SEC (semantic clothing)

Cosmin Vladutu - Cursant MLC

Ion Catalin Condrea – Cursant MLC

Cuprins

Introducere 2

Găsirea Soluției 2

Tehnologii și Principii Folosite 3

.NET MVC 3

DotNetRDF 4

Sparql 4

Entity Framework si Code First 6

Bootstrap Framework 6

SQLServer 7

Protege 8

Stardog 9

GitHub 9

Owasp Top10 10

Ontologia 12

Scenariu de Test 13

Idei de îmbunătățire 17

Concluzii 18

Resurse 18

## Introducere

În acest proiect s-a dorit creerea unei platforme folosind tehnologii cât mai recente, prin care orice utilizator să poată să-și poată programa îmbrăcămintea pentru următoarea zi, să-și creeze propriul „dulap” în care să vadă oricând ce haine deține fără și totul să-i fie doar la un click distanța, inclusiv oferte de noi haine și locul de unde le poate cumpăra.

Interfața s-a dorit a fi cât mai simpla și ușor de folosit, platforma acceptând și aducând avantaje și utilizatorilor neînregistrați cât și celor care sunt înregistrați pe diferite rețele de socializare.

S-a reușit creerea unei ontologii prin care hainele pot fi filtrate după producător, culoare, sex, sezon și stil. Dorința noastră a fost ca sa oferim informația dorita sa ajungă la utilizator cât mai repede fără ca acesta să piardă prea mult timp.

## Găsirea soluției

Pentru găsirea soluției optime am încercat sa construim aplicația sprijiniți de etapele dezvoltării unei aplicații software, învățate la Ingineria Programării din anul 2.

1. Specificarea Problemei

2. Analiza Cerințelor

3. Proiectarea Arhitecturală

4. Proiectarea Detaliată

5. Scrierea Codului

6. Integrarea Componentelor

7. Validarea (testare + verificare)

8. Mentenanță

## Tehnologii folosite

Pentru a înțelege mai bine modul în care proiectul a fost creat dar și funcționalitatea acestuia vom încerca sa prezentam fiecare tehnologie (modul) folosit, în mod individual pentru a observa avantajele folosirii lui.

### .Net MVC

Deja ajuns la versiunea 5, a ajuns să fie un framemork din ce în ce mai folosit în vedere ca include mult mai bine principiul de SOC, fată de învechitul WebForms. Acest principiu se refera la împărțirea responsabilităților. Practic de abia aici se face împărțirea clara intre ce vede utilizatorul și ce se întâmpla în spate. În platforma creata de noi am folosit și principiul de DAL și BLL practic am împărțit și mai bine aplicația în vederea logicii interne față de accesul la baza de date.

În aplicație am folosit modele pentru fiecare tabelă din baza de date SQL dar și pentru fiecare tip din ontologie. Partea de view am folosit-o pentru afisare iar partea de controller face legătura intre view și model, practic controleaza acțiunile cerute de utilizator.

### DotNetRDF

Aceasta este o puternica librărie Open-source care are ca țel principal oferirea unui API ușor de utilizat pentru lucrul cu SPARQL, RDF și Web Semantic în general. Momentan oferta suport pentru AllegroGraph, 4store, Apache Jena Fuseki, Sesame, SPARQL Graph Store HTTP Protocol, Stardog și Virtuoso. Noi l-am folosit împreuna cu Stardog pentru creerea platformei.

### Sparql

Este un limbaj de interogare menit sa interogheze și să manipuleze date memorate în format RDF (Resource Description Framework) . Sintaxa SPARQL este foarte asemănătoare cu sintaxa SQL.

RDF este un limbaj care definește grafuri de cunoștințe prin specificarea unor triplete de tipul Subiect Predicat Atribut, unde subiectul și atributul sunt noduri în graful de cunoștințe, iar predicatul reprezinta relația dintre ele. SPARQL se bazează pe crearea de șabloane ce vor fi căutate în interiorul acestui graf de cunoștințe.

Modelul de bază al RDF:

1. *resurse* – Datele descrise RDF sunt denumite *resurse*
2. *proprietăți* – O *proprietate* reprezintă o caracteristică sau o relație pentru a descrie o resursă.
3. *declarații* – O anumită resursă împreună cu o proprietate a sa având asociată o valoare formează o declarație. Declarația este de forma: *{subiect, predicat, obiect}.*

Există o serie de validatoare disponibile astfel încât utilizatorul poate alege. Un exemplu de validator este cel pus la dispoziție de consorțiul W3C și poate fi găsit la adresa: http://www.w3.org/RDF/Validator/

### Entity Framework și Code First

Entity Framawork este una dintre tehnologiile recomandate de Microsoft și este folosită pentru accesarea datelor. Acesta generează obiecte și entități pe tabelelor din baza de date și returnează un obiect sau o colecție de obiecte mai mai ușor de manipulat în .NET. Permite principiul de CRUD ( create, read, update și delete), este ușor de folosit și are funcții predefinite de generează comenzi SQL.

