



UNIVERSITÀ DI PISA

Corso di Laurea in Informatica Umanistica

RELAZIONE

**Servizio per la creazione e gestione di questionari
on-line.**

Candidato: *Lorenzo Berti*

Relatore: *Gianluigi Ferrari*

Correlatore: *Enrica Salvatori*

Anno Accademico 2011-2012

INDICE

1. Introduzione	1
2. La progettazione	3
3. Come creare un questionario	14
4. Descrizione casi d'uso	17
4.1 Questionario della didattica	17
4.1.1. Definizione obiettivo.....	17
4.1.2. Descrizione Database	18
4.2 Questionario di Indagine Social Network	21
4.2.1 Definizione Obiettivo.....	21
4.2.2. Descrizione Database.....	22
4.3 Questionario di Autovalutazione.....	24
4.3.1 Definizione obiettivo.....	24
4.3.2 Descrizione Database.....	25
4.4 Indagine di mercato per apertura di una libreria	27
4.4.1 Definizione Obiettivo.....	27
4.4.2 Descrizione Database.....	28
5. Conclusioni	28
6. Bibliografia	30
7. Appendice	31
7.1 Guida all'amministratore	31

Ringraziamenti

Ringrazio tutta la mia famiglia che mi ha permesso di raggiungere questo traguardo e un ringraziamento speciale va alla persona che mi ha donato la vita, a cui devo tutto e che nonostante non sia più con me è a lei che dedico tutto.

1. Introduzione

Il progetto di tesi, a cui ho lavorato, ha come obiettivo la creazione di test e sondaggi on-line al fine di valutare statisticamente i dati raccolti.

Ma perché un'azienda o un docente dovrebbe utilizzare un determinato servizio via web invece di utilizzare il tradizionale metodo cartaceo?

La risposta è abbastanza semplice : il servizio via web offre la possibilità di raggiungere un maggior numero di utenti in minor tempo, dato che è reperibile da chiunque munito di connessione ad internet e abbassa notevolmente il lavoro a livello di estrapolazioni dei dati. Il sistema è un sistema flessibile poiché attraverso delle piccole modifiche al codice, cambia l'atteggiamento dell'applicazione.

Le piccole modifiche che ho apportato, rendono quindi il sistema utilizzabile per vari casi. Inoltre è indifferente la piattaforma hardware su cui l'utente sta lavorando oltre ovviamente alla piattaforma operativa, sarà dunque sicuramente utilizzabile sia dall'utente Windows sia dall'utente Mac sia dall'utente Linux ecc.

Si è scelto in particolare di creare dei questionari che permettessero da un lato di individuare gusti, giudizi, abitudini o preferenze di un campione di utenti su determinati argomenti in maniera da poter utilizzare i risultati ottenuti per “disegnare” qualcosa ad *Hoc* per quest'ultimi o fare una indagine di mercato, dall'altro di permettere agli utenti di misurarsi in un test di autovalutazione creato appositamente per preparare lo studente ad un futuro test di ingresso.

Un primo ambito di applicazione è stato quello relativo alla didattica. Sono partito da un semplice questionario della Didattica da porre agli studenti, in modo tale da permettere ai votanti di dare dei giudizi in maniera anonima sulla qualità dei corsi e contestualmente consente agli artefici del test di ottenere sia una media totale di giudizio, sia un giudizio per ogni domanda posta. Se attivato, il questionario sulla didattica fornirebbe dati che permetterebbero di migliorare la didattica nei punti dove gli studenti hanno evidenziato una scarsa qualità del servizio, attraverso la votazione.

È inoltre possibile verificare anche il variare della qualità che la didattica subisce nel tempo, in quanto l'applicazione è stata progettata per memorizzare e effettuare calcoli su un intervallo di tempo mensile.

Proseguendo sulla strada della Didattica, ho modificato l'applicazione in maniera che potesse essere utilizzata per un questionario di autovalutazione.

Quello che in questo caso l'applicazione verifica e calcola, è la percentuale delle risposte corrette o sbagliate che l'utente fornisce, tuttavia oltre al risultato ottenuto, l'applicazione

restituisce la serie di domande in cui l'utente non ha risposto correttamente, indicando la domanda del test, la risposta che l'utente ha dato ed infine la risposta corretta del test. In questa maniera, chiunque può verificare la propria preparazione.

Un'applicazione di questo tipo può essere utilizzata anche per effettuare un test di ingresso o una verifica telematica: a questo fine si dovrebbe solamente modificare il test in modo che risulti aperto on-line solamente per un determinato tempo, ad esempio 2 ore.

Cambiando genere di analisi, l'applicazione creata è stata tarata per servire ad indagini di mercato.

Gli esempi che ho creato sono 2:

- indagine sui Social Network;
- indagine per aprire una Libreria.

L'applicazione sostanzialmente calcola e memorizza in tutte e due i casi d'uso alcuni dati, ovvero memorizza per ogni utente la risposta ad un determinato test, calcola la percentuale di risposte effettuate per ogni test e anche in questo caso è possibile monitorare la variazione di percentuale nel tempo grazie all'intervallo mensile stabilito da un amministratore.

La percentuale che si calcola è chiamata, in statistica, frequenza percentuale, "ossia la frequenza relativa moltiplicata per cento"¹.

La frequenza relativa si ottiene calcolando il rapporto tra il numero di risposte per ciascun quesito e il numero totale di risposte.

Quello che cambia è l'obiettivo, poiché nel primo caso le informazioni che si estraggono dal questionario sono inizialmente per verificare le abitudini dell'utente ponendo domande di base ad esempio quale Social Network l'utente utilizza, per quale motivo lo utilizza ecc. passando poi a domande più specifiche che serviranno per capire se è possibile e se è gradita la creazione di un Social Network Letterario, ovvero un Social Network dove si condividono opere letterarie.

