

Universitatea Tehnică „Gheorghe Asachi” din Iași
Facultatea de Automatică și Calculatoare

~

White Little Dress Magazin online

Studenta: Maria Frentescu

Tema proiectului și motivația alegerii și motivația

Tema abordată constă în realizarea unei aplicații web ce pune la dispoziție o pagină principală de prezentare a mai multor produse destinate miresei. Pagina conține un meniu pentru creare cont și pentru logare.

Site-ul nostru sare în ajutorul viitoarelor mirese care sunt puse în situația de a alege cea mai potrivită rochie pentru ziua cea mare. Astfel, ele pot alege dintr-o varietate mare de rochii și accesorii începând de la un simplu voal până la rochia ideală pentru fiecare doamnă în parte!

Cum ziua nunții se apropie cu pași repezi, venim cu o colecție vastă de rochii de mireasă pentru a le scăpa pe viitoarele mirese de cea mai mare grijă a lor: alegerea rochiei potrivite și personalizate.

Scurtă analiză a tipurilor de aplicații existente din perspectiva temei alese

Această temă este de asemenea o provocare, din cauza faptului că există foarte mulți competitori în acest domeniu. Acești competitori și sursa noastră de inspirație dar la care am adăugat un surplus de originalitate se numără prin urmatoarele:

Wedding Day ~ <https://wedding-day.ro>

Eden Bride ~ <https://www.edenbride.ro>

Donna Alba ~ <https://donnaalba.ro>

Rochii de mireasă | Alice Desing ~ <https://www.alice-design.ro>

Bella Sposa ~ <https://www.bellasposa.ro>

Descrierea generală a implementării (caracteristici așteptate de la aplicație)

Aplicația dezvoltată este o aplicație web care oferă servicii de:

prezentare, reclame, securitate și administrare. Va exista posibilitatea plasării unei comenzi doar în modul user, și desigur crearea unui nou cont de utilizator ce va conține informații precum: numele utilizatorului, adresa de email, parola.

Aplicația va stoca datele esențiale într-o bază de date SQL ce conține tabelele: Utilizator, Produs, Comandă, Info_livrare, Cerere.

Resursele hardware/software utilizate

Aplicația își propune să fie disponibilă pe cele mai noi versiuni ale principalelor browsere web, fără o complexitate mare a implementării. Nu există un nivel egal de muncă pe partea de server și pe front-end, cum aplicația constă într-un joc creat integral în tehnologii web de front-end, iar back-end-ul este utilizat pentru interacțiunea cu baza de date.

Tehnologiile alese pentru implementarea aplicației sunt:

- JavaScript, cu inserții de jQuery pentru adresarea cererilor AJAX către server și pentru localizarea elementelor din DOM. Funcționalitatea este creată utilizând manipularea prin intermediul JavaScript.
- HTML5 cu CSS3 și Bootstrap pentru partea vizuală și de responsiveness a aplicației, pentru adaptarea la cât mai multe tipuri de interfețe și de dispozitive.
- Java, cu framework-ul Spring.

Identificarea actorilor și a componentelor principale

În aplicația web dezvoltată regăsim 2 actori, Guest și User, actorul Guest, în urma înregistrării devenind User, iar User-ul neautentificat fiind considerat Guest. Diferența între actori o reprezintă privilegiile fiecăruia în cadrul site-ului, User-ul putând accesa mai multe componente principale.

Componentele principale ale site-ului sunt prezentate de următoarele:
Înregistrare, Autentificare, Vizualizare produse, Efectuare comandă.

Înregistrare oferă posibilitatea Guest-ului de a-și crea un cont și de a deveni un User. Prin componenta Autentificare, User-ul are posibilitatea de a intra în cont.

Vizualizare produse oferă posibilitatea atât Guest-ului, cât și User-ului să vizualizeze ofertele disponibile de rochii de mireasă, cat și accesorii.

