

Relazione del Progetto di Basi di Dati

Gestione degli Impianti acquatici

*Kevin Mansoldo VR368403,
Matteo Dal Monte VR373431,
Luca Vicentini VR367617.*

Indice

1	Schema Concettuale	3
2	Schema Logico	4
3	Considerazioni Varie	5
4	Dettagli Implementativi del Sito	6
4.1	Login e Sessioni	6
4.2	Pagine	7
4.3	Struttura sito in pseudocodice	8
4.4	Note	14

1 Schema Concettuale

Lo schema concettuale...

2 Schema Logico

Persona (Matricola, Nome, Cognome, DataNascita, Password, LavoraDa, LavoraA, NomeImpianto, DataApertura, DataChiusura);
Impianto (Nome, DataApertura, DataChiusura, Comune, Indirizzo);
Piscina (Codice, DataApertura, DataChiusura, OraApertura, OraChiusura, Tipo, NomeImpianto, DataAperturaImpianto, DataChiusuraImpianto);
Cliente (Codice, Nome, Cognome, TipoDocumento, NumeroDocumento);
Biglietto (Codice, Importo, DataInizio*, DataFine*, IngressiTotali*, IngressiRimasti*, CodiceCliente);
Giorno (Data, NomeImpianto, DataApertura, DataChiusura, OraApertura, OraChiusura, NumeroIngressi);
Iscritto (CodiceCliente, NomeImpianto, DataAperturaImpianto, DataChiusuraImpianto);
Turno (MatricolaPersona, NomeImpianto, DataAperturaImpianto, DataChiusuraImpianto, Data, OraEntrata, OraUscita);
Ingresso (CodiceBiglietto, NomeImpianto, DataAperturaImpianto, DataChiusuraImpianto, DataEntrata, OraEntrata, OraUscita);

Vincoli d'integrità referenziale:

Persona.(NomeImpianto, DataApertura, DataChiusura) → Impianto.
Piscina.(NomeImpianto, DataApertura, DataChiusura) → Impianto.
Biglietto.CodiceCliente → Cliente.
Iscritto.CodiceCliente → Cliente.
Iscritto.(NomeImpianto, DataApertura, DataChiusura) → Impianto.
Turno.(NomeImpianto, DataApertura, DataChiusura) → Impianto.
Ingresso.(NomeImpianto, DataApertura, DataChiusura) → Impianto.
Ingresso.CodiceBiglietto → Biglietto.

3 Considerazioni Varie

- Biglietto con gli attributi di quello singolo aveva 2 figli: abbonamento con attributi data validità e ingressi prepagati con ingressi totali (10 o 50) e ingressi rimasti
- Una persona può lavorare in un solo impianto
- Un cliente può essere iscritto a più impianti con la stessa tessera
- Un abbonamento, 10 ingressi e 50 ingressi, possono essere usati nel solo impianto in cui è stato creato
- L'abbonamento è valido fino alla fine dell'anno corrente
- 10/50 ingressi non ha data di scadenza
- Nel caso ci fossero abbonamenti utilizzabili non è possibile emettere un nuovo biglietto singolo
- Quando viene creato un nuovo abbonamento(10/50/abb) viene scalato un nuovo ingresso da quelli rimasti
- Una persona licenziata non può più accedere all'area riservata del sito
- La relazione turno è legata all'entità giorno piuttosto che all'entità impianto supponendo che ogni turno di lavoro avviene in un giorno di apertura dell'impianto (altrimenti si sarebbe dovuto ripetere la data nella relazione)
- Nell'entità Piscina la chiave è composta dagli attributi codice e data apertura e chiusura permettendo di aprire e chiudere la stessa piscina più volte in un anno
- L'entità Giorno ha come chiave esterna il nome dell'Impianto
- I codici di Cliente, Persona, Biglietto e Piscina sono autoincrementali

4 Dettagli Implementativi del Sito

4.1 Login e Sessioni

Una volta che un amministratore di un impianto fa il login viene associata una sessione al client relativo, nella quale viene mantenuto l'attributo username. L'attributo è una stringa contenente il relativo nome utente. In maniera trasparente viene memorizzato anche un attributo che identifica l'id della sessione, che sarà lo stesso che verrà allocato nella macchina client dell'utente tramite cookie. In questo modo dopo aver fatto un login, è possibile accedere a tutte le pagine private senza dover rinviare user e pass ad ogni richiesta HTTP. Ogni sessione dura 600 secondi(10 minuti), dopo di che essa verrà "eliminata" e se avviene un'altra richiesta da parte di un host a cui è scaduta la sessione, allora esso verrà reindirizzato alla HomePage con il relativo messaggio di sessione scaduta. Da HomePage è anche possibile effettuare il logout, in questo caso verrà eliminato l'attributo username dalla relativa sessione. L'utilizzo di SESSIONI implica che il browser del client debba avere l'accettazione di COOKIE attiva per poter navigare sul sito. La classe AuthFilter oltre a controllare il PATH di index lo fa anche per impianto, quindi queste 2 pagine possono essere visualizzate anche senza essere loggati.

