Clothing store

Web Application

Name:Mîrza Andrei

Table of Contents

[Project Specification 3](#_bookmark1)

[Functional Requirements 3](#_bookmark2)

[Use Case Model 3](#_bookmark3)

[Use Cases Identification 3](#_bookmark4)

UML Use Case Diagrams ………………………………………….3

[Supplementary Specification 4](#_bookmark5)

[Non-functional Requirements 4](#_bookmark6)

[Design Constraints 5](#_bookmark7)

[Glossary 5](#_bookmark8)

[Domain Model 5](#_bookmark10)

[Architectural Design 5](#_bookmark11)

[Conceptual Architecture 5](#_bookmark12)

[Package Design 6](#_bookmark13)

[Component and Deployment Diagram 6](#_bookmark14)

[Design Model 7](#_bookmark16)

[Dynamic Behavior 7](#_bookmark17)

[Class Diagram 8](#_bookmark18)

[Data Model 9](#_bookmark19)

[System Testing 9](#_bookmark20)

[Future Improvements 9](#_bookmark21)

## Project Specification

Se va crea o aplicatie web folosind ca limbaje de programare Java Spring si ReactJS care va gestiona activitatea unui magazine de haine online . Pentru a putea intra in aplicatie trebuie creat un cont si vom avea trei tipuri de utilizatori : client, administrator si angajat. Clientul poate sa vizualizeze produsele in functie de preferinte, acestea pot sa fie sortate dupa diferite categorii ( dupa pret , marime si marca).

Administratorul are dreptul sa adauge produse noi, sa le actualizeze, sa modifice preturile, de asemenea poate sa stearga un produs si gestioneaza baza de date . Iar angajatul poate sa vizualizeze comenzile puse de catre client si sa le gestioneze in cazul in care se gresesc adresele de livrare.

## Functional Requirements

Cerintele functionale specifica ceea ce trebuie sistemul sa faca. Pe langa cele prezentate in specificarea temei, mai sunt si urmatoarele : in momentul in care un client isi creeaza un cont nou datele trebuie sa fie stocate in baza de date, la fel si pentru administrator si angajat. Clientul are un cos de cumparaturi unde adauga produsele pe care doreste sa le comande . Plata produselor se poate face online sau in momentul in care se face livrarea. Administratorul poate modifca prosuele, adauga produse noi, sterge produsele si gestioneaza baza de date. Iar angajatul gestioneaza comenzile si este notificat de fiecare data cand o comanda este pusa, acesta poate modifica o comanda adica sa o anuleze sau sa modifice datele de livrare in cazul in care s-a produs o eroare.

## Use Case Model 1

Use Cases Identification

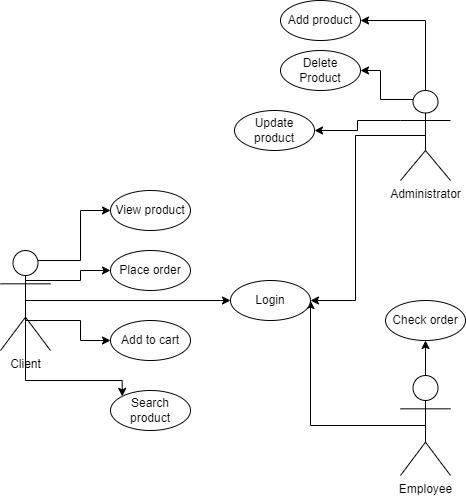
Use case: Prezinta cazurile de utilizare prin capabilitatile ficaruia dintre actori Level : user-goal level

Primary Actor : Actorul principal este determinat in functie de situatie, poate fi clientul, administratorul sau un angajat

Main success scenario : Un scenariu de succes ar fi cel in care un client pune o comanda si angajatul este notificat de acea comanda si nu trebuie sa intervina modificari.

Extensions : Erorile pot sa apara in oricare dintre situatii spre exemplu s-a comandat un produs care nu este pe stoc.

UML Use Case Diagrams



## Supplementary Specification

Non-functional Requirements

Cerintele non-functionale redau calitatile pe care trebuie sa le aiba produsul final. In general cerintele non-functionale ale sistemului sunt deseori numite calitati ale sistemului. Termeni pentru cerintele non-functionale sunt : constrangeri, attribute de calitate, scopuri de calitate si calitatea cerintelor serviciului. Cele 4 cerinte non-functionale din proiect sunt: securitatea(security), modificabilitatea(modificability), testabilitate(testability) si folosinta(usability).

