



SITE PENTRU COMERT ELECTRONIC

Moldovan Adelina-Stefania

Vorniceanu Iuliana-Georgiana

Grupa 30236

CUPRINS

1. Specificatii si analiza sistemului	3
2. Precizarea limbajului de programare ales	4
3. Fundamente teoretice	5
4. Proiectarea aplicatiei	7
4.1 Descrierea arhitecturii sistemului	9
4.2. Identificarea functionalitatilor aplicatiei	11
4.3. Diagrama de interactiune: activitate, secventa, colaborare	12
4.4. Detalii de implementare: diagrama bazei de date	15
4.5. Cazuri de testare	16
5. Testarea propriu zisa	17
6. Manual de instalare si utilizare	20
7. Concluzii si dezvoltari ulterioare	21
8. Bibliografie.	22

1. Specificatii si analiza sistemului

Cerinta proiectului:

Construiti un site pentru comert electronic: utilizatorul trebuie sa fie capabil sa viziteze site-ul si sa vizualizeze toate produsele, sa caute un anumit produs, respectiv categorie de produse, sa comande anumite produse (ulterior punerii acestora in cos), sa-si puna propriile produse pe site in calitate de provider.

Analiza si specificarea au obiective diferite:

- Analiza pune accent pe intelegerea structurii problemei.
- Specificarea arata ce trebuie sa faca sistemul, pune accent pe comportamentul exterior.
-

Această fază definește cerințele sistemului, independent de modul în care acestea vor fi îndeplinite. Aici se definește problema pe care clientul dorește să o rezolve. Rezultatul acestei faze este documentul cerințelor, care trebuie să precizeze clar ce trebuie construit.

Documentul încearcă să redea cerințele din perspectiva clientului, definind scopurile și interacțiunile la un nivel descriptiv înalt, independent de detaliile de implementare, cum ar fi, de exemplu: formularea problemei, așteptările clientului sau criteriile pe care trebuie să le îndeplinească produsul.

Faza de analiză poate fi văzută ca o rafinare a detaliilor. Distincția dintre detaliile de nivel înalt și cele de nivel scăzut sunt puse mai bine în evidență de abordările top-down (unde se merge către detaliile de nivel scăzut) și bottom-up (care tind către detaliile de nivel înalt).

Cerintele proiectului sunt foarte clare, trebuie sa implementam un site web in scopul comertului de diferite bunuri. Am ales ca bunurile respective sa fie produse electronice . Cum ni se prezinta in cerinta, clientul trebuie sa isi poata face cont pe site sau sa se autentifice, daca are deja unul. Poate cauta produse dupa nume sau categoria din care fac parte. Poate adauga oricate produse in cos, dupa care poate da comanda pentru produsele respective. O alta optiune este cea de a pune propriile produse la vanzare pe site.

2. Precizarea limbajului de programare ales

Ce inseamna Front-End(client) și ce inseamna Back-End (server) ?

Front-End:

- Implică tot ceea ce este vizibil pentru utilizatorul final în browser.
- Front-End-ul are două părți: design-ul (partea creativă) și dezvoltarea interfeței (partea de cod sau implementare HTML CSS). Putem remarca diferențierea între partea de design grafic (ce se vede efectiv) și partea de implementare (cum funcționează ce se vede, cu ajutorul HTML, CSS, JavaScript) .

Back-End:

- Reprezintă tot ceea ce utilizatorul final nu vede în aplicație: baza de date, server.
- Back-End-ul de obicei constă în trei părți: un server, o aplicație de interfață și o bază de date. Rolul principal al acestuia este cel de management de conținut.

Pentru dezvoltarea partii de Front-End am folosit urmatoarele limbaje:

HTML: este limbajul utilizat la scrierea paginilor web, el fiind mai mult un limbaj de formatare a textului decat limbaj de programare. Acest limbaj se foloseste de anumite marcaje numite TAG-uri pentru a preciza navigatrolui cum sa interpreteze textul din cadrul lor.

CSS: este un limbaj de stilizare al elementelor html(al tagurilor html). Acest lucru inseamna ca de la culoarea literelor si a backgroundului pana si la positionarea elementelor de pe o pagina web, totul este stilizat prin CSS.

JavaScript: este un limbaj de programare orietant obiect bazat pe conceptul prototipurilor. Este folosit mai ales pentru introducerea unor functionalitati in paginile web, codul JavaScript din aceste pagini fiind rulat de către browser.

Pentru dezvoltarea partii de Back-End am folosit urmatoarele limbaje:

PHP: este un limbaj de uz general, cu cod-sursa deschis (open source), utilizat pe scara larga si care este potrivit in special pentru dezvoltarea aplicatiilor web si poate fi integrat in HTML.

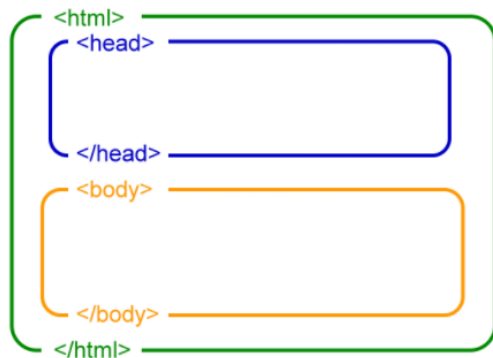
SQL: este un limbaj de programare folosit pentru lucrul și comunicarea cu bazele de date. Prin acest limbaj poți să modifichi tabele și structuri de index din baze de date, să adaugi, să editezi sau să ștergi linii de date și să recuperezi sub-seturi dintr-o bază de date.

3. Fundamente teoretice

Limbajul folosit pentru scrierea paginilor web este **HTML** (HyperText Markup Language). Acesta nu este un limbaj de programare, ci de "formatare" a textului, si foloseste asa numitele marcate, numite TAG-uri in limba engleza. Paginile HTML sunt formate din etichete sau tag-uri și au extensia „.html” sau „.htm”.

Tagurile apar intre semnele < si >, si reprezinta comenzi ce spun navigatorului cum sa interpreteze textul dintre taguri (bold, italic, daca se defineste un tabel, o lista, inserarea unei imagini, etc.) si astfel poate afisa corespunzator pagina web.

Majoritatea tagurilor au doua parti: **tagul de deschidere** ce indica unde incepe textul si **tagul de inchidere** ce spune navigatorului unde se termina textul ce trebuie formatat conform tagului respectiv. Tagurile de inchidere incep cu semnul / (slash) chiar dupa simbolul < . De exemplu:



In acest fel navigatorul stie ca tot ce se afla intre aceste tag-uri este continutul paginii web si il va afisa in fereastra browser-ului.

CSS (*Cascading Style Sheets*) sau **foi de stil în cascadă**, reprezintă un mecanism simplu de a adăuga elemente de stil (tipuri de caractere, culori, aliniere, spațiere etc.) unui document web.

Tag-urile HTML au avut inițial rolul de defini conținutul unui document, de a face distincția între titlul unui paragraf (definit prin tag-urile <Hn>), conținutul paragrafului (definit prin tag-ul <p>), tabele (definite prin tag-ul <table>) etc. Aspectul și modul de afișare al unui site cădeau în sarcina browserului, fără a fi folosite niciun fel de tag-uri de formatare.

Dar cum cele două principale browsere – Internet Explorer și Netscape Navigator – au continuat să adauge noi tag-uri și attribute limbajului HTML original, a devenit tot mai dificil să creezi un site în care conținutul documentului HTML să fie clar separat de modul său de prezentare.

Pentru a rezolva această problemă, consorțiul W3C, responsabil de standardizarea specificațiilor HTML, a creat **stilurile** și le-a adăugat specificațiilor HTML 4.0.

În prezent, toate browserele suportă CSS.

