**Banking Console Application**

***A Project Report submitted in partial fulfilment of the***

***requirements for the award of the degree of***

**Bachelor of Technology**

**in**

**Computer Science and Engineering with Specialization in DA**

**by**

**Anmol Khandelwal**

**Roll No: 181520009**

Under the Guidance of

Department of Computer Engineering & Applications

**Institute of Engineering & Technology**





**GLA University**

**Mathura- 281406, INDIA**

**May, 2022**

**Department of Computer Engineering and Applications**

**GLA University, 17 km Stone, NH#2, Mathura-Delhi Road, P.O. Chaumuhan, Mathura-281406(U.P.)**

**Declaration**

I hereby declare that the work which is being presented in the B.Tech. Project **“Banking Console Application”**, in partial fulfilment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering with Specialization in DA and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my own work carried under the supervision of **Name & Designation of Supervisor(s).**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Candidate: Anmol Khandelwal

University Roll No: 181520009

**Certificate**

This is to certify that the above statements made by the candidate are correct to the best of my/our knowledge and belief.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Supervisor**

(

Designation of Supervisor

Dept. of Computer Engg, & App.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project Co-ordinator Program Co-ordinator**

**(Dr. Mayank Srivastava)** (**Dr Rakesh Kumar Galav**)

Associate Professor Assistant Professor

Dept. of Computer Engg, & App. Dept. of Computer Engg, & App.

Date:

**ACKNOWLEDGEMENT**

Presentation, inspiration and motivation have always played a key role in the success of any venture.

We would like to express sincere gratitude to our teachers, to encourage us to the highest peak, to provide us the opportunity to prepare this project and for providing their invaluable guidance and suggestions throughout the course of project. We are highly indebted to their constant supervision as well as for providing necessary information regarding the project and also their support in completing the project. We would also like to express our special gratitude towards our friends for their kind co-operation and encouragement which helped us in completion of this project.

Last, but not the least, our parents are also an important inspiration to us. So, with due regards, we express our gratitude to them.

Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Candidate: Anmol Khandelwal

University Roll No: 181520009

**ABSTRACT**

The Bank Console is an application for maintaining a person's account in a bank. In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System. To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the user’s work space to have additional functionalities which are not provided under a conventional banking project. The Bank Account Management System undertaken as a project is based on relevant technologies. The main aim of this project is to develop software for Bank Account Management System. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software. This project is developed using C# with .Net framework and SQL Express use for database connection. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization need to effectively define and manage requirements to ensure they are meeting needs of the customer, while proving compliance and staying on the schedule and within budget. The impact of a poorly expressed requirement can bring a business out of compliance or even cause injury or death. Requirements definition and management is an activity that can deliver a high, fast return on investment. The project analyzes the system requirements and then comes up with the requirements specifications. It studies other related systems and then come up with system specifications. The system is then designed in accordance with specifications to satisfy the requirements. The system design is then implemented with C# and SQL Express. The system is designed as an interactive and content management system. The content management system deals with data entry, validation confirm and updating whiles the interactive system deals with system interaction with the administration and users. Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

**CONTENTS**

Declaration i

Certificate ii

Acknowledge iii

Abstract iv

**CHAPTER 1 Introduction 1**

1.1 Overview and Motivation 1

1.2 Objective 1

1.3 Scope 3

**CHAPTER 2 Software Requirement Analysis 4**

2.1 Hardware Requirements 4

2.2 Software Requirements 4

2.3 Functional Requirements 5

2.4 Technologies Used 5

2.4.1 C# 5

2.4.2 SQL Server Database 6

2.4.3 ADO.Net 7

2.5 Workflow of Banking console application 10

2.6 Use Case Diagram 11

2.7 Activity Diagram 12

**CHAPTER 3 Software Design 13**

3.1 Architecture 13

3.2 Models and Its Properties 15

3.3 SQL Server database 18

**CHAPTER 4 Software Testing 21**

**CHAPTER 5 Conclusion 23**

**References 24**

**Chapter 1**

**Introduction**

* 1. **Overview and Motivation**

Bank console application is a simple console-based totally application developed using the C# programming language. Basically, this device includes a C# libraries and a database.

