

Basi della pubblicazione on-line (mod.2)

Applicazione per la gestione del calendario
didattico e delle aule per il corso di SMC

Progetto di

V. D. e Luca Imperatore

| | |
|--|----|
| Analisi dei requisiti | 3 |
| 1. Specifiche di progetto | 3 |
| 2. Operazioni su i dati e frequenza..... | 4 |
| Costruzione dello schema E-R..... | 4 |
| 1. Sviluppo delle entità principali | 4 |
| 2. Schema E-R finale | 6 |
| Diagramma di flusso | 7 |
| Creazione dei fogli di stile | 8 |
| Creazione del Template | 8 |
| Creazione di funzioni esterne..... | 8 |
| 1. Funzioni Javascript | 8 |
| 2. Funzioni PHP | 9 |
| Sviluppo del progetto..... | 10 |
| Problematiche riscontrate..... | 12 |
| Applicazione delle letture al progetto | 13 |
| Creazione dei Trigger..... | 14 |
| Esportazione di una tabella in XML | 14 |

Analisi dei requisiti

1. Specifiche di progetto

Si vuole realizzare un'applicazione web che permetta la gestione del calendario didattico e della disponibilità delle aule per un corso di studi universitario (nello specifico il corso di SMC). L'applicazione sarà impostata per essere utilizzata anche come normale agendario per eventi universitari o come agenda personale.

Un requisito richiesto è la possibilità di integrare l'applicazione alla piattaforma "Life" che rappresenta l'attuale ambiente di comunicazione e di e-learning del corso di SMC. Verranno per questo utilizzate le tabelle che utilizza la piattaforma esistente e gli stili e la grafica il più possibile simili alla stessa.

Il progetto prevede l'utilizzo da parte di due tipologie di utenti: gli amministratori e gli utenti generici. Al momento del rilascio sarà composto da una maschera di registrazione per permettere la registrazione dei nuovi utenti e una di login per l'accesso agli utenti registrati. In fase di integrazione con la piattaforma "Life" questa parte sarà eliminata in quanto l'accesso degli utenti è già gestita a livello più alto in quanto questo progetto rappresenta solamente un modulo da inserire all'interno di un macro sistema.

All'accesso dell'utente sarà visualizzata una tabella che rappresenta l'orario settimanale della didattica impostata per vedere tutti gli anni di corso contemporaneamente della settimana corrente (il giorno corrente sarà evidenziato da una colonna di colore diverso). Inoltre sarà presentato un menù di gestione differente a seconda che il soggetto sia un utente generico o un amministratore. Per il primo è previsto un link che permette l'esportazione della tabella del calendario didattico in un file xml, un menù a tendina (combobox) per la scelta dell'anno di corso e un tasto rappresentato da un calendario che attiva una popup per la scelta del giorno di interesse. Si potranno scorrere i giorni o i mesi sempre attraverso quest'ultima. Per il secondo, oltre al menù precedentemente descritto, sono previste tre scelte che permettono l'inserimento delle aule, dei moduli e del calendario didattico. Al momento dell'inserimento di una nuova aula o di un nuovo modulo viene controllato se questi sono già presenti nella base dati, in caso affermativo viene visualizzato un messaggio di errore. L'opzione di inserimento del calendario didattico consiste nello scegliere il corso, il docente, l'aula di appartenenza, il giorno, l'ora di inizio e fine lezione e il periodo di validità (generalmente un trimestre). Al momento dell'inserimento dei dati, alcune funzioni opportunamente realizzate controlleranno la disponibilità dell'aula (che in caso negativo segnalerà il messaggio) e la correttezza delle date inserite (sia la corrispondenza con il formato standard previsto sia il controllo che la data di fine corso sia maggiore di quella di inizio corso). Modifiche al calendario potranno essere effettuate cliccando direttamente sulla cella corrispondente al giorno/ora da modificare o eliminare.

2. Operazioni su i dati e frequenza

OPERAZIONE 1: Inserimento di un nuovo utente specificando nome, cognome, email, titolo, username e password (ogni volta che si autorizza l'accesso ad un nuovo utente).

OPERAZIONE 2: Inserimento di una nuova aula (1 volta l'anno).

OPERAZIONE 3: Inserimento di un nuovo modulo (1 volta l'anno).

OPERAZIONE 4: Inserimento del calendario didattico (4 volte l'anno).

OPERAZIONE 5: Modifica del calendario didattico (ogni volta che ci sono richieste di modifica orario o aula).