De ce Code First?

Este mult mai ușor să creezi obiectele decât să le manipulezi după tabelele din baza de date. După creere dar și după ce se dorește o modificare o comanda de update face ca maparea cu baza de date sa creeze tabelele și legăturile dintre ele. Principiul de Code First este bine venit aplicațiilor mici și de start-up, în vedere ca nu există o baza de date deja în producție care nu e așa de ușor de modificat.

### Bootstrap Framework

Bootstrap este un framework din ce în ce mai popular pentru developerii front-end. Oferă un bun suport aplicațiilor web dar și celor mobile. Se poate integra într-o aplicație direct din Visual Studio folosind NuGet Manager, și este foarte ușor de folosit.

### SQL Server

Unul dintre cel mai bune servere de SQL, este cel oferit de la Microsoft, varianta folosită de noi fiind Microsoft SQLServer 2008R2. Acesta este un sistem de gestionare a datelor de tip relațional, iar limbajul principal de interogare este T-SQL. Acesta s-a integrat foarte ușor cu celelalte componente de la Microsoft. Crearea tabelelor de useri dar și a celor de legătură folosite pentru memorarea articolelor de îmbrăcăminte ale utilizatorilor a fost făcută direct din code-fist și nu a fost necesar decât conectarea spre serverul bazei de date. Motivația noastră pentru a utiliza acest tip de server de baza de date a fost faptul că MS SQL Server suporta ODBC (Open Database Connectivity), practic ne-am putut conecta la ea remote fără prea multe configurari.

### Protege

Protégé este o platformă open-source ce pune la dispoziție o serie de instrumente pentru a construi modele de cunoștințe bazate pe aplicații cu ontologii. Protégé poate fi particularizat să permită un domeniu suport pentru crearea modelelor de cunoștințe şi pentru introducerea datelor.

O ontologie descrie conceptele și relațiile dintre ele care sunt importante pentru un domeniu particular, furnizând un vocabular pentru domeniul respectiv. În ultimii ani, otologiile au fost adoptate în multe comunități științifice ca o metodă de a împărtăși, reutiliza şi procesa diferite domenii de cunoștințe. Există o mulțime de plug-in-uri pentru a se putea importa ontologii în diferite formate în Protégé incluzând DAG-EDIT, XML, RDF, şi OWL.

Protégé poate fi rulat ca aplicație de sine sau printr-un client Protégé în comunicare cu un server.

Editorul Protégé-OWL permite utilizatorilor:

Încărcare si salvare a OWL (Web Ontology Language) și ontologiilor RDF (Resource Description Framework);

Editarea și vizualizarea claselor și a proprietăților

Definirea caracteristicilor logice ale claselor ca expresii OWL;

Editare individuală OWL pentru web semantic;

### StarDog

Acesta este un sistem de reasoning ce folosește SPARQL ca și limbaj de utilizare. Noi l-am folosit întrucât ne-a oferit un server pe care am putut încărca ontologia și am putut-o interoga cu ușurința în vedere ca integrarea acesteia cu dotNetRdf a fost foarte ușoară. Versiunea 2.1 a fost lansata în ianuarie anul acesta și are ca suport o comunitate destul de bine formata.

### GITHUB

Este un serviciu web de găzduire pentru dezvoltarea de aplicații care are inclus un sistem de control al versiunilor. Github oferă atât conturi gratis pentru proiecte open source cât și plătite pentru proiecte private.

Adresa la care poate fi accesat proiectul nostru este:

https://github.com/cosminvladutu/info2

Aici se poate găsi codul proiectului precum și informații despre versiunile prin care a evoluat platforma și dezvoltatorii care au lucrat la aceasta.

### Owasp Top10

Owasp este o fundație non-profit ce se ocupa cu monitorizarea aplicațiilor web și a atacurilor ce se întâmpla în întreaga lume. Aceasta fundație furnizează cursuri pentru securitatea platformelor web și în fiecare an un nou top 10 al celor mai întâlnite atacuri. Top actual pe care am încercat și noi să-l implementam este acesta:

1. Injecțiile: Pentru a ne proteja de acest tip de atac am folosit entify framework pentru conectarea la baza de date, astfel scăpând de orice fel de injectie la nivel de baza de date.

