



# eCommerce website cu produse apicole

APICultor-ul

Name: Cociubei Mihai  
Group:30237

## Table of Contents

<b>Deliverable 1 .....</b>	<b>3</b>
<b>Project Specification .....</b>	<b>3</b>
<b>Functional Requirements .....</b>	<b>3</b>
<b>Use Case Model 1.....</b>	<b>3</b>
Use Cases Identification .....	3
<b>Use case Model 2.....</b>	<b>4</b>
Use Cases Identification .....	4
<b>Use case Model 3.....</b>	<b>4</b>
Use Cases Identification .....	4
UML Use Case Diagrams .....	4
<b>Supplementary Specification .....</b>	<b>5</b>
Non-functional Requirements.....	5
Design Constraints .....	6
<b>Glossary.....</b>	<b>6</b>
<b>Deliverable 2 .....</b>	<b>6</b>
<b>Domain Model.....</b>	<b>6</b>
<b>Architectural Design.....</b>	<b>7</b>
Conceptual Architecture .....	7
Package Design.....	7
Component and Deployment Diagram .....	8
<b>Deliverable 3 .....</b>	<b>9</b>
<b>Design Model.....</b>	<b>9</b>
Dynamic Behavior .....	9
Class Diagram .....	9
<b>Data Model.....</b>	<b>9</b>
<b>System Testing .....</b>	<b>9</b>
<b>Future Improvements .....</b>	<b>9</b>
<b>Conclusion.....</b>	<b>9</b>
<b>Bibliography.....</b>	<b>9</b>

## Deliverable 1

### Project Specification

Obiectivul acestui proiect este de a realiza un magazin apicol online, unde producătorii din toată țara să poată găsi o piață de desfacere. Magazinul apicol online va ușura distribuția mărfurilor de la producător spre consumator. Un alt avantaj ar fi, ușurința găsirii produselor dorite. Am utilizat Java, MySQL, Spring pentru partea de back end, iar pentru front end React.

### Functional Requirements

Site-ul web într-o variantă inițială va avea trei tipuri de utilizatori (user, vânzător și admin) cu posibilitatea extinderii în viitor. Orice persoană se poate înregistra și autentifica pe site. În momentul înregistrării pe site vei fi întrebat dacă vrei să devii vânzător sau doar utilizator.

Un user poate adăuga produse în coș, finaliza comenzi, lasă o recenzie unui vânzător, filtra produsele după diferite caracteristici.

Vânzătorul poate adăuga produse noi, modifică caracteristicile produselor existente și elimină produse.

Adminul are drepturi depline asupra întregii pagini web putând să vină atât în ajutorul clienților cât și a vânzătorilor. În cazul unor abuzuri poate elimina un vânzător sau ștergerea produselor neconforme.

## Use Case Model 1

### Use Cases Identification

Use-Case: Pune Add new product

Level: Sub function

Primary Actor: Vânzătorul

Main success scenario: Alege datele noului produs adăugă o poză.

Extensions: Se va afișa un mesaj ceva nu a funcționat cum trebuie iar vânzătorul va fi rugat să introducă informațiile unde a omis inițial.

## Use case Model 2

### Use Cases Identification

Use-Case: Căutare produs

Level: User goal

Primary Actor: Customer

Main success scenario: Găsește produsul pe care îl caută și e pe stoc

Extensions: Dă un mesaj vânzătorului dacă produsul nu e pe stoc pentru a afla dată la care produsul va fi disponibil în magazin.