O construcție CSS are două părți: un selector, și una sau mai multe declarații:

```
selector {  
    proprietate_1 : valoare_1 ;  
    proprietate_2 : valoare_2 ;  
    .....  
    proprietate_n : valoare_n ;  
}
```

unde:

- **selector** - această componentă, identifică elementul HTML, căreia i se aplică stilul dat;
- **acoladele** - delimitează, care proprietăți și cu ce valori se aplică elementului HTML;
- **proprietăți** - sunt elementele de aspect care vrem să le modificăm;
- **valori** ale proprietăților - sunt noile valori pe care vrem să le atribuim proprietății respective.

JavaScript (JS) este un limbaj de programare orientat obiect bazat pe conceptul prototipurilor. Este folosit mai ales pentru introducerea unor funcționalități în paginile web, codul JavaScript din aceste pagini fiind rulat de către browser.

Programatorii web pot îngloba în paginile HTML script-uri pentru diverse activități cum ar fi verificarea datelor introduse de utilizatori sau crearea de meniuri și alte efecte animate.

Browselele rețin în memorie o reprezentare a unei pagini web sub forma unui arbore de obiecte și pun la dispoziție aceste obiecte script-urilor JavaScript, care le pot citi și manipula. Arborele de obiecte poartă numele de Document Object Model sau DOM. Există un standard W3C pentru DOM-ul pe care trebuie să îl pună la dispoziție un browser, ceea ce oferă premiza scrierii de script-uri portabile, care să funcționeze pe toate browserele.

PHP-ul este unul din cele mai folosite limbaje de programare server-side. Popularitatea de care se bucură acest limbaj de programare se datorează următoarelor caracteristici:

- **Familiaritatea:** sintaxa limbajului este foarte ușoară combinând sintaxele unora din cele mai populare limbaje Perl sau C
- **Simplitatea:** sintaxa limbajului este destul de liberă. Nu este nevoie de includere de biblioteci sau de directive de compilare, codul PHP inclus într-un document executându-se între marcasele speciale;
- **Eficiența:** PHP-ul se folosește de mecanisme de alocare a resurselor, foarte necesare unui mediu multiutilizator, așa cum este web-ul;
- **Securitate:** PHP-ul pune la dispoziția programatorului un set flexibil și eficient de măsuri de siguranță;

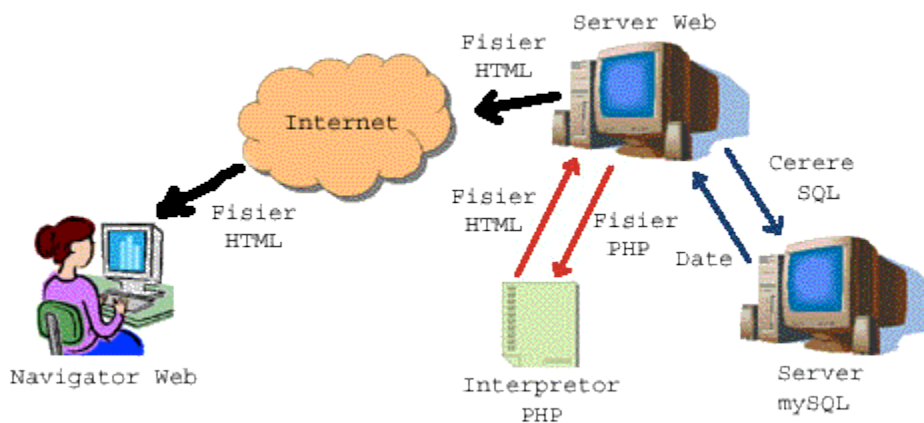
- **Flexibilitate:** fiind apărut din necesitatea dezvoltării web-ului, PHP a fost modularizat pentru a bine pasul cu dezvoltarea diferitelor tehnologii. Nefiind legat de un anumit server web, PHP-ul a fost integrat pentru numeroasele servere web existente: Apache, IIS, Zeus, server, etc.
- **Gratuitate :** este probabil cea mai importantă caracteristică a PHP-ului. Dezvoltarea PHP-ului sub licența open-source a determinat adaptarea rapidă a PHP-ului la nevoile web-ului, eficiențizarea și securizarea codului.

Utilizare:

PHP este simplu de utilizat, fiind un limbaj de programare structurat, ca și C-ul ,Perl-ul sau începînd de la versiunea 5 chiar Java , sintaxa limbajului fiind o combinație a celor trei. Probabil una din cele mai importante facilități ale limbajului este conlucrarea cu majoritatea bazelor de date relaționale, de la MySQL și pînă la Oracle, trecînd prin MS Sql Server, PostgreSQL, sau DB2.

PHP poate rula pe majoritatea sistemelor de operare, de la UNIX ,Windows, sau Mac OS X și poate interacționa cu majoritatea serverelor web. Codul dumneavoastră PHP este interpretat de serverul WEB și generează un cod HTML care va fi văzut de utilizator (clientului -browserului- fiindu-i transmis numai cod HTML).

PHP are rol de interpretator și comunicator între MySQL care gestionează bazele de date, Apache care are rol de server web. Fișierele PHP sunt interpretate de către serverul web iar rezultatul este trimis în formă de text sau cod HTML către browser-ul clientului.



Serverul Web încarcă pagina din sistemul de fișiere propriu și, constatând că este vorba de o pagină PHP o trimite interpretorului PHP. Interpretorul PHP parcurge pagina și execută instrucțiunile PHP întâlnite. Instrucțiunile PHP au ca și scop principal acela de a genera cod HTML. De asemenea, codul HTML din fișierul original este copiat pur și simplu la ieșire. Interpretorul PHP transmite înapoi către server fișierul transformat prin interpretarea codul PHP. Acest fișier va conține doar cod HTML, dar nu și cod PHP.

SQL este un limbaj de programare specific pentru manipularea datelor în sistemele de manipulare a bazelor de date relaționale (RDBMS), iar la origine este un limbaj bazat pe algebra relațională. Acesta are ca scop inserarea datelor, interogații, actualizare și ștergere, modificarea și crearea schemelor, precum și controlul accesului la date. A devenit un standard în domeniu (standardizat ANSI-ISO), fiind cel mai popular limbaj utilizat pentru crearea, modificarea, regăsirea și manipularea datelor de către SGBD-urile (Sistemele de Gestionare a Bazelor de Date) relaționale. SQL permite atât accesul la conținutul bazelor de date, cât și la structura acestora.

SQL pune la dispoziție comenzi pentru o varietate de task-uri incluzând:

- date interogate
- inserarea, extragerea și ștergerea rândurilor într-un tabel.
- crearea, modificarea și ștergerea obiectelor de tip bază de date
- controlul accesului la baza de date și la obiectele de tip bază de date.
- garantarea consistenței bazei de date

MySQL este un sistem de gestionare a bazelor de date relaționale. Limba utilizată pentru a efectua operațiuni într-o bază de date se numește SQL. Diferența cheie între SQL și MySQL este că: SQL este o limbă pentru a gestiona datele dintr-o bază de date relațională și MySQL este un sistem de gestionare a bazelor de date bazate pe relații bazate pe open source pentru a gestiona bazele de date utilizând SQL.

Cu ajutorul aplicației MySQL gestionăm bazele de date, în timp ce bazele de date sunt fișiere structurate specific, create special pentru o modificare rapidă și eficientă. MySQL este o bază de date relațională. Aceasta înseamnă că datele din el sunt în tabele, care pot fi într-o relație reciprocă. Fiecare tabel reprezintă o noțiune a programului nostru și conține rânduri. Rândurile din tabel reprezintă exemple concrete a ceea ce reprezintă tabelul.