2. Autentificare Stricată sau Sistem greșit de management: Nu avem secțiune de admin deci orice utilizator poate vizualiza date. În vederea utilizatorilor înregistrați, pentru a le oferi toată securitatea păstram parolele în hash, criptate, iar pentru utilizatorii de pe rețelele de socializare le memoram un id unic împreuna cu numele lor de utilizator.

3. XSS: Toate datele care pot schimba starea serverului sunt validate.

4. Referințe la obiecte nesigure: Ne-am interesat în legătura cu versiunile framework-urilor folosite și nici unul nu are o vulnerabilitate, și alte referințe externe nu folosim.

5. Configurări greșite de securitate. Toate versiunile de framework-uri sunt la ultima variantă iar datele de conecsiune sunt criptate folosit IIS.

6. Date confidențiale nesecurizate: Parolele le ținem criptat în baza de date.

7. Grad diferit de acces: Fiind o aplicație open-source nu am implementat grade diferite de acces. „Oricine poate spune orice despre orice”.

8. CSRF: Pentru fiecare acțiune ce poate modifica serverul am folosit o cheie unica de autentificare, auto-generata la fiecare refresh al paginii. În cazul în care cheia de pe server nu coincide cu cheia utilizatorului, atunci acțiunea va fi stopata.

9. Folosirea de componente cu vulnerabilități: Nu este cazul. Componentele sunt la ultima varianta, fără a avea branșe de securitate.

10. Redirecționări fără a fi validate: toate redirecționările sunt validate pentru a fi doar interne. Toate redirecționările externe se fac într-o fereastra noua.

### Conceptul de ontologie

Ontologia este definită drept “teoria particulară despre natura existenței” (Woolf, 1981). Obiectivul unei ontologii constă în reprezentarea unei conceptualizări, partajabile și reutilizabile, în care sunt ignorate detaliile specifice aplicațiilor. Sfera de cuprindere a unei ontologii se referă la toate aplicațiile domeniului, nu numai la o singura aplicație.

**Ontologia** folosită de noi a fost construită după în felul următor: În primul rând avem clasele Product și Type.

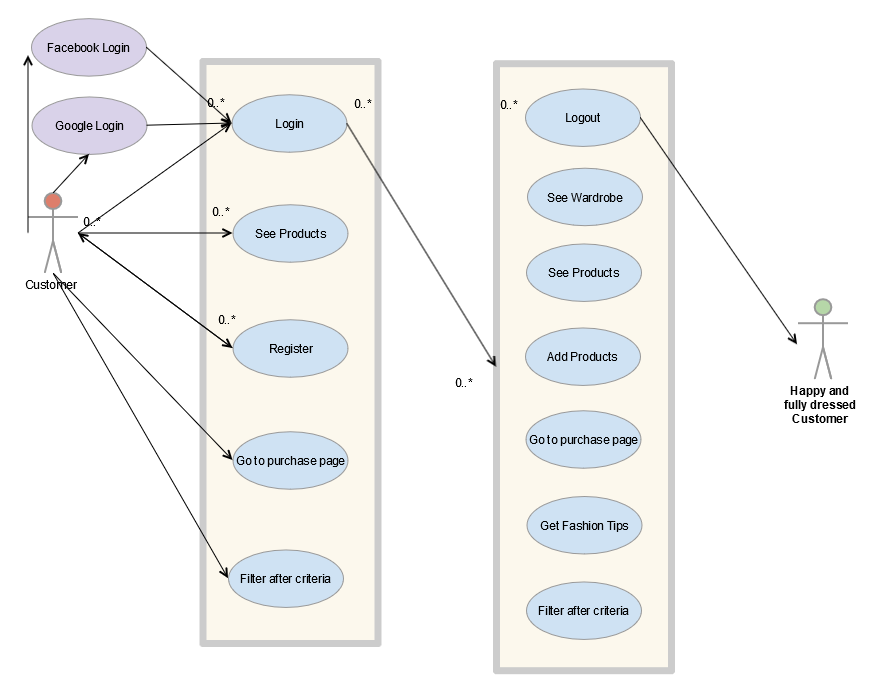
Clasa Product conține subclase cu diferite tipuri de produse ca și caracteristici iar clasa Type conține diferite subclase, fiecare dintre ele concentrându-se asupra unor atribute generice pe care le pot avea indivizii (instanțele de clase). Predicatele din ontologie presupun o serie de legături dintre indivizi și caracteristicile elementelor la care se raportează (sezon, still, culoare, brand).

