**Universitatea din Craiova**

**Facultatea de Automatică, Calculatoare și Electronică**

**Departamentul de Calculatoare și Tehnologia Informației**

**Planul Proiectului de Licenta**

**Aplicatie Web de gestiune a unei Sali de jocuri**

**Tipul proiectului: Cercetare si dezvoltare**

**Numele și semnătura studentului: Vilceanu Ionut-Alexandru**

**Numele și semnătura coordonatorului: Dr. ing. Popescu Elvira**

**Data: 26.12.2019**

**Anul universitar 2019-2020**

Cuprins

[1. Introducere 2](#_Toc28269900)

[2. Obiective 2](#_Toc28269901)

[3. Activități 4](#_Toc28269902)

[4. Mediul de cercetare / dezvoltare 5](#_Toc28269903)

[5. Livrările proiectului 6](#_Toc28269904)

[6. Planificarea activităților 6](#_Toc28269905)

[7. Bibliografie 7](#_Toc28269906)

# 1. Introducere

In ultimii ani, desi platformele de socializare au capatat o foarte mare amploare, in special tinerii si-au cautat si alte medii de socializare, medii in afara realitatii virtuale. Unul dintre acestea o reprezinta salile de jocuri, un loc unde oamenii pot sa-si dezvolte aptitudinile la un anumit joc, pot concura intre ei, unde este competitie fiind si distractie. Dat fiind faptul ca obisnuitii clienti ai acestor sali sunt familiarizati cu tehnologia, este evident ca online-ul reprezinta un aspect foarte important in derularea activitatii lor.

In cadrul acestui proiect de licenta ne propunem realizarea unei aplicatii web de gestiune a activitatilor dintr-o astfel de sala, care sa usureze munca tuturor, plecand de la angajati si sarcinile lor, pana la clienti, carora le este mult mai usor sa-si rezerve totul dinainte.

# 2. Obiective

**O1. Lizibitate si accesibilitate.**Pagina principala va avea un design inovativ, culori vii si un stil atractiv. Astfel, se vor putea accesa foarte usor meniurile principale, dar si meniul de Log In/Register.

**O2. Preferinte.** Dupa ce se va inregistra, fiecare Client(User) isi poate alege jocul preferat, selectand apoi data si ora la care vrea sa faca o rezervare. De asemenea, Clientul(User-ul) si poate alege singur pentru cat timp vrea sa faca o rezervare, iar daca masa/jocul sau favorit este ocupat/a, acesta va putea vizualiza ora la care acestea se elibereaza, dar si sa-si aleaga daca vrea sa ia o gustare, alegand si ora la care vrea ca aceasta sa fie pregatita.

**O3.Istoric al comenzilor.** Fiecare user/client va beneficia de un istoric al comenzilor,

datele fiind stocate.

**O4. Istoric al scorurilor**. Dupa fiecare meci de tenis/biliard/joc electronic, user-ul va avea un meniu in care isi va putea completa scorul. Aplicatia va avea un istoric al scorurilor tuturor partidelor de orice tip pentru fiecare user, astfel el putand sa-si imbunatateasca recordurile.

**O5.Tipul platii.** Dupa fiecare rezervare, clientul(user-ul) poate plati cu cardul in aplicatie, sau poate alege sa plateasca cash, in sala.

**O6.Recomandari(Inteligenta Artificiala).** Aplicatia va beneficia si de inteligenta artificiala. Astfel, recomandarile vor fi facute pe baza similaritatilor intre utilizatori(collaborative filtering), clientul fiind ajutat sa-si gaseasca mai repede jocul/activitatea dorita. Acelasi lucru se va intampla si cu mancarea, fiecare User avand recomandari variate care-i pot face alegerea mai usoara.

**O7. Disponibilitatea stocului.** Userii vor stii exact ce feluri de mancare/bautura sunt disponibile pe stoc, dar si cele pentru care exista un stoc limitat, neexistand posibilitatea de a comanda in aplicatie ceva ce nu exista in sala.

**O8. Distributia rolurilor.** Conturile aplicatiei vor avea 3 roluri: User(clientii obisnuiti, oricine poate deveni User prin inregistrare), Admin(singurul care va avea acces la parolele si datele personale ale Userilor) si Angajat(angajatii salii de jocuri). Angajatii vor avea conturi speciale, acestia facand update-uri in timp real cu situatia meselor/jocurilor ocupate, dar si a stocului de produse. Administratorii vor fi singurii care pot modifica/vedea anumite date personale, date de contact, precum si date financiare.

