MAGAZIN

Document de analiza si design

Numar echipa: 2

Studenti:

* Nicoară Adriana Maria
* Georgiu Georgiana Domnica

**Grupa: 302310**

Contents

[I Specificatia proiectului 3](#_Toc27236737)

[1.1 Diagrama Domain Model 3](#_Toc27236738)

[II Modelul Use-Case 4](#_Toc27236739)

[2.1 Identificarea Use-Case-uri 4](#_Toc27236740)

[2.2 Diagrama UML Use-Case 5](#_Toc27236741)

[III Design arhitectural 6](#_Toc27236742)

[3.1 Arhitectura conceptuala 6](#_Toc27236743)

[3.2 Diagrama de pachete 7](#_Toc27236744)

[3.3 Diagrama de clase 8](#_Toc27236745)

[3.4 Diagrama bazei de date 8](#_Toc27236746)

[3.5 Diagrame de secventa 8](#_Toc27236747)

[3.6 Diagrame de activitati 9](#_Toc27236748)

[IV Specificatii suplimentare 10](#_Toc27236749)

[4.1 Specificatii non-functionale 11](#_Toc27236750)

[4.2 Constrangeri de design 12](#_Toc27236751)

[V Testare 12](#_Toc27236752)

[5.1 Testareafunctionalitatiiaplicatiei 12](#_Toc27236753)

[5.2 Dezvolatariulterioare 12](#_Toc27236754)

[VI Bibliografie 12](#_Toc27236755)

