 : General Accounting Software Sequence Diagram

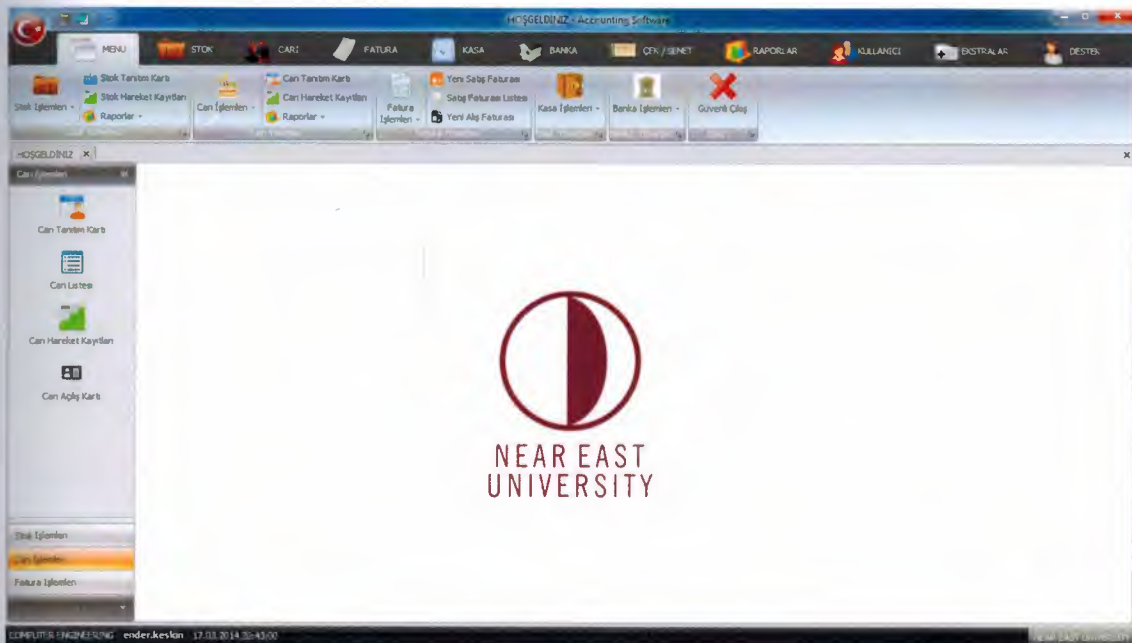


### 3.2.3 Screenshots



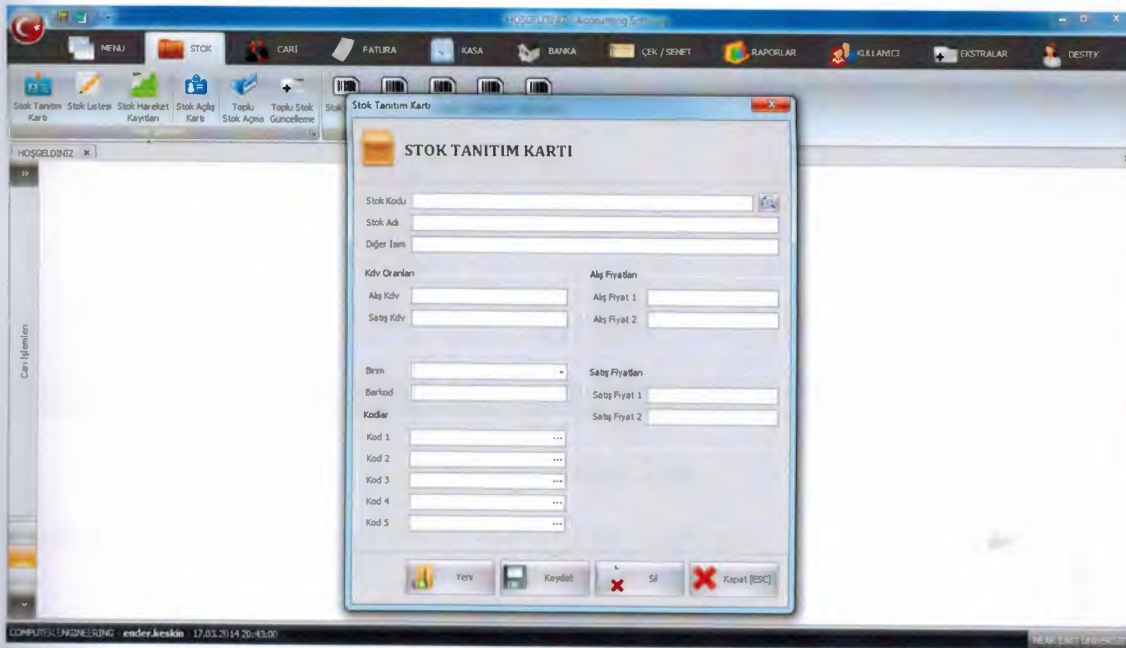
**Fig 15 : User Panel**

**User Panel :** When I opened the program, coming user panel. User name and password required to login to the system. I used a simple encryption algorithm. If you write true user name and password, entered into the system. Otherwise error message is displayed.



**Fig 16 : Main Menu**

**Main Menu :** It has 11 tab menus. Each tab menus have sub tab menus. There are shortcut keys of other menus on the main menu. Features of thi menu are ergonomic and more visually.



**Fig 17 : Stock Id Card**

**Stock Id Card :** This form is used to input the company stock. Stock Id Card makes add, update and delete operation.

STOK KODU	STOK ADI	SATIŞ FİY...	MİKTAR	KOD 1	KOD 2	KOD 3	KOD 4	KOD 5
ST00002	PORTAKAL	1,50	1,50	TOSHIBA	MAHAY	qweq	ASDASD	ASDAWRQ
ST00004	CEP TELEFONU	200,00	200,00	TOSHIBA	MAHAY	qweq	ASDASD	ASDAWRQ
STK5545454	FARE	20,00	20,00	TOSHIBA	MAHAY	qweq	ASDASD	ASDAWRQ
TEST	TEST	20,00	20,00	TOSHIBA	MAHAY	qweq	ASDASD	ASDAWRQ
21	PAZAR	21,00	21,00	PAZAR	PAZAR	PAZAR	PAZAR	PAZAR
9K0213215465555555	MONITOR	150,00	150,00	TURKIYE	TOSHIBA	17 INCH	CLIMARTESI	CLINA

**Fig 18 : Stock List**

**Stock List :** Stock list shows stocks added to the database.

STOK KODU	CARI İSİM	FATURA NO	TARİH	ÜC KOD	GİRİŞ	ÇIKIŞ	BRUT FİYAT	İSK 1(%)	İSK 2(%)	BİRİM FİYAT	TBİ
ST00002	ENDER KESKİN	SF0000000...	22.03.2014	G	10,00	0,00	200,00	10,00	10,00	162,00 S	
ST00004	ENDER KESKİN	SF0000000...	24.03.2014	G	1,00	0,00	200,00	10,00	10,00	162,00 S	
STK5545454	AHMET EKER	SF0000000...	09.04.2014	G	10,00	0,00	200,00	10,00	0,00	180,00 S	
TEST	AHMET EKER	SF0000000...	09.04.2014	G	100,00	0,00	200,00	10,00	5,00	171,00 S	
PAZAR	ENDER KESKİN	SF0000000...	09.04.2014	G	3,00	0,00	200,00	0,00	0,00	200,00 S	
MONITOR	AHMET EKER	SF0000000...	09.04.2014	G	120,00	0,00	171,00	10,00	5,00	146,21 S	
PORTAKAL	ENDER KESKİN	SF0000000...	23.04.2014	G	10,00	0,00	200,00	10,00	10,00	162,00 S	
CEP TELEFONU	ENDER KESKİN	SF0000000...	23.04.2014	G	150,00	0,00	171,00	10,00	10,00	138,51 S	
FARE	ENDER KESKİN	SF0000000...	23.04.2014	G	5,00	0,00	200,00	10,00	5,00	171,00 S	
TEST	ENDER KESKİN	SF0000000...	06.06.2014	C	0,00	100,00	250,00	10,00	5,00	213,75 S	
PAZAR	ENDER KESKİN	SF0000000...	06.06.2014	C	0,00	12,00	200,00	10,00	10,00	162,00 S	
MONITOR	ENDER KESKİN	SF0000000...	07.06.2014	C	0,00	124,00	250,00	0,00	0,00	250,00 S	
PORTAKAL	KESKİN YAZILIM	AF0000000...	08.06.2014	G	150,00	0,00	250,00	0,00	0,00	250,00 A	
CEP TELEFONU	KESKİN YAZILIM	AF0000000...	08.06.2014	G	10,00	0,00	100,00	0,00	0,00	100,00 A	
FARE	KESKİN YAZILIM	A10000000...	08.06.2014	C	0,00	10,00	100,00	0,00	0,00	100,00 A	
TEST	KESKİN YAZILIM	A10000000...	13.06.2014	C	0,00	25,00	100,00	0,00	0,00	100,00 A	

**Fig 19 : Stock Transactions**

**Stock Transactions :** It is a table showing where and how much of the stock is given.



**CARİ TANITIM KARTI**

Carı Kodu:

Carı Adı:

Adres:

İl/İlçe/İlçe:

Telefon/Fax:

Vergi Dairesi/No:

T.C. Kimlik No:

Posta Kodu:

E-Posta/Web:

Tip: ☐ Alıcı ☐ Satıcı ☐ Diğer

Yeni Kaydet Sil Kapat (ESC)

**Fig 20 : Customer Id Card**

**Customer Id Card :** Add customer information in the system. Customer Id Card makes add, update and delete operation.

**CARİ LİSTESİ**

CARİ KODU	CARİ ADI	İL	İLÇE	TELEFON	VERGİ DAİRESİ	TC	TİPİ
32132-454654-545	AHMET EKER	KARAMAN	HERVEZ	0230321	KARAMAN	51065165	SATICI
320-020-211-545	KESKİN YAZILIM	İSTANBUL	SULTANBEYLİ	02169920000	SULTANBEYLİ	1234567890	SATICI
TEST	TEST	TEST	TEST	TEST	TEST	TEST	ALICI

Arama:  Temale

**Fig 21 : Customer List**

**Customer List :** Customer List shows stocks added to the database.

BELGE NUMARASI	TARİH	VADE TARİHİ	BORÇ	ALACAK	BAKIYE	TİPİ	AÇIKLAMA
SP0000000001	08.06.2014	08.06.2014	0,00	44.750,00	44.750,00	A	
AP0000000002	08.06.2014	08.06.2014	0,00	1.180,00	45.930,00	A	
AP0000000002	08.06.2014	08.06.2014	1.180,00	0,00	44.750,00	A	
AP0000000003	08.06.2014	08.06.2014	0,00	11.000,00	55.750,00	A	
AP0000000003	08.06.2014	08.06.2014	11.000,00	0,00	44.750,00	A	
AJ0000000001	08.06.2014	08.06.2014	1.180,00	0,00	43.570,00	AI	
AJ0000000002	13.06.2014	13.06.2014	2.950,00	0,00	40.620,00	AI	
AJ0000000003	13.06.2014	13.06.2014	2.360,00	0,00	37.260,00	AI	
AJ0000000003	13.06.2014	13.06.2014	0,00	2.360,00	40.120,00	AI	
AJ0000000004	13.06.2014	13.06.2014	3.540,00	0,00	36.580,00	AI	
AJ0000000004	13.06.2014	13.06.2014	0,00	3.540,00	40.120,00	AI	
SF0000000033	16.06.2014	16.06.2014	236,00	0,00	39.884,00	S	
SF0000000034	16.06.2014	16.06.2014	2.360,00	0,00	37.524,00	S	
SF0000000034	16.06.2014	16.06.2014	0,00	2.360,00	39.884,00	S	

Toplam Alacak: 64690,00    Toplam Borç: 24806,00    Bakiye: 39884,00

**Fig 22 : Customer Transactions**

**Customer Transactions :** It shows the account transactions of customer. There are total balance in the bottom.

Fatura Numarası: SF0000000035    Cari Kodu:    Cari Ad:    Tarih: 17.06.2014    Vade Tarihi: 17.06.2014    Tutarı: ALISH FATURA

BARKOD	STOK KODU	STOK ADI	MİKTAR	BİRİM	BİRİM FİYAT	İSK 1 (%)	İSK 2 (%)	KDV (%)	TOPLAM
Click here to add a new row									

Record 0 of 0

Ara Toplam: 0,00  
İskonts: 0,00  
Net Fiyat: 0,00  
KDV: 0,00  
Genel Toplam: 0,00

**Fig 23 : Sales Invoice**

**Sales Invoice :** After the trading, the invoice process is done. We can take consisting invoice as printout.



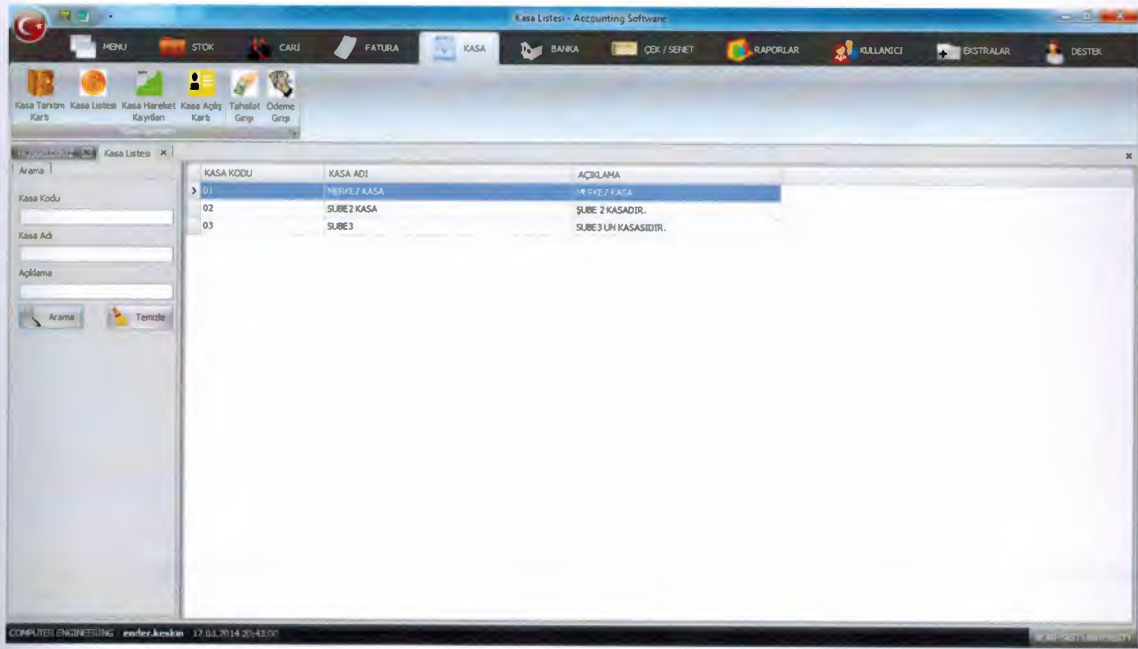
CARI KODU	CARI ADI	FATURA NO	TARİHİ	GENEL TOPLAM	TÜRÜ
10-03-2014	ENDER KESKİN	PT00001	22.03.2014	0,00 A	
10-03-2014	ENDER KESKİN	PT00001	22.03.2014	0,00 A	
10-03-2014	ENDER KESKİN	SF000000001	22.03.2014	0,00 A	
10-03-2014	ENDER KESKİN	SF000000000...	22.03.2014	0,00 A	
10-03-2014	ENDER KESKİN	SF000000000...	22.03.2014	0,00 K	
10-03-2014	ENDER KESKİN	SF000000000...	24.03.2014	411,48 A	
10-03-2014	ENDER KESKİN	SF000000000...	09.04.2014	708,00 A	
10-03-2014	ENDER KESKİN	SF000000000...	04.04.2014	1,77 A	
10-03-2014	ENDER KESKİN	SF000000000...	23.04.2014	1911,60 A	
10-03-2014	ENDER KESKİN	SF000000000...	23.04.2014	52772,31 A	
10-03-2014	ENDER KESKİN	SF000000000...	23.04.2014	3488,40 A	
10-03-2014	ENDER KESKİN	SF000000000...	04.06.2014	2057,40 K	
10-03-2014	ENDER KESKİN	SF000000000...	06.06.2014	16287,75 K	
10-03-2014	ENDER KESKİN	SF000000000...	06.06.2014	2293,92 K	
10-03-2014	ENDER KESKİN	SF000000000...	06.06.2014	3334,50 K	
10-03-2014	ENDER KESKİN	SF000000000...	07.06.2014	3450,44 A	
10-03-2014	ENDER KESKİN	SF000000000...	07.06.2014	36580,00 A	
32132-454654-545	AHMET EKER	SF000000000...	09.04.2014	21027,60 A	
32132-454654-545	AHMET EKER	SF000000000...	09.04.2014	4343,40 A	
32132-454654-545	AHMET EKER	SF000000000...	09.04.2014	1,77 A	
32132-454654-545	AHMET EKER	SF000000000...	09.04.2014	381,00 A	
32132-454654-545	AHMET EKER	SF000000000...	23.04.2014	1911,60 A	
32132-454654-545	AHMET EKER	SF000000000...	23.04.2014	1,77 K	
32132-454654-545	AHMET EKER	SF000000000...	24.04.2014	31275,90 A	
32132-454654-545	AHMET EKER	SF000000000...	04.06.2014	1597,43 A	

**Fig 24 : Sales Invoice List**

**Sales Invoice List :** This is where invoice is kept.

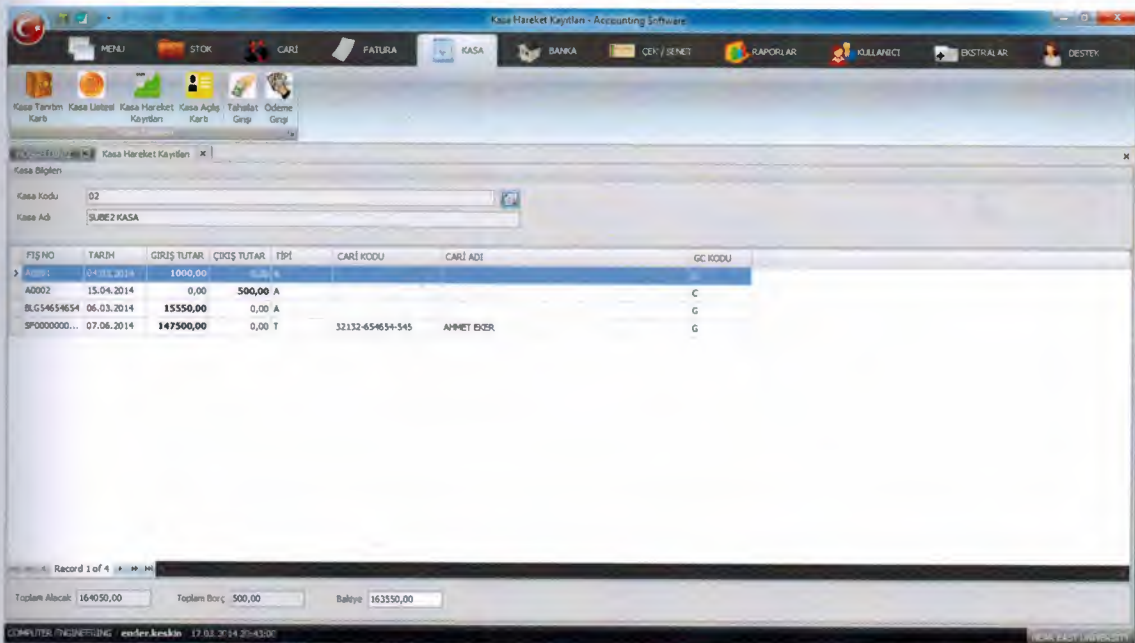
**Fig 25 : Safe Id Card**

**Safe Id Card :** Add safe information in the system.



**Fig 26 : Safe List**

**Safe List :** It shows stocks added to the database.



**Fig 27 : Safe Transactions**

**Safe Transactions :** It shows the account transactions of customer. There are total balance in the bottom.

**Tahsilat Giriş**

Belge No:

Makbuz No:

Kasa Kodu:

Kasa Adı:

Cari Kodu:

Cari Adı:

Tarih: 17.06.2014

Tutar: 147500,00

Açıklama:

Yeni Kaydet Sil Kapat [ESC]

PİS NO	TARİHİ	GİRİŞ TUTAR	ÇIKIŞ TUTAR	TİPİ	CARİ KOD
A0001	04.03.2014	1000,00	0,00	A	
A0002	15.04.2014	0,00	500,00	A	
BLG54654654	06.03.2014	15550,00	0,00	A	
SP0000000...	07.06.2014	147500,00	0,00	T	32132-48

Toplam Alacak: 164050,00 Toplam Borç: 500,00 Bakiye: 163550,00

ender.keşken 17.05.2014 20:43:07

**Fig 28 : Collecting Entry**

**Collecting Entry :** Payments from customer are made safe account with this form.

**Ödeme Giriş**

Belge No:

Makbuz No:

Kasa Kodu:

Kasa Adı:

Cari Kodu:

Cari Adı:

Tarih: 17.06.2014

Tutar: 147500,00

Açıklama:

Yeni Kaydet Sil Kapat [ESC]

PİS NO	TARİHİ	GİRİŞ TUTAR	ÇIKIŞ TUTAR	TİPİ	CARİ KOD
A0001	04.03.2014	1000,00	0,00	A	
A0002	15.04.2014	0,00	500,00	A	
BLG54654654	06.03.2014	15550,00	0,00	A	
SP0000000...	07.06.2014	147500,00	0,00	T	32132-48

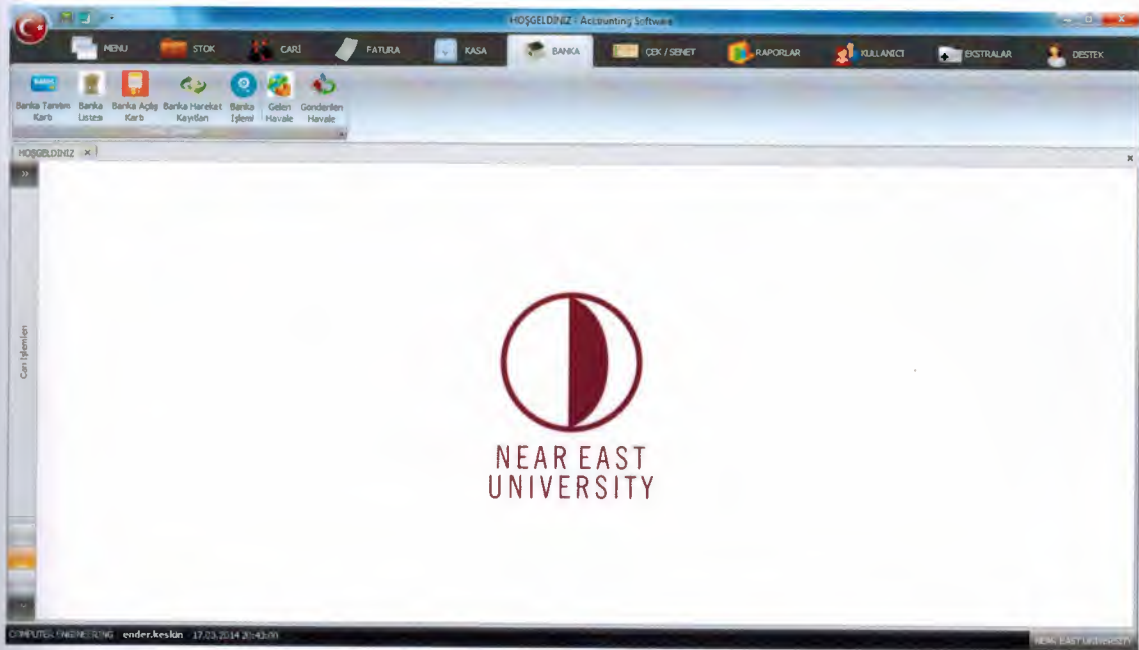
Toplam Alacak: 164050,00 Toplam Borç: 500,00 Bakiye: 163550,00

ender.keşken 17.05.2014 20:43:07

**Fig 29 : Payment Entry**

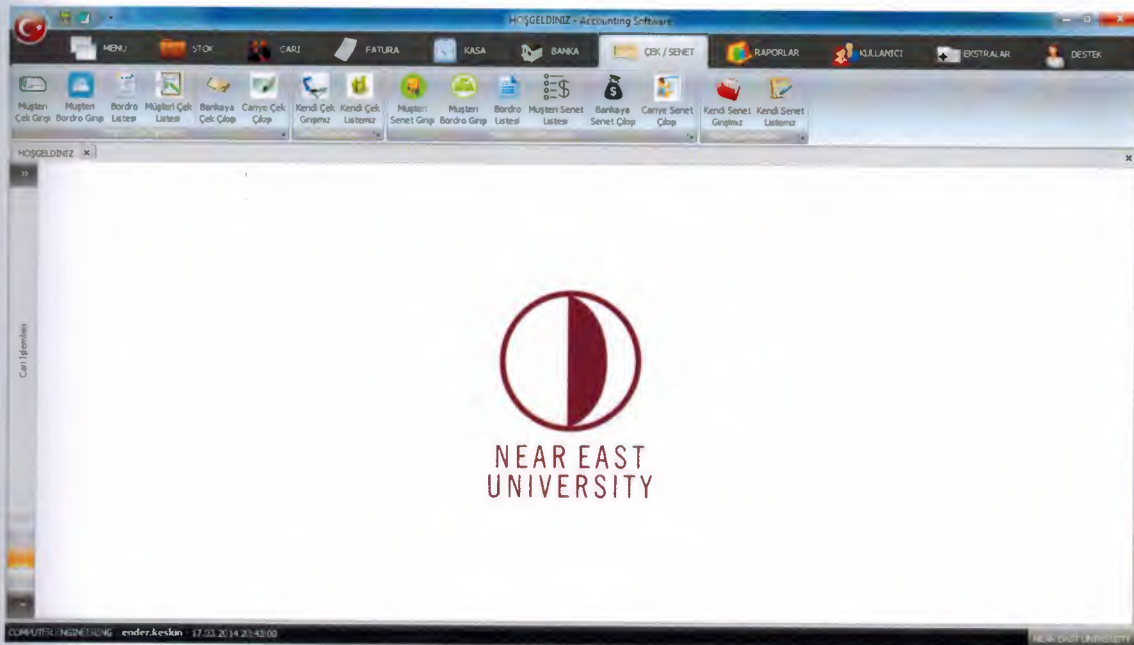
**Payment Entry :** Payments to Customer are made safe account.





