**LUCRARE DE LICENTA**

Aplicatie de gestiune a fluxului de lucru intr-un department pentru suport tehnic

Indrumator stiintific: Absolvent:

Conf. Univ. Dr. Mirel Cosulschi Tereanu Stefan Bogdan

**Cuprins**

[Capitolul 1 Introducere 4](#_Toc35974207)

[1.1 Internetul – indispensabil in lumea moderna 4](#_Toc35974208)

[1.2 Impactul tehnologiei in mediul de afaceri 5](#_Toc35974209)

[1.2.1 Importanta internetului in afaceri de mici dimensiuni 6](#_Toc35974210)

[1.2.2 Importanta internetului in companiile de mari dimensiuni 7](#_Toc35974211)

[1.3 Istoria si utilitatea practicilor ITIL 8](#_Toc35974212)

[1.4 Utilizarea aplicatiei 10](#_Toc35974213)

[Capitolul 2 Aplicatii de gestionare a fluxului de lucru 11](#_Toc35974214)

[Capitolul 3 Tehnologii utilizate 14](#_Toc35974215)

[3.1 IDE-ul Microsoft Visual Studio 2019 14](#_Toc35974216)

[3.2 Limbajul C# 15](#_Toc35974217)

[3.3 Framework-ul ASP.NET MVC 18](#_Toc35974218)

[3.4 HTML, CSS, Javascript si HTML Helpers 19](#_Toc35974219)

[3.5 Entity Framework si Microsoft SQL Server 19](#_Toc35974220)

[3.6 Google Charts 19](#_Toc35974221)

# Capitolul 1 Introducere

## 1.1 Internetul – indispensabil in lumea moderna

Ușurința cu care se realizează comunicarea digitala in zilele noastre este datorata Internetului. Termenul de internet provine din telescoparea termenilor din limba engleza: ***inter****connected* = interconectat și ***net****work* = rețea. Aşadar internetul reprezintă o rețea globală care interconectează miliarde de dispozitive între ele, facilitând accesul la informație si comunicare oricui are acces la acesta. O caracteristică a internetului este decentralizarea, lipsa unui proprietar. Acesta nu este detinut de nici o entitate, deși la nivel local exista instutii care pot reglementa diferite aspecte ale limitărilor acestuia.

Termenul World Wide Web reprezintă o parte a internetului in care documentele si resursele web sunt organizate, unde identificarea acestora se realizează prin intermediul unor coduri de forma Uniform Resource Locators (URL).

Consideram un aspect ca fiind indispensabil daca s-a inradacinat in rutina noastra zilnica atat de puternic incat ne este aproape imposibil sa ne imaginam viața fara el. Odată cu evoluția tehnologiei, internetul a devenit prezent atât in timpul liber al oamenilor, fiind una dintre principalele surse de divertisment, cu o plaja larga de zone de interes, cat si in acțiunile zilnice ale locului de munca. Tot mai multe slujbe depind de internet in însuși natura lor, iar unele actiuni repetitive,necesare locului de munca(exemplu verificarea mail-urilor) sunt ritualizate in jurul folosi internetul. Putem confirma ca timpul de utilizare al internetului ocupa o mare parte din activitățile zilnice, iar repetitivitatea cu care întreprindem aceasta actiune transforma aceasta practica in rutina.

Viitorul ne promite o conexiune mai strânsa a omului cu internetul prin conceptul relativ nou de IoT(Internet of Things). Conceptul de IoT prin intermediul internetului,doreste sa permita interconectarea obiectelor care vor putea primi si trimite date intre ele sub controlul oamenilor.

Putem spune ca internetul a devenit indispensabil in lumea moderna, iar vietile noastre sunt strans legate de existenta sa.

Printre avantajele internetului putem menționa:

* Conectivitatea, ușurința comunicării
* Crearea noilor locuri de munca, munca de acasă, accesul la o piața globala
* Conceptul de Internet of Things
* Căutarea eficienta a informației

## 1.2 Impactul tehnologiei in mediul de afaceri

Tehnologia a revoluționat modul în care companiile își desfășoară activitatea. Acum companiile de mici dimensiuni pot concura cu cele de mari dimensiuni, ambele categorii avand posibilitati de expunere similare. Companiile de mari dimensiuni, prin intermediul tehnologiei, reușesc sa dezvolte o comunicare stabila intre angajati, indiferent de locatia acestora, fapt ce facilitează organizarea si dezvoltarea afacerii.

Am impartit companiile in doua categorii in funcție de dimensiunea acestora:

* Companii de mici dimensiuni = corporații private, parteneriate sau proprietăți unice care au mai puțini angajați și / sau mai puține venituri anuale decât o afacere de dimensiuni regulate sau o corporație.[[1]](#footnote-1)
* Companii de mari dimensiuni = corporații cu un număr mare de angajați si venituri anuale mult mai mari decât o companie de mici dimensiuni. Acestea au in general puncte de lucru in mai multe orașe, tari sau continente ale lumii.

Intern, tehnologia a imbunatatit comunicarea intre afacere si angajat prin usurarea dificultatii modului de a răspândi si monitoriza informația. Acum angajatul nu este nevoit sa fie fizic in birou pentru a afla starea in care se afla afacerea sau pentru a comunica cu alti angajati. Comu-nicarea prin mesaje se poate realiza prin intermediul mesajelor de tip email iar supravegherea afaceri prin aplicatii care folosesc internetul.

Extern, companiile au dezvoltat o multitudine de mijloace cu ajutorul carora sa capteze interesul eventualilor clienti. O buna comunicare intre companie si client inseamna atat o modali-tate de expunere si prezentare a companiei (exemplu: website), cat si o modalitate eficienta de suport pentru eventuale probleme intampinate de clienti (exemplu: livechat).

Nevoia afacerilor de a intelege dorintele clientilor, dar si de a supraveghea aspecte legate de munca angajatilor a dus la dezvoltarea ramurii de statistica in afaceri. Analizele statistice permit afacerilor sa identifice trendurile si sa masoare performantei angajatilor. Prin aceste metode de analiza deciziile devin mai sigure, acum fiind bazate pe date, nu pe presupuneri. O analiza statistica foloseste numerele pentru a reprezenta aspecte ale realitatii, care pot fi ulterior folosite in intelegerea mai corecta a acelor aspecte. Abilitatea cu care numerele pot fi convertite in valori de tip procente, medii, rapoarte, mediene ofera afacerilor o varianta simplificata si explicita a analizelor care favorizeaza intelegerea si interpretarea informatiilor prin intermediul diagrame explicative. Un avantaj al analizelor statistice este dat de posibilitatea de compararare, care ofera numerelor valoare si sens.

Colectarea datelor a devenit o practica intalnita la companii de orice marime. Internetul imbina rapid analiza statistica si colectarea datelor permitand companiilor observarea in timp real a rapoartelor.

Răspândirea internetului a creat din acesta o unelata esentiala in mediul de afaceri care prin versatilitate poate avea utilitate in orice sector si la orice nivel al afacerilor.

Internetul reuseste sa imbunatateasca mediul de afaceri inovand domenii care la baza sunt concepute pentru o lume lipsita de acesta. Deseori intalnim prefixul „E-” in constructia cuvintelor care definesc acele domenii cum ar fi: E-commerce, E-shopping, E-banking, E-sport sau E-mail.

### 1.2.1 Importanta internetului in afaceri de mici dimensiuni

Afacerile de dimensiuni mici au ca public ținta, in general, un număr restrâns de potențiali clienți dintr-un spațiu limitat. In momentul in care afacerile au produse sau servicii similare, cel mai important factor al dezvoltării unei afaceri in detrimentul altora este promptitudinea cu care ajunge la publicul ținta. In acest punct intervine internetul care sprijină afacerile in încercarea lor de a ajunge la oameni.

In zilele noastre, o firma de mici dimensiuni, dispune cel puțin de un website ca mijloc de interacțiune cu potențialii clienți. Pentru o indexare favorabila in topul listei de căutare care duce automat la o vizibilitate crescuta in aria domeniului de interes, afacerile folosesc procese de perfecționare a vizibilității website-urilor in cadrul listei care rezulta in urma utilizarii motorului de căutare. Aceste procese au numele de tehnici de optimizare a motorului de cautare sau Search Engine Optimization(SEO).

Bugetele limitate obliga firmele de dimensiuni mici sa gaseasca solutii eficiente la preturi cat mai reduse pentru promovare si expunere. Internetul ofera suport prin intermediul uneltelor de automatizare a actiunilor umane. Atat pe partea de social media cat si in scopul trimiterii de email-uri, uneltele de automatizare reusesc sa efectueze operatiuni repetitive care favorizează promovarea firmelor, cu costuri reduse.

Intelegerea clientilor se realizeaza prin analize statistice realizate asupra mijloacelor de expunere prin intermediul internetului. Desi afacerile de dimensiuni reduse nu dispun de un analist, aceste pot utiliza o multitudine de unelte de analiza care ofera informatii despre comportamentul clientilor in interactiunea cu mijloacele de expunere ale companiei.