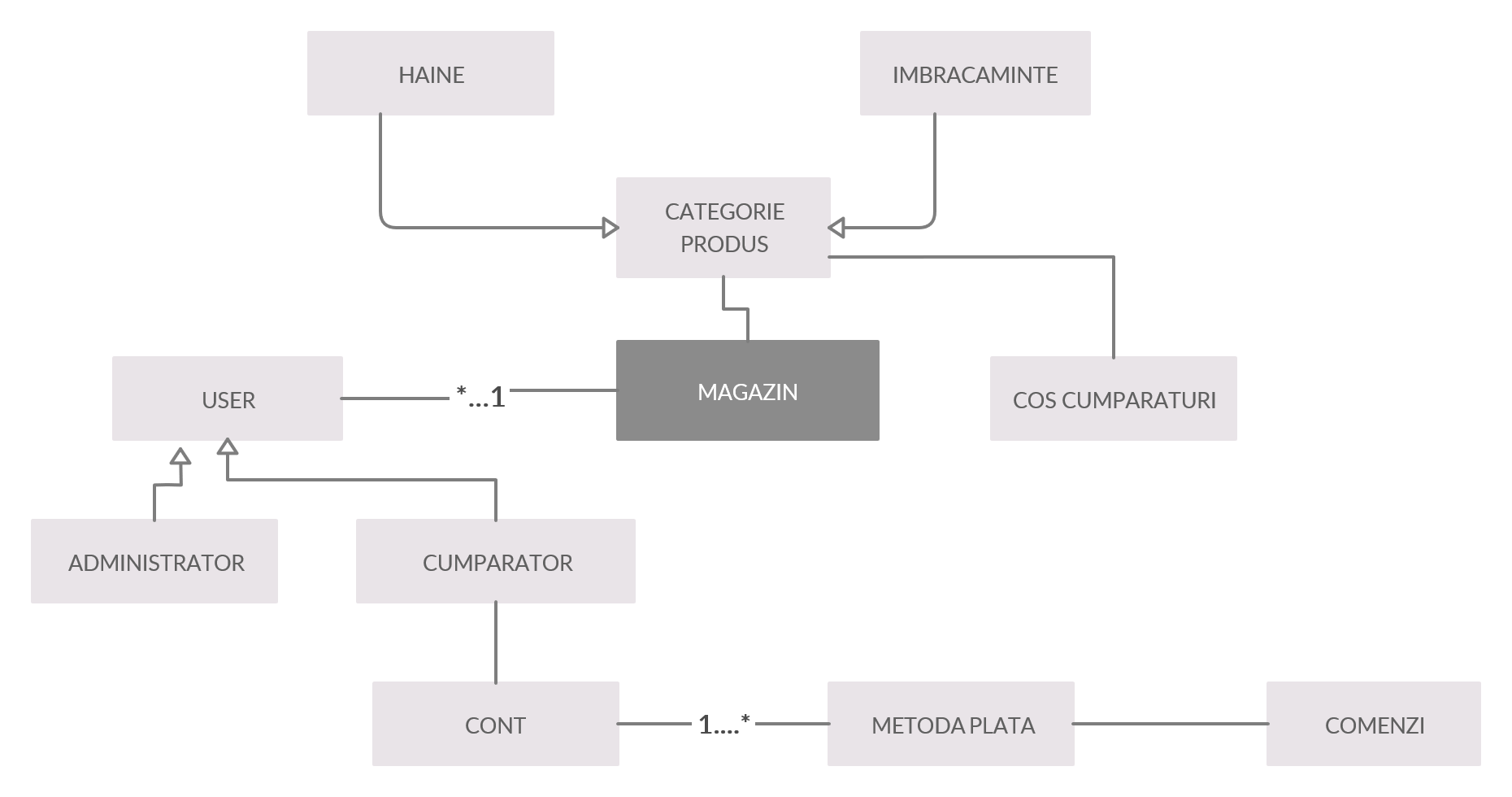
# I Specificatiaproiectului

*Proiectul consta in crearea unui magazin de haine si incaltaminte. La acest magazin au acces mai multi clienti si administratorul magazinului.*

*Fiecare client poate verifica daca un produs este sau nu in stoc, il poate adauga in cosul de cumparaturi , iar mai apoi il poate achizitiona finalizand comanda prin selectarea metodei de plata.*

*Adinistratorul poate face anumite modificari ale produselor , cum ar fii: modificare pret produs, modificare stoc produs.*

## 1.1 DiagramaDomain Model

**

# II Modelul Use-Case

*Principalii utilizatori ai aplicatiei create sunt : admin si client. Admin-ul are dreptul de a adauga/modifica/sterge un produs, de a se inregistra in baza de date, de a vizualiza toti ceilalti utilizatori si toate produsele. Clientul se poate inregistra in baza de date , poate vizualiza produsele, poate selecta un produs si il poate adauga in cosul sau de cumparaturi si poate finzaliza o comanda.*

## 2.1 Identificarea Use-Case-uri

***NumeUse case****:Adaugare unui nou produs.*

***Actor principal****: Admin .*

***Scenariul principal de success****: Adminul se inregistreaza pe pagina magazinului , apoi acesta are drepturi de a adauga un nou produs. Pentru adaugarea unui nou produs, acesta completeaza datele produsului iar apoi il adauga in stoc . Produsul apoi poate fi vizualizat si comandat de catre ceilalti useri ai magazinului.*

***Extensie:*** *Nu s-a reusit adaugarea unui nou produs deoarece datele adaugate au fost invalide .*

***NumeUse case****: Vizualizare produse magazin.*

***Actor principal****: Client .*

***Scenariul principal de success****: După ce utilizatorul se inregistreaza, acceseaza pagina cu produse a magazinului si le vizualizeaza.*

