



# Sapienza – Università di Roma

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica  
Corso di laurea in Ingegneria Informatica

Progetto di Reti e Sistemi Informatici  
A.A. 2014-2015

# SHELF

*“Do it your-shelf!”*

Progetto di:  
Alessio Pece  
Lorenzo Gallucci

## **Descrizione del Progetto**

Shelf è una piattaforma di Social Collecting che permette ad un utente di creare la propria libreria virtuale di libri, film o giochi. Grazie alla natura “Social” dell’applicazione, è possibile seguire ed essere aggiornato su tutte le azioni degli altri utenti e, allo stesso tempo condividere le modifiche alla propria libreria.

### **1 – Requisiti di Sistema**

Per poter eseguire correttamente Shelf è necessario che sulla macchina siano installati i seguenti software:

- sqlite3
- nodeJS
- gcc
- unzip
- Ruby >= 2.2.2
- RubyOnRails >= 4.2.2

L'applicazione è fruibile da un qualsiasi browser moderno (Safari, Opera, Firefox, Chrome, ecc.).

### **2 – Tecnologie impiegate**

L'intero progetto è stato sviluppato sfruttando il framework RubyOnRails basato sul linguaggio Ruby. Si è scelto di utilizzare RubyOnRails vista la semplicità di utilizzo e la possibilità di realizzare applicazioni web dinamiche anche molto complesse.

Le pagine statiche e dinamiche sono state realizzate attraverso l'uso congiunto di linguaggio HTML5 e Ruby sfruttando la funzione di RubyOnRails che permette la stesura di codice misto salvato in alcuni file di estensione “.html.erb” interpretati dal server di RubyOnRails. Massiccio anche l'uso di CSS, sfruttando le classi del framework Bootstrap per una resa finale visivamente più elegante.

Come server SQL è stato utilizzato Sqlite3 poiché molto semplice da configurare.

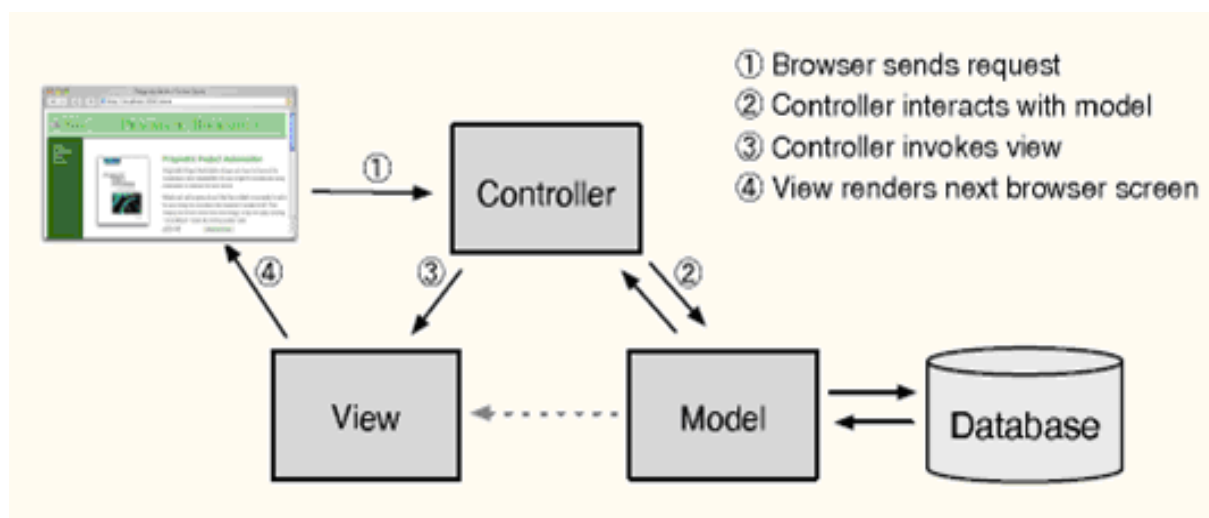
Tutti i dati sensibili, quali passwords e chiavi di sessione, sono salvati nel

database in forma criptata usando la funzione Bcrypt.

