



l'innovativo gestore di liste della spesa

## MEMBRI del GRUPPO

- Tommaso Bosetti **185286**
- Sebastiano Chiari **185858**
- Leonardo Remondini **184563**
- Marta Toniolli **187839**

È importante fare riferimento al branch master per avere delle versioni funzionanti del programma, in quanto negli altri branch non garantiamo che tutte le funzionalità siano implementate e/o funzionanti, perché avendo la possibilità di lavorare sul progetto fino alla sua presentazione, siamo in continua fase di programmazione.

## SCELTE IMPLEMENTATIVE

Durante la realizzazione del progetto Friday, ci siamo trovati davanti a diversi bivi in quanto a scelte implementative per realizzare le varie funzionalità del sito. In alcuni casi le scelte che abbiamo preso si sono rivelate efficaci (almeno per quanto ci riguarda e per la nostra visione d'insieme del progetto), in altri meno.

In particolare, nelle fasi iniziali del progetto, abbiamo scelto di non utilizzare i JavaBean in quanto, nonostante garantiscano un maggiore livello di formalità nella scrittura del codice e in generale nella gestione degli oggetti, a nostro parere ne limitavano la flessibilità all'interno del codice.

Allo stesso modo, per quanto riguarda filtri e listeners, abbiamo inizialmente scelto di non servirci di queste tecnologie, gestendo gli aggiornamenti "in live" delle informazioni e gli eventuali controlli correlati all'interno della stessa pagina tramite web socket, servlet e parametri di sessione. Tuttavia, ci stiamo rendendo conto che per effettuare controlli molto più precisi e sicuri sugli indirizzi navigabili all'interno del sito filtri e listeners sono una componente necessaria da implementare, e quindi stiamo cercando una soluzione alternativa al loro utilizzo che sicuramente risulterà meno elegante, ma che almeno ci permetterà di non stravolgere completamente la struttura del progetto. Siamo comunque consapevoli che alcune pagine siano raggiungibili scrivendo direttamente l'URL se si conosce lo schema del progetto e la suddivisione dei vari file.

Per quanto riguarda l'implementazione effettiva delle funzioni richieste, ad oggi manca ancora la geolocalizzazione. Attualmente, questa è parzialmente implementata e non funzionante. Dato che creava molti problemi al progetto, abbiamo deciso di pubblicarlo senza questa parte e di cercare di implementarla in seguito entro la presentazione. Abbiamo creato un branch apposito per questa funzionalità (geolocalizzazione), visto che lavorando in locale la geolocalizzazione impediva l'esecuzione del progetto e quindi la possibilità di implementare altre funzionalità importanti.

Una cosa infine che non è ancora supportata è l'effettivo caricamento delle immagini. Infatti nella nostra applicazione web è inserita la possibilità di inserire un avatar per un utente, o l'immagine di un prodotto o di una lista, ma in realtà nel database viene solo salvato il nome dell'immagine caricata. Non viene effettivamente caricata e salvata successivamente in una precisa località o salvata all'interno del database (tramite per esempio un tipo BLOB). Di conseguenza, se l'immagine scelta è una di quelle di default che abbiamo già inserito noi all'interno delle cartelle del progetto, allora verrà poi visualizzata correttamente durante la navigazione del sito, altrimenti, non verrà visualizzata.

Le altre funzionalità richieste dalla descrizione del progetto invece sono state tutte implementate. È da segnalare il fatto che alcune non sono ancora perfette e necessitano una limatura finale, o per via di bug che ancora non riusciamo a gestire o per via di soluzioni programmatiche non propriamente eleganti.

Un esempio può essere la modalità attraverso la quale si può diventare Admin, e di conseguenza poter creare nuovi prodotti, nuove categorie di prodotti e nuove categorie di lista per tutti gli utenti. Si tratta di un sistema non adatto ad un progetto del genere (nell'ottica della distribuzione), ma funzionale per dimostrare le corrette funzionalità e differenze date dal ruolo di Admin.