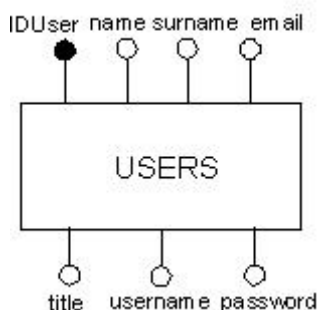
OPERAZIONE 6: Cancellazione del calendario didattico (ogni volta che ci sono richieste di cancellazione orario).

Costruzione dello schema E-R

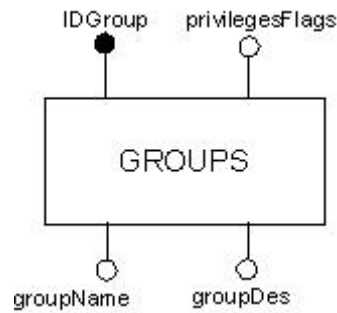
1. Sviluppo delle entità principali

Per rispettare il requisito di integrazione richiesto, le entità utilizzate nel progetto sono state esportate dalla piattaforma "Life" attualmente in esercizio. Le entità interessate allo sviluppo dell'applicazione sono:

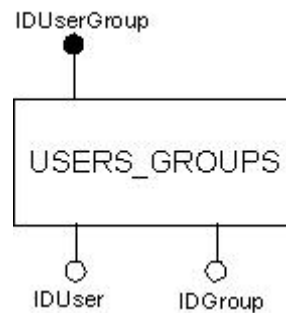
USERS – Rappresenta i dati degli utenti che accedono al sistema. E' composta dai seguenti attributi: IDUser (chiave primaria), name, surname, email, title, username e password.



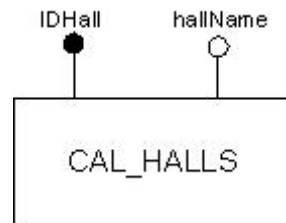
GROUPS – Rappresenta il gruppo di appartenenza dell'utente. E' composta dai seguenti attributi: IDGroup (chiave primaria), groupName, groupDes, privilegesFlags.



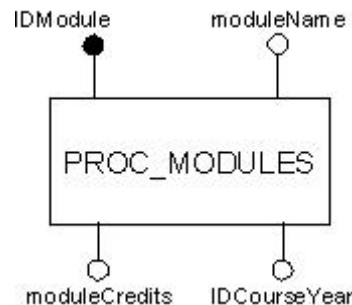
USERS_GROUPS – Rappresenta la relazione tra l’utente e il gruppo di appartenenza. E’ composta dai seguenti attributi: IDUserGroup (chiave primaria), IDUser, IDGroup.



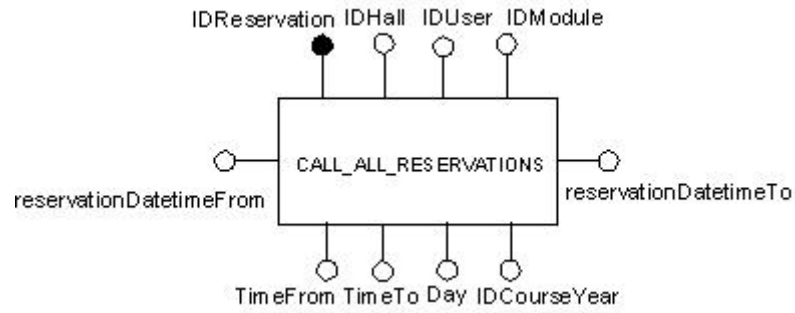
CAL_HALLS – Rappresenta le aule della facoltà. E’ composta dai seguenti attributi: IDHall (chiave primaria), hallName.



PROC_MODULES – Rappresenta i dati del modulo di insegnamento. E’ composta dai seguenti attributi: IDModule (chiave primaria), moduleName, moduleCredits, IDCourseYear.



CAL_ALL_RESERVATIONS – Rappresenta i dati del calendario didattico. E’ composta dai seguenti attributi: IDReservation (chiave primaria), IDHall, IDUser, IDModule, reservationDatetimeFrom, reservationDatetimeTo, IDCourseYear, Day, TimeFrom, TimeTo.