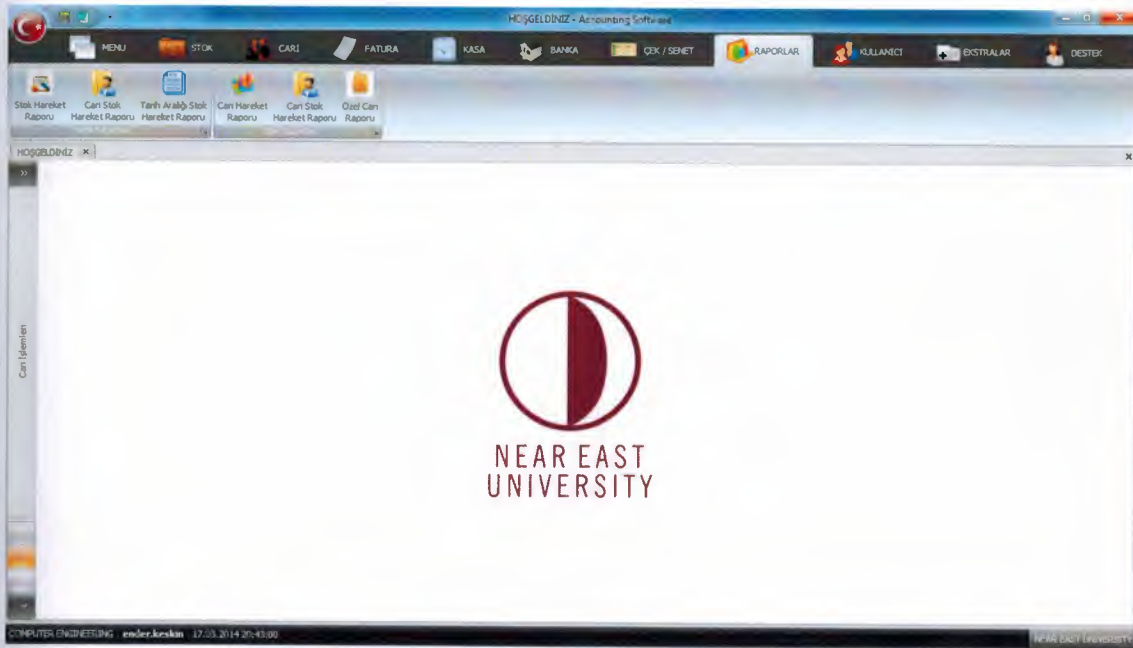
**Fig 30 : Bank Menu**

**Bank Menu :** This menu is all banking transaction. Adding bank, show bank in the database, bank transactions, money transfer operations.



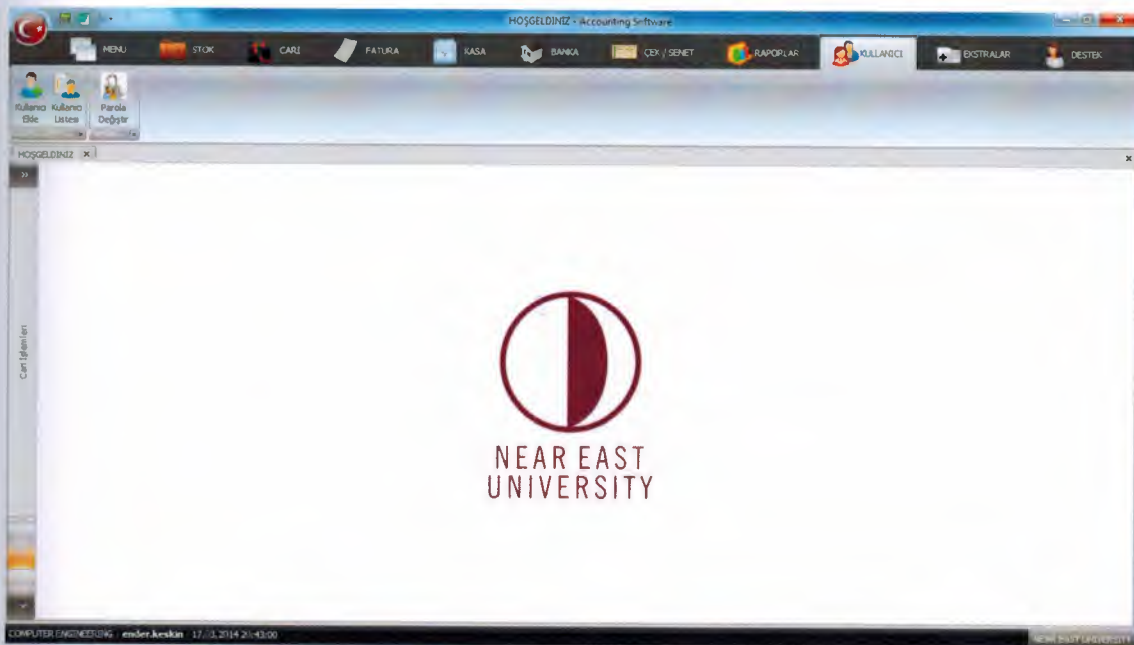
**Fig 31 : Cheque / Bond Menu**

**Cheque / Bond :** All Cheque / Bond. Cheque / Bond from customers, Cheque / Bond to customers, show all cheque and bond in the bank etc.



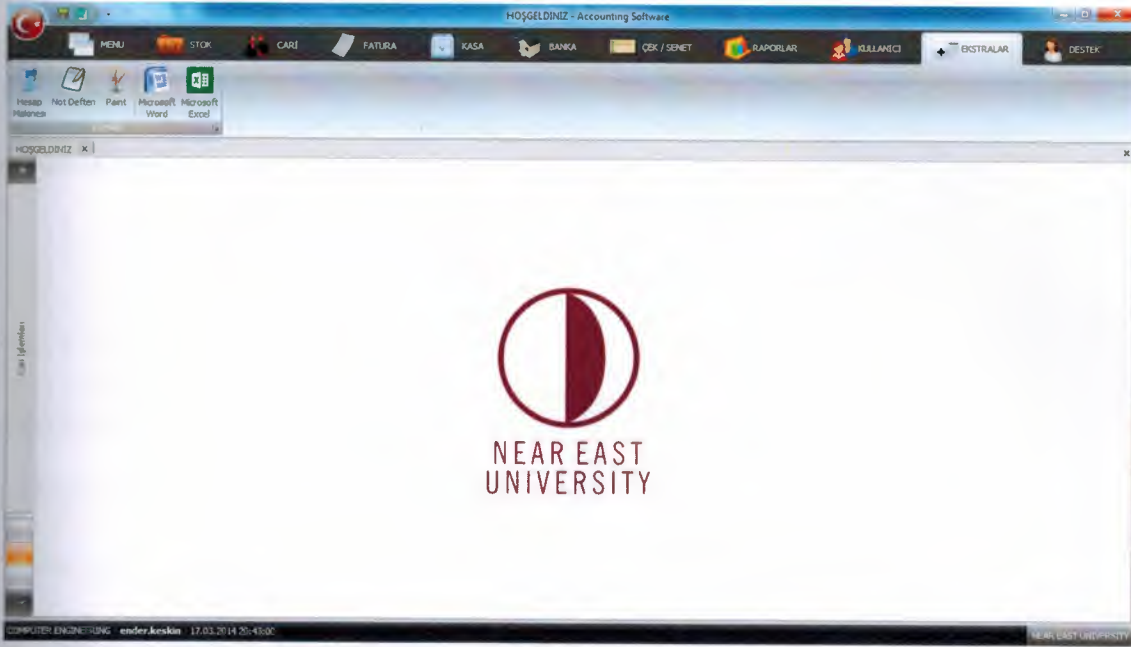
**Fig 32 : Reports Menu**

**Reports Menu :** We can see the reports of all transactions in the database. You can take printout of the reports as Excel, Pdf and Html.



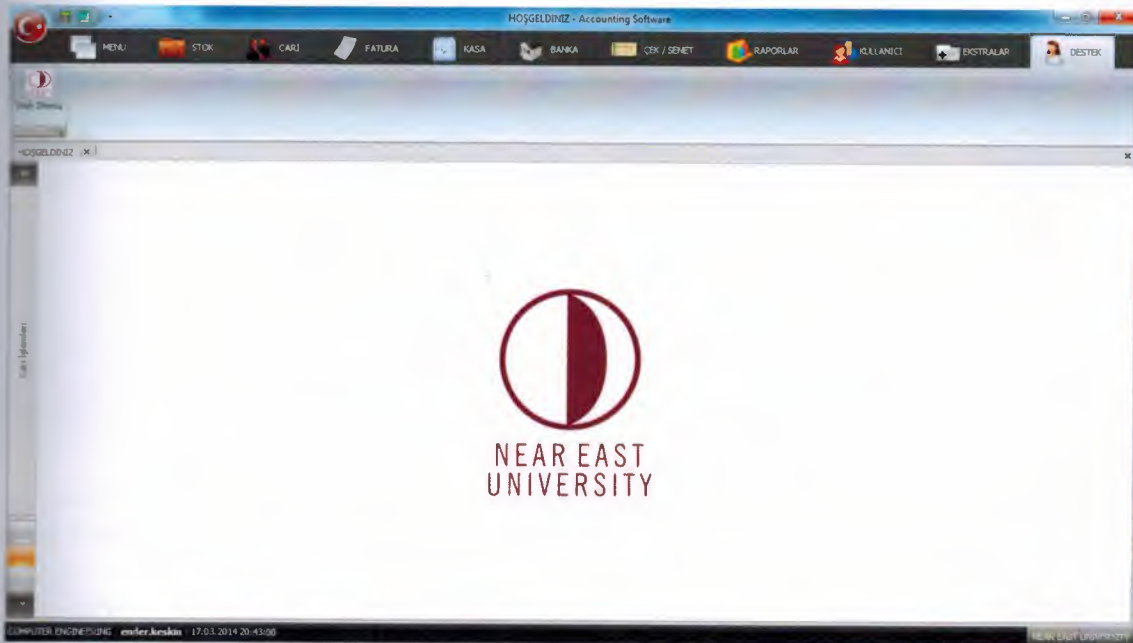
**Fig 33 : User Menu**

**User Menu :** User is logged into the system by admin. We can see registered Users from list. If necessary, change the password.



**Fig 34 : Extras Menu**

**Extras Menu :** Applications are required for accounting operation. For examples, calculator. Calculator is necessary for accounting.



**Fig 35 : Support Menu**

**Support Menu :** When there is a problem with application, you can contact us here.



## CHAPTER FOUR : ACCOUNTING SOFTWARE CODE

### 4. Codes

#### clsBanka.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.SqlClient;

namespace Accounting_Software.BankaModulu.Classlar
{
    class clsBanka
    {
        Accounting_Software.Classlar.Veritabanı VeriTabani = new
        Accounting_Software.Classlar.Veritabanı();

        public Boolean Ekle(string HesapNo, string Iban, string BankaAdi, string
        HesapAdi, string Sube, string Tel, string Ilgili, string Adres, string SaveUser)
        {
            try
            {
                string Sql = "EXECUTE SP_BANKA_EKLE " + HesapNo + "," + Iban +
                "," + BankaAdi + "," + HesapAdi + "," + Sube + "," + Tel + "," + Ilgili + "," +
                Adres + "," + SaveUser;
                VeriTabani.Isle(Sql);
                return true;
            }

            catch (Exception)
            {
                return false;
            }
        }

        public Boolean Guncelle(string Id, string HesapNo, string Iban, string BankaAdi,
        string HesapAdi, string Sube, string Tel, string Ilgili, string Adres, string SaveUser)
        {
            try
            {
                string Sql = "EXECUTE SP_BANKA_GUNCELLE " + Id + "," + HesapNo
                + "," + Iban + "," + BankaAdi + "," + HesapAdi + "," + Sube + "," + Tel + "," +
                Ilgili + "," + Adres + "," + SaveUser;
                VeriTabani.Isle(Sql);
                return true;
            }
        }
    }
}
```

```

        catch (Exception)
        {
            return false;
        }
    }

    public Boolean Sil(string Id)
    {
        try
        {
            string Sql = "EXECUTE SP_BANKA_SIL " + Id;
            VeriTabani.Isle(Sql);
            return true;
        }

        catch (Exception)
        {
            return false;
        }
    }

    public DataTable Listele(string HesapNo, string Iban, string BankaAdi, string HesapAdi, string Sube, string Ilgili)
    {
        string Sql = "SELECT * FROM TBLBANKA WHERE HESAPNO LIKE '%" + HesapNo + "%' AND IBAN LIKE '%" + Iban + "%' AND BANKAADI LIKE '%" + BankaAdi + "%' AND HESAPADI LIKE '%" + HesapAdi + "%' AND SUBE LIKE '%" + Sube + "%' AND ILGILI LIKE '%" + Ilgili + "%' ORDER BY BANKAADI";
        return VeriTabani.TabloCek(Sql);
    }

    public DataRow Ac(string Id)
    {
        string Sql = "SELECT TOP 1 * FROM TBLBANKA WHERE ID=" + Id;
        return VeriTabani.SatirCek(Sql);
    }
}

```

#### **frmBankaTanitimKarti.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

```

```

using DevExpress.XtraEditors;

namespace Accounting_Software.BankaModulu
{
    public partial class frmBankaTanitimKarti : DevExpress.XtraEditors.XtraForm
    {
        Accounting_Software.Classlar.Ekranlar Ekranlar = new
        Accounting_Software.Classlar.Ekranlar();
        Accounting_Software.Classlar.Mesajlar Mesajlar = new
        Accounting_Software.Classlar.Mesajlar();
        Accounting_Software.BankaModulu.Classlar.clsBankaHareketleri Hareketler =
        new Classlar.clsBankaHareketleri();
        Classlar.clsBanka Banka = new Classlar.clsBanka();

        Boolean Edit = false;
        string AcilanBankaId = "-";

        public frmBankaTanitimKarti()
        {
            InitializeComponent();
        }

        private void btnKaydet_Click(object sender, EventArgs e)
        {
            Kaydet();
        }

        void Kaydet()
        {
            if (Edit == true)
            {
                if (Banka.Guncelle(AcilanBankaId, txtHesapNo.Text, txtIban.Text,
                txtBankaAdi.Text, txtHesapAdi.Text, txtSube.Text, txtTelefon.Text, txtIlgili.Text,
                txtAdres.Text, frmAnaForm.UserId) == true)
                {
                    Mesajlar.Bilgi("Banka Tanıtım Kartı başarılı bir şekilde güncellenmiştir.");
                    Temizle();
                    this.Close();
                }

                else
                {
                    Mesajlar.Hata();
                }
            }

            else
            {

```



```

        if ( Banka.Ekle(txtHesapNo.Text, txtIban.Text, txtBankaAdi.Text,
txtHesapAdi.Text, txtSube.Text, txtTelefon.Text, txtIlgili.Text, txtAdres.Text,
frmAnaForm.UserId) == true)
        {
            Mesajlar.Bilgi("Banka Tanıtım Kartı başarılı bir şekilde kaydedilmiştir.");
            Temizle();
            this.Close();
        }

        else
        {
            Mesajlar.Hata();
        }
    }
}

```

```

void Temizle()
{
    txtAdres.Text = "";
    txtBankaAdi.Text = "";
    txtHesapAdi.Text = "";
    txtHesapNo.Text = "";
    txtIban.Text = "";
    txtIlgili.Text = "";
    txtSube.Text = "";
    txtTelefon.Text = "";
    Edit = false;
    txtHesapNo.Focus();
}

```

```

public void Ac(string Id)
{
    DataRow Satir = Banka.Ac(Id);

    AcilanBankaId = Id;

    txtAdres.Text = Satir["ADRES"].ToString();
    txtBankaAdi.Text = Satir["BANKAADI"].ToString();
    txtHesapAdi.Text = Satir["HESAPADI"].ToString();
    txtHesapNo.Text = Satir["HESAPNO"].ToString();
    txtIban.Text = Satir["IBAN"].ToString();
    txtIlgili.Text = Satir["ILGILI"].ToString();
    txtSube.Text = Satir["SUBE"].ToString();
    txtTelefon.Text = Satir["TEL"].ToString();

    Edit = true;

    BakiyeListesi(Id);
}

```

```

void BakiyeListesi(string Id)
{
    ListeBankaBakiye.DataSource = Hareketler.BakiyeListesi(Id);
}

private void txtHesapNo_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
{
    AcilanBankaId = Ekranlar.BankaListesi(true);
    Ac(AcilanBankaId);
}

private void btnSil_Click(object sender, EventArgs e)
{
    if (Mesajlar.Sor("Banka Tanıtım Kartını silmek istediğinizden emin misiniz?"
== true)
    {
        if (Banka.Sil(AcilanBankaId) == true)
        {
            Mesajlar.Bilgi("Banka başarılı bir şekilde silinmiştir.");
            Temizle();
        }

        else
        {
            Mesajlar.Hata();
        }
    }
}

private void frmBankaTanitimKarti_Load(object sender, EventArgs e)
{
}

private void btnKapat_Click(object sender, EventArgs e)
{
    this.Close();
}
}
}

```

#### **clsCariTanitimKarti.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;

```

```

using System.Data.SqlClient;

namespace Accounting_Software.CariModulu.Classlar
{
    class clsCariTanitimKarti
    {
        Accounting_Software.Classlar.Veritabanı VeriTabanı = new
        Accounting_Software.Classlar.Veritabanı();

        public DataTable Listele(string CariKodu, string CariAdi, string İl, string İlce,
        string Tc, string Tipi)
        {
            string Sql = "SELECT * FROM TBLCARILER WHERE CARIKODU LIKE
            '%" + CariKodu + "%' AND CARIISIM LIKE '%" + CariAdi + "%' AND IL LIKE '%"
            + İl + "%' AND ILCE LIKE '%" + İlce + "%' AND TC LIKE '%" + Tc + "%' AND TIPI
            LIKE '%" + Tipi + "%'";
            DataTable Tablo = VeriTabanı.TabloCek(Sql);
            return Tablo;
        }

        public void Ekle(string CariKodu, string CariAdi, string Adres, string İl, string
        İlce, string Ulke, string Telefon, string Fax, string VergiDairesi, string VergiNo, string
        Tc, string PostKodu, string EMail, string Web, string Tipi, string UserId)
        {
            string Sql = "EXECUTE SP_CARI_EKLE " + CariKodu + ", " + CariAdi + ",
            " + Adres + ", " + İl + ", " + İlce + ", " + Ulke + ", " + Telefon + ", " + Fax + ", " +
            EMail + ", " + Web + ", " + Tc + ", " + PostKodu + ", " + VergiDairesi + ", " +
            VergiNo + ", " + Tipi + ", " + UserId;
            VeriTabanı.Isle(Sql);
        }

        public void Guncelle(string Id, string CariKodu, string CariAdi, string Adres,
        string İl, string İlce, string Ulke, string Telefon, string Fax, string VergiDairesi, string
        VergiNo, string Tc, string PostKodu, string EMail, string Web, string Tipi, string
        UserId)
        {
            string Sql = "EXECUTE SP_CARI_GUNCELLE " + Id + ", " + CariKodu + ",
            " + CariAdi + ", " + Adres + ", " + İl + ", " + İlce + ", " + Ulke + ", " + Telefon + ",
            " + Fax + ", " + EMail + ", " + Web + ", " + Tc + ", " + PostKodu + ", " +
            VergiDairesi + ", " + VergiNo + ", " + Tipi + ", " + UserId;
            VeriTabanı.Isle(Sql);
        }

        public void Sil(string Id)
        {
            string Sql = "EXECUTE SP_CARI_SIL " + Id;
            VeriTabanı.Isle(Sql);
        }

        public DataRow Ac(string Id)

```



```

{
    string Sql = "SELECT TOP 1 * FROM TBLCARILER WHERE ID="+Id;
    return VeriTabani.SatirCek(Sql);
}

public string CariIdAl(string CariKodu)
{
    string Sql = "SELECT TOP 1 * FROM TBLCARILER WHERE
CARIKODU='" + CariKodu + "'";
    return VeriTabani.SatirCek(Sql)["ID"].ToString();
}
}
}

```

#### **frmCariTanitimKarti.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraEditors;

namespace Accounting_Software.CariModulu
{
    public partial class frmCariTanitimKarti : DevExpress.XtraEditors.XtraForm
    {
        Accounting_Software.Classlar.Mesajlar Mesajlar = new
Accounting_Software.Classlar.Mesajlar();
        Accounting_Software.Classlar.Ekranlar Ekranlar = new
Accounting_Software.Classlar.Ekranlar();
        Classlar.clsCariTanitimKarti Cari = new Classlar.clsCariTanitimKarti();
        string CariId;
        Boolean Edit = false;

        public frmCariTanitimKarti()
        {
            InitializeComponent();
        }

        private void frmCariTanitimKarti_Load(object sender, EventArgs e)
        {
        }

        private void btnKaydet_Click(object sender, EventArgs e)
        {
            string Tipi = "Alici";

```

```

if (raAlici.Checked == true)
{
    Tipi = "ALICI";
}

else if (raSatici.Checked == true)
{
    Tipi = "SATICI";
}

else
{
    Tipi = "DİĞER";
}

if (Edit == true)
{
    Cari.Guncelle(CariId, txtCariKodu.Text, txtCariAdi.Text, txtAdres.Text,
txtIl.Text, txtIlce.Text, txtUlke.Text, txtTelefon.Text, txtFax.Text, txtVergiDairesi.Text,
txtVergiNo.Text, txtTc.Text, txtPostaKodu.Text, txtEMail.Text, txtWeb.Text, Tipi,
frmAnaForm.UserId);
    Temizle();
}

else
{
    Cari.Ekle(txtCariKodu.Text, txtCariAdi.Text, txtAdres.Text, txtIl.Text,
txtIlce.Text, txtUlke.Text, txtTelefon.Text, txtFax.Text, txtVergiDairesi.Text,
txtVergiNo.Text, txtTc.Text, txtPostaKodu.Text, txtEMail.Text, txtWeb.Text, Tipi,
frmAnaForm.UserId);
    Temizle();
}

Edit = false;
Temizle();
}

void Temizle()
{
    txtCariKodu.Text = "";
    txtCariAdi.Text = "";
    txtAdres.Text = "";
    txtEMail.Text = "";
    txtFax.Text = "";
    txtIl.Text = "";
    txtIlce.Text = "";
    txtPostaKodu.Text = "";
    txtTc.Text = "";
    txtTelefon.Text = "";
}

```

```

txtUlke.Text = "";
txtVergiDairesi.Text = "";
txtVergiNo.Text = "";
txtWeb.Text = "";
}

private void btnCariSec_Click(object sender, EventArgs e)
{
    CariId = Ekranlar.CariListesiAc();
    DataRow Satir = Cari.Ac(CariId);
    txtCariKodu.Text = Satir["CARIKODU"].ToString();
    txtCariAdi.Text = Satir["CARIISIM"].ToString();
    txtAdres.Text = Satir["ADRES"].ToString();
    txtEMail.Text = Satir["EMAIL"].ToString();
    txtFax.Text = Satir["FAX"].ToString();
    txtIl.Text = Satir["IL"].ToString();
    txtIlce.Text = Satir["ILCE"].ToString();
    txtPostaKodu.Text = Satir["POSTAKODU"].ToString();
    txtTc.Text = Satir["TC"].ToString();
    txtTelefon.Text = Satir["TELEFON"].ToString();
    txtUlke.Text = Satir["ULKE"].ToString();
    txtVergiDairesi.Text = Satir["VERGIDAIRESI"].ToString();
    txtVergiNo.Text = Satir["VERGINUMARASI"].ToString();
    txtWeb.Text = Satir["WEB"].ToString();

```

```

    string Tipi = Satir["TIPI"].ToString();

```

```

    if (Tipi == "ALICI")
    {

```

```

        raAlici.Checked = true;
    }

```

```

    else if (Tipi == "SATICI")
    {

```

```

        raSatıcı.Checked = true;
    }

```

```

    else
    {

```

```

        raDiğer.Checked = true;
    }

```

```

    Edit = true;
}

```

```

private void btnSil_Click(object sender, EventArgs e)
{

```

```

    if (Mesajlar.Sor("Seçilen kaydı silmek istediğinizden emin misiniz?") == true)
    {

```

```

        Cari.Sil(CariId);
    }
}

```



```

        Temizle();
    }
}

private void btnYeni_Click(object sender, EventArgs e)
{
    txtCariKodu.Text = "";
    txtCariAdi.Text = "";
    txtAdres.Text = "";
    txtEMail.Text = "";
    txtFax.Text = "";
    txtIl.Text = "";
    txtIlce.Text = "";
    txtPostaKodu.Text = "";
    txtTc.Text = "";
    txtTelefon.Text = "";
    txtUlke.Text = "";
    txtVergiDairesi.Text = "";
    txtVergiNo.Text = "";
    txtWeb.Text = "";
    txtCariKodu.Focus();
}
}
}