***Extensie****: Utilizatorul poate accesa cu succes pagina magazinului doar daca datele sale sunt valide.*

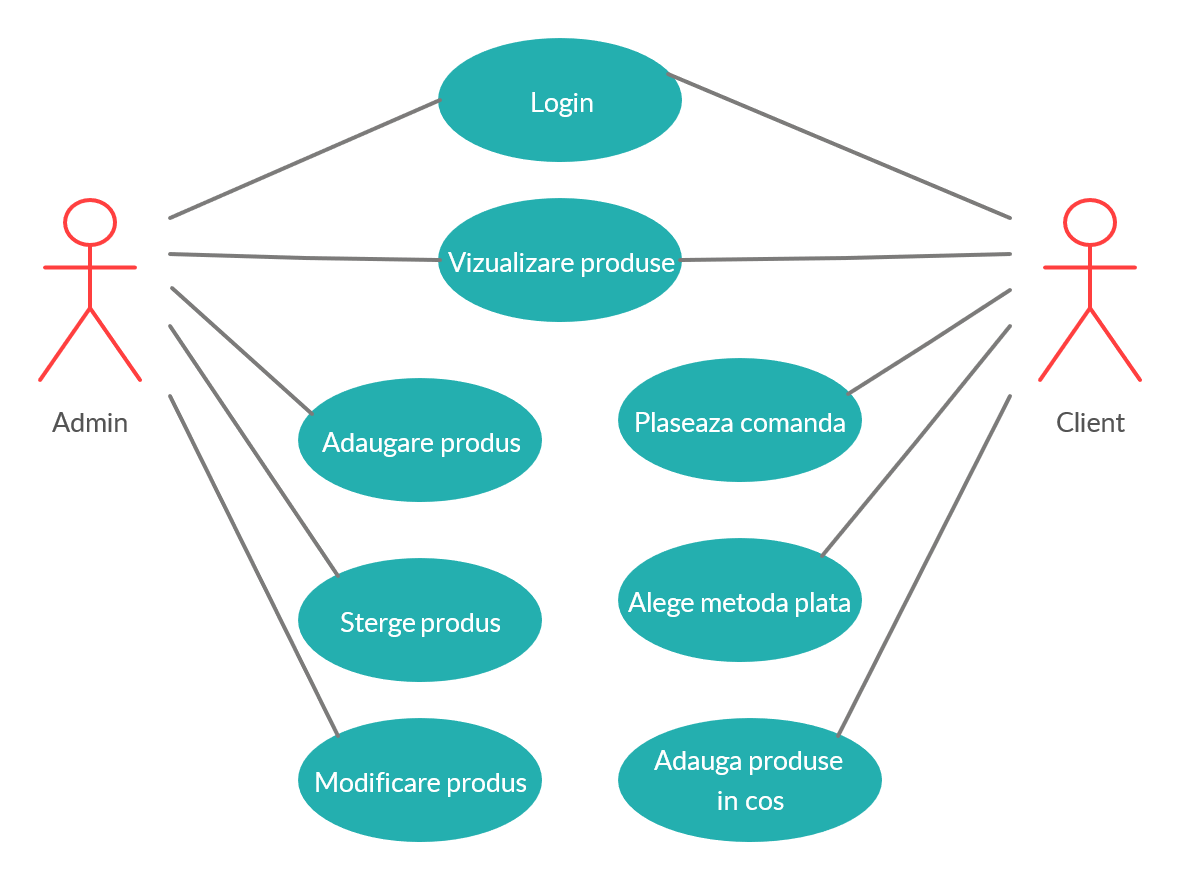
***NumeUse case****: Plasare comanda .*

***Actor principal:*** *Client .*

***Scenariul principal de success:*** *Clientul se inregistreaza pe pagina magazinului si vizualizeaza produsele .După ce clientul a selectat elemente pentru a achiziționa , va trebui sa le adauge in cosul de cumparaturi. Tot clientul va furniza informatii legate de metoda de plata si expediere. Clientul poate avea deja un cont cu informații de facturare și de expediere.*

***Extensie:*** *Comanda nefinalizata deoarece in momentul introduceii datelor pentru finalizarea comenzii , metoda de plata nu a fost selectata.*

## 2.2 Diagrama UML Use-Case

**

# III Design arhitectural

*Pentru acest proiect am folosit un pattern bazat pe mai multe layere, in care fiecare layer are o anumita responsabilitate si asigura anumite functionalitati pentru layerele superioare.*

## 3.1 Arhitecturaconceptuala

*Arhitectura bazata pe layere sau pe mai multe nivele, nu are un numar predefinit de niele, insa urmatoarel, pe care le-am folosit si in aceasta aplicatie, se intalesc cel mai des:*

* *Presentation layer*
* *Service layer*
* *Repository layer*
* *Entity layer*

*Ideea este urmatoarea: utilizatorul instantiaza un anumito obiect in presentation layer,apasand de exemplu pe un buton. Presentation layer apeleaza layer-ul de sub el, in acest caz service layer. Acesta apeleaza repository layer iar acesta, la randul lui entity layer. Fiecare layer este dependent de layer-ul inferior lui, putand sa il apeleze doar pe acesta.*

*Presentation layer contine partea de design grafic a aplicatiei si partea care se ocupa de interactiunea cu utilizatorul, acesta fiin controllerul.*

*Service layer apeleaza doar metodele din repository si prinde si trateaza anumite exceptii, cum ar fi in cazul introducerii unor date gresite de catre un client.*

*Repository layer implementeaza metodele prin care se lucreaza cu baza de date.*

*Entity layer are mai multe clase care reprezinta tocmai tabelele din baza de date.*