## Use case Model 3

### Use Cases Identification

Use-Case: Comandă produs

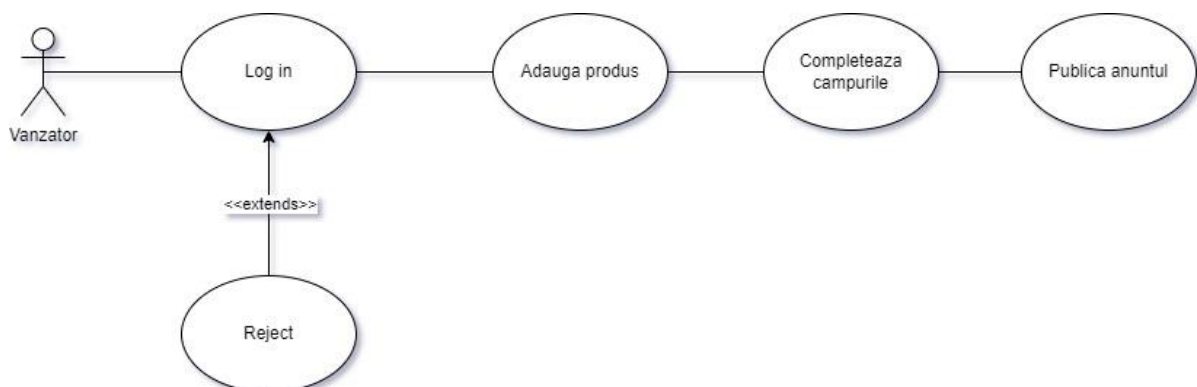
Level: User goal

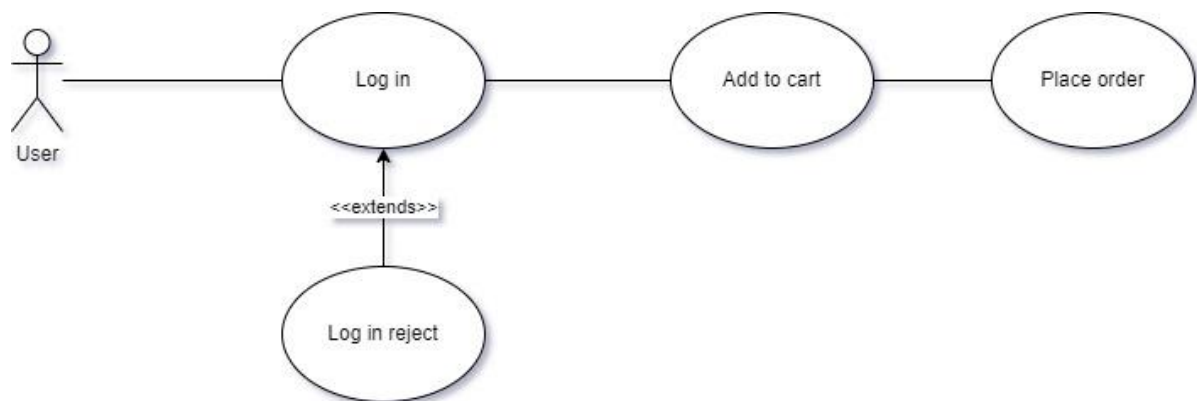
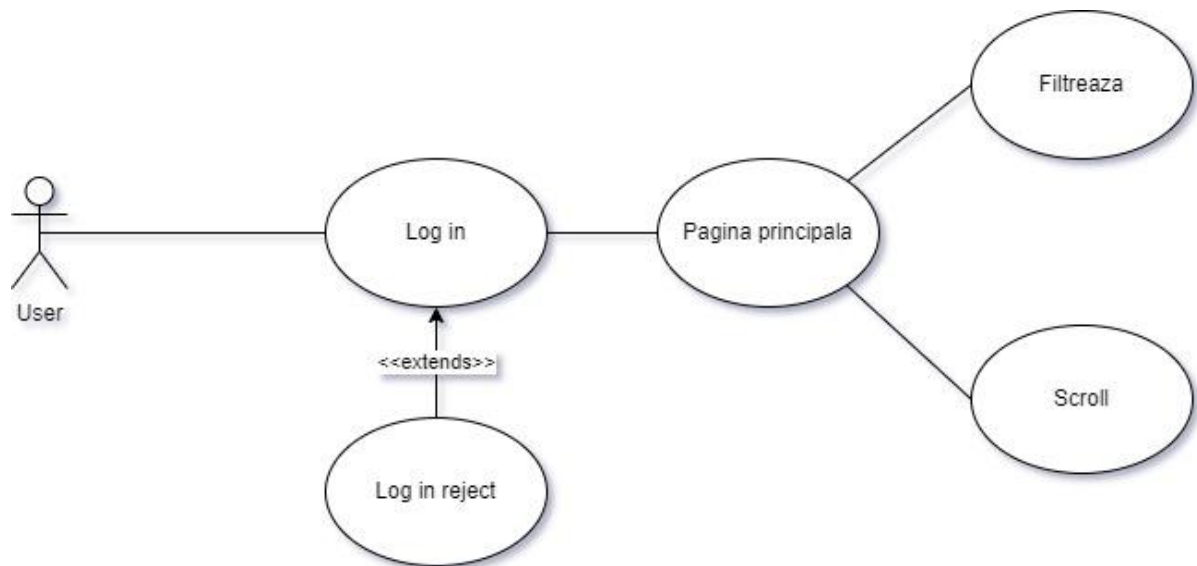
Primary Actor: Customer

Main success scenario: Utilizatorul intră pe contul sau adaugă produsul în coș și pune comandă.