### 1.2.2 Importanta internetului in companiile de mari dimensiuni

Afacerile de dimensiuni mari dispun deja de un numar ridicat de clienti dintr-o multitudine de zone ale lumii. Scopurile pe care doresc sa le indeplineasca sunt comunicarea intre angajati si extinderea pe un plan cat mai robust.

Intr-o afacere care se extinde, nevoia de a găsii noi angajați este ridicata, iar daca aceasta are puncte de lucru in întreaga lume si exista un număr considerabil de aplicații pentru locurile de munca disponibile, întâlnirea de către reprezentanți cu toți posibilii angajați este imposibila. Internetul facilitează modalitati de intervievare si testare eficiente pentru a diminua timpul de căutare si a gasii cea mai buna soluție pentru posturile vacant.

Firmele de mari dimensiuni folosesc un procent considerabil din venituri pentru a lucra cu cele mai actuale unelte care maresc productivitatea si cresc eficienta pe diferite sectoare cum ar fi:

* Sisteme de management IT (ITSM)
* Sisteme de gestiune a relatiilor cu clientii (CRM)
* Social Media Marketing & Management
* Cercetare si analiza in marketing
* Finantare si procesarea platilor[[2]](#footnote-2)

## 1.3 Istoria si utilitatea practicilor ITIL

ITIL(acronim de la  Information Technology Infrastructure Library) a fost creat în anii 1980 de către guvernele britanice CCTA(Agenția Centrală de Calculatoare și Telecomunicații) cu scopul de a asigura o mai bună utilizare a serviciilor și resurselor IT.

Agenția Centrală de Calculatoare și Telecomunicații (CCTA), numită acum Office of Government Commerce (OGC), a primit sarcina de a dezvolta un cadru pentru utilizarea eficientă și responsabilă din punct de vedere financiar a resurselor IT în cadrul guvernului britanic și al sectorului privat.

Cea mai veche versiune a ITIL a fost numită inițial GITIM(acronim de la Government Information Technology Infrastructure Management). Evident, acest lucru a fost foarte diferit de ITIL-ul actual, dar conceptual foarte similar, concentrându-se în jurul asistenței și livrării serviciilor.

Companiile mari și agențiile guvernamentale din Europa au adoptat cadrul foarte repede la începutul anilor ’90. ITIL-ul s-a răspândit, fiind folosit atât în ​​organizații guvernamentale, cât și neguvernamentale. Pe măsură ce a crescut în popularitate, atât în ​​Marea Britanie, cât și în întreaga lume, IT-ul în sine s-a schimbat și a evoluat, la fel și ITIL.

În 2000, CCTA a fuzionat cu OGC, Office for Government Commerce și, în același an, Microsoft a folosit ITIL ca bază pentru dezvoltarea propriului său Microsoft Operations Framework (MOF).

În 2001, versiunea 2 a ITIL a fost lansată. Cărțile de asistență și livrare de servicii au fost reamenajate în volume mai concise. În următorii câțiva ani a devenit, de departe, cea mai utilizată abordare a celor mai utilizate practici de gestionare a serviciilor IT din lume.

În 2007, versiunea 3, dacă a fost publicat ITIL. Aceasta a adoptat mai mult o abordare a ciclului de viață a managementului serviciilor, cu un accent mai mare pe integrarea afacerilor IT.

În 2009, OGC a anunțat oficial că certificarea ITIL Versiunea 2 va fi retrasă și a lansat o consultare majoră în funcție de modul de procedare.

În 2011, a fost publicată o noua ediția a ITIL, oferind o actualizare la versiunea publicată în 2007. OGC nu mai este listat ca titular al ITIL, în urma consolidării OGC în Cabinet Office.

Acum procesele ITIL sunt un set detaliat de practici impartite in 3 categorii:

* Practici generale de management
* Practici de management al serviciilor
* Practici de management tehnic

Practicile utilizate in managementul serviciilor IT (ITSM) se concentrează pe alinierea serviciilor IT cu nevoile actuale in mediul de afaceri.

Service Desk este o solutie bazata pe conceptul de ITSM, care inglobeaza o serie de practici ITIL in furnizarea unui punct de contact care raspund la nevoile de comunicare dintre utilizatori si personalaul IT. Acesta se focuseaza pe raportarea proceselor de tip *incident* sau *request*. Acestea urmeaza sa fie directionate catre personalul IT abilitat in gestionarea si luarea unei decizii.

Un incident este descris printr-o intrerupere neplanificata a unui serviciu IT, o reducere a calitatii unui serviciu IT sau defectiunea unui element de configurare care nu a afectat inca un serviciu IT.[[3]](#footnote-3)

Un request este descris printr-o cerere de a furniza un produs sau serviciu care afecteaza in acel moment productivatate angajatului, urmeaza sa afecteze in scurt timp buna functionare a angajatului in segmentul IT, sau ii poate fi folositor angajatului in cresterea productivitatii sau calitatii.

## Utilizarea aplicatiei

Proiectul „Aplicatie pentru gestionarea fluxului de lucru intr-un departament pentru suport tehnic” consta in dezvolarea unei solutii pentru sporirea productivitatii in departamentul de suport tehnic al companiilor care dispun de acest tip de departament prin implementarea unor serii de practici ITSM. Aplicatia este de tip web , fiind adaptata pentru utilizarea atat pe computerul personal cat si pe telefonul mobil.

Aplicatia ofera solutii pentru nevoile angajatilor companiei in scopul raportarii posibilelor probleme, angajatilor din departamentele destinate suportului tehnic prin posibilitatea realizarii unui flux de lucru care faciliteaza urmarirea progresului problemei si rezolvarea acestea intr-o perioada cat mai scurta dar si administratorului care poate superviza angajatii si analiza rezultatele.

Printre principiile ITSM implementate in aplicatie amintim:

* Administrarea rapoartelor de tip incident si request
* Urmarirea fluxului de lucru si evolutiei unui raport
* Extragerea de statistici in scopul analizarii productivitatii

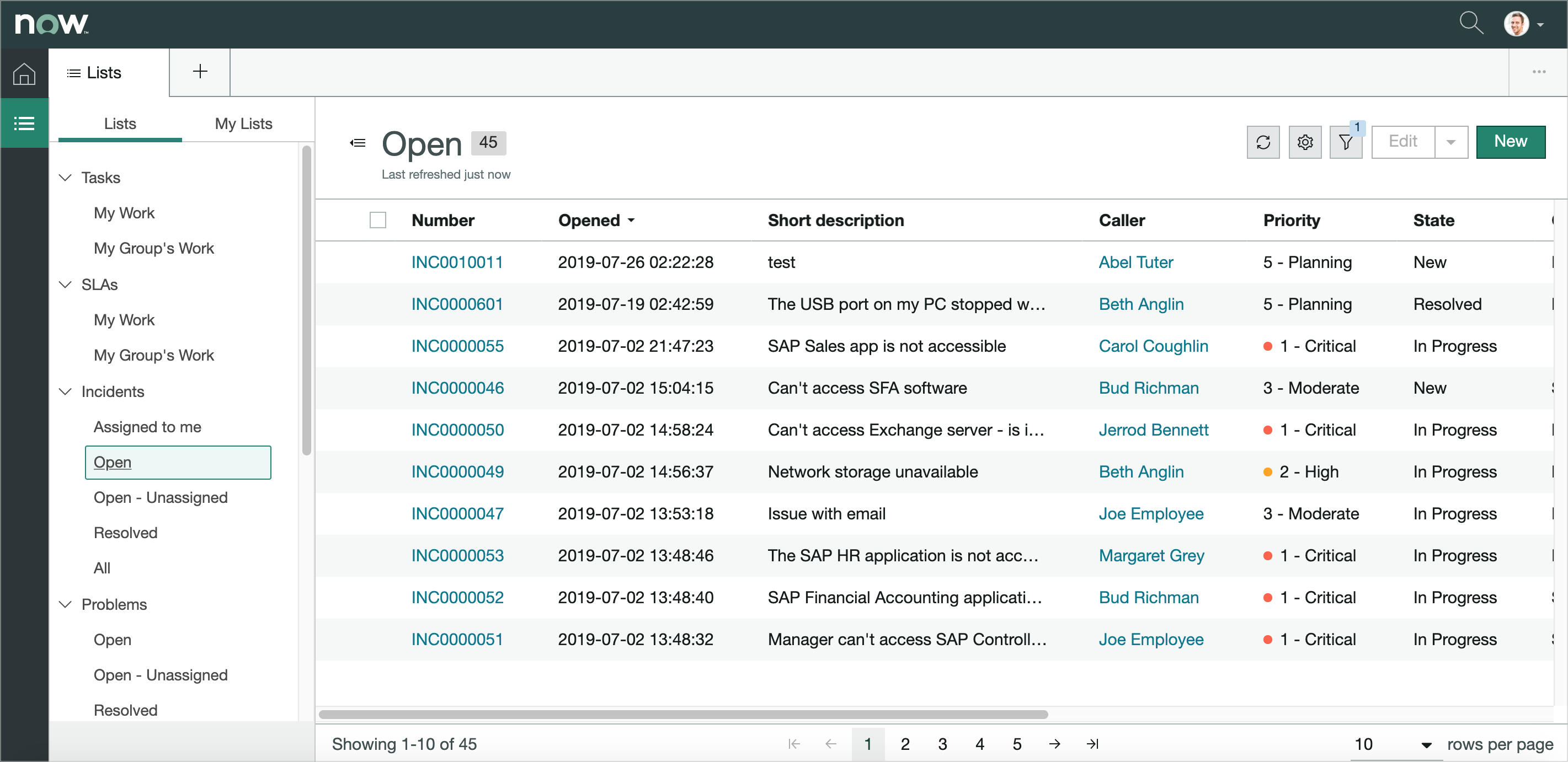
# Capitolul 2 Aplicatii de gestionare a fluxului de lucru