Componenta Efectuare comandă este disponibilă doar User-ului și permite acestuia să efectueze comenzi. Guest-ul este redirecționat către pagina de Login.

Diagramme UML de use-case

Diagrama cazurilor de utilizare prezintă o colecție de situații în care actorii interacționează cu sistemul. Clientul web, aflat în ipostazele de Guest și/sau User dispune de opțiuni de vizualizare a produselor și, doar User, achiziționarea lor. Din punctul de vedere al serviciilor oferite, există o parte de verificare atât a produsului cât și a clientului.

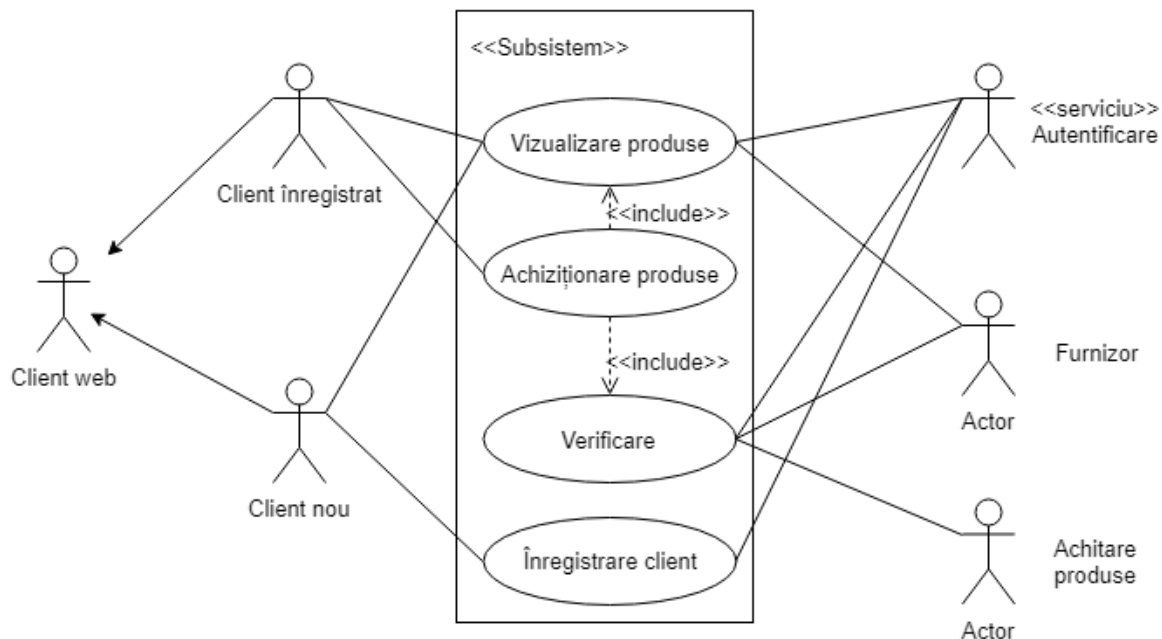
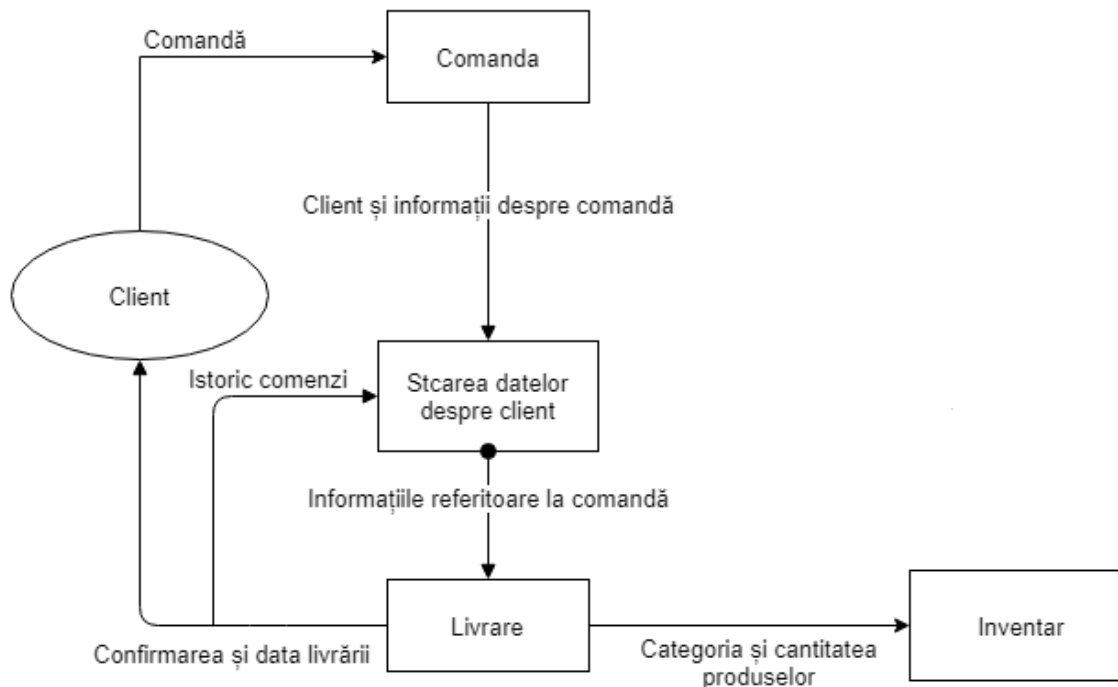


