By
Maminiaina, Christelle, Amanda

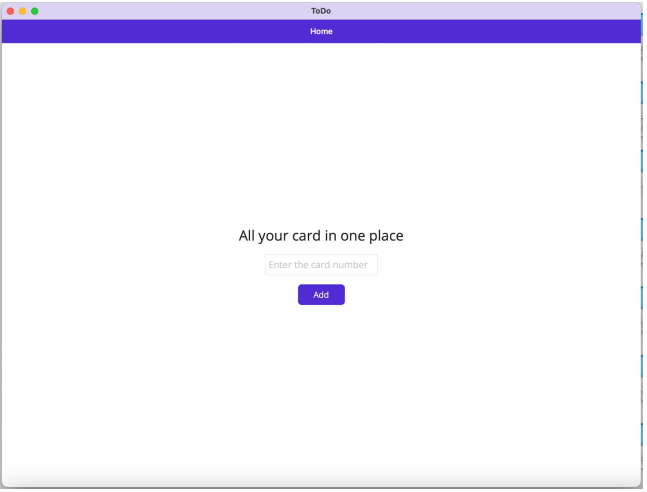
Card Validator Application Using .NET MAUI

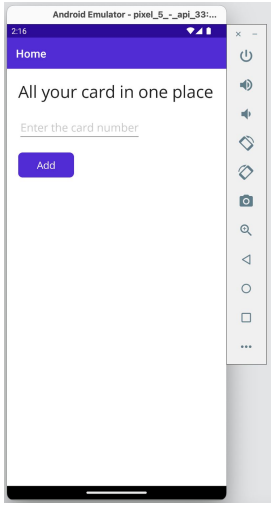
An application that checks for the validity of a credit card number and identifies its type (Visa, MasterCard, Amex or Unknown) using the Luhn algorithm.

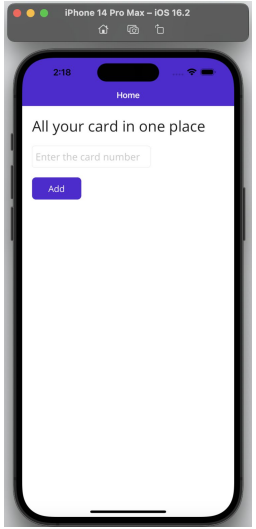
Introduction

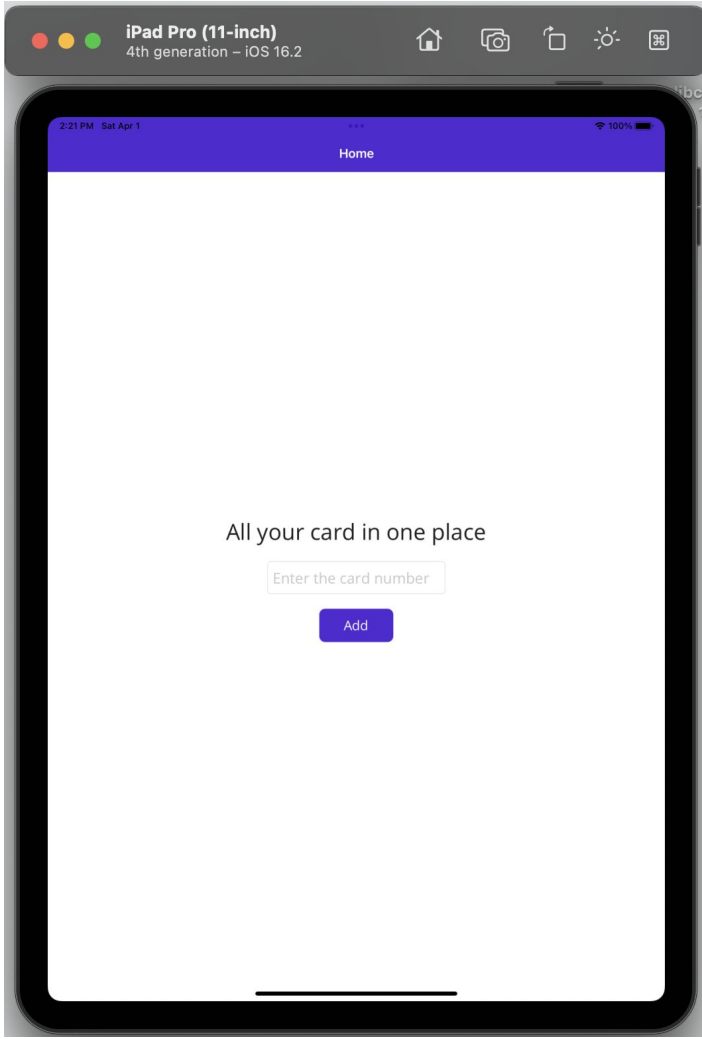
- The Card Validator is a .NET MAUI application that checks the validity of a credit card number and identifies its type, then store it if it is a valid number.