**ServiceNow IT Service Management** este unul din serviciile oferite de compania

ServiceNow. Conform Gartner Magic Quadrant, ServiceNow IT Service Management este cea mai buna alegere in randul serviciilor de tip ITSM in ultimii 6 ani. Solutia celor de la ServiceNow ofera o metoda moderna de gestionare a afacerilor, cu o multitudine de modalitati de customizare a serviciilor pentru a se mula pe nevoile utilizatorului. Aceasta solutie este dezvoltata pe o platforma de tip cloud.

Modalitatile de automatizare a fluxului de lucru si implementarea inteligentei artificiale in scopul rezolvarii problemelor des intalnite cresc productivitatea si reduc costurile, fapt ce motiveaza afacerile in alegerea serviciului ServiceNow IT Service Management.

Serviciul se poate accesa sub forma unei abonament anual, iar accesul la aceste poate fi facut prin intermediul browserului web sau al unui aplicatii mobile.



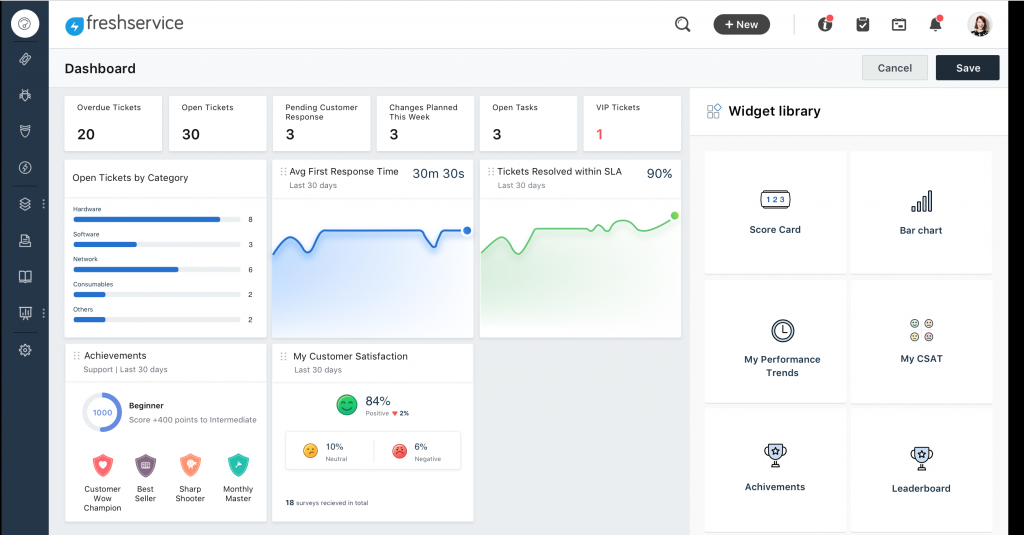
Figură : Prezentare generală ServiceNow

**Freshservice** este o solutie de tip ITSM oferita de compania Freshworks, pentru gestionarea serviciilor, bazata pe cloud care simplifica operatiunule IT in mediul de afaceri. Solutia oferita de Freshworks administreaza incidentele, problemele, schimbarile si comunicarile prin functii de tip adaugare, urmarire , etichetare sau eliminare.

Freshservice ofera de asemenea un modul de transformare a activitatilor intr-un joc, numit “Arcade”. Acest modul activeaza o mecanica de optinere a punctelor si realizarea unui scor pentru realizarea activitatilor. Aceasta solutie permite administratorului sa genereze rapoarte predefinite pentru incidente, folosind diferite filter.

In principal, solutia este creata pentru a fi folosita intr-un browser web, dar are de asemenea oplicatii mobile pentru iOS si Android care faciliteaza administratorului supravegherea de la distanta.

Solutia poate fi utilizata pe baza unui abonament anual care include de asemenea suport prin email si telefonic.



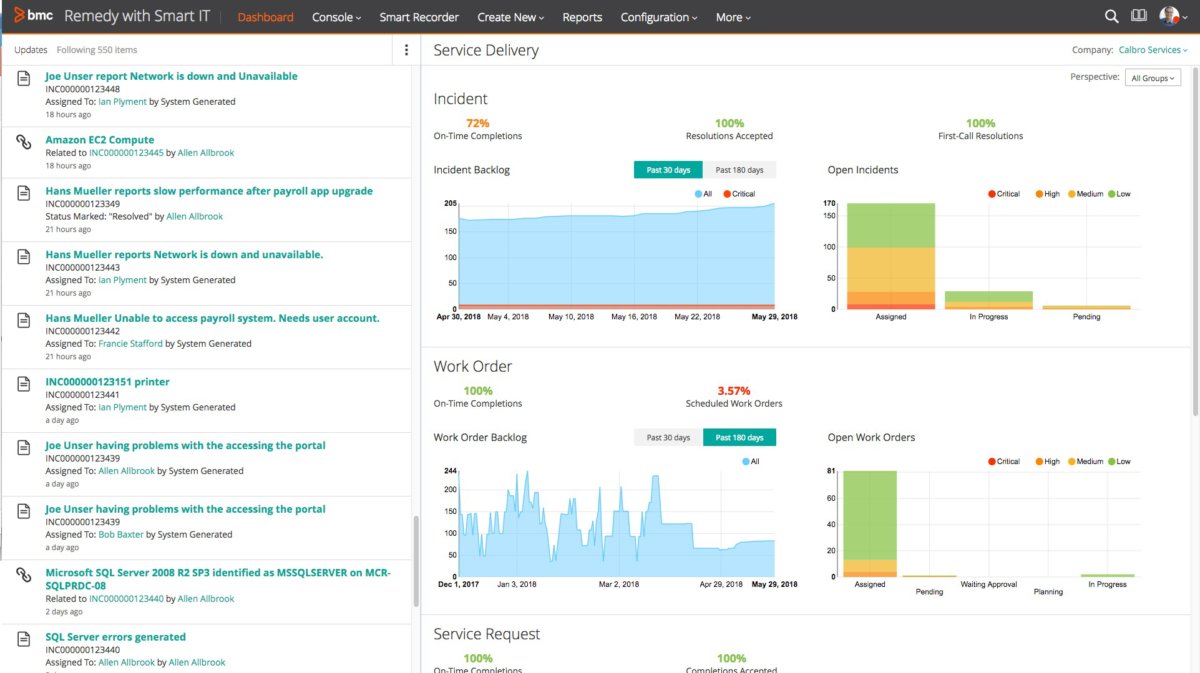
Figură : Prezentare generala Freshservice

BMC Software, Inc. este o companie americană de software destinata mediului de afaceri. BMC oferă software și servicii în domenii precum cloud computing, gestionarea serviciilor IT, automatizare și mainframe. Acțiunile BMC au fost tranzacționate inițial pe NASDAQ sub simbolul BMCS și la Bursa din New York cu simbolul BMC, dar compania este privata din punct de vedere al actinarilor in acest moment. BMC este o firmă multinațională care operează în America de Nord, America de Sud, Australia, Europa și Asia și are mai multe birouri situate în întreaga lume. Sediul internațional al companiei se află în Houston, Texas, Statele Unite.[[4]](#footnote-4)

**BMC Helix** **Suite** este solutia comaniei BMC Software care prin implementarea unor serii de practici ITIL constituie unul din cele mai utilizate aplicatii software pe segmentul IT Service Management.

Solutia BMC Helix Suite include, pe langa caracteristicile de baza pentru crearea unui ITSM si module precum chatbot, microservicii native de tip cloud, posibilitatea automatizarii fluxurilor de lucru. BMC Helix Suite este dotata cu capacitati integrate de invatare inteligenta si machine learning.

Aceasta solutie IT Service Management poate fi operata atat de pe computerul personal cat si printr-o aplicatie mobila.