## Use case

Am pornit de la ideea că aplicația noastră să fie cât mai intuitivă și ușor de folosit pentru utilizatori, iar timpul necesar petrecut utilizând aplicația pentru aflarea informațiilor necesare să fie cât mai scurt posibil.

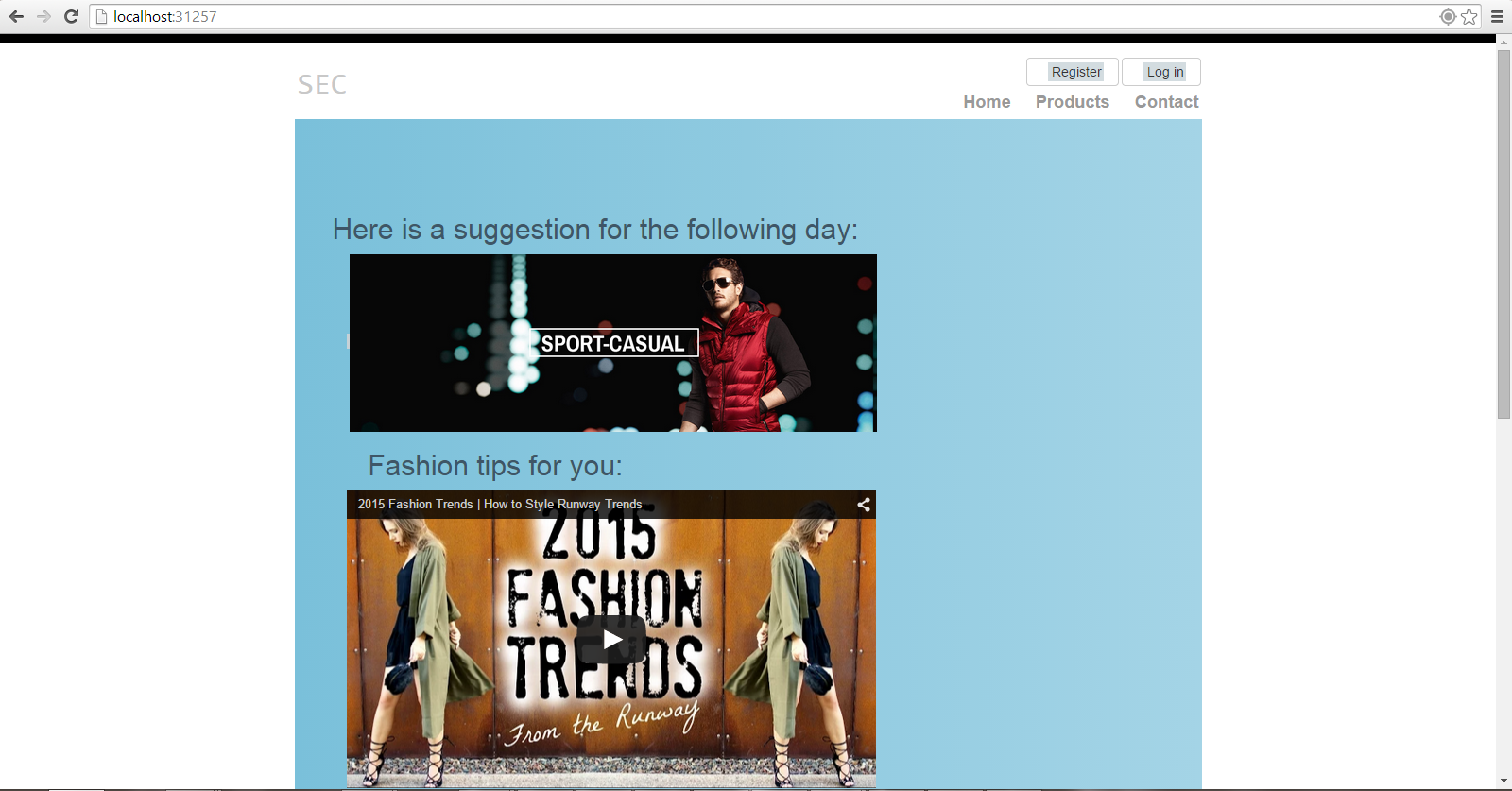
Utilizarea aplicației este una intuitivă deoarece nu au fost creeate meniuri care să îngreuneze navigarea și care să inducă utilizatorul în dileme de utilizare.