2. Schema E-R finale

Lo schema E-R finale si ottiene integrando le entità sviluppate fin'ora con opportune relazioni. Lo schema E-R finale risulta perciò essere il seguente:

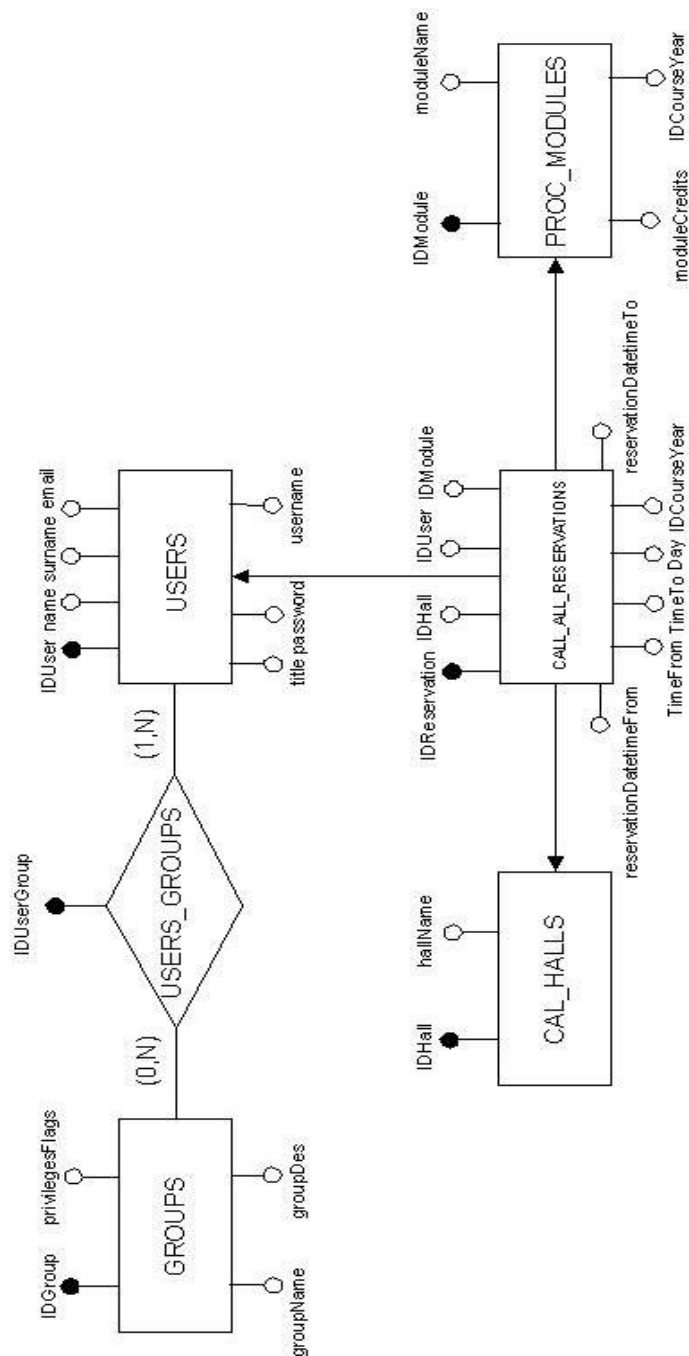
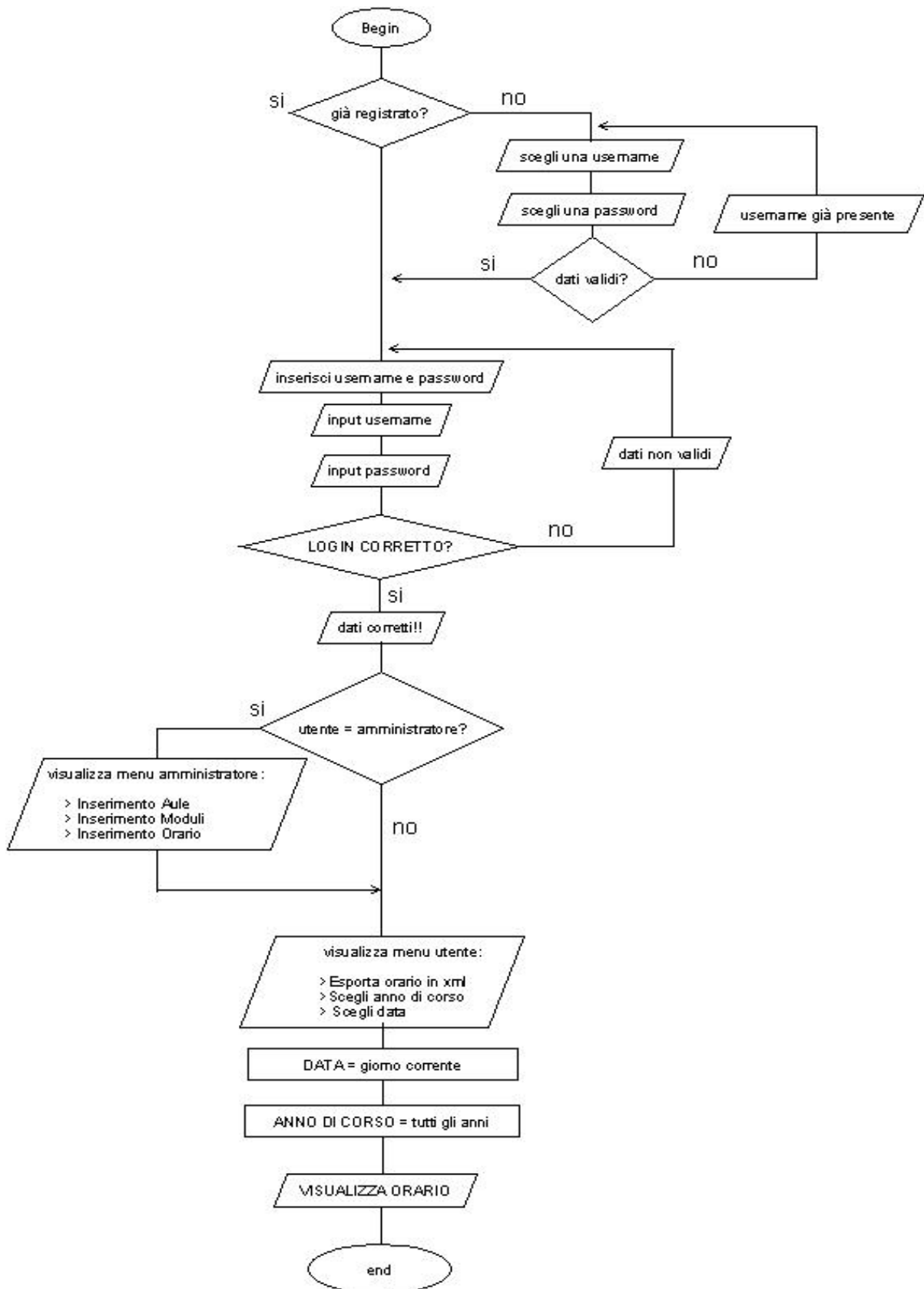