This device is a simple console-based totally device so it’s far very easy to understand and use. Talking about the machine, it includes all of the fundamental features required in a bank. There is a  login system as well in the project.

Besides, this means the person can all those available capabilities without difficulty without any restrictions. It’s too simple to use, the person can look at the facts of total financial institution accounts without problems.

Admins have some extensive access to check the all the bookings done till the date and can modify the room details and hotel details. He can add, update, delete the rooms info according to the need and he can change the availability of rooms as well.

This whole project is built on asp.net core using MVC (Model View Architecture). Visual studio code is the ide we use for it and we used Sql Server 2018 as an on-premise database for local database to check and test the functionality of our web app.

* 1. **Objective**

The goal of the bank management system project is to create an organic and optimal software of interaction between the various banking components.

This is to maximize the profit of the banking mechanism. The implementation of competent bank management procedures is significantly responsible for the successful optimization of the bank’s productivity and activities.  
  
The project’s main goal is to create an online banking system for banks. All banking work is done manually in the current system.

To withdraw or deposit money, the user must go to the bank. Today, it is also hard to find account information for people who have accounts in the banking system.

1. Improve customer care and service at the banks.
2. Increase the bank performance.
3. Reduce the operational costs of the bank.

* To enable online banking via the internet.
* To enable automated data entry methods.
* Ensure efficient and reliable communication within the bank.
* Avoid data entry errors by use of input masks.
* Enable easy authorized modification of data.
* Enforce security measures to avoid unauthorized access to account holder records.
* Enable fast and easy retrieval of account holder’s records and data for fast reference activities
  + - 1. To create Bank and account
      2. To deposit money
      3. To withdraw money
      4. To check balance
      5. To show transaction history
  1. **Scope**

The internet has developed a revolution in the field of bank application services.

Depending on the bank’s policies, bank personnel and/or customers can utilize the Banking Management System. It can be utilized by multiple employees at the same time if they have the necessary permissions. Any web browser with a supported software requirements can be used to access it.

**Chapter 2**

**Software Requirement Analysis**

**2.1 Hardware Requirements**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. The basic hardware requirements are-

* Laptop (above i3 Processor) with at least 4GB RAM
* GPU (usually preferred) with 1.5GHZ Processor with good internet connection.

**2.2 Software Requirements**

Software engineering technology defines a software requirement as a condition or capability needed by a user to solve a problem or achieve an objective. The basic software requirements are

* Intel Pentium 4 and Windows 2010
* Memory 8 GB
* Internet Explorer 11.0 or higher / Chrome
* SQL Express
* Visual Studio 2019
* C#
* .Net Core 3.1

**2.3 Functional Requirements**

* System shall show the error message to the user when the input given is not in the required format.
* It has different restrictions for different type of users like customer and admin.
* Some functionalities are hidden from the customer and extra functionalities are given to admin.
* Also, without registering onto the site user is not able to banking in the application.

**2.4.1. C#**

## What is C#?

C# is pronounced "C-Sharp".

It is an object-oriented programming language created by Microsoft that runs on the .NET Framework.

C# has roots from the C family, and the language is close to other popular languages like [C++](https://www.w3schools.com/cpp/default.asp) and [Java](https://www.w3schools.com/java/default.asp).

The first version was released in year 2002. The latest version, **C# 8**, was released in September 2019.

C# is used for:

* Mobile applications
* Desktop applications
* Web applications
* Web services
* Web sites
* Games
* VR
* Database applications
* And much, much more!

## Why Use C#?

* It is one of the most popular programming language in the world
* It is easy to learn and simple to use
* It has a huge community support
* C# is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs.
* As C# is close to C, [C++](https://www.w3schools.com/cpp/default.asp) and [Java](https://www.w3schools.com/java/default.asp), it makes it easy for programmers to switch to C# or vice versa