Diagramme UML de information flow

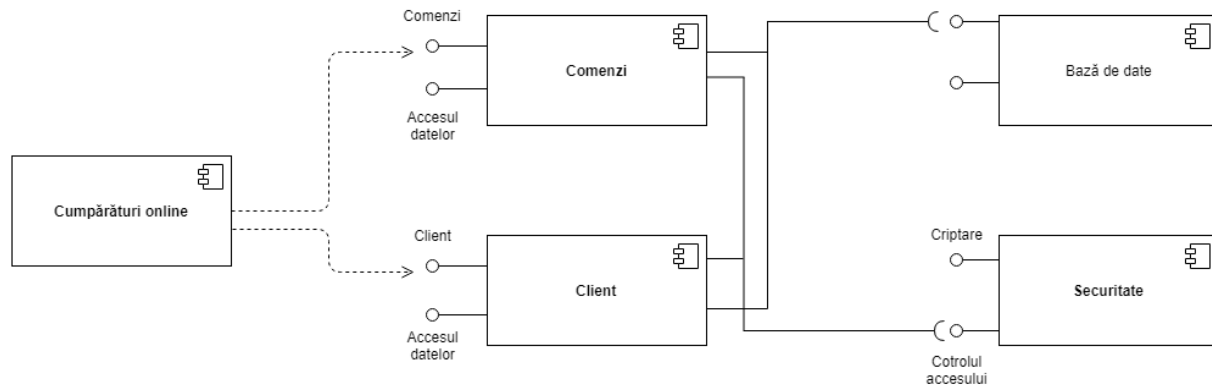
Diagrama fluxului de informații arată modul în care informațiile sunt comunicate dintr-o sursă către un receptor prin intermediul unui mediu. Astfel, la efectuarea unei comenzi de către un client se vor păstra inițial date despre comandă, apoi, o dată cu completarea datelor personale ale acestuia, se vor stoca și ele. O dată făcută verificarea și acceptată se poate realiza livrarea comenzii, ținând cont de cantitatea dorită și de confirmarea datei livrării.



