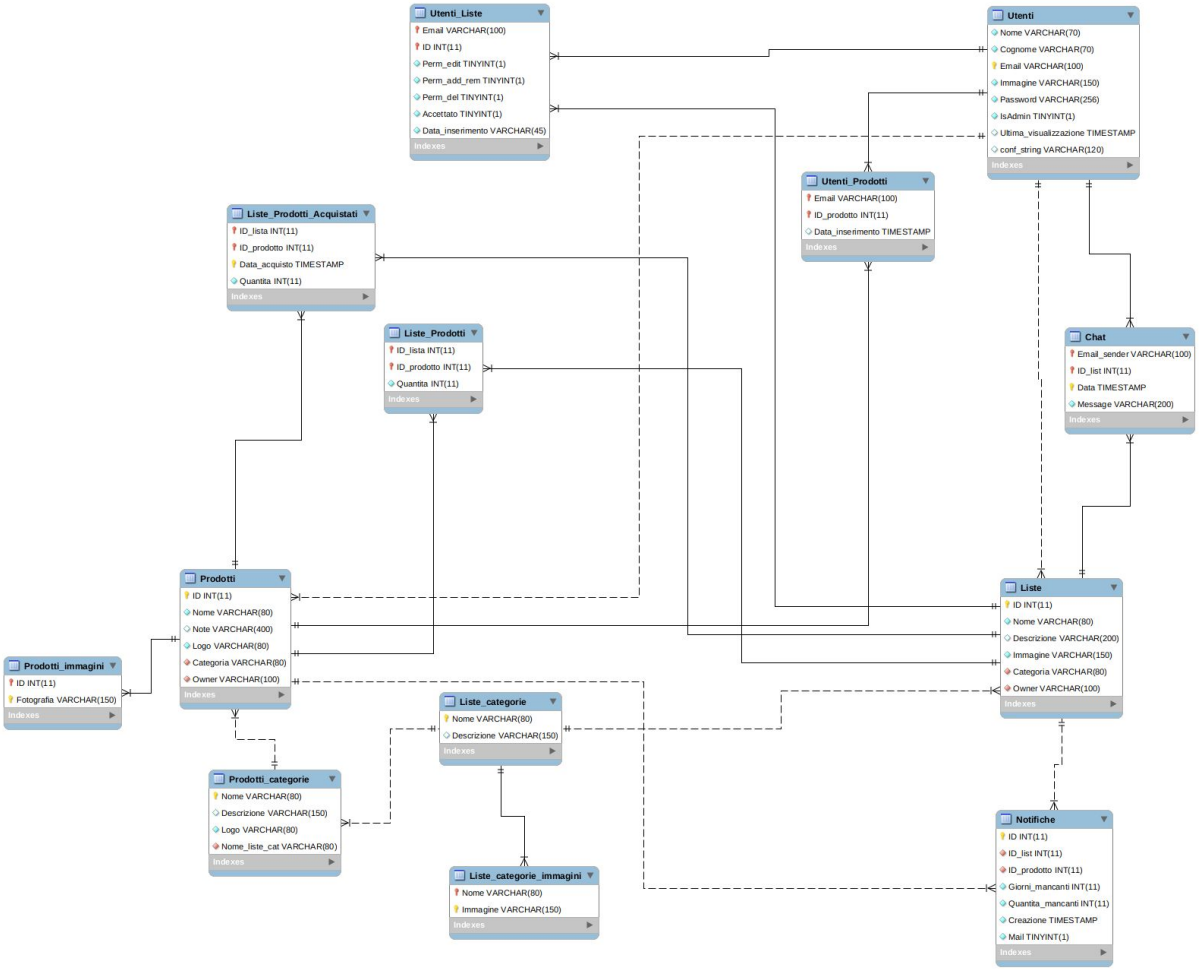
Shopping List - La banda dei Babbi Natale

Alberto Bombardelli
Massimiliano Fronza
Stefano Sega
Carlotta Tagliaro
Davide Zanella

Tecnologie utilizzate:

La nostra implementazione di Shopping List utilizza un database MySql per memorizzare tutte le informazioni sugli utenti, sui prodotti e sulle liste. Lato server utilizziamo Java EE seguendo il pattern MVC, mentre lato client utilizziamo JSP e JSTL ma anche librerie come bootstrap, jquery e sweetalert.
Di seguito proponiamo il diagramma ER del nostro database.



