

**Design-ul intr-o aplicatie Web. User Experience (UX). Alegerea culorilor potrivite. ASP.NET MVC Best Practices.**

## **Design-ul intr-o aplicatie Web**

Pentru realizarea aplicatiilor web, in ceea ce priveste design-ul acestora, se pot respecta anumite reguli:

- **Simplitatea** – prea multa informatie intr-o pagina distrage atentia utilizatorului deoarece acesta trebuie sa citeasca si sa parcurga mult prea multa informatie pana la informatia de care are nevoie. Pastrand paginile simple, aplicatia va fi mult mai usor de folosit.
- **Design-ul este esential** – prima impresie conteaza. Aceasta regula se aplica si in dezvoltarea unei aplicatii web deoarece este primul lucru pe care il observa un utilizator atunci cand acceseaza aplicatia.
- **Culorile sunt foarte importante** – atunci cand alegeti culorile folositi o paleta consistenta de culori. De asemenea, sa va asigurati ca nu exista nuante foarte apropiate care nu pot fi deosebite si mai ales ca exista un contrast puternic intre text si fundal.
- **Navigarea ar trebui sa fie intuitiva** – un utilizator nu trebuie sa caute ceea ce doreste sa acceseze. Paginile trebuie sa fie bine organizate cu un design de tipul top-down, utilizatorii navigand usor printre diferitele sectiuni existente intr-o pagina.
- **Consistenta este extrem de importanta** – utilizatorii nu ar trebui sa aiba sentimentul ca viziteaza un alt site sau o alta aplicatie web de fiecare data cand acceseaza o alta pagina a aplicatiei. Consistenta face navigarea in aplicatie mult mai simpla.

- **Aplicatia trebuie sa fie responsive** – utilizatorii acceseaza aplicatia utilizand o varietate de device-uri, de la smartphone-uri la calculatoare personale, de aceea este esential ca aplicatia sa se incadreze oricarei rezolutii. CSS media queries sunt ideale pentru realizarea rapida a unei aplicatii web responsive.
- **Utilizarea unui continut real in momentul dezvoltarii design-ului** – evitati Lorem Ipsum. Orice aplicatie este bazata pe continut si se dezvolta in jurul continutului. De cele mai multe ori atunci cand se dezvolta design-ul nu se acorda importanta continutului, folosindu-se Lorem Ipsum in locul textului real si placeholder in locul imaginilor. Chiar daca totul arata bine in timpul dezvoltarii design-ului, nu o sa mai fie la fel atunci cand aplicatia va contine datele reale. Scopul in dezvoltarea unei aplicatii web este de a fi cat mai aproape de experienta reala a utilizatorului
- **Fontul** – in general fonturile Sans Serif, cum ar fi Arial si Verdana, sunt mai usor de citit intr-o aplicatie web, fiind fonturi fara finisaje decorative. Dimensiunea fontului pentru citirea cu usurinta este de 16px

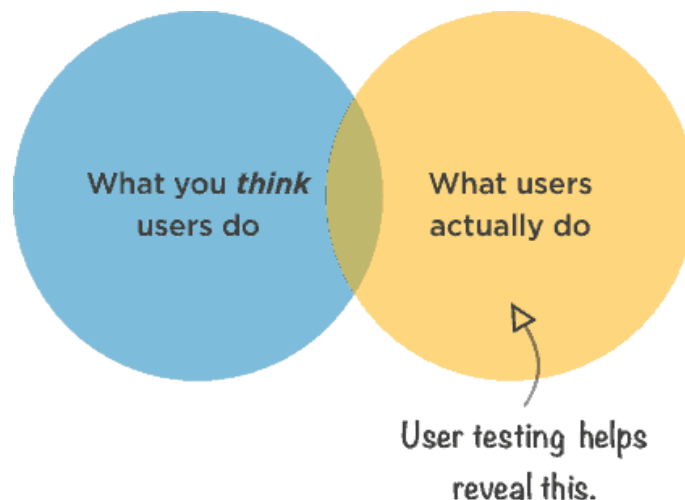


- **Imaginile** – ajuta la o buna interactiune cu utilizatorul
- **“F” Pattern Design** – informatiile sa fie prezentate, in ordinea relevantei, de la stanga la dreapta si de sus in jos, un studiu aratand ca partea de sus-stanga a ecranului este mult mai vizualizata decat cea de jos-dreapta.