Figură : Prezentare Generala BMC ITSM

# Capitolul 3 Tehnologii utilizate

## 3.1 IDE-ul Microsoft Visual Studio 2019

Mediul de dezvoltare Microsoft Visual Studio este utilizat pentru dezvoltarea de programe de sine statatoare, precum si pentru site-uri web, aplicatii web, servicii web si aplicatii mobile. Visual Studio utilizeaza platforme de dezvoltare software Microsoft, precum Windows API[[5]](#footnote-5), Windows Forms[[6]](#footnote-6), Windows Presentation Foundation[[7]](#footnote-7), Windows Store[[8]](#footnote-8) si Microsoft Silverlight[[9]](#footnote-9).

Visual Studio include un editor de cod care suporta IntelliSense precum si refactorizarea codului. Acesta incorporeaza un profilator de coduri, proiectat pentru construirea GUI[[10]](#footnote-10)-ului aplicatiilor, un designer web, un designer de clase si un designer de scheme de baze de date.

Visual Studio accepta plug-in-uri care imbunatatesc functionalitati in aproape toate sectoarele, inclusiv adaugarea de suport pentru sistemele de control sursa (precum Subversion [[11]](#footnote-11)si Git[[12]](#footnote-12)) si adaugarea de noi seturi de instrumente precum editori si designeri vizuali pentru limbaje sau seturi de instrumente specifice domeniului pentru alte aspecte ale dezvoltarii ciclului de viata al aplicatiilor software (precum serverul Azure DevOps: Team Explorer[[13]](#footnote-13)).

Visual Studio acceptă 36 de limbaje de programare diferite și permite editorului de coduri și depanatorului să sprijine (în grade diferite) aproape orice limbaj de programare, cu condiția să existe un serviciu specific limbajului. Limbile încorporate includ C , C ++, C ++ / CLI, Visual Basic .NET, C #, F #, JavaScript, TypeScript, XML, XSLT, HTML și CSS. Asistența pentru alte limbi, cum ar fi Python, Ruby, Node.js și M, printre altele, este disponibilă prin pluginuri. Java a fost acceptat în trecut.[[14]](#footnote-14)

IDE-ul Visual Studio 2019 este publicat in 3 versiuni:

* **Visual Studio 2019 Community Edition** – versiunea gratuita de IDE a celor de la Microsoft destinata mediului academic si programatorilor individuali. Desi este gratuita, aceasta versiune permite dezvolatrea de aplicatii comerciale si monetizarea acestora in cazul in care venitul anual nu depaseste un milion de dolari si programul este instalat pe maxim cinci calculatoare.
* **Visual Studio 2019 Professinal Edition** – aceasta versiune difera prin caracteristici minore fata de varianta precedenta, cum ar fi proprietatea CodeLens[[15]](#footnote-15) care detecteaza schimbarile de cod si genereaza un istoric. Acesul la aceasta versiune se realizeaza prin abonament lunar sau anual, iar programul este destinat afacerilor care depasesc venituri de un milion de dolari anual.
* **Visual Studio 2019 Enterprise Edition –** aceasta versiune aduce o multime de noi proprietati versiunii Professinal precum moduri avansate de depanare si diagnosticare a a erorilor, unelte de testare, diverse proprietati pentru framework-urile de dezvoltare pe multiple platforme si imbunatatiri ale mediului de dezvoltare, in principal pentru dezvoltarea de aplicatii de mari dimensiuni. Acesul la aceasta versiune se realizeaza prin intermediul unui abonament lunar sau anual, iar pretul este mult mai mare in comparatie cu versiunea Professinal.

Desi IDE-ul este devoltat de Microsoft, acesta are suport si pentru produsele firmei Apple de tip Mac sau Macbook.

## 3.2 Limbajul C#

C# este un limbaj modern de programare dezvoltat de Microsoft si lansat in anul 2000 care introduce conceptele de programare orientata pe obiect si deriva din doua limbaje de programare care sunt de mare succes si in zilele noastre: C si C++. Acesta a fost conceput ca o alternativa la limbajul Java care aparuse cu 5 ani in urma, fiind de asemenea inspirat de limbaje precum C si C++.

Denumirea de „C sharp” a fost inspirata de notatia muzicala, unde diez(sharp #) indica faptul ca nota scrisa ar trebui cantata cu un semiton mai ridicat. Numele limbajului este similar cu cel al limbajului C++, unde notatia „++” indica faptul ca o variabila urmeaza a fi incrementata cu 1 dupa evaluare. Simbolul # poate fi ca o concatenare suprapusa a doua formatiuni „++”, fapt ce denota ideea de incrementare asupra limbajului C++.

Limbajul C# a fost dezvoltat de Microsoft pentru propria lor arhitectura numita .NET(pronuntie: dot net). Arhitectura .NET defineste un mediu de programare care permite dezvolatrea de aplicatii pe mai multe platforme. Acesta permite programarea in limbaj mixt si ofera facilitati de securitate si portabilitate programelor.

Limbajul C# este asemănător cu limbajele C si C++ pe care le moștenește in multe aspecte ale structurii si conceptelor dupa cum putem vedea in programul urmator:



Cod sursa : Program C# de afiseare text

Limbajul C# sta la baza creari de aplicatii care utilizeaza arhitectura .NET pe o mutitudine de platforme, imbinandu-se cu o multime de limbaje.

Printre platforme amintim:

* Desktop: WPF, WindowsForms, UWP
* Web: ASP.NET
* Cloud: Azure
* Mobile: Xamarin
* Gaming: Unity
* Internet Of Things: ARM32, ARM64
* Inteligenta artificiala: ML.NET, .NET for Apache Spark.

Fiind un limbaj orienta-obiect, C# respecta urmatoarele principiile programarii orientate obiect:

* Incapsularea = procesul de grupare a datelor si metodelor intr-o singura structura, datele fiind definite alaturi de codul care poate actiona asupra lor. Accesul la date se realizeaza in limbajul C# atat prin crearea metodelor obisnuite de tip set si get cat si printr-o terminologie prescurtata:

|  |  |
| --- | --- |
| Cod sursa : Exemplu incapsulare - metoda prescurtata | Cod sursa : Exemplu incapsulare - metoda obisnuita |

Principalii modificatori de acces in limbajul C# care deservesc principiul de incapsulare sunt:

* Public: acces oriunde in interiorul proiectului
* Protected: acces in interiorul clasei, dar si in clasele care mostenesc acea clasa
* Private: acces doar in interiorul clasei
* Internal: acces in interiorul acelui DLL
* Abstractizarea = procesul de ascundere a anumitor date, oferind utilizatorului accesul doar la informatiile esentiale. Acest principiu se realizeaza utilizand clase abstract sau interfete.
* Mostenirea: procesul prin care o clasa mosteneste atributele si metodele unei clase de baza, folosindu-le pentru a se extinde
* Polimorfismul: procesul prin care o operatie sa se realizeze in mai multe feluri

## 3.3 Framework-ul ASP.NET MVC

ASP.NET MVC este un framework open-source construit de Microsoft pentru a dezvolta aplicația web care permite o buna separare a codului. Framework-ul ASP.NET MVC este cea mai personalizabilă și extensibilă platformă livrată de Microsoft.

Cu alte cuvinte, putem spune că ASP.NET MVC este un framework de dezvoltare a aplicațiilor web de la Microsoft care se bazează pe modelul de design arhitectural MVC (Model-View-Controller) pentru a delimita intr-o modalitate eficienta codul, in scopul crearii unui mediu de lucru potrivit pentru echipe de programatori care lucreaza la acelasi proiect.

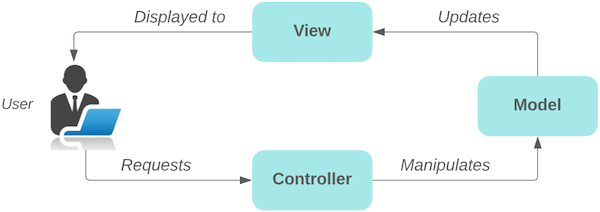
ASP.NET MVC nu este construit de la zero. Acesta este o alternativă completă la tradiționalele ASP.NET WebForms. Este construit utilizant arhitectura ASP.NET, astfel încât dezvoltatorii se bucură de aproape toate caracteristicile ASP.NET în timp ce construiesc aplicația MVC.

Microsoft a anuntat ca lucreaza la acest framework inca din anul 2007, dar lansarea oficiala a fost pe 13 martie 2009 odata cu versiunea ASP.NET MVC 1.0. Ultima versiune lansata inca din 29 noiembrie 2018 este ASP.NET MVC 5.2.7.

Arhitectura Model-View-Controller este un sablon de proiectare ce separa o aplicatie in trei grupuri principale: Models, Views, Controllers. Aceste grupuri ajuta la separarea responsabilitatilor intr-o aplicatie.[[16]](#footnote-16)

Utilizand acest sablon, cererile utilizatorului sunt directionate catre Controller, care este responsabil pentru colaborarea cu Model-ul pentru a efectua actiunile solicitate de utilizator si/sau pentru a prelua rezultatele interogarilor. Controller-ul este responsabil cu alegerea View-ului care urmeaza a fi afisat utilizatorului, alaturi de datele care vor fi furnizate prin intermediul Model-ului.