Come si può notare abbiamo una tabella per gli Utenti, una per i Prodotti e una per le Liste. Queste tabelle sono tra loro collegate con delle relazioni che sono delle tabelle a loro volta (Utenti_Liste, Utenti_Prodotti, Liste_Prodotti, Liste_Prodotti_Acquistati). Liste_Prodotti e Liste_Prodotti_Acquistati si differenziano solamente perchè nella prima ci sono i prodotti

inseriti in una lista, mentre nella seconda sono quelli che sono stati effettivamente acquistati ed erano in precedenza in una lista. Le tabelle Liste_categorie e Prodotti_categorie servono a memorizzare le possibili categorie di lista e di prodotto. Abbiamo poi delle tabelle specializzate nella memorizzazione delle immagini per le varie entità (Prodotti_immagini, Liste_categorie_immagini) che memorizzano i percorsi sul server in cui si trovano le immagini. Infine ci sono le tabelle Chat e Notifiche, la prima salva i messaggi inviati nella chat di una lista salvando anche il mittente, mentre la seconda serve a memorizzare i consigli di acquisto per gli utenti.

Schermate:

Home:

Appena aperto il sito si potranno subito vedere i prodotti che esso propone, sarà possibile farne una ricerca per nome ed inoltre ordinarli per nome o negozio. Per ogni prodotto si può decidere se aggiungerlo ad una propria lista che deve essere della stessa tipologia del prodotto. Se il prodotto è stato creato da noi sarà inoltre possibile condividerlo con altri utenti andando a selezionarli tra la lista di quelli proposti. Per rendere la visualizzazione più pratica abbiamo implementato la un sistema di paginazione dei prodotti, mostrandone solamente una decina per volta.

Shops:

Nella schermata dei negozi ci verranno mostrati i negozi disponibili e una volta aperto uno di essi verranno mostrate le categorie di prodotti di cui è fornito. Per mostrare le foto dei negozi abbiamo utilizzato la modalità carosello di bootstrap. All'interno di una categoria di prodotti ci si troverà in una pagina molto simile e con le stesse funzionalità della schermata home, in cui si possono ricercare e ordinare i prodotti.

My Lists:

In questa schermata vengono mostrate tutte le liste dell'utente o che sono state condivise con lui. Per ogni lista si possono vedere i prodotti inseriti e la relativa quantità. In base ai permessi che un utente ha sulla lista, egli potrà andare a modificare il nome o la descrizione della stessa, condividerla con altri utenti del sito andando a specificare con quali permessi, eliminare la lista e comprare o eliminare prodotti dalla stessa. Un apposito pulsante permette di creare una nuova lista e tutte informazioni necessarie saranno richieste.

My Products: (visibile solo dopo il login)

In questa schermata un utente loggato potrà visualizzare tutti i prodotti da lui creati e se vuole può crearne di nuovi con l'apposito pulsante. Anche in questa pagina è stata implementata la paginazione.

Messages: (visibile solo dopo il login)

Da questa pagina è possibile visualizzare le proprie chat, ovvero una per ogni lista di cui si fa parte. Nella parte sinistra si può selezionare la lista della quale si vuole vedere la chat, mentre nella parte destra si vedranno tutti i messaggi presenti nella chat selezionata. In basso c'è inoltre un campo testuale nel quale si può inserire il testo del messaggio da inviare

oppure se ne può scegliere uno tra quelli proposti. Infine premendo il pulsante invia questo verrà inviato nella chat.
Per implementare questo sistema di messaggistica è stata implementata una servlet che verifica la presenza di nuovi messaggi oppure ne invia uno se richiamata in post coi corretti parametri. Lato client è stata implementata una funzione JavaScript che viene richiamata a intervalli regolari e si occupa di verificare la presenza di nuovi messaggi e eventualmente aggiornare la chat aperta. Per questo motivo la chat può risultare non esattamente in tempo reale ma c'è bisogno di attendere qualche secondo perchè si aggiorni.

Profile: (visibile solo dopo il login)
In questa schermata vengono mostrate le informazioni dell'utente quali nome, cognome, immagine profilo e gli viene permesso di modificarle e di aggiornare la password.