### 3 – Architettura di Sistema

Shelf è un'applicazione basata sul sistema client-server:

- Il client è il browser web
- Il server è offerto da RubyOnRails ed è così strutturato:



- I **Controller** contengono tutta la logica applicativa e gestiscono il comportamento dell'applicazione a seguito di un evento.
- I **Model** si occupano di rappresentare gli “oggetti” della nostra applicazione memorizzati nel database.
- Le **View** hanno il compito di interagire con il client raccogliendo gli input e mostrando il risultato delle computazioni con render HTML.

### 5 – Deploy del sistema

Per mettere in opera il sistema è sufficiente seguire i seguenti passi:

- Scaricare il file “shelf\_app\_prsi-master.zip” dall'indirizzo [https://github.com/alessiopece/shelf\\_app\\_prsi](https://github.com/alessiopece/shelf_app_prsi) e salvarlo nella directory di Download.
- Eseguire da terminale i comandi:

```
~$ cd /home/user/Downloads
```

```
~$ sudo unzip shelf_app_prsi-master -d .
```

```
~$ cd shelf_app_prsi-master
```

```
~$ sudo sh install.sh
```

*N.B Potrebbe essere richiesto di inserire la password di amministratore del*

Adesso sarà necessario solo recarsi da browser all'indirizzo <http://localhost:3000>

Il codice del file di script “install.sh” è il seguente:

```
#!/bin/bash
```

```
#Download ed installazione delle “Ruby Gem” necessarie all'applicazione  
bundle install
```

```
#Creazione delle tabelle nel database e configurazione iniziale dello stesso  
bundle exec rake db:migrate  
bundle exec rake db:seed
```

```
#Comando di start del server RoR  
rails server
```

## 5 – Contributi personali

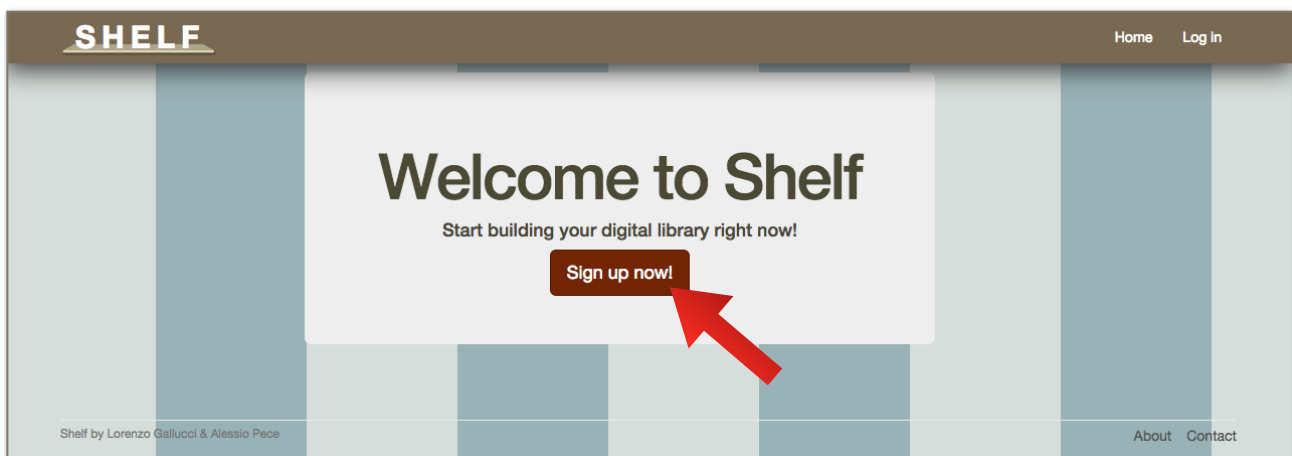
- **Lorenzo Gallucci** si è occupato di modellare gli utenti e di realizzare tutta la logica associata ad essi. La registrazione di un nuovo utente è consentita attraverso la compilazione di una form che raccoglie tutti i parametri necessari alla creazione dell'utente nel database purché essi rispettino i vincoli di formato imposti (ad esempio la lunghezza minima della password o la validità dell'indirizzo email). Di particolare importanza l'utilizzo della funzione Bcrypt per la memorizzazione del campo “password” nel database in forma criptata. Qualora la mail dell'utente fosse associata al servizio “Gravatar” il sistema estrapolerà in automatico l'avatar per l'utente in questione, contrariamente sarà utilizzato un avatar generico. È stato realizzato anche un controller di sessione per l'autenticazione di un utente con conseguente rilascio di un token da depositare nei cookie del browser con scadenza temporale variabile (è presente infatti il tick “remember me” in fase di autenticazione). Tuttavia i cookie persistenti sono vulnerabili al cosiddetto “Session Hijacking”. Dunque il sistema di autenticazione consiste in un sistema di scambio di token e codici identificativi, criptati e non, tra il server e il browser, volto a ridurre la vulnerabilità. Inoltre è stato implementato un sistema di following/follower in stile “Twitter”.
- **Alessio Pece** ha realizzato i modelli associati a tutti gli oggetti che popolano l'applicazione, quali le “shelves” o i vari “item” (libri, film e giochi). Si è occupato di gestire tutta la complessa logica di dipendenze

