

DressAdvisor

Federica Pecci

Chiara Varini

25 maggio 2015

Indice

1 - ANALISI

1.1 Requisiti.....p. 3

1.2 Problematiche.....p. 3

2 - DESIGN

2.1 Architettura.....p. 5

2.2 Design dettagliato.....p. 7

3 - SVILUPPO

3.1 Testing automatizzato.....p. 9

3.2 Divisione dei compiti e metodologia del lavoro.....p. 9

4 - VALUTAZIONI

4.1 Conclusioni e lavori futuri.....p. 11

APPENDICE - GUIDA UTENTE.....p. 12

1 - ANALISI

DressAdvisor nasce grazie alla passione delle sue sviluppatrici per il mondo della moda e come scopo primario vuole essere una innovativa piattaforma a forte vocazione "social".

1.1 Requisiti

L'applicazione è stata concepita fundamentalmente come semplice gestionale di un archivio relativo a negozi di abbigliamento. In particolare, l'applicazione si offre a due categorie di utilizzatori: negozianti, i quali possono gestire le informazioni dei propri punti vendita e clienti, che possono cercare i negozi in base a criteri pre-impostati per poi eventualmente recensirli.

Più in dettaglio, le principali funzionalità offerte sono:

- per i negozianti:
 1. registrazione di uno o più negozi;
 2. cancellazione;
 3. modifica delle specifiche;
- per i clienti:
 1. ricerca di un negozio;
 2. recensione e valutazione di un negozio.

1.2 Problematiche

Le difficoltà affrontate in fase di progettazione sono state le seguenti:

- necessità, per l'utente che intenda inserire informazioni sui propri negozi, di registrarsi e quindi successivamente effettuare il login come negoziante, scegliendo opportunamente username e password. La criticità intrinseca in questa gestione, consiste nell'amministrare opportunamente i diritti di accesso alle sezioni private;

- evitare la ripetizione di due negozi uguali, sviluppando precisi criteri di inserimento di ogni nuovo negozio;
- calcolare i punti assegnati ad ogni negozio in base alle recensioni associate gestendo eventuali punteggi parimerito. Ciò è determinante al fine di stilare correttamente e in maniera univoca la classifica dei negozi presenti in archivio;
- caricare e salvare ogni nuova tipologia di dato inserito durante ogni utilizzo dell'applicazione, in quanto i dati presenti in archivio (negozi e negozianti) devono necessariamente permanere all'interno del software anche dopo la chiusura di quest'ultimo.

2 - DESIGN

2.1 Architettura

DressAdvisor è stata sviluppata seguendo il pattern architetturale *MVC (Model-View-Controller)* poichè questo è in grado di separare la logica di presentazione dei dati dalla logica grafica, rendendo disponibile una suddivisione del codice in blocchi, evidenziando funzionalità tra loro distinte. Ne consegue ovviamente che cambiamenti radicali della *view* non comporta modificazione modifica del *model*.

Nello specifico il *model* contiene i metodi che implementano funzionalità

- creazione di un negozio e di una
- redazione di una recensione
- modifica di un negozio
- gestione degli archivi (aggiunta/rimozione/modifica negozi e negozianti).

Per quanto riguarda le strutture dati impiegate si è fatto uso di:

- mappe, per quanto riguarda la gestione dell'archivio dei negozianti,
- set, per la gestione dell'archivio dei negozi.

L'inserimento di negozi nell'applicazione, si è deciso di svilupparlo partendo dall'identificazione univoca dei negozi in base a criteri oggettivi quali nome, città e indirizzo, in quanto non possono essere associati due esercizi commerciali uguali relativamente alla medesima ubicazione. Di conseguenza il metodo *equals* della classe *NegoziSocial* è stato implementato tenendo conto solo di quest'ultimi parametri. Ovviamente l'inserimento di un nuovo negozio nell'*ArchivioNegozi* si basa proprio su questo metodo.

Nel risolvere il problema della gestione della classifica abbiamo, innanzitutto, creato due enum:

- *Punteggi*, che contiene un insieme di giudizi per il negozio (da scarso a eccellente)
- *Valutazioni*, che racchiude un insieme di campi in base ai quali un generico utente può giudicare un negozio più o meno buono in base a criteri soggettivi quali stile, qualità/prezzo, tessuti....