```

#### **clsCekler.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;

namespace Accounting_Software.CekSenetModulu.Cek.Classlar
{
    class clsCekler
    {
        Accounting_Software.Classlar.Veritabani VeriTabani = new
        Accounting_Software.Classlar.Veritabani();

        public Boolean CekGirisi(string BelgeNo, string CekNo, string Tipi, string
        AlinanCari, string VerilenCari, string VerilenBanka, string AsilCiro, string AsilBorclu,
        string Tarih, string VadeTarihi, string Banka, string Sube, string HesapNo, string Tutar,
        string Aciklama, string Durumu, string TahsilEdildiMi, string BordroNo, string
        SaveUser)
        {
            string Sql = "EXECUTE SP_CEK_EKLE '" + BelgeNo + "', '" + CekNo + "', '"
            + Tipi + "', '" + AlinanCari + "', '" + VerilenCari + "', '" + VerilenBanka + "', '" +
            AsilCiro + "', '" + AsilBorclu + "', '" + Tarih + "', '" + VadeTarihi + "', '" + Banka + "', '"

```

```

+ Sube + ", " + HesapNo + ", " + Tutar.Replace(",", ".") + ", " + Aciklama + ", " +
Durumu + ", " + TahsilEdildiMi + ", " + BordroNo + ", " + SaveUser;
    return VeriTani.Isle(Sql);
}

```

```

public Boolean CekGirisiGuncelle(string Id, string BelgeNo, string CekNo, string
Tipi, string AlinanCari, string VerilenCari, string VerilenBanka, string AsilCiro, string
AsilBorclu, string Tarih, string VadeTarihi, string Banka, string Sube, string HesapNo,
string Tutar, string Aciklama, string Durumu, string TahsilEdildiMi, string BordroNo,
string SaveUser)
{
    string Sql = "EXECUTE SP_CEK_GUNCELLE " + Id + ", " + BelgeNo + ", " +
+ CekNo + ", " + Tipi + ", " + AlinanCari + ", " + VerilenCari + ", " + VerilenBanka
+ ", " + AsilCiro + ", " + AsilBorclu + ", " + Tarih + ", " + VadeTarihi + ", " +
Banka + ", " + Sube + ", " + HesapNo + ", " + Tutar.Replace(",", ".") + ", " +
Aciklama + ", " + Durumu + ", " + TahsilEdildiMi + ", " + BordroNo + ", " +
SaveUser;
    return VeriTani.Isle(Sql);
}

```

```

public Boolean Sil(string Id)
{
    try
    {
        string Sql = "EXECUTE SP_CEK_SIL " + Id;
        VeriTani.Isle(Sql);
        return true;
    }

    catch (Exception)
    {
        return false;
    }
}

```

```

public DataTable MusteriCekListesi(string BelgeNo, string CekNo, string
CariKodu, string CariAdi, string Tarih, string VadeTarihi, string Aciklama, string
Durum, Boolean Bankadakiler, Boolean Portfoydakiler, Boolean Cirodakiler)
{
    string _Banka = "";
    string _Portfoy = "";
    string _Ciro = "";
    string Yersorgusu = "";

    if (Bankadakiler == true)
    {
        _Banka = "B";
    }

    if (Portfoydakiler == true)

```

```

    {
        _Portfoy = "P";
    }

    if (Cirodakiler == true)
    {
        _Ciro = "C";
    }

    YerSorgusu = "AND (YERI IN(" + _Banka + ", " + _Portfoy + ", " + _Ciro +
    ")))";

    try
    {
        string Sql = "SELECT * FROM VWMUSTERICEKLERI WHERE
        BELGENO LIKE '%" + BelgeNo + "%' AND CEKNO LIKE '%" + CekNo + "%' AND
        ALINANCARI LIKE '%" + CariKodu + "%' AND ALINANCARIISIM LIKE '%" +
        CariAdi + "%' AND TARİH LIKE '%" + Tarih + "%' AND VADETARİHI LIKE '%" +
        VadeTarihi + "%' AND ACIKLAMA LIKE '%" + Aciklama + "%' AND DURUMU
        LIKE '%" + Durum + "%' " + YerSorgusu;
        return VeriTabani.TabloCek(Sql);
    }

    catch (Exception)
    {
        throw;
    }
}

public DataTable KendiCekListesi(string BelgeNo, string CekNo, string
VerilenCari, string VerilenBanka, string Tarih, string VadeTarihi, string Aciklama,
string Durum, Boolean Bankadakiler, Boolean Portfoydakiler, Boolean Cirodakiler)
{
    string _Banka = "";
    string _Portfoy = "";
    string _Ciro = "";
    string YerSorgusu = "";

    if (Bankadakiler == true)
    {
        _Banka = "B";
    }

    if (Portfoydakiler == true)
    {
        _Portfoy = "P";
    }

    if (Cirodakiler == true)
    {

```



```

        _Ciro = "C";
    }

    YerSorgusu = "AND (YERI IN(" + _Banka + ", " + _Portfoy + ", " + _Ciro +
    "))";

    try
    {
        string Sql = "SELECT * FROM VWKENDICEKLERI WHERE BELGENO
        LIKE '%" + BelgeNo + "%' AND CEKNO LIKE '%" + CekNo + "%' AND
        VERILENCARIISIM LIKE '%" + VerilenCari + "%' AND VERILENBANKAISIM
        LIKE '%" + VerilenBanka + "%' AND TARİH LIKE '%" + Tarih + "%' AND
        VADETARİHI LIKE '%" + VadeTarihi + "%' AND ACIKLAMA LIKE '%" +
        Aciklama + "%' AND DURUMU LIKE '%" + Durum + "%' " + YerSorgusu;
        return VeriTabani.TabloCek(Sql);
    }

    catch (Exception)
    {
        throw;
    }
}

public DataRow Ac(string Id)
{
    string Sql = "SELECT TOP 1 * FROM VWCEKLER WHERE ID=" + Id;
    return VeriTabani.SatirCek(Sql);
}

public void CekiBankayaVer(string Id, string BankaId, string Tarih, string
BelgeNo)
{
    string Sql = "EXECUTE SP_CEKIBANKAYAVER " + Id + ", " + BankaId + ",
"" + Tarih + "", "" + BelgeNo + """;
    VeriTabani.Isle(Sql);
}

public Boolean CekiBankadanIadeAl(string Id)
{
    try
    {
        string Sql = "EXECUTE SP_CEKIBANKADANIADEAL " + Id;
        VeriTabani.Isle(Sql);
        return true;
    }

    catch (Exception)
    {
        return false;
    }
}

```

```

    }

    public Boolean CekCaridenIadeAl(string Id)
    {
        try
        {
            string Sql = "EXECUTE SP_CEKICARIDENIADEAL " + Id;
            VeriTabani.Isle(Sql);
            return true;
        }

        catch (Exception)
        {
            return false;
        }
    }

    public void CekCariyeVer(string Id, string CariKodu, string Tarih, string
    BelgeNo)
    {
        string Sql = "EXECUTE SP_CEKICARIYEVER " + Id + ", '" + CariKodu + "',
        '" + Tarih + "', '" + BelgeNo + "'";
        VeriTabani.Isle(Sql);
    }

    public Boolean CekDurumu(string CekId, string Durumu)
    {
        try
        {
            string Sql = "EXECUTE SP_CEKINDURUMU " + CekId + ", '" + Durumu +
            VeriTabani.Isle(Sql);
            return true;
        }

        catch (Exception)
        {
            return false;
        }
    }

    public string CekTahsilatiYapilanKasa(string CekId)
    {
        try
        {
            string Sql = "SELECT KASAKODU FROM TBLKASAHAR WHERE
            EVRAKTURU='CEK' AND EVRAKID =" + CekId;
            DataRow Satir = VeriTabani.SatirCek(Sql);
            string KasaKodu = Satir["KASAKODU"].ToString();
            return KasaKodu;
        }
    }

```

```

    }

    catch (Exception)
    {
        return "";
    }
}

public string CekTahsilatiYapilanBanka(string CekId)
{
    try
    {
        string Sql = "SELECT BANKAID FROM TBLBANKAHAR WHERE
EVRAKTURU ='CEK' AND EVRAKID =" + CekId;
        DataRow Satir = VeriTabani.SatirCek(Sql);
        string BankaId = Satir["BANKAID"].ToString();
        return BankaId;
    }

    catch (Exception)
    {
        return "";
    }
}
}
}

```

#### **frmMusteriCeki.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraEditors;

namespace Accounting_Software.CekSenetModulu.Cek
{
    public partial class frmMusteriCeki : DevExpress.XtraEditors.XtraForm
    {
        Classlar.clsCekler Cekler = new Classlar.clsCekler();
        Accounting_Software.Classlar.Formatlar Formatlar = new
Accounting_Software.Classlar.Formatlar();
        Accounting_Software.Classlar.Mesajlar Mesajlar = new
Accounting_Software.Classlar.Mesajlar();
        Accounting_Software.CariModulu.Classlar.clsCariTanitimKarti Cari = new
CariModulu.Classlar.clsCariTanitimKarti();
    }
}

```



```
Accounting_Software.Classlar.Ekranlar Ekranlar = new  
Accounting_Software.Classlar.Ekranlar();
```

```
public MusteriCeki GirilenCek;  
public Boolean BordroIcinSec = false;  
string AsilMiCiroMu = "";  
string AcilanCekId = "-1";  
string CariId = "-1";  
Boolean Edit = false;
```

```
public frmMusteriCeki()  
{  
    InitializeComponent();  
}
```

```
public MusteriCeki Sec(string CariKodu, string CariAdi)  
{  
    BordroIcinSec = true;  
    txtCariKodu.Text = CariKodu;  
    txtCariAdi.Text = CariAdi;  
    txtCariKodu.Properties.ReadOnly = true;  
    txtCariAdi.Properties.ReadOnly = true;  
    raAsilCek.Checked = true;  
    Edit = false;  
    this.ShowDialog();  
    return GirilenCek;  
}
```

```
public MusteriCeki Guncelle(MusteriCeki Cek, string CariKodu, string CariAdi,  
Boolean BordroIcinMi, Boolean DialogAc = true)  
{  
    Edit = true;  
    BordroIcinSec = BordroIcinMi;  
    GirilenCek = Cek;  
    BilgileriTextlereAktar(CariKodu, CariAdi);  
  
    if (DialogAc == true)  
    {  
        this.ShowDialog();  
    }  
  
    return GirilenCek;  
}
```

```
void BilgileriTextlereAktar(string CariKodu, string CariAdi)  
{  
    txtAciklama.Text = GirilenCek.Aciklama;  
    txtAsilBorclu.Text = GirilenCek.AsilBorclu;  
    txtBanka.Text = GirilenCek.Banka;  
    txtBelgeNo.Text = GirilenCek.BelgeNo;
```

```

txtCariAdi.Text = CariAdi;
txtCariKodu.Text = CariKodu;
txtCekNo.Text = GirilenCek.CekNo;
txtHesapNo.Text = GirilenCek.HesapNo;
txtSube.Text = GirilenCek.Sube;
txtTutar.Text = GirilenCek.Tutar.ToString();
txtVadeTarihi.Text = GirilenCek.VadeTarihi;
txtTarih.Text = GirilenCek.Tarih;
}

private void frmMusteriCeki_Load(object sender, EventArgs e)
{
}

private void btnKaydet_Click(object sender, EventArgs e)
{
    if (BordroIcinSec == true)
    {
        BordrolleKaydet();
    }

    else
    {
        if (Mesajlar.Sor("Çeki kaydetmek istediğinizden emin misiniz?") == true)
        {
            if (Edit == true)
            {
                DirektGuncelle();
            }

            else
            {
                DirektKaydet();
            }
        }
    }
}

private void raAsilCek_CheckedChanged(object sender, EventArgs e)
{
    if (raAsilCek.Checked == true)
    {
        txtAsilBorclu.Properties.ReadOnly = true;
        txtAsilBorclu.Text = txtCariAdi.Text;
        AsilMiCiroMu = "A";
    }
}

private void raCiroCek_CheckedChanged(object sender, EventArgs e)

```

```

{
    if (raCiroCek.Checked == true)
    {
        txtAsilBorclu.Properties.ReadOnly = false;
        AsilMiCiroMu = "C";
    }
}

void DirekKaydet()
{
    if (txtCariKodu.Text == "")
    {
        Mesajlar.Uyari("Cari seçmeden kayıt işlemi yapamazsınız!!!");
        CariSec();
    }

    else
    {
        if (Cekler.CekGirisi(txtBelgeNo.Text, txtCekNo.Text, "M",
txtCariKodu.Text, "", "", AsilMiCiroMu, txtAsilBorclu.Text,
Formatlar.IngilizceTarihFormati(txtTarih.Text),
Formatlar.IngilizceTarihFormati(txtVadeTarihi.Text), txtBanka.Text, txtSube.Text,
txtHesapNo.Text, txtTutar.Text, txtAciklama.Text, "B", "", "", firmAnaForm.UserId) ==
true)
        {
            Mesajlar.Bilgi("Çek başarılı bir şekilde kaydedilmiştir.");
        }

        else
        {
            Mesajlar.Hata();
        }
    }
}

void DirekGuncelle()
{
    if (txtCariKodu.Text == "")
    {
        Mesajlar.Uyari("Cari seçmeden kayıt işlemi yapamazsınız!!!");
        CariSec();
    }

    else
    {
        if (Cekler.CekGirisiGuncelle(AcilanCekId, txtBelgeNo.Text, txtCekNo.Text,
"M", txtCariKodu.Text, "", "", AsilMiCiroMu, txtAsilBorclu.Text,
Formatlar.IngilizceTarihFormati(txtTarih.Text),
Formatlar.IngilizceTarihFormati(txtVadeTarihi.Text), txtBanka.Text, txtSube.Text,

```



```
txtHesapNo.Text, txtTutar.Text, txtAciklama.Text, "B", "", "", frmAnaForm.UserId) ==
true)
```

```
    {
        Mesajlar.Bilgi("Çek başarılı bir şekilde güncellenmiştir.");
    }

    else
    {
        Mesajlar.Hata();
    }
}
}
```

```
void BordrolleKaydet()
```

```
{
    string AsilCiro = "A";
    GirilenCek.Aciklama = txtAciklama.Text;
    GirilenCek.AsilBorclu = txtAsilBorclu.Text;
```

```
    if (raAsilCek.Checked == true)
    {
        AsilCiro = "C";
    }
}
```

```
GirilenCek.AsilCiro = AsilCiro;
GirilenCek.Banka = txtBanka.Text;
GirilenCek.BelgeNo = txtBelgeNo.Text;
GirilenCek.CekNo = txtCekNo.Text;
GirilenCek.HesapNo = txtHesapNo.Text;
GirilenCek.Sube = txtSube.Text;
```

```
try
{
    GirilenCek.Tutar = Convert.ToDecimal(txtTutar.Text);
}
}
```

```
catch (Exception)
{
    txtTutar.Text = "0.00";
    GirilenCek.Tutar = 0;
}
}
```

```
GirilenCek.VadeTarihi = txtVadeTarihi.Text;
GirilenCek.Tarih = txtTarih.Text;
GirilenCek.EklemeBasariliMi = true;
```

```
this.Close();
}
```

```
void CariSec()
```

```

{
    CariId = Ekranlar.CariListesiAc();
    DataRow Satir = Cari.Ac(CariId.ToString());
    txtCariKodu.Text = Satir["CARIKODU"].ToString();
    txtCariAdi.Text = Satir["CARIISIM"].ToString();
}

private void txtCariKodu_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
{
    if (BordroIcinSec == false)
    {
        CariSec();
    }
}

private void btnKapat_Click(object sender, EventArgs e)
{
    if (txtBelgeNo.Text != "" || txtCekNo.Text != "" || txtCariKodu.Text != "")
    {
        if (Mesajlar.Sor("İşlemi iptal etmek istediğinizden emin misiniz?") == true)
        {
            this.Close();
        }
    }
    else
    {
        this.Close();
    }
}

private void btnSil_Click(object sender, EventArgs e)
{
    if (Mesajlar.Sor("Çeki silmek istediğinizden emin misiniz?") == true)
    {
        if (BordroIcinSec == true)
        {
            GirilenCek.EklemeBasariliMi = false;
            this.Close();
        }
        else
        {
            Cekler.Sil(AcılanCekId);
            Mesajlar.Bilgi("Çekiniz başarılı bir şekilde silinmiştir.");
            this.Close();
        }
    }
}
}

```

```

public void Ac(string Id, Boolean Dialog = true)
{
    AcilanCekId = Id;

    DataRow Satir = Cekler.Ac(AcilanCekId);
    MusteriCeki AcilanCek = new MusteriCeki();
    string CariKodu, CariAdi;

    CariKodu = Satir["ALINANCARI"].ToString();
    CariAdi = Satir["ALINANCARIISIM"].ToString();
    AcilanCek.Aciklama = Satir["ACIKLAMA"].ToString();
    AcilanCek.AsilBorclu = Satir["ASILBORCLU"].ToString();
    AcilanCek.AsilCiro = Satir["ASILCIRO"].ToString();
    AcilanCek.Banka = Satir["BANKA"].ToString();
    AcilanCek.BelgeNo = Satir["BELGENO"].ToString();
    AcilanCek.CekNo = Satir["CEKNO"].ToString();
    AcilanCek.EklemeBasariliMi = true;
    AcilanCek.HesapNo = Satir["HESAPNO"].ToString();
    AcilanCek.Sube = Satir["SUBE"].ToString();
    AcilanCek.Tarih = Satir["TARIH"].ToString();
    AcilanCek.Tutar = Convert.ToDecimal(Satir["TUTAR"].ToString());
    AcilanCek.VadeTarihi = Satir["VADETARIHI"].ToString();

    Guncelle(AcilanCek, CariKodu, CariAdi, false, Dialog);
}

private void txtBelgeNo_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
{
    AcilanCekId = Ekranlar.MusteriCekListesi(true);
    Ac(AcilanCekId, false);
}
}

```

#### **clsSenetler.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;

namespace Accounting_Software.CekSenetModulu.Senet.Classlar
{
    class clsSenetler
    {

```



```
Accounting_Software.Classlar.Veritabani VeriTabani = new
Accounting_Software.Classlar.Veritabani();
```

```
public Boolean SenetGirisi(string BelgeNo, string SenetNo, string Tipi, string
AlinanCari, string VerilenCari, string VerilenBanka, string AsilCiro, string AsilBorclu,
string Tarih, string VadeTarihi, string Banka, string Sube, string HesapNo, string Tutar,
string Aciklama, string Durumu, string TahsilEdildiMi, string BordroNo, string
SaveUser)
{
    string Sql = "EXECUTE SP_SENET_EKLE " + BelgeNo + ", " + SenetNo +
    ", " + Tipi + ", " + AlinanCari + ", " + VerilenCari + ", " + VerilenBanka + ", " +
    AsilCiro + ", " + AsilBorclu + ", " + Tarih + ", " + VadeTarihi + ", " + Banka + ", " +
    Sube + ", " + HesapNo + ", " + Tutar.Replace(", ", ".") + ", " + Aciklama + ", " +
    Durumu + ", " + TahsilEdildiMi + ", " + BordroNo + ", " + SaveUser;
    return VeriTabani.Isle(Sql);
}
```

```
public Boolean SenetGirisiGuncelle(string Id, string BelgeNo, string SenetNo,
string Tipi, string AlinanCari, string VerilenCari, string VerilenBanka, string AsilCiro,
string AsilBorclu, string Tarih, string VadeTarihi, string Banka, string Sube, string
HesapNo, string Tutar, string Aciklama, string Durumu, string TahsilEdildiMi, string
BordroNo, string SaveUser)
{
    string Sql = "EXECUTE SP_SENET_GUNCELLE " + Id + ", " + BelgeNo + ",
    " + SenetNo + ", " + Tipi + ", " + AlinanCari + ", " + VerilenCari + ", " +
    VerilenBanka + ", " + AsilCiro + ", " + AsilBorclu + ", " + Tarih + ", " + VadeTarihi
    + ", " + Banka + ", " + Sube + ", " + HesapNo + ", " + Tutar.Replace(", ", ".") + ", " +
    Aciklama + ", " + Durumu + ", " + TahsilEdildiMi + ", " + BordroNo + ", " +
    SaveUser;
    return VeriTabani.Isle(Sql);
}
```

```
public Boolean Sil(string Id)
{
    try
    {
        string Sql = "EXECUTE SP_SENET_SIL " + Id;
        VeriTabani.Isle(Sql);
        return true;
    }

    catch (Exception)
    {
        return false;
    }
}
```

```
public DataTable MusteriSenetListesi(string BelgeNo, string SenetNo, string
CariKodu, string CariAdi, string Tarih, string VadeTarihi, string Aciklama, string
Durum, Boolean Bankadakiler, Boolean Portfoydakiler, Boolean Cirodakiler)
```

```

{
    string _Banka = "";
    string _Portfoy = "";
    string _Ciro = "";
    string YerSorgusu = "";

    if (Bankadakiler == true)
    {
        _Banka = "B";
    }

    if (Portfoydakiler == true)
    {
        _Portfoy = "P";
    }

    if (Cirodakiler == true)
    {
        _Ciro = "C";
    }

    YerSorgusu = "AND (YERI IN(" + _Banka + ", " + _Portfoy + ", " + _Ciro +
    "))";

    try
    {
        string Sql = "SELECT * FROM VWMUSTERISENETLERI WHERE
        BELGENO LIKE '%" + BelgeNo + "%' AND SENETNO LIKE '%" + SenetNo + "%'
        AND ALINANCARI LIKE '%" + CariKodu + "%' AND ALINANCARIISIM LIKE
        '%" + CariAdi + "%' AND TARİH LIKE '%" + Tarih + "%' AND VADETARİHI LIKE
        '%" + VadeTarihi + "%' AND ACIKLAMA LIKE '%" + Aciklama + "%' AND
        DURUMU LIKE '%" + Durum + "%' " + YerSorgusu;
        return VeriTabani.TabloCek(Sql);
    }

    catch (Exception)
    {
        throw;
    }
}

```

```

public DataTable KendiSenetListesi(string BelgeNo, string SenetNo, string
VerilenCari, string VerilenBanka, string Tarih, string VadeTarihi, string Aciklama,
string Durum, Boolean Bankadakiler, Boolean Portfoydakiler, Boolean Cirodakiler)
{
    string _Banka = "";
    string _Portfoy = "";
    string _Ciro = "";
    string YerSorgusu = "";

```

```

if (Bankadakiler == true)
{
    _Banka = "B";
}

if (Portfoydakiler == true)
{
    _Portfoy = "P";
}

if (Cirodakiler == true)
{
    _Ciro = "C";
}

YerSorgusu = "AND (YERI IN(" + _Banka + ", " + _Portfoy + ", " + _Ciro +
"))";

try
{
    string Sql = "SELECT * FROM VWKENDISENETLERI WHERE
BELGENO LIKE '%" + BelgeNo + "%' AND SENETNO LIKE '%" + SenetNo + "%'
AND VERILENCARIISIM LIKE '%" + VerilenCari + "%' AND
VERILENBANKAISIM LIKE '%" + VerilenBanka + "%' AND TARİH LIKE '%" +
Tarih + "%' AND VADETARİHI LIKE '%" + VadeTarihi + "%' AND ACIKLAMA
LIKE '%" + Aciklama + "%' AND DURUMU LIKE '%" + Durum + "%' " +
YerSorgusu;
    return VeriTabani.TabloCek(Sql);
}

catch (Exception)
{
    throw;
}

public DataRow Ac(string Id)
{
    string Sql = "SELECT TOP 1 * FROM VWSENETLER WHERE ID=" + Id;
    return VeriTabani.SatirCek(Sql);
}

public void SenetiBankayaVer(string Id, string BankaId, string Tarih, string
BelgeNo)
{
    string Sql = "EXECUTE SP_SENETIBANKAYAYER " + Id + ", " + BankaId +
", " + Tarih + ", " + BelgeNo + """;
    VeriTabani.Isle(Sql);
}

```



```

public Boolean SenetiBankadanIadeAl(string Id)
{
    try
    {
        string Sql = "EXECUTE SP_SENETIBANKADANIADEAL " + Id;
        VeriTabani.Isle(Sql);
        return true;
    }

    catch (Exception)
    {
        return false;
    }
}

public Boolean SenetiCaridenIadeAl(string Id)
{
    try
    {
        string Sql = "EXECUTE SP_SENETICARIDENIADEAL " + Id;
        VeriTabani.Isle(Sql);
        return true;
    }

    catch (Exception)
    {
        return false;
    }
}

public void SenetiCariyeVer(string Id, string CariKodu, string Tarih, string
BelgeNo)
{
    string Sql = "EXECUTE SP_SENETICARIYEVER " + Id + ", '" + CariKodu +
    "', '" + Tarih + "', '" + BelgeNo + "'";
    VeriTabani.Isle(Sql);
}

public Boolean SenetDurumu(string SenetId, string Durumu)
{
    try
    {
        string Sql = "EXECUTE SP_SENETINDURUMU " + SenetId + ", '" +
Durumu + "'";
        VeriTabani.Isle(Sql);
        return true;
    }

    catch (Exception)
    {

```

```

        return false;
    }
}

public string SenetTahsilatiYapilanKasa(string SenetId)
{
    try
    {
        string Sql = "SELECT KASAKODU FROM TBLKASAHAR WHERE
EVRAKTURU='SENET' AND EVRAKID =" + SenetId;
        DataRow Satir = VeriTabani.SatirCek(Sql);
        string KasaKodu = Satir["KASAKODU"].ToString();
        return KasaKodu;
    }

    catch (Exception)
    {
        return "";
    }
}

public string SenetTahsilatiYapilanBanka(string SenetId)
{
    try
    {
        string Sql = "SELECT BANKAID FROM TBLBANKAHAR WHERE
EVRAKTURU='SENET' AND EVRAKID =" + SenetId;
        DataRow Satir = VeriTabani.SatirCek(Sql);
        string BankaId = Satir["BANKAID"].ToString();
        return BankaId;
    }

    catch (Exception)
    {
        return "";
    }
}
}
}

```

#### **frmMusteriSeneti.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraEditors;