**2.4.2. SQL Server Database**

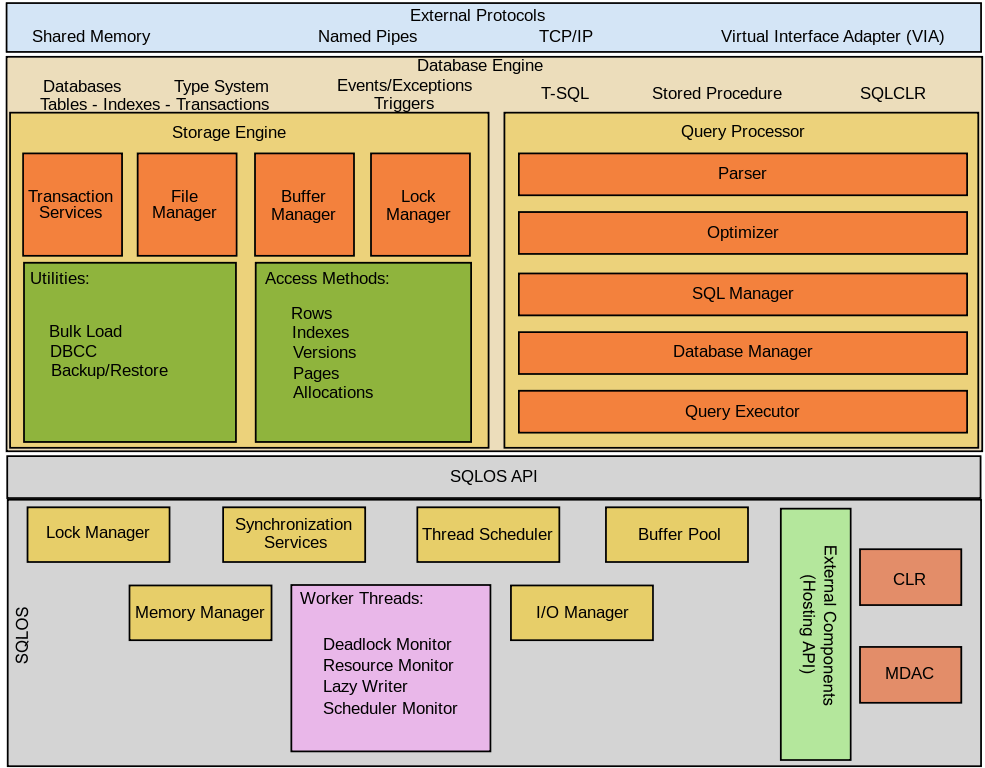
SQL Server is a relational database management system, or RDBMS, developed and marketed by Microsoft.

Similar to other RDBMS software, SQL Server is built on top of SQL, a standard programming language for interacting with the relational databases. SQL server is tied to Transact-SQL, or T-SQL, the Microsoft’s implementation of SQL that adds a set of proprietary programming constructs.

SQL Server works exclusively on Windows environment for more than 20 years. In 2016, Microsoft made it available on Linux. SQL Server 2017 became generally available in October 2016 that ran on both Windows and Linux.

## SQL Server Architecture

The following diagram illustrates the architecture of the SQL Server:



SQL Server consists of two main components:

1. Database Engine
2. SQLOS

2.4.2 **ADO.net**

ADO.NET is a set of classes (a framework) to interact with data sources such as databases and XML files. ADO is the acronym for ActiveX Data Objects. It allows us to connect to underlying data or databases. It has classes and methods to retrieve and manipulate data.

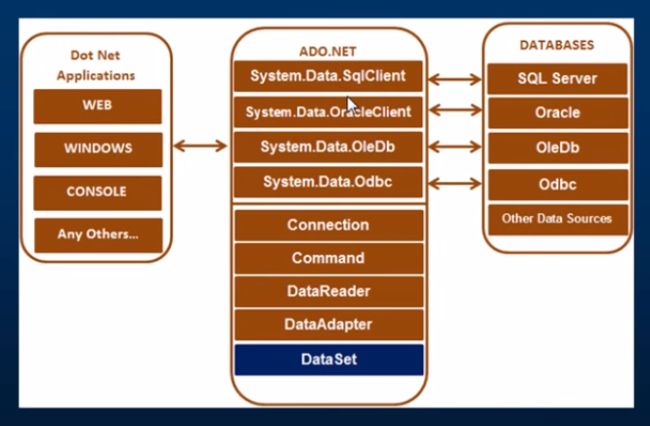
The following are a few of the .NET applications that use ADO.NET to connect to a database, execute commands and retrieve data from the database.