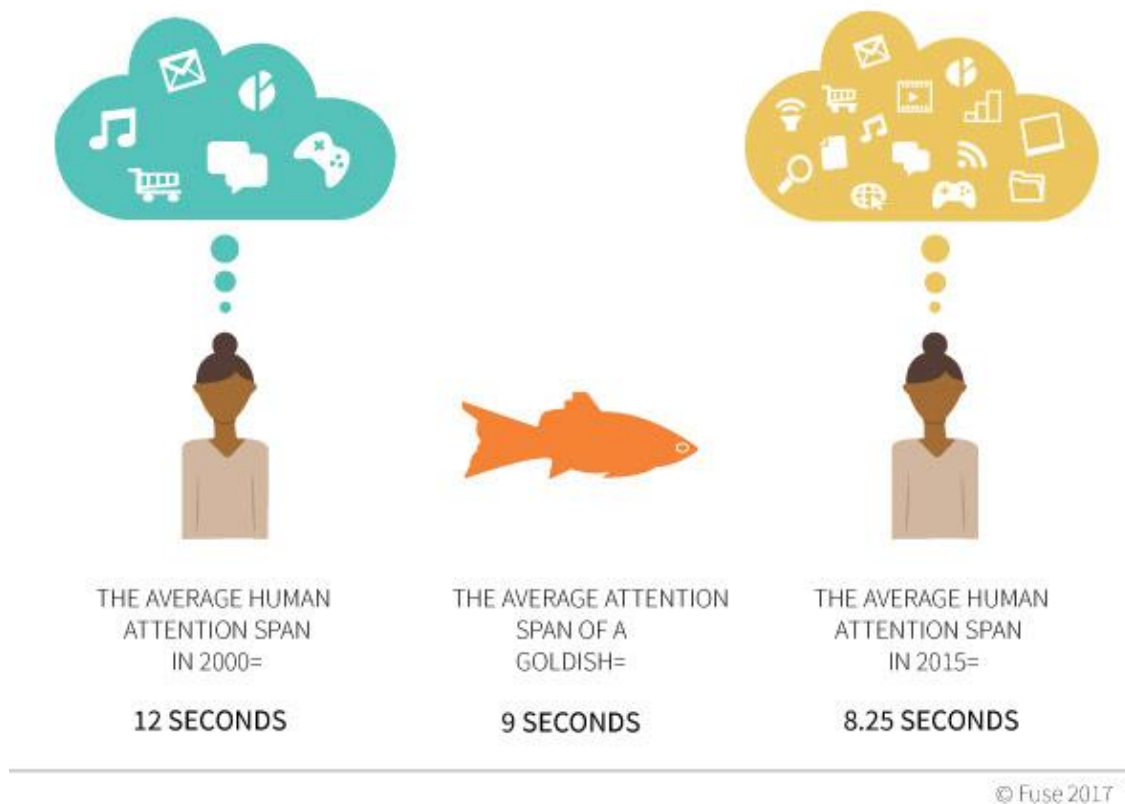
## User Experience (UX)

În primul rând, interfața cu utilizatorul (User Interface) este o parte din UX, existând o diferență semnificativă între cele două. Interfața cu utilizatorul este spațiul unde utilizatorul interacționează cu aplicația, cu diferite componente ale aplicației, iar User Experience este rezultatul final pe care îl produce interacțiunea cu aplicația.