```

```

namespace Accounting_Software.CekSenetModulu.Senet
{
    public partial class frmMusteriSeneti : DevExpress.XtraEditors.XtraForm
    {
        Classlar.clsSenetler Senetler = new Classlar.clsSenetler();
        Accounting_Software.Classlar.Formatlar Formatlar = new
Accounting_Software.Classlar.Formatlar();
        Accounting_Software.Classlar.Mesajlar Mesajlar = new
Accounting_Software.Classlar.Mesajlar();
        Accounting_Software.CariModulu.Classlar.clsCariTanitimKarti Cari = new
CariModulu.Classlar.clsCariTanitimKarti();
        Accounting_Software.Classlar.Ekranlar Ekranlar = new
Accounting_Software.Classlar.Ekranlar();

        public MusteriSeneti GirilenSenet;
        public Boolean BordroIcinSec = false;
        string AsilMiCiroMu = "";
        string AcilanSenetId = "-1";
        string CariId = "-1";
        Boolean Edit = false;

        public frmMusteriSeneti()
        {
            InitializeComponent();
        }

        public MusteriSeneti Sec(string CariKodu, string CariAdi)
        {
            BordroIcinSec = true;
            txtCariKodu.Text = CariKodu;
            txtCariAdi.Text = CariAdi;
            txtCariKodu.Properties.ReadOnly = true;
            txtCariAdi.Properties.ReadOnly = true;
            raAsilSenet.Checked = true;
            Edit = false;
            this.ShowDialog();
            return GirilenSenet;
        }

        public MusteriSeneti Guncelle(MusteriSeneti Senet, string CariKodu, string
CariAdi, Boolean BordroIcinMi, Boolean DialogAc = true)
        {
            Edit = true;
            BordroIcinSec = BordroIcinMi;
            GirilenSenet = Senet;
            BilgileriTextlereAktar(CariKodu, CariAdi);

            if (DialogAc == true)
            {

```



```

        this.ShowDialog();
    }

    return GirilenSenet;
}

void BilgileriTextlereAktar(string CariKodu, string CariAdi)
{
    txtAciklama.Text = GirilenSenet.Aciklama;
    txtAsilBorclu.Text = GirilenSenet.AsilBorclu;
    txtBanka.Text = GirilenSenet.Banka;
    txtBelgeNo.Text = GirilenSenet.BelgeNo;
    txtCariAdi.Text = CariAdi;
    txtCariKodu.Text = CariKodu;
    txtSenetNo.Text = GirilenSenet.SenetNo;
    txtHesapNo.Text = GirilenSenet.HesapNo;
    txtSube.Text = GirilenSenet.Sube;
    txtTutar.Text = GirilenSenet.Tutar.ToString();
    txtVadeTarihi.Text = GirilenSenet.VadeTarihi;
    txtTarih.Text = GirilenSenet.Tarih;
}

private void btnKaydet_Click(object sender, EventArgs e)
{
    if (BordroIcinSec == true)
    {
        BordrolleKaydet();
    }

    else
    {
        if (Mesajlar.Sor("Seneti kaydetmek istediğinizden emin misiniz?") == true)
        {
            if (Edit == true)
            {
                DirektGuncelle();
            }

            else
            {
                DirektKaydet();
            }
        }
    }
}

void DirektKaydet()
{
    if (txtCariKodu.Text == "")
    {

```

```

        Mesajlar.Uyari("Cari seçmeden kayıt işlemi yapamazsınız!!!");
        CariSec();
    }

    else
    {
        if (Senetler.SenetGirisi(txtBelgeNo.Text, txtSenetNo.Text, "M",
txtCariKodu.Text, "", "", AsilMiCiroMu, txtAsilBorclu.Text,
Formatlar.IngilizceTarihFormati(txtTarih.Text),
Formatlar.IngilizceTarihFormati(txtVadeTarihi.Text), txtBanka.Text, txtSube.Text,
txtHesapNo.Text, txtTutar.Text, txtAciklama.Text, "B", "", "", frmAnaForm.UserId) ==
true)
        {
            Mesajlar.Bilgi("Senet başarılı bir şekilde kaydedilmiştir.");
        }

        else
        {
            Mesajlar.Hata();
        }
    }
}

void DirekGuncelle()
{
    if (txtCariKodu.Text == "")
    {
        Mesajlar.Uyari("Cari seçmeden kayıt işlemi yapamazsınız!!!");
        CariSec();
    }

    else
    {
        if (Senetler.SenetGirisiGuncelle(AcilanSenetId, txtBelgeNo.Text,
txtSenetNo.Text, "M", txtCariKodu.Text, "", "", AsilMiCiroMu, txtAsilBorclu.Text,
Formatlar.IngilizceTarihFormati(txtTarih.Text),
Formatlar.IngilizceTarihFormati(txtVadeTarihi.Text), txtBanka.Text, txtSube.Text,
txtHesapNo.Text, txtTutar.Text, txtAciklama.Text, "B", "", "", frmAnaForm.UserId) ==
true)
        {
            Mesajlar.Bilgi("Senet başarılı bir şekilde güncellenmiştir.");
        }

        else
        {
            Mesajlar.Hata();
        }
    }
}

```

```

void BordrolleKaydet()
{
    string AsilCiro = "A";
    GirilenSenet.Aciklama = txtAciklama.Text;
    GirilenSenet.AsilBorclu = txtAsilBorclu.Text;

    if (raAsilSenet.Checked == true)
    {
        AsilCiro = "C";
    }

    GirilenSenet.AsilCiro = AsilCiro;
    GirilenSenet.Banka = txtBanka.Text;
    GirilenSenet.BelgeNo = txtBelgeNo.Text;
    GirilenSenet.SenetNo = txtSenetNo.Text;
    GirilenSenet.HesapNo = txtHesapNo.Text;
    GirilenSenet.Sube = txtSube.Text;

    try
    {
        GirilenSenet.Tutar = Convert.ToDecimal(txtTutar.Text);
    }

    catch (Exception)
    {
        txtTutar.Text = "0.00";
        GirilenSenet.Tutar = 0;
    }

    GirilenSenet.VadeTarihi = txtVadeTarihi.Text;
    GirilenSenet.Tarih = txtTarih.Text;
    GirilenSenet.EklemeBasariliMi = true;

    this.Close();
}

```

```

void CariSec()
{
    CariId = Ekranlar.CariListesiAc();
    DataRow Satir = Cari.Ac(CariId.ToString());
    txtCariKodu.Text = Satir["CARIKODU"].ToString();
    txtCariAdi.Text = Satir["CARIISIM"].ToString();

    txtAsilBorclu.Properties.ReadOnly = true;
    txtAsilBorclu.Text = txtCariAdi.Text;
    AsilMiCiroMu = "A";
}

```

```

private void txtCariKodu_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)

```



```

    }

    catch (Exception)
    {
        return "";
    }
}

public string CekTahsilatiYapilanBanka(string CekId)
{
    try
    {
        string Sql = "SELECT BANKAID FROM TBLBANKAHAR WHERE
EVRAKTURU ='CEK' AND EVRAKID =" + CekId;
        DataRow Satir = VeriTabani.SatirCek(Sql);
        string BankaId = Satir["BANKAID"].ToString();
        return BankaId;
    }

    catch (Exception)
    {
        return "";
    }
}
}
}

```

#### **frmMusteriCeki.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraEditors;

namespace Accounting_Software.CekSenetModulu.Cek
{
    public partial class frmMusteriCeki : DevExpress.XtraEditors.XtraForm
    {
        Classlar.clsCekler Cekler = new Classlar.clsCekler();
        Accounting_Software.Classlar.Formatlar Formatlar = new
Accounting_Software.Classlar.Formatlar();
        Accounting_Software.Classlar.Mesajlar Mesajlar = new
Accounting_Software.Classlar.Mesajlar();
        Accounting_Software.CariModulu.Classlar.clsCariTanitimKarti Cari = new
CariModulu.Classlar.clsCariTanitimKarti();
    }
}

```

```
Accounting_Software.Classlar.Ekranlar Ekranlar = new  
Accounting_Software.Classlar.Ekranlar();
```

```
public MusteriCeki GirilenCek;  
public Boolean BordroIcinSec = false;  
string AsilMiCiroMu = "";  
string AcilanCekId = "-1";  
string CariId = "-1";  
Boolean Edit = false;
```

```
public frmMusteriCeki()  
{  
    InitializeComponent();  
}
```

```
public MusteriCeki Sec(string CariKodu, string CariAdi)  
{  
    BordroIcinSec = true;  
    txtCariKodu.Text = CariKodu;  
    txtCariAdi.Text = CariAdi;  
    txtCariKodu.Properties.ReadOnly = true;  
    txtCariAdi.Properties.ReadOnly = true;  
    raAsilCek.Checked = true;  
    Edit = false;  
    this.ShowDialog();  
    return GirilenCek;  
}
```

```
public MusteriCeki Guncelle(MusteriCeki Cek, string CariKodu, string CariAdi,  
Boolean BordroIcinMi, Boolean DialogAc = true)  
{  
    Edit = true;  
    BordroIcinSec = BordroIcinMi;  
    GirilenCek = Cek;  
    BilgileriTextlereAktar(CariKodu, CariAdi);  
  
    if (DialogAc == true)  
    {  
        this.ShowDialog();  
    }  
  
    return GirilenCek;  
}
```

```
void BilgileriTextlereAktar(string CariKodu, string CariAdi)  
{  
    txtAciklama.Text = GirilenCek.Aciklama;  
    txtAsilBorclu.Text = GirilenCek.AsilBorclu;  
    txtBanka.Text = GirilenCek.Banka;  
    txtBelgeNo.Text = GirilenCek.BelgeNo;
```

```

txtCariAdi.Text = CariAdi;
txtCariKodu.Text = CariKodu;
txtCekNo.Text = GirilenCek.CekNo;
txtHesapNo.Text = GirilenCek.HesapNo;
txtSube.Text = GirilenCek.Sube;
txtTutar.Text = GirilenCek.Tutar.ToString();
txtVadeTarihi.Text = GirilenCek.VadeTarihi;
txtTarih.Text = GirilenCek.Tarih;
}

private void frmMusteriCeki_Load(object sender, EventArgs e)
{
}

private void btnKaydet_Click(object sender, EventArgs e)
{
    if (BordroIcinSec == true)
    {
        BordrolleKaydet();
    }

    else
    {
        if (Mesajlar.Sor("Çeki kaydetmek istediğinizden emin misiniz?") == true)
        {
            if (Edit == true)
            {
                DirektGuncelle();
            }

            else
            {
                DirektKaydet();
            }
        }
    }
}

private void raAsilCek_CheckedChanged(object sender, EventArgs e)
{
    if (raAsilCek.Checked == true)
    {
        txtAsilBorclu.Properties.ReadOnly = true;
        txtAsilBorclu.Text = txtCariAdi.Text;
        AsilMiCiroMu = "A";
    }
}

private void raCiroCek_CheckedChanged(object sender, EventArgs e)

```



```

{
    if (raCiroCek.Checked == true)
    {
        txtAsilBorclu.Properties.ReadOnly = false;
        AsilMiCiroMu = "C";
    }
}

void DirekKaydet()
{
    if (txtCariKodu.Text == "")
    {
        Mesajlar.Uyari("Cari seçmeden kayıt işlemi yapamazsınız!!!");
        CariSec();
    }

    else
    {
        if (Cekler.CekGirisi(txtBelgeNo.Text, txtCekNo.Text, "M",
txtCariKodu.Text, "", "", AsilMiCiroMu, txtAsilBorclu.Text,
Formatlar.IngilizceTarihFormati(txtTarih.Text),
Formatlar.IngilizceTarihFormati(txtVadeTarihi.Text), txtBanka.Text, txtSube.Text,
txtHesapNo.Text, txtTutar.Text, txtAciklama.Text, "B", "", "", firmAnaForm.UserId) ==
true)
        {
            Mesajlar.Bilgi("Çek başarılı bir şekilde kaydedilmiştir.");
        }

        else
        {
            Mesajlar.Hata();
        }
    }
}

void DirekGuncelle()
{
    if (txtCariKodu.Text == "")
    {
        Mesajlar.Uyari("Cari seçmeden kayıt işlemi yapamazsınız!!!");
        CariSec();
    }

    else
    {
        if (Cekler.CekGirisiGuncelle(AcilanCekId, txtBelgeNo.Text, txtCekNo.Text,
"M", txtCariKodu.Text, "", "", AsilMiCiroMu, txtAsilBorclu.Text,
Formatlar.IngilizceTarihFormati(txtTarih.Text),
Formatlar.IngilizceTarihFormati(txtVadeTarihi.Text), txtBanka.Text, txtSube.Text,

```

```
txtHesapNo.Text, txtTutar.Text, txtAciklama.Text, "B", "", "", frmAnaForm.UserId) ==
true)
```

```
    {
        Mesajlar.Bilgi("Çek başarılı bir şekilde güncellenmiştir.");
    }

    else
    {
        Mesajlar.Hata();
    }
}
}
```

```
void BordrolleKaydet()
```

```
{
    string AsilCiro = "A";
    GirilenCek.Aciklama = txtAciklama.Text;
    GirilenCek.AsilBorclu = txtAsilBorclu.Text;
```

```
    if (raAsilCek.Checked == true)
    {
        AsilCiro = "C";
    }
}
```

```
GirilenCek.AsilCiro = AsilCiro;
GirilenCek.Banka = txtBanka.Text;
GirilenCek.BelgeNo = txtBelgeNo.Text;
GirilenCek.CekNo = txtCekNo.Text;
GirilenCek.HesapNo = txtHesapNo.Text;
GirilenCek.Sube = txtSube.Text;
```

```
try
{
    GirilenCek.Tutar = Convert.ToDecimal(txtTutar.Text);
}
}
```

```
catch (Exception)
{
    txtTutar.Text = "0.00";
    GirilenCek.Tutar = 0;
}
}
```

```
GirilenCek.VadeTarihi = txtVadeTarihi.Text;
GirilenCek.Tarih = txtTarih.Text;
GirilenCek.EklemeBasariliMi = true;
```

```
this.Close();
}
```

```
void CariSec()
```

```

{
    CariId = Ekranlar.CariListesiAc();
    DataRow Satir = Cari.Ac(CariId.ToString());
    txtCariKodu.Text = Satir["CARIKODU"].ToString();
    txtCariAdi.Text = Satir["CARIISIM"].ToString();
}

private void txtCariKodu_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
{
    if (BordroIcinSec == false)
    {
        CariSec();
    }
}

private void btnKapat_Click(object sender, EventArgs e)
{
    if (txtBelgeNo.Text != "" || txtCekNo.Text != "" || txtCariKodu.Text != "")
    {
        if (Mesajlar.Sor("İşlemi iptal etmek istediğinizden emin misiniz?") == true)
        {
            this.Close();
        }
    }
    else
    {
        this.Close();
    }
}

private void btnSil_Click(object sender, EventArgs e)
{
    if (Mesajlar.Sor("Çeki silmek istediğinizden emin misiniz?") == true)
    {
        if (BordroIcinSec == true)
        {
            GirilenCek.EklemeBasariliMi = false;
            this.Close();
        }
        else
        {
            Cekler.Sil(AcılanCekId);
            Mesajlar.Bilgi("Çekiniz başarılı bir şekilde silinmiştir.");
            this.Close();
        }
    }
}
}

```



```

public void Ac(string Id, Boolean Dialog = true)
{
    AcilanCekId = Id;

    DataRow Satir = Cekler.Ac(AcilanCekId);
    MusteriCeki AcilanCek = new MusteriCeki();
    string CariKodu, CariAdi;

    CariKodu = Satir["ALINANCARI"].ToString();
    CariAdi = Satir["ALINANCARIISIM"].ToString();
    AcilanCek.Aciklama = Satir["ACIKLAMA"].ToString();
    AcilanCek.AsilBorclu = Satir["ASILBORCLU"].ToString();
    AcilanCek.AsilCiro = Satir["ASILCIRO"].ToString();
    AcilanCek.Banka = Satir["BANKA"].ToString();
    AcilanCek.BelgeNo = Satir["BELGENO"].ToString();
    AcilanCek.CekNo = Satir["CEKNO"].ToString();
    AcilanCek.EklemeBasariliMi = true;
    AcilanCek.HesapNo = Satir["HESAPNO"].ToString();
    AcilanCek.Sube = Satir["SUBE"].ToString();
    AcilanCek.Tarih = Satir["TARIH"].ToString();
    AcilanCek.Tutar = Convert.ToDecimal(Satir["TUTAR"].ToString());
    AcilanCek.VadeTarihi = Satir["VADETARIHI"].ToString();

    Guncelle(AcilanCek, CariKodu, CariAdi, false, Dialog);
}

private void txtBelgeNo_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
{
    AcilanCekId = Ekranlar.MusteriCekListesi(true);
    Ac(AcilanCekId, false);
}
}

```

#### **clsSenetler.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;

namespace Accounting_Software.CekSenetModulu.Senet.Classlar
{
    class clsSenetler
    {

```

```
Accounting_Software.Classlar.Veritabani VeriTabani = new
Accounting_Software.Classlar.Veritabani();
```

```
public Boolean SenetGirisi(string BelgeNo, string SenetNo, string Tipi, string
AlinanCari, string VerilenCari, string VerilenBanka, string AsilCiro, string AsilBorclu,
string Tarih, string VadeTarihi, string Banka, string Sube, string HesapNo, string Tutar,
string Aciklama, string Durumu, string TahsilEdildiMi, string BordroNo, string
SaveUser)
{
    string Sql = "EXECUTE SP_SENET_EKLE " + BelgeNo + ", " + SenetNo +
    ", " + Tipi + ", " + AlinanCari + ", " + VerilenCari + ", " + VerilenBanka + ", " +
    AsilCiro + ", " + AsilBorclu + ", " + Tarih + ", " + VadeTarihi + ", " + Banka + ", " +
    Sube + ", " + HesapNo + ", " + Tutar.Replace(", ", ".") + ", " + Aciklama + ", " +
    Durumu + ", " + TahsilEdildiMi + ", " + BordroNo + ", " + SaveUser;
    return VeriTabani.Isle(Sql);
}
```

```
public Boolean SenetGirisiGuncelle(string Id, string BelgeNo, string SenetNo,
string Tipi, string AlinanCari, string VerilenCari, string VerilenBanka, string AsilCiro,
string AsilBorclu, string Tarih, string VadeTarihi, string Banka, string Sube, string
HesapNo, string Tutar, string Aciklama, string Durumu, string TahsilEdildiMi, string
BordroNo, string SaveUser)
{
    string Sql = "EXECUTE SP_SENET_GUNCELLE " + Id + ", " + BelgeNo + ",
    " + SenetNo + ", " + Tipi + ", " + AlinanCari + ", " + VerilenCari + ", " +
    VerilenBanka + ", " + AsilCiro + ", " + AsilBorclu + ", " + Tarih + ", " + VadeTarihi
    + ", " + Banka + ", " + Sube + ", " + HesapNo + ", " + Tutar.Replace(", ", ".") + ", " +
    Aciklama + ", " + Durumu + ", " + TahsilEdildiMi + ", " + BordroNo + ", " +
    SaveUser;
    return VeriTabani.Isle(Sql);
}
```

```
public Boolean Sil(string Id)
{
    try
    {
        string Sql = "EXECUTE SP_SENET_SIL " + Id;
        VeriTabani.Isle(Sql);
        return true;
    }

    catch (Exception)
    {
        return false;
    }
}
```

```
public DataTable MusteriSenetListesi(string BelgeNo, string SenetNo, string
CariKodu, string CariAdi, string Tarih, string VadeTarihi, string Aciklama, string
Durum, Boolean Bankadakiler, Boolean Portfoydakiler, Boolean Cirodakiler)
```

```

{
    string _Banka = "";
    string _Portfoy = "";
    string _Ciro = "";
    string YerSorgusu = "";

    if (Bankadakiler == true)
    {
        _Banka = "B";
    }

    if (Portfoydakiler == true)
    {
        _Portfoy = "P";
    }

    if (Cirodakiler == true)
    {
        _Ciro = "C";
    }

    YerSorgusu = "AND (YERI IN(" + _Banka + ", " + _Portfoy + ", " + _Ciro +
    "))";

    try
    {
        string Sql = "SELECT * FROM VWMUSTERISENETLERI WHERE
        BELGENO LIKE '%" + BelgeNo + "%' AND SENETNO LIKE '%" + SenetNo + "%'
        AND ALINANCARI LIKE '%" + CariKodu + "%' AND ALINANCARIISIM LIKE
        '%" + CariAdi + "%' AND TARİH LIKE '%" + Tarih + "%' AND VADETARİHI LIKE
        '%" + VadeTarihi + "%' AND ACIKLAMA LIKE '%" + Aciklama + "%' AND
        DURUMU LIKE '%" + Durum + "%' " + YerSorgusu;
        return VeriTabani.TabloCek(Sql);
    }

    catch (Exception)
    {
        throw;
    }
}

```

```

public DataTable KendiSenetListesi(string BelgeNo, string SenetNo, string
VerilenCari, string VerilenBanka, string Tarih, string VadeTarihi, string Aciklama,
string Durum, Boolean Bankadakiler, Boolean Portfoydakiler, Boolean Cirodakiler)
{
    string _Banka = "";
    string _Portfoy = "";
    string _Ciro = "";
    string YerSorgusu = "";

```



```

if (Bankadakiler == true)
{
    _Banka = "B";
}

if (Portfoydakiler == true)
{
    _Portfoy = "P";
}

if (Cirodakiler == true)
{
    _Ciro = "C";
}

YerSorgusu = "AND (YERI IN(" + _Banka + ", " + _Portfoy + ", " + _Ciro +
"))";

try
{
    string Sql = "SELECT * FROM VWKENDISENETLERI WHERE
BELGENO LIKE '%" + BelgeNo + "%' AND SENETNO LIKE '%" + SenetNo + "%'
AND VERILENCARIISIM LIKE '%" + VerilenCari + "%' AND
VERILENBANKAISIM LIKE '%" + VerilenBanka + "%' AND TARİH LIKE '%" +
Tarih + "%' AND VADETARİHI LIKE '%" + VadeTarihi + "%' AND ACIKLAMA
LIKE '%" + Aciklama + "%' AND DURUMU LIKE '%" + Durum + "%' " +
YerSorgusu;
    return VeriTabani.TabloCek(Sql);
}

catch (Exception)
{
    throw;
}

}

public DataRow Ac(string Id)
{
    string Sql = "SELECT TOP 1 * FROM VWSENETLER WHERE ID=" + Id;
    return VeriTabani.SatirCek(Sql);
}

public void SenetiBankayaVer(string Id, string BankaId, string Tarih, string
BelgeNo)
{
    string Sql = "EXECUTE SP_SENETIBANKAYAVER " + Id + ", " + BankaId +
", " + Tarih + ", " + BelgeNo + """;
    VeriTabani.Isle(Sql);
}

```

```

public Boolean SenetiBankadanIadeAl(string Id)
{
    try
    {
        string Sql = "EXECUTE SP_SENETIBANKADANIADEAL " + Id;
        VeriTani.Isle(Sql);
        return true;
    }

    catch (Exception)
    {
        return false;
    }
}

public Boolean SenetiCaridenIadeAl(string Id)
{
    try
    {
        string Sql = "EXECUTE SP_SENETICARIDENIADEAL " + Id;
        VeriTani.Isle(Sql);
        return true;
    }

    catch (Exception)
    {
        return false;
    }
}

public void SenetiCariyeVer(string Id, string CariKodu, string Tarih, string
BelgeNo)
{
    string Sql = "EXECUTE SP_SENETICARIYEVER " + Id + ", '" + CariKodu +
    "', '" + Tarih + "', '" + BelgeNo + "'";
    VeriTani.Isle(Sql);
}

public Boolean SenetDurumu(string SenetId, string Durumu)
{
    try
    {
        string Sql = "EXECUTE SP_SENETINDURUMU " + SenetId + ", '" +
Durumu + "'";
        VeriTani.Isle(Sql);
        return true;
    }

    catch (Exception)
    {

```

```

        return false;
    }
}

public string SenetTahsilatiYapilanKasa(string SenetId)
{
    try
    {
        string Sql = "SELECT KASAKODU FROM TBLKASAHAR WHERE
EVRAKTURU='SENET' AND EVRAKID =" + SenetId;
        DataRow Satir = VeriTabani.SatirCek(Sql);
        string KasaKodu = Satir["KASAKODU"].ToString();
        return KasaKodu;
    }

    catch (Exception)
    {
        return "";
    }
}

public string SenetTahsilatiYapilanBanka(string SenetId)
{
    try
    {
        string Sql = "SELECT BANKAID FROM TBLBANKAHAR WHERE
EVRAKTURU='SENET' AND EVRAKID =" + SenetId;
        DataRow Satir = VeriTabani.SatirCek(Sql);
        string BankaId = Satir["BANKAID"].ToString();
        return BankaId;
    }

    catch (Exception)
    {
        return "";
    }
}
}
}

