



UNIVERSITÀ DEGLI STUDI DI TRENTO

Dipartimento di Ingegneria
e Scienza dell'Informazione

Corso di Introduzione alla programmazione web

PROGETTO FINALE

"ShoppingLesto"

ShoppingLesto

Anno accademico 2017-2018

Battisti Eliana, Castellan Samuele, Dietre Alberto, Fabris Pietro, Gerevini Alessandro, Gridea Robert

INDICE

Introduzione	3
Features	3
Utilizzo del sistema	3
Sistema suggerimenti	3
Sistema messaggistica	3
Sistema gestione permessi	3
Configurazione Iniziale	4
Utilizzo in modalità “utente registrato”	4
Utilizzo in modalità “utente anonimo”	6
Interfaccia admin	6
Considerazioni sulle scelte effettuate	7
Front End	7
Back End	8
Database	8
MVC	8
Suggerimenti	9
Geolocalizzazione	9
Chat	10
Ricerca dei prodotti	10
Modello E-R	11

Introduzione

ShoppingLesto è una web app pensata per aiutare l'utente a creare e condividere liste della spesa. È facilmente utilizzabile sia su dispositivi fissi che su dispositivi mobili grazie al suo design responsive.

Features

Utilizzo del sistema

Il sistema permette di lavorare sia come utente registrato che come utente anonimo. L'utente anonimo avrà la possibilità di creare una sola lista, senza la possibilità di condividerla e di usare il servizio di chat con gli altri utenti.

Se si decide invece di accedere come utente registrato si avranno tutte le funzionalità disponibili (creazione illimitata di liste, condivisione di liste, accesso al sistema di suggerimenti delle scadenze e possibilità di chattare con gli altri utenti)

Sistema suggerimenti

Il nostro sistema presenta una funzionalità di **suggerimenti su prodotti in esaurimento**: il sistema determina in autonomia quando avvisare l'utente che un certo prodotto che la persona acquista abitualmente è in esaurimento, indicandogli in quale lista il prodotto deve essere inserito. In più è presente una pagina da cui l'utente (sia anonimo che registrato) può **controllare i punti vendita vicini** alla propria posizione, inerenti alle categorie delle sue liste.

Sistema messaggistica

Per dare la possibilità agli utenti di comunicare all'interno del sistema senza affidarsi a servizi esterni, **abbiamo creato una chat**. Questa è presente in ogni lista creata da un utente registrato. Permette di comunicare con gli altri utenti con cui è stata condivisa. Il funzionamento della chat viene spiegato al capitolo 5.

Sistema gestione permessi

Quando un utente registrato crea una lista, in fase di condivisione può decidere di assegnare dei permessi agli utenti che verranno collegati. I permessi sono di tre tipi:

- **aggiunta:** l'utente può aggiungere prodotti alla lista.
- **modifica:** l'utente può modificare i dettagli della lista come il nome e la descrizione, ma anche la categoria e la foto.
- **condivisione:** l'utente può a sua volta condividere la lista con altri utenti e scegliere quali permessi dare agli altri utenti.

Configurazione Iniziale

Utilizzo in modalità *"utente registrato"*

1. Premere il bottone *"register"* sulla schermata che si presenta come landing page
2. Inserire i dati richiesti, se questi rispettano le condizioni richieste (sicurezza della password e dimensione dei campi), verrà inviata una mail con un link per confermare la propria registrazione
3. Confermare la registrazione tramite il link ricevuto via mail
4. Confermata la registrazione, si verrà reindirizzati sulla propria home page
5. Da qui si possono gestire le liste, basterà cliccare sul bottone **"AGGIUNGI NUOVA LISTA"**
6. Verrà chiesto di inserire i dati della lista e di confermare con il pulsante **"NUOVA LISTA"**
7. A questo punto nella propria home page sarà presente la nuova lista con l'immagine (se inserita, altrimenti sarà messa quella di default), il nome, la descrizione e l'utente che l'ha creata.