Diagramma di flusso

Il diagramma di flusso dell'applicazione è il seguente:



Creazione dei fogli di stile

Per rispettare lo standard di layout della piattaforma “Life” in cui il progetto dovrà integrarsi sono stati utilizzati i fogli di stile della piattaforma stessa. Per non modificare quelli originali, è stato inoltre aggiunto un foglio di stile personalizzato che viene utilizzato in alcune parti interne dell’applicazione. I fogli di stile utilizzati sono perciò i seguenti: life.css, life_cal.css, life_day.css, styles.css.

Creazione del Template

Alcune parti del layout dell’applicazione si ripetono per tutte le pagine. E’ stato perciò realizzato un Template formato dall’intestazione html comune a tutte le pagine, dal livello header (intestazione grafica della pagina), dal livello contentPage (contenitore della pagina) e dal livello footer (parte bassa con i nomi degli autori). Questo ha comportato un risparmio di tempo a livello di programmazione in quanto per ogni pagina si è dovuto scrivere solamente la parte di codice del livello corpo (parte centrale della pagine diversa per ognuna di essa).

Creazione di funzioni esterne

1. Funzioni Javascript

Durante lo sviluppo del progetto si è reso necessario l’utilizzo di alcune funzioni Javascript che sono state raccolte in un unico file denominato azioni.js

Nello specifico le funzioni utilizzate sono le seguenti:

function viewBrowserDetails – Funzione che controlla il tipo di browser utilizzato dal client in modo da poter prevedere diverse soluzioni in caso di diverse interpretazioni del codice da parte dello stesso.

function cancellaCampo – Funzione che pulisce i campi username e password al momento in cui l’utente si posiziona sulla casella per l’inserimento dei dati.