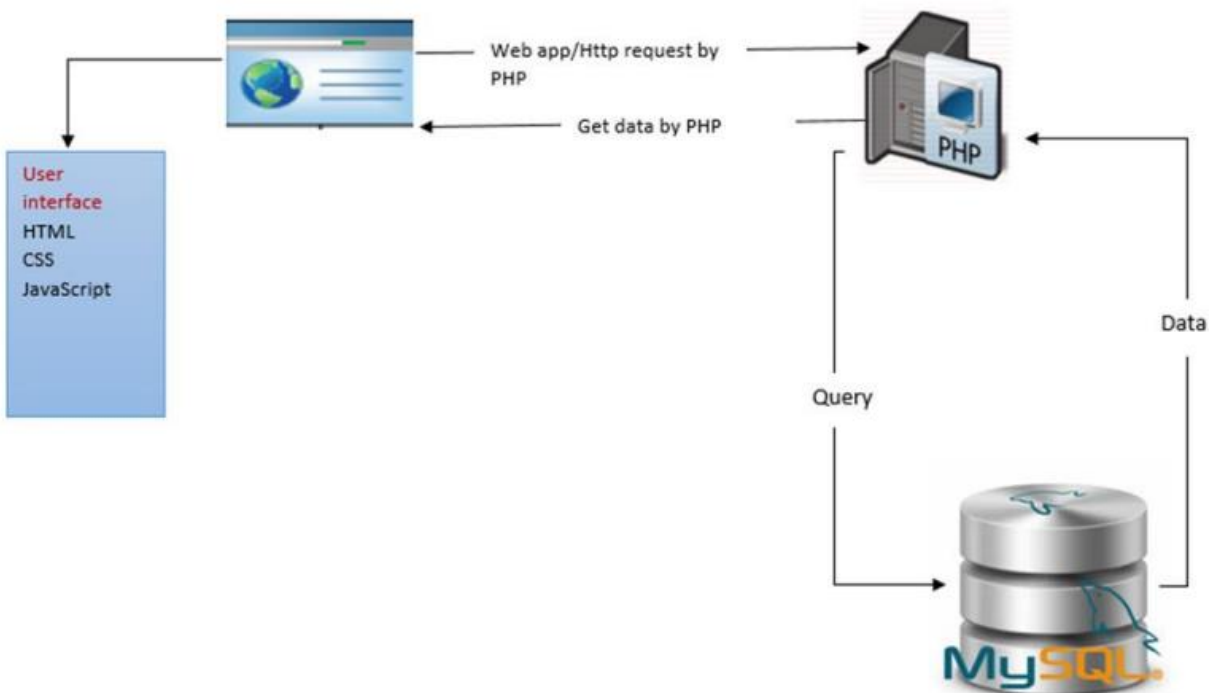
MySQL este aplicație server. Alte aplicații se conectează la ea local sau prin rețea, apoi are loc un schimb reciproc de mesaje prin intermediul limbajului SQL.

4. Proiectarea aplicatiei

4.1 Descrierea arhitecturii sistemului

Arhitectura sistemului este împărțită în două parti: clientul care conține interfața și serverul care conține paginile web scrise în diferite limbaje de programare și baza de date. Prin interogările SQL scrise în paginile PHP se poate accesa datele din baza de date. Tehnologiile utilizate au fost explicate anterior la capitolul de fundamente teoretice.

În continuare vom prezenta o *schema a arhitecturii sistemului* :

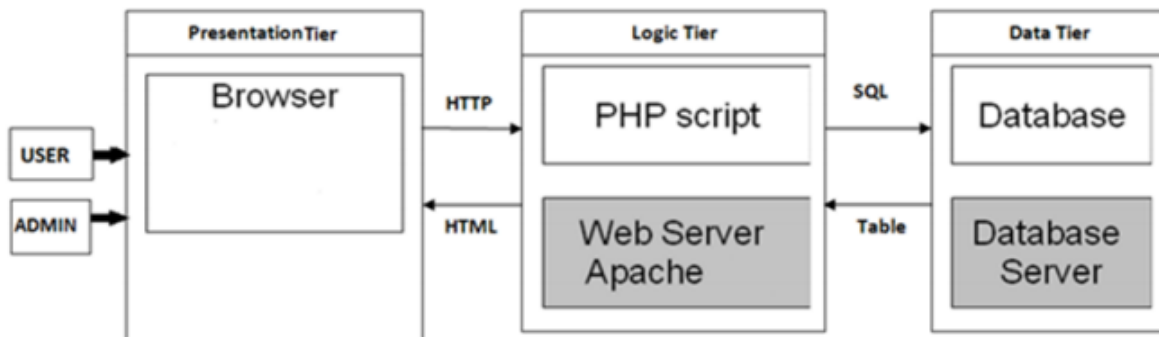


Pentru serverul Web am folosit o arhitectură pe 3 niveluri. Trei niveluri este o arhitectură client-server în care interfața utilizator, procesul de afaceri, stocarea datelor și accesul la date sunt dezvoltate și menținute ca module independente sau cel mai adesea pe platforme separate.

Cele 3 niveluri sunt:

1. Nivelul de prezentare care afișează site-ul web la cererea clientului.
2. Nivelul logic este stratul de mijloc care include logica sistemului. Acest nivel trimite cererea clientului către nivelul de date și nu permite clienților să acceseze direct baza de date.
3. Nivelul de date are dreptul să efectueze modificări asupra bazei de date cum ar fi: crearea de tabele, stergerea lor, modificarea unor constrângeri și valori din baza de date.

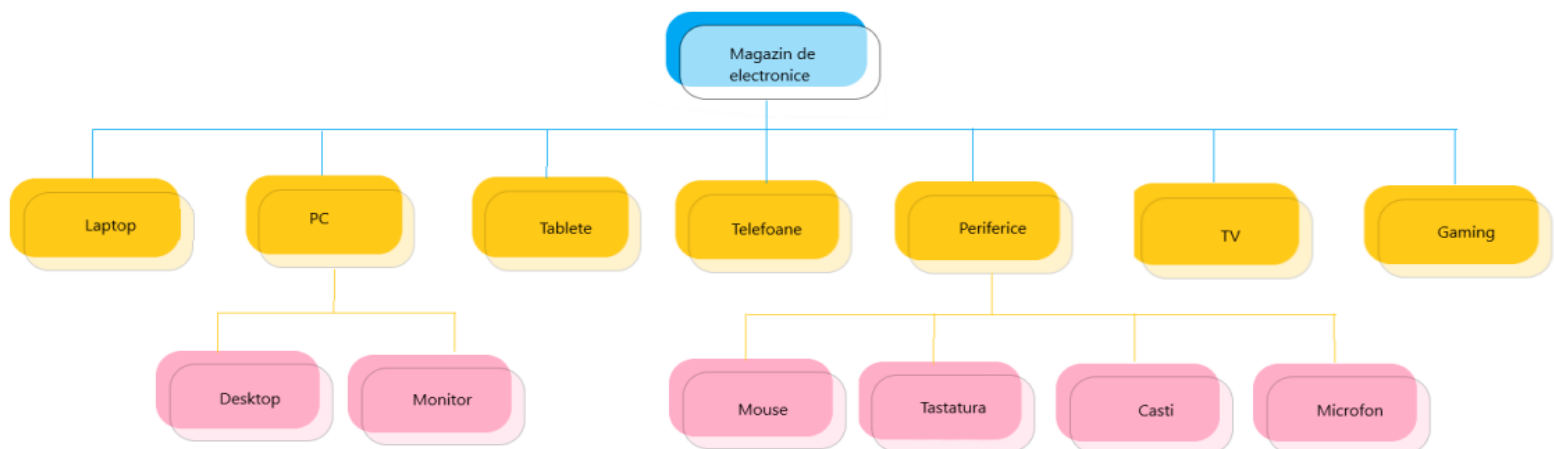
Schema arhitecturii pe 3 nivele:



Am ales pentru construirea site-ului o arhitectura de tip sub-folder. Acest tip de arhitectură este cel mai des întâlnit și în general caracterizează site-urile de mărime mică sau medie. Folosind acest tip de structură se oferă avantajul punerii în valoare a celor mai importante secțiuni ale site-ului.