Liste

AGGIUNGI NUOVA LISTA



Compleanno

Festa di compleanno Pietro

Samuele Castellan

8. Cliccando sulla lista si accederà alla pagina del dettaglio, dove sarà possibile visionare, modificare, eliminare e condividere la lista.



Lista: Compleanno

Descrizione: Festa di compleanno Pietro

Proprietario: Samuele Castellan

Categoria: Chips e Patatine

nella parte centrale della pagina sarà possibile visionare, cercare e aggiungere prodotti alla lista

Visualizza elementi

Cerca:

Nome	Descrizione	Immagini	Prezzo	Rimuovi
------	-------------	----------	--------	---------

Nessun dato presente nella tabella

Vista da 0 a 0 di 0 elementi



Precedente

Successivo

AGGIUNGI NUOVO PRODOTTO

mentre nella parte inferiore della pagina è presente una chat per comunicare con gli altri utenti con cui è stata condivisa la lista.

Chat



9. tornando alla propria home page, nel menù in alto a destra, sono presenti i bottoni: “LE TUE LISTE”, “TROVA NEGOZI”, la campanella delle notifiche e il bottone con il proprio nome.
 - a. Il bottone LE TUE LISTE, porta alla propria home page dove sono presenti tutte le liste create e condivise.
 - b. il bottone TROVA NEGOZI porta alla pagina dei suggerimenti basati sulla propria posizione (vedi prossimo punto)
 - c. la campanella delle notifiche se premuta visualizza le notifiche presenti
 - d. il bottone con il proprio nome, apre un menù a tendina dove è possibile fare il logout e, cliccando su “IL TUO ACCOUNT” permette di modificare i propri dati come il nome e la password e di eliminare l’account
10. Premendo il bottone “TROVA NEGOZI” si aprirà una pagina dove é presente una mappa dinamica in base alla posizione dell’utente. Di default la mappa mostra i supermercati in zona. È possibile, anche, scegliere la lista per la quale si vogliono i suggerimenti semplicemente cliccando su di essa, così da visualizzare i punti vendita relativi alla categoria della lista selezionata.

Utilizzo in modalità “*utente anonimo*”

1. Per usare ShoppingLesto in modalità utente anonimo (senza registrazione) basterà, dalla landing page, cliccare il bottone “ANONIMO”
2. L’utente così creato avrà solamente la possibilità di creare una lista, ma non avrà la possibilità di condividerla e non potrà utilizzare il servizio di suggerimenti della scadenza dei prodotti.

Il funzionamento é basato sull'utilizzo di un cookie di sessione che permette di tenere traccia quale lista ha creato l'utente. Ovviamente quella lista é disponibile solo su quel browser di quel dispositivo.

Interfaccia admin

Per questioni relative alla moderazione dei contenuti, è stato deciso di non permettere ai singoli utenti di poter creare nuove categorie di prodotti e nuove categorie di liste, pertanto questi permessi sono forniti solo all’amministratore del sistema.

Accedendo con la propria mail e password, l'admin ha a disposizione tre pagine per la gestione del sito.

Potrà aggiungere, modificare, rimuovere prodotti, categorie di prodotto e categorie di lista, semplicemente usando il menù in alto a destra per selezionare cosa modificare e dalla pagina che si aprirà con il bottone "+" aggiungere, con il bottone con l'icona della penna, modificare e con il bottone con l'icona del cestino, eliminare il prodotto, categoria di lista o categoria di prodotto.

Considerazioni sulle scelte effettuate

Front End

Il progetto è basato sul framework **Bootstrap 4**. La scelta è stata effettuata considerando diversi fattori, tra i quali:

- Possibilità di creare interfacce responsive
- Comodità del layout a griglia
- Qualità della documentazione presente in rete e della community di supporto
- Possibilità di utilizzare elementi predefiniti ma lasciando anche al designer un elevato grado di personalizzazione
- Semplicità di utilizzo
- Facile implementazione con altri framework e piattaforme

Nonostante Bootstrap abbia permesso di risparmiare tempo utilizzando componenti "ready-to-use", diverse parti dell'interfaccia grafica sono state create da zero in HTML/CSS per soddisfare pienamente i requisiti.