Login:
In questa schermata vengono richieste solamente l'email e la password di un utente già registrato in precedenza. Inoltre si può decidere se memorizzare il proprio login, nel caso questa opzione venga selezionata verrà inviato un cookie al client, in questo modo, attraverso un apposito filtro ogni volta che si visiterà il sito in un arco temporale relativamente ampio l'utente verrà automaticamente loggato nel sistema senza la richiesta di email e password da quel client.
Nel caso in cui l'utente non si sia registrato sul portale in precedenza può farlo da questa schermata cliccando sulla scritta "register", in questo modo verrà reindirizzato a un form di registrazione e una volta compilato e inviato dovrà confermare la propria email aprendo il link che gli è stato inviato via email contenente un identificativo univoco che aggiornerà il suo record nel db.
Nel caso in cui invece l'utente non ricordi più la sua password dovrà cliccare su "reset password", in modo che, una volta inserita la sua email, gli verrà inviata una nuova password via email.

Logout: (visibile solo dopo il login)
Questa servlet non fa altro che invalidare la sessione dell'utente in modo da fargli fare logout e rimandarlo sulla schermata home.

Ulteriori funzionalità:
Notifiche via WEB:
In qualsiasi pagina del sito, in alto a destra è presente una campanella che se cliccata mostrerà una tendina contenente la lista di notifiche che interessano l'utente in questione. Queste notifiche possono essere di vario tipo:

- Chat:
  in una chat ci sono dei nuovi messaggi che l'utente non ha ancora visualizzato.
- Condivisione Prodotto:
  un utente ha condiviso un suo prodotto con l'utente loggato.
- Condivisione Lista:
  un utente ha condiviso una lista con l'utente loggato, premendo sul titolo della notifica sarà possibile scegliere se accettare o rifiutare la condivisione.

- Consigli di acquisto:
  Il portale consiglia all'utente, in base al suo storico di acquisti, se acquistare un certo prodotto per una certa lista, questo viene fatto con una stored procedure in MySql che, tenendo conto delle quantità e degli intervalli di acquisto di ogni prodotto in ogni lista, sa consigliare l'utente.

Notifiche via Email:

A ogni utente registrato, in seguito a controlli giornalieri, viene inviata un'email se, secondo il portale, l'utente potrebbe avere necessità di acquistare un determinato prodotto. Questo viene fatto con delle funzioni di java schedulate.

Geolocalizzazione:

Il browser chiederà all'utente il permesso di poter visualizzare la sua posizione, questo perché, tramite una funzione JavaScript schedulata, andrà a richiedere al server di cercare se ci sono negozi vicini che sono compatibili con le lista dell'utente loggato o anonimo che sia. Per fare questo, lato server c'è una servlet apposita che, con delle chiamate alle API di HERE maps riesce a capire se ci sono dei negozi sufficientemente vicini ed in caso segnalarlo all'utente con dei bootstrap alert.

Dettagli implementativi:

Custom tags:

Abbiamo implementato dei custom tag JSTL parametrizzati con gli oggetti che avevamo necessità di mostrare all'utente. Per questo abbiamo creato dei custom tag per mostrare le lista, i prodotti, i negozi e le categorie di prodotti. In questo modo abbiamo reso più comodo lo sviluppo di molte pagine rendendole più intuitive e leggibili.

Servlet pages:

Per mantenere distinta la business logic dalla view abbiamo creato delle servlet che si interpongono tra le pagine jsp e il DAO in questo modo i livelli rimangono distinti e tutti i controlli e l'elaborazione dei dati rimangono scollegati dalla loro visualizzazione.

Servlets:

Abbiamo creato numerose servlets specializzate ognuna in una specifica funzione, queste servlets vengono richiamate tramite submit di un form in una pagina jsp oppure tramite chiamate JavaScript e solitamente si occupano di reindirizzare a altre pagine oppure inviare dei dati come risposta una volta terminata la loro elaborazione.

Filters:

Abbiamo creato alcuni filtri il cui scopo è quello di fare dei controlli in modo da evitare questo compito alle servlets e rendere più sicuro e mantenibile l'intero portale. In particolare un filtro si occupa di verificare che chi cerca di accedere a certe pagine abbia effettuato l'accesso onde evitare che chiunque riesca a vedere dati di altri o pagine di cui non ha l'autorizzazione. Un secondo filtro invece si occupa di verificare se il client ha inviato uno

specifico cookie in modo da far fare il login automatico al client così da evitargli una lunga procedura (come spiegato in precedenza nella sezione del login).

DAO:

Per gestire la connessione al database abbiamo utilizzato la metodologia DAO, definendo le entità del database, delle interfacce per le classi DAO, una loro implementazione JDBC ed infine il DAO Factory in modo che se in futuro decidessimo di cambiare la tipologia di database questa migrazione potrebbe avvenire in maniera facile e con pochi cambiamenti.