* Securitatea ( security ) - este o masura a capacitatii sistemului de a rezista la folosirea necorespunzatoare mentinand calitatea serviciilor in ciuda acestei folosiri. Securitatea sistemului va fi implementata prin crearea unor autentificari pentru utilizatori prin intermediul unui user name si a unei parole.
* Modificabilitatea (modificability) – este modul in care sistemul se comporta in cazul in care se doreste modificarea sa. Se doreste o modificabilitate cat mai buna.
* Testabilitate ( testability ) – este proprietatea sistemului in cazul in care trece cu bine prin teste. Este o proprietate rezultata in mare parte din celelalte.
* Folosinta(usability) – se refera la folosirea sistemului de catre utilizator la final care doreste sa se acomodeze cu sistemul si implicit la modul in care se comporta sistemul.



Design Constraints

Aplicatia web are o baza de date cat mai optimizata cu cat mai putine query- uri, de aemenea are si constrangeri de securitate care determina modul in care trebuie protejat continutul web. Aceste proprietati asociaza constrangeri de securitate cu una sau mai multe colectii de resurse web. De asemenea calitatea aplicatiei trebuie sa fie masurata atat din perspectiva tehnica cat si din punct de vedere commercial precum claritatea codului, mentabilitatea, performanta si satisfactia utilizatorilor. O alta constrangere este limbajul de programare si anume Java care este un limbaj orientat pe obiecte si de asemenea s-a folosit ca framework Spring boot . Ca metoda de testare avem testarea unitare cu Junit.

## Glossary

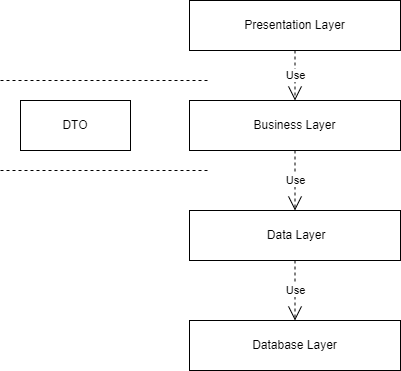
Spring framework – este un cadru de aplicatii open-source care ofera suport de infrastructura pentru dezvoltarea aplicatiilor Java.

JUnit – este un cadru de testare unitara pentru limbajul de programare Java. Joaca un rol in dezvoltarea bazata pe testare si este o familie de cadre de testare unitara cunoscuta colectiv sub numele de xUnit.

