

GESTIONE DI UN LIDO

PROGETTO INDIVIDUALE

Antonio Guarisco (0705419)

DESCRIZIONE GENERALE

Il sistema mostrerà una schermata principale in cui sarà possibile registrarsi o effettuare il login. Gli utenti si potranno registrare mentre gli impiegati dovranno essere aggiunti manualmente dall'Admin. Fatta l'autenticazione con opportuni controlli e interrogazioni al Database, l'utente loggato sarà reindirizzato alla sua pagina personale basata sul "Tipo" che gli è stato assegnato in fase di registrazione / inserimento da parte dell'amministratore. L'utente potrà effettuare azioni basate sul proprio privilegio. Si identificano 4 tipologie di utenti: client, cashier, lifeguard e admin. L'utente client potrà prenotare un "Seat", liberarlo, ordinare qualcosa da mangiare, tenere traccia dello stato dei suoi ordini e aggiornare le credenziali di accesso. Il "Lifeguard" avrà a disposizione una schermata interattiva per monitorare lo stato delle Seats e visualizzarne le informazioni. Il "Cashier" potrà visualizzare tutti gli acquisti che gli utenti effettuano e visualizzare il flusso di cassa totale. Il "Cook" tramite una schermata interattiva visualizzerà gli ordini non completati e tramite pulsante potrà cambiare lo stato dell'ordine in "completato". L'Admin potrà aggiungere i nuovi dipendenti, cancellare gli utenti e visualizzare tutti gli utenti registrati

ARCHITETTURA DEL SISTEMA

Tecnologie utilizzate:

1. classi Java servlet, per l'accettazione di richieste http, per il processo di elaborazione dei dati inseriti dal client e per il forwarding a pagine JSP;
2. JSP per la rappresentazione dei risultati elaborati dalle servlet;
3. Java Beans per la creazione, lo storage degli Utenti, delle Sdraio e degli Ordini e per la condivisione di quest'ultimi tra le varie classi della web application;
4. HTML, CSS, JavaScript e jQuery per la gestione dell'interfaccia utente;
5. AJAX per la realizzazione di schermate interattive;
6. Apache Tomcat, quale middleware per il deployment della web application.
7. MySQL come DBMS, InnoDB per la memorizzazione dei dati.

REQUISITI FUNZIONALI

Generali:

1. Dovranno essere controllati tutti i casi d'accesso;
2. Dovranno essere controllati e gestiti i possibili errori;
3. I dati inseriti dall'utente dovranno essere verificati;
4. Gli schermi interattivi saranno ricaricati periodicamente ed in maniera asincrona;
5. Gli errori tecnici dovuti all'interrogazione al Database annulleranno le azioni e restituiranno una schermata di errore;

Utente "client":

1. Sarà possibile registrarsi tramite form;
2. Sarà possibile effettuare il login;

3. Saranno visibili le proprie informazioni;
4. Si potrà aggiornare la password di accesso;
5. Sarà possibile prenotare un "seat";
6. Sarà possibile liberare un "seat" ed uscire;
7. Potrà ordinare da mangiare / bere;
8. Potrà visualizzare i propri acquisti e il loro stato.

Utente "Lifeguard":

1. Potrà effettuare il login;
2. Potrà visualizzare in maniera interattiva lo stato delle "seat" dell'impianto;
3. Potrà visualizzare le informazioni inerenti ad un "seat".

Utente "Cashier":

1. Potrà effettuare il login;
2. Potrà visualizzare lo stato degli "orders" e il flusso di cassa totale.

Utente "Cook":

1. Potrà effettuare il login;
2. Potrà visualizzare in maniera interattiva lo stato degli "orders";
3. Potrà modificare lo stato degli ordini.

Utente "Admin":

1. Potrà effettuare il login;
2. Potrà aggiungere i nuovi dipendenti;
3. Potrà eliminare gli utenti dal sistema;
4. Potrà visualizzare tutti gli utenti registrati.

Server side:

1. Suddivisione in package: Servlets, Model, Database;
2. Elaborazione dei dati forniti dal client attraverso l'uso di classi java servlet;
3. Gestione degli errori e di eccezioni;
4. Metodi di utilità per interrogare il DB;
5. Configurazione di un Data Source per gestire le connessioni;
6. Uso del PreparedStatement per la sicurezza;
7. Autenticazione tramite Login;
8. Java Beans per la creazione e la condivisione di oggetti tra le classi;
9. JSP per la visualizzazione dei dati elaborati;
10. Implementazione del pattern MVC;

MODELLO DEI DATI E ALTRE SPECIFICHE

Xampp (Apache, MySQL, Apache Tomcat v9.0);

Il sistema è stato reso il più semplice possibile per velocizzare l'implementazione:

user(ID, username, password, type);

orders(ID, idclient, description, quantity, price, status);

seat(ID, idclient, posX, posY, available);

orders: Status rappresenta lo stato dell'ordine ("Incomplete", "Complete"), sarà il Cook ad aggiornare lo stato di un ordine;

user: ogni Utente avrà un Type ("client" nel caso sia un cliente semplice, "admin", "cook", "lifeguard", "cashier" nel caso si tratti di impiegati con mansioni specifiche);

seat: posX, posY rappresentano le "coordinate" della sdraio specifica; Available (se la sdraio è libera allora nel DB il valore sarà pari ad **1**, al contrario sarà uguale a **0**).