

Relazione progetto finale FPW

Osama Bounabri 60/61/65849

Credenziali amministratore: username: bounabri, password: 65849

Credenziali utenti semplici:

username	password
susy84	susanna84
sara01	sarav01
nicolar101	nicolarossi101
romina85	rominapilia85
lucamenta89	lucaluca89

DATABASE

il database è strutturato in 6 tabelle:

tabella: utente / prenotazione / fattura / slot / bagnino / messaggio

chiave primaria: username / id / id / id / nome / nessuno

chiavi esterne: la fattura, la prenotazione e il messaggio sono in relazione con l'utente attraverso un vincolo di chiave esterna che rappresenta l'username dell'utente a cui è stata fatta la fattura, che ha prenotato o che ha inviato il messaggio. La prenotazione inoltre presenta una relazione con lo slot che è stato prenotato attraverso un vincolo di chiave esterna che si riferisce all'id dello slot prenotato

FUNZIONALITA'

Per gestire i due tipi di accesso, mi sono servito di un booleano nella tabella utente del database che mi indicasse se l'utente fosse un amministratore (true) o no (false) e in base a ciò, dopo il login che rimanda a una servlet, quest'ultima mi rimanda all'area personale dell'utente o alla home dell'amministratore se l'utente ha passato tutti i controlli sugli input di login, settando quindi una sessione che dura 30min.

Nella home del sit, ho gestito il fatto di non poter prenotare, se non loggati, rimandando alla pagina di login se non fosse presente una sessione e se non fosse valido l'attributo di sessione che rappresenta l'username.

La registrazione è possibile attraverso una pagina jsp che rimanda a una servlet che prende i dati di registrazione inseriti e se validi (attraverso i controlli gestiti dalla classe Utils, con il controllo sull'uguaglianza delle password con il metodo equals(), e con il controllo sul fatto che i restanti input non fossero nulli come la scelta della fattura) mi inserisce i dati nella tabella utente del database settando a false il booleano che indica sia un amministratore.

Attraverso i metodi della classe Utils, gestisco i controlli sugli input, come lunghezze, controllo della sufficienza di posti nello slot da prenotare rispetto a quelli che si vuole prenotare, conversioni di date e orari e controllo su range di grandezza dei numeri.

Per la prenotazione degli slot, ho visualizzato gli slot nella home prelevandoli come una lista dal database, attraverso i css gli ho racchiusi ognuno in una sottospecie di box e ho inserito un menu dove l'utente indica quale slot vuole prenotare (inserendo l'id dello slot che vede nelle info di ogni slot) e quanti posti vuole prenotare. Inviando alla fine i dati con un tasto, i dati arrivano alla servlet, vengono controllati per vedere se validi e viene generata una prenotazione nel database con tutti i dati. Se lo stesso utente prenota lo stesso slot, vengono solo aggiornati il numero dei posti della prenotazione fatta in precedenza dallo stesso utente nella stessa data e orario. Le prenotazioni vengono visualizzate dall'amministratore quando vuole compilare le fatture.

La pagina personale dell'utente è realizzata controllando sempre prima che la sessione sia valida attraverso un controllo nella jsp e nella servlet (a cui rimanda la servlet di login) e poi stampando tutte le informazioni dell'utente prelevandole dal database in base all'username e la password passate in login. Contiene anche una barra di navigazione tra le pagine che nella parte del sito dove è sempre uguale è realizzata attraverso una jsp unica che poi cambia in base ai controlli su che pagina siamo.

Per visualizzare le fatture, ho creato una lista delle fatture solo dell'utente, prelevando, dalla lista di tutte le fatture (prese dal database dalla funzione nella factory delle fatture), solo quelle che hanno nel campo relativo all'id del cliente, l'username dell'utente della sessione corrente. Un controllo nella jsp verifica se la scelta della fattura è "si" e le visualizza, altrimenti mostra solo un messaggio che informa il cliente sulla sua non volontà di ricevere fatture.

Per la sezione messaggistica, ho creato una pagina con un'area di testo dove poter scrivere il messaggio e attraverso una servlet salvarlo nel database insieme all'autore del messaggio. In una pagina dedicata all'amministratore, stampo poi la lista di tutti i messaggi prelevati dal database.

L'amministratore ha una pagina a cui attraverso un box può inserire un nuovo slot inserendo tutti i dati che, con un apposito tasto di invio, vanno in una servlet che ne gestisce l'inserimento nel database nella tabella degli slot. I bagnini vengono prelevati dal database e inseriti come una lista di opzioni in una select, cosicché l'amministratore possa selezionare solo quelli inseriti nel database. Se uno slot è già presente, con un controllo lato server sull'uguaglianza di data e orario, vengono solo aggiornati i posti dello slot che ha queste informazioni uguali.