Diagrame UML de componente

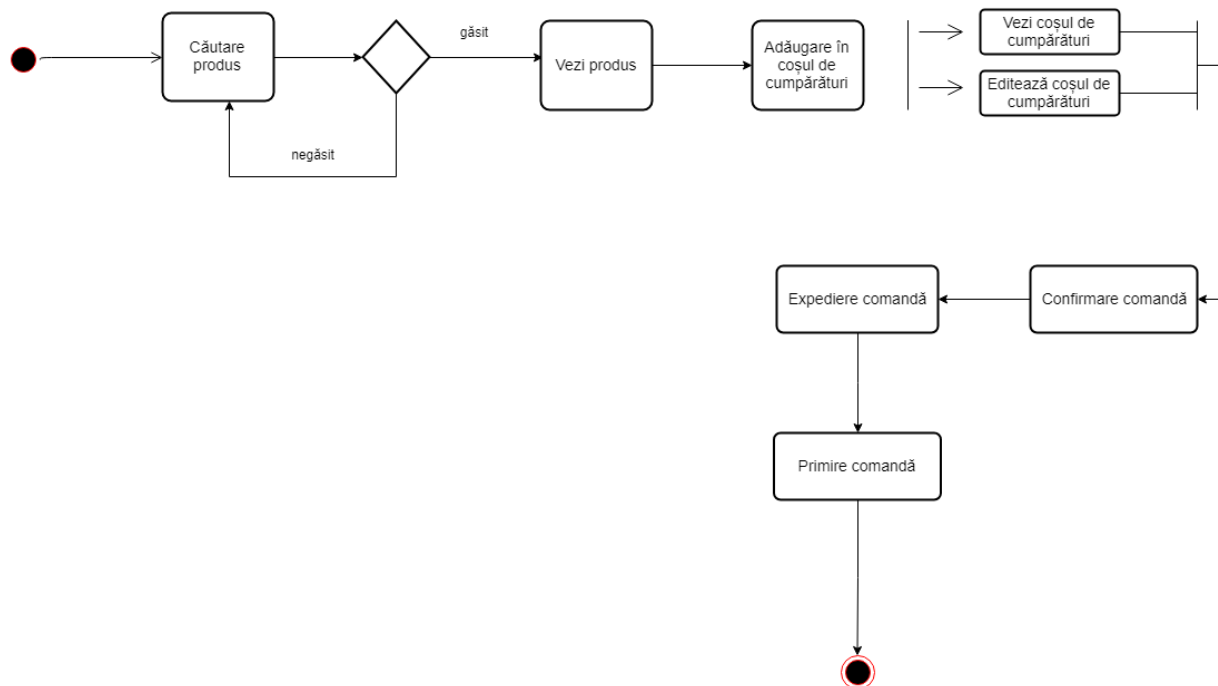
Diagrama de componente descrie modul în care un sistem este descompus în părțile sale componente și arată dependențele dintre acestea. Din această diagramă putem distinge componentele aplicației: Cumpărături online, Comenzi, Client, Bază de date și Securitate.

Componenta Cumpărături Online depinde de componentele Comenzi și Client. Componentele Comenzi și Client folosesc interfețele componentelor Bază de Date și Securitate.



