

***E-commerce Website***

Name: Marinescu Roxana

Group: 30236

Table of Contents

[Deliverable 1 3](#_Toc64843130)

[Project Specification 3](#_Toc64843131)

[Functional Requirements 3](#_Toc64843132)

[Use Case Model 3](#_Toc64843133)

[Use Cases Identification 3](#_Toc64843134)

[UML Use Case Diagrams 3](#_Toc64843135)

[Supplementary Specification 3](#_Toc64843136)

[Non-functional Requirements 3](#_Toc64843137)

[Design Constraints 3](#_Toc64843138)

[Glossary 3](#_Toc64843139)

[Deliverable 2 3](#_Toc64843140)

[Domain Model 3](#_Toc64843141)

[Architectural Design 4](#_Toc64843142)

[Conceptual Architecture 4](#_Toc64843143)

[Package Design 4](#_Toc64843144)

[Component and Deployment Diagram 4](#_Toc64843145)

[Deliverable 3 4](#_Toc64843146)

[Design Model 4](#_Toc64843147)

[Dynamic Behavior 4](#_Toc64843148)

[Class Diagram 4](#_Toc64843149)

[Data Model 4](#_Toc64843150)

[System Testing 4](#_Toc64843151)

[Future Improvements 4](#_Toc64843152)

[Conclusion 4](#_Toc64843153)

[Bibliography 4](#_Toc64843154)

# Deliverable 1

## Project Specification

Proiectul va reprezenta un site online(ex:Emag,Amazon) in care user-ul va avea posibilitatea sa comande dintr-o gama larga de categorii si produse. Site-ul web va fi realizat in Java si va avea in spate o baza de date realizata in MySQL.

## Functional Requirements

Cerintele functionale pentru proiect sunt urmatoarele:

* Operații de CRUD pe entități

Interfetele UserRepo, ProductRepo si CategoryRepo extind CrudRepository. Interfete UserService, ProductService si CategoryService definesc metodele pentru operatiile de delete ,update si create, iar clasele UserServiceImpl, ProductServiceImpl si CategoryServiceImpl implementeaza aceste interfete si metodele aferente folosind metodele din CRUD.

* Folosirea ORM

Am folosit adnotarile manyToMany, manyToOne, oneToOne si oneToMany pentru a crea relatiile corepunzatoare intre tabele.

* Acces user/admin

Utilizatorii pot fi admini sau customeri, lucru care este determinat cu ajutorul unui enum. Admin-ul are posibilitatea de a crea, sterge sau modifica produse,categorii sau useri. User-ul va putea sa isi faca un cont sau sa se logheze, dupa care va putea selecta categoriile sale de interes si adauga produse in cos pentru a plasa comanda.

* Unit tests

Am realizat teste pentru delete(in cazul in care exista user-ul sau nu exista), pentru update(in cazul in care exista user-ul sau nu exista) si pentru create, pentru toate cele trei service-uri.

## Use Case Model 1

### Use Cases Identification

Use-Case: Admin-ul se poate inregistra sau conecta la cont si poate gestioneaza informatiile despre produse,categorii,useri.  
Level: User Goal  
Primary Actor: Admin  
Main success scenario:

* + Primul pas e ca admin-ul sa se inregistreze/conecteze.
  + Apoi admin-ul o sa aiba posibilitatea de a adauga, sterge sau modifica informatii despre useri, produse sau categorii dupa caz.
  + Cand termina de facut modificarile trebuie sa le salveze.

Use-Case: Customer-ul se poate inregistra sau conecta la cont si poate vede produsele si categoriile afisate pe site, poate adauga produse in cos si plasa comanda.

Level: User Goal  
Primary Actor: Customer  
Main success scenario:

* + Primul pas e ca customer-ul sa se inregistreze/conecteze.
  + Apoi acesta poate vedea categoriile si selecta categoria care il intereseaza.
  + Dupa care acesta poate selecta produsele din categorie pentru a le vedea si/sau adauga in cos.
  + Acesta poate intra in cos si plasa o comada in cazul in care doreste.

### UML Use Case Diagrams

O imagine care conține diagramă

Descriere generată automat

## Supplementary Specification

### Non-functional Requirements

* + Securitate – Autentificarea prin username si parola reprezinta o masura de securitate importanta pentru proiect si de asemenea faptul ca doar userii care au rol de admin pot modifica baza de date.
  + Fiabilitatea - site-ul trebuie să funcționeze fără niciun bug pe tot parcursul utilizarii acestuia de catre customer, astfel customer-ul va simti ca site-ul este unul de incredere.
  + Scalabilitate – este o cerinta non-functionala importanta deoarece site-ul trebuie să evolueze, sa i se adauge cat mai multe caracteristici si numărul de utilizatorii care acceseaza site-ul sau care isi fac cont pot creste de asemenea.
  + Uzabilitatea - utilizatorul poate accesa toate informatiile despre produse de pe site(pret,brand,descriere,etc) si poate sa plaseze o comanda cu produsele de care are nevoie.

### Design Constraints

Pentru proiect am ales sa folosesc limbajul de programare Java. Ca pattern arhitectural am ales sa folosesc MVC (model-view-controller). Am folosit ca java based framework Spring si ca tool am folosit libraria Lombok. Datele sunt tinute in baza de date din MySQL.

## Glossary

# Deliverable 2

## Domain Model

## Architectural Design

### Conceptual Architecture

[Define the system’s conceptual architecture; use an architectural style and pattern - highlight its use and motivate your choice.]

### Package Design

[Create a package diagram]

### Component and Deployment Diagram

[Create the component and deployment diagrams.]

# Deliverable 3

## Design Model

### Dynamic Behavior

[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]

### Class Diagram

[Create the UML class diagram; apply GoF patterns and motivate your choice]

## Data Model

[Create the data model for the system.]

# System Testing

[Describe the testing methides and some test cases.]

# Future Improvements

[Present some features that apply to the application scope.]

# Conclusion

# Bibliography