Urmatoarea diagrama prezinta cele trei componentele, alaturi de fluxul lor de lucru in urma cererilor utilizatorului.



Figură : Arhitectura Model-View-Controller

Aceasta modalitate de delimitare a responsabilitatilor ajuta in extinderea aplicatiilor din punt de vedere al complexitatii, deoarece este mai usor de codat, depanat si testat o functionalitate bine definita intr-una din cele trei componente(Model, View sau Controller), aceasta are o singura responsabilitate.

Dificultatea creste odata cu incercare de actolizare,testare si depanare a codului care are numeroase dependente.

Responsabilitatile celor trei componente sunt impartite astfel:

* Responsabilitatea componentei Model

Componentele View si Controller depind de componenta Model, dar componenta Model nu depinde de clelelate. Acesta aspect al separarii responsabilitatilor, ofera beneficii in dezvoltarea si testarea independenta de partea vizuala a componentelor de tip Model.

Acesta componenta este responsabila de incapsularea si gestionarea datelor, logicii si regulilor aplicatiei.

* Responsabilitatea componentei View

Componenta View este responsabila de orice reprezentarea a continutului din interfata utilizatorului, cum ar fi texte, imagini, diagrame sau tabele.

Rolul acestei componente este de a evidentia informatii obtinute de la model sau informatiile proprii, si de a le trimite la Controller daca este nevoie.

* Responsabilitatea componentei Controller

Componenta Controller se ocupa de gestionarea cererilor utilizatorului. Acesta colaboreaza cu componenta Model pentru a furniza informațiile care urmeaza a fi redate in componenta View. Acesta primeste dete de intarare in urma unei cereri, le valideaza daca este cazul, iar in functie de rezultatul validarii datelor controllerul alege modelul care urmeaza a fi randat in View, catre utilizator.

## 3.4 Microsoft SQL Server si Entity Framework

**Microsoft SQL Server** este un sistem relațional de gestionare a bazelor de date dezvoltat de compania Microsoft. Ca server destinat gestiunii bazelor de date, acesta este un produs software care are ca scop stocare și preluare a datelor, așa cum solicita alte aplicații software care poate rula fie pe același computer, fie pe un computer de pe o alta rețea (inclusiv pe Internet).

Microsoft comercializează o multitudine de ediții diferite de Microsoft SQL Server, destinate unor audiențe diverse, in functie de scopul in care acestia doresc sa utilizeze un sistem de gestiune a bazelor de date. Microsoft, prin aplicatia Microsoft SQL Server, pune la dispozite atat editii gratuite destinate proiectelor de mici dimensiuni, sau utilizarii in scop didactic, cat si editii specializate destinate aplicatiilor complexe, utilizate de un numar impresionant de utilizatori.[[17]](#footnote-17)

Printre editiile aplicatiei amintim:

* Enterprise:

SQL Server Enterprise Edition include atat functionalitatile de baza ale aplicatiei cât și serviciile suplimentare, cu o serie de instrumente pentru crearea și gestionarea unui cluster SQL Server. Poate gestiona bazele de date de până la 524 printr-un maxim de 640 de procesoare(nuclee de procesor).

* Standard:

SQL Server Standard include functionalitatile de baza ale aplicatiei. Acesta diferă de ediția Enterprise prin faptul că acceptă mai puține instanțe active (număr de noduri dintr-un cluster) și nu include unele funcții, cum ar fi aceea care permite adăugarea memoriei în timp ce serverul este încă în funcțiune și indecși paraleli.

* Web:

SQL Server Web este o editie cu functionalitati limitate, cu costuri totale reduse pentru cei care gazduiesc pagini Web și VAP-urile Web care ofera functionalitati de scalabilitate, accesibilitate și gestionare pentru proprietăți Web de mici dimensiuni.

* Developer:

SQL Server Developer include aceleași caracteristici ca SQL Server Enterprise, dar este limitat de licență pentru a fi utilizat doar ca sistem de dezvoltare și testare, și nu ca server de producție. Începând cu începutul anului 2016, Microsoft a făcut această ediție gratuită publicului.

* Express:

SQL Server Express este o editie care nu are limitări cu privire la numărul de baze de date sau utilizatori suportați, dar este limitat din punct de vedere al resurselor la utilizarea unui singur procesor, 1 GB de memorie și 10 GB fișiere de baze de date. Acesta este oferit gratuit de catre Microsoft.

Microsoft incepe dezvoltarea Microsoft SQL Server in anul 1986 in urma unei colaborari cu Sybase, Inc[[18]](#footnote-18), companie ce dezvoltase deja un sistem de gestinare al bazelor de date inca din anul 1984, iar in anul 1989 anunta intrarea Microsoft pe piața bazelor de date relaționale prin produsul Microsoft SQL Server 1.0, care trebuia sa concureze cu Oracle si IBM.

In anul 1993 parteneriatul dintre Sybase si Microsoft se incheie. Microsoft primieste o copie a codului sursa SQL Server, dezvoltat in parteneriat pana in acel moment, iar Sybase primeste posibilitatea de a dezvolta gratuit pe platforma x86. In acel moment Sybase SQL Server 4.2 si Microsoft SQL Server erau identice. Amblele aplicatii imparteau acelasi limbaj de programare numit T-SQL (Transact-SQL) si aceeasi arhitectura. Aplicatiile incep ulterior sa se diferentieze, Microsoft adaugand o serie de functionalitati Windows, iar Sybase adaugand functii destinate mediului de afaceri, acestia schimbandu-si ulterior numele în Adaptive Server Enterprise, pentru a evita confuzia cu Microsoft SQL Server

**Versiunea SQL Server 7.0** a adus modificari majore aplicatiei celor de la Microsoft, prin rescrierea partiala utilizand limbajul C++ a vechii aplicatii care era scrisa in limbajul C, adaugand printe altele o modalitate proprie de gestionare a firelor de executie si o interfata grafica.

**SQL Server 2000** a inclus mai multe modificări și extensii la codul sursa al celor de la Sybase, adăugând suport pentru arhitectura IA-64 . SQL Server 2000 a introdus, de asemenea, multe îmbunătățiri de limbaj T-SQL, cum ar fi variabile de tabel, funcții definite de utilizator, vizualizări indexate, declanșatoare INSTEAD OF, constrângeri referențiale în cascadă și suport de bază pentru limbajul XML.

In **versiunea SQL Server 2005**, codul Sybase a fost complet rescris. SQL Server 2005 a fost lansat în noiembrie 2005. Acesta a inclus suport nativ pentru gestionarea datelor XML, in concomitent cu datele relaționale. În acest scop, a definit un tip de date numit xml care ar putea fi utilizat fie ca tip de date în coloanele bazei de date, fie ca literal în interogări. Coloanele XML puteau fi asociate cu scheme XSD. XML este convertit într-un tip intern de date binare înainte de a fi stocat în baza de date. Pentru datele XML au fost disponibile metode de indexare specializate. Datele XML sunt solicitate folosind XQuery. SQL Server 2005 a adăugat câteva extensii la limbajul T-SQL pentru a permite încorporarea interogărilor XQuery în T-SQL. De asemenea, acesta a definit o nouă extensie la XQuery, numită XML DML, care permite modificări bazate pe interogare la datele XML. SQL Server 2005 permite, de asemenea, ca un server de baze de date să fie expus peste serviciile web folosind pachete de date Tabular Data Stream (TDS) încapsulate în cererile SOAP (protocol). Când datele sunt accesate prin intermediul serviciilor web, rezultatele sunt returnate sub forma XML.

Integrarea Common Language Runtime (CLR) a fost introdusă cu această versiune, permițând scrierea de cod SQL ca un cod gestionat de CLR. Pentru date relaționale, T-SQL au fost adaugate caracteristici de gestionare a erorilor (try / catch) și suport pentru interogări recursive cu CTEs (Common Table Expressions). De asemenea, SQL Server 2005 a fost îmbunătățit cu noi algoritmi de indexare si sintaxă. Permisiunile și controlul accesului au fost făcute mai granulare, iar procesatorul de interogare gestionează executarea simultană a interogărilor într-un mod mai eficient. Partițiile din tabele și indexuri sunt acceptate nativ, astfel încât extinderea unei baze de date într-un cluster este mai ușoară. SQL CLR a fost introdus cu SQL Server 2005 pentru a-i permite să se integreze cu .NET Framework.

SQL Server Management Studio (SSMS) este o aplicație software lansată pentru prima dată cu Microsoft SQL Server 2005 care este utilizată pentru configurarea, administrarea tuturor componentelor din Microsoft SQL Server printr-o noua interfata grafica. Acesta este succesorul aplicatiei Enterprise Manager din verisunea SQL 2000 sau versiunile anterioare anterior. Aplicatia include atât editori de script, cât și instrumente grafice care funcționează cu obiecte și caracteristici ale serverului.