Nel secondo caso, le domande sono poste allo scopo di ottenere informazioni sui gusti degli utenti, in maniera da capire quali sono i libri/autori preferiti o i generi preferiti, con lo scopo di indirizzare la propria attività su quei particolari gusti.

Questo tipo di applicazione potrebbe essere molto utile ad una qualunque azienda per misurare il gradimento dei consumatori.

1 (http://www.dm.unito.it/quadernididattici/garetto/quaderno_statistica.pdf , Maria Garetto, STATISTICA lezione ed esercizi)

2. La Progettazione

Il sistema creato si basa su un sistema Client-Server, basato su un database creato in SQLite. “Un sistema Client-Server (letteralmente cliente-servente) è un'architettura di rete nella quale un computer Client istanzia l'interfaccia utente di un'applicazione connettendosi ad una server application o ad un sistema di database.”²

“SQLite è una libreria software scritta in linguaggio C che implementa un DBMS SQL(...)incorporabile all'interno di applicazioni. Il suo creatore, D. Richard Hipp, lo ha rilasciato nel pubblico dominio, rendendolo utilizzabile quindi senza alcuna restrizione”³.

Questa scelta è dovuta alla portabilità del progetto e alla maggior semplicità del linguaggio SQLite rispetto al più comune MySQL.

L'architettura del sistema è basata, a livello di applicazione, sul protocollo HTTP.

“HTTP(hypertext transfer protocol), (...)è il cuore del Web, definisce in che modo i Client web richiedono le pagine ai server web e come quest'ultimi le trasferiscono ai Client”⁴

A livello di trasporto HTTP si affida a TCP.

TCP è un protocollo a livello di trasporto affidabile, ovvero che garantisce che i dati inviati arrivino sicuramente a destinazione.

Come ho detto inizialmente è un sistema ad architettura Client-Server, in questo tipo di architettura il Client, tramite browser fa una richiesta tramite protocollo HTTP al server, da qui il server, tramite motore PHP e SQL richiederà al database i dati richiesti dal cliente. Infine il server invierà i dati al Client che visualizzerà sullo schermo ciò che stava cercando.



Immagine. 1 Architettura client-server Db Sql

² (Wikipedia, voce Sistema client/server)

³ (Wikipedia, voce SQLite)

⁴ (Reti calcolatori e internet un approccio Top Down, Kurose, James F., Ross, Keith W., B. Mondadori S.p.A 2008)

Nel database vengono inserite le domande relative ai questionari e vengono inserite come porzione di codice HTML.

Potrebbe essere interessante capire il motivo per cui è stata presa la decisione di inserire nel database una porzione di codice invece che i singoli dati.

La risposta a questa domanda è abbastanza semplice.

Questa scelta permettere all'amministratore di :

1. inserire manualmente le domande e quindi una maggiore personalizzazione;
2. rendere più rapido il processo di inserimento o di modifica una volta che si sono memorizzate le domande nel file di inserimento.

Infatti l'amministratore una volta che ha deciso quale domande inserire, lo può fare semplicemente tramite una esecuzione di Query.

Il linguaggio utilizzato per la connessione al database e per l'inserimento è PHP.

“PHP (acronimo di "Hypertext Preprocessor", preprocessore di ipertesti;) è un linguaggio di programmazione interpretato, (...)concepito per la programmazione Web ovvero la realizzazione di pagine web dinamiche.”⁵

“PHP ebbe inizio come progetto condotto e ideato da Rasmus Lerdorf(...).Si trattava di una piccola raccolta di funzioni che aiutavano a rendere automatica la creazione e la gestione di semplici Home Page dell'allora nascente Internet.”⁶

La progettazione del sistema di registrazione dell'utente è uguale in tutti i casi d'uso.

In tutti i casi d'uso l'utente può registrarsi semplicemente tramite Nickname e una volta registrato potrà accedere nuovamente al servizio facendo il Login con lo stesso Nickname.

Questa scelta è stata presa con la consapevolezza che un sistema del genere ha bisogno che il Nickname sia unico per evitare che due utenti abbiano lo stesso Nickname e quindi creare uno scambio di persona.

⁵ (Wikipedia, voce PHP)

⁶ (PHP per professionisti, Peter MacIntyre, Brian Danchilla, Mladen Gogala, Apogeo 2012)

Per fare ciò è stato creata la pagina di gestione dei Login che, nel caso un nuovo utente provi a effettuare la registrazione, verifica che quel Nickname non sia stato già utilizzato:

```
$querycontrollo =
    "SELECT NickName
    FROM Utente
    WHERE NickName = '". $nomeutente. "'";
$controllo = sqlite_query ($querycontrollo, $database);
if($controllo == FALSE){ echo "errore" ;
}else{
    $risultato = sqlite_query ($querycontrollo, $database);
    $num_rows = sqlite_num_rows($risultato);
    if($num_rows>0) {
        $_SESSION['registrato'] = 0;
        unset($nomeutente);
    }else{
        $risultati = sqlite_query ($inserimento , $database );
        if ($risultati==FALSE) {
            echo "Errore nella query: " . mysql_error();
        }else{
            $_SESSION['registrato']=1; $_SESSION['autorizzato'] = 1;
            $user = $_SESSION['nome'];
        }
    }
}
}
```

Il pezzo di codice è abbastanza chiaro, prima si controlla che non ci siano risultati con quel Nickname poi nel caso il risultato vada a buon fine, cioè il numero dei record trovati sia uguale a 0, l'utente viene registrato.

