

Enalyzer ATM assignment

Beskrivelse af opgaven og den måde jeg har løst den på.

Til Enalyzer opgaven er der brugt C# og Bootstrap til design opbygning, og består af en master page og en default side. Masterpagen indeholder overskrift og selve side designet. Jeg har taget mig lidt frihed i forhold til at overskriften går igen på alle side, samt der er et beskedfelt til evt. fejlbeskeder. Der er desuden brugt diverse CSS og javascript til brug for f.eks runde knapper og bootstrap

Så er der lavet en default side, som indeholder et multiview med 3 views.

View 1:

Denne side, indeholder kun en knap, som starter simuleringen af processen med at hæve penge.

View 2:

Denne side, indeholder følgende kontroller ud over tekst som er ifl. Design oplægget.

En literal til visning af det indtastede beløb.

4 rækker af 3 knapper med tal og tilbageknap, der skal simulere et tastatur på en hæveautomat.

Derud over er der en Submit knap, som laver udregningen, og sender brugeren videre til View 3

Der er lavet fejlhåndtering, så brugeren ikke kan indtaste nul. Det er desuden forholdsvis let at tilføje flere sedler/mønter. De skal bare indsættes i den rigtige rækkefølge i det array der er brugt. Man kunne selvfølgelig lave en sortering, så man altid får den rigtige rækkefølge med de største beløb først.

View 3

I dette view, vises det antal sedler og mønter som brugeren vil modtage, samt felter til beskeder.

Der er desuden en tilbageknap, som sender brugeren tilbage til forrige side.

Guide

På side 1 trykkes på knappen "Start withdrawal", hvorefter siden med valg af beløb. På denne side, vælges et beløb ved at klikke på de respektive knapper med tal. Ønskes et andet beløb trykkes på knappen med pilen tilbage. Når det ønskede beløb er indtastet, trykkes på knappen "Submit" og næste side vises med det antal sedler og mønter der skal bruges for udbetalingen af det korrekte beløb.

Ønskes en ny udregning, trykkes på knappen med pilen tilbage i toppen af siden, hvorefter man kan indtaste et nyt beløb.

Test:

Der er testet med følgende beløb, vist i nedenstående matrix

Indtastet	578	2500	1945	8436	42	651
1000	0	2	1	8	0	0
500	1	1	1	0	0	1
200	0	0	2	2	0	0
100	0	0	0	0	0	1
50	1	0	0	0	0	1
20	1	0	2	1	2	0
10	0	0	0	1	0	0
5	1	0	1	1	0	0
2	1	0	0	0	1	0
1	1	0	0	1	0	1
Resultat	578	2500	1945	8436	42	651

Manglende funktionalitet eller ændringer.

Der er ikke lavet hvide kanter på knapperne, og der er ikke lavet visning af sedler og mønter som ifl. Design oplægget.

Mulige udvidelser/tilføjelser

Det ville selvfølgelig være ideelt at bruge en database til at holde de sedler og mønter der skal bruges.

Jeg ville også bruge et objekt af sedler/mønter, som man kunne arbejde med. Og så ville jeg lægge alt kode, som ikke har med f.eks onclick at gøre, ned i class filer.

Enalyzer ATM Assignment

Description of the task and how I have solved it.

For the Enalyzer task, C# and Bootstrap were used for the design setup, and it consists of a master page and a default page. The master page contains the header and the overall page design. I have taken some liberty in having the header appear on all pages, as well as a message field for potential error messages. Additionally, various CSS and JavaScript have been used for features like round buttons and Bootstrap.

A default page has been created, which contains a multiview with 3 views.

View 1:

This page only contains a button to start the simulation of the withdrawal process.

View 2:

This page contains the following controls in addition to text according to the design layout.

A literal for displaying the entered amount.

4 rows of 3 buttons with numbers and a back button to simulate an ATM keyboard.

There is also a Submit button that performs the calculation and takes the user to View 3.

Error handling has been implemented so the user cannot enter zero. It is also relatively easy to add more bills/coins. They just need to be inserted in the correct order in the array used. One could, of course, arrange them so that the correct order with the largest amounts first is always obtained.

View 3

In this view, the number of bills and coins the user will receive is shown, as well as fields for messages.

There is also a back button that takes the user to the previous page.

Guide

On page 1, press the "Start withdrawal" button, then the page for selecting the amount appears. On this page, an amount is selected by clicking on the respective buttons with numbers. If a different amount is desired, press the back arrow button. When the desired amount is entered, press the "Submit" button, and the next page shows the number of bills and coins required for the correct amount withdrawal.

If a new calculation is desired, press the back arrow button at the top of the page, and you can enter a new amount.

Test:

It has been tested with the following amounts, shown in the matrix below:

Entered	578	2500	1945	8436	42	651
1000	0	2	1	8	0	0
500	1	1	1	0	0	1
200	0	0	2	2	0	0
100	0	0	0	0	0	1
50	1	0	0	0	0	1
20	1	0	2	1	2	0
10	0	0	0	1	0	0
5	1	0	1	1	0	0
2	1	0	0	0	1	0
1	1	0	0	1	0	1
Result	578	2500	1945	8436	42	651

Missing features or changes.

White borders on the buttons have not been made, and displays of bills and coins have not been made according to the design layout.

Possible extensions/additions

It would, of course, be ideal to use a database to manage the bills and coins needed.

I would also use an object of bills/coins to work with. Then I would put all code that is not related to, for example, onclick into class files.