La scala di valutazione è stata fissata partendo da un minimo di 0 punti fino ad un massimo di 5. All'interno della classe Recensione sono stati inseriti opportuni campi per poter tener traccia del giudizio espresso.

Essendo ogni negozio associato a delle recensioni, si è pensato di accedere all'insieme dei valori relativi alla totalità delle recensioni per farne una media e giungere quindi al calcolo e alla redazione della classifica. Quest'ultima viene creata ordinando i negozi in base ad un punteggio e, in caso di parità, ricorrendo all'ordine alfabetico: tutto questo è implementato dal `NegoziComparator` che viene utilizzato dal `Set<INegoziSoial>` situato nell'`ArchivioNegozi`. L'algoritmo specifico che si occupa del calcolo dei punti rispetto ad ogni recensione è stato isolato utilizzando il pattern strategy. Il set ordinato viene poi inserito in una mappa, anch'essa presente nell'archivio, tenendo conto della posizione di ogni negozio.

La *view* è composta da un'interfaccia principale che offre due percorsi diversi all'utente: uno riservato al negoziante ed uno al cliente. Entrambi i percorsi portano ad una serie di altre interfacce con le quali l'utente può interagire. Per la gestione approfondita della view si rimanda all'appendice A.

Nell'affrontare il caricamento e salvataggio dei dati, sono state serializzate le classi del model, così da poterle scrivere su file. Di seguito sono elencati alcuni metodi gestiti dalla classe Controller:

- `carica(nomeArchivioNegozi, nomeArchivioNegozianti)`: metodo invocato della classe Main all'avvio del programma che passa i due file binari salvati nella cartella archivi del progetto
- `salva(nomeArchivioNegozi, nomeArchivioNegozianti)`: metodo che si occupa del salvataggio di tutte le modifiche apportate all'applicazione durante il suo utilizzo. Il Salvataggio delle modifiche avviene con una interfaccia grafica di chiusura.

- `chiudi()`: viene invocato quando si vuole chiudere il `JFrame` dell'applicazione e si occupa di invocare il metodo `salva()`, in seguito anche `System.exit(0)`.

2.2 Design dettagliato

La struttura di `DressAdvisor` si fonda su due concetti cardine:

- un primo aspetto riguarda l'associazione tra i negozianti ed i negozi, perchè ad ogni negoziante è associato un set di negozi;
- un secondo aspetto è la relazione tra recensioni e negozi, in quanto ad ogni negozio è associato ad un insieme di recensioni.

Le strategie messe in atto per risolvere i problemi descritti in analisi sono rispettivamente le seguenti:

- nella parte di controller è stata modellata la classe `ControllaLogin` che gestisce la registrazione di nuovi oggetti della classe `User`, il cui tipo è `INegoziante`, e quindi il loro accesso. La classe `User` rappresenta un generico utente che usufruisce dell'applicazione e la classe `Negoziante` delinea la figura dell'utente che vuole inserire i propri negozi;
- nella parte di model è stata aggiunta la classe `NegoziSocial` che dà forma ad un generico negozio poichè estende la classe `Negozi` aggiungendo i concetti di recensione, punteggio e valutazione. A tal proposito, i metodi più rilevanti della classe `NegoziSocial` sono `getPunteggio()` e `getPuntiValutazione()`: il primo prende in ingresso un dato valore di punteggio e restituisce il numero di recensioni che hanno votato quello specifico valore, mentre il secondo prende in ingresso una determinata valutazione e restituisce il punteggio medio che le recensioni hanno assegnato rispetto al valore;
- per il confronto di due negozi è stata creata una classe `NegoziComparator` che rappresenta un comparatore fra due classi che implementano l'interfaccia `INegoziSocial`;

- per la gestione dell'archivio sono state create alcune classi nel model, le quali implementano l'interfaccia Serializable.

Al fine di garantire che ci sia una sola istanza di una certa classe è fatto ricorso al pattern Singleton.

3 - SVILUPPO

3.1 *Testing automatizzato*

Sebbene siano stati realizzati dei test non automatizzati sull'applicazione, si è ritenuto opportuno non inserirli nel progetto finale.