function funzioneControllo – Funzione che controlla la correttezza grammaticale della mail acquisendo il campo di inserimento come stringa, processandolo carattere per carattere in modo da poter verificare l’avvenuto inserimento del simbolo “@” e del punto

function controlloVuoto – Questa funzione è attivata in modo dinamico alla digitazione di un dato in un campo o alla selezione di una combo box e dà un messaggio visivo di risposta in “real-time”. Attraverso l’indice passato come argomento, controlla che sia avvenuto l’inserimento del relativo

campo nel form. Sempre con questo indice e attraverso l'individuazione DOM, la funzione va a modificare la corrispettiva icona di controllo d'inserimento ponendola a verde se il dato è inserito o lasciandola rossa fino a che il campo non sia stato riempito con qualche dato

function catturaAnno – Funzione che prende il valore corrispondente all'anno di corso scelto dall'utente per la visualizzazione del calendario didattico.

Una sola funzione denominata **function aggiorna**, che si occupa di ricaricare la pagina visualizza.php con i parametri anno di corso e giorno della settimana aggiornati, è stata implementata direttamente nella pagina suddetta. Inoltre è stata utilizzata una funzione denominata `ts_picker` per creare il calendario in una popup che permette di scegliere il giorno di interesse. Questa funzione è stata importata dal sito *html.it* in quanto disponibile come standard usato in molti siti che fanno uso di calendari.

2. Funzioni PHP

Le funzioni PHP utilizzate nello sviluppo del progetto sono state raccolte in un file denominato `Funzioni.php` e vengono richiamate dalle pagine che ne richiedono l'utilizzo.

Nello specifico le funzioni utilizzate sono le seguenti:

function OpenDBMSConnection – Funzione che permette la connessione al Database.

function userNonPresente – Funzione che permette di controllare la presenza nel Database di un utente.

function login Corretto – Funzione che permette di controllare la correttezza del login effettuato da un utente registrato.

function palla – La funzione controlla : che il campo passato in argomento sia stato riempito e nel caso di un campo selezione che questa sia stata effettuata. In fase di invio dei dati la funzione modifica visivamente la corrispettiva icona di controllo di inserimento dei dati posta vicino ad ogni campo.

function ControlloEmail – Funzione che fa la stessa cosa della funzione `Controllo` implementata in javascript. La differenza sta nel fatto che questa effettua il controllo del corretto formato della mail in fase di invio dei dati, validando il formato della stessa.

ControlloData- - Funzione che effettua il controllo del corretto inserimento della data in modo che prima di scrivere il dato sul DB il campo sia inserito in modo corretto con il formato previsto dal sistema che la dovrà gestire.

function inserisciDatiRegistrazione – Funzione che permette di inserire nella tabella `USERS` i dati di un nuovo utente che si registra.

function inserisciDatiAula – Funzione che permette di inserire nella tabella CALL_HALLS i dati relativi ad una nuova aula.

function inserisciDatiModulo – Funzione che permette di inserire nella tabella PROC_MODULES i dati relativi ad un nuovo modulo.

function inserisciDatiOrario – Funzione che permette di inserire i dati relativi ad una specifica fascia oraria nella tabella CAL_ALL_RESERVATIONS.

function modificaDatiOrario – Funzione che permette di modificare i dati relativi ad una specifica fascia oraria nella tabella CAL_ALL_RESERVATIONS.

function cancellaDatiOrario – Funzione che permette di cancellare i dati relativi ad una specifica fascia oraria nella tabella CAL_ALL_RESERVATIONS rendendo di nuovo disponibile l'aula in quella determinata fascia.

function aula Occupata – Funzione che permette di controllare se in un determinato giorno ad una determinata fascia oraria l'aula scelta, in inserimento o aggiornamento, è disponibile oppure no.

function data_ita – Funzione che trasforma il formato di una data da quello anglosassone a quello italiano.

function data_eng – Funzione che trasforma il formato di una data da quello italiano a quello anglosassone.

Sviluppo del progetto

Lo sviluppo del progetto è stato suddiviso in 3 fasi:

FASE 1 : Analisi, progettazione e implementazione del modulo riguardante la registrazione e l'accesso degli utenti

La prima parte del progetto affrontata è stata quella di prevedere la registrazione e il conseguente login degli utenti. Come già specificato in fase di analisi, con l'integrazione nella piattaforma "Life" questa parte sarà eliminata in quanto l'accesso degli utenti è già gestito a livello più alto.