Mentre se per ipotesi un altro utente ha già utilizzato quel particolare Nickname, il risultato del record sarà maggiore di 0 e l'utente non verrà registrato.

```
$_SESSION['registrato']=1;
```

Vorrei mettere in evidenza questa riga poiché ci sono diversi modi per memorizzare i dati in PHP.

Uno sviluppatore potrebbe decidere di memorizzare i dati tramite Cookies o tramite sessione, tuttavia “i dati memorizzati in **\$_COOKIES** non possono essere ritenuti affidabili(...) dato che i Cookies si trovano sul lato Client e si possono modificare con facilità”⁷, per questo motivo, per “i dati sensibili andrebbero impiegati gli array **\$_SESSION** sul lato server.”⁸

7 (PHP per professionisti, Peter MacIntyre, Brian Danchilla, Mladen Gogala, Apogeo 2012)

8 (PHP per professionisti, Peter MacIntyre, Brian Danchilla, Mladen Gogala, Apogeo 2012)

L'avvenuta registrazione viene rappresentata da in questa maniera:



Immagine. 2 View Registrazione avvenuta

L'avvenuta registrazione viene mostrata attraverso un Alert creato con un plug-in di jQuery chiamato "jAlert".

"jQuery è una libreria di funzioni che permette di integrare in una pagina web le possibilità interattive di JavaScript con poche, eleganti, righe di codice."⁹

Spiegato il sistema di registrazione potrebbe essere interessante affrontare le varie modalità di risposta che all'utente vengono proposte.

Le modalità di risposta che l'amministratore può scegliere sono tre:

La modalità "Default", dove le domande vengono poste una dopo l'altra controllando che all'utente non venga mai posta una domanda che ha già fatto.

Il controllo avviene grazie ad una Query:

```
(SELECT id_test, Test_Question
FROM Admin)
    EXCEPT
(SELECT id_test, Test_Question
FROM Admin JOIN RispostaUtente
WHERE id_test = id_testUtente AND NickNameRisposta = '". $nick.'" )
order by id_test ASC
```

L'idea è quella di prendere tutti i test(domande) eccetto quelli a cui l'utente, che ha effettuato il Login(**\$nick**) , non ha ancora risposto.

Differente è la modalità scelta per coprire il più possibile il Database:

In questa modalità le prime domande che vengono poste all'utente sono quelle che non hanno ancora avuto una risposta, successivamente vengono poste in ordine di risposta, cioè vengono ordinate secondo il numero di risposte che hanno ricevuto, quelle con meno risposte vengono proposte prima.

9 (jQuery Guida completa, Earle Castledine, Craig Sharkie, Apogeo 2011)

Anche in questo caso all'utente non vengono mai mostrate nuovamente le domande a cui ha già risposto.

In questo caso la prima cosa da fare è fare un conteggio delle domande per risposte e ordinarle secondo questo conteggio:

```
("SELECT id_test, Test_Question, COUNT(*) AS co
FROM
(SELECT id_test, Test_Question
FROM Admin JOIN RispostaUtente
WHERE id_test=id_testUtente AND id_test = '". $idtestM[$indiceY]."' )
GROUP BY id_test");
```

idtestM[] è l'array degli id che a quell'utente mancano.

Fatto questo bisogna creare degli array con i risultati ottenuti:

```
$IdtestMancanti[] = $contenuti['id_test'];
$IdtestMancantiConteggio[] = $contenuti['co'];
```

In questo caso il primo array rappresenterebbe gli "Id" del test e il secondo relativo agli "Id" con il conteggio delle risposte.

Fatti questi due array con i relativi risultati si possono ordinare entrambi, mantenendo le relazioni di indice con la funzione

```
array_multisort($IdtestMancanti, SORT_ASC, $IdtestMancantiConConteggio);
```

Con quest'ultima Query, viene infine preso il test giusto.

```
SELECT id_test ,Test_Question, NickNameRisposta, COUNT(*) AS Conteggio
FROM
(SELECT DISTINCT id_test, Test_Question, id_testUtente, NickNameRisposta
FROM Admin JOIN RispostaUtente
WHERE id_test = id_testUtente )
WHERE id_test=id_testUtente AND id_test='". IdtestMancantiConteggio[$z]."'
GROUP BY id_test
order by Conteggio ASC LIMIT 1
```

Ovviamente **\$z** viene definito prima ed è l'indice che scorrerà l' array dei test.

L'ultima modalità è la modalità di disambiguazione delle domande, ovvero vengono poste prima le domande che hanno avuto maggiori diversi tipi di risposta.

Per spiegare questo bisogna semplicemente spiegare come si ottiene l'indice per ordinarli.

Gli indici si ottengono tramite il calcolo di un indice di una statistica: Fleiss' kappa.

Per calcolare l'indice necessario quello che bisogna fare è:

- per ogni test calcolare il numero di tutte le risposte ricevute per quel test;
- calcolare per ogni tipo di risposta di questo test, il numero ed elevarlo al quadrato;
- sommare i quadrati di ogni tipo di risposta e alla somma di questi bisogna sottrarci il numero di tutte le risposte ricevute per quel test;
- calcolare il valore ottenuto dall'operazione:
 $(1/ \text{Numero totale risposte per test}(\text{Numero totale risposte per test}-1))$;
- moltiplicare quest'ultimo valore con il valore ottenuto in precedenza dalla somma delle potenze e sottrazione;
- il valore ottenuto sarà un valore che andrà da 0 a 1, più il valore si avvicina a 1 e più i tipi di risposta sono coerenti, cioè uguali e quindi il test è più affidabile.