3.2 *Divisione dei compiti e metodologia del lavoro*

Le funzionalità di DressAdvisor sono state suddivise tra le due proponenti come di seguito riportato:

Chiara Varini

- realizzazione delle interfacce grafiche per la parte della registrazione, del login e dell'aggiunta e/o modifica di un negozio;
- realizzazione della parte di model relativa al negozio;
- realizzazione del controller pertinente ai negozi e negozianti

Federica Pecci

- realizzazione delle interfacce grafiche per la ricerca di un negozio e l'aggiunta di una recensione;
- realizzazione della parte di model relativa alla recensione;
- realizzazione del controller pertinente alle recensioni;
- realizzazione User.

Le parte sviluppate in comune sono le seguenti:

- schermata iniziale dell'applicazione GUI;

- classi archivi;
- controller gestione archivi;
- classe main;

Inizialmente le due sviluppatrici hanno pensato a quali GUI era necessario realizzare per garantire all'utente un'esperienza gradevole nell'interazione con l'applicazione e hanno impostato insieme la GUI che rappresenta la schermata principale di DressAdvisor. In seguito, sono state realizzate separatamente le varie interfacce grafiche, le interfacce del model la relativa implementazione, le interfacce-implementazione del controller secondo la ripartizione di cui sopra. Per la realizzazione del progetto è stato utilizzato Il DVCS Mercurial.

4 - VALUTAZIONI

4.1 Conclusioni e lavori futuri

In fase di progettazione del software è stato ponderato l'utilizzo di vari pattern e, dopo un'accurata valutazione, sono stati scelti e utilizzati quelli ritenuti più idonei per l'applicazione in questione.

DressAdvisor è ovviamente un progetto molto vasto in termini di packages ed elementi in essi contenuti in quanto le funzionalità di base precedentemente elencate nei requisiti, sono state arricchite con strumenti adeguati per la loro personalizzazione. Ad esempio, per quanto concerne gli aspetti legati alle recensioni, l'interfaccia per l'aggiunta di una recensione offre al cliente la possibilità di inserire un punteggio per il negozio, da scarso a ottimo, una valutazione dei campi stile, tessuti, qualità/prezzo, raggiungibilità.

Sebbene l'appendice A offra una semplice ed intuitiva guida utente, le due sviluppatrici ritengono di aver realizzato il software in modo tale che si possa prendere confidenza sin dal primo momento con la logica di funzionamento dell'applicazione, che è stata intesa sin dal principio quanto più possibile *user-friendly*.

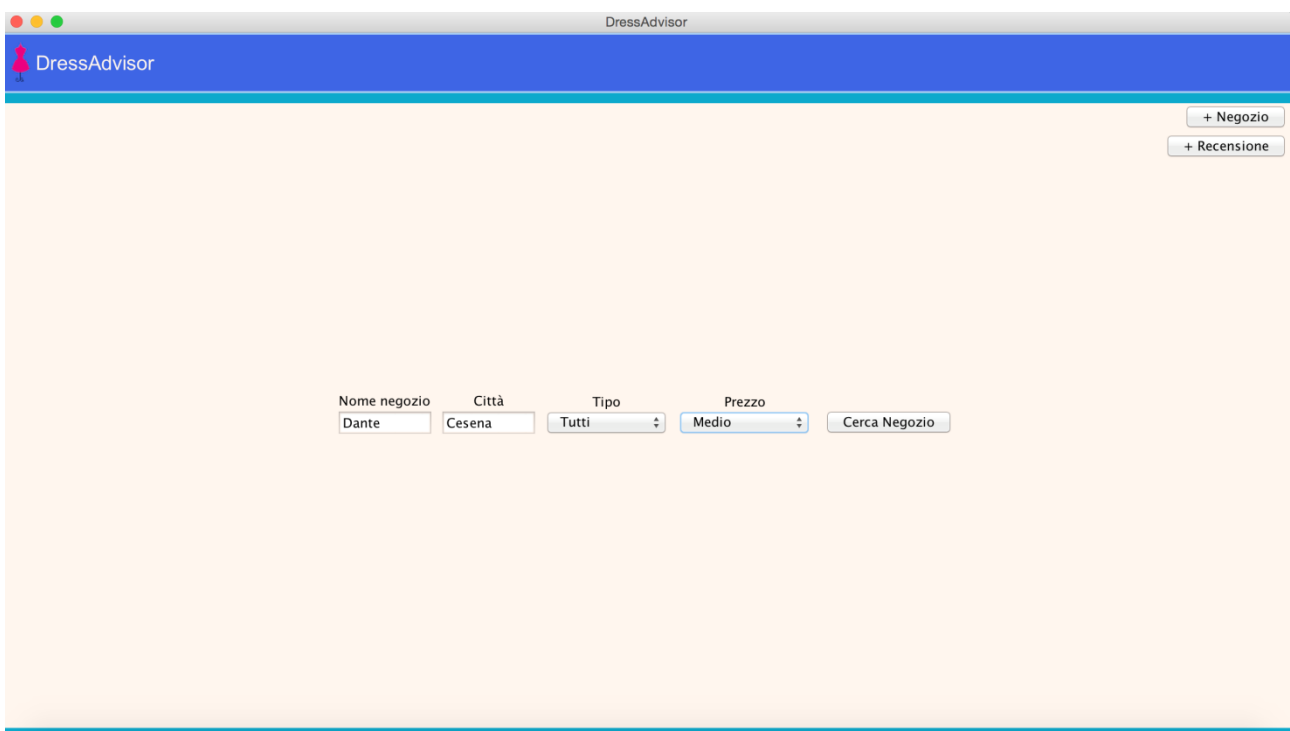
Un punto debole risiede certamente nel fatto che un negoziante possa recensire il proprio negozio, il che ovviamente mostra delle pecche dal punto di vista concettuale.