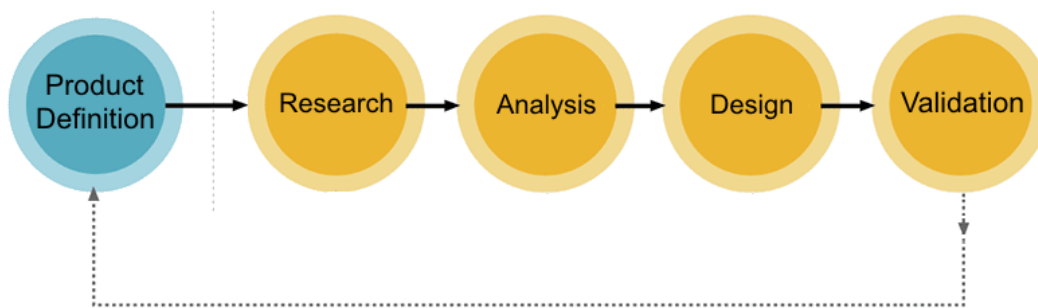
- **Scopul aplicației** – cel mai important lucru este scopul aplicației; Cine o să fie utilizatorii finali? Care sunt nevoile lor? Ce își doresc ei? Etc. De aceea, dezvoltarea unei aplicații poate începe cu un “user research”, fiind un proces esențial pentru UX.
- **Tu nu ești utilizatorul final** – este un lucru esențial în dezvoltarea unei aplicații. Dezvoltatorul nu este utilizatorul final, iar cel mai important lucru este testarea de către un utilizator real. Un dezvoltator nu se va comporta niciodată ca un utilizator real deoarece are experiența și va utiliza aplicația exact cum a fost dezvoltată. Utilizatorii care vor folosi aplicația au moduri diferite de gândire, un background diferit, experiența diferită sau chiar scopuri diferite, ceea ce face testarea cu utilizatori reali cea mai bună formă de testare pentru o aplicație web. (usability testing)



- **Adaptarea design-ului pentru atentie de scurta durata** – nu incarcati utilizatorii cu foarte multa informatie. O perioada de atentie inseamna acea perioada de timp in care cineva se concentreaza asupra unei sarcini fara sa fie distras. Un studiu realizat de Microsoft in 2015 a aratat faptul ca media atentiei umane a scazut de la 12 secunde la 8 secunde, ceea ce inseamna ca avem o perioada de atentie mai scurta decat cea a unui pestisor auriu. Acest lucru inseamna ca este necesar ca designerii sa afiseze oamenilor informatiile necesara cat mai repede posibil. De aceea, lucrurile care nu sunt necesare pot fi eliminate. Acest lucru nu inseamna ca trebuie sa limitam experientele utilizatorilor intr-o aplicatie web, ci doar ca informatiile trebuie sa fie relevante.




- **UX depinde de fiecare proiect in parte** – nu exista un UX general care poate fi aplicat oricarui proiect. Fiecare proiect este unic, are cerinte unice, ceea ce duce la un UX diferit fiecărei aplicatii. De exemplu, atunci cand dezvoltati un nou proiect este necesara alocarea unui timp suplimentar pentru a cerceta ce tipuri de utilizatori vor accesa aplicatia, dar si care sunt specificatiile aplicatiei.



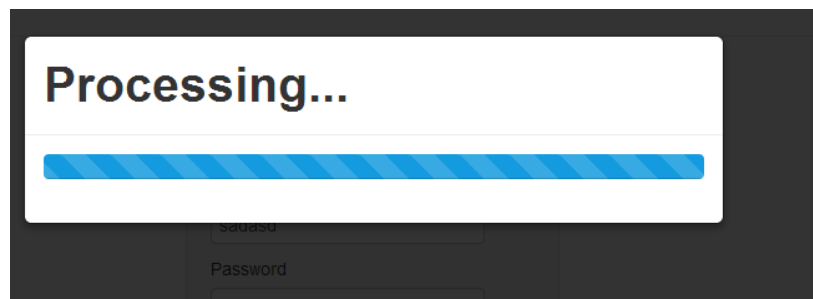
- **Prevenirea erorilor este mai usoara decat rezolvarea lor** – multe erori apar din cauza utilizatorilor (email incorect, parole care nu respecta anumite reguli, etc.). In acest caz aplicatia poate fi dezvoltata oferind sugestii de completare si notificand utilizatorii de fiecare data.

### Register Yourself



You have entered an invalid e-mail address. Please try again.

- **Afisarea feedback-ului** – utilizatorii trebuie sa fie mereu informati de tot ceea ce se intampla in cadrul unei aplicatii: mesaje dupa trimiterea formularelor, mesaje in momentul editarii/stergerii datelor, feedback vizual. Acesta din urma este foarte important fiind un mijloc bun de informare.