Per fare un esempio pratico, si può prendere l'esempio proposto da Wikipedia alla voce Fleiss' kappa:

	Tipo della Risposta					Indice PI
	A	B	C	D	E	/
Test N. 1	0	2	6	4	2	0,25
Test N.2	0	0	3	5	6	0,31
Test N.3	3	2	6	3	0	0,24
Totale	3	4	15	12	8	//
Indice PJ	0,07	0,09	0,35	0,28	0,19	

Tab. 1 Tabella Calcolo Indici

Calcolo indice PI:

$$(1/ 14(14-1)) \times (0^2 + 2^2 + 6^2 + 4^2 + 2^2 - 14) = 0,25$$

I test verranno messi in ordine crescente secondo quell'indice.

Oltre a quell'indice viene calcolato un secondo indice(PJ), che verrà calcolato invece che per righe, per colonne. Si ottiene dividendo la somma di ogni riga(Tipo di risposta) per quel test per il numero delle risposte totali, nell'esempio sopra:

$$0,07 = 3 / 42$$

Dove 3 è il totale delle risposte di tipo A e 42 è il totale delle risposte.

Anche questo indice sarà utile al calcolo definitivo del valore K, valore che determinerà l'attendibilità del questionario, calcolando anch'esso un valore da 0 a 1. Più il valore si avvicina all'uno più il questionario è attendibile, cioè tanti utenti hanno risposto in maniera simile.

La K si ottiene calcolando diversi valori che si ottengono dagli indici:

$$P1 = (1 / (\text{Numero dei test})) * \text{Somma Indici PI};$$

$$P2 = \text{Elevare al quadrato gli indici PJ e sommarli}$$

$$K = (P1 - P2) / (1 - P2)$$

Ora che sono concluse le modalità di risposta si può passare alle statistiche.

Le statistiche sono globali o mensili.

Le statistiche mensili, a differenza di quelle globali che vengono calcolate senza dar importanza alla data, vengono calcolate confrontando la data del mese stabilito con il mese delle risposte effettuate.

Per prima cosa quindi bisogna prendere ad esempio il mese precedente:

```
$data=date('d/m/Y', mktime(0,0,0,date(n)-1,date(d)-1,date(Y)));
```

Poi possiamo quindi prendere il valore del mese tramite una sottostringa in maniera da prendere solamente il numero del mese.

```
$mese=substr($data, 4,6);
```

Fatto ciò quando si estrarrà una qualsiasi statistica basterà confrontare i mesi:

```
SELECT DataIscrizione,COUNT(NickName) AS Conto
FROM Utente
WHERE SUBSTR(DataIscrizione, 4, 7) = ".$mese."
GROUP BY DataIscrizione ORDER BY DataIscrizione"
```

In questo particolare caso, restituisce una statistica relativa alle iscrizioni, cioè ci dice quanti utenti si sono iscritti nel mese scelto.

Ho deciso di prendere il mese come sottostringa in maniera che fosse più semplice confrontare i mesi, evitando così di confrontare giorno ecc.

Le statistiche variano dai casi d'uso, ad esempio nel caso della didattica quello che otteniamo è una media delle votazioni, mentre nei casi d'uso dei questionari su Social Network e sull'indagine di mercato per una libreria otteniamo invece, una percentuale sulle risposte.

Nel Questionario della didattica, la media della risposta viene restituita tramite un valore da Insufficiente a Ottimo, questo valore è possibile calcolarlo in questa maniera:

```
$query=("SELECT COUNT(*) AS totRisposte FROM(
        SELECT Id_testUtente FROM RispostaUtente)
");
$risultato=sqlite_query($database,$query);
$y=0;
while($contenuto[$y]=sqlite_fetch_array($risultato)) {
$totale=$contenuto[$y]['totRisposte'];

$query2="SELECT SUM(Answer) AS ris FROM
        (SELECT id_test,Answer,NickNameRisposta
         FROM Admin JOIN RispostaUtente
         WHERE id_testUtente = id_test
        )";
$risultato2=sqlite_query($database,$query2);
$j=0;

while($contenuto2[$j]=sqlite_fetch_array($risultato2)) {
    $massimo = $contenuto2[$j]['ris'];
    $finale = "";
    if($massimo>0){
        $media=$totale/$massimo;
        $valore=$massimo/$totale;
        if($valore <= 1){ $finale = "Insufficiente";}
        if(($valore<2)&&($valore>1) ){ $finale = "Sufficiente";}
        if(($valore < 3)&&( $valore >= 2) ){ $finale = "Buono";}
        if(($valore <=4)&&($valore >= 3) ){ $finale = "Ottimo";}
    }
}
}
```

In questo codice si può notare come si effettua la media dei voti, con la prima Query si estrapola il numero totale delle risposte ottenute nel Database, mentre con la seconda si estrae la somma dei valori delle risposte, che è stato definito nel Database come intero, in

maniera che dividendo la somma per il numero totale di risposte si ottiene la media che poi a seconda del risultato verrà stampato come voto.

Nei casi d'uso dell'indagine per la libreria e del Social Network la cosa è diversa.

Quello che bisogna calcolare, come ho detto precedentemente, è la percentuale.

Questa si calcola prendendo prima il numero delle risposte di un determinato tipo per un test dividendole poi per tutte le risposte riferite a quel test, il risultato viene moltiplicato per cento e quello che si ottiene è la percentuale corrispondente a quella risposta per quel test.

Nell'ultimo caso d'uso, quello relativo al test di autovalutazione sui vari campi, l'utente collegandosi al sistema, può andare nella sezione delle risposte dove vengono mostrate le sezioni relative al test e dove vorrebbe esercitarsi, in questo caso ho inserito tre sezioni.



Immagine. 3 View Scelta sezioni

Questo è un caso in cui l'utente pone il mouse sopra la sezione numero 1 quella relativa alla comprensione del testo.

