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 8](#_Toc27236748)

[IV Specificatii suplimentare 9](#_Toc27236749)

[4.1 Specificatii non-functionale 9](#_Toc27236750)

[4.2 Constrangeri de design 9](#_Toc27236751)

[V Testare 9](#_Toc27236752)

[5.1 Testareafunctionalitatiiaplicatiei 9](#_Toc27236753)

[5.2 Dezvolatariulterioare 10](#_Toc27236754)

[VI Bibliografie 10](#_Toc27236755)

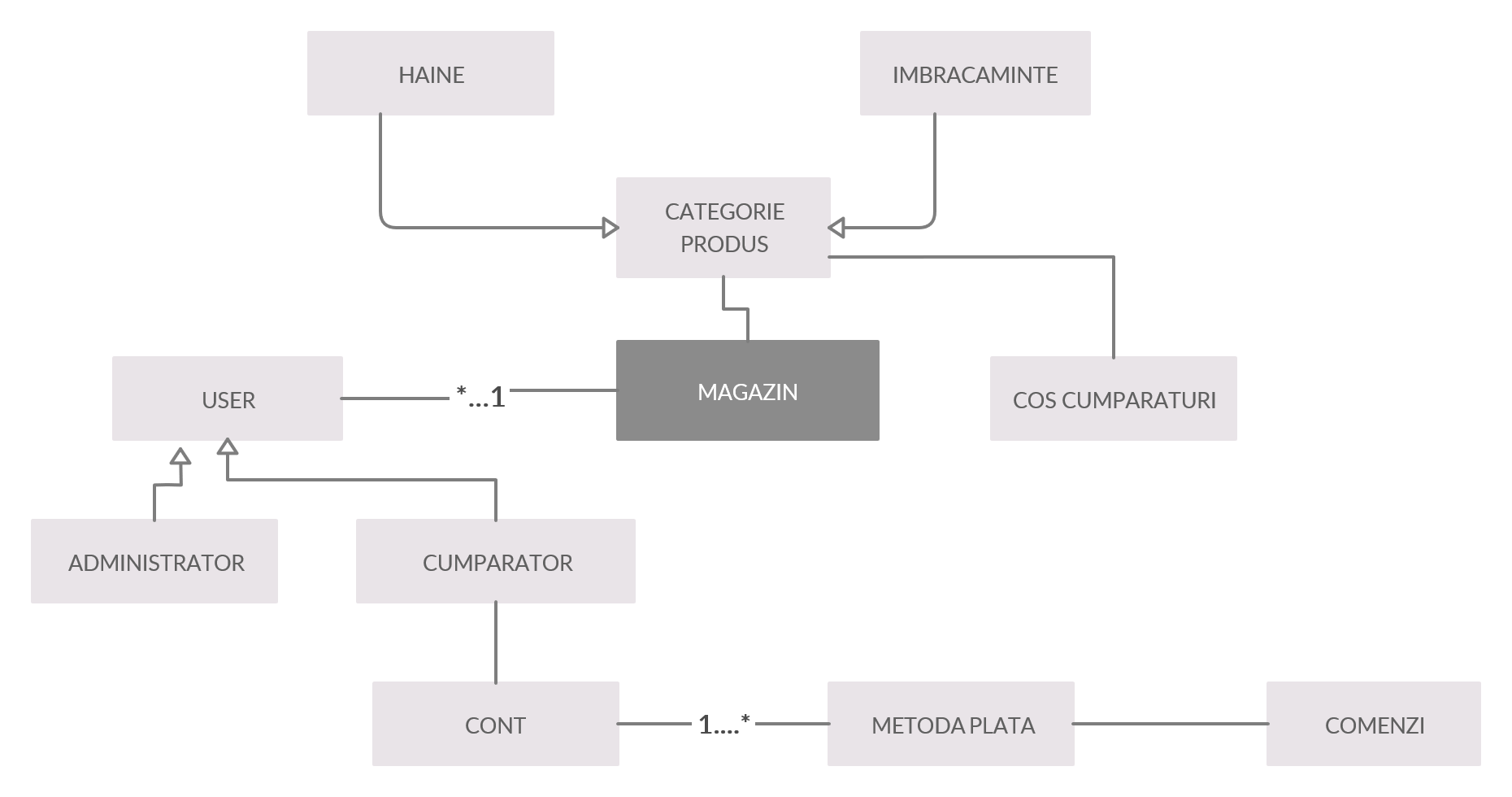
# I Specificatia proiectului

*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 Diagrama Domain Model

**

# II Modelul Use-Case

*< Se vascrie o mica introducere./>*

## 2.1 Identificarea Use-Case-uri

*<Aici se vorprezenta 3-4 use-case-urimaiimportante din applicatiedupaurmatorul model*