```

#### **frmMusteriSeneti.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraEditors;

```



```

namespace Accounting_Software.CekSenetModulu.Senet
{
    public partial class frmMusteriSeneti : DevExpress.XtraEditors.XtraForm
    {
        Classlar.clsSenetler Senetler = new Classlar.clsSenetler();
        Accounting_Software.Classlar.Formatlar Formatlar = new
Accounting_Software.Classlar.Formatlar();
        Accounting_Software.Classlar.Mesajlar Mesajlar = new
Accounting_Software.Classlar.Mesajlar();
        Accounting_Software.CariModulu.Classlar.clsCariTanitimKarti Cari = new
CariModulu.Classlar.clsCariTanitimKarti();
        Accounting_Software.Classlar.Ekranlar Ekranlar = new
Accounting_Software.Classlar.Ekranlar();

        public MusteriSeneti GirilenSenet;
        public Boolean BordroIcinSec = false;
        string AsilMiCiroMu = "";
        string AcilanSenetId = "-1";
        string CariId = "-1";
        Boolean Edit = false;

        public frmMusteriSeneti()
        {
            InitializeComponent();
        }

        public MusteriSeneti Sec(string CariKodu, string CariAdi)
        {
            BordroIcinSec = true;
            txtCariKodu.Text = CariKodu;
            txtCariAdi.Text = CariAdi;
            txtCariKodu.Properties.ReadOnly = true;
            txtCariAdi.Properties.ReadOnly = true;
            raAsilSenet.Checked = true;
            Edit = false;
            this.ShowDialog();
            return GirilenSenet;
        }

        public MusteriSeneti Guncelle(MusteriSeneti Senet, string CariKodu, string
CariAdi, Boolean BordroIcinMi, Boolean DialogAc = true)
        {
            Edit = true;
            BordroIcinSec = BordroIcinMi;
            GirilenSenet = Senet;
            BilgileriTextlereAktar(CariKodu, CariAdi);

            if (DialogAc == true)
            {

```

```

        this.ShowDialog();
    }

    return GirilenSenet;
}

void BilgileriTextlereAktar(string CariKodu, string CariAdi)
{
    txtAciklama.Text = GirilenSenet.Aciklama;
    txtAsilBorclu.Text = GirilenSenet.AsilBorclu;
    txtBanka.Text = GirilenSenet.Banka;
    txtBelgeNo.Text = GirilenSenet.BelgeNo;
    txtCariAdi.Text = CariAdi;
    txtCariKodu.Text = CariKodu;
    txtSenetNo.Text = GirilenSenet.SenetNo;
    txtHesapNo.Text = GirilenSenet.HesapNo;
    txtSube.Text = GirilenSenet.Sube;
    txtTutar.Text = GirilenSenet.Tutar.ToString();
    txtVadeTarihi.Text = GirilenSenet.VadeTarihi;
    txtTarih.Text = GirilenSenet.Tarih;
}

private void btnKaydet_Click(object sender, EventArgs e)
{
    if (BordroIcinSec == true)
    {
        BordrolleKaydet();
    }

    else
    {
        if (Mesajlar.Sor("Seneti kaydetmek istediğinizden emin misiniz?") == true)
        {
            if (Edit == true)
            {
                DirektGuncelle();
            }

            else
            {
                DirektKaydet();
            }
        }
    }
}

void DirektKaydet()
{
    if (txtCariKodu.Text == "")
    {

```

```

        Mesajlar.Uyari("Cari seçmeden kayıt işlemi yapamazsınız!!!");
        CariSec();
    }

    else
    {
        if (Senetler.SenetGirisi(txtBelgeNo.Text, txtSenetNo.Text, "M",
txtCariKodu.Text, "", "", AsilMiCiroMu, txtAsilBorclu.Text,
Formatlar.IngilizceTarihFormati(txtTarih.Text),
Formatlar.IngilizceTarihFormati(txtVadeTarihi.Text), txtBanka.Text, txtSube.Text,
txtHesapNo.Text, txtTutar.Text, txtAciklama.Text, "B", "", "", frmAnaForm.UserId) ==
true)
        {
            Mesajlar.Bilgi("Senet başarılı bir şekilde kaydedilmiştir.");
        }

        else
        {
            Mesajlar.Hata();
        }
    }
}

void DirekGuncelle()
{
    if (txtCariKodu.Text == "")
    {
        Mesajlar.Uyari("Cari seçmeden kayıt işlemi yapamazsınız!!!");
        CariSec();
    }

    else
    {
        if (Senetler.SenetGirisiGuncelle(AcilanSenetId, txtBelgeNo.Text,
txtSenetNo.Text, "M", txtCariKodu.Text, "", "", AsilMiCiroMu, txtAsilBorclu.Text,
Formatlar.IngilizceTarihFormati(txtTarih.Text),
Formatlar.IngilizceTarihFormati(txtVadeTarihi.Text), txtBanka.Text, txtSube.Text,
txtHesapNo.Text, txtTutar.Text, txtAciklama.Text, "B", "", "", frmAnaForm.UserId) ==
true)
        {
            Mesajlar.Bilgi("Senet başarılı bir şekilde güncellenmiştir.");
        }

        else
        {
            Mesajlar.Hata();
        }
    }
}

```



```

void BordrolleKaydet()
{
    string AsilCiro = "A";
    GirilenSenet.Aciklama = txtAciklama.Text;
    GirilenSenet.AsilBorclu = txtAsilBorclu.Text;

    if (raAsilSenet.Checked == true)
    {
        AsilCiro = "C";
    }

    GirilenSenet.AsilCiro = AsilCiro;
    GirilenSenet.Banka = txtBanka.Text;
    GirilenSenet.BelgeNo = txtBelgeNo.Text;
    GirilenSenet.SenetNo = txtSenetNo.Text;
    GirilenSenet.HesapNo = txtHesapNo.Text;
    GirilenSenet.Sube = txtSube.Text;

    try
    {
        GirilenSenet.Tutar = Convert.ToDecimal(txtTutar.Text);
    }

    catch (Exception)
    {
        txtTutar.Text = "0.00";
        GirilenSenet.Tutar = 0;
    }

    GirilenSenet.VadeTarihi = txtVadeTarihi.Text;
    GirilenSenet.Tarih = txtTarih.Text;
    GirilenSenet.EklemeBasariliMi = true;

    this.Close();
}

```

```

void CariSec()
{
    CariId = Ekranlar.CariListesiAc();
    DataRow Satir = Cari.Ac(CariId.ToString());
    txtCariKodu.Text = Satir["CARIKODU"].ToString();
    txtCariAdi.Text = Satir["CARIISIM"].ToString();

    txtAsilBorclu.Properties.ReadOnly = true;
    txtAsilBorclu.Text = txtCariAdi.Text;
    AsilMiCiroMu = "A";
}

```

```

private void txtCariKodu_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)

```

```

{
    if (BordroIcinSec == false)
    {
        CariSec();
    }
}

private void btnKapat_Click(object sender, EventArgs e)
{
    if (txtBelgeNo.Text != "" || txtSenetNo.Text != "" || txtCariKodu.Text != "")
    {
        if (Mesajlar.Sor("İşlemi iptal etmek istediğinizden emin misiniz?") == true)
        {
            this.Close();
        }
    }

    else
    {
        this.Close();
    }
}

private void btnSil_Click(object sender, EventArgs e)
{
    if (Mesajlar.Sor("Seneti silmek istediğinizden emin misiniz?") == true)
    {
        if (BordroIcinSec == true)
        {
            GirilenSenet.EklemeBasariliMi = false;
            this.Close();
        }

        else
        {
            Senetler.Sil(AcilanSenetId);
            Mesajlar.Bilgi("Senetiniz başarılı bir şekilde silinmiştir.");
            this.Close();
        }
    }
}

public void Ac(string Id, Boolean Dialog = true)
{
    AcilanSenetId = Id;

    DataRow Satir = Senetler.Ac(AcilanSenetId);
    MusteriSeneti AcilanSenet = new MusteriSeneti();
    string CariKodu, CariAdi;

```

```

CariKodu = Satir["ALINANCARI"].ToString();
CariAdi = Satir["ALINANCARIISIM"].ToString();
AcilanSenet.Aciklama = Satir["ACIKLAMA"].ToString();
AcilanSenet.AsilBorclu = Satir["ASILBORCLU"].ToString();
AcilanSenet.AsilCiro = Satir["ASILCIRO"].ToString();
AcilanSenet.Banka = Satir["BANKA"].ToString();
AcilanSenet.BelgeNo = Satir["BELGENO"].ToString();
AcilanSenet.SenetNo = Satir["SENETNO"].ToString();
AcilanSenet.EklemeBasariliMi = true;
AcilanSenet.HesapNo = Satir["HESAPNO"].ToString();
AcilanSenet.Sube = Satir["SUBE"].ToString();
AcilanSenet.Tarih = Satir["TARIH"].ToString();
AcilanSenet.Tutar = Convert.ToDecimal(Satir["TUTAR"].ToString());
AcilanSenet.VadeTarihi = Satir["VADETARIHI"].ToString();

```

```

Guncelle(AcilanSenet, CariKodu, CariAdi, false, Dialog);

```

```

}

```

```

private void txtBelgeNo_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
{
    AcilanSenetId = Ekranlar.MusteriSenetListesi(true);
    Ac(AcilanSenetId, false);
}

```

```

private void raAsilSenet_CheckedChanged_1(object sender, EventArgs e)
{
    if (raAsilSenet.Checked == true)
    {
        txtAsilBorclu.Properties.ReadOnly = true;
        txtAsilBorclu.Text = txtCariAdi.Text;
        AsilMiCiroMu = "A";
    }
}

```

```

private void raCiroSenet_CheckedChanged_1(object sender, EventArgs e)
{
    if (raCiroSenet.Checked == true)
    {
        txtAsilBorclu.Properties.ReadOnly = false;
        AsilMiCiroMu = "C";
    }
}

```

```

private void frmMusteriSeneti_Load_1(object sender, EventArgs e)
{
    txtTarih.Text = DateTime.Now.ToShortDateString();
}
}

```



```
}
```

### **Ekranlar.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Accounting_Software.Classlar
{
    class Ekranlar
    {
        // STOK MODULU
        public string StokKod1Ac(Boolean Sec)
        {
            Accounting_Software.StokModulu.frmStokKod1 Form = new
            StokModulu.frmStokKod1();

            if (Sec == true)
            {
                Form.SecimIcinAcildiMi = true;
                Form.ShowDialog();
                return frmAnaForm.AraDegiskenString;
            }

            else
            {
                Form.SecimIcinAcildiMi = false;
                Form.ShowDialog();
            }

            return "";
        }

        public string StokKod2Ac(Boolean Sec)
        {
            Accounting_Software.StokModulu.frmStokKod2 Form = new
            StokModulu.frmStokKod2();

            if (Sec == true)
            {
                Form.SecimIcinAcildiMi = true;
                Form.ShowDialog();
                return frmAnaForm.AraDegiskenString;
            }

            else
            {
                Form.SecimIcinAcildiMi = false;
                Form.ShowDialog();
            }
        }
    }
}
```

```

    }

    return "";
}

public string StokKod3Ac(Boolean Sec)
{
    Accounting_Software.StokModulu.frmStokKod3 Form = new
StokModulu.frmStokKod3();

    if (Sec == true)
    {
        Form.SecimIcinAcildiMi = true;
        Form.ShowDialog();
        return frmAnaForm.AraDegiskenString;
    }

    else
    {
        Form.SecimIcinAcildiMi = false;
        Form.ShowDialog();
    }

    return "";
}

public string StokKod4Ac(Boolean Sec)
{
    Accounting_Software.StokModulu.frmStokKod4 Form = new
StokModulu.frmStokKod4();

    if (Sec == true)
    {
        Form.SecimIcinAcildiMi = true;
        Form.ShowDialog();
        return frmAnaForm.AraDegiskenString;
    }

    else
    {
        Form.SecimIcinAcildiMi = false;
        Form.ShowDialog();
    }

    return "";
}

public string StokKod5Ac(Boolean Sec)
{

```

```
Accounting_Software.StokModulu.frmStokKod5 Form = new  
StokModulu.frmStokKod5();
```

```
if (Sec == true)  
{  
    Form.SecimIcinAcildiMi = true;  
    Form.ShowDialog();  
    return frmAnaForm.AraDegiskenString;  
}
```

```
else  
{  
    Form.SecimIcinAcildiMi = false;  
    Form.ShowDialog();  
}
```

```
return "";  
}
```

```
public void StokTanitimKartiAc(bool Edit = false, string UrunId = "-1")  
{  
    Accounting_Software.StokModulu.frmStokTanitimKarti Form = new  
StokModulu.frmStokTanitimKarti();
```

```
if (Edit == true)  
{  
    Form.Ac(UrunId);  
}
```

```
Form.ShowDialog();  
}
```

```
public string StokListesiAc(Boolean Sec = false)  
{  
    Accounting_Software.StokModulu.frmStokListesi Form = new  
StokModulu.frmStokListesi();
```

```
if (Sec == true)  
{  
    Form.SecimIcinAc = true;  
    Form.ShowDialog();  
    return frmAnaForm.AraDegiskenString;  
}
```

```
else  
{  
    Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;  
    Form.SecimIcinAc = false;  
    Form.Show();  
    return "";
```



```

    }
}

public void StokHareketKayitlari(Boolean StokGonder = false, string UrunId = "-1")
{
    Accounting_Software.StokModulu.frmStokHareketleri Form = new
    StokModulu.frmStokHareketleri();

    if (StokGonder == true)
    {
        Form.Ac(UrunId);
    }

    Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;
    Form.Show();
}

public void StokAcilisKarti(string Id = "-1", Boolean Ac = false)
{
    Accounting_Software.StokModulu.frmStokAcilisKarti Form = new
    StokModulu.frmStokAcilisKarti();

    if (Ac == true)
    {
        Form.Ac(Id);
    }

    Form.ShowDialog();
}

public void TopluStokAcma()
{
    Accounting_Software.StokModulu.frmTopluStokAcma Form = new
    StokModulu.frmTopluStokAcma();
    Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;
    Form.Show();
}

public void TopluStokGuncelleme()
{
    Accounting_Software.StokModulu.frmTopluStokGuncelleme Form = new
    StokModulu.frmTopluStokGuncelleme();
    Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;
    Form.Show();
}

// CARI MODULU

public void CariTanitimKartiAc()

```

```

{
    Accounting_Software.CariModulu.frmCariTanitimKarti Form = new
CariModulu.frmCariTanitimKarti();
    Form.ShowDialog();
}

public string CariListesiAc(Boolean Dialog = true)
{
    Accounting_Software.CariModulu.frmCariListesi Form = new
CariModulu.frmCariListesi();

    if (Dialog == true)
    {
        Form.ShowDialog();
    }

    else
    {
        Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;
        Form.Show();
    }

    return Accounting_Software.frmAnaForm.AraDegiskenString;
}

public void CariHareketKayitlari(Boolean CariKoduGonder = false, string Id = "-
1")
{
    Accounting_Software.CariModulu.frmCariHareketler Form = new
CariModulu.frmCariHareketler();

    if (CariKoduGonder == true)
    {
        Form.CariSec(Id);
    }

    Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;
    Form.Show();
}

public void CariAcilisKarti(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.CariModulu.frmCariAcilisKarti Form = new
CariModulu.frmCariAcilisKarti();

    if (Ac == true)
    {
        Form.Ac(Id);
    }
}

```

```

    Form.ShowDialog();
}

// FATURA MODULU

public void SatisFaturasiAc()
{
    Accounting_Software.FaturaModulu.Satis.frmSatisFaturasi Form = new
FaturaModulu.Satis.frmSatisFaturasi();
    Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;
    Form.Show();
}

public string SatisFaturasiListesi(Boolean Dialog = true)
{
    Accounting_Software.FaturaModulu.Satis.frmSatisFaturasiListesi Form = new
FaturaModulu.Satis.frmSatisFaturasiListesi();

    if (Dialog == true)
    {
        Form.ShowDialog();
    }

    else
    {
        Form.MdiParent = frmAnaForm.ActiveForm;
        Form.SecimIcinAc = false;
        Form.Show();
    }

    return frmAnaForm.AraDegiskenString;
}

public void SatisIadeFaturasiAc()
{
    Accounting_Software.FaturaModulu.SatisIade.frmSatisIadeFaturasi Form = new
FaturaModulu.SatisIade.frmSatisIadeFaturasi();
    Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;
    Form.Show();
}

public string SatisIadeFaturasiListesi(Boolean Dialog = true)
{
    Accounting_Software.FaturaModulu.SatisIade.frmSatisIadeFaturasiListesi Form
= new FaturaModulu.SatisIade.frmSatisIadeFaturasiListesi();

    if (Dialog == true)
    {
        Form.ShowDialog();
    }
}

```



```

else
{
    Form.MdiParent = frmAnaForm.ActiveForm;
    Form.SecimIcinAc = false;
    Form.Show();
}

return frmAnaForm.AraDegiskenString;
}

public void AlisFaturasiAc()
{
    Accounting_Software.FaturaModulu.Alis.frmAlisFaturasi Form = new
FaturaModulu.Alis.frmAlisFaturasi();
    Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;
    Form.Show();
}

public string AlisFaturasiListesi(Boolean Dialog = true)
{
    Accounting_Software.FaturaModulu.Alis.frmAlisFaturasiListesi Form = new
FaturaModulu.Alis.frmAlisFaturasiListesi();

    if (Dialog == true)
    {
        Form.ShowDialog();
    }

    else
    {
        Form.MdiParent = frmAnaForm.ActiveForm;
        Form.SecimIcinAc = false;
        Form.Show();
    }

    return frmAnaForm.AraDegiskenString;
}

public void AlisIadeFaturasiAc()
{
    Accounting_Software.FaturaModulu.AlisIade.frmAlisIadeFaturasi Form = new
FaturaModulu.AlisIade.frmAlisIadeFaturasi();
    Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;
    Form.Show();
}

public string AlisIadeFaturasiListesi(Boolean Dialog = true)
{

```

```
Accounting_Software.FaturaModulu.AlisIade.frmAlisIadeFaturasiListesi Form  
= new FaturaModulu.AlisIade.frmAlisIadeFaturasiListesi();
```

```
if (Dialog == true)
```

```
{  
    Form.ShowDialog();  
}
```

```
else
```

```
{  
    Form.MdiParent = frmAnaForm.ActiveForm;  
    Form.SecimIcinAc = false;  
    Form.Show();  
}
```

```
return frmAnaForm.AraDegiskenString;
```

```
}
```

```
// KASA MODULU
```

```
public void KasaTanitimKarti(Boolean Ac = false, string Id = "-1")
```

```
{
```

```
    Accounting_Software.KasaModulu.frmKasaTanitimKarti Form = new  
KasaModulu.frmKasaTanitimKarti();
```

```
if (Ac == true)
```

```
{  
    Form.Ac(Id);  
}
```

```
Form.ShowDialog();
```

```
}
```

```
public string KasaListesi(Boolean Dialog = false)
```

```
{
```

```
    Accounting_Software.KasaModulu.frmKasaListesi Form = new  
KasaModulu.frmKasaListesi();
```

```
if (Dialog == true)
```

```
{  
    Form.SecimIcinAc = true;  
    Form.ShowDialog();  
    return frmAnaForm.AraDegiskenString;  
}
```

```
else
```

```
{  
    Form.SecimIcinAc = false;  
    Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;  
    Form.Show();  
}
```

```

    }

    return "";
}

public void KasaHareketKayitlari(Boolean Ac = false, string KasaId = "-1")
{
    Accounting_Software.KasaModulu.frmKasaHareketKayitlari Form = new
KasaModulu.frmKasaHareketKayitlari();

    if (Ac == true)
    {
        Form.Ac(KasaId);
    }

    Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;
    Form.Show();
}

public void KasaAcilisKarti(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.KasaModulu.frmKasaAcilisKarti Form = new
KasaModulu.frmKasaAcilisKarti();

    if (Ac == true)
    {
        Form.Ac(Id);
    }

    Form.ShowDialog();
}

public void KasaTahsilatGirisi(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.KasaModulu.frmKasaTahsilat Form = new
KasaModulu.frmKasaTahsilat();

    if (Ac == true)
    {
        Form.Ac(Id);
    }

    Form.ShowDialog();
}

public void KasaOdemeGirisi(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.KasaModulu.frmKasaOdeme Form = new
KasaModulu.frmKasaOdeme();
}

```



```

    if (Ac == true)
    {
        Form.Ac(Id);
    }

    Form.ShowDialog();
}

// BANKA MODULU

public void BankaTanitimKarti(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.BankaModulu.frmBankaTanitimKarti Form = new
BankaModulu.frmBankaTanitimKarti();

    if (Ac == true)
    {
        Form.Ac(Id);
    }

    Form.ShowDialog();
}

public string BankaListesi(Boolean Dialog = false)
{
    Accounting_Software.BankaModulu.frmBankaListesi Form = new
BankaModulu.frmBankaListesi();

    if (Dialog == true)
    {
        Form.SecimIcinAc = true;
        Form.ShowDialog();
        return frmAnaForm.AraDegiskenString;
    }

    else
    {
        Form.SecimIcinAc = false;
        Form.MdiParent = Accounting_Software.frmAnaForm.ActiveForm;
        Form.Show();
    }

    return "";
}

public void BankaAcilisKarti(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.BankaModulu.frmBankaAcilisKarti Form = new
BankaModulu.frmBankaAcilisKarti();

```

```

        if (Ac == true)
        {
            Form.Ac(Id);
        }

        Form.ShowDialog();
    }

    public void BankaHareketKayitlari()
    {
        Accounting_Software.BankaModulu.frmBankaHareketleri Form = new
BankaModulu.frmBankaHareketleri();
        Form.MdiParent = frmAnaForm.ActiveForm;
        Form.Show();
    }

    public void BankaIslemi(Boolean Ac = false, string Id = "-1")
    {
        Accounting_Software.BankaModulu.frmBankaIslemi Form = new
BankaModulu.frmBankaIslemi();

        if (Ac == true)
        {
            Form.Ac(Id);
        }

        Form.ShowDialog();
    }

    public void GelenHavale(Boolean Ac = false, string Id = "-1")
    {
        Accounting_Software.BankaModulu.frmGelenHavale Form = new
BankaModulu.frmGelenHavale();

        if (Ac == true)
        {
            Form.Ac(Id);
        }

        Form.ShowDialog();
    }

    public void GonderilenHavale(Boolean Ac = false, string Id = "-1")
    {
        Accounting_Software.BankaModulu.frmGonderilenHavale Form = new
BankaModulu.frmGonderilenHavale();

        if (Ac == true)
        {
            Form.Ac(Id);
        }
    }

```

```

    }

    Form.ShowDialog();
}

// ÇEK MODULU

public void MusteriCekGirisi(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.CekSenetModulu.Cek.frmMusteriCeki Form = new
CekSenetModulu.Cek.frmMusteriCeki();

    if (Ac == true)
    {
        Form.Ac(Id);
    }

    else
    {
        Form.ShowDialog();
    }
}

public void KendiCekGirisi(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.CekSenetModulu.Cek.frmKendiCekimiz Form = new
CekSenetModulu.Cek.frmKendiCekimiz();

    if (Ac == true)
    {
        Form.Ac(Id);
    }

    else
    {
        Form.ShowDialog();
    }
}

public void MusteriCekBordrosuGirisi(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.CekSenetModulu.Cek.frmMusteriCekBordrosu Form =
new CekSenetModulu.Cek.frmMusteriCekBordrosu();

    Form.MdiParent = frmAnaForm.ActiveForm;

    if (Ac == true)
    {
        Form.BordroAc(Id);
    }
}

```



```

    Form.Show();
}

public MusteriCeki BordroIcinCekSec(string CariKodu, string CariAdi)
{
    Accounting_Software.CekSenetModulu.Cek.frmMusteriCeki Form = new
    CekSenetModulu.Cek.frmMusteriCeki();

    return Form.Sec(CariKodu, CariAdi);
}

public MusteriCeki BordroIcinCekGuncelle(MusteriCeki Cek, string CariKodu,
string CariAdi)
{
    Accounting_Software.CekSenetModulu.Cek.frmMusteriCeki Form = new
    CekSenetModulu.Cek.frmMusteriCeki();

    return Form.Guncelle(Cek, CariKodu, CariAdi, true);
}

public string CekBordroListesi(Boolean Dialog = false)
{
    Accounting_Software.CekSenetModulu.Cek.frmCekBordroListesi Form = new
    CekSenetModulu.Cek.frmCekBordroListesi();

    if (Dialog == true)
    {
        Form.SecimIcinAc = true;
        Form.ShowDialog();
        return frmAnaForm.AraDegiskenString;
    }

    else
    {
        Form.MdiParent = frmAnaForm.ActiveForm;
        Form.Show();
    }

    return "";
}

public string MusteriCekListesi(Boolean Dialog = false, Boolean Bankadakiler =
false, Boolean Portfoydekiler = false, Boolean CiroEdilenler = false)
{
    Accounting_Software.CekSenetModulu.Cek.frmMusteriCekListesi Form = new
    CekSenetModulu.Cek.frmMusteriCekListesi();

    if (Dialog == true)
    {

```

```

Form.SecimIcinAc = true;
Form.Bankadakiler = Bankadakiler;
Form.Portfoydakiler = Portfoydakiler;
Form.Cirodakiler = CiroEdilenler;
Form.ShowDialog();
return frmAnaForm.AraDegiskenString;
}

```

```

else
{
    Form.Bankadakiler = true;
    Form.Portfoydakiler = true;
    Form.Cirodakiler = true;
    Form.MdiParent = frmAnaForm.ActiveForm;
    Form.Show();
}

```

```

return "";
}

```

```

public string KendiCekListesi(Boolean Dialog = false, Boolean Bankadakiler =
false, Boolean Portfoydakiler = false, Boolean CiroEdilenler = false)

```

```

{
    Accounting_Software.CekSenetModulu.Cek.frmKendiCekListesimiz Form =
new CekSenetModulu.Cek.frmKendiCekListesimiz();

```

```

if (Dialog == true)
{
    Form.SecimIcinAc = true;
    Form.Bankadakiler = Bankadakiler;
    Form.Portfoydakiler = Portfoydakiler;
    Form.Cirodakiler = CiroEdilenler;
    Form.ShowDialog();
    return frmAnaForm.AraDegiskenString;
}

```

```

else
{
    Form.Bankadakiler = true;
    Form.Portfoydakiler = true;
    Form.Cirodakiler = true;
    Form.MdiParent = frmAnaForm.ActiveForm;
    Form.Show();
}

```

```

return "";
}

```

```

public void BankayaCekCikisi()