- The app can run on IOS, Android, MacOs, Windows





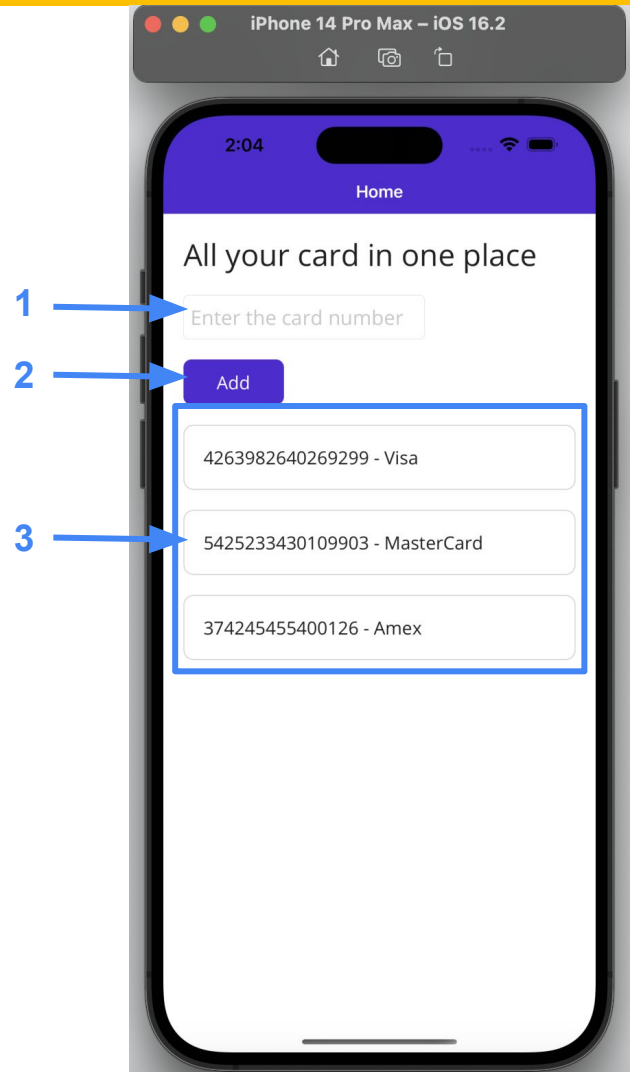




User Interface

The application has a simple user interface

- 1) a single text box to input the credit card number
- 2) a button to submit the number for validation.
- 3) When the validation is successful, a label is added to the stack layout showing the credit card number and its type.