**SQL Server 2008** a fost lansat pe 6 august 2008 și își propunea să facă managementul datelor mult mai audomatizat prin dezvoltarea tehnologiilor *SQL Server Always On*, pentru a nu exista perioade de nefunctionare a serviciului. SQL Server 2008 include, de asemenea, suport pentru date structurate și semi-structurate, inclusiv formate media digitale pentru imagini, audio, video și alte date multimedia. În versiunile curente, astfel de date multimedia pot fi stocate ca formatiuni de tip BLOB (obiecte mari binare).

Microsoft SQL Server 2008 a adus o mai bună asistență pentru date nestructurate și semi-structurate folosind noul tip de date FILESTREAM, care poate fi utilizat pentru a face referire la orice fișier stocat în sistemul de fișiere. Datele structurate și metadatele sunt stocate în baza de date SQL Server, în timp ce componentele nestructurate sunt stocate în sistemul de fișiere. Astfel de fișiere pot fi accesate atât prin intermediul API-urilor de gestionare a fișierelor Win32, cât și prin SQL Server folosind T-SQL, cele din urma accesand datele fișierului ca BLOB.

SQL Server 2008 include funcții de compresie mai bune, care ajută la îmbunătățirea scalabilității. Acesta a îmbunătățit algoritmii de indexare și a introdus noțiunea de indexuri filtrate. De asemenea, include Resource Governor care permite rezervarea resurselor pentru anumiți utilizatori sau functionalitati pentru imbunatatirea fluxuri de lucru si capabilități pentru criptarea transparentă a datelor (TDE).

SQL Server 2008 introduce pentru prima oara ADO.NET Entity Framework și instrumentele de raportare, replicarea și definirea datelor vor fi construite în jurul modelului de date al *entității*.

Versiunea din 2008 a aplicatiei celor de la Microsoft vine cu o noua versiune SQL Server Management Studio care acceptă IntelliSense pentru interogări. SQL Server 2008 face, de asemenea, bazele de date disponibile prin intermediul Windows PowerShell, astfel încât serverul și toate instanțele în curs să poată fi gestionate prin linie de comanda.

**SQL Server 2012** a fost lansat la fabricare pe 6 martie 2012. Acesta urmeaza sa fie ultima versiune care acceptă nativ OLE DB și, în schimb, preferă ODBC pentru conectivitate nativă. Printre noile caracteristici și îmbunătățiri ale SQL Server 2012 includem un set de opțiuni destinat îmbunătățirii disponibilitatea bazelor de date, îmbunătățiri ale programabilității, performanței dar si de securitate, inclusiv în timpul configurararii, permisiuni noi, gestionarea îmbunătățită a rolurilor și atribuirea schemei implicite pentru grupuri.

**SQL Server 2014** a fost lansat pentru publicul larg pe 1 aprilie 2014. Aceasta a fost disponibila in 3 versiuni: versiunea destinata Microsoft Azure, SQL Server 2014 CAB și SQL Server 2014 ISO . SQL Server 2014 a fist ultima versiune disponibilă pe arhitectura x86 / IA32.

**SQL Server 2016** a fost lansata pe 1 iunie 2016, iar disponibilitatea a fost redusa doar procesoare x64. Procesoarele x86 nu mai erau compatibile cu SQL Server 2016.

Microsoft a lansat **SQL Server 2017** pe 2 octombrie 2017 ,aplicatie care avea suport pentru sistemul de operare Linux.

Ultima versiune apare pe 4 noiembrie 2019 si se numeste **SQL Server 2019**.[[19]](#footnote-19)

**Entity Framework**

Entity Framework este un set de tehnologii din ADO.NET care sprijină dezvoltarea aplicațiilor software orientate către date. Arhitecții și dezvoltatorii de aplicații orientate către date s-au luptat de obicei cu necesitatea de a atinge două obiective foarte diferite. Ei trebuie să modeleze entitățile, relațiile și logica problemelor de afaceri pe care le rezolvă și, de asemenea, trebuie să lucreze cu baze de date, pentru stocarea și preluarea datelor. Datele pot cuprinde mai multe sisteme de stocare, fiecare cu propriile protocoale.

Entity Framework, care din versiunea Entity Framework 6 nu mai face parte din framework-ul .NET, permite dezvoltatorilor să lucreze cu date sub formă de obiecte și proprietăți specifice domeniului, cum ar fi clienții și adresele clienților, fără a fi nevoiți să se preocupe de tabelele și coloanele de bază ale bazei de date unde sunt stocate aceste date. Cu Entity Framework, dezvoltatorii pot lucra la un nivel mai înalt de abstractizare atunci când se ocupă de date și pot crea și menține aplicații care dispun de baze de date cu mai puțin cod si mai putine cunostinte legate de baze de date decât în ​​aplicațiile tradiționale.[[20]](#footnote-20)

Acest set de tehnologii ofera trei modalitati de abordare a crearii entitatilor unei baze de date, fiecare dintre acestea avand atuurile si slabiciunule lor. Cele trei abordari sunt:

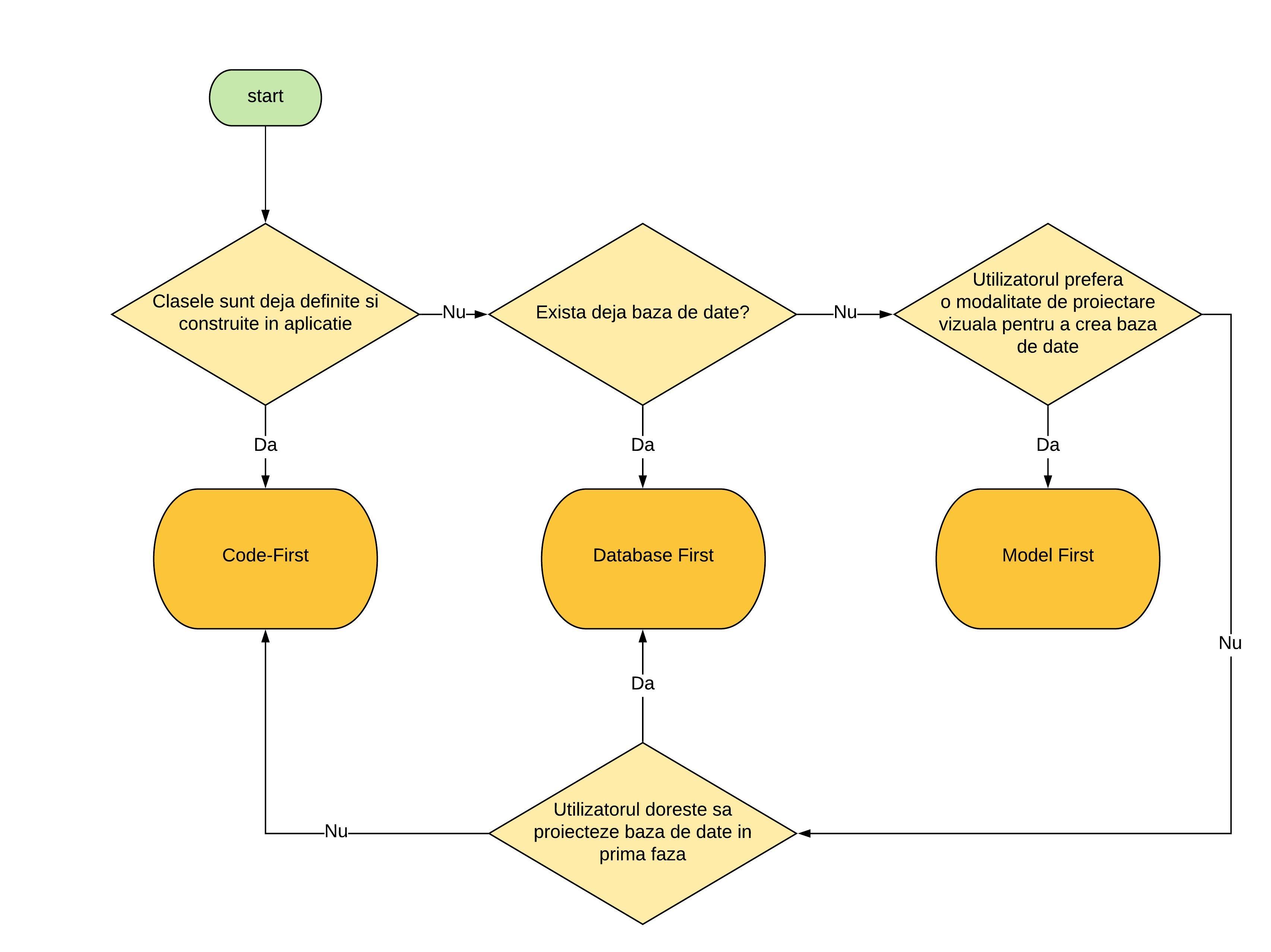
* Database First
* Model First
* Code First

Abordarea Database First reduce cantitatea de cod care necesara pentru a conecta baza de date la aplicatii si creaza, de asemenea, si entitatile care se mapeaza modelului bazei de date. Aceasta este cea mai populara abordare in momentul in care exista deja baza de date si pemite o modalitate simpla de actualizare in scenariul in care este modificata arhitectura bazei de date. Abordarea Database First nu este recomandata in proiecte cu arhitecturi complexe, deoarece fiind limitat din punct de vedere al presonalizarii, este dificil de intretinut pe termen lung.