In queste diverse sezioni sono presentate diverse domande, nella sezione della comprensione, quello che l'utente deve fare è scegliere un brano su cui esercitarsi, scelto il brano gli verranno messe di fronte le domande relative a quel brano e potrà consultare il brano attraverso il link offerto dal sistema.

Nelle altre sezioni vengono poste delle domande ordinate per sottosezioni, ad esempio le prime 5 domande di Conoscenza della Grammatica saranno sulla grammatica ed ortografia alcune successive sull'analisi grammaticale e così via, in maniera da dare un test più conforme possibile a quelli che andrà a effettuare durante il test di ingresso.

A fine di ogni sezione verrà offerto una correzione delle risposte appena effettuate con una percentuale delle risposte giuste e sbagliate.

Le domande sono poste casualmente, ovvero, vengono scelte tra un set di domande, cinque domande di ciascuna sotto categoria.

Ho scelto di farle visualizzare casualmente e che un utente possa effettuare una sola volta il test, in maniera che l'utente possa avere di fronte un questionario differente da un altro utente, così da rendere utilizzabile l'applicazione anche per una verifica telematica.

Nel caso un utente utilizzi il sistema come autovalutazione, una volta completate le sezioni, gli basterà iscriversi con un altro NickName per effettuare nuovamente i test.

Questa applicazione potrebbe quindi essere utilizzata nel tempo, ovvero ogni anno l'amministratore può inserire domande differenti senza dover cancellare le precedenti, mantenendole nel file di inserimento.

È utile la divisione in sezioni perché ad esempio un anno potrebbe esserci una sezione mentre in un successivo anno no, allora è compito dell'amministratore dover scegliere le sezioni da mostrare all'utente.

Per i diversi casi d'uso, sono state prese differenti decisioni riguardo al database poiché al variare dell'obiettivo variavano alcuni dati da estrarre, come ad esempio nel questionario della didattica: l'obiettivo è quello di estrarre un punteggio medio che va da insufficiente a ottimo quindi sarà certamente necessario che nel database, nella tabella delle risposte sia presente un campo della risposta di tipo INT poiché alla fine dovremo effettuare una somma e calcolare una media matematica .

Mentre nel caso d'uso dell'indagine di mercato non dobbiamo recuperare nessuna valutazione ma solamente delle preferenze, quindi sarà necessario che il campo delle risposte non sia come nel campo precedente un intero, ma che sia un campo di stringhe.

Infine nella progettazione del caso d'uso per i test di autovalutazione ho deciso di porre i quesiti relativi ai brani uno ad uno, mentre per le altre sezioni è stato necessario mostrare i risultati come unico questionario per permettere di visualizzare un numero di domande casuali e che non si ripetessero.

Per una buona progettazione del sistema, ogni caso d'uso ha bisogno di un suo diagramma.

“I diagrammi dei casi d'uso mostrano dal punto di vista orientato all'attività dell'utente”¹⁰ ovvero rappresenta un'attività che l'utente sta per compiere con il sistema progettato.

Io ho creato un diagramma di esempio per il caso d'uso del test di autovalutazione.

¹⁰ (PHP6 Guida per lo sviluppatore, Ed Lecky-Thompson, Steven D.Nowicki, Thomas Myer, Ulrico Hoepli Editore 2009)

In questo caso l'utente può effettuare 3 azioni:

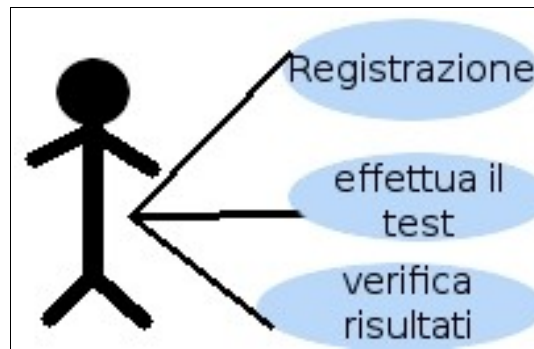


Immagine. 4 Diagramma caso d'uso Test Autovalutazione

1. l'utente accede al sito Web e si iscrive;
2. l'utente va nella sezione dei test e risponde alle domande che gli vengono poste;
3. l'utente finito il test verifica il proprio risultato.

Questo tipo di diagramma dovrebbe essere creato per semplificare e schematizzare ciò che il sistema deve permettere di fare all'utente.

Infine si può aggiungere che il sistema creato è un sistema modulare. Questo sistema ci torna utile quando abbiamo a che fare con siti che presentano tante pagine con la stessa struttura, ma differenti tra loro solo per i contenuti.

3. Come creare un questionario

Per costruire un Questionario che sia utile ad un sondaggio di mercato o a ottenere dei dati da utilizzare per una valutazione, si devono seguire alcuni passi. Prima di tutto bisogna stabilire il campo d'indagine e stabilire gli obiettivi da raggiungere, ovvero a cosa serve il questionario che vogliamo creare.

Prendendo come esempio i casi delle mie ricerche, è stato creato un questionario che ha come obiettivo il raccoglimento dei dati per monitorare la soddisfazione degli studenti. Fatto questo occorre perciò stabilire anche a chi rivolgere il questionario che, nel caso sopra citato, saranno sicuramente gli studenti.

Per quanto riguarda il linguaggio da utilizzare in un questionario è utile che “il linguaggio non sia dispregiativo o elogiativo oppure troppo complesso.”¹¹

È importante che le domande vengano poste una alla volta e che non ci siano più domande in un quesito.