* ASP.NET Web Applications
* Console Applications
* Windows Applications.

## Various Connection Architectures

There are the following two types of connection architectures:

1. **Connected architecture:** the application remains connected with the database throughout the processing.
2. **Disconnected architecture:** the application automatically connects/disconnects during the processing. The application uses temporary data on the application side called a DataSet.

****

In this diagram, we can see that there are various types of applications (Web Application, Console Application, Windows Application and so on) that use ADO.NET to connect to databases (SQL Server, Oracle, OleDb, ODBC, XML files and so on).

## Important Classes in ADO.NET

We can also observe various classes in the preceding diagram. They are:

1. Connection Class
2. Command Class
3. DataReader Class
4. DataAdaptor Class
5. DataSet.Class

### 1. Connection Class

In ADO.NET, we use these connection classes to connect to the database. These connection classes also manage transactions and connection pooling. To learn more about connection classes, start here: [Connection in ADO.NET](https://www.c-sharpcorner.com/article/exploring-connection-in-ado-net/).

### 2. Command Class

The Command class provides methods for storing and executing SQL statements and Stored Procedures. The following are the various commands that are executed by the Command Class.

* **ExecuteReader:** Returns data to the client as rows. This would typically be an SQL select statement or a Stored Procedure that contains one or more select statements. This method returns a DataReader object that can be used to fill a DataTable object or used directly for printing reports and so forth.
* **ExecuteNonQuery:** Executes a command that changes the data in the database, such as an update, delete, or insert statement, or a Stored Procedure that contains one or more of these statements. This method returns an integer that is the number of rows affected by the query.
* **ExecuteScalar:**This method only returns a single value. This kind of query returns a count of rows or a calculated value.
* **ExecuteXMLReader:** (SqlClient classes only) Obtains data from an SQL Server 2000 database using an XML stream. Returns an XML Reader object.

### 3. DataReader Class

The DataReader is used to retrieve data. It is used in conjunction with the Command class to execute an SQL Select statement and then access the returned rows. Learn more here: [Data Reader in C#.](https://www.c-sharpcorner.com/article/datareader-in-C-Sharp/)

### 4. DataAdapter Class

The DataAdapter is used to connect DataSets to databases. The DataAdapter is most useful when using data-bound controls in Windows Forms, but it can also be used to provide an easy way to manage the connection between your application and the underlying database tables, views and Stored Procedures. Learn more here: [Data Adapter in ADO.NET.](https://www.c-sharpcorner.com/article/dataadapter-in-C-Sharp/)

### 5. DataSet Class

The DataSet is the heart of ADO.NET. The DataSet is essentially a collection of DataTable objects. In turn each object contains a collection of DataColumn and DataRow objects. The DataSet also contains a Relations collection that can be used to define relations among Data Table Objects.

### How to Connect to a Database using ADO.NET

Now let us learn how to connect to a database using ADO.NET. To create a connection, you must be familiar with connection strings. A connection string is required as a parameter to SQLConnection. A ConnectionString is a string variable (not case sensitive).

This contains key and value pairs, like provider, server, database, userid and word as in the following:

Server="nameof the server or IP Address of the server"  
  
Database="name of the database"  
  
userid="user name who has permission to work with database"  
  
word="the word of userid"

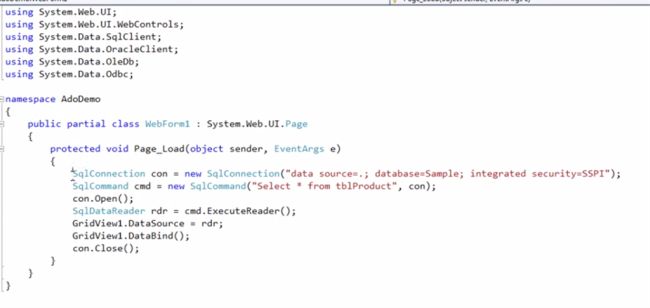
**Example**

**SQL Authentication**String constr="server=.;database=institute;user id=rakesh;word=abc@123";  
  
Or:  
  
String constr="data source=.;initial catalog=institute;uid=rakesh;pwd=abc@213";  
  