După abordarea pe care am avut-o asupra aplicației utilizatorii pot avea următoarele capabilități:

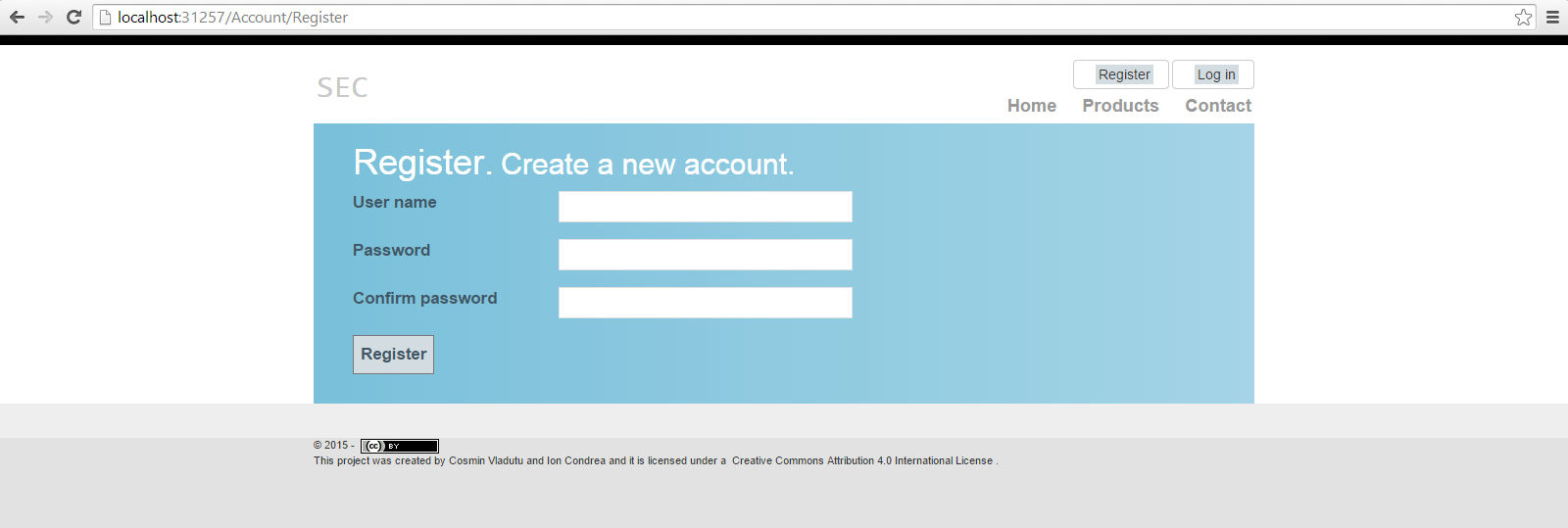


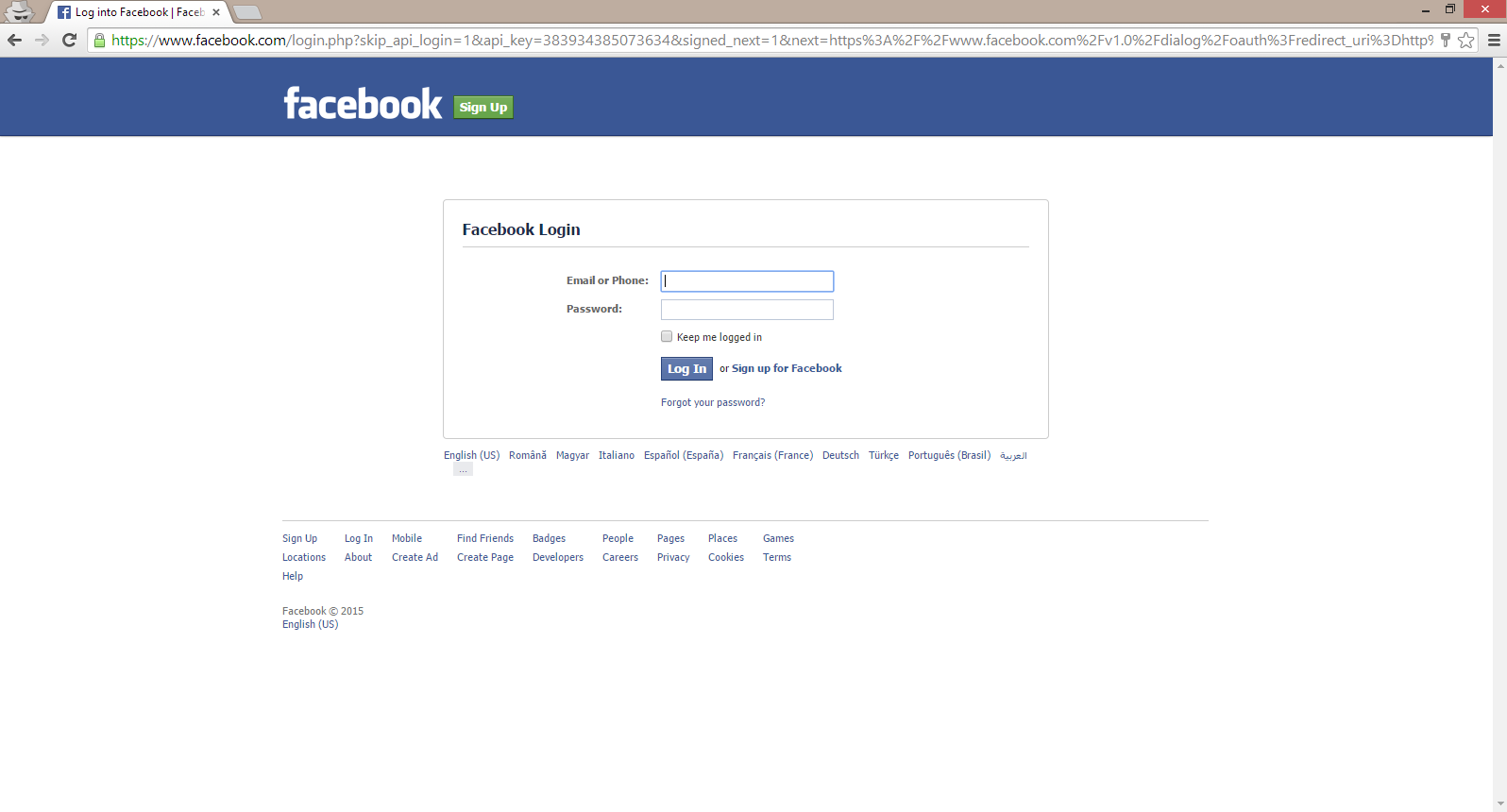
După cum puteți vedea, utilizatorul chiar și neînregistrat are acces pentru a vizualiza produsele din ontologie pe care le poate filtra după dorința sa și poate să viziteze site-ul de pe care poate cumpăra produsele care ii atrag atenția.

Prima pagina a aplicației devine clipuri video în legătura cu moda și temperatura în zona în care utilizatorul se afla.