tra i diversi “oggetti” del sistema grazie alle funzioni “belongs\_to”, “has\_many” e “dependent” offerte da ruby. Ha implementato il sistema di eventi all’interno dell’applicazione, quali la creazione/rimozione di “shelves” sulla propria pagina utente e l’aggiunta/rimozione di “items”. Alcuni eventi all’interno dell’app generano un “feed”, ovvero un oggetto dotato di un messaggio che descrive l’azione appena effettuata. Questi feed vanno a popolare la home dell’utente in base alla relazione following/follower in modo da mostrare tutte le attività svolte dalle persone “seguite”. Si è inoltre occupato di curare nel dettaglio l’interfaccia grafica dell’intera applicazione, in particolare sfruttando le classi CSS offerte dal framework Bootstrap. Grazie a questo framework è stato semplice creare la struttura “a libreria” delle pagine profilo di ogni utente, creando un layout scalabile a seconda del dispositivo (desktop o mobile).

## 6 - Casi di test

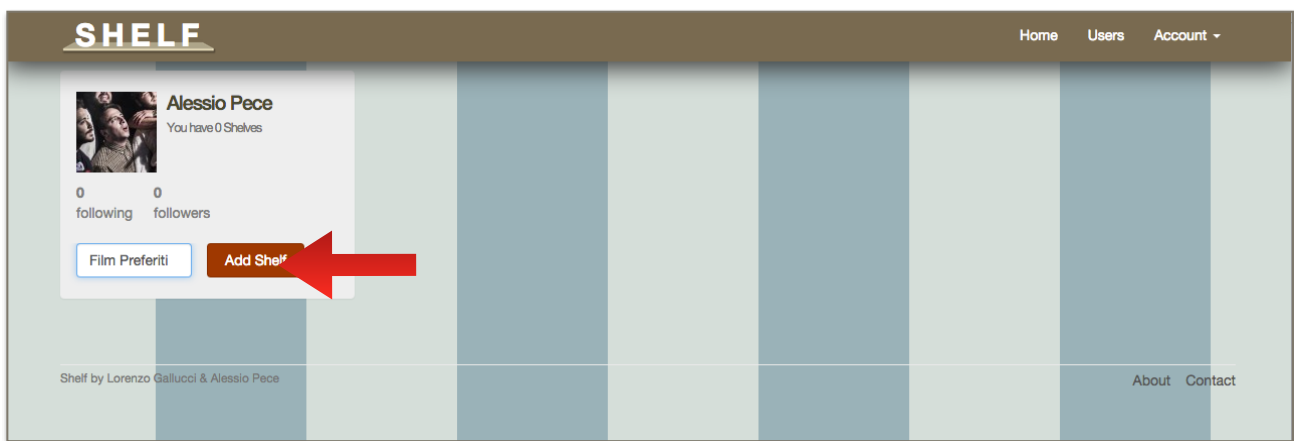
È possibile testare l’applicazione iscrivendosi alla piattaforma ed eseguendo le possibili azione che essa offre:

- Recarsi all’indirizzo “<http://localhost:3000>”.
- Registrarsi cliccando su “Sign up now!” o effettuare il login se si è già

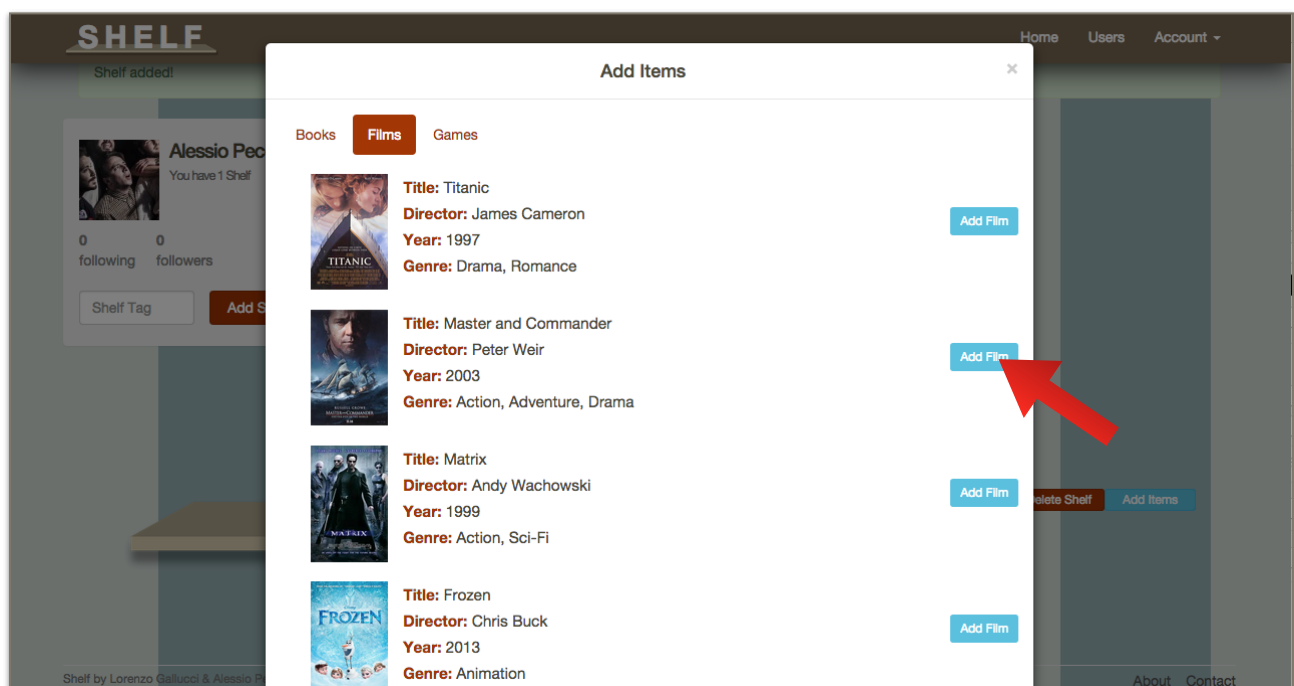
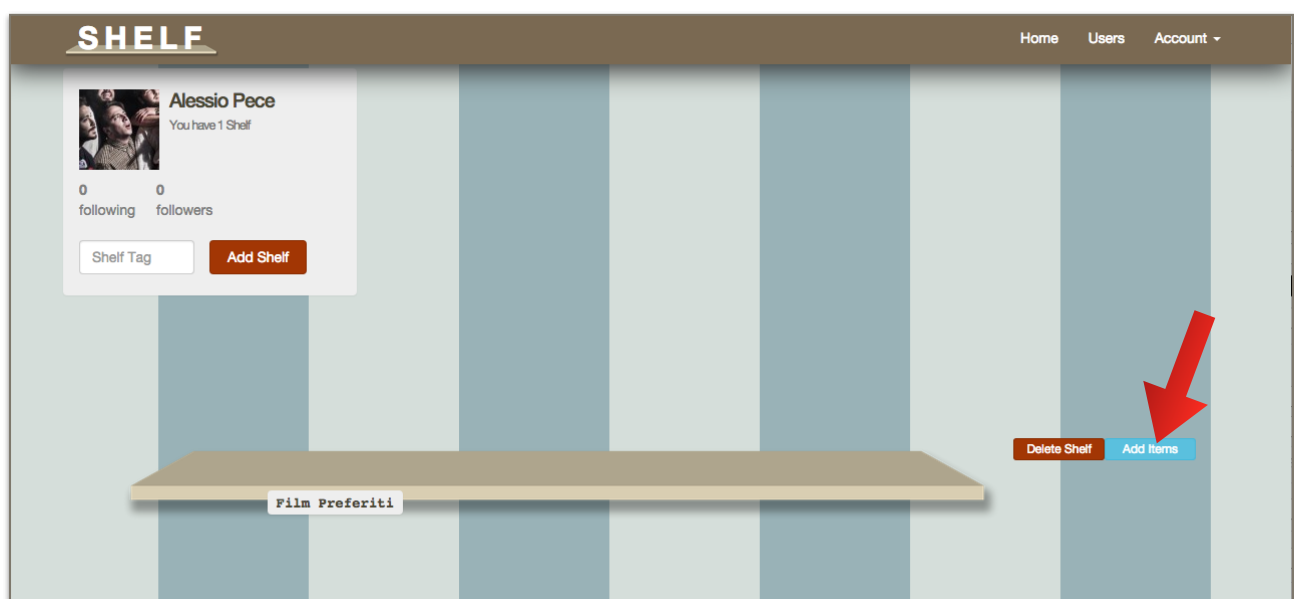