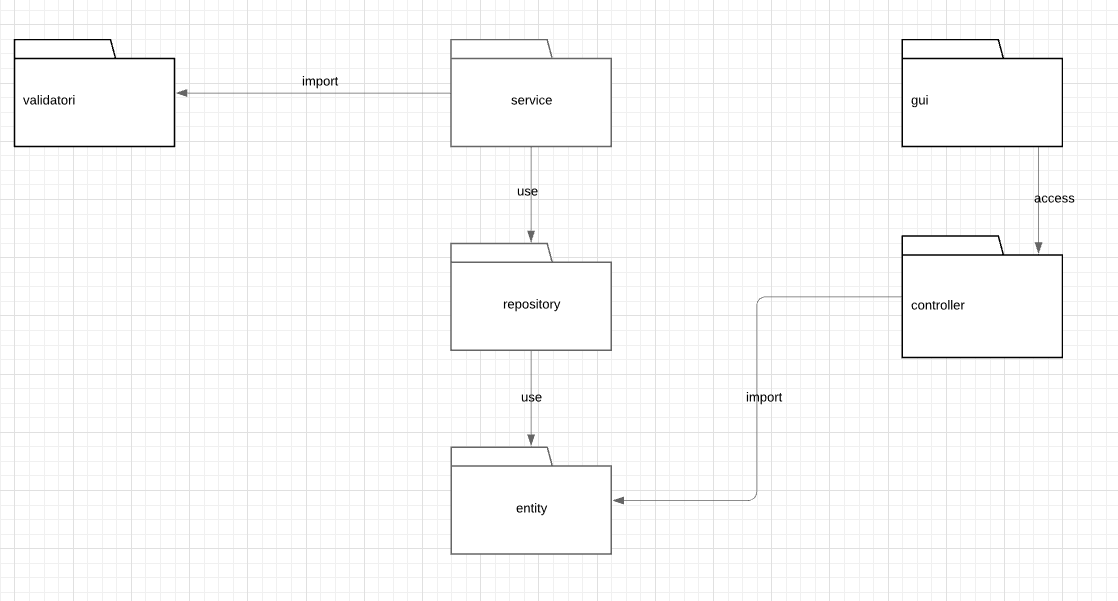
*Aceasta aplicatie este una desktop, lucreaza cu baza de date. Conexiunea la baza de date se face prin utilizarea maven si hibernate. Datele de acces la baza de date pe langa alte informatii se scriu intr-un fisier denumit „persistence.xml” .*

*Fisierul „pom.xml” contine informatii despre proiect si configuratii importante, cum ar fi dependinte sau plugin-uri, utilizate de Maven pentru a construi proiectul.*

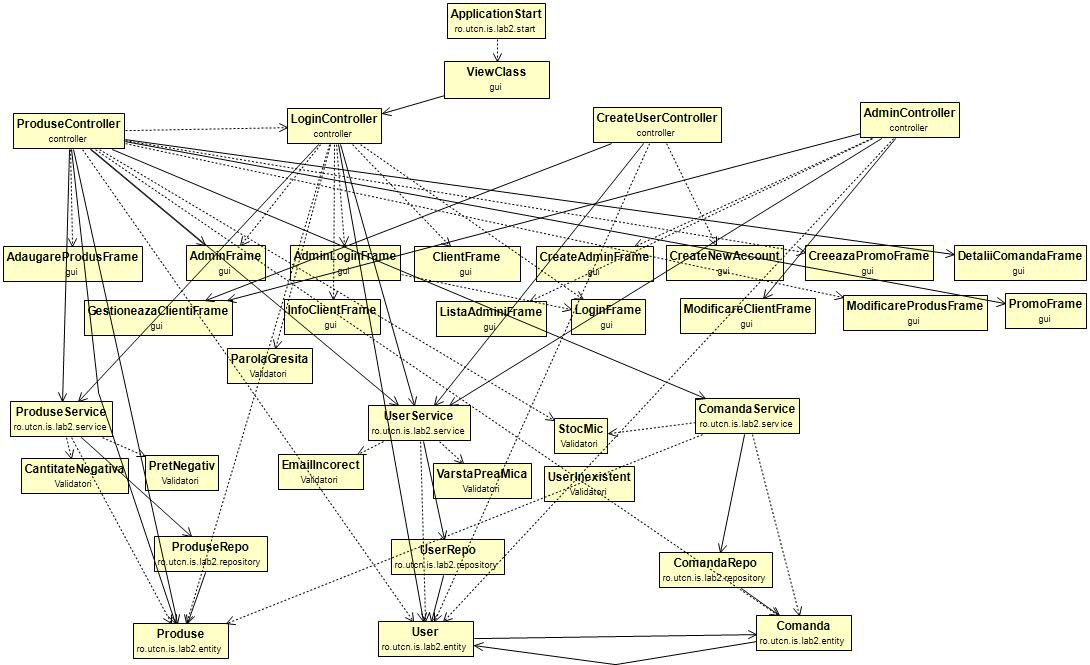
*In acest proiect se intalnesc 2 componente, Clientul sau utilizatorul si server-ul prin care se face comunica cu baza de date.*