Extensions: În cazul unui cont invalid va fi atenționat. Situația în care produsul nu se mai află pe stoc cumpărătorul va primi un mesaj de eroare indicând stoc insuficient.

## UML Use Case Diagrams





## Supplementary Specification

### Non-functional Requirements

**Scalability** – În viitor acest proiect va putea fi scalat prin adăugarea unor noi cazuri de utilizare și mai multe roluri.

**Usability**- Interfața siteului web va fi intuitivă astfel încât experiența utilizării să fie una plăcută și ușurată.

**Reliability**- Este o caracteristică foarte importantă pentru a ne asigura că sistemul va fi funcțional în parametri optimi.

**Availability**- Pagina web va fi disponibilă în toată lumea 24/7 având o echipă dedicată care se va asigura de acest lucru mareș.

Suportability- Alegand un design structurat și bine definit proiectul respecta cele mai înalte standarde.

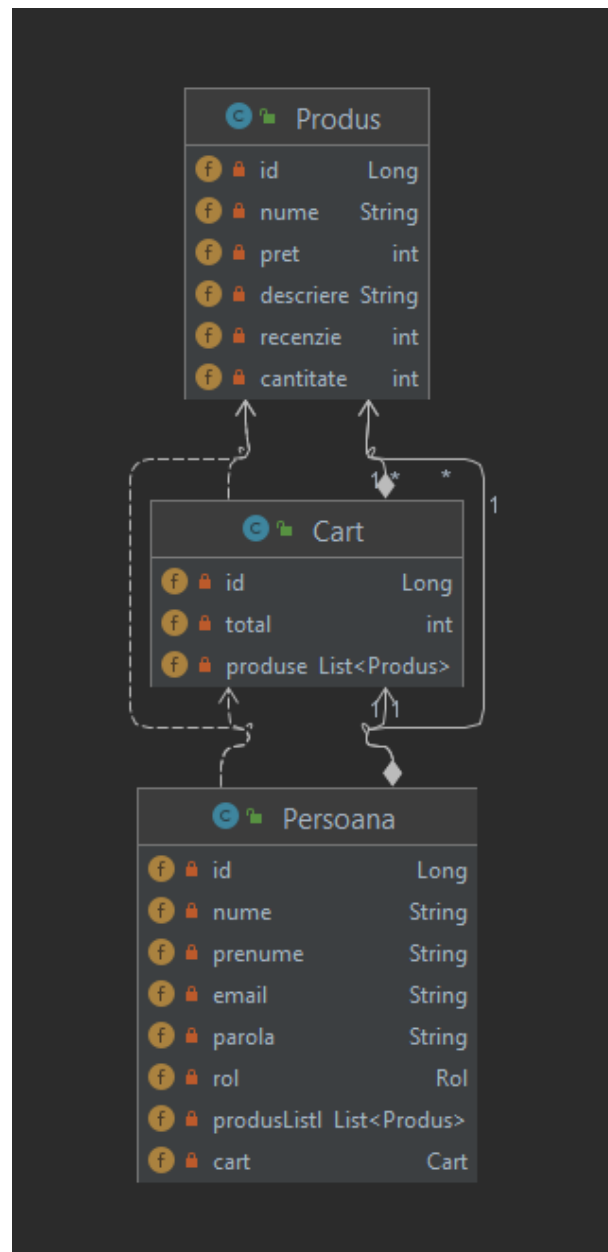
### Design Constraints

Toate clasele au fost structurate în pachete pentru o mai bună mentenanță și o ușoară scalare a proiectului. Partea de back end a fost realizată cu Spring și Java iar partea de front-end va fi implemenata cu React.

