SEDCO

Presented to the SEDCO

Prepared by : Mohammad al-mesherqe

CVM department

2/18/2019

Abstraction:

KIOSK screen configurator application, is an application responsible to configure Kiosk screen

Through add, edit, delete buttons and manages the activity for each button.

 KIOSK screen configurator application have intuitive user interface to support the usability of our application.

Table of Contents

Chapter 1: user manual

Task name: open KIOSK screen configurator.

Activity:

1- Install the application on end user machine(By administrator "chapter2 for detailed description "
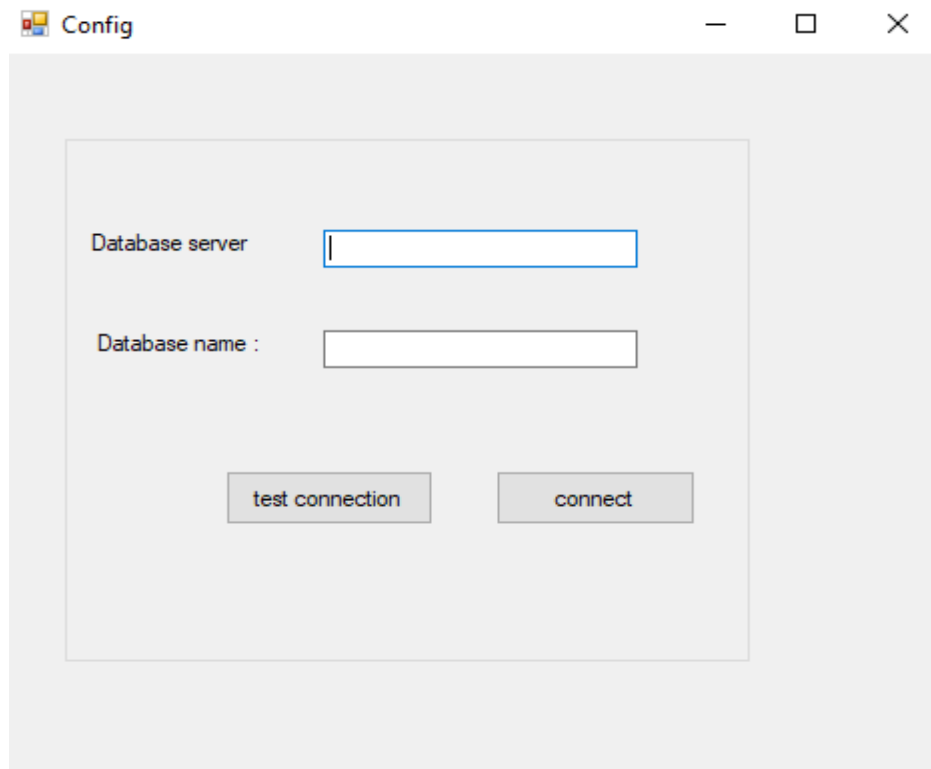2- Open path where the application installed and open it .

Post condition:

The application is now opened and you can try its feature.

Screen description:

-pre condition: open the application.