In abordarea Model First, tabele si relatiile dintre acestea sunt create mai întâi folosind proiectantul ORM , care ofera o vizuala de a inlocui codul in scopul desenarii unei digrame, iar baza de date fizică urmeaza a fi generată folosind diagrama rezultata. Abordarea Model First rezulta in generarea de cod care va fi ulterior folosit crearii bazei de date de aplicatii. Acest model este dificil de abordat deoarece nu exista control intre diagrama si baza de date,iar in cazul modificarilor tabelelor din diagrama, acestea nu se sincronizeaza cu baza de date, fapt ce poate duce la pierderea datelor preexistente.

Abordarea Code First permite utilizatorului sa creeze modele și relația intre ele folosind clase și apoi să creeze baza de date din prin intermediul acestor clase. Aceasta abordare permite analiza structurii bazei de date într-o manieră orientată spre obiect. Aceasta abordare permite un control total asupra modelului bazei de date prin intermediul codului, fara o interfata intermediara de proiectare. Controlul codului face aceasta abordare sa fie cea mai personalizabila.

Fiecare dintre aceste abordari are avantaje si dezavantaje, importanta fiind alegere celei mai potrivite abordari in functie de resursele detinute in momentul alegerii, dar si de preferintele utilizatorilor.



Figură : Diagrama alegere abordare Entity Framework

Prima versiune a Entity Framework a fost inclusă odata cu Visual Studio 2008 lansat la data de 11 august 2008. Această versiune a fost criticată pe larg, atrăgând un val de critici si nemultumiri.

A doua versiune a Entity Framework a aparut la data te 12 aprilie 2010, si a fost numită Entity Framework 4.0. Aceasta aborda o serie de aspecte criticate in prima versiune.

O a treia versiune a Entity Framework, versiunea 4.1, a fost lansată pe 12 aprilie 2011, cu suportul pentru abordarea Code First. O actualizare a versiunii 4.1, denumită Entity Framework 4.1 a fost lansata ulterior, la putin timp distanta. Acesta corecta o serie de erori aparute in versiunea curenta.

Versiunea 4.3.1 a fost lansată pe 29 februarie 2012. Versiunea aceasta includea suportul pentru migrare.

Versiunea 5.0.0 a fost lansată pe 11 august 2012 și fiind destinata .NET Framework 4.5. De asemenea, disponibilitatea pentru .Net Framework 4 nu a fost afectata, dar au existat mici limitari fata de utlizarea impreuna cu .NET Framwork 4.5.

Versiunea 6.0 a fost lansată pe 17 octombrie 2013, creând oportunitatea comunitatii sa ajute la dezvoltare realizarea proiectului ca fiind Open Source licențiat sub Apache License v2. Ca și ASP.NET MVC, codul sursă este găzduit de platforma GitHub folosind Git. Această versiune are o serie de îmbunătățiri pentru suportul Code First.

Microsoft a decis apoi să aducă platforma .NET pentru Linux, OSX și în alte părți, ceea ce duce la o rescriere completa a următoarei versiuni de Entity. La 27 iunie 2016, aceasta a fost lansată ca Entity Framework Core 1.0, alături de ASP.NET Core 1.0 și .NET Core 1.0. Inițial a fost numit Entity Framework 7, dar a fost redenumit pentru a evidenția faptul că era o rescriere completă, mai degrabă decât o actualizare incrementală și nu o inlocuire a Entity Framework 6.

EF Core 1.0 este licențiat sub Apache License v2 și este construit în întregime în mod deschis pe GitHub. În timp ce EF Core 1.0 împărtășește unele asemănări conceptuale cu versiunile anterioare ale Entity Framework, este o bază de cod complet nouă, concepută pentru a fi mai eficientă, mai puternică, flexibilă și extensibilă, putand fi rulata pe Windows, Linux și OSX și sustinand o nouă gamă de depozite de date relaționale și NOSQL.

EF Core 2.0 a fost lansat pe 14 august 2017 împreună cu Visual Studio 2017 15.3 odata cu ASP.NET Core 2.0

EF Core 3.0 a fost lansat pe 23 septembrie 2019 împreună cu Visual Studio 2019 16.3 și ASP.NET Core 3.0.

## 3.5 HTML - CSS – JavaScript

**HTML** (acronim de la HyperText Markup Language) este o modalitate de a structura si prezenta vizual informatii prin intermediul unei pagini dintr-un browser web. Aceasta este prezentat ca fiind un limbaj de marcare, deoarece contine elemente distinctive semantic de textul propriu-zis care nu urmeaza a fi afisate. Acestea au ca scop formatarea textelor, si nu numai, pentru a fi afisate intr-o maniera personalizata.

Tim Berners-Lee a reusit sa puna bazele internetului pe care il utilizam in zilele noastre printr-o serie de concepte fundamentale printre care si limbajul HTML in 1990.

Elementele distinctive utlizate pentru a marca o portiune de cod HTML sunt numite etichete si sunt cuprinse intre simbolurile „<” si „>”.

Printre cele mai utlizate etichete amintim:

* Titlu: <title> - eticheta utilizata pentru definirea unui titlu in limbajul HTML
* Paragraf: <p> - eticheta utilizata pentru definirea unui paragraf in limbajul HTML
* Header <h1>,<h2>,<h3>,<h4>,<h5>,<h6> - eticheta utilizata pentru definirea unui antet in limbajul HTML. Acest antet poate fi definit in 6 dimensiunui, <h1> reprezentand dimensiunea maxima, iar <h6> dimensiunea minima.
* Ancora <a href=”…”> - Eticheta utilizata pentru a crea un link catre o pagina web.
* Liste ordonate <ol> si neordonate <ul> - Etichete utlizata pentru a crea liste. Pentru a adauga elemente in aceste liste se utilizeaza eticheta <li>
* Tabel <table> - Eticheta utilizata pentru a crea un tabel intr-o pagina web. Acesta este imbrica eticheta <tr> (rand tabel) care la randul ei imbrica etichetele <td>(data tabel) si <th>(antet tabel).
* Selectie/Diviziune <div> Eticheta utilizata la incapsula altor etichete pentru a fi utilizate ca un intreg.

Majoritatea elementelor de tip HTML sunt de forma <nume\_eticheta> continut </nume\_eticheta>, dar exista o serie de elemente de forma <nume\_eticheta> cum ar fi:

* Rupere rand <br> - Eticheta utilizata pentru spatierea cu un rand a continutului ce o urmeaza, fata de continutul ce o preceda
* Linie orizontala <hr> - Eticheta utilizata pentru spatierea continutului ce urmeaza, fata de continutul ce o preceda, printr-o linie orizontala
* Imagine <img scr=”…”> Eticheta utilizata pentru adaugrarea unei imagini

Urmatorul exemplu prezinta codul unei pagini HTML, urmat de rularea acestuia intr-o pagina web:

Tabel : Exemplu utilizare HTML

|  |  |
| --- | --- |
| 1. **<html>** 2. **<body>** 3. **<h1>**Numele meu este**</h1>** 4. **<h2>**Tereanu Bogdan**</h2>** 5. **<p>**Exemplu paragraf**</p>** 6. **</body>** 7. **</html>** | C:\Users\Boogie\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ex.png |

Elementele HTML, pe langa etichete, dispun de o serie de, numite atribute. Atributele sunt proprietati optionale localizeazate in interiorul elementelor HTML. Daca o eticheta nu are nici un atribut, atunci browser-ul va lua in considerare valorile implicite ale respectivei etichete.

Atributele se integraza in elementele HTML astfel:

<eticheta **nume\_atribut=”valoare”**></ eticheta >

Atributele pot fi folosite pentru schimbarea culorilor elementelor sau fundalului, aliniere, ascundere, creare sau modificare de margini, redimensionare si multe altele. Printre cele mai cunoscute atribute amintim:

* bgcolor – Atribut utilizat pentru schimbarea culorii fundalului in etichetele precum <body>, <table>, <td>, <th>.
* align – Atribut utlizat pentru alinierea textului. Acesta poate primi una din urmatoarele valori: left,right,center,justify.
* href – Atribut utilizat in interiorul etichetei <a> pentru a specifica un link
* scr – Atribut utilizat in interiorul etichetei <img> pentru a specifica adresa imaginii
* alt – Atribut care seteaza un text, ca alternativa in cazul in care o imagine nu poate fi randata
* width si height – Atribute utilizate pentru a seta dimensiunile unei etichete
* style – Atribut utilizat pentru stilizarea unui element prin intermediul culorilor, dimensiunii, fontului, etc.