- **Designul trebuie sa fie simplu si intuitiv**, utilizatorii folosind de fiecare data intuitia. Steve Krug spunea ca principalul motiv pentru care utilizatorii folosesc intuitia atunci cand aceseaza o aplicatie este ca nu le pasa. *“If we find something that works, we stick to it. It doesn't matter to us if we understand how things work, as long as we can use them”*. S-a constatat ca utilizatorii nu citesc informatia, ci doar o scaneaza, cautand cateva puncte de referinta care sa ii ghideze prin continutul paginii.

- **Spatiile albe sunt importante** – informatia este perceputa mult mai repede deoarece un utilizator, dupa cum am mentionat anterior, scaneaza informatia si o imparte in sectiuni usor de interpretat. Structurile complexe sunt greu de citit, scanat, analizat si lucrat cu ele. Daca aveti de ales intre a separa doua sectiuni cu o linie vizibila sau cu spatii albe, a doua varianta este cea potrivita.

## Alegerea culorilor potrivite

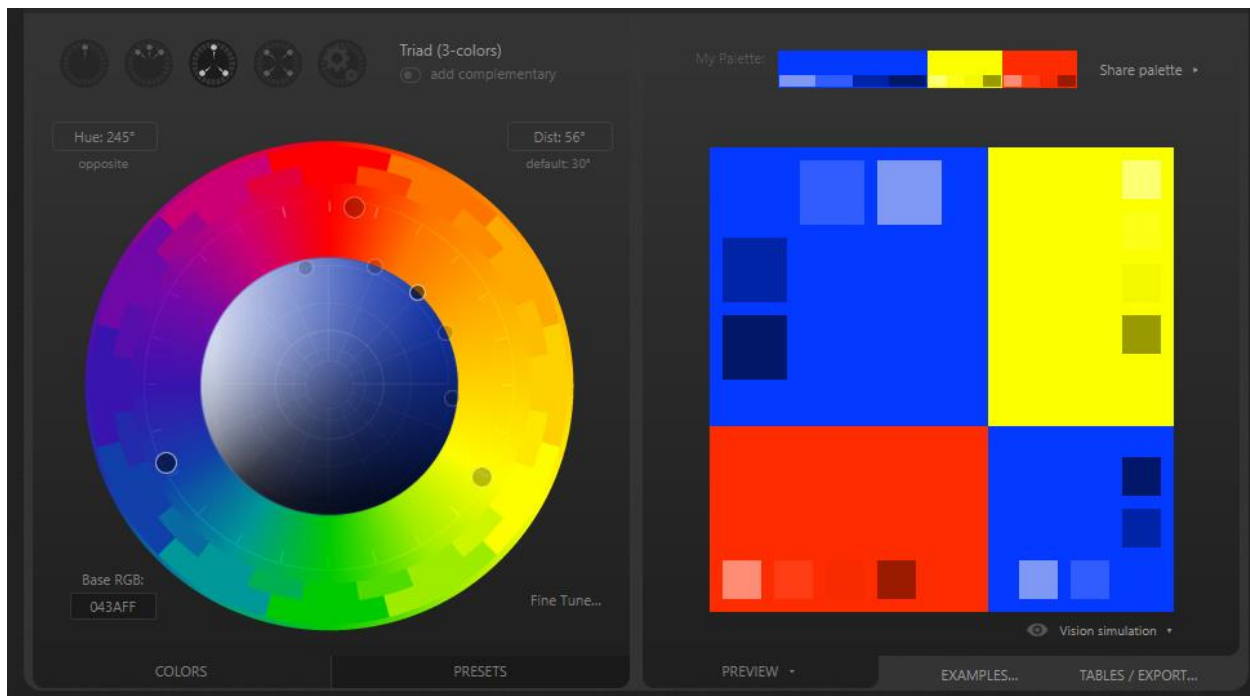
Alegerea culorilor in dezvoltarea unei aplicatii web, de cele mai multe ori, nu este atat de simpla. In continuare o sa urmarim cele mai importante reguli in acest sens: (<https://flatuicolors.com/> )