Nella struttura del codice abbiamo cercato di rispettare il pattern MVC utilizzando sempre i DAO per connetterci al database e rispettando la separazione tra business logic e la parte di presentazione. Siamo abbastanza convinti di esserci riusciti, ma siamo anche consapevoli che molto probabilmente certe soluzioni adottate non sono esattamente quelle che ci si potrebbe aspettare. Forse, da questo punto di vista, la scelta di non implementare la parte relativa ai JavaBean ci ha messo in difficoltà nelle fasi più avanzate del progetto.

In generale siamo a conoscenza di diversi bug e consapevoli di funzionalità mancanti che andrebbero implementate e come già detto sopra, se si trattasse della realizzazione di un progetto reale, allora questa versione sarebbe solo una prima release e nelle successive implementeremmo tutte le funzionalità mancanti e perfezioneremmo quelle già esistenti cercando di non rendere più complesso un programma che lo è già molto.

Per quanto riguarda la scelta del database, abbiamo deciso di utilizzare un database SQL, sfruttando MySQLWorkbench, puramente per il fatto che sono state le tecnologie insegnateci nel corso di Basi di Dati.

Per quanto riguarda le possibilità all'interno della nostra applicazione nel caso di un utente anonimo, abbiamo deciso che l'utente anonimo è impossibilitato alla creazione di nuovi prodotti. Questi, come da specifiche, possono essere creati dagli utenti registrati e disponibili per tutti, nel caso di utente admin, oppure, nel caso di utenti regolari, disponibili solamente per gli utenti con cui è stata condivisa una lista con il nuovo prodotto creato al suo interno. L'utente che condivide una lista, attraverso il form di condivisione, sceglie i vari permessi che ciascun altro utente può avere. Questi vengono elargiti tramite checkbox nel form di condivisione e prevedono la possibilità di modificare le impostazioni della lista, aggiungere/rimuovere prodotti dalla lista e eliminare la lista.

Per quanto riguarda la gestione della sessione e tempistiche relative, se un utente seleziona l'opzione "ricordami" in fase di login verrà ricordato per un giorno. Altrimenti, la sua sessione ha una durata pari a mezz'ora, dopodiché dovrà rieseguire il login.

## UTILIZZO

In primo luogo, è necessario creare il database e inizializzarlo in modo tale che sia operativo per l'applicativo web. Abbiamo fornito un file `database.sql` che all'interno contiene sia il create schema per creare il database che i vari import che vanno a popolarlo secondo le specifiche richieste. È importante, ai fini della connessione, che il database rispetti questi parametri: come nome utente "root", come password "root81097" e come porta di riferimento "3306".

Una cosa importante da segnalare è che attualmente nel database esiste un utente **admin** proprietario di tutti i prodotti e di tutte le categorie di lista che però non è accessibile nel nostro sito web, in quanto la sua password non è criptata e quindi il relativo login fallirà ogni volta. Abbiamo tentato di risolvere questo dettaglio prima della consegna ma per qualche motivo a noi ignoto la modifica di questa entry del database creava grossi problemi al funzionamento dell'applicazione web. Pertanto abbiamo deciso di consegnare una prima versione con ancora questo utente nel database, ma contiamo per la presentazione di eliminare questo problema.

Come richiesto nel database, sono inseriti 10 utenti tutti correttamente registrati. Le loro credenziali di accesso si trovano nel file "credenziali utenti registrati.txt" e sono un insieme abbastanza eterogeneo di amministratori, utenti senza liste, utenti con liste di loro proprietà e/o liste condivise con altri utenti e utenti con solo liste condivise.

Da segnalare inoltre che nel database esiste un utente `aaa@cercasi.cercasi` che non è inserito nel file .txt di cui sopra perché in realtà è utente di test che utilizziamo per fare vari test e che abbiamo lasciato nel database.