```

```

{

```

```

        Accounting_Software.CekSenetModulu.Cek.frmBankayaCekCikisi Form = new
CekSenetModulu.Cek.frmBankayaCekCikisi();
        Form.ShowDialog();
    }

    public void CariyeCekCikisi()
    {
        Accounting_Software.CekSenetModulu.Cek.frmCariyeCekCikisi Form = new
CekSenetModulu.Cek.frmCariyeCekCikisi();
        Form.ShowDialog();
    }

    public void BankayaCekCikisiTek(string CekId)
    {
        Accounting_Software.CekSenetModulu.Cek.frmBankayaCekCikisiTek Form =
new CekSenetModulu.Cek.frmBankayaCekCikisiTek();
        Form.CekAc(CekId);
    }

    public void CariyeCekCikisiTek(string CekId)
    {
        Accounting_Software.CekSenetModulu.Cek.frmCariyeCekCikisiTek Form =
new CekSenetModulu.Cek.frmCariyeCekCikisiTek();
        Form.CekAc(CekId);
    }

    public void BankayaCekCikisiIade(string CekId)
    {
        Accounting_Software.CekSenetModulu.Cek.frmBankayaCekCikisiIade Form =
new CekSenetModulu.Cek.frmBankayaCekCikisiIade();
        Form.CekAc(CekId);
    }

    public void CariyeCekCikisiIade(string CekId)
    {
        Accounting_Software.CekSenetModulu.Cek.frmCariyeCekCikisiIade Form =
new CekSenetModulu.Cek.frmCariyeCekCikisiIade();
        Form.CekAc(CekId);
    }

    public void CekDurumu(string CekId, string Yeri)
    {
        if (Yeri == "P")
        {
            Accounting_Software.CekSenetModulu.Cek.frmCekDurumuPortfoy Form =
new CekSenetModulu.Cek.frmCekDurumuPortfoy();
            Form.CekAc(CekId);
        }

        else if (Yeri == "B")

```



```

    {
        Accounting_Software.CekSenetModulu.Cek.frmCekDurumuBanka Form =
new CekSenetModulu.Cek.frmCekDurumuBanka();
        Form.CekAc(CekId);
    }
}

//SENET

public void MusteriSenetGirisi(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.CekSenetModulu.Senet.frmMusteriSeneti Form = new
CekSenetModulu.Senet.frmMusteriSeneti();

    if (Ac == true)
    {
        Form.Ac(Id);
    }

    else
    {
        Form.ShowDialog();
    }
}

public void KendiSenetGirisi(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.CekSenetModulu.Senet.frmKendiSenetimiz Form = new
CekSenetModulu.Senet.frmKendiSenetimiz();

    if (Ac == true)
    {
        Form.Ac(Id);
    }

    else
    {
        Form.ShowDialog();
    }
}

public void MusteriSenetBordrosuGirisi(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.CekSenetModulu.Senet.frmMusteriSenetBordrosu Form
= new CekSenetModulu.Senet.frmMusteriSenetBordrosu();

    Form.MdiParent = frmAnaForm.ActiveForm;

    if (Ac == true)
    {

```

```

        Form.BordroAc(Id);
    }

    Form.Show();
}

public MusteriSeneti BordroIcinSenetSec(string CariKodu, string CariAdi)
{
    Accounting_Software.CekSenetModulu.Senet.frmMusteriSeneti Form = new
    CekSenetModulu.Senet.frmMusteriSeneti();

    return Form.Sec(CariKodu, CariAdi);
}

public MusteriSeneti BordroIcinSenetGuncelle(MusteriSeneti Senet, string
CariKodu, string CariAdi)
{
    Accounting_Software.CekSenetModulu.Senet.frmMusteriSeneti Form = new
    CekSenetModulu.Senet.frmMusteriSeneti();

    return Form.Guncelle(Senet, CariKodu, CariAdi, true);
}

public string SenetBordroListesi(Boolean Dialog = false)
{
    Accounting_Software.CekSenetModulu.Senet.frmSenetBordroListesi Form =
    new CekSenetModulu.Senet.frmSenetBordroListesi();

    if (Dialog == true)
    {
        Form.SecimIcinAc = true;
        Form.ShowDialog();
        return frmAnaForm.AraDegiskenString;
    }

    else
    {
        Form.MdiParent = frmAnaForm.ActiveForm;
        Form.Show();
    }

    return "";
}

public string MusteriSenetListesi(Boolean Dialog = false, Boolean Bankadakiler =
false, Boolean Portfoydekiler = false, Boolean CiroEdilenler = false)
{
    Accounting_Software.CekSenetModulu.Senet.frmMusteriSenetListesi Form =
    new CekSenetModulu.Senet.frmMusteriSenetListesi();

```

```

if (Dialog == true)
{
    Form.SecimIcinAc = true;
    Form.Bankadakiler = Bankadakiler;
    Form.Portfoydakiler = Portfoydakiler;
    Form.Cirodakiler = CiroEdilenler;
    Form.ShowDialog();
    return frmAnaForm.AraDegiskenString;
}

else
{
    Form.Bankadakiler = true;
    Form.Portfoydakiler = true;
    Form.Cirodakiler = true;
    Form.MdiParent = frmAnaForm.ActiveForm;
    Form.Show();
}

return "";
}

public string KendiSenetListesi(Boolean Dialog = false, Boolean Bankadakiler =
false, Boolean Portfoydakiler = false, Boolean CiroEdilenler = false)
{
    Accounting_Software.CekSenetModulu.Senet.frmKendiSenetListesimiz Form =
new CekSenetModulu.Senet.frmKendiSenetListesimiz();

    if (Dialog == true)
    {
        Form.SecimIcinAc = true;
        Form.Bankadakiler = Bankadakiler;
        Form.Portfoydakiler = Portfoydakiler;
        Form.Cirodakiler = CiroEdilenler;
        Form.ShowDialog();
        return frmAnaForm.AraDegiskenString;
    }

    else
    {
        Form.Bankadakiler = true;
        Form.Portfoydakiler = true;
        Form.Cirodakiler = true;
        Form.MdiParent = frmAnaForm.ActiveForm;
        Form.Show();
    }

    return "";
}

```



```

public void BankayaSenetCikisi()
{
    Accounting_Software.CekSenetModulu.Senet.frmBankayaSenetCikisi Form =
new CekSenetModulu.Senet.frmBankayaSenetCikisi();
    Form.ShowDialog();
}

public void CariyeSenetCikisi()
{
    Accounting_Software.CekSenetModulu.Senet.frmCariyeSenetCikisi Form =
new CekSenetModulu.Senet.frmCariyeSenetCikisi();
    Form.ShowDialog();
}

public void BankayaSenetCikisiTek(string SenetId)
{
    Accounting_Software.CekSenetModulu.Senet.frmBankayaSenetCikisiTek Form
= new CekSenetModulu.Senet.frmBankayaSenetCikisiTek();
    Form.SenetAc(SenetId);
}

public void CariyeSenetCikisiTek(string SenetId)
{
    Accounting_Software.CekSenetModulu.Senet.frmCariyeSenetCikisiTek Form =
new CekSenetModulu.Senet.frmCariyeSenetCikisiTek();
    Form.SenetAc(SenetId);
}

public void BankayaSenetCikisiIade(string SenetId)
{
    Accounting_Software.CekSenetModulu.Senet.frmBankayaSenetCikisiIade
Form = new CekSenetModulu.Senet.frmBankayaSenetCikisiIade();
    Form.SenetAc(SenetId);
}

public void CariyeSenetCikisiIade(string SenetId)
{
    Accounting_Software.CekSenetModulu.Senet.frmCariyeSenetCikisiIade Form =
new CekSenetModulu.Senet.frmCariyeSenetCikisiIade();
    Form.SenetAc(SenetId);
}

public void SenetDurumu(string SenetId, string Yeri)
{
    if (Yeri == "P")
    {
        Accounting_Software.CekSenetModulu.Senet.frmSenetDurumuPortfoy Form
= new CekSenetModulu.Senet.frmSenetDurumuPortfoy();
        Form.SenetAc(SenetId);
    }
}

```

```

        else if (Yeri == "B")
        {
            Accounting_Software.CekSenetModulu.Senet.frmSenetDurumuBanka Form
= new CekSenetModulu.Senet.frmSenetDurumuBanka();
            Form.SenetAc(SenetId);
        }
    }

```

```

//RAPORLAR

```

```

//*****STOK RAPORLARI*****

```

```

public void Rapor_StokHareket()
{
    Accounting_Software.StokModulu.Raporlar.frmRapStokHareket Form = new
StokModulu.Raporlar.frmRapStokHareket();
    Form.MdiParent = frmAnaForm.ActiveForm;
    Form.Show();
}

```

```

public void Rapor_CariStokHareket()
{
    Accounting_Software.StokModulu.Raporlar.frmRapCariStokHareket Form =
new StokModulu.Raporlar.frmRapCariStokHareket();
    Form.MdiParent = frmAnaForm.ActiveForm;
    Form.Show();
}

```

```

public void Rapor_TarihAraligiStokHareket()
{
    Accounting_Software.StokModulu.Raporlar.frmRapTarihAraligiStokHareket
Form = new StokModulu.Raporlar.frmRapTarihAraligiStokHareket();
    Form.MdiParent = frmAnaForm.ActiveForm;
    Form.Show();
}

```

```

//*****CARI RAPORLARI*****

```

```

public void Rapor_CariHareket()
{
    Accounting_Software.CariModulu.Raporlar.frmRapCariHareket Form = new
CariModulu.Raporlar.frmRapCariHareket();
    Form.MdiParent = frmAnaForm.ActiveForm;
    Form.Show();
}

```

```

public void Rapor_OzelCariRaporu()
{
    Accounting_Software.CariModulu.Raporlar.frmRapCariOzelRaporu Form =
new CariModulu.Raporlar.frmRapCariOzelRaporu();
}

```

```

    Form.MdiParent = frmAnaForm.ActiveForm;
    Form.Show();
}

/** KULLANICI İŞLEMLERİ **/

public void KullaniciEkle()
{
    Accounting_Software.KullaniciModulu.frmKullanici Form = new
KullaniciModulu.frmKullanici();
    Form.Edit = false;
    Form.ShowDialog();
}

public void KullaniciGuncelle(string Id)
{
    Accounting_Software.KullaniciModulu.frmKullanici Form = new
KullaniciModulu.frmKullanici();
    Form.Edit = true;
    Form.Ac(Id);
    Form.ShowDialog();
}

public string KullaniciListesi(Boolean Dialog = false)
{
    Accounting_Software.KullaniciModulu.frmKullaniciListesi Form = new
KullaniciModulu.frmKullaniciListesi();

    if (Dialog == true)
    {
        Form.SecimIcinAc = true;
        Form.ShowDialog();
        return frmAnaForm.AraDegiskenString;
    }

    else
    {
        Form.MdiParent = frmAnaForm.ActiveForm;
        Form.Show();
    }

    return "";
}

public void KullaniciYetkileri(Boolean Ac = false, string Id = "-1")
{
    Accounting_Software.KullaniciModulu.frmYetkiler Form = new
KullaniciModulu.frmYetkiler();
    Form.MdiParent = frmAnaForm.ActiveForm;

```



```

        if (Ac == true)
        {
            Form.Ac(Id);
        }

        Form.Show();
    }

    public void ParolaDegistir()
    {
        frmParolaDegistir Form = new frmParolaDegistir();
        Form.ShowDialog();
    }

    public void ServerAyarlari()
    {
        frmServerAyarlari Form = new frmServerAyarlari();
        Form.ShowDialog();
    }
}

```

### ExcellIslemleri.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.SqlClient;
using System.Data.OleDb;

namespace Accounting_Software.Classlar
{
    class ExcellIslemleri
    {
        public DataTable Veriler(string Yol)
        {
            string BaglantiStringi = "Provider = Microsoft.Jet.OLEDB.4.0; Data Source = "
+ Yol + "; Extended Properties = 'Excel 8.0;';";
            OleDbConnection Baglanti = new OleDbConnection(BaglantiStringi);
            Baglanti.Open();

            OleDbDataAdapter Adapter = new OleDbDataAdapter("SELECT * FROM
[Sayfa1$A:Z]", Baglanti);
            DataTable Veriler = new DataTable();
            Adapter.Fill(Veriler);
            return Veriler;
        }
    }
}

```

}

## ExportIslemleri.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraBars;
using System.Diagnostics;
```

```
namespace Accounting_Software.Classlar
```

```
{
```

```
class ExportIslemleri
```

```
{
```

```
    Mesajlar Mesajlar = new Mesajlar();
```

```
    SaveFileDialog Save = new SaveFileDialog();
```

```
    public void Excel2003(string RaporAdi, DevExpress.XtraGrid.GridControl Liste)
```

```
    {
        Save.Filter = "Excel 2003 | *.xls";
        Save.FileName = RaporAdi + "_" + DateTime.Now.ToShortDateString();
```

```
        if (Save.ShowDialog() == DialogResult.OK)
```

```
        {
            Liste.ExportToXls(Save.FileName);
```

```
            if (Mesajlar.Sor("Dosyayı açmak ister misiniz?") == true)
```

```
            {
                Process.Start(Save.FileName);
```

```
            }
```

```
        }
```

```
    }
```

```
    public void Excel2007(string RaporAdi, DevExpress.XtraGrid.GridControl Liste)
```

```
    {
        Save.Filter = "Excel 2007 | *.xlsx";
        Save.FileName = RaporAdi + "_" + DateTime.Now.ToShortDateString();
```

```
        if (Save.ShowDialog() == DialogResult.OK)
```

```
        {
            Liste.ExportToXlsx(Save.FileName);
```

```
            if (Mesajlar.Sor("Dosyayı açmak ister misiniz?") == true)
```

```
            {
                Process.Start(Save.FileName);
```

```
            }
```

```
        }
```

```
    }
```





```

string Gun, Ay, Yil, Saat, Dakika, Saniye;
Gun = Tarih.Day.ToString();
Ay = Tarih.Month.ToString();
Yil = Tarih.Year.ToString();
Saat = Tarih.Hour.ToString();
Dakika = Tarih.Minute.ToString();
Saniye = Tarih.Second.ToString();

```

```

string YeniTarih = Ay + "." + Gun + "." + Yil + " " + Saat + ":" + Dakika + ":" +

```

```

Saniye;
return YeniTarih;
}
}
}

```

## GirisIslemi.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Windows.Forms;

```

```

namespace Accounting_Software.Classlar

```

```

{
    class GirisIslemi
    {
        Veritabanı VeriTabani = new Veritabanı();
        Mesajlar Mesajlar = new Mesajlar();

        public Boolean KullaniciGirisi(string KullaniciAdi, string Parola)
        {
            string Sql = "SELECT TOP 1 * FROM TBLKULLANICILAR WHERE KULLANICIADI = '" + KullaniciAdi + "' AND PAROLA = '" + Parola + "'";
            try
            {
                DataRow Satir = VeriTabani.SatirCek(Sql);
                frmAnaForm.UserId = Satir["ID"].ToString();
                frmAnaForm.UserName = KullaniciAdi;
                frmAnaForm.GirisDurumu = true;
                return true;
            }

            catch (Exception)
            {
                frmAnaForm.UserId = "-1";
                frmAnaForm.UserName = "";
                frmAnaForm.GirisDurumu = false;
            }
        }
    }
}

```

```

        return false;
    }

    return false;
}

public Boolean ParolaDegistir(string UserId, string EskiParola, string YeniParola)
{
    string Sql = "SELECT * FROM TBLKULLANICILAR WHERE ID=" +
    UserId;

    try
    {
        DataRow Satir = VeriTabani.SatirCek(Sql);

        string VtParola = Satir["PAROLA"].ToString();

        if (EskiParola == VtParola)
        {
            return VeriTabani.Isle("UPDATE TBLKULLANICILAR SET
PAROLA=" + YeniParola + " WHERE ID=" + UserId);
        }

        else
        {
            Mesajlar.Uyari("Eski Parolanızı yanlış girdiniz. Lütfen tekrar deneyiniz.");
            return false;
        }
    }

    catch (Exception)
    {
        Mesajlar.Hata();
        return false;
    }

    return false;
}
}
}

```

### **Mesajlar.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Accounting_Software.Classlar

```

```

{
    class Mesajlar
    {
        public Boolean Sor(string Mesaj)
        {
            DialogResult Sor = new DialogResult();
            Sor = MessageBox.Show(Mesaj, "Uyarı", MessageBoxButtons.YesNo,
            MessageBoxIcon.Question, MessageBoxDefaultButton.Button2);

            if (Sor == DialogResult.Yes)
            {
                return true;
            }

            return false;
        }

        public void Bilgi(string Mesaj)
        {
            MessageBox.Show(Mesaj, "Bilgi", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        }

        public void Hata()
        {
            MessageBox.Show("Beklenmedik bir hata oluştu. Lütfen tekrar deneyiniz.",
            "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }

        public void Uyari(string Mesaj)
        {
            MessageBox.Show(Mesaj, "Bilgi", MessageBoxButtons.OK,
            MessageBoxIcon.Warning);
        }
    }
}

```

### **Veritabani.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.SqlClient;

namespace Accounting_Software.Classlar
{
    class Veritabani
    {

```



```

Mesajlar Mesajlar = new Mesajlar();

public SqlConnection Baglan()
{
    string Provider = @"Data Source=Ender-Pc;Initial
Catalog=Accounting_Software;Integrated Security=True;User ID="";Password="";
    SqlConnection Connection = new SqlConnection(Provider);

    try
    {
        Connection.Open();
        frmAnaForm.ErisimBilgileri = false;
    }

    catch (Exception)
    {
        Mesajlar.Uyari("Server ile bağlantı kurulamadı. Lütfen erişim bilgilerinizi
kontrol ediniz.");
        frmAnaForm.ErisimBilgileri = true;
    }

    return Connection;
}

public DataTable TabloCek(string Sql)
{
    DataTable Tablo = new DataTable();
    SqlDataAdapter Adapter = new SqlDataAdapter(Sql, this.Baglan());
    Adapter.Fill(Tablo);
    return Tablo;
}

public DataSet DataSetCek(string Sql, string Tablo)
{
    DataSet DSet = new DataSet();
    SqlDataAdapter Adapter = new SqlDataAdapter(Sql, this.Baglan());
    Adapter.Fill(DSet, Tablo);
    return DSet;
}

public DataRow SatirCek(string Sql)
{
    try
    {
        DataRow Satir = TabloCek(Sql).Rows[0];
        return Satir;
    }

    catch (Exception)
    {

```

```

        throw;
    }
}

public Boolean Isle(string Sql)
{
    try
    {
        SqlCommand Islem = new SqlCommand(Sql, this.Baglan());
        Islem.ExecuteNonQuery();
        return true;
    }

    catch (Exception)
    {
        return false;
    }
}
}
}

```

#### **clsSatisFaturasi.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Windows.Forms;

namespace Accounting_Software.FaturaModulu.Classlar
{
    class clsSatisFaturasi
    {
        Accounting_Software.Classlar.Veritabani VeriTabani = new
        Accounting_Software.Classlar.Veritabani();
        Accounting_Software.Classlar.Formatlar Formatlar = new
        Accounting_Software.Classlar.Formatlar();
        Accounting_Software.StokModulu.Classlar.clsStokHareketleri StokHareketleri =
        new StokModulu.Classlar.clsStokHareketleri();

        public DataTable Listele(string CariKodu, string CariAdi, string FaturaNo)
        {
            string Sql = "SELECT * FROM VWFATURALAR WHERE CARIKODU
            LIKE '%" + CariKodu + "%' AND CARIISIM LIKE '%" + CariAdi + "%' AND
            FATURANO LIKE '%" + FaturaNo + "%' AND TIPI='S'";
            return VeriTabani.TabloCek(Sql);
        }
    }
}

```

```

public Boolean Ekle(string FaturaNo, string CariKodu, string Tarih, string
VadeTarihi, string GenelToplam, string Turu, string Aciklama, DataTable Detaylar,
string OdemeYeri, string OdemeYeriId)

```

```

{
    try
    {
        string Tipi = "S";
        string Sql = "EXECUTE SP_FATURA_EKLE " + FaturaNo + ", " +
CariKodu + ", " + Formatlar.IngilizceTarihFormati(Tarih) + ", " +
Formatlar.IngilizceTarihFormati(VadeTarihi) + ", " + GenelToplam.Replace(",",".")
+ ", " + Turu + ", " + Tipi + ", " + OdemeYeri + ", " + OdemeYeriId + ", " +
Aciklama + ", " + Accounting_Software.frmAnaForm.UserId.ToString();
        VeriTabani.Isle(Sql);
        StokHareketleri.Ekle(FaturaNo, "C", Detaylar,
Formatlar.IngilizceTarihFormati(Tarih), Tipi);

```

```

        return true;

```

```

    }

```

```

    catch (Exception)

```

```

    {

```

```

        return false;

```

```

    }

```

```

}

```

```

public Boolean Guncelle(string Id, string FaturaNo, string CariKodu, string Tarih,
string VadeTarihi, string GenelToplam, string Turu, string Aciklama, DataTable
Detaylar, string OdemeYeri, string OdemeYeriId)

```

```

{

```

```

    try

```

```

    {

```

```

        string Tipi = "S";

```

```

        string Sql = "EXECUTE SP_FATURA_GUNCELLE " + Id + ", " +

```

```

FaturaNo + ", " + CariKodu + ", " + Formatlar.IngilizceTarihFormati(Tarih) + ", " +

```

```

Formatlar.IngilizceTarihFormati(VadeTarihi) + ", " + GenelToplam.Replace(",",".") +

```

```

", " + Turu + ", " + Tipi + ", " + OdemeYeri + ", " + OdemeYeriId + ", " + Aciklama

```

```

+ ", " + Accounting_Software.frmAnaForm.UserId.ToString();

```

```

        VeriTabani.Isle(Sql);

```

```

        if (FaturaDetaylariniSil(FaturaNo) == true)

```

```

        {

```

```

            StokHareketleri.Ekle(FaturaNo, "C", Detaylar,

```

```

Formatlar.IngilizceTarihFormati(Tarih), Tipi);

```

```

        }

```

```

        return true;

```

```

    }

```

```

    catch (Exception)

```

```

    {

```



```

        return false;
    }
}

public void Sil(string Id)
{
    string Sql = "EXEC SP_FATURA_SIL" + Id;
    VeriTabani.Isle(Sql);
}

public DataRow Ac(string FaturaNo)
{
    string Sql = "SELECT TOP 1 * FROM VWFATURALAR WHERE
FATURANO='" + FaturaNo + "'";
    return VeriTabani.SatirCek(Sql);
}

public DataTable KalemleriAc(string FaturaNo)
{
    string Sql = "SELECT * FROM VWSTOKHAR WHERE FATURANO='" +
FaturaNo + "'";
    return VeriTabani.TabloCek(Sql);
}

public Boolean FaturaDetaylariniSil(string FaturaNo)
{
    try
    {
        string Sql = "EXEC SP_FATURAKALEMLERI_SIL '" + FaturaNo + "'";
        VeriTabani.Isle(Sql);
        return true;
    }

    catch (Exception)
    {
        return false;
    }
}
}
}

```

#### **frmSatisFaturasi.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

```

```
using DevExpress.XtraEditors;
```

```
namespace Accounting_Software.FaturaModulu.Satis
```

```
{  
    public partial class frmSatisFaturasi : DevExpress.XtraEditors.XtraForm  
    {  
        Accounting_Software.StokModulu.Classlar.clsStokTanitimKarti Stok = new  
StokModulu.Classlar.clsStokTanitimKarti();  
        Accounting_Software.CariModulu.Classlar.clsCariTanitimKarti Cari = new  
CariModulu.Classlar.clsCariTanitimKarti();  
        Accounting_Software.Classlar.Ekranlar Ekranlar = new  
Accounting_Software.Classlar.Ekranlar();  
        Accounting_Software.Classlar.Mesajlar Mesajlar = new  
Accounting_Software.Classlar.Mesajlar();  
        Accounting_Software.BankaModulu.Classlar.clsBanka Banka = new  
BankaModulu.Classlar.clsBanka();  
        Accounting_Software.KasaModulu.Classlar.clsKasa Kasa = new  
KasaModulu.Classlar.clsKasa();  
        Dizaynlar.DizaynYazdir Yazdir = new Dizaynlar.DizaynYazdir();  
        Classlar.clsSatisFaturasi SatisFaturasi = new Classlar.clsSatisFaturasi();  
        Classlar.clsNumaralar Numaralar = new Classlar.clsNumaralar();  
  
        Boolean Edit = false;  
        string UrunId, CariId;  
        string SecilenBankaId = "-1";  
        string SecilenKasaId = "-1";  
        string SecilenBankaAdi = "";  
        string SecilenKasaAdi = "";  
        string AcilacakFaturaNo = "-1";  
        string AcilacakFaturaId = "-1";  
        Boolean KapaliFatura = false;  
        string OdemeTuru = "KASA";  
  
        public frmSatisFaturasi()  
        {  
            InitializeComponent();  
        }  
  
        private void frmSatisFaturasi_Load(object sender, EventArgs e)  
        {  
            YeniFatura();  
        }  
  
        private void btnStokSec_ButtonClick(object sender,  
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)  
        {  
            StokSec();  
        }  
  
        void StokSec()
```

```

{
    UrunId = Ekranlar.StokListesiAc(true);

    if (UrunId != "")
    {
        DataRow Satir = Stok.Ac(UrunId);

        gridView1.AddNewRow();
        gridView1.SetRowCellValue(gridView1.FocusedRowHandle,
        "STOKKODU", (Satir["STOKKODU"].ToString()));
        gridView1.SetRowCellValue(gridView1.FocusedRowHandle, "STOKADI",
        (Satir["STOKADI"].ToString()));
        gridView1.SetRowCellValue(gridView1.FocusedRowHandle, "BARKOD",
        (Satir["BARKOD"].ToString()));
        gridView1.SetRowCellValue(gridView1.FocusedRowHandle,
        "BIRIMFIYAT", (Satir["SATISFIYAT1"].ToString()));
        gridView1.SetRowCellValue(gridView1.FocusedRowHandle, "BIRIM",
        (Satir["BIRIM"].ToString()));
        gridView1.SetRowCellValue(gridView1.FocusedRowHandle, "KDV",
        (Satir["SATISKDV"].ToString()));
    }
}

void Hesapla()
{
    decimal BirimFiyat, Miktar, Isk1, Isk2, Toplam, IskontoOncesiToplam,
    IskontoToplam, KdvOrani, KdvToplam, SatirToplam;

    try
    {
        IskontoOncesiToplam = 0;
        Toplam = 0;
        KdvToplam = 0;

        for (int i = 0; i < gridView1.RowCount; i++)
        {
            BirimFiyat = Convert.ToDecimal(gridView1.GetRowCellValue(i,
            "BIRIMFIYAT").ToString());
            Miktar = Convert.ToDecimal(gridView1.GetRowCellValue(i,
            "MIKTAR").ToString());
            KdvOrani = Convert.ToDecimal(gridView1.GetRowCellValue(i,
            "KDV").ToString());