Avem un site prin intermediul caruia vindem produse electronice. Astfel pot fi delimitate articolele în: laptopuri, tablete, telefoane, desktop, monitor, mouse, tastaturi, casti, microfon, TV, gaming, foto. În felul acesta, vizitatorii vor dispune de o navigare ușoară pe site și ajung foarte rapid la secțiunea căreia îi aparține. Nimic mai simplu nu? Faptul că s-a creat o structură de genul acesta le permite atât vizitatorilor cât și roboților de căutare să focalizeze rapid. Vizitatorii vor avea de câștigat prin ușurința de a naviga pe site iar motoarele de căutare vor ști unde să așeze ca importanță fiecare sub-folder. Unul dintre cele mai mari avantaje ale acestui tip de arhitectură este că poate fi aranjat conținutul în același site și implicit nu duce la diluarea importanței paginilor.

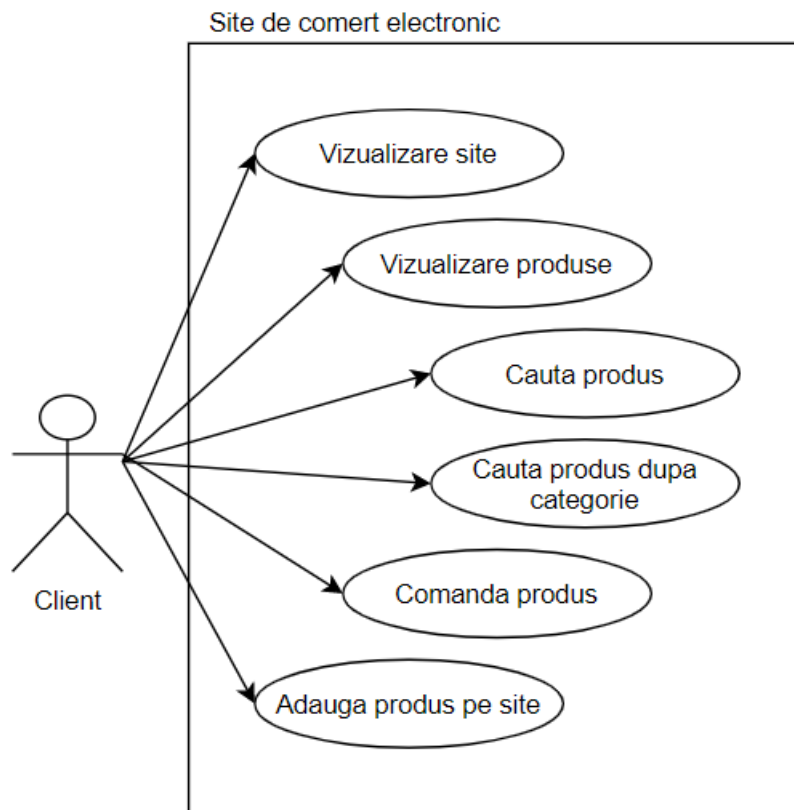
Ierarhia de navigare a site-ului:



4.2. Identificarea functionalitatilor aplicatiei

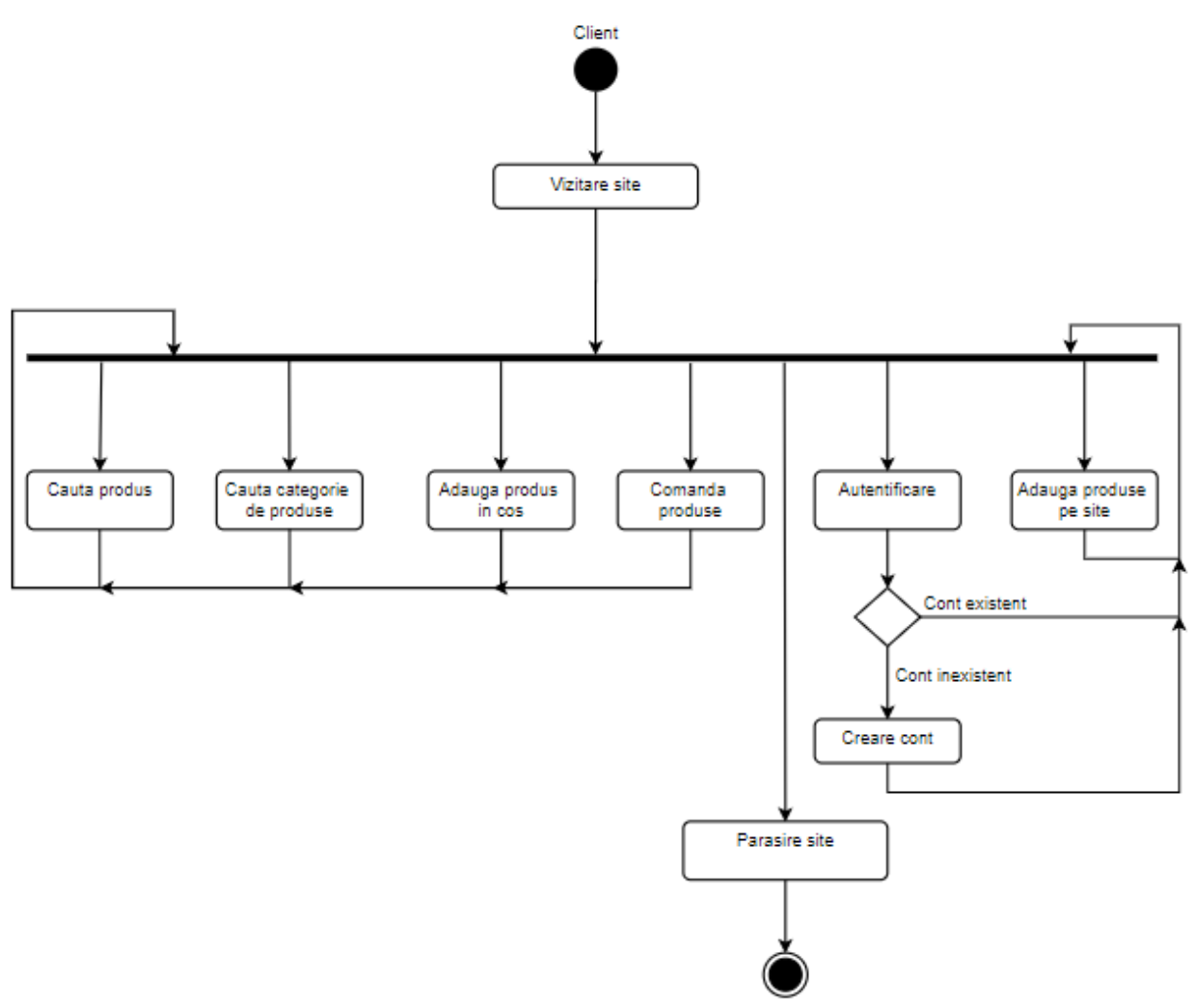
O diagrama a cazurilor de utilizare (use case diagram) prezinta o colectie de cazuri de utilizare si actori care: ofera o descriere generala a modului in care va fi utilizat sistemul, furnizeaza o privire de ansamblu a functionalitatilor ce se doresc a fi oferite de system, arata cum interactioneaza sistemului cu unul sau mai multi actori si asigura faptul ca sistemul va produce ceea ce s-a dorit.

Functionalitatile aplicaiei sunt prezentate cu ajutorul unei **diagramme a cazurilor de utilizare**:



4.3. Diagrama de interacțiune: activitate, secvența, colaborare

Diagrama de activitate este o variantă a diagramei de stare și este folosită pentru a modela dinamica unui proces sau a unei operații. Ajută la reprezentarea vizuală a secvențelor de acțiuni prin care se dorește obținerea unui rezultat. Descrie fluxul de lucru dintr-un punct de plecare până într-un punct de terminare, detaliind căile de decizie care pot apărea într-o activitate. Nu se construiește pentru fiecare caz de utilizare și scenariu, deoarece nu este necesar, ci numai pentru cele importante.