“La difficoltà successiva è quella della scelta delle domande da utilizzare per ottenere uno specifico tipo di risposte e di informazioni(..)”¹², quindi, stabilire se per la nostra ricerca o valutazione sia più utile inserire domande aperte o domande a risposta multipla.

“In funzione della loro forma tecnica, le domande possono essere classificate in:

1. **Domande aperte:** non prevedono risposte predefinite dal ricercatore e che consentono piena libertà (...) al soggetto intervistato
2. **Domande chiuse:** prevedono un ventaglio di risposte definite a priori dal ricercatore.”¹³

Ciò dipende da come si vuole impiegare e controllare i dati ottenuti: ad esempio, nel caso in cui un docente volesse utilizzare l'applicazione per una verifica telematica a risposta aperta, deve sapere che poi sarà lui a dover verificare le risposte, mentre nel caso delle risposte multiple potrebbe essere il sistema a correggere il tutto e restituire il giudizio.

11 (www.istat.it/it/files/2010/09/lineeguida.pdf , Linee guida metodologiche per rilevazioni statistiche, Marco Fortini, Istituto Nazionale di Statistica, 2000)

12 (www.urp.it/Sezione.jsp?idSezione=922&idSezioneRif=62 , Il processo di costruzione di un questionario)

13 (www.urp.it/Sezione.jsp?idSezione=922&idSezioneRif=62 , Il processo di costruzione di un questionario)

Questo perché con domande a risposta multipla, il sistema riceve dagli studenti risposte ben precise da poter confrontare con le risposte giuste ed è quindi in grado di stabilire tramite algoritmi se una risposta data è corretta o errata, mentre nel caso di risposta aperta, non esiste una vera e propria risposta corretta e sarà quindi compito del docente giudicare.

Non esiste un metodo migliore dell'altro in quanto tutti i due i metodi hanno sia vantaggi che svantaggi:

- nel caso delle domande a risposta multipla, come ho detto sarà possibile analizzare i dati in maniera più rapida e più semplice, ma non dobbiamo dimenticare che nel caso che un utente non sappia cosa rispondere può rispondere a caso;
- nel caso delle domande a risposta aperta è certamente impossibile che un utente risponda a caso, ma a quel punto sarà più complicato controllare le risposte con il sistema e sarà quindi compito di un amministratore controllare le risposte.

Importante è inoltre la sequenza delle domande poiché “i quesiti che implicano uno sforzo di memoria andrebbero collocati verso la metà del questionario, per evitare che all’inizio il rispondente non sia ancora disponibile a tale impegno e alla fine sia troppo stanco.”¹⁴

“I quesiti su temi delicati da affrontare andrebbero collocati verso la fine, per sfruttare la maggiore confidenza e disponibilità(...)”¹⁵ dell'utente.

Il passo successivo sarà scegliere le domande da porre, ovvero che cosa chiedere all’utente in maniera che i risultati siano utili al nostro obiettivo.

Quest’ultimo passo è sostanzialmente il più importante, in quanto sbagliando le domande poste, i risultati saranno completamente inutili al nostro caso di studio.

I quesiti, soprattutto in una indagine di mercato, sono in genere presentati in maniera che prima vengano poste delle domande base che saranno utili ad avere caratteristiche generali sull’utente che risponde, ad esempio, sempre partendo da un mio esperimento su una potenziale indagine di mercato per l’apertura di una libreria, all’utente verrà richiesto di specificare età, professione, quanto tempo dedica alla lettura ecc. I successivi quesiti saranno invece più utili per la nostra ricerca, ovvero come nel caso dell’indagine di mercato per la libreria, sarà certamente utile capire quale è il proprio autore preferito o il genere di libro che preferisce o ancora il titolo del libro preferito.

14 (http://www.ds.unifi.it/concorsoscuole/materiale/intro_al_questionario.pdf , Carla Rampichini, Dipartimento di Statistica ‘G. Parenti’)

15 (http://www.ds.unifi.it/concorsoscuole/materiale/intro_al_questionario.pdf , Carla Rampichini, Dipartimento di Statistica ‘G. Parenti’)

In alcuni casi, potrebbe essere però utile modificare l'ordine delle domande, potrebbe essere utile, nel caso di un questionario con numerosi quesiti, porre quest'ultimi ordinati secondo il numero di risposte ricevute, cioè porre per prima i quesiti che hanno ricevuto meno risposte, con lo scopo di ottenere un Database più completo possibile.

Un'altra possibilità potrebbe essere quella di poter porre i quesiti ordinandoli secondo risposte differenti, ovvero mostrare prima le domande a cui sono state date più tipi di risposte diverse in maniera da poter indirizzare verso un risultato più certo. Questo compito spetterà all'amministratore del questionario che dovrà decidere quale secondo lui, al momento, è il miglior modo di agire per ottenere i propri risultati.

Lo studio del questionario è fondamentale anche per un caso di valutazione, se il questionario è troppo semplice o troppo difficile, risulterà inefficiente e i risultati non saranno utilizzabili.

“Prima di rilasciare la versione definitiva del questionario occorre valutare se:

1. risponde alle esigenze (...) dell'indagine;
2. sono state omesse domande;
3. linguaggio e struttura delle domande sono adeguati;
4. è facilmente comprensibile per gli intervistati e semplice da gestire per gli intervistatori.”¹⁶

Infine, per quanto riguarda i test di autovalutazione, vorrei evidenziare il fatto che, negli ultimi anni, stanno moltiplicando la propria presenza sul web grazie alla loro utilità e comodità. On-line si trovano guide su come progettare un buon test di verifica che quindi ci “permette di comprendere se e in quale misura gli obiettivi didattici specifici cui si riferisce sono stati raggiunti.”¹⁷

16 (<http://www.istat.it/it/files/2010/09/lineeguida.pdf> , Linee guida metodologiche per rilevazioni statistiche, Marco Fortini, Istituto Nazionale di Statistica, 2000)

17 (<http://www.exelearning.it>, Progettiamo correttamente un test di verifica, Furio Petrossi)

4. Descrizione casi d'uso

4.1. Questionario didattica

4.1.1. Definizione obiettivo

L'obiettivo di questo caso d'uso è di poter permettere agli studenti di effettuare questionari di valutazione didattica, elettronicamente e non tramite uso cartaceo.

