

# Relazione Secondo Progetto

Introduzione alla programmazione web

AA 2012/2013

Daniel Frassinelli 145687

Casera Eric 150027

Alessandro Lotto 145445

## Struttura del progetto

### Cartelle dei file :

- PrimoProgetto/Jsp/\* : cartella delle pagine jsp
- PrimoProgetto/Images-site : cartella dove vengono salvate le immagini utilizzate solo per il sito
- PrimoProgetto/Images : cartella delle immagini usate per categorie/prodotti

### Pattern delle servlets:

- GuestController?op : gestisce login , recupero password , registrazione nuovo utente
- LogoutController : gestisce il logout
- General/GeneralController?op : gestisce le pagine che possono essere visitate da tutti
- User/UserController?op : gestisce le pagine per le offerte e l'aggiunta di una nuova asta
- User/Upload : servlet per l'upload delle immagini
- User/AccController?op : gestisce tutte le pagine dell'account di un utente
- Admin/AdminController?op : gestisce le pagine dell'amministratore

### Database:

- Il database è stato creato utilizzando derby
- Il database è la cartella database/DBSecondoProgetto
- Autenticazione : username : database , password : database
- Per connettersi usare i client driver

Il sito è stato sviluppato secondo un pattern mvc. Ogni request viene indirizzata ad una servlet che ne controlla l'input , fa le query , setta gli attributi nella request e fa il forward alla pagina jsp corrispondente.

La pagina jsp si occupa solo della visualizzazione dei dati settati dalla servlet tramite tag JSTL.

Le pagine non sono accessibili se non tramite servlet (un filtro blocca qualsiasi request su jsp/\*).

La gestione delle aste è organizzata così :

- Una nuova asta viene inserita tra le aste attive e automaticamente copiata tra le aste finite.
- Tutti i riferimenti vengono fatti sulla tabella delle aste finite (un asta viene rimossa dalle aste attive quando scade).
- Le offerte utilizzano il sistema di ebay , un utente mette un prezzo massimo (la congruenza con l'incremento viene controllata direttamente dalla pagina) e il sito esegue l'autoincremento in base alle offerte degli altri utenti.
- Ogni minuto un metodo controlla tutte le aste nella tabella delle aste attive , prende quelle scadute e le elimina. Dopo questo setta vari flag nella tabella delle aste finite (se l'asta è stata venduta , cancellata o ritirata) , in caso di vendita aggiunge un record nella tabella delle vendite.
- Quando un admin cancella un asta , l'asta rimane comunque tra le aste attive fino alla sua scadenza. Ovviamente l'asta apparirà come "cancellata" tra le aste attive sul sito e quindi sarà impossibile per gli utenti fare offerte.

## **Le classi sono suddivise nei seguenti package:**

- Beans
- Managers
- Filters
- Listeners
- Servlets

### **Beans**

- Contiene tutti i beans usati per trasportare i dati agevolmente.

### **Filters**

- AdminFilter , Admin/\* : Filtra tutte le request per le pagine dell'amministratore. Solo un amministratore può accedere a questo pattern , altrimenti viene ridirezionato alla homepage
- UserFilter , User/\* : Filtra tutte le request per le pagine accessibili solo da un utente. Se non si è loggati come utente si viene ridirezionati alla homepage o alla pagine di login.
- GuestFilter , GuestController : Come per gli altri filtri. Non permette ad un utente loggato di accedere alle pagine di Login , Recupero password , Iscrizione.
- JspFilter , jsp/\* : Blocca solo le request ma non i forward. Solo le servlet possono quindi accedere alle pagine jsp

### **Managers**

- DBManager : classe condivisa da tutte le servlet in cui ci sono tutte le query
- DBChecker : metodo che ogni minuto controlla il database. Per ogni asta scaduta decide se ritirarla o venderla. Dopo questo cancella dalla tabella "aste attive" tutte le aste scadute e le inserisce nelle "aste terminate"
- EmailManager : classe che gestisce l'invio delle mail e la creazione al volo del foglio excel

### **Listener**

- ContextListener: Inizializza il DBManager e esegue la schedulazione del DBChecker.

### **Servlets**

- AccountServlet : gestisce il pattern /User/AccController
- UserServlet : gestisce il pattern /User/UserController
- UploadServlet : gestisce il pattern /User/Upload
- GuestServlet : gestisce i pattern GuestController , LogoutController
- AdminServlet : gestisce il pattern AdminController
- GeneralServlet : gestisce il pattern /General/GeneralController