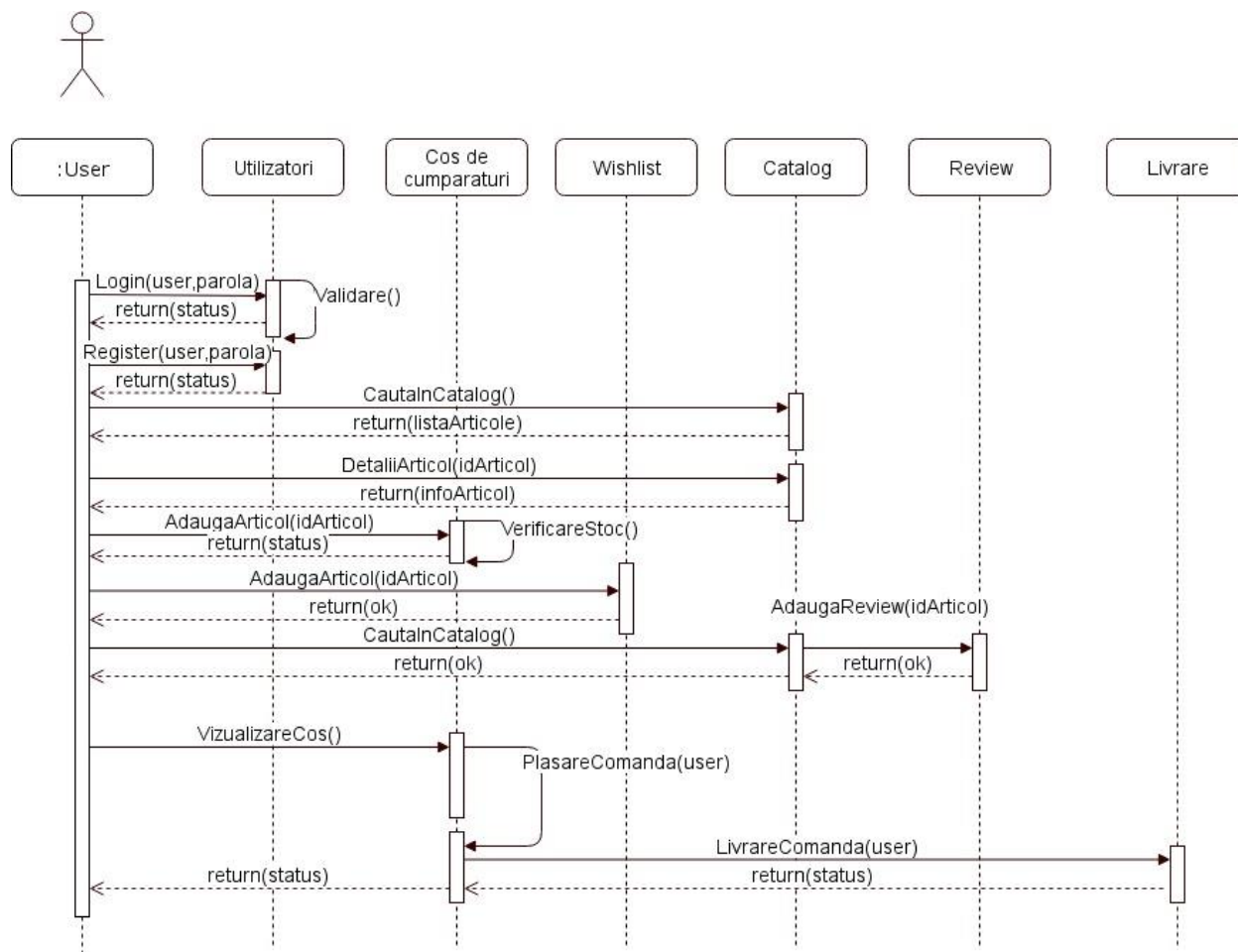
Diagrame UML de activitate

Diagrama de activitate descrie succesiunea de activități perationale ale componentelor unui sistem. Din această diagramă se observă că, clientul poate să caute produse, să vadă un anumit produs, să-l adauge în coșul de cumpărături, să vadă sau să modifice coșul de cumpărături și să plaseze comanda.



