INTEGRATED CASE STUDY

By

T-[VENKATESHWARI1@cognizant.com](mailto:VENKATESHWARI1@cognizant.com)

BANK ACCOUNT MANAGEMENT & TRANSACTIONS

|  |  |
| --- | --- |
| Introduction |  |
| Project OverView |  |
| Scope |  |
| Traget Audience |  |
| SOFTWARE & HARDWARE REQUIREMENTS |  |
| HARDWARE |  |
| SOFTWARE |  |
| DATA SPECIFICATIONS |  |
| Data Design |  |
| Components required |  |
| Use Cases |  |
| Design Constraints |  |
| SUBMISSION |  |

# INTRODUCTION:

ENJOY CREATING AN ACCOUNT IN YOUR BANK APP.

PERFORM TASK LIKE CREATE NEW ACCOUNT, DEPOSIT,WITHDRAWLS,

BALANCE ENQUIRY ETC.

# PROJECT OVERVIEW:

This project captures the various concepts, techniques and skills learned and help to put them into practice using ASP.NET MVC . This would be at a scaled-down level since the purpose is to let the associate experience the concepts DATA BASE TABLES,ASP.NET MVC FORMS learned in ASP.NET MVC as an individual. The individual associate is expected to carry out the knock out challenge and complete it within 3 hours.

# SCOPE:

🡪ASP.NET MVC

🡪MS SQL SERVER

🡪HTML,CSS,JAVASCRIPT

# TARGET AUDIENCE

🡪Beginner Level

# Hardware and Software Requirements

### Hardware Requirements

|  |  |  |
| --- | --- | --- |
| # | Item | Specification/Version |
|  | PROCESSOR | I5 |
|  | RAM | 8GB |
|  | OS | WINDOWS 10 |
|  |  |  |

### Software Requirements

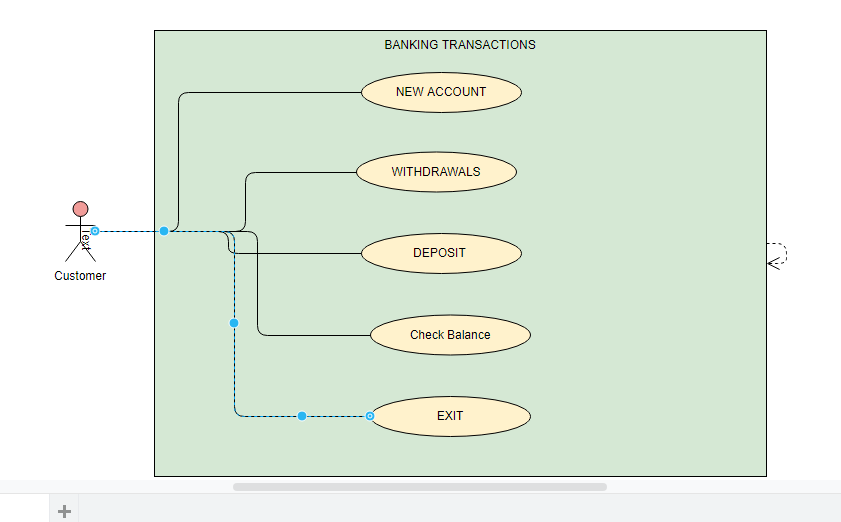
|  |  |  |  |
| --- | --- | --- | --- |
| # | Item | Specification/Version | |
| 1. | Visual Studio2017 | Enterprise Edition | |
|  | | |
|  |  |  | |

# Functional Requirements

## Functional Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. # | Req. Name | Req. Description | Actors / Users | Comments |
| 1 | ADD ACCOUNT | ADD ACCOUNT, will automatically adds minimum amount of 500/rs | Users |  |
| 2 | DEPOSIT | Will take account number to deposit specified amount, to corresponding account. |  | Will add amount to existing amount |
| 3 | WITHDRAWLS | Will take account number to withdraw from specified ac no |  | Will substract from existing amount of the specified ac.  Withdrawl amont should not be more than existing amount. |
| 4 | BALANCE ENQUIRY | Takes account number and displays available balance in ac. |  | -- |
| 5 |  | --------- |  |  |
|  |  |  |  |  |

## USE CASE:



## Data Design

## TABLE STRUCTURE

|  |  |
| --- | --- |
| Table name: BANK\_USER | |
| Column Name | **Data type** |
| ID | Int (PK) IDENTITY |
| NAME | varchar(25) |
| Address | varchar(25) |
|  |  |
| AccountType | Varchar(25) |
| BalanceAmount | Money |

USE [BANK\_APP]

GO

CREATE TABLE [dbo].[BANK\_USER](

[AccountID] [int] IDENTITY(1,1) NOT NULL,

[Name] [char](25) NOT NULL,

[Address] [varchar](50) NOT NULL,

[AccountType] [char](10) NOT NULL,

[BalanceAmount] [money] NOT NULL,

CONSTRAINT [PK\_BANK\_USER] PRIMARY KEY CLUSTERED

(

[AccountID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

## Design Constraints:

* Use MSSQL SERVER database to store the data. The database name should be “BANK\_APP”.
* The above table has been already created. To create the table in your local machine, you can avail the script from “script.sql, which will be provided as part of the code skeleton.
* The table name and the column names should be the same as specified in the table structure.
* Database connections should be configurable; it should not be hard coded.

## MVC Components Design for identified Use cases

|  |  |  |
| --- | --- | --- |
| Component Name(Model) | Method  (OR)  Property | Desc |
| BANKCLASS | Accountno, Accountholdername, Accountholderaddress,  Accounttype, Balanceamount,  public long balanceamountdump = 500; |  |
|  |  |  |
| INTERFACE ALLMETHODS | --string mainmenu();  --void addnewaccount();  -- void balanceenquiry(allmethodsdetails allmethoddetailsbalanceobj);  --void withdrawamount(allmethodsdetails allmethoddetailswithdrawobj);  --void depositamount(allmethodsdetails allmethoddetailsdepositobj);  --void exit(); |  |
| class allmethodsdetails : allmethods | This class implements methods of allmethods interface |  |
|  |  |  |

## General Design Constraints

* 1. The attribute/method/class name should be correctly specified as given in the document.