L'Home page di accesso al sistema prevede una piccola spiegazione delle funzionalità del progetto e una maschera in cui inserire username e password se si è un utente registrato e un link ad una pagina di registrazione nel caso non si è ancora registrati. La registrazione prevede la compilazione di una form la cui conferma produce una INSERT nella tabella USERS. La correttezza nell'inserimento dei dati è controllata da alcune funzioni javascript e segnalata in tempo reale da delle icone rosse poste accanto ad ogni campo del form che diventano verdi quando il dato risulta essere corretto. Nel codice php è presente un altro controllo dei dati in fase di "POST", all'aggiornamento della pagina a seguito dell'invio quindi sarà visualizzato un messaggio d'errore nel caso di dati inseriti in modo non corretto e

verrà modificato in rosso anche il colore dell'icona posta vicino ad ogni campo. In caso di inserimento corretto a video verrà visualizzato il relativo messaggio e sarà presente un link per tornare alla pagina di login. La maschera di accesso per utenti già registrati prevede invece un controllo effettuato da una funzione php che effettua una SELECT sulla tabella USERS con le condizioni che l'utente e la password siano quelle inserite. Il controllo viene inviato ad un'altra pagina rispetto a quella dove vengono inseriti i dati che funziona come punto di controllo logico. Se la risposta della query è positiva viene consentito l'accesso al sistema e viene chiamata la pagina di visualizzazione dell'orario, in caso contrario l'errore è segnalato da un messaggio posto in alto di colore rosso, richiamando la pagina iniziale con la maschera di accesso.

FASE 2: Analisi, progettazione e implementazione della tabella di visualizzazione del calendario didattico

Considerando che normalmente l'orario didattico è articolato su base settimanale, si è deciso di implementare una tabella che avesse come periodo minimo la settimana. E' stato perciò seguito lo standard già utilizzato nell'attuale presentazione dell'orario che prevede una tabella che ha per colonne i giorni della settimana e per righe gli orari delle lezioni.

I dati che popolano la tabella provengono dal Database e nello specifico dalla tabella CAL_ALL_RESERVATIONS popolata tramite l'inserimento del calendario didattico da un'apposita maschera. All'accesso dell'utente viene visualizzato l'orario settimanale della didattica impostata per vedere tutti gli anni di corso contemporaneamente della settimana corrente (il giorno corrente viene evidenziato da una colonna di colore diverso). Successivamente la scelta di un diverso giorno di interesse o anno di corso producono una nuova query che aggiorna l'orario stesso. Tramite una SELECT vengono estratti i record dalla tabella CAL_ALL_RESERVATIONS che rispondono alle condizioni dettate dall'utente. I record estratti vengono memorizzati in un recordset. A questo punto, tramite due cicli annidati (uno che controlla le ore e uno i giorni) viene creata dinamicamente la tabella dell'orario didattico.

FASE 3: Analisi, progettazione e implementazione dei menù di gestione

Per quanto riguarda i menù di gestione si è scelto di differenziarli a seconda che l'utente fosse un utente generico o l'amministratore. Naturalmente il secondo è abilitato a vedere anche il menù del primo mentre non è vero il contrario. Il menù dell'utente generico prevede un link per esportare il calendario didattico in un file xml, una combo box per la scelta dell'anno di corso (per default è selezionata la voce "Tutti gli anni") e un tasto rappresentato da un calendario che attiva una popup per la scelta del giorno di interesse. Il link per esportare il calendario è semplicemente il puntamento ad un file che ha del codice php che permette proprio tale operazione e verrà descritto nei dettagli successivamente. La costruzione del menù a tendina è avvenuta associando a quattro opzioni i relativi anni di corso (il 4 rappresenta tutti e tre gli anni). In base alla scelta effettuata il valore servirà come condizione nella SELECT sulla tabella CAL_ALL_RESERVATIONS e l'aggiornamento della pagina avverrà in modo dinamico ad ogni scelta dell'utente eliminando così la presenza di uno scomodo pulsante di invio. Il tasto per selezionare il giorno di interesse invece è associato ad una funzione javascript che è stata importata dal sito *html.it* in quanto disponibile come standard usato in molti siti che fanno uso di calendari. Al momento della scelta, il giorno, dopo essere stato trasformato dal formato anglosassone al formato italiano, viene usato come condizione nella SELECT sulla tabella CAL_ALL_RESERVATIONS.