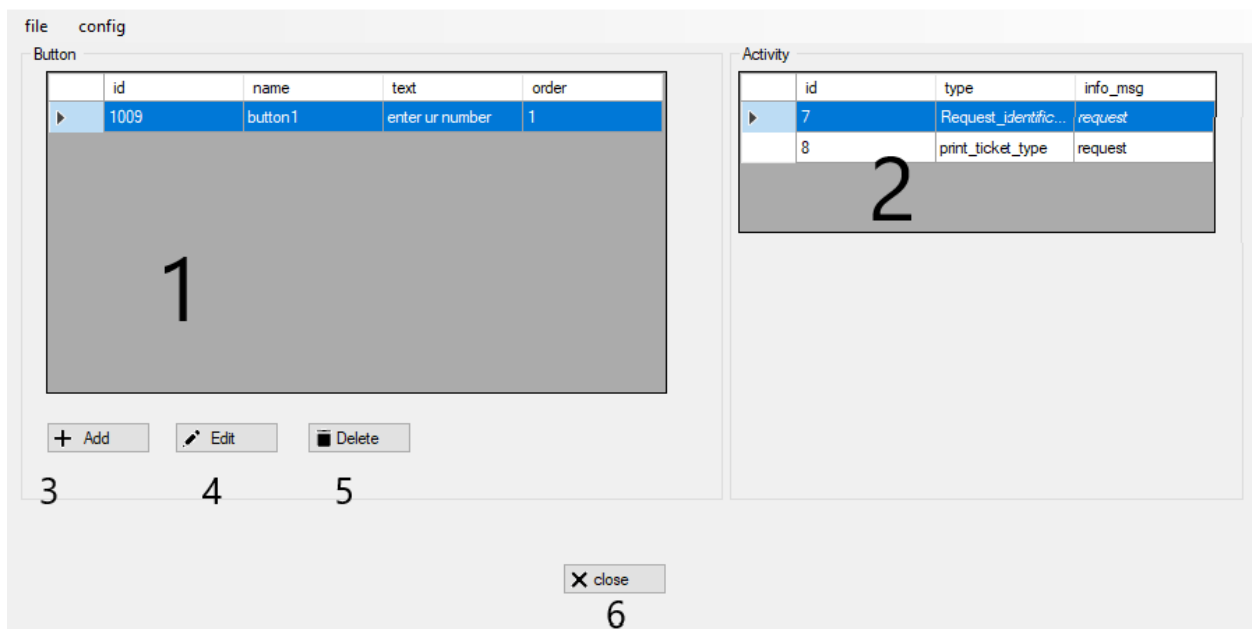If that the first time of you in our application this screen will be appear:

This image bellow for main form components

This screen for administrator to config your connection with database.

1- Button list: list preview the button list.
2- Activity list: list preview the activity list for each button.
3- Add button: open new window to add new button.
4- Edit Button: to edit selected button in the list.
5- Delete Button: to delete selected button in the list.
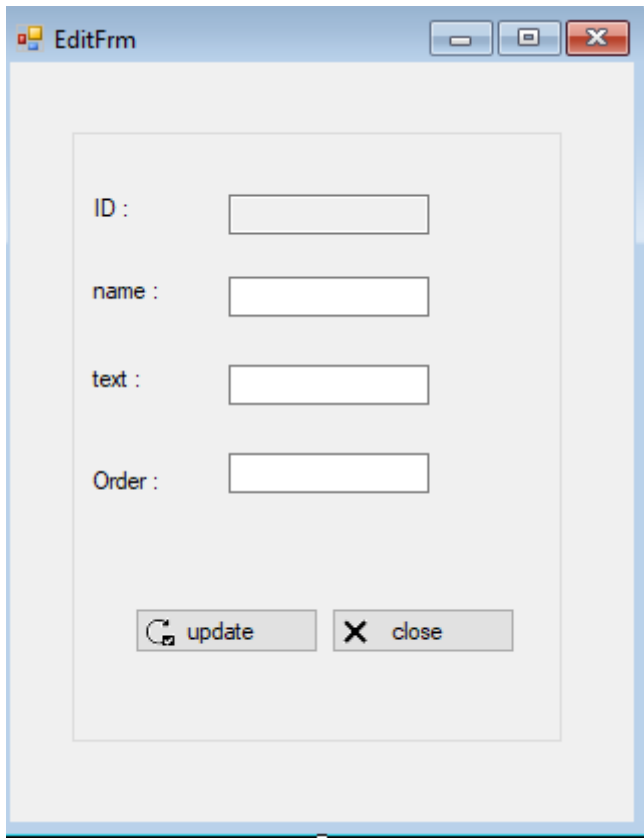6- Close Button: to close the application.

Add Button Form :

*note:

You must enter button first then you can add activity.