## 

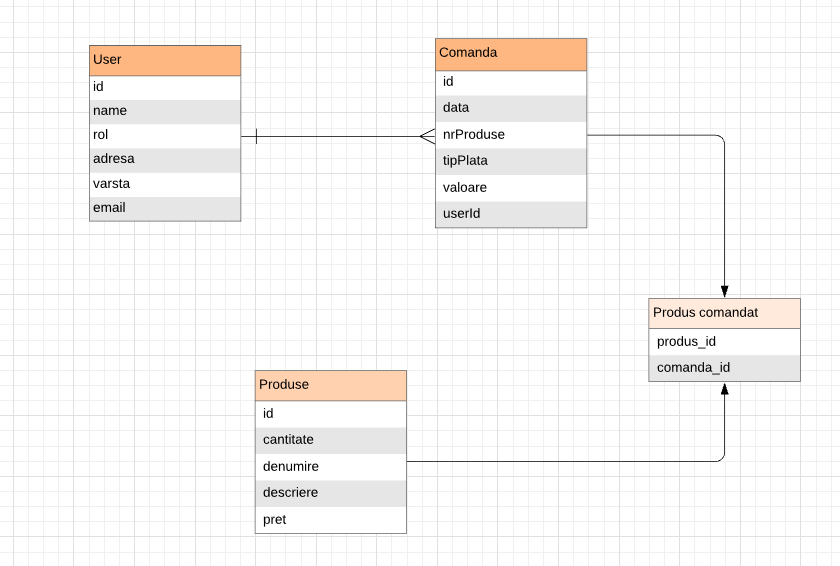
## 3.2 Diagrama de pachete



## 3.3 Diagrama de clase

**

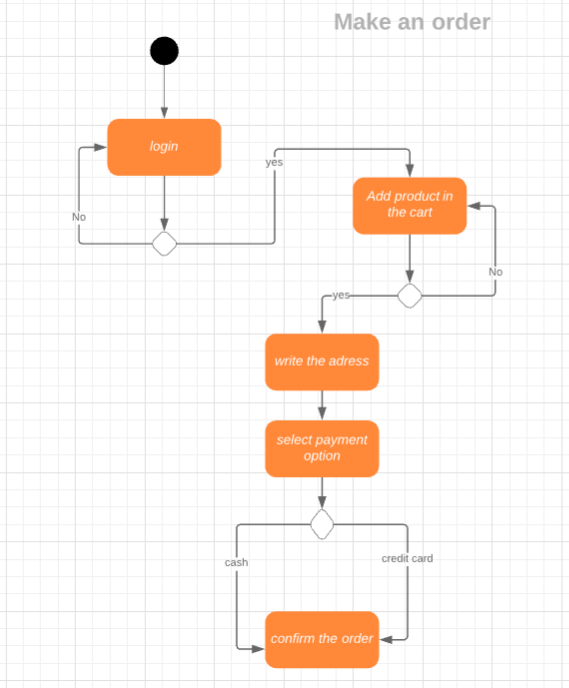
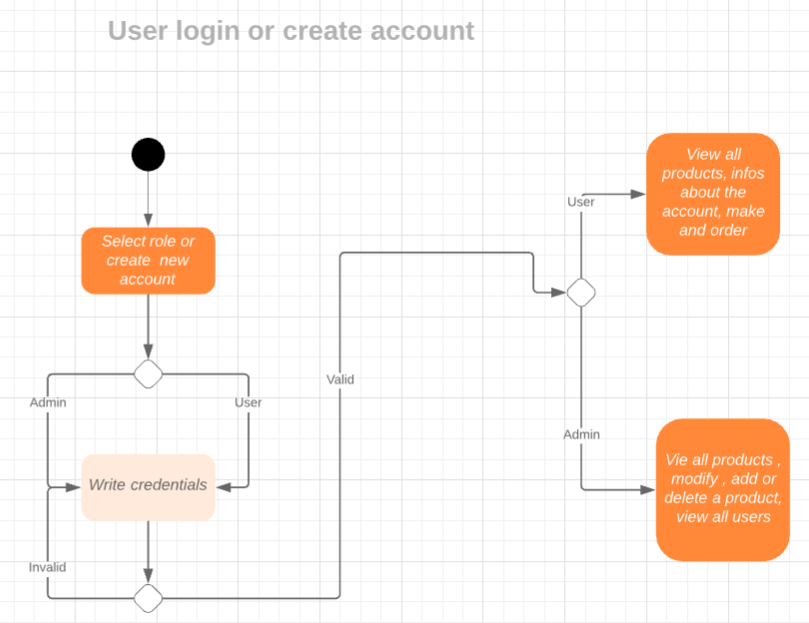
## 3.4 Diagrama bazei de date