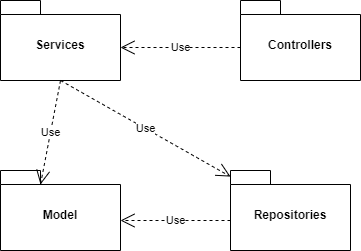
## Deliverable 2 Domain Model

## Architectural Design

Conceptual Architecture

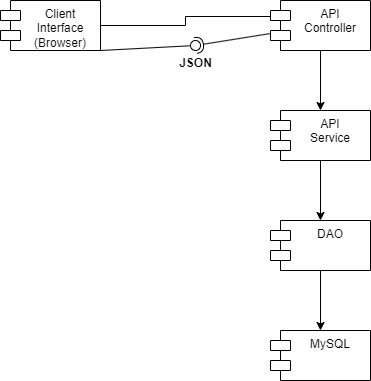


Package Design

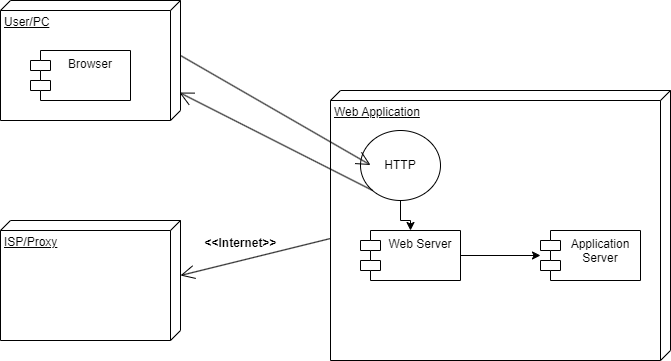


Component and Deployment Diagram

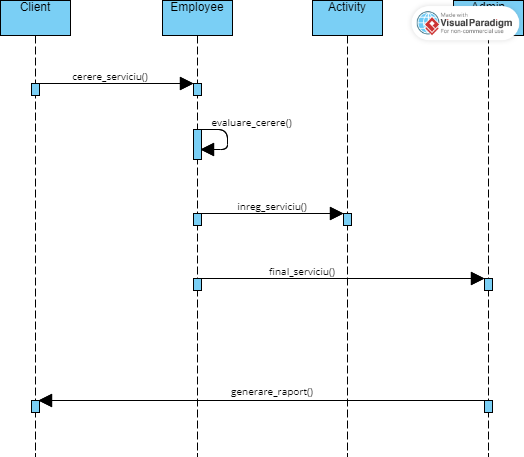
Component Diagram:



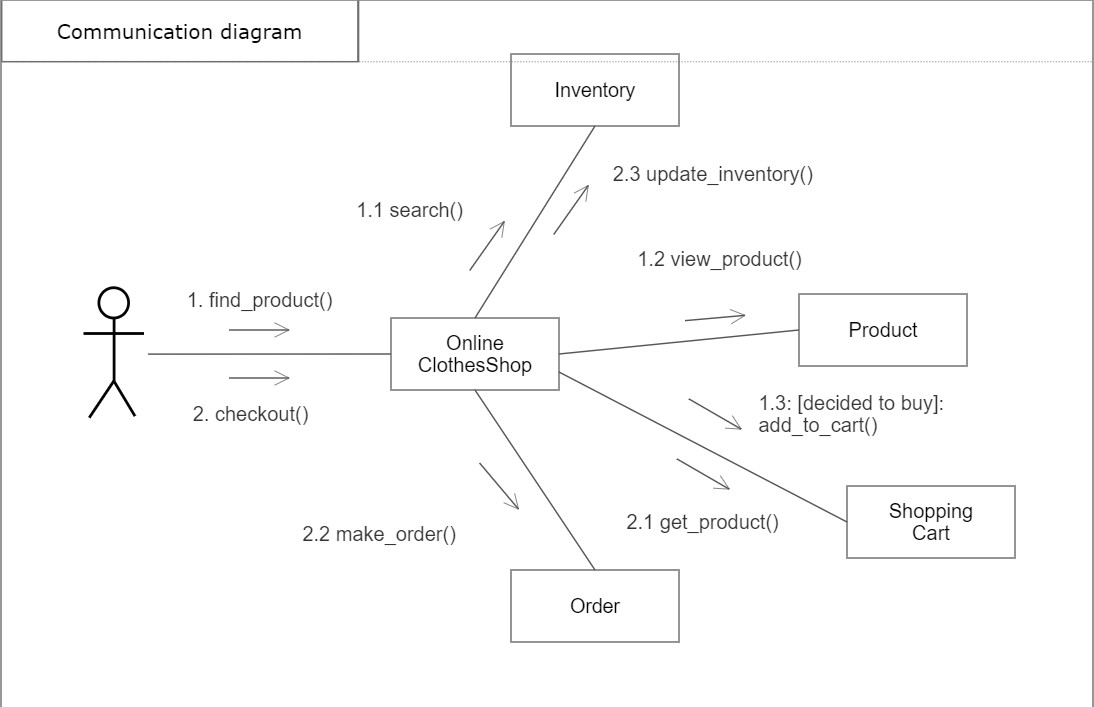
Deployment Diagram :