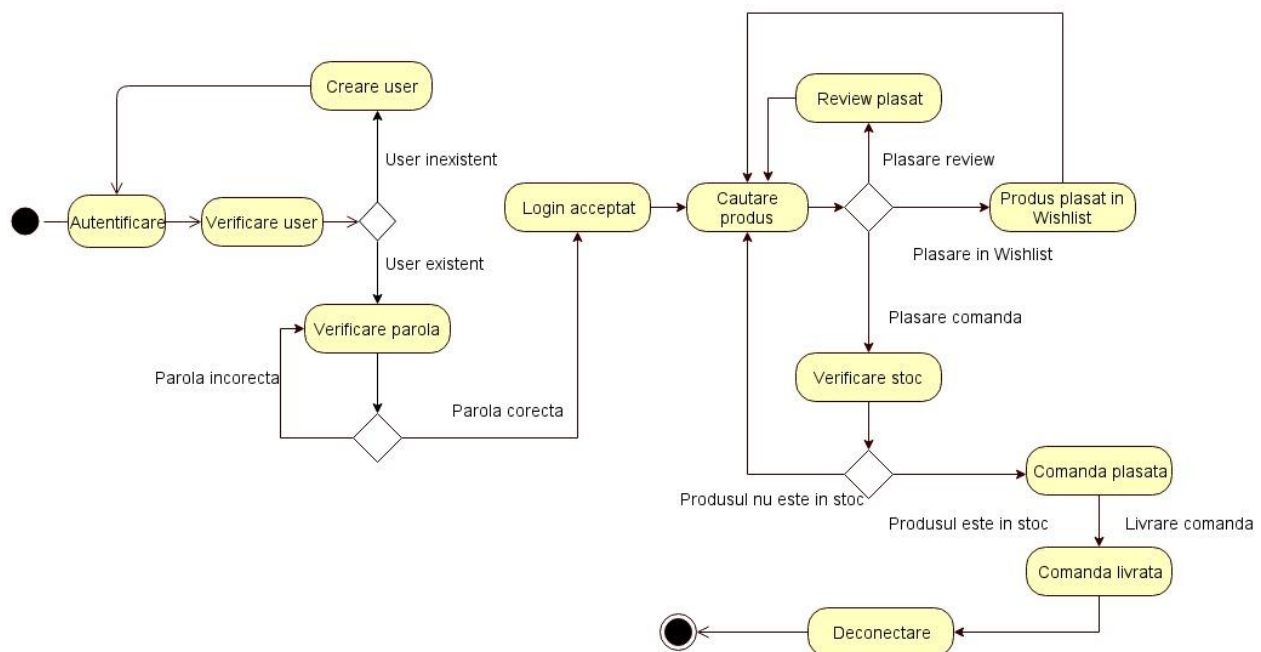
Diagrame UML de secvență

Diagrama de secvență are rolul de a modela scenariile posibile. Crearea obiectelor se reprezintă cu săgeata mesajului de creare indicând obiectul nou creat, iar perioadele de activitate se reprezintă prin benzi dreptunghiulare plasate pe liniile vieții, ca de exemplu perioada de așteptare din momentul în care lansezi o căutare a unui produs până în momentul în care se returnează rezultatul căutării. Trimiterea mesajelor se reprezintă cu săgeți orizontale indicând apelul unei operații. Durata de viață a unui obiect se reprezintă printr-o linie punctată vertical, iar timpul se scurge de sus în jos.



Diagrame UML de stare

Diagrama de stări este folosită pentru a modela comportamentul unui obiect și specifică o secvență de stări prin care trece un obiect de-a lungul vieții sale, ca răspuns la apariția unor evenimente împreună cu răspunsul la aceste evenimente. Diagrama începe și se termină cu noduri specifice. Stările ilustrează anumite acțiuni îndeplinite de protagonist care așteaptă ca răspuns un eveniment. Stările sunt reprezentate prin intermediul unor dreptunghiuri rotunjite la capete, iar tranzițiile între stări prin intermediul săgeților.



Evoluția proiectului

Am creat partea de login și logout folosind JSP, Servlet si MySQL.