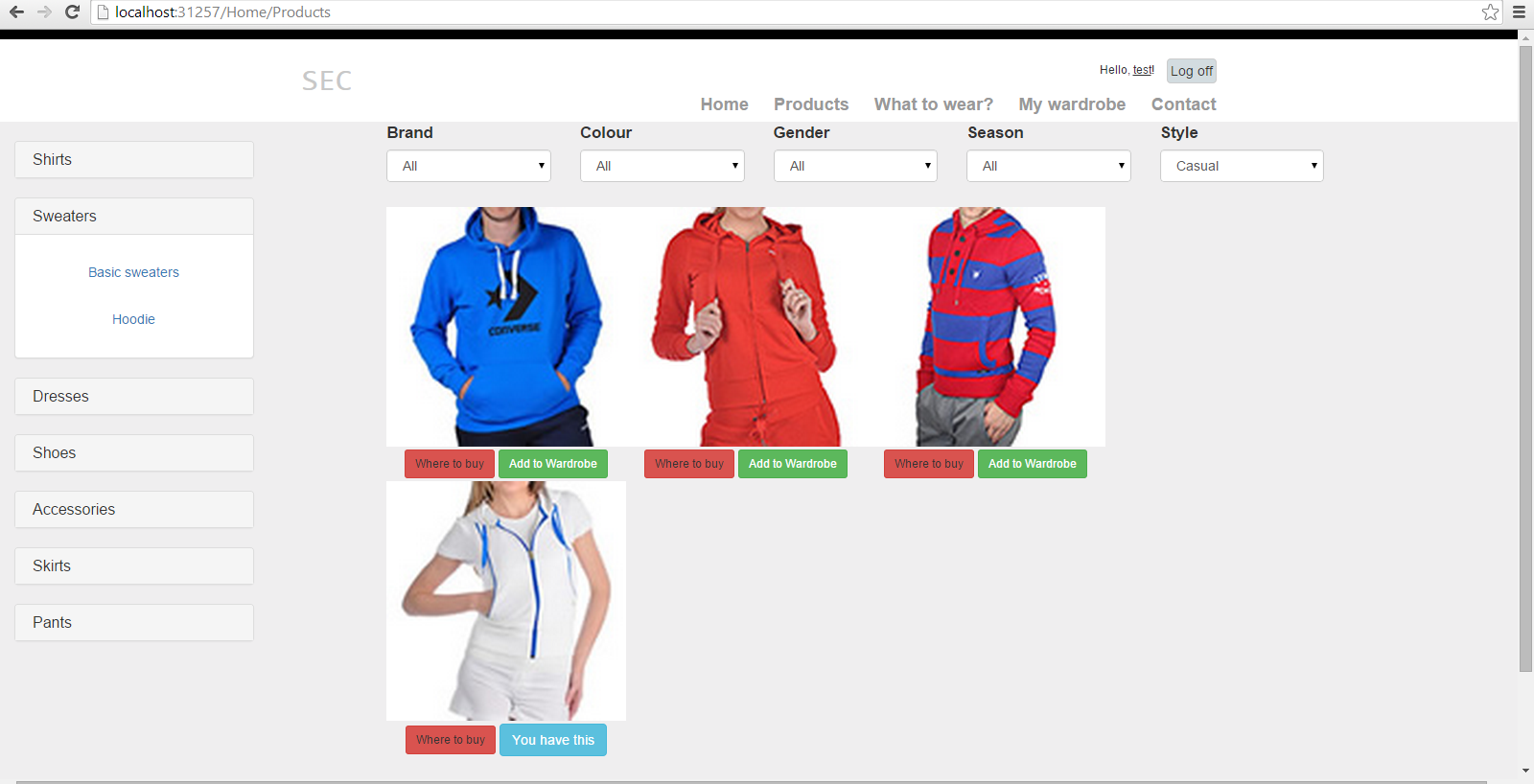


Înregistrarea utilizatorului se poate face direct prin intermediul site-ului completând un formular de înregistrare

