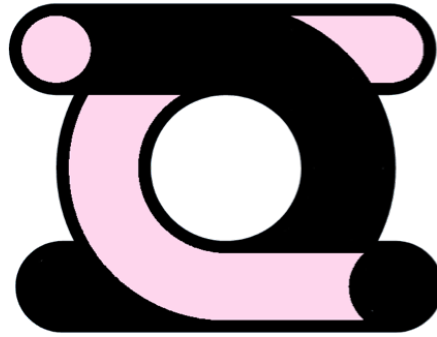


Università degli Studi di Trento

Introduzione alla Programmazione per il web



Tum4World

Pagina di descrizione

Gruppo 19

Anno accademico 2022/23

1 Utilizzo del pattern MVC

L'intera progettazione e strutturazione del sito web è stata fatta seguendo il pattern MVC, ovvero cercando di separare le parti di presentazione, business logic e rappresentazione dei dati trattati. Di seguito, si riporta una breve descrizione di ogni elemento software appartenente a Model, View e Controller.

- **Model**

La parte Model del progetto consiste nelle classi *Donazione* e *User* che non sono altro che dei BEAN (memorizzati nella cartella *model*). Nel dettaglio:

- *Donazione*, per gestire le donazioni effettuate dagli aderenti all'associazione; quando è necessario ottenere delle donazioni dal database viene istanziato (per ogni donazione ottenuta) un oggetto di questa classe, impostando i vari attributi.
- *User*, per gestire gli utenti registrati al sito. Anche in questo caso tutti i vari attributi degli utenti selezionati dal database sono messi in questo BEAN per renderli più semplicemente utilizzabili dalle parti dedicate alla presentazione.

- **View**

La parte View del progetto comprende file JSP, CSS e JavaScript che definiscono l'aspetto e l'interfaccia utente del sito. In particolare:

- i file JSP creano la struttura della presentazione e le singole pagine del nostro sito web. Il codice Java, per rispettare il pattern MVC, è ridotto al minimo: vengono utilizzati solamente dei blocchi di codice Java per effettuare scelte dal punto di vista grafico e della presentazione (infatti in *intestazione.jsp* viene verificata la validità o meno della sessione dell'utente per decidere quali parti del menu' di navigazione mostrare oppure per decidere lo stile della pagina). Inoltre si utilizza la funzione *encodeUrl* di *HttpServletResponse* per l'URL rewriting;
- i file CSS sono usati per lo stile della presentazione. Sono organizzati nella cartella *css* e sono divisi per tipologia di utente;
- infine i vari file JavaScript che animano la presentazione lato client sono organizzati nella cartella *scripts*.

- **Controller**

La parte Controller del progetto rappresenta la business logic dell'intero progetto.

Abbiamo innanzitutto un POJO, la classe *Counter* che rappresenta un contatore di hits. Si è scelto di utilizzare dei contatori separati per ogni pagina del sito (quindi tante istanze di *Counter*, una per pagina) al fine di aumentare la modularità del codice e rispettare i principi della OOP. I vari contatori vengono poi salvati nel *ServletContext* della nostra applicazione. Abbiamo poi varie servlet, responsabili della gestione delle richieste degli utenti provenienti dalla View e della coordinazione delle interazioni tra il Model e la View. In particolare, le servlet responsabili dell'interazione con il database sono inserite nella cartella denominata *db* e creano una connessione con il database ogni volta che vengono create (nel metodo *init*) e la distruggono alla loro uscita (nel metodo *destroy*).

Molto importante è la servlet *Index* che rappresenta l'entry point nel nostro sito web e che recupera dal database i valori dei contatori, per poi serializzarli (salvandoli nel database) quando viene distrutta.

La parte Controller comprende infine anche i filtri (nella cartella *filters*), necessari per proteggere l'accesso a pagine che necessitano che l'utente abbia eseguito il login e che la sessione sia ancora attiva. Molto importante è il filtro *SessionFilter* che gestisce il timeout della sessione (visto che a causa delle richieste AJAX per recuperare le varie frasi ispiranti la sessione non verrebbe mai invalidata). In questo modo seguiamo anche il paradigma AOP.

2 Parti non sviluppate

Il progetto è stato sviluppato per intero, seguendo con metodo rigoroso le specifiche fornite nella consegna. Pertanto, non sono presenti parti del progetto non sviluppate.

3 Contributo dei membri del gruppo

Ciascun membro del gruppo ha contribuito allo sviluppo sia della parte di back-end che di front-end.

Gabriele Volani si è occupato maggiormente della parte di back-end, dedicandosi alla gestione del database, delle sessioni, dei filtri e del conteggio dell'hitCounter. Federico Menegoz si è concentrato principalmente sulla parte di front-end, sui fogli di stile e sul design della View. Giulio Lo Cigno si è occupato maggiormente del front-end delle pagine private e dei grafici della pagina dell'amministratore. Enrico Cescato si è dedicato principalmente al front-end delle pagine di descrizione dell'associazione e alla procedura di testing.

Dopo aver progettato le rispettive parti individuali, abbiamo adottato una metodologia pseudo-agile per affrontare i diversi bug che sono emersi. Durante questa fase, ogni membro del gruppo ha contribuito allo sviluppo su tutte le aree del progetto.