### Glossary

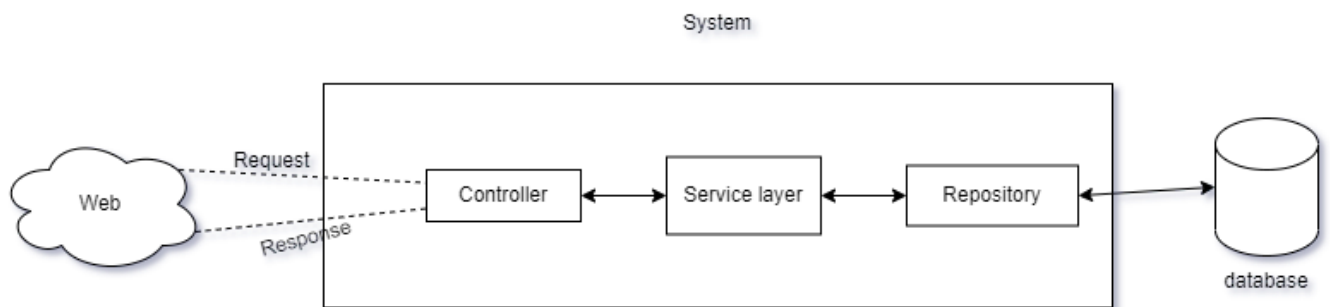
## Deliverable 2

### Domain Model

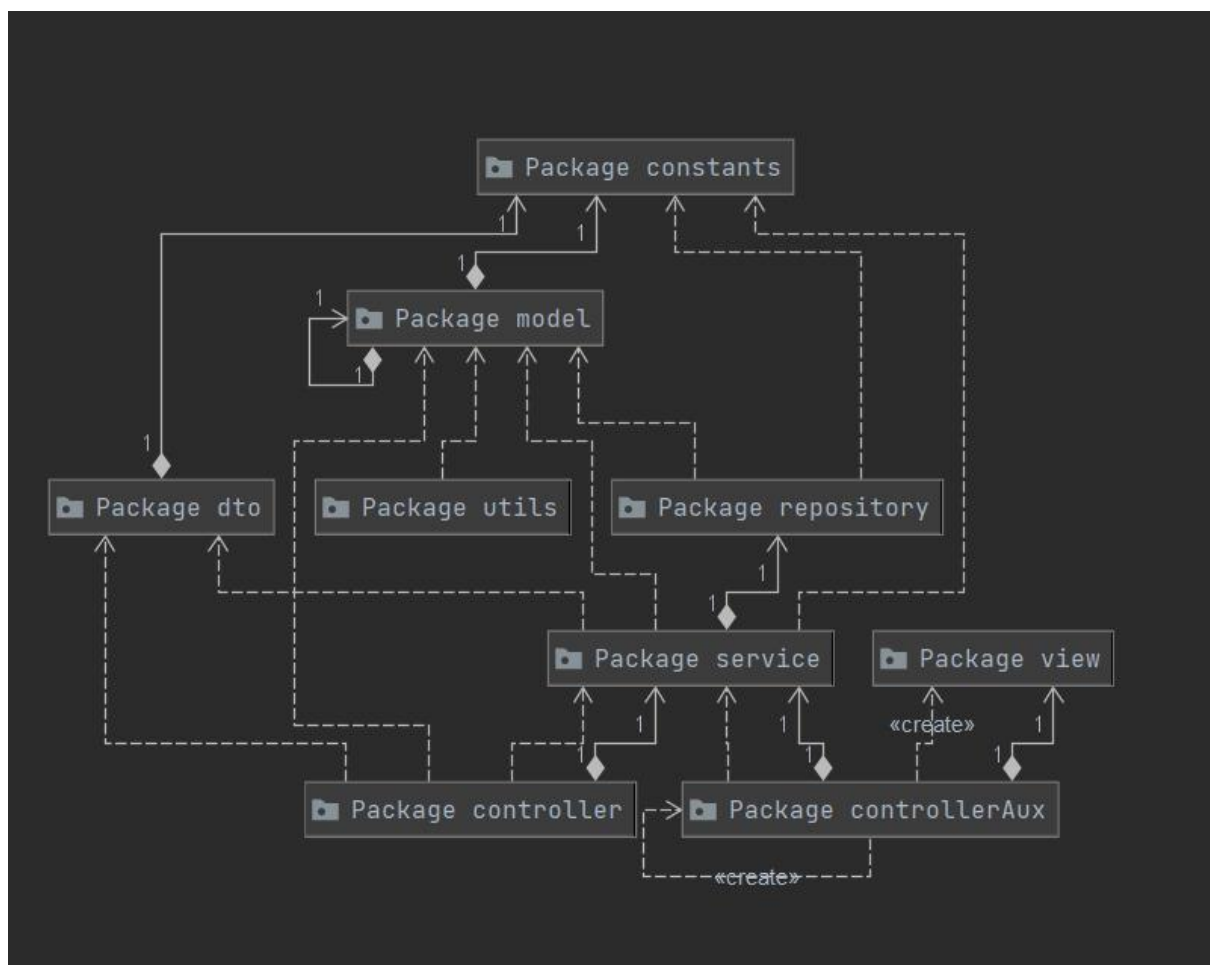


## Architectural Design

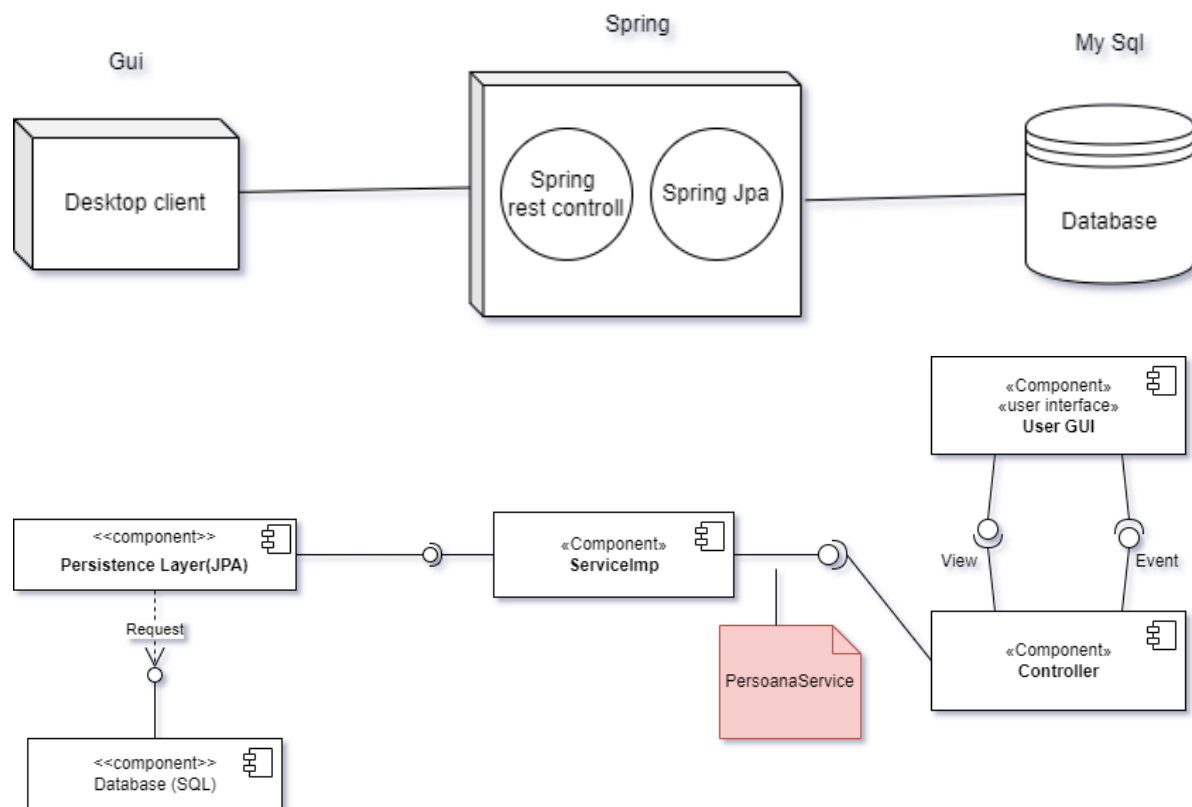
### Conceptual Architecture



## Package Design



## Component and Deployment Diagram





## 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 methods and some test cases.]*

### Future Improvements

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

### Conclusion

### Bibliography