

Apicola Store

Name: Munteanu Ionel Andrei

Deliverable 1

Project Specification

Proiectul "Magazin online apicol" este o platformă online care permite utilizatorilor să achiziționeze produse apicole de calitate, de la furnizori de încredere. Utilizatorii pot naviga prin produse, pot adăuga produse în coșul de cumpărături, pot finaliza comanda și pot verifica istoricul comenzilor. Administratorii pot gestiona produsele, pot actualiza informațiile despre acestea, pot șterge produse și pot gestiona comenzile.

Pentru a realiza acest proiect, vom folosi framework-ul Java Spring, care oferă un set de instrumente și biblioteci pentru a dezvolta rapid aplicații web și API-uri. Aplicația va fi construită folosind modelul de arhitectură MVC (Model-View-Controller), care separă logica aplicației în trei componente diferite, pentru a facilita dezvoltarea și întreținerea aplicației.

Aplicația va fi construită pe baza mai multor module, care vor fi responsabile pentru gestionarea diferitelor funcționalități ale magazinului online. Vom utiliza baza de date pentru a stoca informațiile despre utilizatori, produse și comenzile acestora. Utilizatorii vor putea căuta produse după nume sau după anumite criterii (preț, cantitate, etc.), pot adăuga produse în coșul de cumpărături, pot finaliza comanda și pot vizualiza istoricul comenzilor.Un alt aspect important al proiectului este securitatea. Pentru a proteja datele utilizatorilor și tranzacțiile, vom implementa un sistem de autentificare și autorizare, pentru a limita accesul la anumite funcționalități ale aplicației.

În concluzie, proiectul "Magazin online apicol" va fi o aplicație web modernă, construită folosind Java Spring, care oferă utilizatorilor o experiență de cumpărare plăcută și sigură pentru produse apicole de calitate.

Functional Requirements

* Utilizatorii trebuie să poată vizualiza produsele disponibile în magazinul online.
* Utilizatorii trebuie să poată adăuga produse în coșul de cumpărături și să poată finaliza comanda.
* Utilizatorii trebuie să poată vizualiza istoricul comenzilor și starea comenzii curente.
* Administratorii trebuie să poată adăuga noi produse în magazin, să le actualizeze sau să le șteargă.
* Administratorii trebuie să poată gestiona comenzile și să actualizeze starea comenzilor*.*

Use Case Model 1

Use Cases Identification

Utilizatorul vizualizează produsele disponibile.

Utilizatorul adaugă produse în coșul de cumpărături.

Utilizatorul finalizează comanda.

Utilizatorul vizualizează istoricul comenzilor.

Utilizatorul vizualizează starea comenzii curente.

Administratorul adaugă produse noi în magazin.

Administratorul actualizează informațiile despre produse.

Administratorul șterge produse din magazin.

Administratorul gestionează comenzile.

Administratorul actualizează starea comenzilor*.*

UML Use Case Diagrams

* UseCase Diagram pentru Administrator



* UseCase Diagram pentru Client



Supplementary Specification

Non-functional Requirements

* Securitatea: toate datele utilizatorilor și informațiile privind tranzacțiile trebuie să fie protejate.
* Scalabilitate: magazinul online trebuie să poată gestiona un număr mare de utilizatori și tranzacții simultane.
* Performanță: magazinul online trebuie să răspundă rapid la solicitările utilizatorilor.
* Disponibilitate: magazinul online trebuie să fie disponibil 24/7.

Design Constraints

* Accesul la baza de date trebuie să fie eficient și sigur, astfel încât să se evite pierderea datelor și să se minimizeze timpul de accesare a acestora.
* Structura bazei de date trebuie să fie bine definită și organizată pentru a permite operarea eficientă a operațiilor CRUD.
* Repozitoriul trebuie să fie proiectat pentru a permite adăugarea, citirea, actualizarea și ștergerea datelor într-un mod sigur și controlat.
* Repo-urile CRUD trebuie să fie proiectate pentru a se potrivi cu modelul de afaceri și logica aplicației.
* Magazinul online trebuie să fie ușor de utilizat și să ofere o experiență de utilizare plăcută.
* Design-ul trebuie să fie responsiv, astfel încât magazinul online să funcționeze bine pe toate dispozitivele.
* Magazinul online trebuie să fie integrat cu un sistem de plată securizat și cu un sistem de livrare a produselor.

Deliverable 2

Domain Model

*[Define the domain model and create the conceptual class diagrams]*

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