***NumeUse case: <Nume use-case>.***

***Nivel: < User-Goal, Subfunction, Summary > .***

***Actor principal: <Actorulscenariului> .***

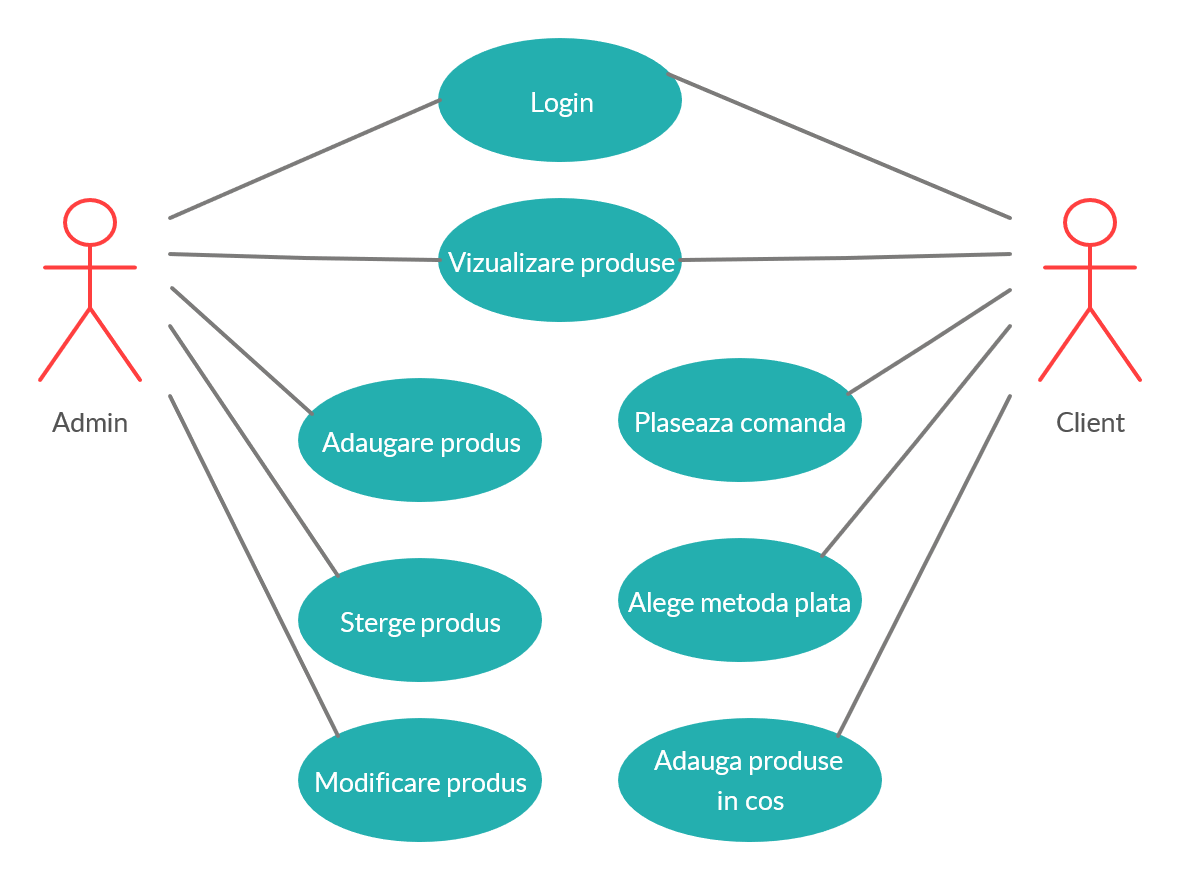
***Scenariul principal de success: <Descrieredetaliata a scenariului>.***

