

## Documento di descrizione del software

Il sistema fa un ampio utilizzo del pattern Model, View, Controller. Vediamo le funzionalità del sito che sfruttano questo pattern:

- **phraseFetcher.java**: la servlet che si occupa di gestire le frasi da mostrare nell'header è un primo esempio di MVC. La pagina che desidera ricevere una nuova frase da mostrare invia una richiesta al server. Qui la servlet (controller) andrà a recuperare dal database le frasi (model), ne sceglierà una in maniera casuale e la invierà come risposta. Una volta ricevuta la risposta, la frase verrà mostrata nell'header della pagina (**header.jsp**) tramite la funzione **updatePhrase** contenuta nel file **headerScript.js** (view).
- **servletLogin.java**: la procedura di login è personalizzata rispetto al tipo di utente. Una volta che l'utente inserisce le sue credenziali, la richiesta viene inviata alla servlet (controller) che si occupa di confrontare le informazioni ricevute con i dati salvati nel database. Nel caso di un match, si verifica il tipo di utente (model) e in base a quel campo si viene reindirizzati alla pagina apposita, una tra **AreaPersonaleAdmin.jsp**, **AreaPersonaleSim.jsp** e **AreaPersonaleAderente.jsp** (view).
- **paginaPrivataServlet.java**: la pagina privata di simpatizzanti e aderenti mostra agli utenti le informazioni relative al loro account. Per questo motivo, la servlet è stata suddivisa in due sottoclassi, ognuna delle quali gestisce la pagina privata a seconda del tipo di utente (**AreaPersonaleAdmin.jsp**, **AreaPersonaleSim.jsp** e **AreaPersonaleAderente.jsp**), comportandosi come controller. Essa recupera i dati relativi alle iscrizioni (model) di quell'utente e li restituisce alla pagina per mostrarli. È possibile inoltre richiedere le informazioni personali (model) relative all'account, che con lo stesso processo vengono restituite e mostrate in una sezione apposita (view).
- **statisticsServlet.java**: anche la pagina privata dell'amministratore (**AreaPersonaleAdmin.jsp**) sfrutta il pattern MVC per mostrare le informazioni relative al sito. Nel momento in cui l'admin vuole visualizzare delle informazioni relative agli utenti iscritti (totali o per tipo), alle donazioni o alle visite tramite tabelle o grafici (view, implementati in **statistics.js** utilizzando la libreria **highcharts.js**), la richiesta viene intercettata dalla servlet (controller) che si occupa del recupero dei dati richiesti dal database (model). Questi vengono inviati alla pagina dell'admin (**AreaPersonaleAdmin.jsp**), la quale si occupa di mostrare i dati tramite AJAX (contenuto nello stesso file javascript **statistics.js**) in base al tipo di dato ricevuto (view).

Funzionalità non implementate/con bug: le funzionalità sono state implementate tutte, solo due note:

1. La finestra dell'alert del cookie potrebbe ripresentarsi nella pagina successiva nonostante il click accetta/rifiuta, questo bug può essere generato dall'asincronicità della richiesta ma non siamo riusciti a risolvere il problema, tuttavia se si attende circa un secondo dopo il click nella pagina successiva dovrebbe funzionare tutto.
2. Le funzionalità della pagina personale Simpatizzante e Aderente non funzionano quando l'utente rifiuta i cookie, non siamo riusciti ad implementare un modo per gestire questo problema, si consiglia di non usare le seguenti pagine dopo aver rifiutato i cookie (non permette persino il logout).

Gli studenti hanno contribuito nel seguente modo:

- Riccardo Fusiello: ha gestito il gruppo organizzando il lavoro e i vari recap con i membri, contatti, email, descrizione associazione, email, filtro visite, servlet di inizializzazione, problem solving, bug fixing/membro di riferimento nel caso di dubbi o problemi
- Denise Comincioli: logo, pagine attività con immagini, sign in (servlet+jsp+ajax+validazione), registrazione confermata, script di validazione, log out, CSS, interazione col DB
- Cesare Lever: login (servlet+jsp+ajax+validazione), pagina privata admin (servlet+beans+jsp+ajax+grafici), sessione, volantino, documentazione(descrizione progetto, info sull'avvio), footer, interazione col DB
- Silvanus Bordignon: header statico, frasi (servlet+jsp+ajax), cookie (servlet+jsp+ajax), documentazione (annex, descrizione progetto, info sull'avvio)