



1) Descrizione Pattern MVC

Model: La parte di Model del Progetto è stata sviluppata e raggruppata in una cartella omonima comprendente 15 classi JAVA. All'interno di essa si trova la classe astratta "GenericDAO" che definisce gli elementi base per la connessione al database: il path e le funzioni per aprire e chiudere la connessione. All'interno del database verranno gestite 7 tabelle: "Utente", "Attività", "Donazioni", "Frase", "Iscrizioni", "Messaggi", "Visite", ad ogni tabella corrispondono due classi. La prima classe (di tipo Serializable) ha lo stesso nome della tabella ed implementa il Bean corrispondente ad essa includendo come parametri le colonne della tabella, i rispettivi metodi get and set ed il costruttore di default. La seconda classe è del tipo "nomecolonnaDAO" ed estendendo il GenericDao si occupa di gestire tutte le interazioni che si avranno con il database riguardo a quello specifico oggetto. In questo modo un membro del controller per creare un oggetto, utilizza la prima classe per costruirlo e la seconda per salvarlo nel database. Vi sono poi due classi che estendono la classe Exception.

Files: tutti quelli presenti nella cartella "model".

View: La presentazione è gestita dai file jsp presenti nella cartella webapp, ogni url accessibile nel sito è mappato (grazie al web.xml) ad una pagina jsp, ognuna di esse contiene solitamente 6 riferimenti ad altri file: header.jsp, footer.html (footer_static.html) e base.js gestiscono header e footer comuni per tutte le pagine, un include condizionale per il cookieBanner.html, infine i propri fogli stile (in assets/styles) e i fogli di script (presenti in assets/scripts). Il file "skel.jsp" è lo schema base che mostra la base di implementazione di tutte le pagine. Vi sono poi nella stessa cartella dei file html che sono posti separatamente per essere utilizzati da più pagine jsp, oltre a quelli appena evidenziati vi sono infatti "SimpatizzanteAderente.html" e "backToTop.html". All'interno della cartella assets si possono trovare le immagini (assets/images) presenti nel sito, il file download.pdf (volantino) ed un suono.

Files: tutti quelli presenti nella cartella "webapp" tranne la directory WEB-INF.

Controller: La gestione di quale file jsp deve essere mostrato come risposta ad una richiesta http è gestita dal container, seguendo le istruzioni specificate nel web.xml. Si è deciso di non utilizzare una classe java per risorsa per evitare di avere un elevato numero di classi la cui unica funzione fosse di includere la pagina jsp. Sono presenti diverse classi Java che agiscono come controller in diverse occasioni e tre filtri. Il filtro "CookiesFilter" si occupa di verificare quando mostrare il CookieBanner "UserFilter" impedisce accessi non autorizzati, "VisiteFilter" aggiorna le visite effettuate alle varie pagine. Fanno parte del controller anche le varie servlet chiamate tramite javascript per interagire con il database.

Files: tutti quelli presenti nella cartella "control".

2) Divisione Ruoli Partecipanti

Bregola Matteo: Front-end e Back-end delle seguenti pagine: **ChiSiamo**, **Attività**, **Attività1/2/3**, **Simpatizzante e Aderente**, **LogOut**. Servlet **AderenteDonazione**, **cancellaiscrizione**, **IscriviAttività**, **SessionManager**, **TrovaAttività**, **VisualizzaDati**, **logOut**. Filtro ed Html dei **Cookie**. **Css** della pagina di **Contatti** e revisione di Admin. Documento **Annex**, **Testing**, **Descrizione**, **Database**. Impostazione struttura progetto iniziale e revisione finale.

Frizzera Matteo: Gestione **Model** (tutte le classi all'interno della cartella). **Filtro** contatore **visite**. Servlet **fraseRandom**, **visiteJSON**, **donazioniJSON**, **popolaDB**. Pagine **login** (front-end, back-end, **elaboraLogin**), **signup** (frontend, back-end, **elaboraSignup**).

Libanora Riccardo: **Header:** gestione responsiveness con menu a tendina. **Footer:** stile, helper pappagallos **citazioni** dal **database**, click sul pappagallos, opzione di footer dinamico oppure statico. Creazione dei file-template **skel.jsp**, **base.css** e **base.js**, base per tutte le **pagine del progetto**. **Grafiche:** palette, loghi e sfondi per 4 potenziali temi. Funzione per **switchare** tra i temi. **Filtro utenti**. Pagina **homepage** (front-end,back-end). **Revisione generale frontend**.

Pontalti Matteo: **Contatti**. Gestione **mail** (servlet EmailSending, EmailUtility). Pagina **amministratore**(front-end, backend, resetContattori e quelle di visualizzazione tranne VisualizzaDati).