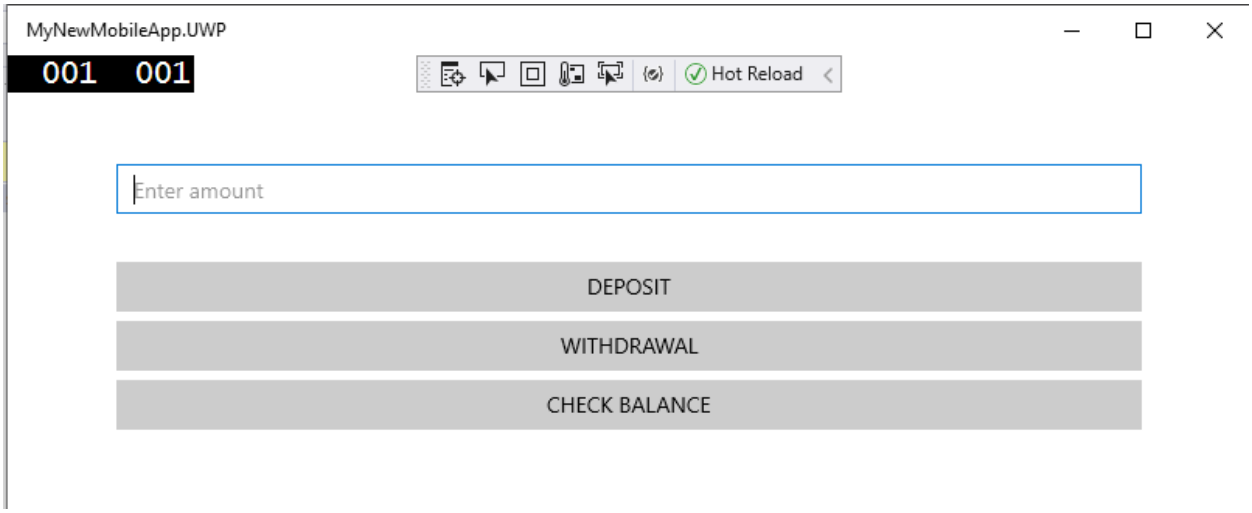
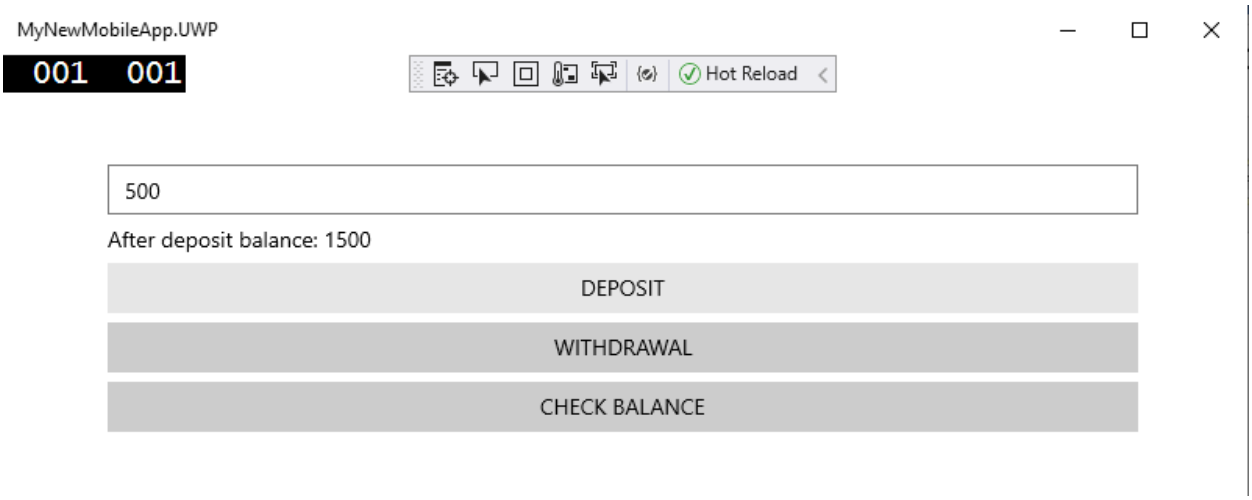
Q: Create one application for banking transaction:

- Withdrawal operation and
- Deposit operation





Solution:

Design Page: Page11.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="MyNewMobileApp.Page11" Padding="70">
    <ContentPage.Content>
        <StackLayout>

<Entry Placeholder="Enter amount" x:Name="txtAmount" />

<Label Text="" x:Name="lblResult"  />

<Button Text="DEPOSIT" Clicked="btnDeposit" />
```

```
<Button Text="WITHDRAWAL" Clicked="btnWithdrawal" />
<Button Text="CHECK BALANCE" Clicked="btnCheckBalance" />
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

Code behind page: Page11.xaml.cs

```
using System;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace MyNewMobileApp
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class Page11 : ContentPage
    {
        public Page11()
        {
            InitializeComponent();
        }
        Bank b1 = new Bank();
        double balance;
        public void btnDeposit(object sender,EventArgs args)
        {
            double amt =double.Parse(txtAmount.Text);
            balance=b1.deposit(amt);
            lblResult.Text = "After deposit balance: " + balance.ToString();

        }
        public void btnWithdrawal(object sender, EventArgs args)
        {
            double amt = double.Parse(txtAmount.Text);
            balance= b1.withdrawal(amt);
            lblResult.Text ="After withdrawal balance: "+ balance.ToString();
        }
        public void btnCheckBalance(object sender, EventArgs args)
        {
            balance= b1.checkbalance();
            lblResult.Text ="Balance:"+ balance.ToString();
        }
    }
}
```

Class File: Bank.cs

```
using System;

namespace MyNewMobileApp
{
    class Bank
    {
        static double balanceAmount=1000;
        public double checkbalance()
        {
```

```
         return balanceAmount;
        }
        public double deposit(double depositamount)
        {
            balanceAmount = balanceAmount + depositamount;
            return balanceAmount;
        }
        public double withdrawal(double withdrawalamount)
        {
            if(withdrawalamount<balanceAmount)
            {
                balanceAmount = balanceAmount - withdrawalamount;
            }

            return balanceAmount;
        }
    }
}
```