Edit Form:
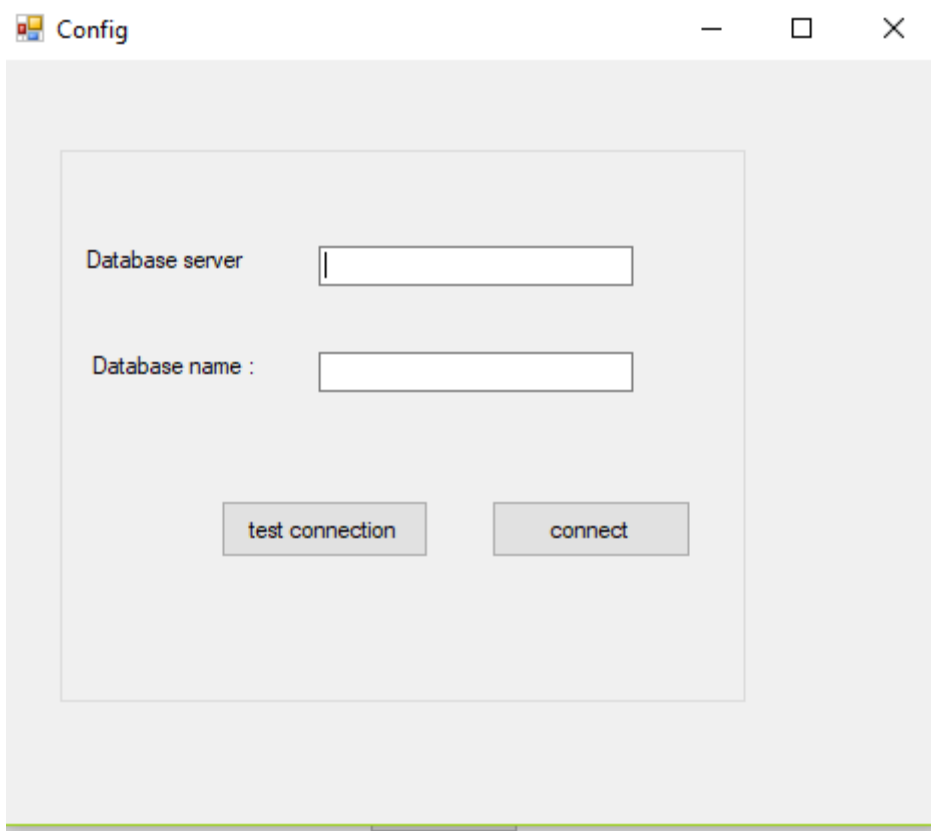
Edit button information then click update to save change in database.

Chapter 2: Administrator guide

With project file you have folder database which is contains the database of our application attach it to sql server in user machine.

We are designed a interface to enter the connection string of user machine as follow:

- Database server : the name of database server
- Database name : the name of database
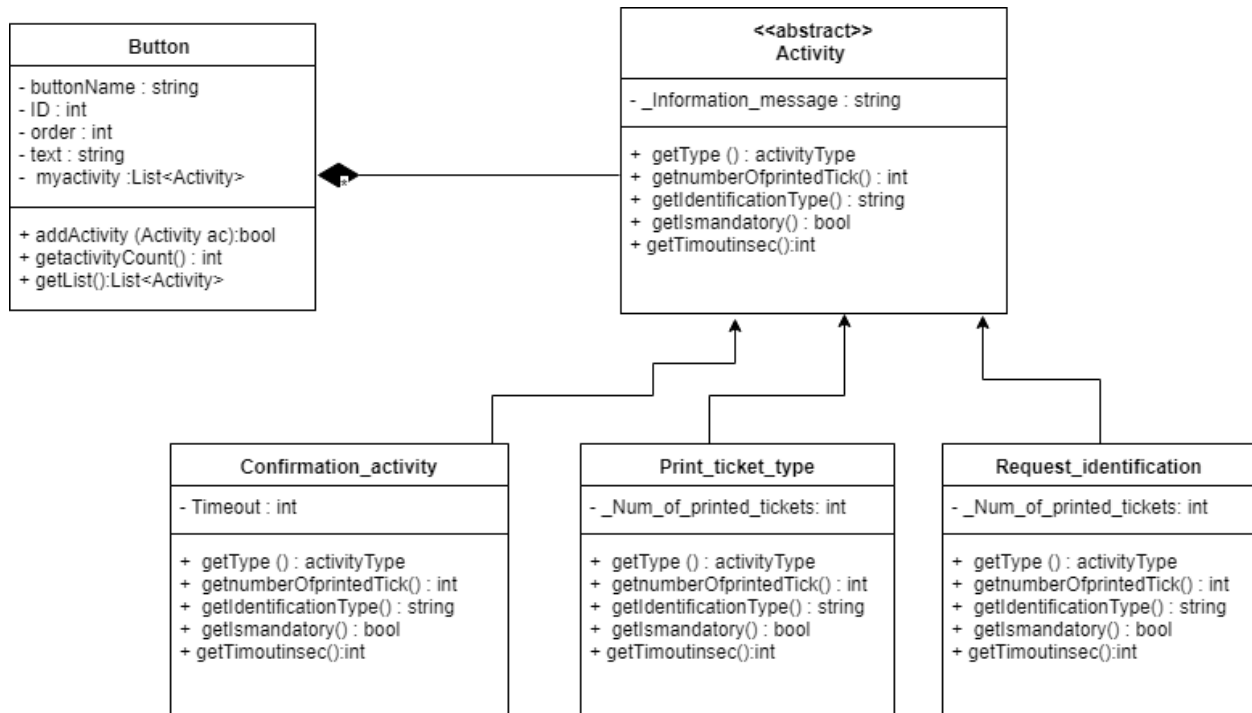
Then click connect.