Per l'interfaccia grafica destinata alla gestione dei prodotti e delle loro categorie (pannello di controllo dell'Admin) si è optato per l'utilizzo del framework **Datatables**.

La scelta è stata effettuata considerando i vantaggi che porta con sé il framework:

- non solo la possibilità di visualizzare i dati delle tabelle ma anche di filtrare e cercare contenuti
- semplice integrazione con Bootstrap e la modalità responsive.

Back End

Il progetto è stato sviluppato con la tecnologia **Servlet/JSP/JSTL** e opera su un web server (**Tomcat**). Queste tecnologie permettono la generazione di pagine web dinamiche a seconda dei parametri di richiesta inviati dal browser dell'utente.

Database

Per la realizzazione del layer di persistenza, l'RDBMS che abbiamo usato é **MySQL**. Questa scelta è stata fatta basandosi sulla nostra familiarità d'uso con tale sistema.

La gestione del layer di persistenza avviene applicando il pattern **DAO**, il quale è un pattern architetturale che permette di stratificare ed isolare l'accesso ad una tabella dell'RDBMS tramite classi Java e query.

Il vantaggio nell'uso del pattern DAO è che quest'ultimo permette di mantenere una rigida separazione tra le componenti dell'applicazione e facilita la sua manutenibilità.

Nella libreria dei DAO, per istanziare gli oggetti, viene utilizzato il Factory Method, uno dei design pattern fondamentali e che indirizza il problema della creazione degli oggetti senza specificare l'esatta classe. Questo pattern viene implementato tramite delle classi Java che vengono inizializzate all'avvio dell'applicazione tramite un Event Listener che le attiva le rende disponibili come attributo di applicazione, così da poterle richiamare da tutti i componenti dell'applicazione.

La connessione al DBMS viene effettuata tramite il **driver JDBC**, il quale permette l'accesso e la gestione della persistenza dei dati nei database indipendentemente dal tipo di DBMS usato.

MVC

L'applicazione è stata sviluppata adottando il pattern architetturale **MVC** (Model-View-Controller) utilizzando un approccio Servlet-centric: le pagine JSP sono usate solo per la parte di visualizzazione (View), mentre le servlet forniscono la parte di controllo dell'applicazione (Controller).

Le Servlet gestiscono le richieste che arrivano dagli utenti tramite l'URL, a cui rispondono, elaborando i dati necessari a soddisfarle, interagendo con i DAO per prelevare o salvare i dati nel database e utilizzando dei POJO (Model) per rappresentare le informazioni. Infine le informazioni vengono trasferite alle pagine JSP la quali si occupano di presentare visivamente i risultati.

Il sistema è stato pensato per essere usato da tre diverse categorie di utenti: "anonimi", "registrati" e "amministratori".

Per questo motivo è stato adottato un processo di autorizzazione basato su ruoli tramite il quale ciascuna categoria di utente può svolgere diverse operazioni e accedere solo alle risorse per le quali ha i permessi. Per fare questo si fa uso di specifici **filtri** che controllano se effettivamente l'utente, che fa la richiesta di un dato, abbia i permessi per potervi accedere.

Gli utenti amministratori, dopo aver effettuato il login, vengono indirizzati all'apposito pannello di controllo, dal quale possono visualizzare, aggiungere, modificare ed eliminare i dati legati ai prodotti, alle categorie di prodotto e alle categorie di lista.

Per implementare la gestione delle notifiche via mail, abbiamo scelto di usare **Apache Velocity**: un template engine Java-based che permette di usare un linguaggio basato su template per referenziare oggetti definiti in Java. Questo permette di creare template dinamici in HTML che al momento dell'invio della mail vengono compilati nei loro campi con gli oggetti Java collegati all'utente o alla situazione corrente.