Per la compilazione delle fatture, ho visualizzato in una pagina tutte le prenotazioni con relativo id prelevandole dal database e ho inserito un menu in cui l'amministratore può inserire l'id della prenotazione che vuole fatturare. La fattura ha una compilazione automatica attraverso una funzione generaFattura() della servlet FatturaFactory che viene poi richiamata nella servlet in cui viene passato l'id della prenotazione. La funzione prende una prenotazione e, in base alla scelta della fattura da parte dell'utente che ha fatto quella prenotazione, genera una fattura e la inserisce nel database con tutti i dati e una descrizione che sarà sempre uguale per tutte le fatture, tranne per i dati relativi alla data e all'orario prenotati, se la scelta è "si", altrimenti non la genera e in ogni caso, la prenotazione viene eliminata dal database con una funzione nella eliminaPrenotazione(). L'id della prenotazione da fatturare passato dall'amministratore viene controllato che sia valido non potendo superare il totale delle prenotazioni effettuate e restituisca una prenotazione esistente controllando tutte le prenotazioni del database.

Per aggiungere nuovi bagnini, l'amministratore ha a disposizione una pagina creata sempre con le jsp in cui ha un box dove può inserire i dati del bagnino e inviarli con un tasto alla servlet che si occuperà di fare i controlli sui dati passati e inserire il bagnino nel database.

Per la visualizzazione di tutti gli utenti, ho creato una pagina jsp in cui ho inserito un menu di ordinamento con la scelta a tendina e a seguire la lista di tutti gli utenti visualizzati ognuno con le sue informazioni in piccoli box gestito da jsp e css. Con un controllo nella jsp sull'uguaglianza del cliente in una lista di prenotazioni e l'username dell'utente con i tag speciali della libreria jstl, stampo per ogni utente oltre alle sue informazioni, la lista di prenotazioni non processate dall'amministratore. Se l'amministratore decide di ordinare gli utenti secondo il cognome, rimando alla servlet che si occupa di ricavare nuovamente la lista di utenti dal database, con un controllo su di esso nella query con cui faccio l'interrogazione al database, ordinata in base alla scelta dell'amministratore. Gli utenti ricavati saranno sempre quelli con il booleano settato a false (cioè utenti semplici). Per ordinarli invece in base ai posti, ho utilizzato un'interrogazione al database unendo la tabella degli utenti e delle prenotazioni e selezionando solo i dati dell'utente raggruppati per username e ordinati in base alla somma dei posti delle loro prenotazioni (in base alla scelta dell'utente), salvando il tutto in una lista (che avrà tutti gli utenti che hanno almeno una prenotazione non processata) che confronto con la lista di tutti gli utenti e se l'utente non è presente nella lista degli utenti con prenotazioni, lo aggiungo, altrimenti no. Così avrò prima gli utenti con prenotazioni ordinati e poi tutti quelli senza.

Per la ricerca dell'utente in base al nome e cognome ho creato una pagina jsp con un campo testuale che sotto ha nascosta (attraverso css) la lista di tutti i nomi e cognomi. Questa lista la ricavo in un file javascript e con dei controlli creo una lista con solo i nomi e cognomi che iniziano con le lettere inserite dall'amministratore che stampo sotto il campo di ricerca e, sempre nel file javascript, se seleziono un elemento di questa nuova lista, l'input di ricerca prenderà il suo valore. Infine il nome e cognome selezionato o inserito viene mandato a una servlet che controlla per ogni utente nel database (attraverso una funzione nella UtenteFactory) che il nome e cognome sia contenuto nella stringa ricercata dall'amministratore e in caso positivo mi ridà l'utente che setto in una variabile di sessione. Setto in una variabile di sessione anche la lista di tutte le sue fatture controllando che per tutte le fatture ricavate dal database, l'username coincida con quello dell'utente. Così posso riprendere questi due elementi nella jsp e stampare le informazioni dell'utente e le sue fatture.

NOTE:

Per ogni operazione, sia dell'utente che dell'amministratore, l'utente viene informato della buona o cattiva riuscita con una pagina jsp di successo che prende un messaggio diverso settato in ogni specifica servlet, oppure queste ultime lanciano un'eccezione con un messaggio su cosa sia andato storto.

L'amministratore è stato inserito manualmente direttamente nel database

La visualizzazione di elementi presenti in tabelle del database rimanda sempre a delle servlet che si occupano di riempire delle liste con questi elementi se questa (con un controllo con tag speciali nella jsp) risulta vuota,