**O9. Evaluare si reclamatii.** Dupa fiecare rezervare, clientul va avea posibilitatea de evaluare a salii de jocuri si personalului. Angajatii vor avea de asemenea functia de reclamatie catre un anumit client prin intermediul aplicatiei. Aceasta reclamatie va ajunge la Administrator care o va analiza si in functie de gravitatea celor intamplate va decide daca clientul trebuie interzis in sala si contul sau blocat/sters. La fel se va intampla si cu angajatii, care, daca vor avea o medie a review-urilor slaba, vor putea fi sanctionati sau dati afara si implicit, conturile lor sa fie blocate/sterse.

**O10. Sectiunea de joburi.** Pagina va avea si sectiunea de joburi. Aici Administratorii vor posta anuminte joburi, la care oricine poate aplica apasand pe butonul „Apply”. Dupa apasarea butonului, se va deschide un formular, pe care potentialul angajat trebuie sa-l completeze. Oricine poate aplica pentru joburi, nefiind nevoie de cont pentru a intra in acest meniu.

# 

# 3. Activități

.

**A1. Crearea la nivel de concept a unei baze de date.** Aplicatia fiind „Code First”, primul lucru de care are nevoie este o baza de date de inceput, conceptuala, aceasta putand fi modificata pe parcurs in functie de ce avem nevoie.

**A2. Realizarea modelelor aplicatiei.** Bazandu-ne pe modelul arhitectural Model-View-Controller(MVC), primele care vor fi create, vor fi modelele aplicatiei. Acestea vor fi realizate conform bazei de date concepute mai sus.

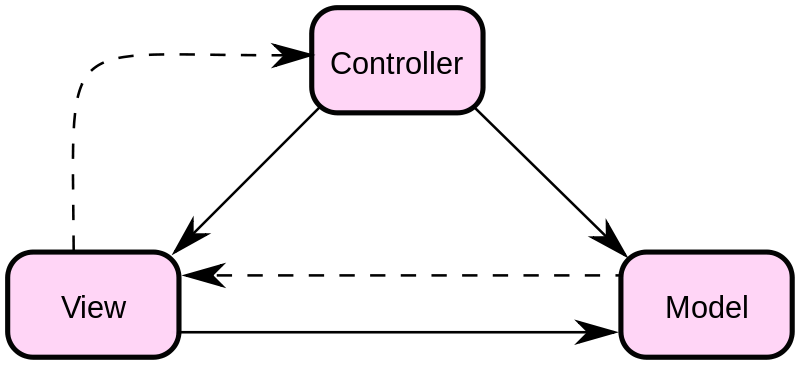


Fig.1 Modelul arhitectural Model-View-Controller

**A3. Controllere si View-uri.** Pe baza modelelor realizate, vom crea Controller-ele, pentru fiecare dintre acestea urmand sa fie actiuni predefinite(precum Index, Create, Delete, Edit), dar si actiuni noi, care urmeaza a fi implementate. Fiecare actiune va avea cate un View.

**A4. Interfata aplicatiei.** Dupa ce am terminat partea de Back-End a aplicatiei, urmeaza partea de Front-End. Aici vom concepe si implementa interfata aplicatiei, partea cu care utilizatorii vor interactiona(Fig. 2).