Pentru modelarea dinamicii sistemului, UML furnizează tipuri de diagrame, și anume, diagramele de interacțiune : diagrama de secvență și diagrama de colaborare. Principala menire a acestor diagrame este de a arăta cum realizează sistemul un caz de utilizare sau un scenariu particular dintr-un caz de utilizare. Pentru fiecare caz de utilizare se pot realiza mai multe scenarii (din descrierea cazului de utilizare). Pentru fiecare astfel de scenariu se pot întocmi, nu este obligatoriu, o diagramă de secvență sau o diagramă de colaborare.

Diagramele de secvențe transformă evenimentele identificate în scenariile cazurilor de utilizare într-o reprezentare grafică a utilizărilor sistemelor de către actor. Diagrama de secvență descrie cronologic interacțiunea obiectelor, identificând mesajele schimbate între obiecte ca răspuns la un eveniment, împreună cu secvența mesajelor. Diagramele secvențiale cuprind obiectele care fac parte dintr-o anumită colaborare și descriu secvența de stimuli transmiși între obiecte în cadrul unei interacțiuni. Ele cuprind și dimensiunea temporală, deoarece fiecărui obiect îi corespunde o linie de viață, trasată vertical sub numele obiectului.

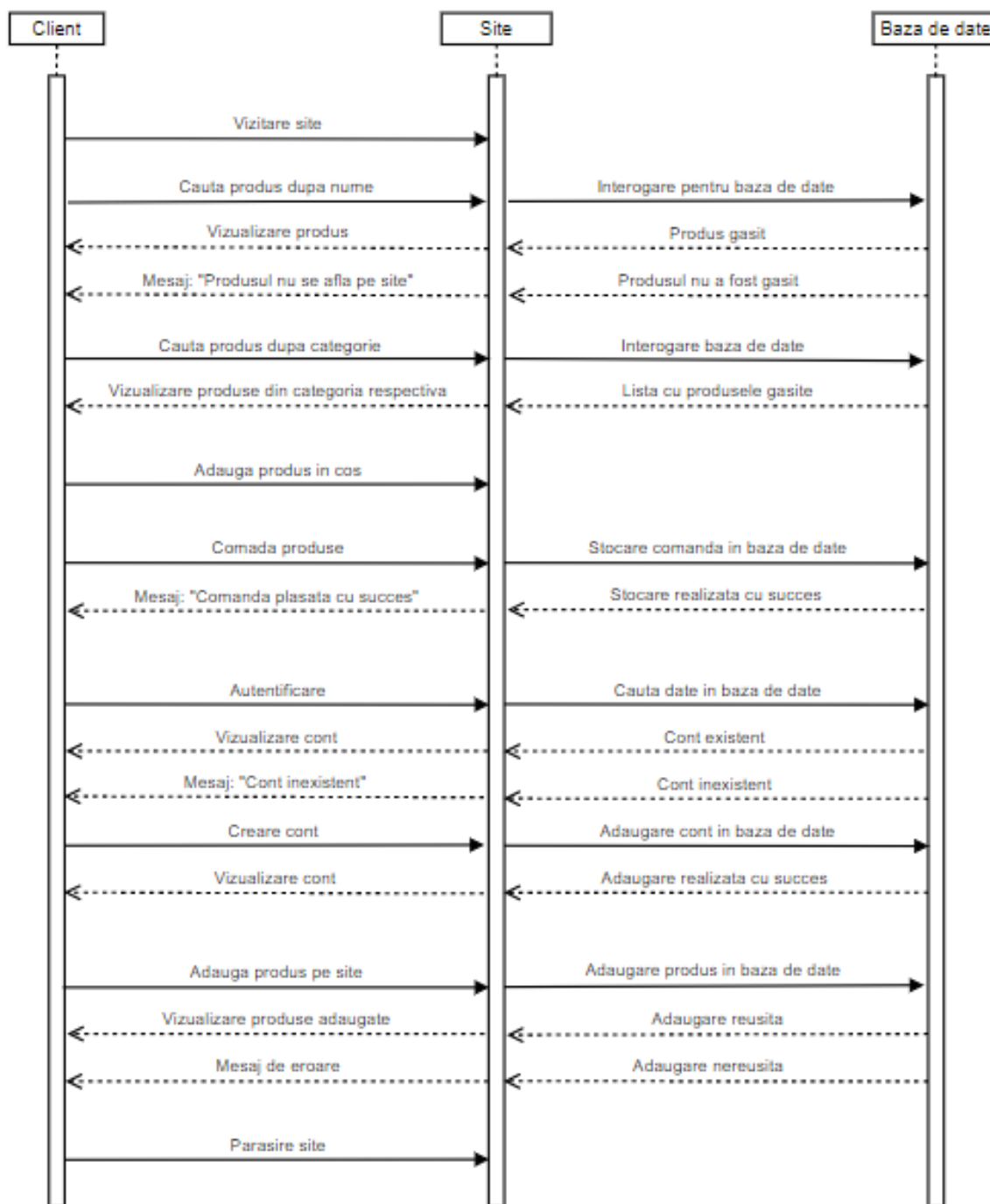
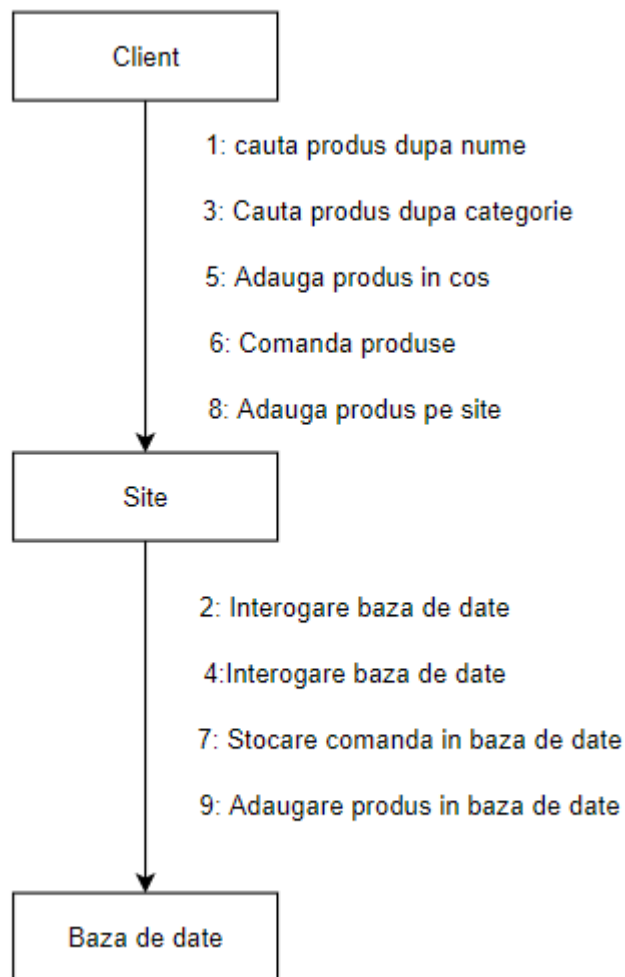


Diagrama de colaborare este o diagramă de interacțiuni care pune accentul pe organizarea structurală a obiectelor care participă la interacțiune. Diagramele de colaborare au multe asemănări cu diagramele de secvență, exprimând aceleași informații dar într-un alt format. Pot fi create la nivele diverse de detaliu și în diferite stadii de dezvoltare a procesului software. Deoarece au un conținut similar, pot fi folosite pentru generarea diagramelor de secvență și viceversa. Diferența semnificativă față de diagrama de secvență este aceea că diagrama de colaborare arată explicit legăturile dintre obiecte. De asemenea, la diagrama de colaborare timpul nu are o dimensiune explicită. Din acest motiv, ordinea în care sunt trimise mesajele este reprezentată prin numere de secvență.