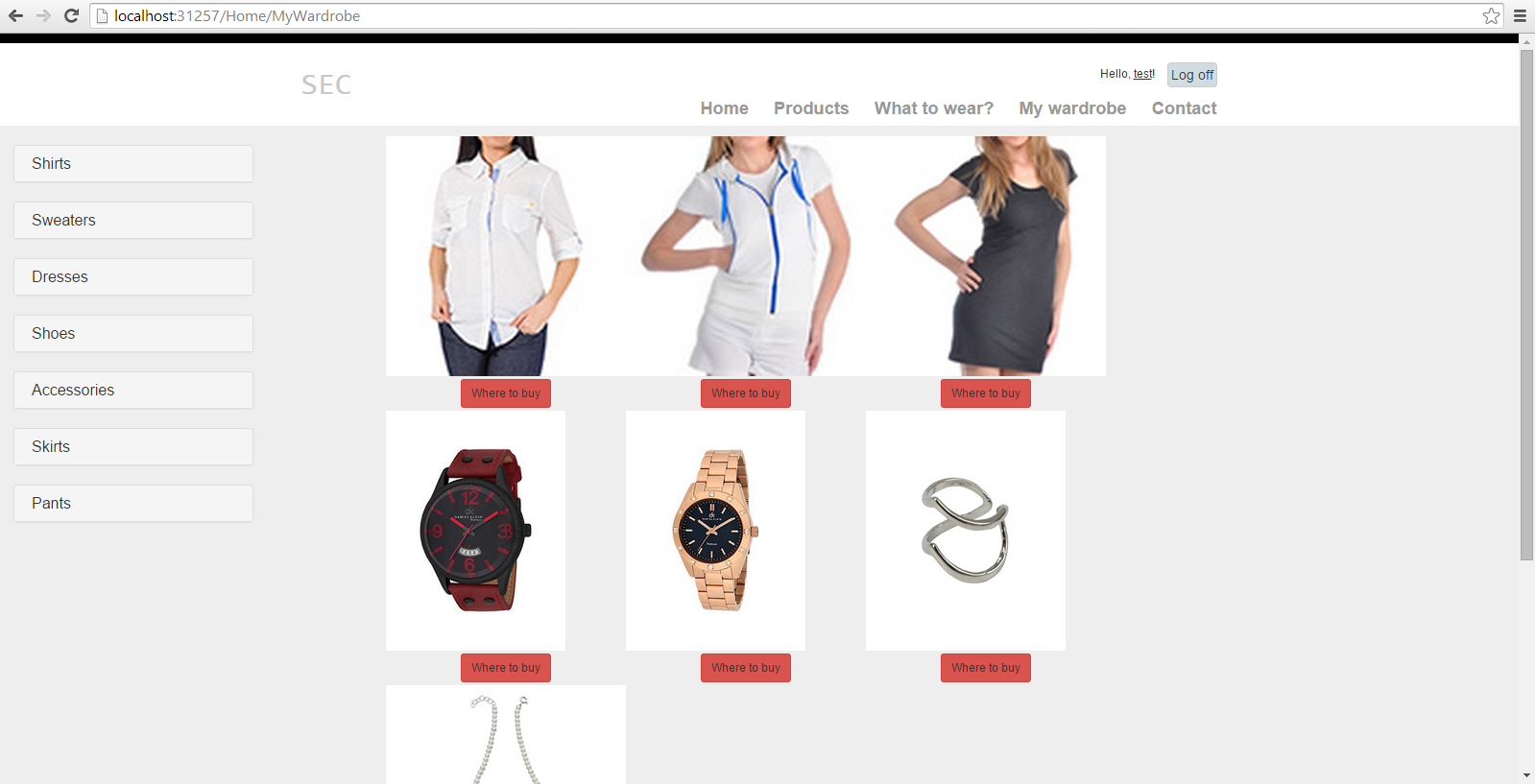
dar și utilizând un cont pe care deja îl deține într-o rețea de socializare.