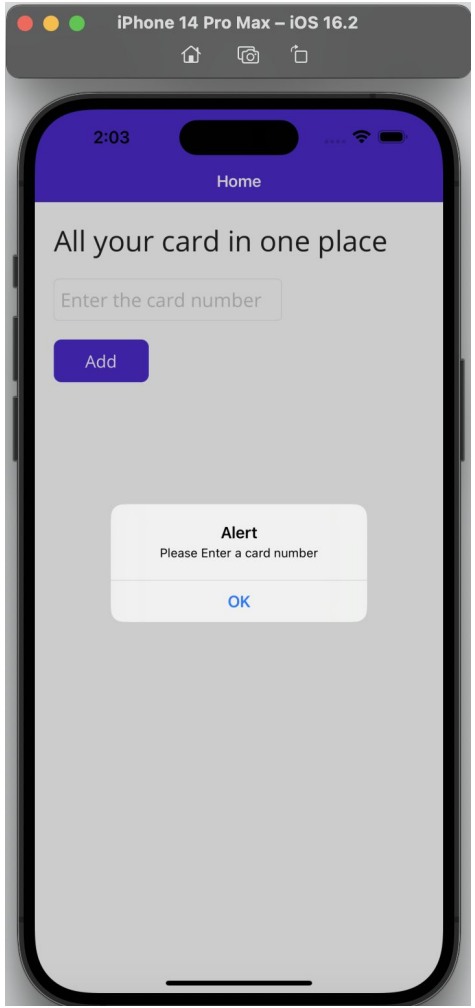
Usage

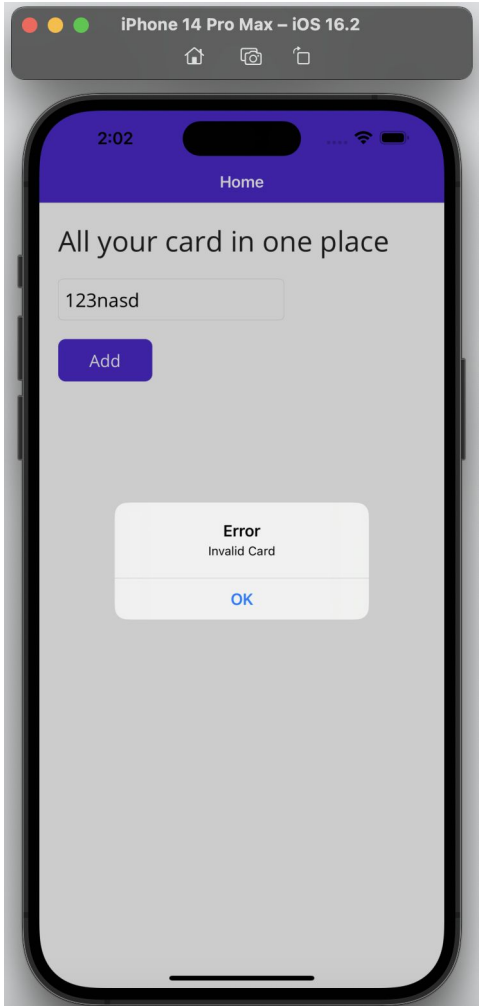
- To use the application, run it on your local machine or emulator

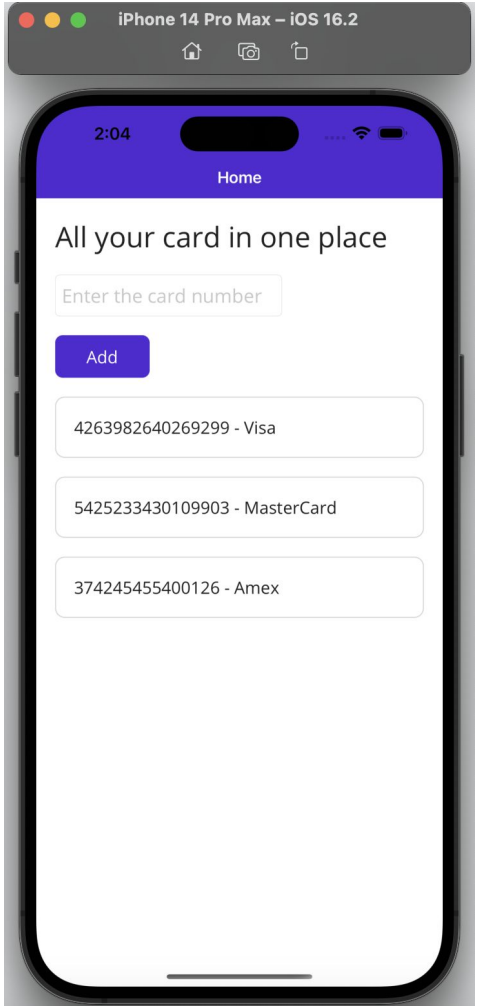
- Input a credit card number into the text box provided.

- The application will add a label showing the card number and its type (Visa, MasterCard, Amex or Unknown) to the stack layout if the card number is valid.