4.2 Pagine

- Home.jsf: Visualizza la lista degli impianti già completi con le rispettive piscine e giorni di apertura e permette di loggarsi. Tramite la funzione `loginBean.getImpianti` riceve la lista degli impianti presenti nel sistema. Tutte le informazioni su ognuno di essi è rappresentato da un div che si rimpaginano in base alla dimensione dello schermo. All'interno di questi troviamo:
 - Immagini random;
 - Nome impianto;
 - Comune;
 - Indirizzo;
 - Tipologia delle piscine presenti.

In fondo alla pagina è presente il form di login che permette di inserire matricola e PW e accedere alla pagina di backend.

- Impianto.jsf: Riceve l'impianto selezionato nella homepage e visualizza le piscine con le date e gli orari di apertura e chiusura e gli ingressi totali giornalieri del mese corrente con la possibilità di filtrare i dati.
- Nuovo_cliente.jsf: Form che invia i dati di un nuovo cliente che viene iscritto all'impianto corrente oppure l'iscrizione di un cliente già esistente nel sistema in altri impianti.
- Lista_clienti.jsf: Riceve la lista dei clienti iscritti al sistema e i loro accessi.
- Nuovo_accesso.jsf: Permette di selezionare un cliente iscritto all'impianto corrente e cercare se sono disponibili abbonamenti o ingressi da 10/50 e utilizza un ingresso di questi, altrimenti viene creato un nuovo biglietto o abbonamento a scelta.

Riceve la lista dei clienti tramite la funzione `loginBean.clientiImpianto` e imposta il cliente selezionato tramite la funzione `loginBean.clienteSelNuovoAcc`. Vengono poi visualizzati gli ingressi o abbonamenti disponibili tramite la funzione `loginBean.ricercaIngressi`. Nel caso non ci fossero risultati è possibile creare un nuovo abbonamento o biglietto singolo o 10/50 ingressi. Per utilizzare un abbonamento si preme il tasto relativo e automaticamente viene scalato un ingresso. La funzione che si occupa di ciò è la funzione `loginBean.inserisciIngresso` che riceve il codice e il tipo di biglietto.

- Lista_accessi.jsf: Riceve la lista degli iscritti all'impianto corrente che hanno acceduto
- Chiusura_giornata.jsf: Permette di scegliere tra le persone che lavorano nell'impianto corrente quelle che hanno lavorato oggi come turno. Oppure chiude la giornata premendo sul bottone relativo e inserisce il totale degli ingressi di oggi nella tabella giorno nel campo del giorno di oggi del DB.

4.3 Struttura sito in pseudocodice

```
index.jsf(  
  menu(  
    link : #impianti(redirect to div)  
    link : #login(redirect to div)  
  )  
  impianti(  
    repeat[loginBean.getImpianti(), var i, <div>] (  
      i.getCitta()  
      i.getIndirizzo()  
      i.getTipi()  
      link : [loginBean.setImpiantoSel(i)] impianto.jsf  
    )  
  )  
  login(  
    form(  
      text - username : loginBean.nome()  
      text - password : loginBean.pwd()  
      link : LOGIN, loginBean.login() (  
        if(ok) redirect to area-riservata.jsf  
      )  
    )  
  )  
);
```



```

    impianto.jsf(
        menu(
            link : index.jsf
        )
        piscine_presenti(
            repeat[loginBean.getImpiantoSel(), var i, <table>] (
                repeat[i.getPiscine, var p, <tr>] (
                    tipologia : p.getTipoString()
                    data_apertura : p.getDataA()
                    data_chiusura : p.getDataC()
                    ora_apertura : p.getOraA()
                    ora_chiusura : p.getOraC()
                )
            )
        )
        ingressi_giornalieri(
            form(
                date - inizioPeriodo : loginBean.dataI()
                date - finePeriodo : loginBean.dataF()
                link : FILTRA-PERIODO, loginBean.cercaIngressi() (
                    repeat[loginBean.getImpiantoSel(), var i, <table>] (
                        repeat[i.getGiorni(), var g, <tr>] (
                            data : g.getData()
                            n - ingressi : g.getNingr()
                        )
                    )
                )
            )
        );