- **Tehnica 60-30-10** – aceasta tehnica se aplica si in decorarea caselor si este una simpla. Pentru a fi in balanta, culorile trebuie combinate in proportie de 60%-30%-10%. Cea mai mare parte reprezinta culoarea dominanta, urmata de culoarea secundara, iar procentul de 10% il reprezinta culoarea care ajuta la realizarea accentelor in aplicatie.
- **Contrastul** – prin folosirea acestuia se aduce individualitate fiecarui element de interfata, facandu-le pe fiecare in parte vizibile si accesibile. Contrastul este folosit pentru punerea in evidenta a anumitor sectiuni sau pentru butoane de tip call-to-action
- **Psihologia culorilor** – fiecare culoare influenteaza utilizatorul intr-un anumit fel si livreaza catre acesta un anumit mesaj. De exemplu:
  - **rosu** simbolizeaza atat un sentiment pozitiv (incredere) cat si un sentiment negativ (erori, folosire incorecta a unui element)
  - **verde** simbolizeaza sentimente de succes, calmitate (un task executat cu succes, folosirea corecta a aplicatiei, etc)



- **alb** reprezinta puritate si claritate (ofera sentimentul de simplitate in cadrul unei aplicatii. Cum a spus Steve Jobs, inspirat de Leonardo da Vinci: “simplitatea este sofisticarea absoluta”)

Culorile se pot alege folosind tool-uri existente. De exemplu, [palleton.com](http://palleton.com) este o astfel de aplicatie, online, care asista utilizatorul sa genereze paleta de culori corecte plecand de la o culoare de baza. Plecand de la culoarea de baza **albastru** observam ca aplicatia ne alege culorile complementare **rosu** si **galben** si genereaza si paleta de culori aferenta acestor culori.



Aceste culori pot fi folosite pentru a crea interfata curata si pentru a evidientia anumite elemente in cadrul unei aplicatii (atat web cat si de alta natura).

# ASP.NET MVC Best Practices

## Controller-ul

- Crearea unui Controller de baza poate ajuta la definirea unor metode necesare pentru toate Controller-ele aplicatiei. Acest Controller de baza poate fi mostenit mai apoi de orice alt Controller din aplicatie:

```
public class BaseController : Controller
{
    protected virtual LoginPrincipal LoggedUser
    {
        get { return HttpContext.User as LoginPrincipal; }
    }

    protected int GetUserId()
    {
        int userId = -1;

        if (Request.IsAuthenticated)
        {
            userId = LoggedUser.UserId;
        }

        return userId;
    }
}

public class InterviewController : BaseController
```

- Atentie la verbele HTTP definite in Controller si la rutele aferente acestora. Cea mai buna practica pentru procesarea datelor pe server este metoda **PRG** (Post-Redirect-Get: Datele sunt **trimise** (post) la server. Metoda procesareaza informatiile aferente si face **redirect** catre o ruta diferita (redirect) care va **afisa** continutul nou creat sau o alta pagina (get)).

## View-ul

- View-ul nu trebuie sa contina foarte multa logica specifica Controller-ului
- Parzialele pot scuti foarte mult timp de implementare a componentelor comune

## Modelul

- Modelele necesita o validare foarte buna (pentru siguranta si consistenta datelor)
- Fiecare model trebuie definit in fisierul lui pentru o mai buna consistenta
- Relatiile trebuie sa poarte numele modelului pentru ca framework-ul sa faca asocierea intre tabele in mod automat.  
**Exemplu:** pentru FK UserId trebuie sa avem relatie cu modelul User. Pentru CategoryId trebuie sa avem relatie cu modelul Category, etc.

## Sistemul de rutare

- Rutele trebuie sa fie definite intr-un mod foarte simplu si intuitiv
- Nu este necesar crearea unei rute pentru fiecare componenta a aplicatiei daca ruta Default sau o alta ruta definita poate sa rezolve acel URL
- In fisierul de rute este folositor sa se scrie in comentarii, pentru fiecare ruta in parte, ce URL-uri poate sa rezolve din cadrul aplicatiei dezvoltate
- Pentru un debug optim al rutelor, la rulare, eliminati cat mai multe rute sau lasati doar ruta Default pentru a observa cum se rezolva acel URL