- An error message will be displayed if the card number is invalid.


iPhone 14 Pro Max – iOS 16.2
2:03
Home
All your card in one place
Enter the card number
Add
Alert
Please Enter a card number
OK


iPhone 14 Pro Max – iOS 16.2
2:02
Home
All your card in one place
123nasd
Add
Error
Invalid Card
OK


iPhone 14 Pro Max – iOS 16.2
2:04
Home
All your card in one place
Enter the card number
Add
4263982640269299 - Visa
5425233430109903 - MasterCard
374245455400126 - Amex

The C# Code

Let’s dig into the code

```
1   using Microsoft.Maui.Controls;
2
3   namespace ToDo
4   {
5
6       public partial class MainPage : ContentPage
7       {
8           public MainPage()
9           {
10              InitializeComponent();
11          }
12
13          // Check for valid card
14  1 ->    static bool IsValid(string cardNumber)
15          {
16              int sum = 0;
17              bool isSecondDigit = false;
18              for (int i = cardNumber.Length - 1; i >= 0; i--)
19              {
20                  if (cardNumber[i] < '0' || cardNumber[i] > '9')
21                  {
22                      // Invalid character, return false
23                      return false;
24                  }
25                  int digit = cardNumber[i] - '0';
26                  if (isSecondDigit)
27                  {
28                      digit *= 2;
29                      if (digit > 9)
30                      {
31                          digit -= 9;
32                      }
33                  }
34                  sum += digit;
35                  isSecondDigit = !isSecondDigit;
36              }
37              return sum % 10 == 0;
38          }
39
40          // Get the card type
41          static string GetCardType(string cardNumber)...
60
61          // Our button onclick
62          private async void OnSubmitButtonClicked(object sender, EventArgs e)...
101     }
102 }
103
```

```
1   using Microsoft.Maui.Controls;
2
3   namespace ToDo
4   {
5
6       public partial class MainPage : ContentPage
7       {
8           public MainPage()
9           {
10              InitializeComponent();
11          }
12
13          // Check for valid card
14          static bool IsValid(string cardNumber)...
39
40          // Get the card type
41  2 ->    static string GetCardType(string cardNumber)
42          {
43              if (cardNumber.Length == 15 && (cardNumber.StartsWith("34") || cardNumber.StartsWith("37")))
44              {
45                  return "Amex";
46              }
47              else if (cardNumber.Length == 16 && (cardNumber.StartsWith("51") || cardNumber.StartsWith("52") || cardNumber.StartsWith("53") |
48              {
49                  return "MasterCard";
50              }
51              else if ((cardNumber.Length == 13 || cardNumber.Length == 16) && cardNumber.StartsWith("4"))
52              {
53                  return "Visa";
54              }
55              else
56              {
57                  return "Unknown";
58              }
59          }
60
61          // Our button onclick
62          private async void OnSubmitButtonClicked(object sender, EventArgs e)...
101     }
102 }
103
```

```
{
    InitializeComponent();
}

// Check for valid card
static bool IsValid(string cardNumber)...

// Get the card type
static string GetCardType(string cardNumber)...

// Our button onclick
3 -> private async void OnSubmitButtonClicked(object sender, EventArgs e)
{

    string cardNumber = entryCardNumber.Text;

    if (cardNumber.Length == 0)
    {
        await DisplayAlert("Alert", "Please Enter a card number", "OK");
        return;
    }

    if (IsValid(cardNumber))
    {
        // await DisplayAlert("Card Type", $"{GetCardType(cardNumber)}", "OK");

        // Create a new label with the value of the entry
        Label newLabel = new Label
        {
            Text = cardNumber + " - " + GetCardType(cardNumber),
            FontSize = 18,
        };

        Frame frame = new Frame
        {
            Content = newLabel,
            BorderColor = Colors.LightGray,
            CornerRadius = 10,
        };


        // Add the new label to the stack layout
        stackLayout.Children.Add(frame);
    }

    else
        await DisplayAlert("Error", "Invalid Card", "OK");

    entryCardNumber.Text = "";
}
```

How does the code works

- The application uses the Luhn algorithm to check the validity of the entered credit card number.

- It identifies the type of the credit card based on the first few digits of the card number.

- A new label is created with the card number and its type, and added to the stack layout if the card number is valid.

Issues Encountered

- Compatibility issues with macOS version, Xcode and .NET MAUI installation errors were encountered while working on a macOS desktop.

- Few issues related to dependencies or libraries were also encountered.

- They were resolved by updating the software, reinstalling or repairing the installation, and troubleshooting dependencies and libraries.

Future improvements

- Adding support for additional credit card types
- improving the user interface
- adding support for international credit card numbers.
- expiration date validation, CVV validation, and more.

Photo by Pixabay

Conclusion

Microsoft .NET MAUI is a powerful cross-platform framework that allows developers to create high-quality applications for multiple devices and platforms with ease.

Ressources

Microsoft .NET MAUI website is an excellent resource for developers looking to learn more about this framework.

Thank you for your time 😊