registrati.

- Si verrà reindirizzati sulla propria pagina del profilo, dove sarà possibile creare nuove “shelves” ed associarvi un’etichetta, cliccando “Add Shelf!”.

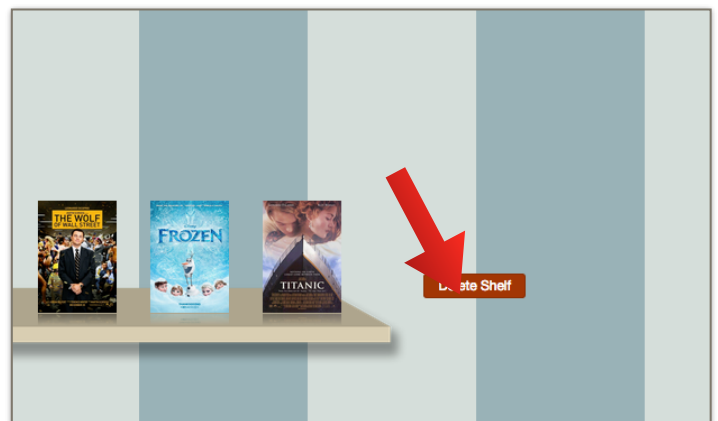
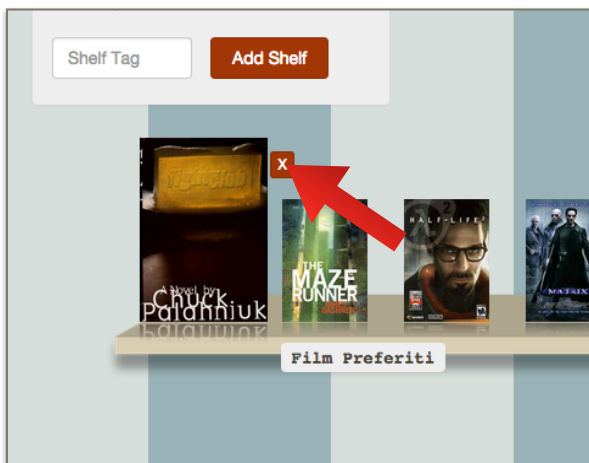