4.4. Detalii de implementare: diagrama bazei de date

O **bază de date** este o colecție organizată de informații sau de date structurate, stocate electronic într-un computer. O bază de date este controlată, de regulă, de un sistem de management al bazelor de date (DBSM). Cumulat, datele, DBMS și aplicațiile asociate reprezintă un sistem de baze de date, denumit prescurtat bază de date.

Datele din cele mai obișnuite tipuri de baze de date sunt distribuite de regulă pe linii și coloane, în diferite tabele, pentru eficientizarea procesării și interogării datelor. Datele pot fi accesate, gestionate, modificate, actualizate, controlate și organizate cu ușurință. Majoritatea bazelor de date utilizează un limbaj structurat de interogare (SQL) pentru scrierea și interogarea datelor.

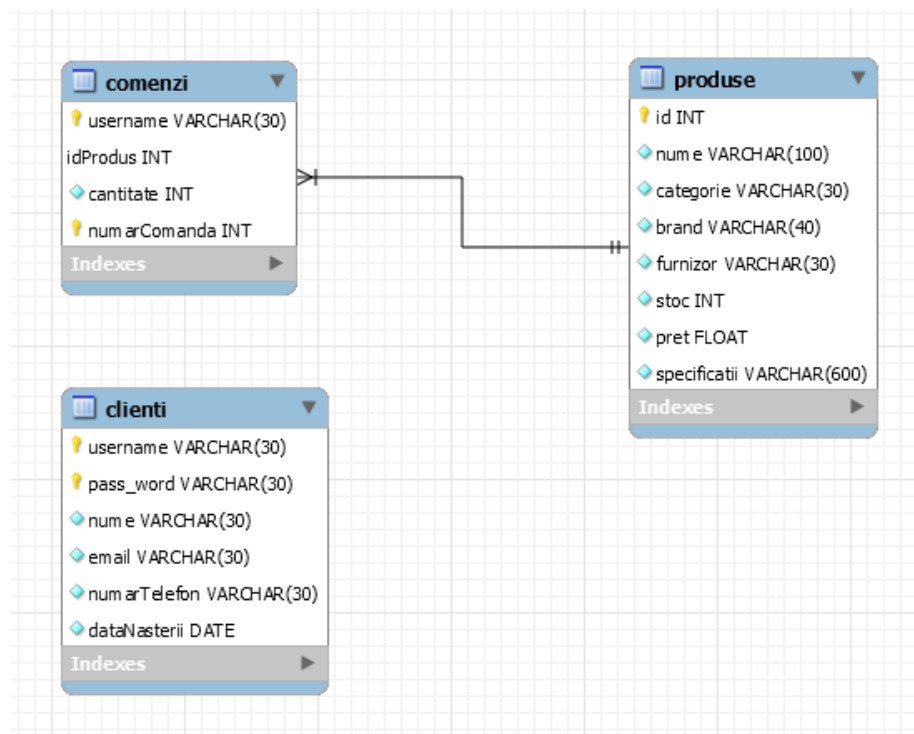
Există multe tipuri de baze de date. Cea mai bună bază de date pentru o anumită organizație depinde de modul în care organizația intenționează să o utilizeze. Pentru acest proiect a fost implementată o bază de date relatională.

SQL este un limbaj de programare utilizat de aproape toate bazele de date relaționale, pentru interogarea, gestionarea și definirea datelor, precum și pentru controlul oferirii accesului.

O bază de date necesită, de regulă, un software cuprinzător pentru aceasta, program cunoscut drept sistem de management al bazei de date (DBMS). Software-ul folosit pentru baza de date a proiectului este MySQL.

MySQL este un sistem open source de management al bazelor de date relaționale bazat pe SQL. A fost proiectat și optimizat pentru aplicațiile web și poate rula pe orice platformă. Deoarece este creat pentru a procesa milioane de interogări și mii de tranzacții, MySQL reprezintă o opțiune populară pentru companiile de comerț electronic.

Diagrama bazei de date:



In tabela Produse stocam produsele disponibile pe site.

In tabela Clienti, date despre Clienti.

Iar in tabela Comenzi, salvam comenzile date pe site.

4.5. Cazuri de testare

Cazurile de testare ale aplicatiei sunt urmatoarele:

- vizitare site
- vizualizare produse
- cautarea unui produs dupa nume
- cautarea unui produs dupa categorie
- punerea unui anumit produs in cos
- punerea unui produs pe site in calitate de provider

Toate aceste cazuri de testare vor fi prezentate si exemplificate in sectiunea urmatoare.

7. Testarea propriu-zisa a aplicatiei

- Vizitare site

- pagina principala a site-ului vizibila in momentul in care intram pe site



Ossas Electronics

Acasa

Produse



CASTI GAMING HYPERX CLOUD II PRO

Indiferent daca aveti nevoie de sunet surround 7.1 sau stereo, exista un model Cloud facut pentru dvs. Fiecare model Cloud este proiectat sa va mentina mai mult timp in joc, oferind confort premiat cu ajutorul spumei de memorie HyperX™.

Cumpara acum!



Laptop ASUS ExpertBook

Specificatii:

Procesor AMD Ryzen 3 3250U

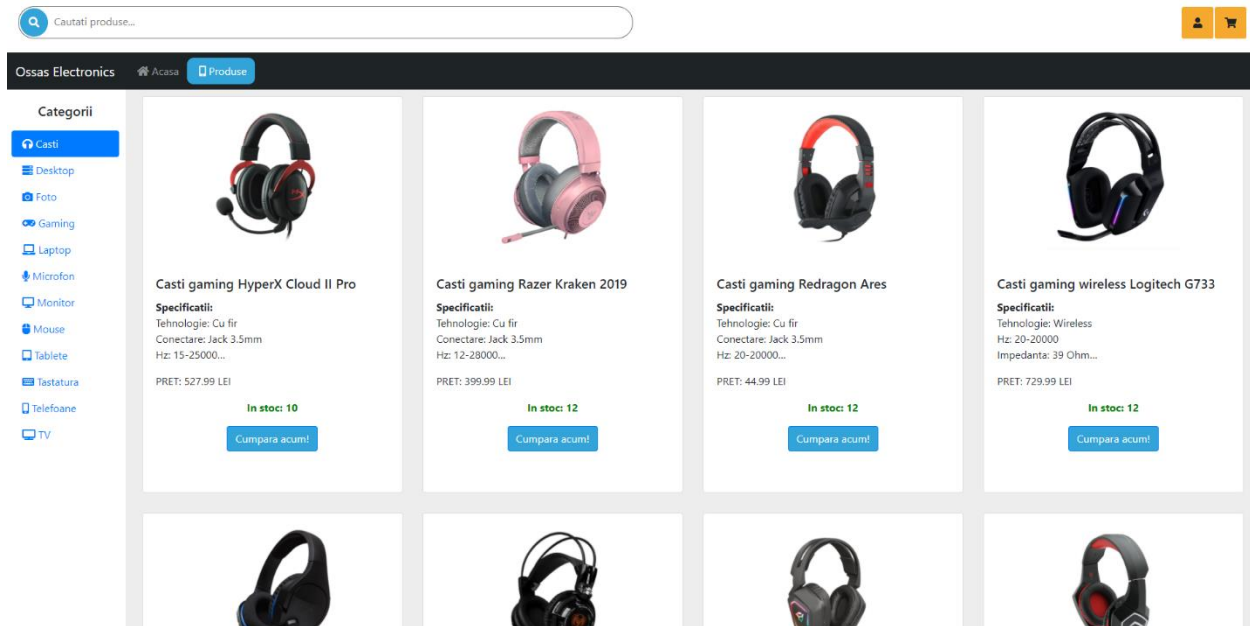


Laptop ultraportabil Dell

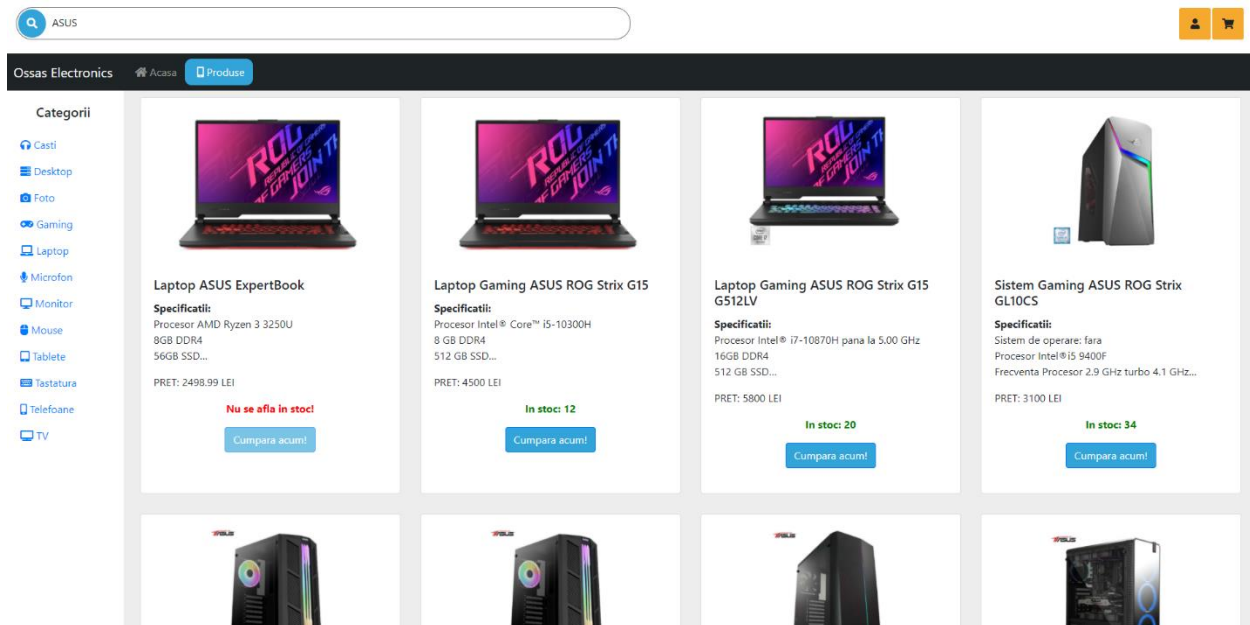
Specificatii:

Procesor Intel® Celeron™ 5205U 1.90 GHz

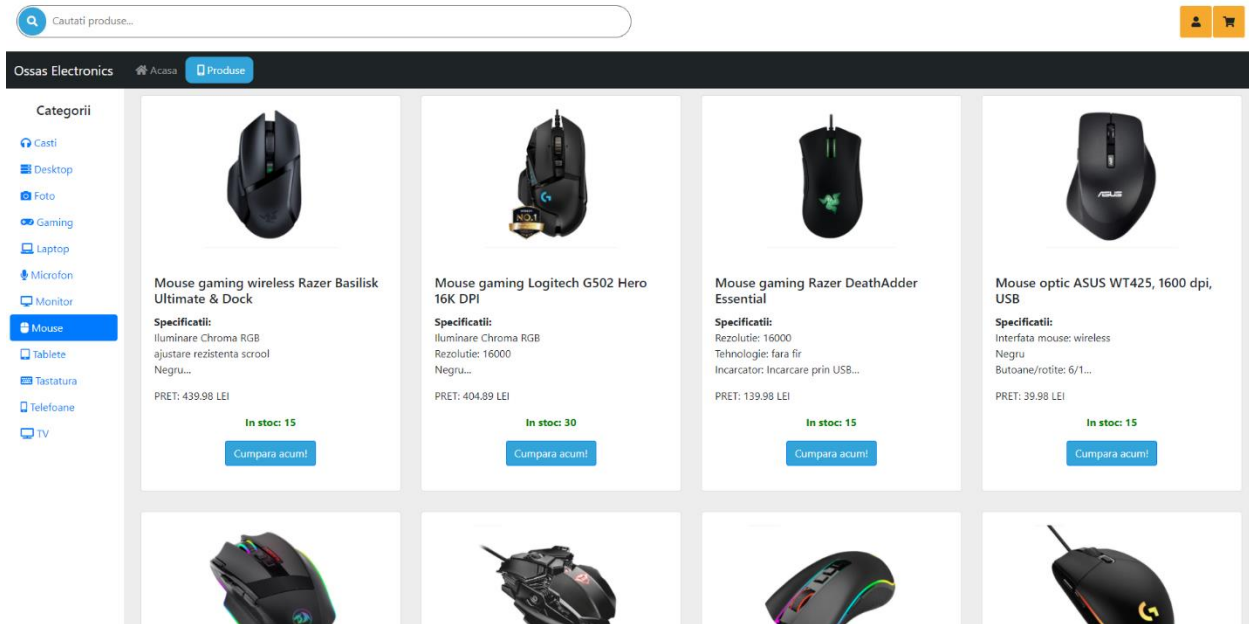
- Vizualizare produse
 - pagina de vizualizare produse este una diferita fata de cea principala, de aici putem selecta categoriile de produse



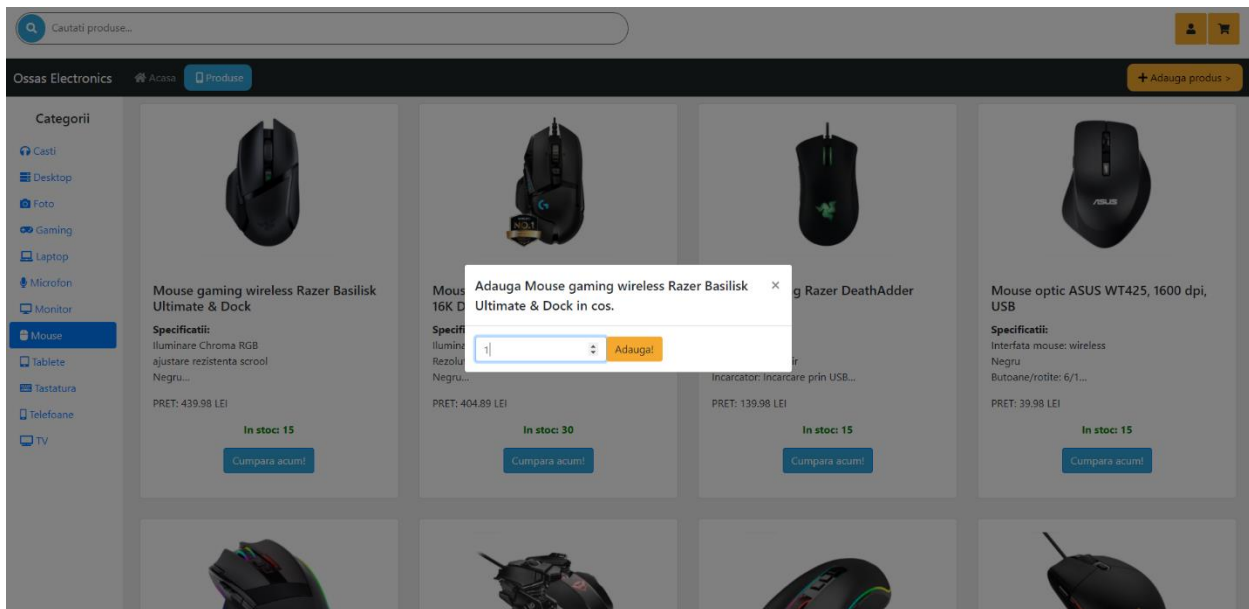
- Cautarea unui produs dupa nume
 - putem introduce in bara de cautare categoria unui produs, brand-ul sau numele produsului
 - in exemplul de mai jos am introdus cuvantul 'ASUS' si am primit ca rezultat produse ale acestui brand existente pe site