If you have this error

Couldn't process file presentation layer\mainForm.resx due to its being in the Internet or Restricted zone or having the mark of the web on the file. Remove the mark of the web if you want to process these files. Watch this video to solve it
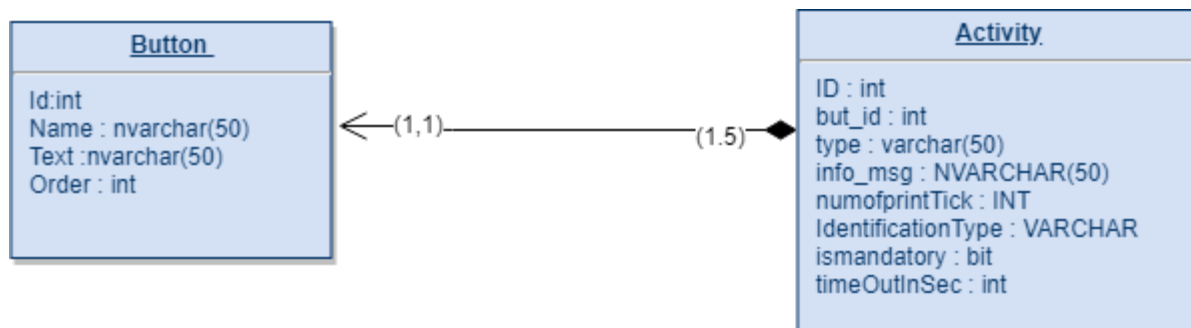
https://www.youtube.com/watch?v=0UKlpN6WhQY

Chapter 3 Developer guide

UML diagrams:

   1- Class diagram :

Button

- buttonName : string
- ID : int
- order : int
- text : string
- myactivity :List<Activity>

+ addActivity (Activity ac):bool
+ getactivityCount() : int
+ getList():List<Activity>

<>
Activity

- _Information_message : string

+ getType () : activityType
+ getnumberOfprintedTick() : int
+ getIdentificationType() : string
+ getIsmandatory() : bool
+ getTimoutinsec():int

Confirmation_activity

- Timeout : int

+ getType () : activityType
+ getnumberOfprintedTick() : int
+ getIdentificationType() : string
+ getIsmandatory() : bool
+ getTimoutinsec():int

Print_ticket_type

- _Num_of_printed_tickets: int

+ getType () : activityType
+ getnumberOfprintedTick() : int
+ getIdentificationType() : string
+ getIsmandatory() : bool
+ getTimoutinsec():int

Request_identification

- _Num_of_printed_tickets: int

+ getType () : activityType
+ getnumberOfprintedTick() : int
+ getIdentificationType() : string
+ getIsmandatory() : bool
+ getTimoutinsec():int

   2- ER-diagram

Button

Id:int
Name : nvarchar(50)
Text :nvarchar(50)
Order : int

—(1,1)———————(1.5)—

Activity

ID : int
but_id : int
type : varchar(50)
info_msg : NVARCHAR(50)
numofprintTick : INT
IdentificationType : VARCHAR
ismandatory : bit
timeOutInSec : int

Design and architecture:

1- Code support quality attribute like (usability, modifiability)
2- I am using singleton design pattern to create data Access layer object
3- I am applying solid design principle and make the code cane contain new type from activity without change through add new type and don't touch existing code("polymorphism").
4- Single responsibility principle applying

Notes :

- Source code is separated and organizing using regions and comments
- Variable naming conventions ass following :
    1- Multiple-word identifiers.
    2- Delimiter-separated words
    3- CamelCase.

The end