Le voci di menù dedicate all'amministratore sono tre: Inserimento Aule, Inserimento Moduli, Inserimento Orario. Tutte e tre le voci sono dei link ai rispettivi file che mostrano le rispettive maschere di inserimento. Ogni maschera è costituita da una form di inserimento dati che al momento del salvataggio effettua una INSERT nella rispettiva tabella del Database. Anche qui come nel modulo di registrazione la correttezza nell'inserimento dei dati è controllata da alcune funzioni javascript e segnalata in tempo reale da dei semafori rossi che diventano verdi quando il dato risulta essere corretto. Le form di inserimento moduli e inserimento orario prevedono delle combo box in alcune voci di scelta. Tali caselle sono state realizzate dinamicamente applicando una SELECT sulla rispettiva tabella in modo da visualizzare i dati corretti in fase di inserimento. La modifiche dell'orario avviene invece cliccando direttamente sulla cella corrispondente alla voce da modificare. Il link porta ad una diversa maschera di modifica/cancellazione che permette di cambiare alcuni o tutti i campi o di cancellare il record selezionato. Per effettuare correttamente tali operazioni è stato necessario passare il parametro IDReservation della tabella CAL_ALL_RESERVATIONS corrispondente al record da modificare, alla pagina che contiene la form di modifica/cancellazione. Una volta che si preme il tasto corrispondente viene effettuata una UPDATE o DELETE sulla tabella CAL_ALL_RESERVATIONS.

Problematiche riscontrate

Nell'implementazione della tabella che visualizza il calendario didattico si è dovuto affrontare il problema della gestione delle date. In particolare ci si è concentrati sul fatto che la funzione javascript che crea un calendario in una popup per la scelta del giorno di interesse restituisce la data nel formato anglosassone. Per una corretta usabilità dell'applicazione però si è voluto presentare la data nel formato italiano. Si è per questo scritta una funzione javascript che trasformasse il formato della data da inglese a italiano e viceversa. Alla stessa maniera quando viene mostrata la settimana di interesse con le date nel formato italiano si è reso necessario trasformarle nel formato anglosassone per effettuare la query nel Database che opera con quest'ultimo formato.

Nell'implementare la parte relativa all'inserimento dell'orario ci si è resi conto che occorre gestire la disponibilità o meno delle aule. In particolare al momento di inserire un nuovo orario didattico si doveva verificare che l'aula scelta non fosse già occupata in tale orario già da un altro corso. Per fare ciò è stata realizzata una funzione php che al momento della conferma di inserimento o modifica orario effettua un controllo sulla tabella CAL_ALL_RESERVATIONS effettuando una SELECT con le condizioni specificate. Se tale query produce dei record allora significa che l'aula è già occupata e viene mostrato un messaggio di segnalazione.

Nell'implementazione dei controlli si è fatta la scelta di utilizzare una tecnica di visualizzazione immediata dell'errore "real-time" durante la digitazione oltre che nella fase di invio dei dati. Per ottenere ciò si è dovuto far condividere i controlli in javascript con quelli fatti in php utilizzando come interfaccia di segnalazione degli errori la visualizzazione di una icona accanto ad ogni campo di inserimento. A seconda del colore l'utente può capire se il dato inserito manca o se il campo richiede una particolare attenzione nell'inserimento dei dati (ad es. e-mail, date). La soluzione è stata trovata utilizzando il sistema di riferimento DOM del javascript per far sì che cambi il valore dell'url dell'immagine a cui punta il tag . Quindi ci si è accorti che quando veniva ricaricata la pagina le icone tornavano di colore rosso anche se il campo era già stato inserito. Così è stato implementato in php una memorizzazione dei campi corretti già inseriti e al posto dell'url dell'immagine relativa ad ogni

campo nel form è stato messo del codice php che verifica se quel determinato dato sia stato inserito correttamente modificando il valore url anche in fase di aggiornamento della pagina dovuto al POST dei dati.

Applicazione delle letture al progetto

Le letture che hanno affrontato gli autori del progetto sono : “Don’t make me think!” di Steve Krug e “Identità digitali” di Phillip J. Windley.