```

```
area - riservata.jsf(  
    menu(  
        link : lista - clienti.jsf  
        link : nuovo - cliente.jsf  
        link : lista - accessi.jsf  
        link : nuovo - accesso.jsf  
        link : chiusura - giornata.jsf  
        form(  
            logout : loginBean.logout() goto index.jsf  
        )  
    )  
);  
  
lista - clienti.jsf(  
    menu(  
        link : area - riservata.jsf  
        form(  
            link : LOGOUT, loginBean.logout() goto index.jsf  
        )  
    )  
    clienti - sistema(  
        repeat[loginBean.getClienti(), var c, <table>] (  
            nome : c.getNomeCliente()  
            cognome : c.getCognomeCliente()  
            data : c.getDataEntrata()  
            ora : c.getOraEntrata()  
            impianto : c.getImpianto()  
        )  
    )  
);
```

```
nuovo – cliente.jsf(  
    menu(  
        link : area – riservata.jsf  
        form(  
            link : LOGOUT, loginBean.logout()gotoindex.jsf  
        )  
    )  
    nuovocliente(  
        form(  
            text – nome : clienti.setNomeCliente()  
            text – cognome : clienti.setCognomeCliente()  
            text – tipoDocumento : cliente.setTipoDocumento()  
            text – numeroDocumento : cliente.setnumeroDocumento()  
            link : INVIA, clienti.insertUtente(loginBean.getImpianto()))(  
        )  
    )  
    iscriviti – cliente – esistente(  
        form(  
            cliente : items(loginBean.getClientiEsterni()), loginBean.setClienteSel()  
            link : INVIA, loginBean.iscriviCliente()  
        )  
    )  
);
```

```
lista ← accessi.jsf(  
  menu(  
    link : area ← riservata.jsf  
    form(  
      link : LOGOUT, loginBean.logout()gotoindex.jsf  
    )  
  )  
  listaaccessi(  
    form(  
      cliente : items(loginBean.getClientiAccessi(),  
                      loginBean.setClienteSelListaAcc())  
      datainizio : loginBean.setDataInizioAccessi()  
      datafine : loginBean.setDataFineAccessi()  
      link : INVIA, loginBean.filtraAccessi()  
    )  
    repeat[loginBean.getAccessi(), varc, < table >](  
      nome : c.getNomeCliente()  
      cognome : c.getCognomeCliente()  
      data : c.getDataEntrata()  
      ora : c.getOraEntrata()  
      impianto : c.getImpianto()  
    )  
  )  
);
```

```

nuovo - accesso.jsf(
    menu(
        link : area - riservata.jsf
        form(
            link : LOGOUT, loginBean.logout()gotoindex.jsf
        )
    )
    nuovo - accesso(
        form(
            cliente : items(loginBean.getClientiImpianto()),
                        loginBean.setClienteSelNuovoAcc()
            link : CERCA, loginBean.ricercaIngressi()
        )
        formhiddenIf : [loginBean.getClienteSelNuovoAcc() != 0
                        AND!loginBean.getAbbonamenti().isEmpty()](
            repeat[loginBean.getAbbonamenti(), varc, < table >](
                codbiglietto : loginBean.getCodBiglietto()
                tipobiglietto : loginBean.getTipoBigliettoString()
                inizioabbonamento : loginBean.getDataInizioAbbonamento()
                fineabbonamento : loginBean.getDataFineAbbonamento()
                ingressitotali : loginBean.getTotaleIngressi()
                ingressirimanenti : loginBean.getIngressiRimnenti()
                link : UTILIZZA,
                        loginBean.inserisciIngresso(c.getCodBiglietto, c.getTipoBiglietto)
            )
        )
        form(
            hiddenIf : [loginBean.getClienteSelNuovoAcc() != 0
                        ANDloginBean.getAbbonamenti().isEmpty()](
                link : BIGLIETTOSINGOLO, loginBean.nuovoBiglietto('s')
                link : ABBONAMENTO, loginBean.nuovoBiglietto('a')
                link : 10INGRESSI, loginBean.nuovoBiglietto('d')
                link : 50INGRESSI, loginBean.nuovoBiglietto('c')
            )
        )
    );

```

```
chiusura - giornata.jsf(  
    menu(  
        link : area - riservata.jsf  
        form(  
            link : LOGOUT, loginBean.logout()gotoindex.jsf  
        )  
    )  
    turnipersonaleoggi(  
        form(  
            checkbox(values : loginBean.getPersonale(),  
                    loginBean.setPersonaleOggi())  
            link : INVIA, loginBean.registraTurni()  
        )  
    )  
    chiudigiornata(  
        form(  
            link : CHIUDIGIORNATA, loginBean.chiudiGiornata()  
        )  
    )  
);
```

4.4 Note

- Ogni inserimento sul db viene notificato a schermo sia nel caso sia andato a buon fine che in caso di errore.
- Abbiamo scaricato un template gratis da un sito di Template e adattato al nostro caso.