- Si verrà a creare una “shelf” sulla quale posizionare nuovi oggetti. Questo è reso possibile cliccando sul tasto “Add Item”, che aprirà un popup che mostrerà la lista di possibili libri/film/giochi da poter posizionare sulla

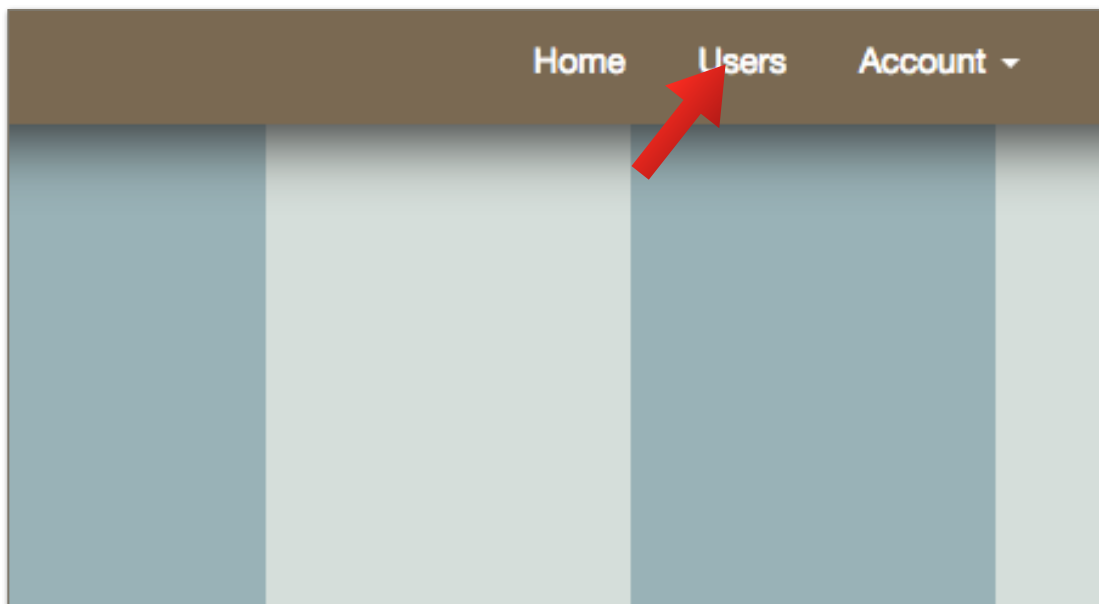


mensola.

- È possibile posizionare fino ad un massimo di 8 “items” per mensola. Gli oggetti saranno già disposti sulla “shelf” creata e sarà possibile eliminarli cliccando sul pulsante X in comparsa sui singoli “items” al passaggio del mouse. È inoltre possibile eliminare l’intera “shelf” (con i relativi “items”) cliccando il pulsante “Delete Shelf”.