După autentificarea via facebook (în acest caz) aplicația primește un token de validare criptat prin care își da seama că, credențialele utilizatorului sunt valide și se poate inregistra numai cu userul retelei de socializare.

După înregistrare utilizatorul poate vedea toate produsele din ontologie, le poate adauga în garderoba sa în cazul în care le deține deja sau poate vizita magazinul virtual de pe care sa cumpere produsul (evident poate folosi filtrări per produs, marca, culoare, sex, sezon și stil pentru a găsi produsul dorit mult mai ușor).



După adăugarea produselor pe care le deține, utilizatorul poate sa le vizualizeze având acces la aceleași filtre având avantajul de a avea legătura directa spre magazinul online ce deține acea piesă de îmbrăcăminte în cazul în care vrea să vadă evoluția prețului.



Scopul principal al aplicației a fost creerea utilizatorilor înregistrați de a avea un mod prin care sa aleagă garderoba mult mai ușor. Acest țel a fost îndeplinit folosind opțiunea „What to wear?” ce oferă idei de îmbrăcăminte bazându-se pe produsele din garderoba și deabia apoi pe cele din ontologie dacă utilizatorului ii lipsește un anumit tip de îmbrăcăminte.

## 

## Idei de îmbunătățire

- În primul rând folosind o bază de cunoaștere este clar ca cel mai important lucru este acela ca baza noastră să cuprindă cât mai multe informații, deci adăugarea de noi produse ar fi primul lucru care ar trebui făcut în aplicație.

- O alta idee ar fi creerea unui editor prin care utilizatorul normal să poată introduce noi informații în ontologie pe baza unui reguli clare de validare.

- Un sistem de notare a produselor, pentru ca alți utilizatori sa poată cauta produse după un anumit număr de steluțe.

- Un sistem de memorare al preferințelor utilizatorului logat în legătura cu culoarea, producătorii preferați , etc. sistem bazat pe căutările utilizatorului nu pe un formular tipizat care se poate schimba fără ca utilizatorului să-și mai reînnoiască preferintelor completate la logare.

- Integrarea mai activa cu rețelele de socializare în ideea în care utilizatorul sa poată scrie comentarii și să sa împartă linkuri cu piese de îmbrăcăminte de pe site direct din aplicație.

## Concluzii

Aplicația în versiunea actuală acoperă puțin din nevoile unui utilizator real, însă poate fi considerată a fi o bază nu doar pentru Semantic Cloathing ci și pentru componente ale Web-ului Semantic.

Menționăm că proiectul și-a ales scopul inițial, ajutarea utilizatorului în a lua decizii în legătura cu îmbrăcămintea mult mai ușor și rapid, totodată fiind un foarte bun exercițiu în lucrul cu Web-ul Semantic pentru noi.

## Bibliografie și Resurse

* http://www.w3.org/2001/sw/
* http://semanticweb.org
* http://docs.stardog.com/
* sit: http://www.w3.org/2001/sw/
* <https://www.owasp.org/index.php/Top_10_2013-Top_10>
* <https://www.gliffy.com/>
* <http://www.slideshare.net/>
* <http://www.techsmith.com/tutorial-camtasia-8.html>
* [www.w3schools.com](http://www.w3schools.com/)
* <http://stackoverflow.com/>