Per il futuro le autrici si propongono di integrare il progetto con mappe, utilizzando le API di Google, per individuare dove fisicamente sono dislocati i negozi sul territorio. Infine sarebbe fortemente auspicabile sviluppare l'applicazione dal punto di vista web e mobile.

APPENDICE - GUIDA UTENTE

All'avvio dell'applicazione, si apre la schermata iniziale del programma e l'utente può decidere di svolgere tre azioni differenti.

La prima azione è inserire uno o più campi nella parte centrale dell'interfaccia e poi premere il bottone "Cerca Negozio" per completare la ricerca. Se la ricerca viene effettuata senza inserire alcun parametro, allora verranno mostrati tutti i negozi presenti nell'archivio attualmente.



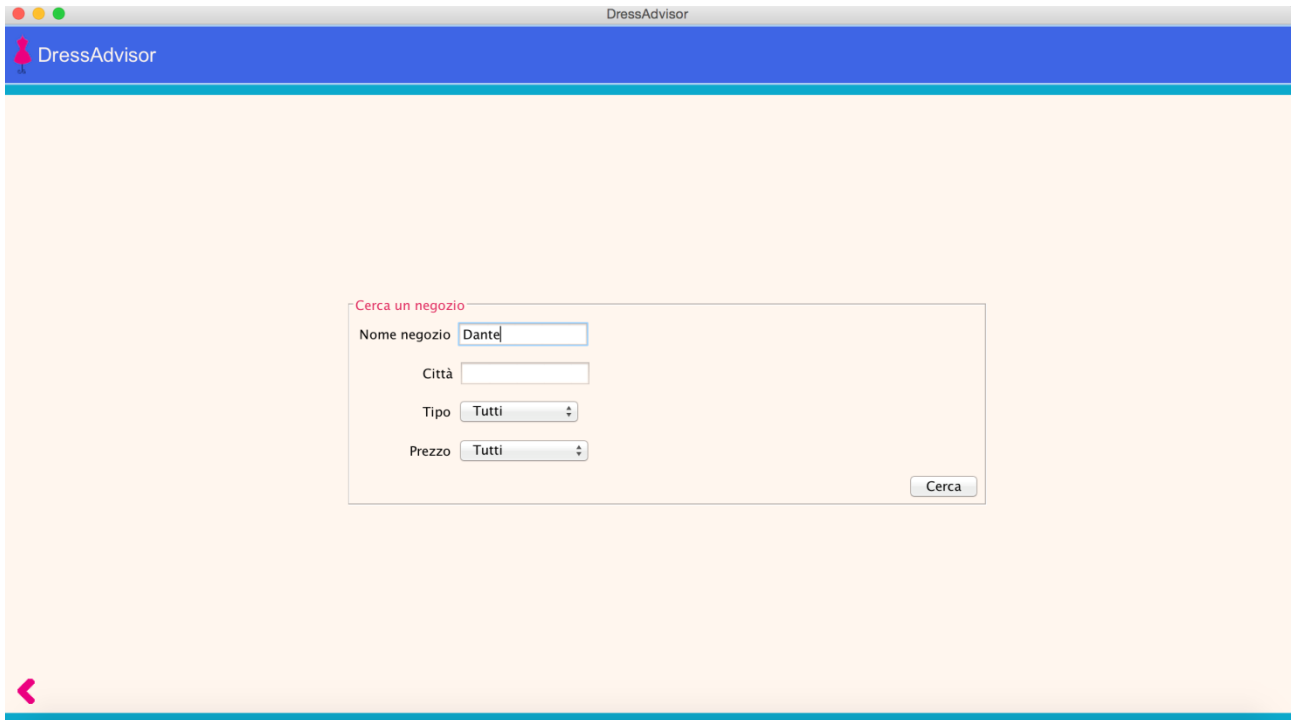
La seconda azione disponibile è premere il bottone "+ Negozio", in alto a destra, così che l'utente possa registrarsi come negoziante. La GUI che appare è la seguente:

The screenshot shows a web browser window titled "DressAdvisor". The page has a blue header with the "DressAdvisor" logo. The main content area is light beige and contains a registration form titled "Registrazione". The form has four input fields: "Nome" (Federica), "Cognome" (Pecci), "Mail" (federica.pecci@hotmail.it), and "Password" (password). A "Registra" button is located at the bottom right of the form. A pink arrow points to the bottom left corner of the page.

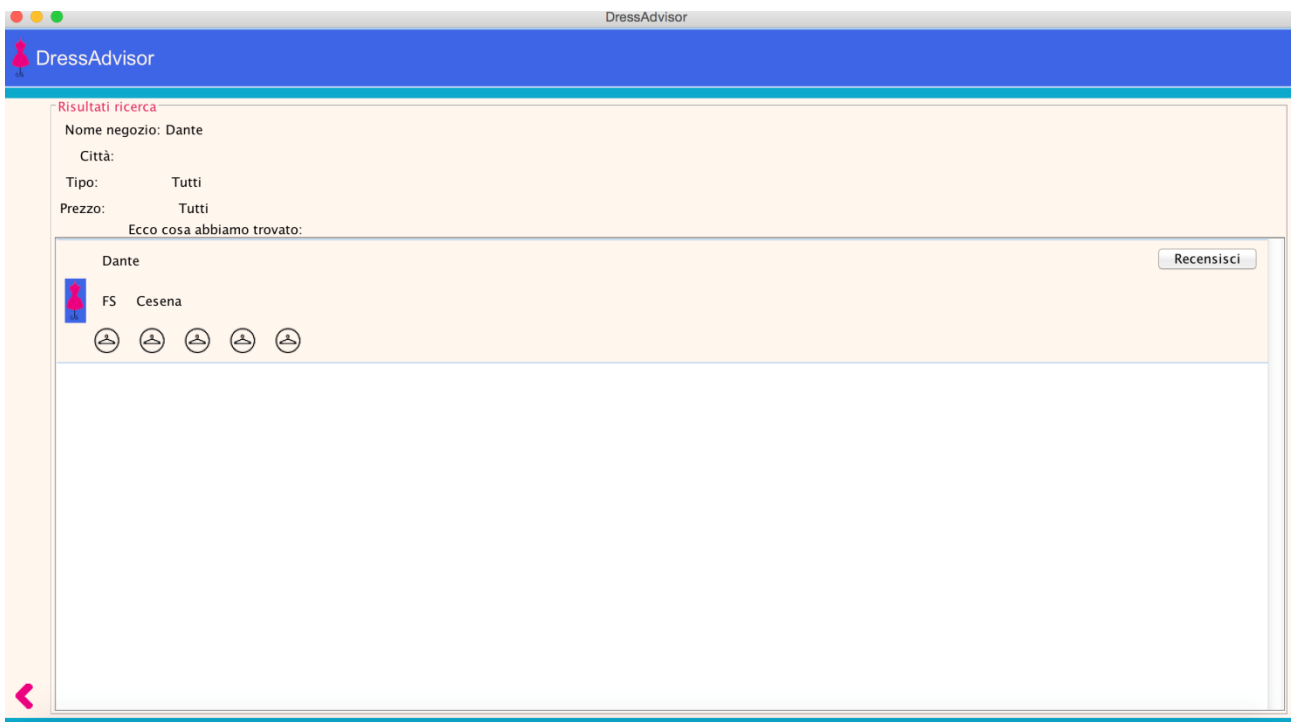
In seguito apparirà, in alto a destra, un messaggio di benvenuto per il negoziante ed esso potrà direttamente aggiungere i negozi che desidera aggiugendoli al proprio profilo.