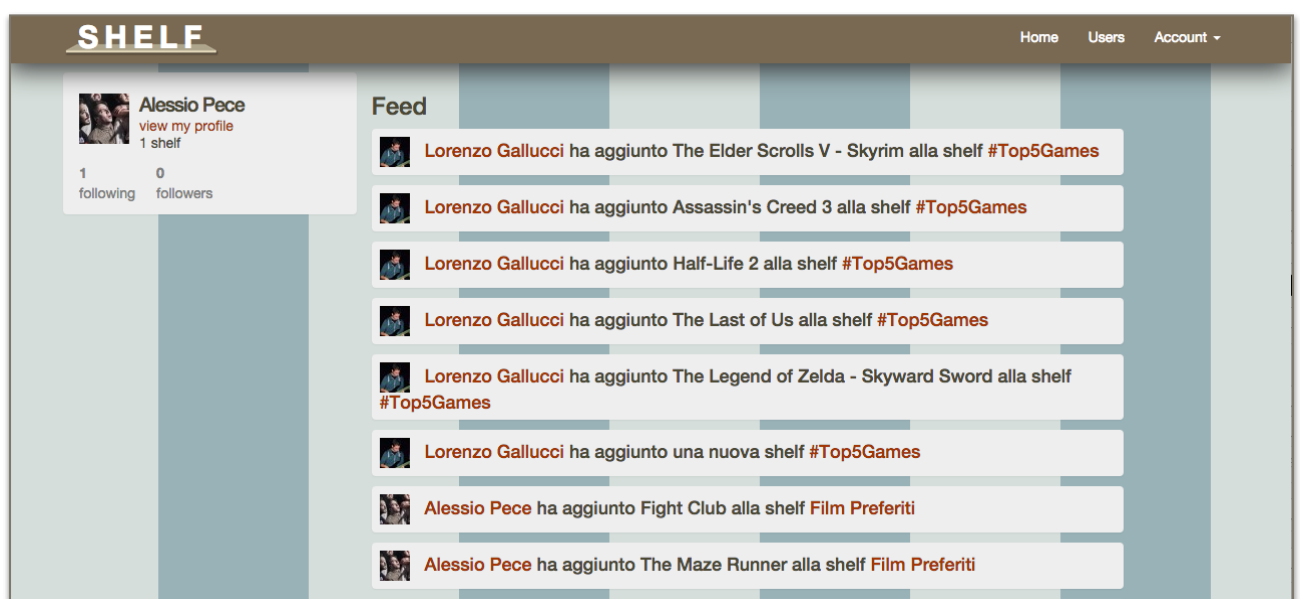
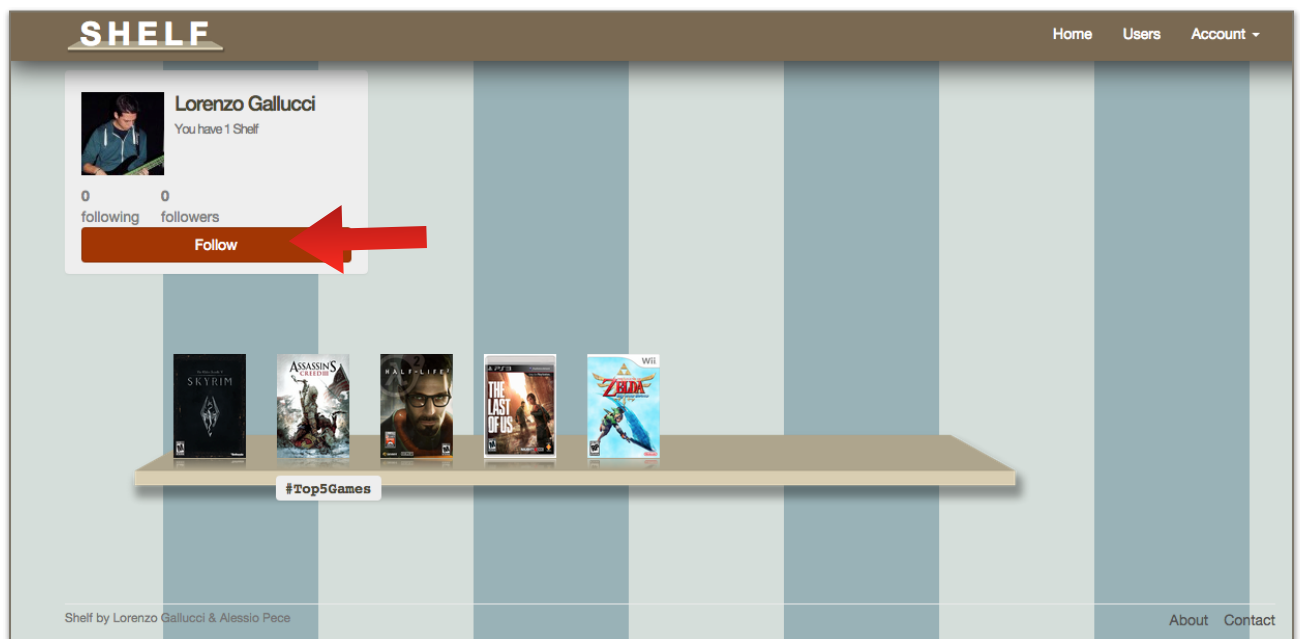
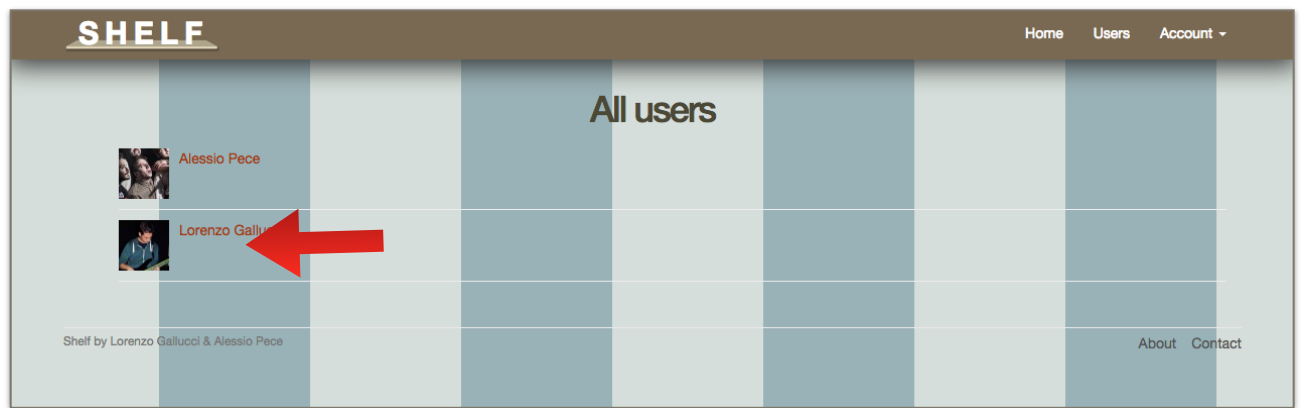


