

# Fondamenti di Programmazione Web

## Progetto Finale

Melis Michele, matricola n°65798

## Pagine

- Suddivisione delle pagine in:
  - ◆ utente: pagine contenute nella directory di default
  - ◆ amministratore: pagine contenute nella directory "Amministratore"
- header.jsp contiene l'header che verrà visualizzato da tutti gli utenti. Le pagine dell'amministratore, invece, non lo visualizzeranno in quanto superfluo
- La pagina principale viene popolata con le sessioni disponibili da una servlet. Ogni sessione viene rappresentata come form, contenente le informazioni della sessione (utili all'utente) e un elemento nascosto contenente l'id della sessione, che viene utilizzato per identificare quest'ultima e successivamente mostrare la pagina con le rispettive informazioni
- Tutti i file .jsp il quale nome contiene "JSON" vengono utilizzati per risposte da parte delle servlet a delle richieste AJAX
- error.jsp si occupa di mostrare sul browser il messaggio di errore che è stato generato da una servlet e la sua possibile causa
- Viene utilizzata la medesima pagina sia per la registrazione sia per la modifica delle informazioni di un utente, in quanto possiedono gli stessi medesimi campi. L'unica differenza è che in "modalità modifica" non è possibile modificare né username né codice fiscale
- La pagina dei messaggi (sia dell'utente, sia dell'amministratore) sono divise in due div dove: uno presenta il form per l'invio di un messaggio e l'altro invece mostra i messaggi ricevuti
- I possibili destinatari di un messaggio da parte dell'amministratore sono:
  - ◆ Utenti disponibili: utenti che possono effettuare delle donazioni (ossia possiedono i requisiti per donare)
  - ◆ Utente singolo: deve essere specificato tramite username
  - ◆ Utenti prenotati in tale data: reminder
- Sia utente che amministratore possono effettuare un logout dalla propria area personale (o area amministratore)
- La pagina che mostra i donatori (utenti) accessibile dall'amministratore oltre alla specifica relativa all'ordinamento degli utenti, presenta anche la funzione per la ricerca di un utente presente nel database e visualizzarne le informazioni, nonché le sue donazioni effettuate
- Pagine come "login.jsp" non vengono generate da una servlet, poiché non hanno bisogno di pre-caricare nessun dato
- Ogni file jsp controlla se lo user ha effettuato l'accesso come utente o come amministratore, in modo da impedire all'utente di visualizzare le pagine come amministratore e all'amministratore di visualizzare le pagine come utente. È stato fatto tramite la libreria JSTL

## Stile

- I fogli di stile permettono di visualizzare correttamente le pagine fino ad un minimo di 480 pixel di larghezza
- base.css contiene lo stile della maggior parte degli elementi presenti nei file jsp. Lo stile di alcuni elementi è stato definito nell'intestazione quando gli elementi in questione presentano delle regole uniche, non adottate da altri elementi presenti in altre pagine, per suddividere meglio il tutto e renderlo più comprensibile (es.: Amministratore/listaUtenti.jsp). Alcuni elementi, invece, aventi regole di stile uniche e con 1 / 2 proprietà, presentano le regole direttamente nel tag (es.: textarea di Amministratore/messaggi.jsp)
- Ogni pagina ha uno stile molto semplice e pensato per rendere le interazioni il più intuitivi possibile

## Script

- Gli script utilizzati sono stati suddivisi in più file in modo da tenere il codice più ordinato
- Nella pagina listaUtenti.jsp, accessibile come amministratore, viene utilizzato AJAX per effettuare una richiesta alla servlet che restituirà i dati delle donazioni effettuate dall'utente cercato. L'idea principale era quella di rispondere con un array di oggetti avente ciascun dato assegnato a ciascun attributo, ma non riuscendoci ho optato per una soluzione più semplice:
  - ◆ I dati relativi alla sessione vengono uniti in un'unica stringa ponendo "/\*-\*/" come divisore;
  - ◆ I dati vengono inoltrati tramite listaDonazioniJSON.jsp;
  - ◆ AJAX riceve i dati e rimuove il divisore "/\*-\*/" dalla stringa, dividendola in più stringhe (pari al numero di informazioni)
- Nelle pagine in cui è previsto un feedback sul corretto invio e ricezione dei dati, viene utilizzato confermaJSON.jsp per la trasmissione della risposta. Quest'ultima può avere due valori:
  - ◆ positivo: l'operazione è avvenuta con successo
  - ◆ negativo: messaggio di errore
- Ogni pagina che può fare una richiesta con AJAX prevede anche a segnalare qualsiasi tipo di errore tramite un alert box, in modo tale da sapere se è andato tutto a buon fine oppure si è verificato un qualche tipo di errore (es.: pagina dei messaggi)
- script.js contiene il codice utile a controllare che la pagina registrazione.jsp (in fase di registrazione) presenti tutti i dati inseriti in modo corretto dall'utente, andando a segnalare a quest'ultimo qualsiasi campo non soddisfi i requisiti richiesti. Questi controlli vengono comunque effettuati anche tramite una servlet (RegistrazioneServlet.java)