Se il negoziante effettua il logout dovrà rifare il login per accedere nuovamente alla sua area negoziante. Per fare il login devono essere inseriti lo username e la password, ricordando che lo username è dato dalla combinazione di nome e cognome separati da un punto, quindi se il nome del negoziante è "Federica" ed il cognome "Pecci", allora lo username sarà "Federica.Pecci".

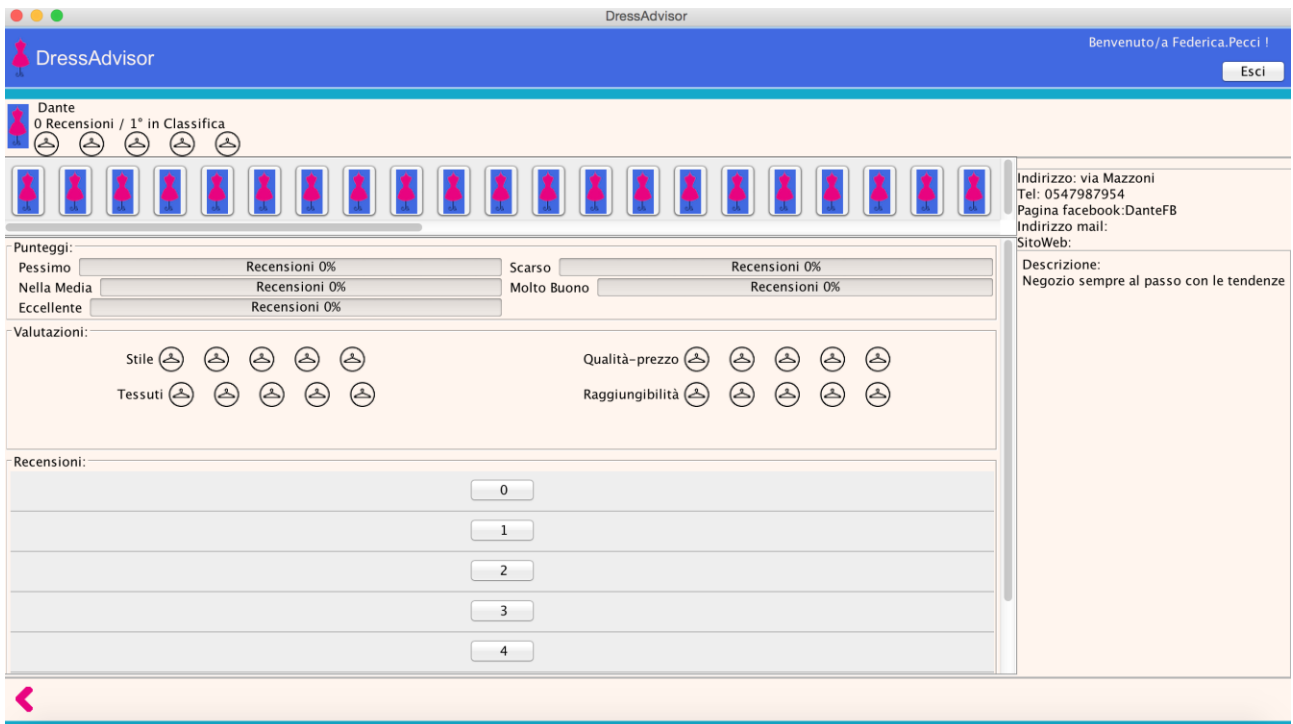
La terza azione che può essere eseguita è premere il bottone "+ Recensione" posto sotto il bottone "+ Negozio", per poter effettuare la ricerca in base ai campi inseriti. La schermata che si apparirà sarà la seguente:



Poi, una volta premuto il bottone "Cerca", verrà visualizzata un'altra GUI che mostra una serie di negozi frutto della ricerca. Tra questi negozi sarà possibile selezionarne uno e recensirlo.



Conseguentemente all'aggiunta di un negozio o di una recensione il programma ricondurrà sempre alla pagina del negozio appena creato o appena recensito dall'utente.



Si offre infine la modalità di visualizzazione, se si desidera, delle recensioni che sono già state aggiunte per il negozio in questione, ricordando che una volta inserite, non sarà possibile rimuoverle.