# Proiect IS

BookShop

Continut:

1. **Prezentare Generala**
2. **Functionalitate**
3. **Arhitectura si Designul Sistemului – prezentarea diagramelor**
4. **Designe Pattern**

1. **Prezentare Generala**

Functionalitatiile principale ale aplicatiei sunt: logare, inregistrare, cautare de produse si cumparare de produse.

Aplicatia are o arhitectura de tipul Client-Server folosind protocolul http. Serverul BookShopRepository este scris in Java cu ajutorul frameworkului SpringBoot si are rolul de a oferi back-endul aplicatiei si datele care provin din baza de date. Aceasta a fost dezvolata in ide-urile Eclipse si InteliJI.

Pentru utilizare va trebui mai intai pornita aceasta aplicatie, care va functiona pe portul 8080. Clientul BookShopAngularApp este interfata proiectului scrisa cu ajutorul frameworkului angularjs. Partea de interfata este o aplicatie de tip web astfel este scrisa in limbajele html, css si specific frameworkului TypeScript.

Aceasta aplicatie este dezvoltata in ide-ul vsCode si se porneste din linia de comanda, din folderul aplicatiei BookShopAngularApp cu comanda "ng serve". Aplicatia este de tip secure:https si va afisa infotmatiile pe pagina cu url-ul:https://localhost:4200. Cand se porneste aplicatia aceasta conduce utilizatorul pe pagina de pornire Home.

Pe langa aceste abilitati un administrator poate vedea si modifica toate comenziile, utilizatorii si produsele.

Pagina de cont ii permite unui utilizator sa-si vada informattile de cont, sa vada cat a cheltuit pe produsele cumparate, sa isi creeze un wishlist si sa primeasca recomadari.

1. **Functionalitate**

* **Logarea:**

Este implementata doar pentru logarea clasica cu un username si o parola. Mai exista posibilitatea de logare cu un cont de facebook dar nu este implementata. In caz de erori, acestea apar pe ecran intr-o casuta rosie. In caz de reusita se ajunge pe pagina de home unde se va vedea ca acesta este logat. Se poate intra in pagina de admin sau user in functie de rolul acestuia.

* **Inregistrarea**

Se poate ajunge la pagina de inregistrare din pagina de home, se introduc datele, campul de role si de check password se lasa necompletate. Toate celalate campuri sunt obligatorii si genereaza erori daca sunt lasate necompletate. Daca apar erori se afiseaza un mesaj cu acestea intr-o casuta rosie deasupra rubricii. Administratorii se pot inregistra numai din back-end.

* **Cautare**

Pentru cautare se folosesc butoanele Search, Filter si bara de cautare. Hovering pe butonul de Filter descopera un meniu cu tipurile de filtre de cautare care se pot folosi: numele produsului, categoria, autorul, editura, anul aparitiei si 3 filtre legate de pretul produselor, nu se poate folosi decat un singur tip de filtru. Filtrele se fixeaza inainte de cautare si va aparea scris pe buton filtrul folosit. In casuta de cautare se scrie cererea dorita si se apasa butonul search. Pe pagina principala vor aparea apoi numai noile poze. Pentru a vedea informatii despre un produs se apasa pe poza produsului care il conduce pe utilizator in pagina produsului.

* **Cumparare**

Pentru a putea cumpara un produs este necesar sa se mearga pe pagina unui produs si a se adauga produsul in cos apasand butonul **Adauga in cos**. Pentru a se putea realiza o comanda se merge pe pagina cosului de cumparuturi **buying cart** ,se apasa pe butonul **Finalizare plata** si se merge pe pagina de completare a informatiilor pentru plata si dupa apasarea butonul finalizare comanda, comanda se realizeaza.

* **Pe pagina principala(home)**

Un user vede imaginile produselor si reclamele, de pe pagina princiapala poate fi accesata orice alta pagina din aplicatie si se pot cauta produse. Pentru cautarea de produse exista o bara de cautare care vine impreuna cu un meniu de filtere. De pe pagina principala un user poate vedea cosul de cumparaturi, pagina contului, daca este logat, sa se logheze, sa se delogheze si sa se inregistreze ca si utilizator.