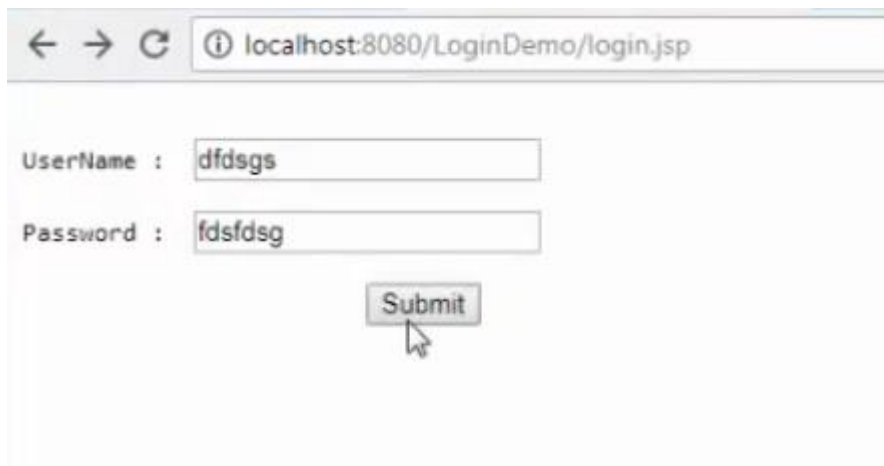
Configurarea serverului a fost făcută cu Apache Tomcat v8.5. Apache Tomcat este o aplicație open source a tehnologiilor Java Servlet, JavaServer Pages, Java Expression Language și Java WebSocket. Acesta permite numeroase aplicații web pe o scară largă de industrii și organizații.

În primă fază, am creat un proiect de tipul Dynamic Web Project numit Magazin care conține clase specifice serviciilor web, oferind un răspuns în format XML.

Există 4 metode utilizate pentru definirea serviciilor de tip RESTful, acestea se numesc servicii CRUD și sunt: POST(serviciu folosit pentru încărcarea sau modificarea unei noi resurse), PUT(serviciu folosit pentru crearea unei noi resurse), GET(serviciu folosit pentru citirea unei noi resurse) și DELETE(serviciu folosit pentru ștergerea unei resurse).

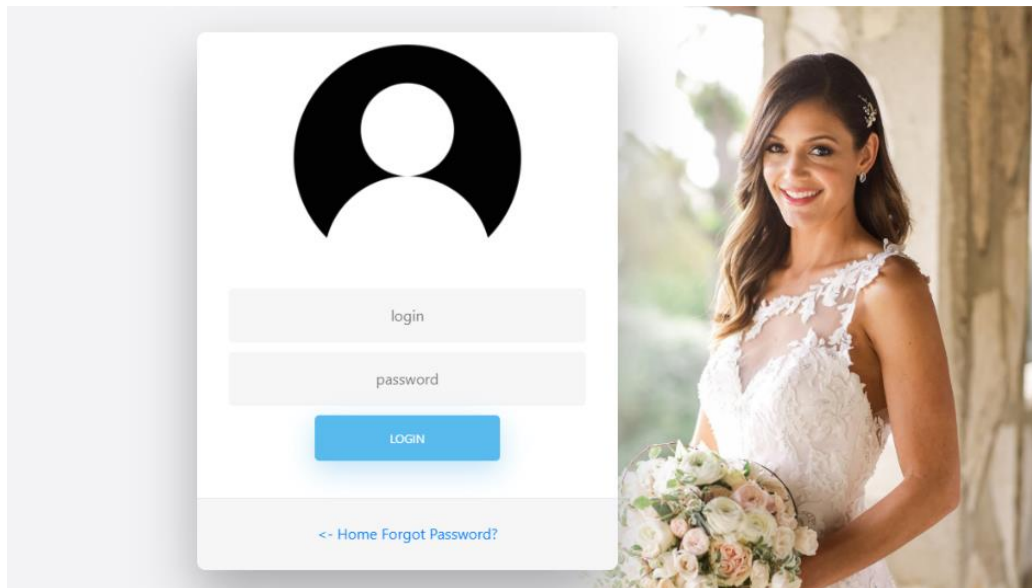
În continuare s-a folosit o bază de date. La proiectul deja creat în Eclipse, s-a adăugat o clasă nouă cu numele UserProfile care va permite conectarea la o bază de date MySQL.