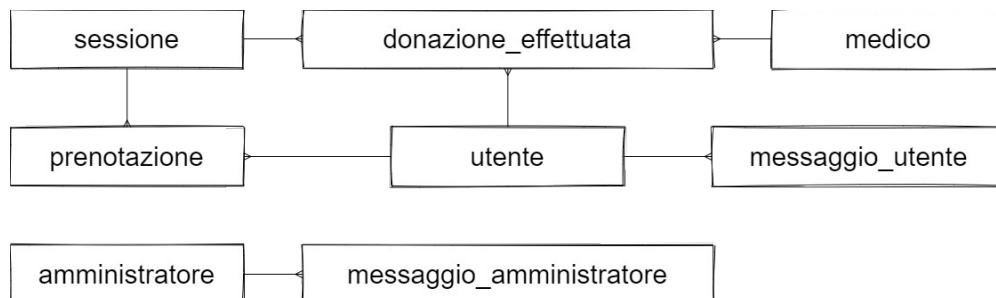
# Servlet

- Le servlet sono suddivise per user:
    - ◆ Utente:
      - AreaPersonaleServlet
      - ContattiServlet
      - ControllaNomeECodiceServlet\*
      - CronologiaUtenteServlet
      - IndexServlet
      - LoginServlet
      - LogoutServlet
      - MessaggioUtenteServlet\*
      - ModificaServlet
      - PrenotazioneServlet
      - RegistrazioneServlet
      - SessioneServlet
    - ◆ Amministratore:
      - AreaAmministratoreServlet
      - DonazioniEffettuateServlet
      - InviaMessaggioAmministratoreServlet\*
      - ListaUtentiServlet
      - LoginAmministratoreServlet
      - LogoutAmministratoreServlet
      - MediciAmministratoreServlet
      - MessaggiAmministratoreServlet
      - NuovaSessioneServlet\*
      - NuovoMedicoServlet\*
      - RichiestaDonazioniServlet\*
      - SalvaPrenotazioneServlet
      - SessioniAmministratoreServlet
- (\* servlet per gestione richieste AJAX)

- All'interno di tutte le servlet viene eseguito un controllo sull'esistenza della sessione: se esiste la sessione si procede normalmente, altrimenti si viene reindirizzati alla pagina principale del sito (o a quella di login)
- Vengono utilizzate delle servlet per il reindirizzamento ad una pagina ogniqualvolta quest'ultima abbia bisogno di informazioni dal database (ad esempio la pagina principale "index" utilizza una servlet per caricare e mostrare al suo interno le sessioni disponibili a cui l'utente può prenotarsi). Tutte le pagine dell'amministratore vengono visualizzate in questo modo poiché hanno tutte qualcosa da caricare a priori (sessioni, utenti, medici, messaggi etc.)
- Per recuperare l'ora a cui vengono spediti i messaggi vengono usate le classi "DateTimeFormatter" e "LocalDateTime"

## Database

- Entità e relazioni:



- Tutti i modelli sono stati rappresentati come classi e ciascuno ha il proprio factory per comunicare con il database
- Le date vengono salvate come stringhe poiché queste ultime vengono utilizzate e visualizzate molto più spesso come tali

*Il progetto è stato sviluppato usando come base il sito fatto a laboratorio*

*Credenziali di accesso come utente:*

**Username:** PastiGino **Password:** 12345

**Username:** Mirketto96 **Password:** 12345

**Username:** MichiLela **Password:** 12345

**Username:** Mirkk **Password:** 12345

*Credenziali di accesso come amministratore:*

**Username:** Melis **Password:** 65798