Tot pe pagina de home exista butonul **"I am feeling lucky"**: alege o carte la intamplare din seria cartiilor existente din toata cartiile prezente pe pagina.

* **In pagina produsului**

Utilizatorul poate realiza urmatoarele actiuni: sa adauge produsul in cosul de cumparaturi, se adauge intr-un wish list, te poti intoarce pe pagina principala sau sa mergi la cosul de cumparaturi. Aici se pot vedea informatiile produsului: titlul, categoria, autorul, anul aparitiei, editura, pretul si o descriere.

* **In pagina utilizatorului**

Un utilizator poate sa-si vada informatiile cu caracter public (parola este ascunsa) si sa le modifice, poate vedea comenziile pe care le-a realizat, poate vedea si sterge produse din wishlistul acestuia si eventual recoamndari daca acestea exista.

* **In pagina administratorului**

Se afla informatiile personale ale administratorului (fara parola), se afla informatiile relevante si publice despre utlizatori (fara parola), despre comenzi si despre produse.

Administratorul trebuie sa fie capabil sa faca modificari pe acestea sau chiar sa le stearga din baza de date. Administratorul mai poate genera noi produse. Acestea se relizeaza prin apasarea butoanelor denumite sugestiv.

* **In pagina cosului de cumparaturi**

Se pot vedea produsele selectate in timpul sesiunii de cumparare intr-un tabel, exista un buton de stergere a produselor, de modificare a cantitatii si butonul de trecere la scrierea informatiilor de plata.

* **In pagina pentru informatiile de plata**

Se completeaza campurile si se apasa butonul: **Finalizare Plata**

1. **Arhitectura si Designul Sistemului – prezentarera diagramelor**

* **Diagrama use case**

Reprezinta actiunile pe care userul le poate efectua in utilizarea programului.

Diagram

Description automatically generated

* **Diagrama de obiecte si clase**

**Diagram

Description automatically generated**

* **Diagrama de secventa**

Diagrama de secvente arata secventa de mesaje schimbate intre setul de obiecte care incearca sa indeplineasca un anumit task si ordinea in care apar.

**Diagram

Description automatically generated**

* **Diagrama de activitate**

Arata ca o schema logica.  Fluxul de control este declansat de incheierea actiunilor in cadrul sistemului. Fluxul poate fi secvential, simultan sau cu ramificare. Diagramele de activitate pot arata modul in care evenimentele unui caz de utilizare sunt legate intre ele sau modul in care o colectie de cazuri de utilizare se poate coordona pentru a reprezenta un flux de lucru de business.

Diagram

Description automatically generated

* **Diagrama de comunicare**

Descrie interactiunea dintre mai multi participanti la implementarea unei anumite functionalitati. Impreuna cu diagrama de secventa face parte din asa- numitele diagrame de interactiune, care sunt adesea utilizate pentru a specifica modul in care diferitele componente software functioneaza impreuna.

Diagram

Description automatically generated

1. **Designe Pattern**

Am ales sa implementam urmatoarele designe-uri pattern:

* **Observer**

Este un design pattern comportamental. Specifica comunicarea intre obiecte: observable si observers. Un observable este un obiect care informeaza observers despre modificarile starii sale, obiectele sale dependente vor fi notificate automat. Se utilizeaza atunci cand exista o relatie one-to-many intre obiecte.

* **Singleton**

Face parte din modelul creational. Acest tip de designe pattern ofera una dintre cele mai bune modalitati de a crea o singura instanta a unui obiect.

* **Builder**

Face parte din modelul creational. El construieste un obiect complex folosind obiecte simple si folosind o abordare pas cu pas. Foloseste constructor care este independent de alte obiecte**.**

**GIT**

https://github.com/TestOrganizationForStudens