**

## 3.5 Diagrame de secventa

*< (Sequence Diagram)/>*

## 3.6 Diagrame de activitati



# IV Specificatiisuplimentare

Acest magazin este unul nou aparut deci este la fel si pentru aplicatie. Deoarece se doreste cea mai buna interactiune dintre clienti si aplicatie, aceasta va avea update-uri periodice care vor rezolva anumite probleme descoperite ulterior sau care au fost semnalate de catre clienti.

## 4.1 Specificatii non-functionale

* ***Utilizabilitate*** – aceasta aplicatie a magazinului este usor de folosit pentru toate tipurile de persoane deoarece fiecare pas este ghidat in interfata grafica. Aceasta contine butoane cu comenzi clare care pot fi efecuate. De exemplu daca se doreste logarea, se apasa butonul login si se continua cu introducerea credentialelor in casutele denumite special pentru acest lucru, dupa care se apasa un buton ok.
* ***Portabilitate*** – deocamdata este disponibila doar varianta desktop a aplicatiei, insa in viitor se doreste ca aceasta sa fie extinsa si sa fie valabila si pe android, ios sau web.
* ***Securitate*** – pentru aceasta prima versiune a aplicatiei, login-ul nu este securizat, insa acest lucru este printre primele care vor fi rezolvate la versiunile urmatoare
* ***Toleranta*** la erori – am incercat sa prindem si sa rezolvam toate erorile si exceptiile, insa este posibil sa apara erori noi neprevazute de noi, insa aplicatia ramane functionala, adica aceasta nu va avea un mesaj de „Not responding”
* ***Reutilizare*** – anumite parti de cod din aceasta aplicatie, cum ar fi metodele de create, delete, update, view all, pot fi reutilizate si pentru alte aplicatii care utilizeaza bazele de date. De asemenea si design-ul arhitectural pe nivele este recomandat sa fie utilizat in toate aplicatiile care comunica cu baze de date
* ***Compatibilitatea platformei*** – aplicatia este scrisa in Java, deci aceasta poate fi rulata atat pe windows cat si pe linux sau alte platforme
* ***Internationalizare si localizare*** – aceasta prima versiune functioneaza doar pe teritoriul romaniei si are doar limba romana, insa deoarece acesta este un magazin nou aparut, se doreste exinderea sa chiar si international deci imediat ce se deschid magazine in alte tari, vor aparea versiunile de aplicatie aferente
* ***Open source*** – aceast aplicatie nu este open source, iar codul este vizibil si poate fi modificat doar de persoanele care au creato

## 4.2 Constrangeri de design

*< Se vadiscuta la laborator./>*

# V Testare

## 5.1 Testareafunctionalitatiiaplicatiei

Pentru testarea aplicatiei am introdus in baza de date mai multi utilizatori, clienti cat si admini. Am introdus de asemenea si produse. In continuare am testat fiecare functionalitate in parte, ne-am logat ca si client sau ca si admin, am creat conturi noi pentru ambele tipuri de utilizatori din aplicatie. Ca si client am facut o comanda noua, am vizualizat toate produsele, am vizualizat profilul personal. Ca si admin, am modificat produse, am adaugat produse noi, am creeat promotii, am sters produse, am vizualizat toate produsele si toti utilizatorii.

## 5.2 Dezvolatariulterioare

# VI Bibliografie

* https://examples.javacodegeeks.com/enterprise-java/hibernate/hibernate-crud-operations-tutorial/
* https://www.dineshonjava.com/hibernate/crud-operations-using-hibernate-3/