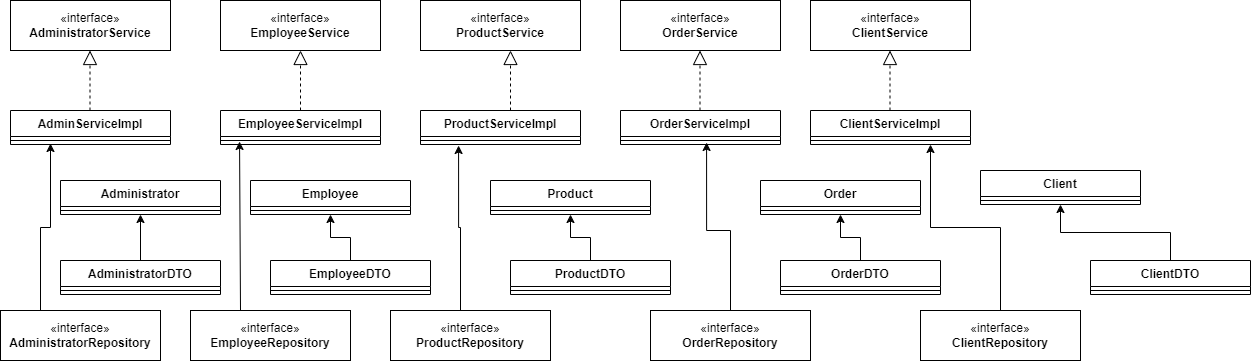
Deliverable 3 Design Model Dynamic Behavior Sequence diagram :



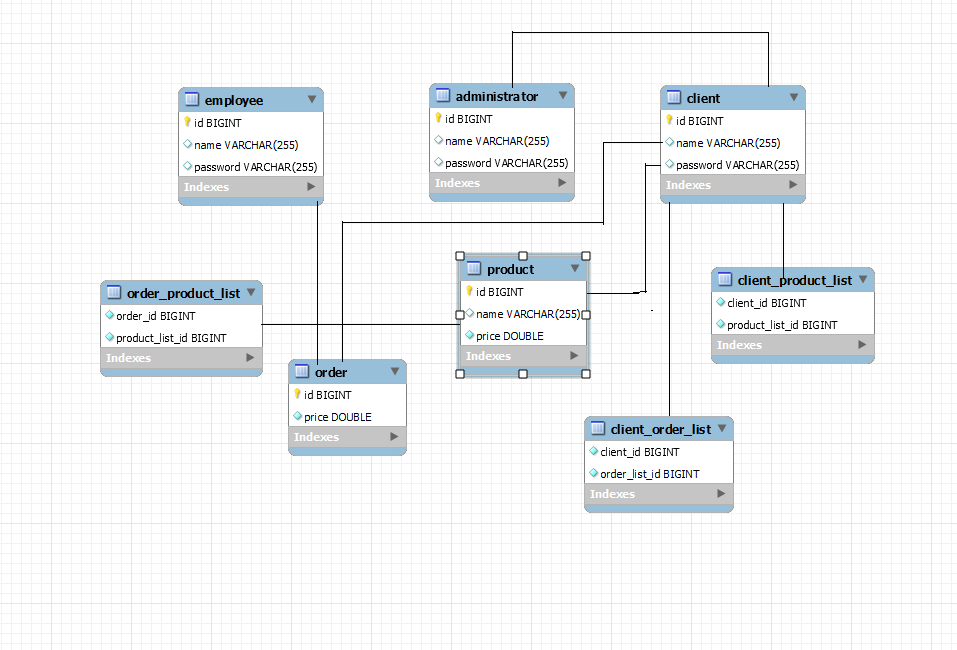
Communication diagram:



Class Diagram



## Data Model



# System Testing

Sistemul va fi testat prin simularea unor utilizari reale ale sistemului. Astfel se vor creea conturi pentru angajati si pentru administrator, se va incerca logarea pe baza informatiilor existente in baza de date corespunzatoare lor, se vor emite comenzi ale unor produse si se va verifica daca comanda a fost plasata cu success . In functie de numele clientului se va verifica in baza de date loialitatea sa si i se vor oferi reduceri in anumite conditii. Se va urmari apoi ca angajatii sa primeasca formulare in care sa isi inregistreze activitatile si orele de munca prestate in cadrul lor. La anumite intervale administratorului i se va cere sa intocmeasca rapoarte referitor la activitatile angajatilor sau la detalii legate de clienti

# Future Improvements

Ca dezvoltari viitoare se va crea o aplicatie mobila dupa modelul aplicatiei web in care clientii sa poata naviga mult mai usor printre produse si sa plaseze comenzi. De asemenea in momentul descarcarii aplicatiei acestia primesc reduceri pentru fidelitate.

# 