Questo metodo permette inoltre un'analisi più rapida e semplice dei risultati ottenuti. I dati ottenuti, verranno infatti analizzati dal sistema fornendo una media del voto, sia complessiva che in un intervallo di tempo scelto, i voti degli utenti andranno dal 1 a 4 che corrisponderanno da insufficiente a Ottimo.

Verranno inoltre mostrati dati come numero di Studenti che effettuano il Questionario, un numero di risposte globali effettuate insieme al numero di domande a cui non è stata ancora data una risposta, in questa maniera si può controllare se tutti gli utenti hanno completato il Questionario, nel caso in cui ci siano ancora domande senza risposta possiamo procedere a cambiare metodo di risposta in maniera da completare il questionario.

Verrà mostrata anche una classifica di utenti per risposte, in maniera da verificare quali utenti hanno risposto al numero maggiore di test posti.

Verranno mostrate sia la classifica dei test per numero di risposte, sia un particolare indice di statistica: il Fleiss'kappa.

“Dal nome di Joseph L. Fleiss , il Fleiss'Kappa, è una misura statistica per valutare l'accordo tra un numero fisso di valutatori durante l'assegnazione di giudizi, la misura calcola il grado di accordo(..)”.¹⁸

Questa statistica è molto utile in quanto è possibile notare l'affidabilità del risultato finale. Se per esempio tutti gli studenti sosterranno che un particolare corso è organizzato bene, si può dire con particolare certezza che questo corso sia effettivamente organizzato bene, mentre se alla medesima domanda abbiamo molta discordanza tra i votanti, quello che ci sarà da verificare sarà perché per alcuni è un buon corso mentre per altri non lo è.

Lo scopo principale è infine quello di poter, tramite il giudizio degli studenti, migliorare la didattica e verificare miglioramenti o peggioramenti in determinati intervalli di tempo.

18 (Wikipedia, voce Fleiss'_kappa)

4.1.2. Descrizione Database

Tabella Admin

Campo	Tipo	Indice	Auto_Increment
Id_test	INT	PRIMARY KEY	Si
Test_Question	CLOB	//	//

Tab. 2 Tabella delle domande

La tabella “**Admin**” è formata da due campi che corrispondono ad un ID univoco del quesito e la domanda (Test_Question).

La domanda viene registrata nel Database come porzione di codice HTML.

Quando poi verrà stampata a video, il codice HTML verrà formattato nella maniera che più si desidera tramite CSS.

Quello che all’utente verrà mostrato nel mio caso d’uso sarà ad esempio:

Domanda Numero:2

Adeguatezza delle proprie conoscenze iniziali rispetto a quelle richieste dai corsi di insegnamento

Insufficiente
 Sufficiente
 Buono
 Ottimo

Immagine 5. Esempio quesito

Tabella **RispostaUtente**

Campo	Tipo	Indice	Auto_Increment
Id_testUtente	INT	FOREIGN KEY (reference Admin.id_test)	//
NickNameRisposta	CHAR	//	//
Answer	INT	//	//
Data	DATE	//	//

Tab. 3 Tabella delle risposte

PRIMARY KEY(id_testUtente, NickNameRisposta)

La tabella “**RispostaUtente**” contiene tutte le risposte che vengono date da ciascun utente. “Id_testUtente” corrisponde al test a cui è stata data la risposta ed è una FOREIGN KEY proprio perché fa riferimento a “Id_test” della tabella “**Admin**”.

In “**NickNameRisposta**” andrà l’utente che ha risposto alla domanda, mentre in “**Answer**” andrà il valore della risposta, in questo caso c’è un intero come tipo, poiché il valore che il form contiene è un intero.

Nel campo “Data” andrà la data in cui l’utente risponde. La PRIMARY KEY di questa tabella è la coppia (id_testUtente, NickNameRisposta) in quanto un utente può rispondere una sola volta a quel Test.

Un esempio di record potrebbe essere:

Id_testUtente	NickNameRisposta	Answer	Data
1	Lorenzo	2	09/10/2012

Tab. 4 Esempio Record utente

Tabella Opzioni

Campo	Tipo	Indice	Auto_Increment
Modalità	CHAR	PRIMARY KEY	//

Tab. 5 Modalità di Risposta

Nella tabella opzioni viene inserita la Modalità di risposta.

Come ho detto nella sezione riguardante la progettazione, l'amministratore può scegliere il metodo che più preferisce per mostrare le domande all'utente, nel mio progetto ho inserito 3 metodi di risposta:

1. **Default:** i quesiti vengono mostrati in ordine consecutivo e all'utente viene posto il primo test a cui non ha risposto.
2. **Coprire Dataset:** i quesiti vengono mostrati in ordine di numero di risposte, il quesito con minor risposte o con 0 risposte viene posto prima.
3. **Risolvere Ambiguità:** i quesiti vengono proposti secondo la differenza delle risposte, ovvero il test che ha ricevuto maggiori risposte differenti viene presentato per primo.

Tabella LogAdmin

Campo	Tipo	Indice	Auto_Increment
Name	CHAR	PRIMARY KEY	//

Tab. 6 Login Amministratore

La tabella "**LogAdmin**" contiene il nome degli amministratori che accederanno all'area Amministratori attraverso un Login, da qui potranno scegliere il metodo di porre le domande sopra citato.