***Extensie: <Uncaz particular al scenariului, fie pozitivsaunegativ>***

*Se vadiscuta la laboratormaidetaliat.*

*/>*

## 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 Arhitectura conceptuala

*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.*

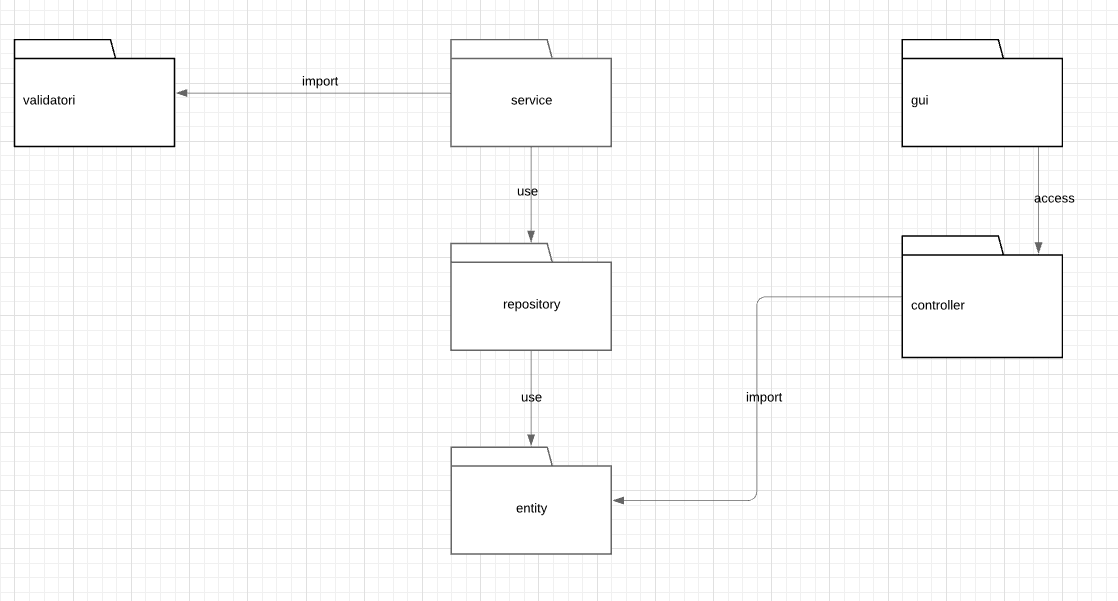
*Entity layer are mai multe clase care reprezinta tocmai tabelele din baza de date.*

*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.*

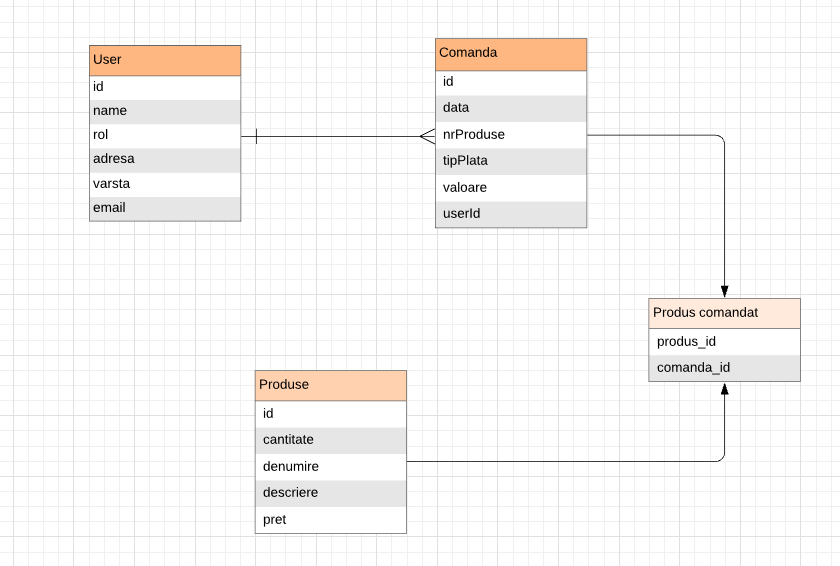
*In acest proiect se intalnesc 2 componente, Clientul sau utilizatorul si server-ul prin care se face comunica cu baza de date.*

## 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

*< (Activity Diagram)/>*

# IV Specificatii suplimentare

*< Se vascrie o mica introducere./>*

## 4.1 Specificatii non-functionale

*<Specificatiile non-functionale ale aplicatiei. Se vadiscuta la laborator./>*

## 4.2 Constrangeri de design

*< Se vadiscuta la laborator./>*

# V Testare

*< Se vadiscuta la laborator./>*

## 5.1 Testareafunctionalitatiiaplicatiei

## 5.2 Dezvolatariulterioare

# VI Bibliografie