Pagina de login fără elementele de HTML5, CSS, JavaScript:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/LoginDemo/login.jsp'. The page content is minimal, featuring two text input fields. The first field is labeled 'UserName :' and contains the text 'dfdsgs'. The second field is labeled 'Password :' and contains the text 'fdsfdsf'. Below these fields is a single 'Submit' button. A mouse cursor is visible over the button. The overall appearance is that of a simple, unstyled web form.

Pagina de login după adăugarea elementelor de HTML5, CSS, JavaScript:



În Etapa III urmează crearea și configurarea unui proiect părinte de tip Maven.

În această etapă va fi implementat magazinul propriu-zis în care va exista posibilitatea vizualizării produselor, plasării unei comenzi doar în modul user, și desigur crearea unui nou cont de utilizator ce va conține informații precum: numele utilizatorului, adresa de email, parola.

Probleme speciale sau dificultăți întâmpinate:

În special ne-am confruntat cu probleme în încercarea de a conecta baza de date.

Problemele au fost rezolvate în urma folosirii unor adnotari corespunzătoare:

`@Id, @Column(name="example"),`

`@GeneratedValue(strategy=GeneratedType.IDENTITY).`

Design Patterns:

Pentru a minimiza dependențele de alte clase am folosit unul din principiile SOLID, Dependency Inversion Principle. Pentru aceasta, am folosit CDI (Context Dependency Injection) cu implementarea Weld. Am utilizat un `EntityManagerFactory` și i-am aplicat un Singleton design pattern. Prin intermediul acelui singleton, obținem un `entityManager`, necesar injectării variabilelor adnotate

cu `@Inject`. La conectarea cu baza de date, am adăugat o nouă dependență în pom-ul din `FashionStoreCore`, `GSON` (Google JSON) , pentru a putea crea un constructor ce primește ca parametru un JSON sub forma unui string. În date SQL am utilizat `JPA`(Java Persistence API) cu `Hibernate`.

Concluzii și modalități de extindere a aplicației:

Am început acest proiect cu intenția de a oferi utilizatorilor (predominant femeilor) o experiență cât mai plăcută pe un site unde își pot găsi ținuta completă pentru o nuntă de vis.

Am putea, pe viitor, să trimitem cumpărătorului pe adresa de e-mail informații despre statusul comenzii sau newsletter despre apariția unor noi produse în stoc.

De asemenea putem face o secțiune unde clienții pot lăsa review-uri despre produsele cumpărate.

Bibliografie:

- [1] Alexandrescu Adrian, `BookManager` [Online], Disponibil la adresa: <https://bitbucket.org/avalexandrescu/bookmanager.git>, Accesat: 2019.
- [2] Alexandrescu Adrian, `BookManager` [Online], Disponibil la adresa: <https://bitbucket.org/avalexandrescu/bookmanager.git>, Accesat: 2019.
- [3] Tutorialspoint, `JSP - Form Processing` [Online], Disponibil la adresa: https://www.tutorialspoint.com/jsp/jsp_form_processing.htm, Accesat: 2019.
- [4] W3schools, `Font Awesome Introduction` [Online], Disponibil la adresa: https://www.w3schools.com/icons/fontawesome_icons_intro.asp, Accesat: 2019.
- [5] W3schools, `jQuery Tutorial` [Online], Disponibil la adresa: <https://www.w3schools.com/jquery/default.asp>, Accesat: 2018.
- [6] Joakim Erdfelt, `How to upload files to server using JSP/Servlet` [Online], Disponibil la adresa: <https://stackoverflow.com/questions/2422468/how-to-upload-files-to-server-using-jsp-servlet>, Accesat: 2019.
- [7] Apache, `Using FileUpload` [Online], Disponibil la adresa: <https://commons.apache.org/proper/commons-fileupload/using.html>, Accesat: 2019.