**Windows Authentication**  
  
String constr="server=.;database=institute;trusted\_connection=true"  
  
Or:  
  
String constr="server=.;initial catalog=institute;integrated security=true"

**How to retrieve and display data from a database**

1. Create a SqlConnection object using a connection string.
2. Handle exceptions.
3. Open the connection.
4. Create a SQLCommand. To represent a SQLCommand like (select \* from studentdetails) and attach the existing connection to it. Specify the type of SQLCommand (text/storedprocedure).
5. Execute the command (use executereader).
6. Get the Result (use SqlDataReader). This is a forwardonly/readonlydataobject.
7. Close the connection
8. Process the result
9. Display the Result

The following is code for connecting to a SQL Database:



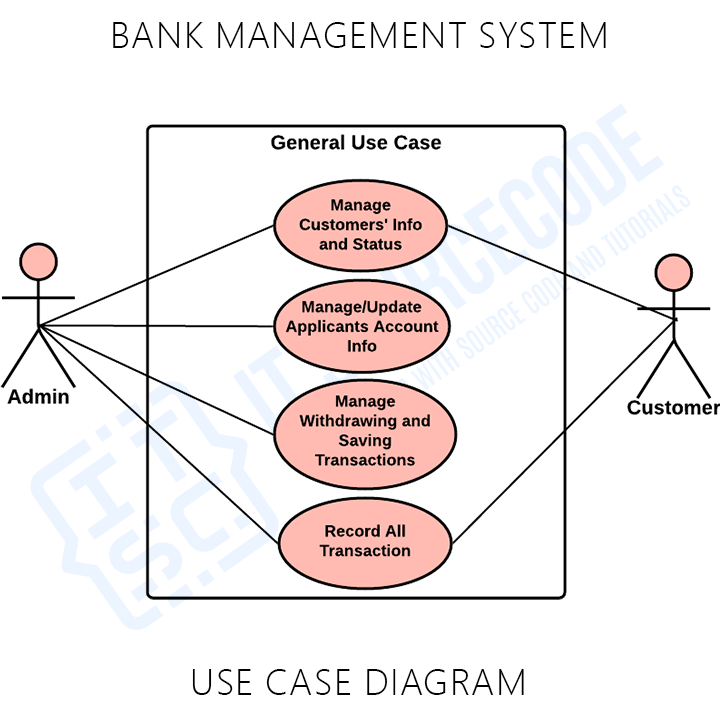
**2.5 Workflow of Banking console application**

**Diagram

Description automatically generated**

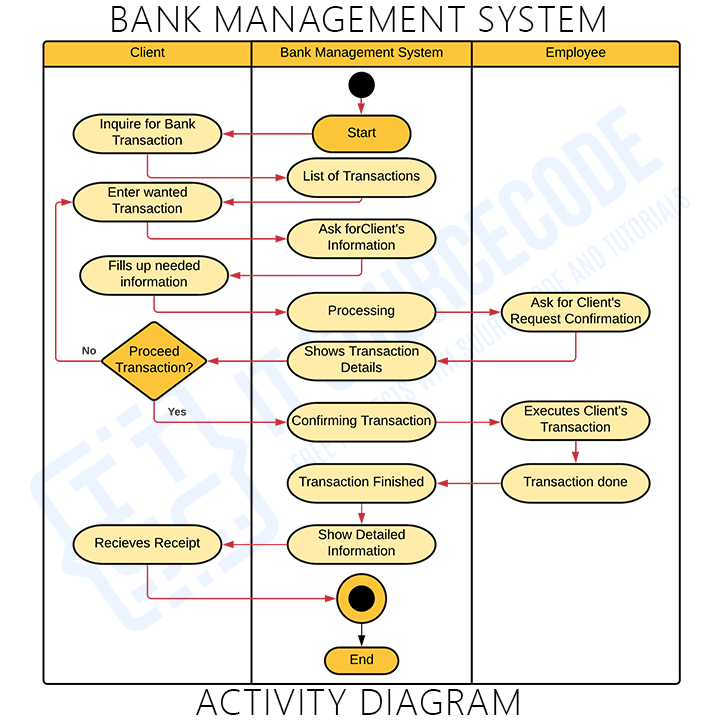
**2.6 Use Case diagram**

This use case diagram is a visual representation of how a user might interact with the bank management system. It depicts the system’s numerous use cases and different sorts of users. The circles or ellipses are used to depict the use cases.