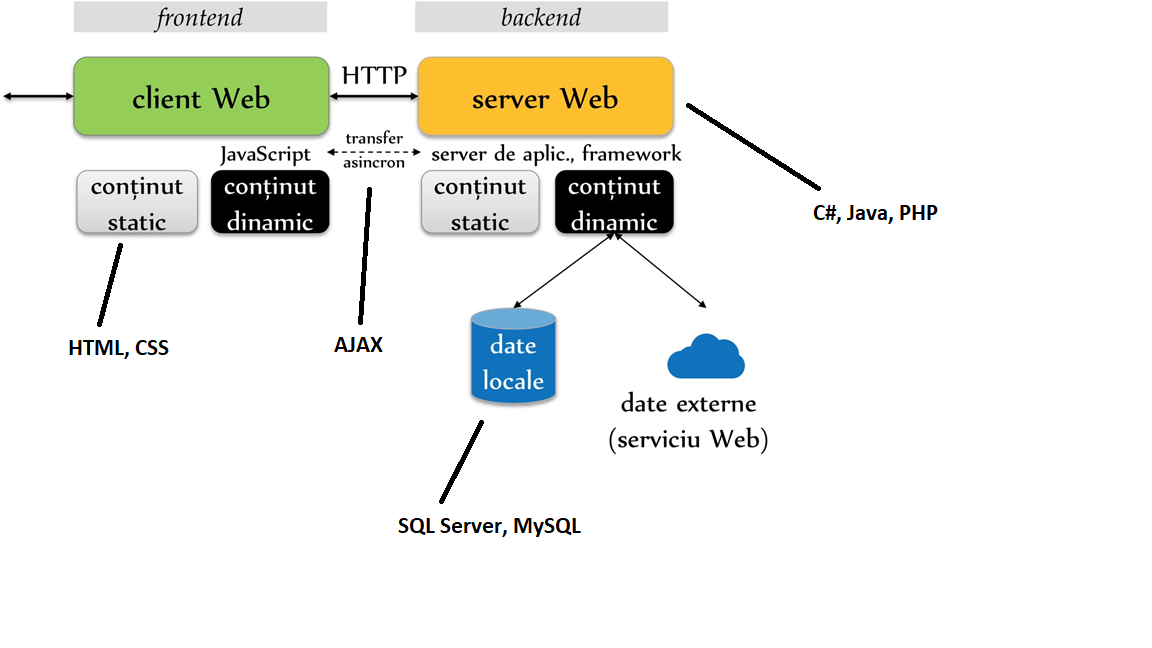


Fig.2 Arhitectura aplicatiilor Web

**A5. Versiunea finala.** Dupa ce finalizam procesele de mai sus, dar si anumite teste, vom ajunge la versiunea finala a aplicatiei, aceasta fiind acum functionala.

**A6. Redactarea unui raport final.** Ca rezultat al activitatii de cercetare/implementare se va incerca redactarea unei lucrari care sa cumuleze rezultatele obtinute, dar si pasii urmati in directia implementarii aplicatiei.

# 4. Mediul de cercetare / dezvoltare

Partea aplicativa a proiectului de diploma se va realiza folosind limbajul **ASP .NET** si integrand diverse librarii pentru implementarea functionalitatilor dorite. Ca IDE pentru realizarea proiectului se va folosi Visual Studio.

ASP.NET este o tehnologie Microsoft pentru crearea de aplicații web și servicii web. ASP.NET este succesorul lui ASP și beneficiază de puterea platformei de dezvoltare .NET, și de setul de instrumente oferite de mediul de dezvoltarea al aplicației „Visual Studio .NET”. Aceasta a fost lansata initial in anul 2002, fiind compatibila cu sisteme de operare precum Microsoft Windows, Linux sau Mac OS X.

**Visual Studio** include un set complet de instrumente de dezvoltare pentru generarea de aplicații ASP.NET, Servicii Web XML, aplicații desktop și aplicații mobile. Visual Basic, Visual C++, Visual C# și Visual J# toate folosesc același mediu de dezvoltare integrat (IDE) care le permite partajarea instrumentelor și facilitează crearea de soluții folosind mai multe limbaje de programare. Aceste limbaje permit să beneficieze de caracteristicile .NET Framework care oferă acces la tehnologii cheie care simplifica dezvoltarea de aplicații web ASP și XML Web Services cu Visual Web Developer.

# 5. Livrările proiectului

Ca livrari ale proiectului avem:

L1. **Codul sursa.**

L2. **Raportul final.** O lucrare care sa cumuleze rezultatele obtinute, dar si pasii urmati in directia implementarii aplicatiei.

L3. **Manualul de utilizare.** Un mic rezumat care va imbunatati interactiunea utilizatorilor cu aplicatia.

# 6. Planificarea activităților

Mai jos este prezentat tabelul in care este specificata organizarea actvitatilor in functie de luna si numarul de ore alocat

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2020 | Ianuarie | | | | Februarie | | | | Martie | | | | Aprilie | | | | Mai | | | | Iunie | | | | Total |
| Activitati | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |  |
| Cercetare bibliografica | 4 | 4 | 6 | 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 20 |
| Planul Proiectului |  |  | 5 | 5 | 5 | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 20 |
| Cerinte |  |  |  |  |  |  | 4 | 4 | 4 | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 17 |
| Implementare funcționalități |  |  |  |  |  |  |  |  |  | 8 | 8 | 8 | 8 | 8 |  |  |  |  |  |  |  |  |  |  | 32 |
| Cod si teste unitare |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 | 6 | 6 | 6 |  |  |  |  |  |  | 24 |
| Integrare si teste |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 | 4 | 4 | 4 |  |  |  |  | 16 |
| Raportul final |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 | 5 |  |  |  | 10 |
| Prezentare Finala |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 8 | 8 |
| Total | 4 | 4 | 11 | 11 | 5 | 5 | 4 | 4 | 4 | 13 | 8 | 8 | 8 | 8 | 6 | 6 | 10 | 10 | 4 | 9 | 5 |  |  | 8 | 127 |

# 7. Bibliografie

1. <https://ro.wikipedia.org/wiki/Microsoft>
2. <https://ro.wikipedia.org/wiki/Microsoft_Visual_Studio>
3. <https://ro.wikipedia.org/wiki/ASP.NET>