            IskontoOncesiToplam = IskontoOncesiToplam + (BirimFiyat * Miktar);
            SatirToplam = Convert.ToDecimal(gridView1.GetRowCellValue(i,
            "TOPLAM").ToString());
            Toplam += SatirToplam;
            KdvToplam += SatirToplam / 100 * KdvOrani;
        }
    }
}

```



```

txtAraToplam.Text = IskontoOncesiToplam.ToString("0.00");
txtNetFiyat.Text = Toplam.ToString("0.00");
IskontoToplam = Toplam / IskontoOncesiToplam;
IskontoToplam = (1 - IskontoToplam) * 100;
txtIskonto.Text = IskontoToplam.ToString("0.00");
txtKdv.Text = KdvToplam.ToString("0.00");
txtGenelToplam.Text = (KdvToplam + Toplam).ToString("0.00");
}

catch (Exception)
{

}

}

private void gridView1_CellValueChanged(object sender,
DevExpress.XtraGrid.Views.Base.CellValueChangedEventArgs e)
{
    if (e.Column.Name != "TOPLAM")
    {
        try
        {
            decimal BirimFiyat, Miktar, Isk1, Isk2;

            BirimFiyat =
Convert.ToDecimal(gridView1.GetRowCellValue(gridView1.FocusedRowHandle,
"BIRIMFIYAT").ToString());
            Miktar =
Convert.ToDecimal(gridView1.GetRowCellValue(gridView1.FocusedRowHandle,
"MIKTAR").ToString());
            Isk1 =
Convert.ToDecimal(gridView1.GetRowCellValue(gridView1.FocusedRowHandle,
"ISK1").ToString());
            Isk2 =
Convert.ToDecimal(gridView1.GetRowCellValue(gridView1.FocusedRowHandle,
"ISK2").ToString());

            decimal AraToplam;

            AraToplam = BirimFiyat * Miktar;
            AraToplam = AraToplam - (AraToplam / 100 * Isk1);
            AraToplam = AraToplam - (AraToplam / 100 * Isk2);

            gridView1.SetRowCellValue(gridView1.FocusedRowHandle, "TOPLAM",
AraToplam);

            Hesapla();
        }

        catch (Exception)

```

```

    {
    }
}

private void gridView1_InitNewRow(object sender,
DevExpress.XtraGrid.Views.Grid.InitNewRowEventArgs e)
{
    gridView1.SetRowCellValue(gridView1.FocusedRowHandle, "MIKTAR", 1);
    gridView1.SetRowCellValue(gridView1.FocusedRowHandle, "ISK1", 0);
    gridView1.SetRowCellValue(gridView1.FocusedRowHandle, "ISK2", 0);
}

private void gridView1_RowCountChanged(object sender, EventArgs e)
{
    Hesapla();
}

private void btnCariSec_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
{
    CariId = Ekranlar.CariListesiAc();
    DataRow Satir = Cari.Ac(CariId);
    btnCariSec.Text = Satir["CARIKODU"].ToString();
    txtCariAdi.Text = Satir["CARIISIM"].ToString();
}

private void btnKaydet_Click(object sender, EventArgs e)
{
    Kaydet();
}

void Kaydet()
{
    Classlar.clsSatisFaturasi SatisFaturasi = new Classlar.clsSatisFaturasi();

    string Turu = "K";

    if (comTuru.SelectedIndex == 0)
    {
        Turu = "A";
    }

    string OdemeYeri = "";
    string OdemeYeriId = "-1";

    if (comOdemeTuru.SelectedIndex == 1)
    {
        OdemeYeri = "B";
        OdemeYeriId = SecilenBankaId;
    }
}

```

```

    }

    else
    {
        OdemeYeri = "K";
        OdemeYeriId = SecilenKasaId;
    }

    DataTable StokKalemleri = new DataTable();
    StokKalemleri = (DataTable)Liste.DataSource;

    if (Edit == true)
    {
        if (SatisFaturasi.Guncelle(AcilacakFaturaId, txtFaturaNo.Text,
            btnCariSec.Text, txtTarih.Text, txtVadeTarihi.Text, txtGenelToplam.Text, Turu,
            txtAciklama.Text, StokKalemleri, OdemeYeri, OdemeYeriId) == true)
        {
            Mesajlar.Bilgi("Faturanız başarılı bir şekilde güncellenmiştir.");
        }

        else
        {
            Mesajlar.Hata();
        }
    }

    else
    {
        if (SatisFaturasi.Ekle(txtFaturaNo.Text, btnCariSec.Text, txtTarih.Text,
            txtVadeTarihi.Text, txtGenelToplam.Text, Turu, txtAciklama.Text, StokKalemleri,
            OdemeYeri, OdemeYeriId) == true)
        {
            Mesajlar.Bilgi("Faturanız başarılı bir şekilde kaydedilmiştir.");
        }

        else
        {
            Mesajlar.Hata();
        }
    }

    if (cheBasim.Checked == true)
    {
        Yazdir.SatisFaturasi(txtFaturaNo.Text);
    }

    YeniFatura();
}

private void btnSec_Click(object sender, EventArgs e)

```



```

{
    try
    {
        AcilacakFaturaNo = Ekranlar.SatisFaturasiListesi();
        DataRow Ac = SatisFaturasi.Ac(AcilacakFaturaNo);
        AcilacakFaturaId = Ac["ID"].ToString();
        txtAciklama.Text = Ac["ACIKLAMA"].ToString();
        txtCariAdi.Text = Ac["CARIISIM"].ToString();
        btnCariSec.Text = Ac["CARIKODU"].ToString();
        txtFaturaNo.Text = AcilacakFaturaNo;

        DateTime Tarih = Convert.ToDateTime(Ac["TARIH"].ToString());
        DateTime VadeTarihi =
Convert.ToDateTime(Ac["VADETARIHI"].ToString());

        txtTarih.EditValue = Tarih;
        txtVadeTarihi.EditValue = VadeTarihi;

        string Turu = Ac["TURU"].ToString();

        if (Turu == "A")
        {
            comTuru.SelectedIndex = 0;
            PanelFaturaKapatma.Visible = false;
        }

        else
        {
            comTuru.SelectedIndex = 1;
            PanelFaturaKapatma.Visible = true;

            string OdemeYeri = Ac["ODEMEYERI"].ToString();

            if (OdemeYeri == "B")
            {
                OdemeTuru = "BANKA";
                lblYer.Text = "Banka Seçimi";
                SecilenBankaAdi = Ac["TAHSILYERI"].ToString();
                txtHedef.Text = SecilenBankaAdi;
                comOdemeTuru.SelectedIndex = 1;
            }

            else
            {
                OdemeTuru = "KASA";
                lblYer.Text = "Kasa Seçimi";
                SecilenKasaAdi = Ac["TAHSILYERI"].ToString();
                txtHedef.Text = SecilenKasaAdi;
                comOdemeTuru.SelectedIndex = 0;
            }
        }
    }
}

```

```

    }

    Liste.DataSource = SatisFaturasi.KalemleriAc(AcilacakFaturaNo);
    Edit = true;
    btnYazdir.Enabled = true;
}

catch (Exception)
{
    Edit = false;
    btnYazdir.Enabled = false;
}
}

void YeniFatura()
{
    btnYazdir.Enabled = false;
    txtTarih.Text = DateTime.Now.ToShortDateString();
    txtVadeTarihi.Text = DateTime.Now.ToShortDateString();
    txtFaturaNo.Text = Numaralar.GetYeniSatisFaturasiNumarasi();
    Liste.DataSource = SatisFaturasi.KalemleriAc("-1");
    txtAciklama.Text = "";
    txtCariAdi.Text = "";
    btnCariSec.Text = "";
    txtTarih.EditValue = DateTime.Now.ToShortDateString();
    txtVadeTarihi.EditValue = DateTime.Now.ToShortDateString();
    comTuru.SelectedIndex = 0;
    txtFaturaNo.Focus();
    Edit = false;
}

private void btnSil_Click(object sender, EventArgs e)
{
    if (Edit == true)
    {
        if (Mesajlar.Sor("Faturayı silmek istediğinizden emin misiniz?") == true)
        {
            SatisFaturasi.Sil(AcilacakFaturaId);
            YeniFatura();
        }
    }
}

private void comTuru_SelectedIndexChanged(object sender, EventArgs e)
{
    if (comTuru.SelectedIndex == 1)
    {
        KapaliFatura = true;
        PanelFaturaKapatma.Visible = true;
    }
}

```

```

else
{
    KapaliFatura = false;
    PanelFaturaKapatma.Visible = false;
}
}

```

```

private void comOdemeTuru_SelectedIndexChanged(object sender, EventArgs e)

```

```

{
    if (comOdemeTuru.SelectedIndex == 1)
    {
        OdemeTuru = "BANKA";
        lblYer.Text = "Banka Seçimi";
        txtHedef.Text = SecilenBankaAdi;
    }

```

```

else
{
    OdemeTuru = "KASA";
    lblYer.Text = "Kasa Seçimi";
    txtHedef.Text = SecilenKasaAdi;
}
}

```

```

void BankaSec()

```

```

{
    SecilenBankaId = Ekranlar.BankaListesi(true);
    DataRow Satir = Banka.Ac(SecilenBankaId);
    SecilenBankaAdi = Satir["BANKAADI"].ToString();
    txtHedef.Text = SecilenBankaAdi;
}

```

```

void KasaSec()

```

```

{
    SecilenKasaId = Ekranlar.KasaListesi(true);
    DataRow Satir = Kasa.Ac(SecilenKasaId);
    SecilenKasaAdi = Satir["ADI"].ToString();
    txtHedef.Text = SecilenKasaAdi;
}

```

```

private void txtHedef_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)

```

```

{
    if (OdemeTuru == "KASA")
    {
        KasaSec();
    }

```

```

else if (OdemeTuru == "BANKA")

```



```

    {
        BankaSec();
    }
}

private void btnYazdir_Click(object sender, EventArgs e)
{
    Yazdir.SatisFaturasi(AcilacakFaturaNo);
}
}
}

```

### **clsKasa.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;

namespace Accounting_Software.KasaModulu.Classlar
{
    class clsKasa
    {
        Accounting_Software.Classlar.Veritabani VeriTabani = new
        Accounting_Software.Classlar.Veritabani();
        Accounting_Software.Classlar.Mesajlar Mesajlar = new
        Accounting_Software.Classlar.Mesajlar();

        public Boolean Ekle(string Kodu, string Adi, string Aciklama, string SaveUser )
        {
            string Sql = "EXECUTE SP_KASA_EKLE '"+ Kodu + "','"+ Adi + "','+
            Aciklama + "','+ SaveUser +'"";

            try
            {
                VeriTabani.Isle(Sql);
                return true;
            }

            catch (Exception)
            {
                return false;
            }
        }

        public Boolean Guncelle(string Id, string Kodu, string Adi, string Aciklama, string
        SaveUser)
        {

```

```
        string Sql = "EXECUTE SP_KASA_GUNCELLE " + Id + ", '" + Kodu + "','" +  
Adi + "','" + Aciklama + "','" + SaveUser + "'";
```

```
    try  
    {  
        VeriTabani.Isle(Sql);  
        return true;  
    }
```

```
    catch (Exception)  
    {  
        return false;  
    }
```

```
}
```

```
public Boolean Sil(string Id)
```

```
{  
    string Sql = "EXECUTE SP_KASA_SIL " + Id ;
```

```
    if (HareketVarMi(KasaKoduAl(Id)) == true)  
    {  
        Mesajlar.Uyari("Hareketi olan kasayı silemezsiniz!!!");  
        return false;  
    }
```

```
    try  
    {  
        VeriTabani.Isle(Sql);  
        return true;  
    }
```

```
    catch (Exception)  
    {  
        return false;  
    }
```

```
}
```

```
public DataTable Listele(string Kodu, string Adi, string Aciklama)
```

```
{  
    string Sql = "SELECT * FROM TBLKASA WHERE KODU LIKE '%" + Kodu  
+ "%' AND ADI LIKE '%" + Adi + "%' AND ACIKLAMA LIKE '%" + Aciklama +  
"%' ORDER BY ADI";
```

```
    return VeriTabani.TabloCek(Sql);  
}
```

```
public string IdAl(string Adi)
```

```
{  
    try  
    {
```

```

        string Sql = "SELECT TOP 1 ID FROM TBLKASA WHERE ADI = '" + Adi
+ ""';
        return VeriTabani.SatirCek(Sql)["ID"].ToString();
    }

    catch (Exception)
    {
        return "-1";
    }
}

public string KasaKoduAl(string Id)
{
    try
    {
        string Sql = "SELECT TOP 1 KODU FROM TBLKASA WHERE ID = '" +
Id + ""';
        return VeriTabani.SatirCek(Sql)["KODU"].ToString();
    }

    catch (Exception)
    {
        return "-1";
    }
}

public string IdAl_KoduIle(string KasaKodu)
{
    try
    {
        string Sql = "SELECT TOP 1 ID FROM TBLKASA WHERE KODU = '" +
KasaKodu + ""';
        return VeriTabani.SatirCek(Sql)["ID"].ToString();
    }

    catch (Exception)
    {
        return "-1";
    }
}

public DataRow Ac(string Id)
{
    string Sql = "SELECT TOP 1 * FROM TBLKASA WHERE ID =" + Id;
    return VeriTabani.SatirCek(Sql);
}

Boolean HareketVarMi(string KasaKodu)
{

```



```

{
    if (Edit == true)
    {
        if (Kasa.Guncelle(AcilanKasaId, txtKasaKodu.Text, txtKasaAdi.Text,
txtAciklama.Text, frmAnaForm.UserId) == true)
        {
            Mesajlar.Bilgi("Kasa başarılı bir şekilde güncellenmiştir.");
            Temizle();
            Edit = false;
        }

        else
        {
            Mesajlar.Hata();
        }
    }

    else
    {
        if (Kasa.Ekle(txtKasaKodu.Text, txtKasaAdi.Text, txtAciklama.Text,
frmAnaForm.UserId) == true)
        {
            Mesajlar.Bilgi("Kasa başarılı bir şekilde tanımlanmıştır.");
            Temizle();
            Edit = false;
        }

        else
        {
            Mesajlar.Hata();
        }
    }

    Edit = false;
}

private void txtKasaKodu_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
{
    AcilanKasaId = Ekranlar.KasaListesi(true);
    Ac(AcilanKasaId);
}

public void Ac(string Id)
{
    AcilanKasaId = Id;

    DataRow Satir = Kasa.Ac(Id);

    txtAciklama.Text = Satir["ACIKLAMA"].ToString();
}

```

```

AcilanKasaKodu = Satir["KODU"].ToString();
txtKasaKodu.Text = AcilanKasaKodu;
txtKasaAdi.Text = Satir["ADI"].ToString();

GetKasaBakiye(AcilanKasaKodu);

Edit = true;
}

void GetKasaBakiye(string KasaKodu)
{
    ListeKasaBakiye.DataSource = Kasa.KasaBakiye(AcilanKasaKodu);
}

void Temizle()
{
    AcilanKasaId = "-1";
    AcilanKasaKodu = "-1";

    txtAciklama.Text = "";
    txtKasaAdi.Text = "";
    txtKasaKodu.Text = "";
    Edit = false;
    txtKasaKodu.Focus();

    GetKasaBakiye(AcilanKasaKodu);
}

private void btnSil_Click(object sender, EventArgs e)
{
    if (Mesajlar.Sor("Kasayı silmek istediğinizden emin misiniz?") == true)
    {
        if (Kasa.Sil(AcilanKasaId) == true)
        {
            Temizle();
        }
    }
}

private void btnYeni_Click(object sender, EventArgs e)
{
    if (txtKasaKodu.Text != "")
    {
        if (Mesajlar.Sor("Geçerli işlemi iptal etmek istediğinizden emin misiniz?") ==
true)
        {
            Temizle();
            Edit = false;
        }
    }
}

```

```

        else
        {
            Temizle();
            Edit = false;
        }
    }

    private void btnKapat_Click(object sender, EventArgs e)
    {
        if (Mesajlar.Sor("Geçerli işlemi iptal etmek istediğinizden emin misiniz?") ==
true)
        {
            this.Close();
        }
    }

    private void frmKasaTanitimKarti_Load(object sender, EventArgs e)
    {
    }
}

```

#### **clsKullanici.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;

namespace Accounting_Software.KullaniciModulu
{
    class clsKullanici
    {
        Accounting_Software.Classlar.Veritabani VeriTabani = new Classlar.Veritabani();

        public Boolean Ekle(string Adi, string Soyadi, string Unvani, string EMail, string
Tel, string KullaniciAdi, string Parola, string Aciklama, string UserId)
        {
            string Sql = "EXECUTE SP_KULLANICI_EKLE '" + Adi + "', '" + Soyadi + "',
'" + Unvani + "', '" + EMail + "', '" + Tel + "', '" + KullaniciAdi + "', '" + Parola + "', '" +
Aciklama + "', '" + UserId;
            return VeriTabani.Isle(Sql);
        }

        public Boolean Guncelle(string Id, string Adi, string Soyadi, string Unvani, string
EMail, string Tel, string KullaniciAdi, string Parola, string Aciklama, string UserId)
        {

```



```

        string Sql = "EXECUTE SP_KULLANICI_GUNCELLE " + Id + ", '" + Adi +
        ", '" + Soyadi + ", '" + Unvani + ", '" + EMail + ", '" + Tel + ", '" + KullaniciAdi + ",
        " + Parola + ", '" + Aciklama + ", '" + UserId;
        return VeriTani.Isle(Sql);
    }

    public Boolean Sil(string Id)
    {
        string Sql = "EXECUTE SP_KULLANICI_SIL " + Id;
        return VeriTani.Isle(Sql);
    }

    public DataTable Listele(string Adi, string Soyadi, string Unvani, string EMail)
    {
        string Sql = "SELECT * FROM TBLKULLANICILAR WHERE ADI LIKE
        '%" + Adi + "%' AND SOYADI LIKE '%" + Soyadi + "%' AND UNVANI LIKE '%" +
        Unvani + "%' AND EMAIL LIKE '%" + EMail + "%' ORDER BY ADI ASC";
        return VeriTani.TabloCek(Sql);
    }

    public DataRow Ac(string Id)
    {
        string Sql = "SELECT TOP 1 * FROM TBLKULLANICILAR WHERE ID=" +
        Id;
        return VeriTani.SatirCek(Sql);
    }
}

```

#### frmKullanici.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraEditors;

namespace Accounting_Software.KullaniciModulu
{
    public partial class frmKullanici : DevExpress.XtraEditors.XtraForm
    {
        Accounting_Software.Classlar.Mesajlar Mesajlar = new Classlar.Mesajlar();
        Accounting_Software.Classlar.Ekranlar Ekranlar = new Classlar.Ekranlar();
        clsKullanici Kullanici = new clsKullanici();

        public Boolean Edit = false;
        string AcilanKullaniciId = "-1";
    }
}

```

```

public frmKullanici()
{
    InitializeComponent();
}

public void Ac(string Id)
{
    AcilanKullaniciId = Id;
    DataRow Satir = Kullanici.Ac(AcilanKullaniciId);

    txtAciklama.Text = Satir["ACIKLAMA"].ToString();
    txtAdi.Text = Satir["ADI"].ToString();
    txtEMail.Text = Satir["EMAIL"].ToString();
    txtKullaniciAdi.Text = Satir["KULLANICIADI"].ToString();
    txtParola.Text = Satir["PAROLA"].ToString();
    txtSoyadi.Text = Satir["SOYADI"].ToString();
    txtTelefon.Text = Satir["TEL"].ToString();
    txtUnvani.Text = Satir["UNVANI"].ToString();

    Edit = true;
}

void Kaydet()
{
    if (Mesajlar.Sor("Kullanıcıyı kaydetmek istediğinizden emin misiniz?") == true)
    {
        if (Kullanici.Ekle(txtAdi.Text, txtSoyadi.Text, txtUnvani.Text, txtEMail.Text,
txtTelefon.Text, txtKullaniciAdi.Text, txtParola.Text, txtAciklama.Text,
frmAnaForm.UserId) == true)
        {
            Mesajlar.Bilgi("Kullanıcı başarılı bir şekilde kaydedildi.");
            Temizle();
        }
        else
        {
            Mesajlar.Hata();
        }
    }
}

void Guncelle()
{
    if (Mesajlar.Sor("Kullanıcıyı güncellemek istediğinizden emin misiniz?") ==
true)
    {
        if (Kullanici.Guncelle(AcilanKullaniciId, txtAdi.Text, txtSoyadi.Text,
txtUnvani.Text, txtEMail.Text, txtTelefon.Text, txtKullaniciAdi.Text, txtParola.Text,
txtAciklama.Text, frmAnaForm.UserId) == true)

```

```

    {
        Mesajlar.Bilgi("Kullanıcı başarılı bir şekilde güncellendi.");
        Temizle();
    }

    else
    {
        Mesajlar.Hata();
    }
}

void Sil()
{
    if (Mesajlar.Sor("Kullanıcıyı silmek istediğinizden emin misiniz?") == true)
    {
        if (Kullanici.Sil(AcilanKullaniciId) == true)
        {
            Mesajlar.Bilgi("Kullanıcı başarılı bir şekilde silindi.");
            Temizle();
        }

        else
        {
            Mesajlar.Hata();
        }
    }
}

void Temizle()
{
    txtAciklama.Text = "";
    txtAdi.Text = "";
    txtEMail.Text = "";
    txtKullaniciAdi.Text = "";
    txtParola.Text = "";
    txtSoyadi.Text = "";
    txtTelefon.Text = "";
    txtUnvani.Text = "";
    txtAdi.Focus();
}

private void btnKaydet_Click(object sender, EventArgs e)
{
    if (Edit == true)
    {
        Guncelle();
    }

    else

```



```

        {
            Kaydet();
        }
    }

    private void btnSil_Click(object sender, EventArgs e)
    {
        if (Edit == true)
        {
            Sil();
        }
    }

    private void btnKapat_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void btnYeni_Click(object sender, EventArgs e)
    {
        if (txtAdi.Text != "")
        {
            if (Mesajlar.Sor("Geçerli işlemi iptal etmek istediğinizden emin misiniz?") ==
true)
            {
                Edit = false;
                Temizle();
            }
        }
    }

    private void txtAdi_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
    {
        Ac(Ekranlar.KullaniciListesi(true));
    }

    private void frmKullanici_Load(object sender, EventArgs e)
    {
    }
}

```

#### **clsStokTanitimKarti.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```
using System.Data;
using System.Data.SqlClient;
```

```
namespace Accounting_Software.StokModulu.Classlar
{
```

```
    class clsStokTanitimKarti
    {
```

```
        Accounting_Software.Classlar.Veritabanı Database = new
Accounting_Software.Classlar.Veritabanı();
        Accounting_Software.Classlar.Mesajlar Mesajlar = new
Accounting_Software.Classlar.Mesajlar();
```

```
        public DataTable Listele(string StokKodu, string StokAdi, string Kod1, string
Kod2, string Kod3, string Kod4, string Kod5)
        {
            string Sql = "SELECT * FROM VWSTOKLAR WHERE STOKKODU LIKE
'" + StokKodu + "%' AND STOKADI LIKE '" + StokAdi + "%' AND KOD1 LIKE
'" + Kod1 + "%' AND KOD2 LIKE '" + Kod2 + "%' AND KOD3 LIKE '" + Kod3 +
%"' AND KOD4 LIKE '" + Kod4 + "%' AND KOD5 LIKE '" + Kod5 + "%'";
            DataTable VeriTablosu = Database.TabloCek(Sql);
            return VeriTablosu;
        }
```

```
        public void Ekle(string StokKodu, string StokAdi, string DigerIsim, string Kod1,
string Kod2, string Kod3, string Kod4, string Kod5, string Birim, string Barkod, string
AlisKdv, string SatisKdv, string AlisFiyat1, string AlisFiyat2, string SatisFiyat1, string
SatisFiyat2, string UserId)
        {
            string Sql = "EXECUTE SP_STOK_EKLE '" + StokKodu + "', '" + StokAdi + "',
'" + DigerIsim + "', '" + Kod1 + "', '" + Kod2 + "', '" + Kod3 + "', '" + Kod4 + "', '" +
Kod5 + "', '" + Birim + "', '" + Barkod + "', " + AlisKdv.Replace(",", ".").ToString() + ",
" + SatisKdv.Replace(",", ".").ToString() + ", " + AlisFiyat1.Replace(",", ".").ToString()
+ ", " + AlisFiyat2.Replace(",", ".").ToString() + ", " + SatisFiyat1.Replace(",",
".").ToString() + ", " + SatisFiyat2.Replace(",", ".").ToString() + ", " + UserId;
            Database.Isle(Sql);
        }
```