**2.7 Activity diagram**

The bank management system activity diagram is a designed illustration that shows the system’s behavioral aspect. It shows the bank management system’s behavior in terms of responding to its users or clients.



**Chapter 3**

**System Design**

**3.1 Architecture**

.NET Core is a new version of .NET Framework, which is a free, open-source, general-purpose development platform maintained by Microsoft. It is a cross-platform framework that runs on Windows, macOS, and Linux operating systems.

.NET Core Framework can be used to build different types of applications such as mobile, desktop, web, cloud, IoT, machine learning, microservices, game, etc.

.NET Core is written from scratch to make it modular, lightweight, fast, and cross-platform Framework. It includes the core features that are required to run a basic .NET Core app. Other features are provided as NuGet packages, which you can add it in your application as needed. In this way, the .NET Core application speed up the performance, reduce the memory footprint and becomes easy to maintain.

## Why .NET Core?

There are some limitations with the .NET Framework. For example, it only runs on the Windows platform. Also, you need to use different .NET APIs for different Windows devices such as Windows Desktop, Windows Store, Windows Phone, and Web applications. In addition to this, the .NET Framework is a machine-wide framework. Any changes made to it affect all applications taking a dependency on it. Learn more about the motivation behind .NET Core.

Today, it's common to have an application that runs across devices; a backend on the web server, admin front-end on windows desktop, web, and mobile apps for consumers. So, there is a need for a single framework that works everywhere. So, considering this, Microsoft created .NET Core. The main objective of .NET Core is to make .NET Framework open-source, cross-platform compatible that can be used in a wide variety of verticals, from the data center to touch-based devices.

## .NET Core Characteristics

**Open-source Framework:** .NET Core is an [open-source framework](https://dotnet.microsoft.com/platform/open-source) maintained by Microsoft and available on GitHub under [MIT](https://github.com/dotnet/runtime/blob/master/LICENSE.TXT) and [Apache 2](https://www.apache.org/licenses/LICENSE-2.0) licenses. It is a [.NET Foundation project](https://dotnetfoundation.org/).

You can view, download, or contribute to the source code using the following GitHub repositories:

* Language compiler platform Roslyn: <https://github.com/dotnet/roslyn>
* .NET Core runtime: <https://github.com/dotnet/runtime>
* .NET Core SDK repository. <https://github.com/dotnet/sdk>
* ASP.NET Core repository. <https://github.com/dotnet/aspnetcore>

**Cross-platform:** .NET Core runs on Windows, macOS, and Linux operating systems. There are different runtime for each operating system that executes the code and generates the same output.

**Consistent across Architectures:** Execute the code with the same behavior in different instruction set architectures, including x64, x86, and ARM.

**Wide-range of Applications:** Various types of applications can be developed and run on .NET Core platform such as mobile, desktop, web, cloud, IoT, machine learning, microservices, game, etc.

**Supports Multiple Languages:** You can use C#, F#, and Visual Basic programming languages to develop .NET Core applications. You can use your favorite IDE, including Visual Studio 2017/2019, Visual Studio Code, Sublime Text, Vim, etc.

**Modular Architecture:** .NET Core supports modular architecture approach using NuGet packages. There are different NuGet packages for various features that can be added to the .NET Core project as needed. Even the .NET Core library is provided as a NuGet package. The NuGet package for the default .NET Core application model is [Microsoft.NETCore.App](https://www.nuget.org/packages/Microsoft.NETCore.App).

This way, it reduces the memory footprint, speeds up the performance, and easy to maintain.

**CLI Tools:** .NET Core includes [CLI tools](https://www.tutorialsteacher.com/core/net-core-command-line-interface) (Command-line interface) for development and continuous-integration.

**Flexible Deployment:** .NET Core application can be deployed user-wide or system-wide or with [Docker Containers](https://docs.microsoft.com/en-us/dotnet/core/docker/introduction).