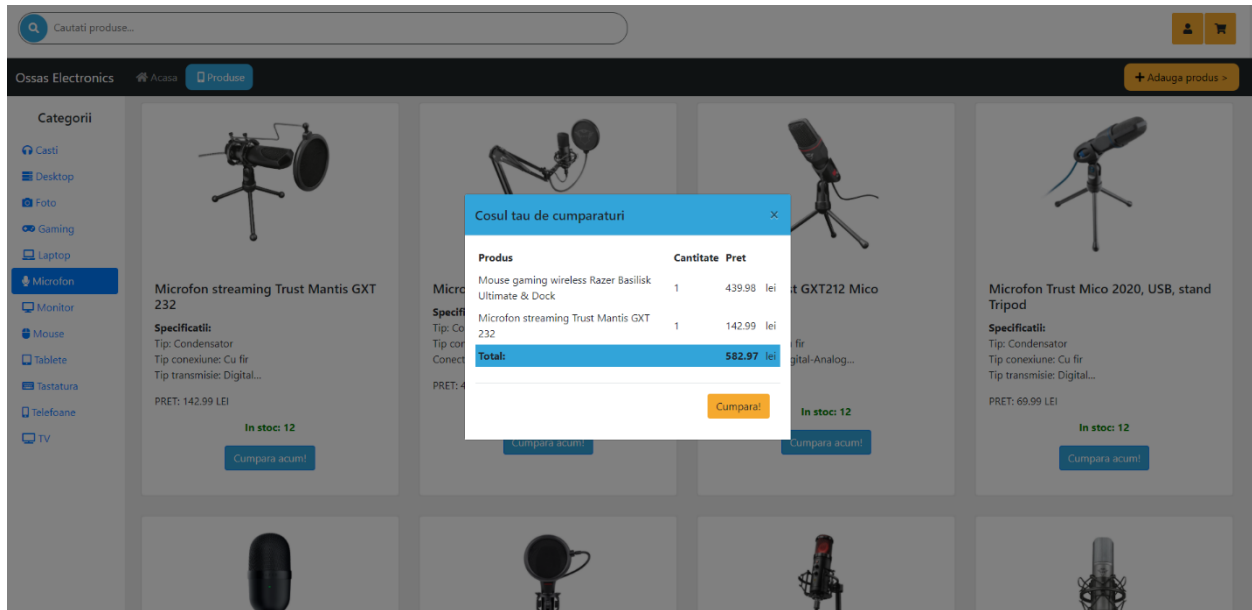
- Cautarea unui produs dupa categorii
 - pentru a cauta un produs dupa categorii este suficient sa selectam categoria din meniul disponibil in partea stanga



- Punerea unui anumit produs in cos
 - pentru a adauga un produs in cos apasam butonul 'Cumpara acum' disponibil sub produsul respectiv si introducem cantitatea dorita, dupa care apasam butonul 'Adauga'

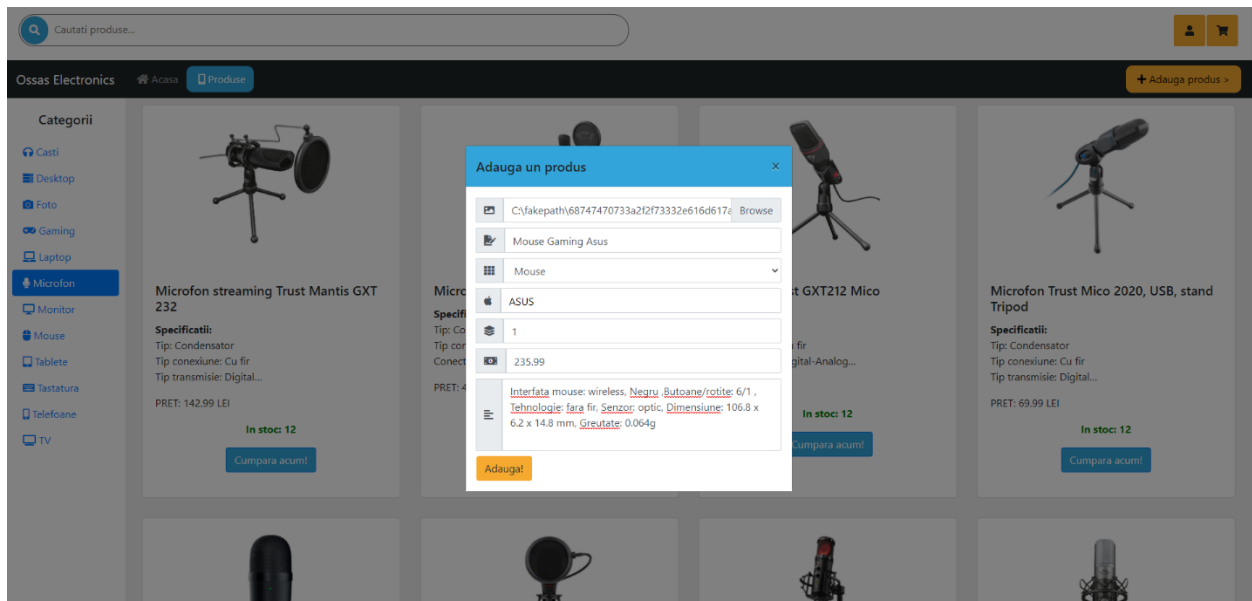


- dupa ce am adaugat produsele dorite in cos putem da click pe butonul ‘Cosul meu’ pentru a vizualiza atat produsele cat si pretul acestora, dar si pretul total al produselor



- Punerea unui produs pe site in calitate de provider

- pentru a aduga un produs pe site trebuie sa te loghezi cu un cont de utilizator, dupa care va fi disponibil in dreapta sus butonul ‘Adauga produs’, dupa care va fi disponibila o fereastră in care vom introduce date despre produsul pe care dorim sa il punem pe site




6. Manual de instalare si utilizare


Manual de utilizare:

In calitate de client sunt disponibile urmatoarele actiuni:


- creare cont sau log in


Conecteaza-te! ×


 Username


 Parola


Log-in



 E-mail


 Username

 Parola

 Nume

 Prenume


 mm/dd/yyyy 


 Numar de telefon


Register


- vizualizare date cont si deconectare


Bun venit Adelina Moldovan! ×

 **Nume de utilizator:** ademoldovan

 **Nume:** Adelina Moldovan

 **Email:** adelinamoldovan59@yahoo.com

 **Numar de telefon:** 0754697657

 **Data de nastere:** 1999-12-22

Deconecteaza-te!

- daca nu esti conectat nu se pot adauga produse in cos si nici nu se pot adauga produse pe site

- in modul log in sunt disponibile actiunile de mai sus: adaugare produs in cos, vizualizare cos si plasare comanda, de asemenea poti adauga propriile produse pe site

7. Concluzii si dezvoltari ulterioare

In concluzie, am reusit sa dezvoltam corespunzator cerintele aplicatiei si anume: vizualizare site, vizualizare produse, cautarea unor produse dupa numele acestora sau categoria din care fac parte, login si logout pentru client, adaugarea unor produse in cos si plasarea comenzii , adaugarea unor produse pe site de catre client.

In ceea ce priveste dezvoltarile ulterioare ale aplicatiei acestea pot fi multiple. De exemplu: am putea largi gama de produse existente in baza de date a magazinului, de asemenea am putea adauga cateva optiuni pentru clienti: acestia ar putea adauga review pentru produsele achizitionate, am putea adauga o sectiune de intrebari adresate de catre vizitatorii site-ului pentru moderatori sau alti clienti.

Am putea adauga o sectiune cu produse care se afla la oferta pe o anumita perioada de timp, de asemenea se poate adauga varianta de plata online a comenzilor sau posibilitatea de a achizitiona produse in rate.

8. Bibliografie

<https://www.javatpoint.com/sql-tutorial>

<https://www.w3schools.com/html/default.asp>

<https://tutorialehtml.com/ro/tutoriale-css-introductere-in-css/>

<https://www.javascript.com/>

<https://www.wikipedia.org/>

<https://stackoverflow.com/>