Il primo libro detta una serie di principi guida sull’usabilità delle pagine web. Tali principi sono stati rigorosamente applicati nello sviluppo del progetto. In particolare si è cercato di facilitare la navigazione dell’utente usando poche voci di menù chiare e esplicative. La registrazione dell’utente avviene in pochi secondi completando solamente 6 campi. Si è inserita nella Home page una chiara descrizione dell’applicazione. Il logo della piattaforma Life risiede in alto a sinistra e il suo clic riporta alla pagina principale. Nelle maschere di inserimento, modifica e cancellazione dei campi sono state previste delle segnalazioni di errore o di corretta procedura ben visibili e facilmente comprensibili. Accanto al titolo dell’applicazione è previsto un help segnalato da un “?” che da informazioni sul progetto. Non si è fatto mai ricorso di attributi specifici ma sono stati usati solo CSS. Nel rispetto anche dei principi di accessibilità è stato usato l’attributo “alt” sulle immagini che producono link ad altre pagine o a funzioni. Per aumentare il livello di usabilità e accessibilità ci si è assicurati che l’applicazione funzionasse allo stesso modo su diversi browser. Nella fattispecie si è testata con Internet Explorer 7.0 e con Mozilla Firefox 5.0. Infine si è testata anche la velocità di elaborazione delle query installando l’applicazione su di un server remoto provando a fare inserimenti e consultazioni ottenendo ottimi risultati.

Il secondo libro si occupa invece delle politiche di sicurezza e della gestione delle identità digitali. Le tecniche applicabili al progetto in questione si riducono al fatto che nel nostro Database prevediamo una registrazione dei dati utente e un successivo login con inserimento di ID e PASSWORD. Come descritto nel libro anche nel nostro progetto vi è una policy enforcement point (PEP) che è rappresentata dai controlli effettuati in fase di login. Dopo il login i dati vengono subito inviati dalla pagina index.php alla pagina login.php. In questa pagina vi è un policy decision point “PDP” in cui vengono processati i dati e se ne verifica la validità rimandando il risultato alla pagina precedente in caso di dati non corretti o alla pagina di visualizzazione in caso di dati corretti. A seconda dell’appartenenza dell’utente al tipo amministratore o meno le funzionalità del sito passano dalla semplice consultazione di un orario alla possibilità di effettuare modifiche sui dati. Per migliorare la privacy e la sicurezza avremo potuto prevedere una criptazione della password tramite un meccanismo di hash tipo MD5, inoltre per salvaguardare la sicurezza del codice avremo potuto cambiare le impostazioni del server php non concedendo l’accesso diretto alle sottocartelle del sito e permettendo quindi solo l’accesso alla root principale dove dovrebbe essere presente solo la pagina iniziale. Infatti tutte le altre pagine dovrebbero trovarsi su delle sottocartelle con una “policy” di non accesso. In questo modo si può evitare che i file presenti sul server vengano scaricati in locale e il codice possa essere visualizzato ed usato in modo fraudolento per accedere a dei dati riservati come la user e la password dell’amministratore o la user e la password per accedere al DB.

Creazione dei Trigger

I trigger implementati nella base dati sviluppata per il progetto sono due e vengono eseguiti ogni volta che si ha un inserimento o un aggiornamento nella tabella CAL_ALL_RESERVATIONS. Essi sono del tipo 'BEFORE INSERT' e 'BEFORE UPDATE', quindi si attivano prima di eseguire un 'insert' o un 'update' nella suddetta tabella. Nello specifico essi controllano che il periodo minimo di riferimento per qualsiasi evento sia la settimana (o comunque multipli della settimana). Viene quindi controllata se la data inserita nel campo 'reservationDatetimeFrom' sia un lunedì ed in caso negativo la impostano al lunedì appena trascorso, mentre nel campo 'reservationDatetimeTo' controllano se la data inserita sia una domenica ed in caso negativo la impostano alla domenica prossima ventura.

Esportazione di una tabella in XML

Nei requisiti del progetto era richiesta la possibilità di esportare una tabella in un file xml. Si è scelto di esportare la tabella CAL_ALL_RESERVATIONS in quanto quella che rappresenta l'intero calendario didattico e di conseguenza quella più utile ad una ipotetica importazione in una agenda personale. La funzione che effettua l'export è stata inserita in un file e per facilitarne l'uso è stato inserito un link a tale file dal menù generale (quello visualizzato sia dall'amministratore che dall'utente generico). Il codice php per l'esportazione effettua due query fondamentali per la produzione del file. Una prima query di selezione sulla tabella COLUMNS viene effettuata facendo una connessione al DB Information Schema che permette di estrarre la struttura della tabella CAL_ALL_RESERVATIONS poi effettuando il link al Database del progetto una seconda query di selezione sulla tabella stessa per estrarre i dati. Si è scelto di estrarre i dati dei nomi dei tag in modo dinamico in modo da non rimanere vincolati alla struttura di essi e alle future modifiche. Per fare questa operazione è stata trovata una soluzione originale con il collegamento al Database generale "information schema".