- Cliccando su “Users” si può visionare un elenco di tutti gli utenti attualmente iscritti all’applicazione. Da qui sarà possibile accedere ai profili

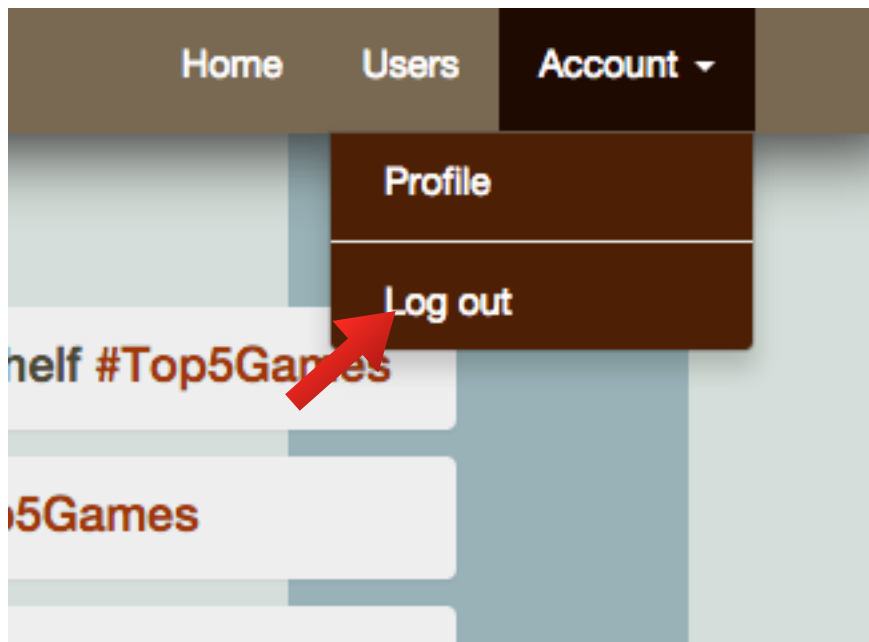


dei  
vari

utenti iscritti, con l’opzione di “seguirli”.







- Seguendo un utente, tutte le azioni effettuate da esso genereranno un feed sulla nostra home
- È infine possibile eseguire il log out.