Suggerimenti

Per la realizzazione dei suggerimenti dei prodotti in esaurimento si è deciso di basarsi sulla **frequenza di acquisto del prodotto** (ovvero quando viene rimosso dalla lista si presuppone che sia stato comprato in quel momento). Il sistema quindi calcolerà in base alla frequenza con cui vengono acquistati i vari prodotti, quando riproporli all'utente con una notifica, indicando che il prodotto potrebbe essere in esaurimento.

L'algoritmo che è stato ideato, segue le seguenti fasi:

1. quando un prodotto viene acquistato si salva nel database l'avvenuta rimozione dalla lista e si calcola quanto tempo è passato dall'ultima volta che è stato acquistato.
2. Dopo un certo numero iniziale di volte che viene acquistato il prodotto, si calcola una media di tempo che intercorre tra un acquisto e l'altro.
3. passato un tempo leggermente inferiore al valor medio, il sistema invierà una notifica all'utente ricordandogli che tale prodotto potrebbe essere in esaurimento.

Geolocalizzazione

Per la parte della geolocalizzazione abbiamo usato una funzionalità offerta da HTML5. La funzione [**navigator.geolocation.getCurrentposition**](#). Questa feature permette di avere la locazione dell'utente direttamente tramite l'uso di javascript. A

questa funzione vengono passati callback di successo, errori ed opzioni. Noi abbiamo associato alla callback di successo l'aggiornamento della mappa in base alla categoria della lista selezionata. Di default la pagina si apre cercando i supermercati nella zona. Poi sarà l'utente, che in base alle liste da lui create, può scegliere che punti vendita visualizzare sulla mappa.

Non é presente un sistema di notifiche per la localizzazione dei punti vendita ma in questo modo l'utente ha una pagina apposita per controllare, quando vuole, dove andare a fare la spesa.

Abbiamo deciso di non usare le API Google, per via del fatto che in modalità free il loro utilizzo è molto limitato. Qualora il nostro sistema venisse commercializzato per un utilizzo massivo, e si volessero usare le API Google consigliamo di modificare la parte della gestione della Geolocalizzazione ed utilizzare la versione premium di queste, per non avere problemi sul numero di richieste effettuabili.

Chat

Per quanto riguarda invece la feature della **messaggistica** via chat è stata creata una componente all'interno della pagina dove viene data all'utente la possibilità di scambiare messaggi con gli altri utenti con cui è stata condivisa la lista. Un utente ha la possibilità sia di mandare dei messaggi preimpostati (come da specifiche), sia di mandare dei messaggi personalizzati. I messaggi vengono trattati come oggetti Java e vi é un DAO apposito per la gestione degli stessi.

Quando un messaggio viene inserito nella chat, parte una chiamata AJAX ad una servlet apposita per la registrazione del messaggio del messaggio all'interno del database. Mentre per la ricezione dei messaggi è presente una chiamata AJAX all'apposita servlet per la ricerca ed il controllo di nuovi messaggi non ancora visualizzati all'interno della chat stessa.

Ricerca dei prodotti

Per la ricerca dei prodotti da aggiungere ad una lista all'utente vengono presentate due possibilità di ricerca tra prodotti esistenti ed una modalità di creazione di un prodotto custom.

La ricerca é presente nelle seguenti modalità:

- ricerca di un prodotto tramite il nome: vi è una barra di ricerca dove l'utente può inserire il nome del prodotto che vuole aggiungere e partono delle chiamate in ajax ogni volta che il campo di testo cambia. Così da popolare la tabella con solo i prodotti interessati.

- è presente anche il tasto che permette di mostrare tutti i prodotti presenti nel database e successivamente cercare l'item da aggiungere navigando nella tabella.

La creazione di un prodotto custom avviene attraverso un modal con un form che permette di inserire tutti i dati del prodotto.

Modello E-R

Di seguito il diagramma ER del nostro database