**Compatibility:** Compatible with .NET Framework and Mono APIs by using [.NET Standard specification](https://docs.microsoft.com/en-us/dotnet/standard/net-standard).

**3.2 Models and its Properties:**

**A screenshot of a computer

Description automatically generatedAccount Class**

**A screenshot of a computer

Description automatically generatedBank Class**

**A screenshot of a computer

Description automatically generatedCurrency Class**

**A screenshot of a computer

Description automatically generatedStaff Class**

**A screenshot of a computer

Description automatically generated with medium confidenceTransaction Class**

**A screenshot of a computer

Description automatically generated with medium confidenceUser Class**

**SQL Server Database:**

**Text

Description automatically generated with medium confidenceText

Description automatically generated with low confidenceGraphical user interface, text, application

Description automatically generatedGraphical user interface, text

Description automatically generatedGraphical user interface, text, application

Description automatically generated**

So finally there are 5 tables as follows:-

1. Bank

2. Staff

3. Account Holder

4. Currency

5. Transaction

My Banking console application fetch data from these tables and show the result on console.

All the username , password validations are stored in bank, staff, account table,

All the transactions of users stored in Transaction table.

**Chapter 4**

**Software Testing**

**4.1 TESTING**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive. A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software.

Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively applied to both strategic to both large- and small-scale systems.

Every time after creating the whole project in the end of the day we test it manually and removes the bugs that we get. For the testing of functions, we created some unit test cases for directly checking the functionality of functions by passing the values.

The software engineering process can be viewed as a spiral. Initially, system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behaviour, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software, we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progresses by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture.

Taking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are 53 validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested as a whole. Table Given below outlines the tests that were performed on the system to ensure correctness and unearth errors, which were subsequently debugged.

**UNIT TESTING**

Unit Testing will be done to test field validations, navigation, functionality of the programs and its blocks. These tests are applied on various functions within each program and other critical program blocks.

**MODULE TESTING**

Module testing will be done to test the interaction between the various programs within one module. It checks the functionality of each program with relation to other programs within the same module. It then tests the overall functionality of each module.

**INTEGRATION TESTING**

Integration testing is done to test the functionality and interfacing between the modules. The system is built up of various modules, which work together to automate the activities of the hotel management system. These modules should work together in a seamless way to achieve the desired results. Integration testing will test for this property of the modules.

The modules display a cause-and-effect relationship, if data in one module is changed, then it affects the data to change in some other module also. Integration testing needs to check if the modifications do not adversely affect some other modules.

**ACCEPTANCE TESTING**

Acceptance testing was done after the implementation of the system. The acceptance testing will check if the system works correctly in the user environment and if the entire user specified functionalities are present. It also tests if the system adheres to the company policies and quality standard.

**Chapter 5**

**Conclusion**

I have successfully designed, develop and implemented this Banking console application which provides a more secured approach in managing bank customer’s information and strengthens the relationships between banks and their customers by providing the right solutions that uses a multilevel security to improve customer satisfaction. I therefore encourages other developers of similar application to think twice on how best they can improve in developing a more secured system that will meet the challenges we face today especially on the banking sector and other financial institutions.

**References**

1. [**https://www.c-sharpcorner.com/UploadFile/18fc30/understanding-the-basics-of-ado-net/**](https://www.c-sharpcorner.com/UploadFile/18fc30/understanding-the-basics-of-ado-net/)
2. [**https://www.c-sharpcorner.com/technologies/dotnetcore#:~:text=FOLLOW-,.,Articles(416)**](https://www.c-sharpcorner.com/technologies/dotnetcore#:~:text=FOLLOW-,.,Articles(416))
3. [**https://www.guru99.com/c-sharp-access-database.html**](https://www.guru99.com/c-sharp-access-database.html)
4. [**https://en.wikipedia.org/wiki/SQL\_Server\_Express**](https://en.wikipedia.org/wiki/SQL_Server_Express)
5. [**https://www.entityframeworktutorial.net/what-is-entityframework.aspx**](https://docs.microsoft.com/en-us/azure/virtual-machines/)

**Github Link Of this Project:-**

[**https://github.com/anmol26/ATM**](https://github.com/anmol26/ATM)