```
        public void Guncelle(string Id, string StokKodu, string StokAdi, string DigerIsim,
string Kod1, string Kod2, string Kod3, string Kod4, string Kod5, string Birim, string
Barkod, string AlisKdv, string SatisKdv, string AlisFiyat1, string AlisFiyat2, string
SatisFiyat1, string SatisFiyat2, string UserId)
        {
            string Sql = "EXECUTE SP_STOK_GUNCELLE " + Id + ", '" + StokKodu + "',
'" + StokAdi + "', '" + DigerIsim + "', '" + Kod1 + "', '" + Kod2 + "', '" + Kod3 + "', '" +
Kod4 + "', '" + Kod5 + "', '" + Birim + "', '" + Barkod + "', " + AlisKdv.Replace(",",
".").ToString() + ", " + SatisKdv.Replace(",", ".").ToString() + ", " +
AlisFiyat1.Replace(",", ".").ToString() + ", " + AlisFiyat2.Replace(",", ".").ToString() +
", " + SatisFiyat1.Replace(",", ".").ToString() + ", " + SatisFiyat2.Replace(",",
".").ToString() + ", " + UserId;
            Database.Isle(Sql);
        }
```

```

    }

    public Boolean Sil(string Id)
    {
        if (StokHareketiVarMi(Id) == true)
        {
            Mesajlar.Uyari("Hareket görmüş Stok Kartını silemezsiniz!!!");
            return false;
        }

        else
        {
            string Sql = "EXECUTE SP_STOK_SIL " + Id;
            Database.Isle(Sql);
            return true;
        }
    }

    public DataRow Ac(string Id)
    {
        string Sql = "SELECT * FROM VWSTOKLAR WHERE ID=" + Id;
        return Database.SatirCek(Sql);
    }

    string StokKoduAl(string Id)
    {
        string Sql = "SELECT STOKKODU FROM TBLSTOKLAR WHERE ID=" +
Id;
        return Database.SatirCek(Sql)["STOKKODU"].ToString();
    }

    public Boolean StokHareketiVarMi(string Id)
    {
        string StokKodu = StokKoduAl(Id);
        string Sql = "SELECT COUNT (*) AS ADET FROM VWSTOKHAR WHERE
STOKKODU=" + StokKodu + """;
        int Adet = Convert.ToInt32(Database.SatirCek(Sql)["ADET"].ToString());

        if (Adet > 0)
            return true;
        return false;
    }
}
}
}

```

**frmStokTanitimKarti.cs**

using System;



```

using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraEditors;

namespace Accounting_Software.StokModulu
{
    public partial class frmStokTanitimKarti : DevExpress.XtraEditors.XtraForm
    {
        Accounting_Software.Classlar.Ekranlar Ekranlar = new
Accounting_Software.Classlar.Ekranlar();
        Accounting_Software.StokModulu.Classlar.clsStokKod1 StokKod1 = new
Classlar.clsStokKod1();
        Accounting_Software.StokModulu.Classlar.clsStokKod2 StokKod2 = new
Classlar.clsStokKod2();
        Accounting_Software.StokModulu.Classlar.clsStokKod3 StokKod3 = new
Classlar.clsStokKod3();
        Accounting_Software.StokModulu.Classlar.clsStokKod4 StokKod4 = new
Classlar.clsStokKod4();
        Accounting_Software.StokModulu.Classlar.clsStokKod5 StokKod5 = new
Classlar.clsStokKod5();
        Classlar.clsStokTanitimKarti Stok = new Classlar.clsStokTanitimKarti();
        Accounting_Software.Classlar.Mesajlar Mesajlar = new
Accounting_Software.Classlar.Mesajlar();

        string UrunId;
        Boolean Edit = false;

        public frmStokTanitimKarti()
        {
            InitializeComponent();
        }

        private void frmStokTanitimKarti_Load(object sender, EventArgs e)
        {
        }

        private void btnKod1Sec_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
        {
            btnKod1Sec.Text = StokKod1.KoduAl(Ekranlar.StokKod1Ac(true));
        }

        private void btnKod2Sec_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
        {

```

```

        btnKod2Sec.Text = StokKod2.KoduAl(Ekranlar.StokKod2Ac(true));
    }

    private void btnKod3Sec_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
    {
        btnKod3Sec.Text = StokKod3.KoduAl(Ekranlar.StokKod3Ac(true));
    }

    private void btnKod4Sec_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
    {
        btnKod4Sec.Text = StokKod4.KoduAl(Ekranlar.StokKod4Ac(true));
    }

    private void btnKod5Sec_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
    {
        btnKod5Sec.Text = StokKod5.KoduAl(Ekranlar.StokKod5Ac(true));
    }

    private void btnKaydet_Click(object sender, EventArgs e)
    {
        if (Edit == true)
        {
            Stok.Guncelle(UrunId, txtStokKodu.Text, txtStokAdi.Text, txtDiger.Text,
StokKod1.IdAl(btnKod1Sec.Text), StokKod2.IdAl(btnKod2Sec.Text),
StokKod3.IdAl(btnKod3Sec.Text), StokKod4.IdAl(btnKod4Sec.Text),
StokKod5.IdAl(btnKod5Sec.Text), txtBirim.Text, txtBarkod.Text, txtAlisKdv.Text,
txtSatisKdv.Text, txtAlisFiyat1.Text, txtAlisFiyat2.Text, txtSatisFiyat1.Text,
txtSatisFiyat2.Text, frmAnaForm.UserId);
        }

        else
        {
            Stok.Ekle(txtStokKodu.Text, txtStokAdi.Text, txtDiger.Text,
StokKod1.IdAl(btnKod1Sec.Text), StokKod2.IdAl(btnKod2Sec.Text),
StokKod3.IdAl(btnKod3Sec.Text), StokKod4.IdAl(btnKod4Sec.Text),
StokKod5.IdAl(btnKod5Sec.Text), txtBirim.Text, txtBarkod.Text, txtAlisKdv.Text,
txtSatisKdv.Text, txtAlisFiyat1.Text, txtAlisFiyat2.Text, txtSatisFiyat1.Text,
txtSatisFiyat2.Text, frmAnaForm.UserId);
        }

        Temizle();
    }

    void Temizle()
    {
        txtStokKodu.Text = "";
        txtStokAdi.Text = "";
    }

```

```

txtDiger.Text = "";
txtAlisFiyat1.Text = "";
txtAlisFiyat2.Text = "";
txtAlisKdv.Text = "";
txtBarkod.Text = "";
txtBirim.SelectedIndex = -1;
txtSatisFiyat1.Text = "";
txtSatisFiyat2.Text = "";
txtSatisKdv.Text = "";
btnKod1Sec.Text = "";
btnKod2Sec.Text = "";
btnKod3Sec.Text = "";
btnKod4Sec.Text = "";
btnKod5Sec.Text = "";
}

```

```

private void btnStokSec_Click(object sender, EventArgs e)
{
    UrunId = Ekranlar.StokListesiAc(true);

    if (UrunId != "")
    {
        Ac(UrunId);
    }
}

```

```

public void Ac(string Id)
{
    UrunId = Id;
    DataRow Satir = Stok.Ac(UrunId);
    txtStokKodu.Text = Satir["STOKKODU"].ToString();
    txtStokAdi.Text = Satir["STOKADI"].ToString();
    txtAlisFiyat1.Text = Satir["ALISFIYAT1"].ToString();
    txtAlisFiyat2.Text = Satir["ALISFIYAT2"].ToString();
    txtAlisKdv.Text = Satir["ALISKDV"].ToString();
    txtBarkod.Text = Satir["BARKOD"].ToString();
    txtBirim.Text = Satir["BIRIM"].ToString();
    txtDiger.Text = Satir["DIGERISIM"].ToString();
    txtSatisFiyat1.Text = Satir["SATISFIYAT1"].ToString();
    txtSatisFiyat2.Text = Satir["SATISFIYAT2"].ToString();
    txtSatisKdv.Text = Satir["SATISKDV"].ToString();
    btnKod1Sec.Text = Satir["KOD1"].ToString();
    btnKod2Sec.Text = Satir["KOD2"].ToString();
    btnKod3Sec.Text = Satir["KOD3"].ToString();
    btnKod4Sec.Text = Satir["KOD4"].ToString();
    btnKod5Sec.Text = Satir["KOD5"].ToString();
    Edit = true;
}

```

```

private void btnSil_Click(object sender, EventArgs e)

```



```

{
    if (Mesajlar.Sor("Seçilen kaydı silmek istediğinizden emin misiniz?") == true)
    {
        Stok.Sil(UrunId);
        Temizle();
    }
}

private void btnYeni_Click(object sender, EventArgs e)
{
    Edit = false;
    txtStokKodu.Focus();
    Temizle();
}
}
}

```

#### frmAnaForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraBars;
using System.Diagnostics;

namespace Accounting_Software
{
    public partial class frmAnaForm : DevExpress.XtraBars.Ribbon.RibbonForm
    {
        /*TEMEL ERİŞİM DEĞİŞKENLERİ*/
        public static Boolean GirişDurumu = false;
        public static string AraDegiskenString;
        public static string UserId = "-1";
        public static string UserName = "";
        public static Boolean ErisimBilgileri = false;
        /*TEMEL ERİŞİM DEĞİŞKENLERİ*/

        Accounting_Software.KullaniciModulu.clsYetkiler Yetkiler = new
        KullaniciModulu.clsYetkiler();
        Accounting_Software.Classlar.Ekranlar Ekranlar = new Classlar.Ekranlar();

        public frmAnaForm()
        {
            InitializeComponent();
        }
    }
}

```

```

private void frmAnaForm_Load(object sender, EventArgs e)
{
    this.Hide();
    GirişIslemi();
    frmMasaüstü Form = new frmMasaüstü();
    Form.MdiParent = this;
    Form.Show();
}

private void barStokTanitimKarti_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.StokTanitimKartiAc();
}

private void barCariTanitimKarti_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.CariTanitimKartiAc();
}

private void barTopluStokAcma_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.TopluStokAcma();
}

private void barSatisFaturasi_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.SatisFaturasiAc();
}

private void barCariHareketKayitlari_ItemClick(object sender,
ItemClickEventArgs e)
{
    Ekranlar.CariHareketKayitlari(false);
}

private void barCariAcilisKarti_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.CariAcilisKarti();
}

private void barCariListesi_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.CariListesiAc(false);
}

private void barStokHareketKayitlari_ItemClick(object sender,
ItemClickEventArgs e)
{
    Ekranlar.StokHareketKayitlari();
}

```

```

private void barStokAcilisKarti_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.StokAcilisKarti();
}

private void barStokListesi_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.StokListesiAc(false);
}

private void barTopluStokGuncelleme_ItemClick(object sender,
ItemClickEventArgs e)
{
    Ekranlar.TopluStokGuncelleme();
}

private void barStokKod1_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.StokKod1Ac(false);
}

private void barStokKod2_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.StokKod2Ac(false);
}

private void barStokKod3_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.StokKod3Ac(false);
}

private void barStokKod4_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.StokKod4Ac(false);
}

private void barStokKod5_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.StokKod5Ac(false);
}

e) private void barKasaTanitimKarti_ItemClick(object sender, ItemClickEventArgs
{
    Ekranlar.KasaTanitimKarti();
}

private void barKasaListesi_ItemClick(object sender, ItemClickEventArgs e)
{

```



```

    Ekranlar.KasaListesi(false);
}

private void barKasaHareketKayitlari_ItemClick(object sender,
ItemClickEventArgs e)
{
    Ekranlar.KasaHareketKayitlari();
}

private void barKasaAcilisKarti_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.KasaAcilisKarti();
}

private void barTahsilatGirisi_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.KasaTahsilatGirisi();
}

private void barKasaOdemeGirisi_ItemClick(object sender, ItemClickEventArgs
e)
{
    Ekranlar.KasaOdemeGirisi();
}

private void barBankaTanitimKarti_ItemClick(object sender, ItemClickEventArgs
e)
{
    Ekranlar.BankaTanitimKarti();
}

private void barBankaListesi_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.BankaListesi();
}

private void barBankaAcilisKarti_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.BankaAcilisKarti();
}

private void barBankaHareketKayitlari_ItemClick(object sender,
ItemClickEventArgs e)
{
    Ekranlar.BankaHareketKayitlari();
}

private void barBankaIslemi_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.BankaIslemi();
}

```

```

    }

    private void barGelenHavale_ItemClick(object sender, ItemClickEventArgs e)
    {
        Ekranlar.GelenHavale();
    }

    private void barGonderilenHavale_ItemClick(object sender, ItemClickEventArgs
e)
    {
        Ekranlar.GonderilenHavale();
    }

    private void barMusteriCekGirisi_ItemClick(object sender, ItemClickEventArgs e)
    {
        Ekranlar.MusteriCekGirisi();
    }

    private void barMusteriBordroGirisi_ItemClick(object sender,
ItemClickEventArgs e)
    {
        Ekranlar.MusteriCekBordrosuGirisi();
    }

    private void barBordroListesi_ItemClick(object sender, ItemClickEventArgs e)
    {
        Ekranlar.CekBordroListesi();
    }

    private void barMusteriCekListesi_ItemClick(object sender, ItemClickEventArgs
e)
    {
        Ekranlar.MusteriCekListesi();
    }

    private void barBankayaCekCikisi_ItemClick(object sender, ItemClickEventArgs
e)
    {
        Ekranlar.BankayaCekCikisi();
    }

    private void barCariyeCekCikisi_ItemClick(object sender, ItemClickEventArgs e)
    {
        Ekranlar.CariyeCekCikisi();
    }

    private void barKendiCekGirisimiz_ItemClick(object sender, ItemClickEventArgs
e)
    {
        Ekranlar.KendiCekGirisi(false);
    }

```

```

    }

    private void barKendiCekListemiz_ItemClick(object sender, ItemClickEventArgs
e)
    {
        Ekranlar.KendiCekListesi();
    }

    private void barMusteriSenetGirisi_ItemClick(object sender, ItemClickEventArgs
e)
    {
        Ekranlar.MusteriSenetGirisi();
    }

    private void barMusteriBordroGirisimiz_ItemClick(object sender,
ItemClickEventArgs e)
    {
        Ekranlar.MusteriSenetBordrosuGirisi();
    }

    private void barSenetBordroListesi_ItemClick(object sender, ItemClickEventArgs
e)
    {
        Ekranlar.SenetBordroListesi();
    }

    private void barMusteriSenetListesi_ItemClick(object sender, ItemClickEventArgs
e)
    {
        Ekranlar.MusteriSenetListesi();
    }

    private void barBankayaSenetCikisi_ItemClick(object sender, ItemClickEventArgs
e)
    {
        Ekranlar.BankayaSenetCikisi();
    }

    private void barCariyeSenetCikisi_ItemClick(object sender, ItemClickEventArgs
e)
    {
        Ekranlar.CariyeSenetCikisi();
    }

    private void barKendiSenetGirisimiz_ItemClick(object sender,
ItemClickEventArgs e)
    {
        Ekranlar.KendiSenetGirisi();
    }

```



```

e) private void barKendiSenetListemiz_ItemClick(object sender, ItemClickEventArgs
    {
        Ekranlar.KendiSenetListesi();
    }

private void barSatisIadeFaturasi_ItemClick(object sender, ItemClickEventArgs e)
    {
        Ekranlar.SatisIadeFaturasiAc();
    }

private void barAlisFaturasi_ItemClick(object sender, ItemClickEventArgs e)
    {
        Ekranlar.AlisFaturasiAc();
    }

private void barAlisIadeFaturasi_ItemClick(object sender, ItemClickEventArgs e)
    {
        Ekranlar.AlisIadeFaturasiAc();
    }

private void barStokHareketRaporu_ItemClick(object sender, ItemClickEventArgs
e)
    {
        Ekranlar.Rapor_StokHareket();
    }

private void barCariStokHareketRaporu_ItemClick(object sender,
ItemClickEventArgs e)
    {
        Ekranlar.Rapor_CariStokHareket();
    }

private void barTarihAraligi_ItemClick(object sender, ItemClickEventArgs e)
    {
        Ekranlar.Rapor_TarihAraligiStokHareket();
    }

private void barCariHareketRaporu_ItemClick(object sender, ItemClickEventArgs
e)
    {
        Ekranlar.Rapor_CariHareket();
    }

private void barOzelCariRaporu_ItemClick(object sender, ItemClickEventArgs e)
    {
        Ekranlar.Rapor_OzelCariRaporu();
    }

```

```

private void barSatisFaturasiListesi_ItemClick(object sender, ItemClickEventArgs
e)
{
    Ekranlar.SatisFaturasiListesi(false);
}

private void barSatisIadeFaturasiListesi_ItemClick(object sender,
ItemClickEventArgs e)
{
    Ekranlar.SatisIadeFaturasiListesi(false);
}

private void barAlisFaturasiListesi_ItemClick(object sender, ItemClickEventArgs
e)
{
    Ekranlar.AlisFaturasiListesi(false);
}

private void barAlisIadeFaturasiListesi_ItemClick(object sender,
ItemClickEventArgs e)
{
    Ekranlar.AlisIadeFaturasiListesi(false);
}

private void barKullaniciEkle_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.KullaniciEkle();
}

private void barKullaniciListesi_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.KullaniciListesi(false);
}

private void barKullaniciYetkileri_ItemClick(object sender, ItemClickEventArgs
e)
{
    Ekranlar.KullaniciYetkileri();
}

void GirişIslemi()
{
    frmGirisFormu GirisFormu = new frmGirisFormu();
    GirisFormu.ShowDialog();

    if (GirisDurumu == false)
    {
        Application.Exit();
    }
}

```

```

else
{
    this.Opacity = 100;
    this.Show();
}
}

private void barParolaDegistir_ItemClick(object sender, ItemClickEventArgs e)
{
    Ekranlar.ParolaDegistir();
}

private void barMicrosoftExcel_ItemClick(object sender, ItemClickEventArgs e)
{
    Process.Start("excel");
}

private void barMicrosoftWord_ItemClick(object sender, ItemClickEventArgs e)
{
    Process.Start("winword");
}

private void barPaint_ItemClick(object sender, ItemClickEventArgs e)
{
    Process.Start("mspaint");
}

private void barNotDefteri_ItemClick(object sender, ItemClickEventArgs e)
{
    Process.Start("notepad");
}

private void barHesapMakinesi_ItemClick(object sender, ItemClickEventArgs e)
{
    Process.Start("calc");
}

private void barWeb_ItemClick(object sender, ItemClickEventArgs e)
{
    Process.Start("http://www.neu.edu.tr");
}

private void barHesapMakinesi1_ItemClick(object sender, ItemClickEventArgs e)
{
    Process.Start("calc");
}

private void barNotDefteri1_ItemClick(object sender, ItemClickEventArgs e)
{
    Process.Start("notepad");
}

```



```

    }
}
}

```

## frmGirisFormu.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraEditors;

namespace Accounting_Software
{
    public partial class frmGirisFormu : DevExpress.XtraEditors.XtraForm
    {
        Accounting_Software.Classlar.Ekranlar Ekranlar = new Classlar.Ekranlar();
        Accounting_Software.Classlar.Mesajlar Mesajlar = new Classlar.Mesajlar();
        Accounting_Software.Classlar.GirisIslemi Giris = new Classlar.GirisIslemi();

        public frmGirisFormu()
        {
            InitializeComponent();
        }

        private void frmGirisFormu_Load(object sender, EventArgs e)
        {
            /*SetStyle(ControlStyles.SupportsTransparentBackColor, true);
            BackColor = Color.Gray;
            TransparencyKey = BackColor;*/

            if (Accounting_Software.Properties.Settings.Default.Animsa == true)
            {
                txtKullaniciAdi.Text =
Accounting_Software.Properties.Settings.Default.KullaniciAdi;
                txtParola.Text = Accounting_Software.Properties.Settings.Default.Parola;
                cheAnimsa.Checked = true;
            }
        }

        private void btnIptal_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnGiris_Click(object sender, EventArgs e)
        {

```

```

        if (Giris.KullaniciGirisi(txtKullaniciAdi.Text, txtParola.Text))
        {
            Accounting_Software.Properties.Settings.Default.Animsa =
(bool)cheAnimsa.Checked;
            Accounting_Software.Properties.Settings.Default.KullaniciAdi =
txtKullaniciAdi.Text;
            Accounting_Software.Properties.Settings.Default.Parola = txtParola.Text;
            Accounting_Software.Properties.Settings.Default.Save();
            this.Close();
        }

        else
        {
            if (frmAnaForm.ErisimBilgileri == false)
            {
                Mesajlar.Uyari("Yanlış giriş yaptınız. Lütfen tekrar deneyiniz.");
            }
        }
    }

    private void btnServerAyarlari_Click(object sender, EventArgs e)
    {
        Ekranlar.ServerAyarlari();
    }
}

```

#### **frmMasaustu.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraEditors;

```

```

namespace Accounting_Software
{

```

```

    public partial class frmMasaüstü : DevExpress.XtraEditors.XtraForm
    {
        Accounting_Software.Classlar.Ekranlar Ekranlar = new Classlar.Ekranlar();

        public frmMasaüstü()
        {
            InitializeComponent();
        }
    }

```

```
private void navStokTanitimKarti_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
{
    Ekranlar.StokTanitimKartiAc();
}
```

```
private void navStokListesi_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
{
    Ekranlar.StokListesiAc();
}
```

```
private void navStokHareketleri_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
{
    Ekranlar.StokHareketKayitlari();
}
```

```
private void navTopluStokAcma_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
{
    Ekranlar.TopluStokAcma();
}
```

```
private void navTopluStokGuncelleme_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
{
    Ekranlar.TopluStokGuncelleme();
}
```

```
private void navStokKod1_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
{
    Ekranlar.StokKod1Ac(false);
}
```

```
private void navStokKod2_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
{
    Ekranlar.StokKod2Ac(false);
}
```

```
private void navStokKod3_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
{
    Ekranlar.StokKod3Ac(false);
}
```

```
private void navStokKod4_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
```



```

    {
        Ekranlar.StokKod4Ac(false);
    }

    private void navStokKod5_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
    {
        Ekranlar.StokKod5Ac(false);
    }

    private void navCariTanitimKarti_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
    {
        Ekranlar.CariTanitimKartiAc();
    }

    private void navCariListesi_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
    {
        Ekranlar.CariListesiAc();
    }

    private void navCariHareket_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
    {
        Ekranlar.CariHareketKayitlari();
    }

    private void navCariAcilisKarti_LinkClicked(object sender,
DevExpress.XtraNavBar.NavBarLinkEventArgs e)
    {
        Ekranlar.CariAcilisKarti();
    }
}

```

#### **frmParolaDegistir.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraEditors;

namespace Accounting_Software
{
    public partial class frmParolaDegistir : DevExpress.XtraEditors.XtraForm

```

```

{
    Accounting_Software.Classlar.GirisIslemi Parola = new Classlar.GirisIslemi();
    Accounting_Software.Classlar.Mesajlar Mesajlar = new Classlar.Mesajlar();
    public frmParolaDegistir()
    {
        InitializeComponent();
    }

    private void btnKaydet_Click(object sender, EventArgs e)
    {
        if (txtYeni.Text == txtTekrar.Text)
        {
            if (Parola.ParolaDegistir(frmAnaForm.UserId, txtEski.Text, txtYeni.Text) ==
true)
            {
                Mesajlar.Bilgi("Parolanız başarılı bir şekilde güncellenmiştir.");
                this.Close();
            }
        }
        else
        {
            Mesajlar.Uyari("Girmiş olduğunuz parolalar uyuşmamaktadır. Lütfen tekrar
deneyiniz.");
        }
    }
}

```

#### **frmServerAyarlari.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using DevExpress.XtraEditors;

namespace Accounting_Software
{
    public partial class frmServerAyarlari : DevExpress.XtraEditors.XtraForm
    {
        public frmServerAyarlari()
        {
            InitializeComponent();
        }

        private void btnKaydet_Click(object sender, EventArgs e)

```

```

    {
        Accounting_Software.Properties.Settings.Default.ServerIpAdresi =
txtSqlServerIp.Text;
        Accounting_Software.Properties.Settings.Default.VeriTabani =
txtVeriTabaniAdi.Text;
        Accounting_Software.Properties.Settings.Default.VeriTabaniKullaniciAdi =
txtKullaniciAdi.Text;
        Accounting_Software.Properties.Settings.Default.VeriTabaniSifresi =
txtParola.Text;
        Accounting_Software.Properties.Settings.Default.Save();
        this.Close();
    }

    private void frmServerAyarlari_Load(object sender, EventArgs e)
    {
        txtSqlServerIp.Text =
Accounting_Software.Properties.Settings.Default.ServerIpAdresi;
        txtVeriTabaniAdi.Text =
Accounting_Software.Properties.Settings.Default.VeriTabani;
        txtKullaniciAdi.Text =
Accounting_Software.Properties.Settings.Default.VeriTabaniKullaniciAdi;
        txtParola.Text =
Accounting_Software.Properties.Settings.Default.VeriTabaniSifresi;
    }

    private void btnKapat_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}
}

```



## CONCLUSION

Previously, companies were worked accounting transactions with book. This is a waste of time and paperwork confusion. Nowadays accounting procedures are performed by software. The complexity of paperwork has been terminated by this software. But this software has been missing. I chose this topic in the project. I decided to develop in the field of accounting software. I tried to correct the deficiencies to a large extent. I have been partly successful. I have developed my knowledge in C # and Sql Server with my project. I have added new topics to knowledge base. this project aroused interest me against any other programming language. I want to develop my career in software. I will continue to produce such accounting software project.

## REFERENCES

- 1- [1] [http://en.wikipedia.org/wiki/Accounting\\_information\\_system](http://en.wikipedia.org/wiki/Accounting_information_system)
- 2- [2] [http://www.ehow.com/about\\_6328213\\_history-computerized-accounting.html](http://www.ehow.com/about_6328213_history-computerized-accounting.html)
- 3- [3] <http://smallbusiness.chron.com/advantages-disadvantages-computerized-accounting-4911.html>
- 4- [4] [http://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio)
- 5- [5] [http://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server)
- 6- [6] <http://en.wikipedia.org/wiki/DevExpress>