**CSS** (acronim de la Cascading Style Sheets) este un limbaj de stilizare al paginilor web care extinde posibilitatile de modificare a elementelor intr-un limbaj de marcare cum ar fi HTML. Acesta vine ca un suport pentru limbajul HTML, principalele imbunatatiri aduse fiind îmbunătățirea accesibilitatii conținutului, flexibilitate și control în specificarea caracteristicilor de prezentare, permiterea mai multor pagini web să partajeze formatarea prin specificarea unei referinte catre un fișier cu extensia .css separat, fapt ce conduce la reduce complexitatea și redundantei conținutului.

CSS a fost propus pentru prima dată de Håkon Wium Lie la 10 octombrie 1994. La acea vreme Lie lucra impreuna cu inventatorul limbajului HTML, Tim Berners-Lee, la compania CERN.

Sintaxa limbajului este simplista, formata dintr-o serie de cuvinte intuitive in limba engleza, urmate de caracteristicile sau valorile acestora. Acestea pot fi numite de asemenea, reguli si sunt dispuse in interiorul unor etichete ale limbajului care au scop de selectatori.

Selectatorii pot fi orice etichete ale limbajului HTML, etichete care au unul din atributele *id*(identificator unic de element) sau *class*(identificator multiplu de element) si etichete HTML raportate la pozitia lor in raport cu alte etichete.

Pentru a integra CSS intr-o pagina HTML exista trei modalitati:

* Integrare externa – se realizeaza prin intermediul unui fisier cu extensia .css asociat paginii HTML
* Integrare interna – se realizeaza prin itermediul etichetei *<style>*, acesta fiind destinata pentru a interpreta codul interior ca fiind CSS.
* Integrare in linie sau in eticheta – se realizeaza prin intermediul atributului *style* in interiorul etichetei. Continutul atributului va fi tratat ca si cod CSS aplicat etichetei.

Integrarea externa clasica a unui fisier CSS in pagina HTML se relizeaza prin intermediul urmatoarei etichete HTML introdusa in interiorul partii *head* a fisierului:

<**link** href="path/to/file.css" rel="stylesheet" type="text/css">

Urmatorul exemplu prezinta modificari aduse la Figura 6 utilizand CSS prin intermediul modalitatilor prezentate anterior:

Tabel : Exemplu utilizare CSS

|  |  |  |
| --- | --- | --- |
| Integrarea interna | Integrare externa | Integrare in eticheta |
| 1. **<html>** 2. **<head>** 3. **<style>** 4. h1{ 5. color:red; 6. } 7. h2{ 8. font-style: italic; 9. } 10. p{ 11. background: cyan; 12. } 13. **</style>** 14. **</head>** 15. **<body>** 16. **<h1>**Numele meu este**</h1>** 17. **<h2>**Tereanu Bogdan**</h2>** 18. **<p>**Exemplu paragraf**</p>** 19. **</body>** 20. **</html>** | Codul urmator va fi situat intr-un fisier extern cu extensia .css   1. h1{color:red;} 2. h2{ 3. font-style: italic; 4. } 5. p{background: cyan;}   Iar modificarea fata de codul din Figura 6 este introducerea codului urmator in intre etichetele html si head   1. **<head>** 2. **<link** href="../exemplu.css" rel="stylesheet"  type="text/css"**>** 3. **<head>** | 1. **<html>** 2. **<body>** 3. **<h1** style="color:red;"**>**   Numele meu este**</h1>**   1. **<h2** style="font-style: italic;"**>**   Tereanu Bogdan**</h2>**   1. **<p** style="background:cyan;"**>**   Exemplu paragraf**</p>**   1. **</body>** 2. **</html>** |
| Rezultatul va fi: | | |
|  | | |

**JavaScript** este un limbaj de programare utilizat in general in dezvolatare de pagini web. Acesta confera dinamicitate paginilor, largind orizonturile posibilitatilor oferite de HTML si CSS prin programarea la nivel de client. Codul JavaScript este rulat de browser odata ce pagina web este incarcata, fara a comunica cu serverul.

JavaScript alaturi de CSS si HTML reprezinta tehnologiile de baza ale dezvoltarii unei pagini web.

La momentul lansarii, in 1995 limbajul se numea Mocha, fiind creat de unul din angajatii comaniei Netscape Communications numit Brendan Eich. Acesta s-a inspirat din limbaje precum Java, Scheme si Self care la acea vreme aveau o popularitate ridicata. La acea vreme, compania Netscape se bucura de dominanta la capitolul browsere web, iar dorinta acestora era sa creeze pagini web mult mai dinamice si interactive. Limbajul era, mai degraba destinat proiectantilor grafici de pagini web, fapt ce a dus la o sintaxa mai scurta, mai simplista si mai usor de implementat decat limbajele de programare clasice ca Java si C.

Fiind un limbaj de programare destinat interactiunii utilizatorului cu o pagina web dinamica, acesta este utilizat pentru a realiza si trata o serie de evenimente fundamentale printre care includem:

* *onClick -* Evenimentul are loc atunci când utilizatorul face clic pe un element.
* *onMouseDown –Evenimentul are loc atunci cand utilizatorul*
* *onMouseUp,*
* *onKeyDown- Evenimentul are loc cand se apasa o tasta*
* *onKeyUp- Evenimentul are loc in momentul in care o tasta anterior apasata este eliberata*
* *onFocus- Evenimentul are loc atunci cand un obect este focalizat/in focus.*
* *onSubmit*- Evenimentul are loc atunci cand se realizeaza un apel catre server
* onLoad- Evenimentul are loc atunci cand se incarca pagina
* onChange- Evenimentul are loc atunci cand se intampla o schimbare de stare

În timp ce JavaScript standard este încă utilizat pentru îndeplinirea funcțiilor de bază pe partea clientului, popular pentru dezvoltatori a devenit utilizarea de biblioteci JavaScript, cum ar fi jQuery, pentru a adăuga elemente dinamice mai avansate pe site-urile web.

## 3.6 Razor View Engine in ASP.NET MVC

Razor View Engine este un limbaj de marcare care permite scrierea de cod HTML și cod C# sau VB.NET in aceeasi pagina web. Cu toate acestea, Razor View Engine nu face parte din framework-ul ASP.NET MVC.

Paginile rezultate, care reprezinta partea vizuala a aplicatiilor ASP.NET MVC au extensia .cshtml.

O parte dintre regulile de sintaxă Razor pentru limbajul C# sunt.

* Codul C# este incadrat in interiorul urmatoarei sintaxe @{...}
* Declarațiile se incheie cu „ ; ”
* Fișierele au extensie .cshtml.
* Variabilele sunt declarate prin cuvântul cheie var
* Expresiile inline (variabile și funcții) încep direct cu @ fara a mai fi nevoie de acolade
* Codul C# este case-sensitive( exista diferentiere intre majuscule si minuscule)

## 3.7 Google Charts

1. https://en.wikipedia.org/wiki/Small\_business [↑](#footnote-ref-1)
2. https://www.thinkific.com/blog/top-online-business-tools/ [↑](#footnote-ref-2)
3. Bruton, Noel, How to Manage the IT Helpdesk — A Guide for User Support and Call Center Managers. [↑](#footnote-ref-3)
4. <https://en.wikipedia.org/wiki/BMC_Software> [↑](#footnote-ref-4)
5. <https://en.wikipedia.org/wiki/Windows_API> [↑](#footnote-ref-5)
6. <https://en.wikipedia.org/wiki/Windows_Forms> [↑](#footnote-ref-6)
7. <https://en.wikipedia.org/wiki/Windows_Presentation_Foundation> [↑](#footnote-ref-7)
8. <https://en.wikipedia.org/wiki/Microsoft_Store_(digital)> [↑](#footnote-ref-8)
9. <https://en.wikipedia.org/wiki/Microsoft_Silverlight> [↑](#footnote-ref-9)
10. <https://en.wikipedia.org/wiki/Graphical_user_interface> [↑](#footnote-ref-10)
11. <https://en.wikipedia.org/wiki/Apache_Subversion> [↑](#footnote-ref-11)
12. <https://en.wikipedia.org/wiki/Git> [↑](#footnote-ref-12)
13. <https://en.wikipedia.org/wiki/Azure_DevOps_Server> [↑](#footnote-ref-13)
14. <https://en.wikipedia.org/wiki/Microsoft_Visual_Studio> [↑](#footnote-ref-14)
15. <https://docs.microsoft.com/en-us/visualstudio/ide/find-code-changes-and-other-history-with-codelens?view=vs-2019> [↑](#footnote-ref-15)
16. <https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-3.1> [↑](#footnote-ref-16)
17. <https://en.wikipedia.org/wiki/Microsoft_SQL_Server> [↑](#footnote-ref-17)
18. <https://en.wikipedia.org/wiki/Sybase> [↑](#footnote-ref-18)
19. <https://en.wikipedia.org/wiki/History_of_Microsoft_SQL_Server> [↑](#footnote-ref-19)
20. <https://en.wikipedia.org/wiki/Entity_Framework> [↑](#footnote-ref-20)