Tabella Utente

Campo	Tipo	Indice	Auto_Increment
idUtente	INT	PRIMARY KEY	SI
NickName	CHAR	//	//
DataIscrizione	Date	//	//

Tab. 7 Tabella degli utenti

Nella tabella utente vengono memorizzati i dati dell'utente, ovvero un Id univoco, un NickName che sarà solo suo in modo che se per qualche motivo dovesse interrompere il questionario potrà completarlo in un secondo momento facendo il Login al sistema e ripartirà da dove era rimasto.

Nel campo "Data iscrizione" andrà la data di quando il NickName è stato registrato in maniera da poter monitorare il numero di Iscritti in un certo intervallo di tempo.

4.2. Questionario di Indagine Social Network

4.2.1. Definizione Obiettivo

L'obiettivo di questo caso d'uso è quello di poter effettuare un sondaggio su un determinato campione di consumatori, al fine di capire se è possibile creare un Social Network Letterario, se quest'ultimo verrebbe utilizzato e in che maniera, in modo da poterlo costruire su misura per i nostri utenti.

In questo genere di indagine, è spesso necessario dapprima "valutare se esistono spazi di mercato per la costituzione di una nuova attività"¹⁹, poi occorrerà recuperare dati che riguardano le generalità dell'utente che risponde, in maniera da capire quale poi sarà il tipo di utente che utilizzerà il Social Network, ed è per questo che alcune domande del questionario sono poste per recuperare dati sull'età o la professione di quest'ultimo oltre a cercare di capire come utilizza generalmente i Social Network più in voga del momento.

Oltre ai dati generali, occorre ovviamente porre domande specifiche sull'argomento che vogliamo approfondire, in questo caso domande su come gli utenti utilizzerebbero un Social Network Letterario, quali sarebbero le funzioni che vorrebbero avere, per quale scopo lo utilizzerebbero, per quale ragione si iscriverebbero a questo particolare Social Network e così via.

L'obiettivo da raggiungere è quello di raccogliere dati e preferenze degli utenti per capire effettivamente cosa gli utenti vorrebbero avere di fronte quando si collegano al nuovo Social Network.

I dati sono raccolti tramite un calcolo percentuale su ogni test, così che si possa avere una visione globale del sondaggio, ma potrebbe essere utile avere anche una visione unitaria della statistica, occorrerebbe quindi sapere, utente per utente o test per test le risposte singole ed è per questo che sono state create delle statistiche generali in cui l'amministratore può scegliere come visualizzare le risposte, può scegliere se visualizzarle per Utente o per Id del Test.

¹⁹ (www.formaper.it/sociale2/ricerca/ricerca.html)

Un ultimo obiettivo è cercare di controllare le preferenze a intervalli costanti, cioè ad esempio potrebbe essere utile verificare il variare delle preferenze mese per mese.

Proprio a questo proposito l'amministratore ha a disposizione una sezione dedicata alle statistiche mensili, che fanno riferimento ad un mese da lui scelto, ovvero se volesse controllare ad esempio le statistiche per il mese di Settembre, non dovrebbe far altro che aprire la pagina "StatisticheMensili.php" e modificare la data memorizzata nella variabile "\$data" ed inserire ad esempio "\$data = 01/09/2012", in modo che il sistema prenda solamente le statistiche riferite a quel particolare mese, e restituisca la percentuale di risposte per quel mese e statistiche giorno per giorno.

4.2.2. Descrizione Database

Il database di questo caso d'uso è molto simile a quello del caso d'uso precedente.

Anche il funzionamento del Database è molto simile, differisce particolarmente nel risultato ottenuto con i dati, poiché nel caso precedente ottenevamo una votazione da parte degli utenti, mentre adesso avremo diverse risposte che serviranno per calcolare le percentuali di ogni risposta.

La tabella che differisce dal precedente caso d'uso è la tabella RispostaUtente, ovvero dove vengono memorizzate tutte le risposte per ogni utente.

Tabella **RispostaUtente**:

Campo	Tipo	Indice	Auto_Increment
Id_testUtente	INT	FOREIGN KEY (reference Admin.id_test)	//
NickNameRisposta	CHAR	//	//
Answer	VARCHAR	//	//
Data	Date	//	//

Tab. 8 Tabella risposte con tipo differente

PRIMARY KEY(id_testUtente, NickNameRisposta, Answer)

Come è possibile notare la sola modifica che è stata apportata è sul tipo di dato che è contenuto nel campo "Answer", mentre nel caso precedente il campo "Answer" era di tipo

INT dato che si trattava di votazioni, in questo caso il campo è di tipo VARCHAR questo perché nel questionario ci sono domande che non hanno un valore numerico.

Ad esempio, prendendo una Domanda del questionario:

“Quale funzione vorresti aggiungere al tuo/tuoi Social Network?”

Le risposte possono essere varie ma non avranno mai un valore numerico come risultato, ma sempre una stringa.

Le domande a risposta aperta, come quella che ho appena citato, possono però avere uno svantaggio, ovvero può non essere semplice poi confrontarle per restituire percentuali precise, quello che si può fare quindi è dare delle indicazioni su come scrivere per cercare di limitare le differenze.

Ad esempio nel quesito precedente potremmo aggiungere tra parentesi l’istruzione di risposta:

“Quale funzione vorresti aggiungere al tuo/tuoi Social Network?(scrivere lista separata da virgola es. commentare,condividere video,condividere foto, leggere note ecc.)”

In maniera da limitare le differenze tra una risposta e l’altra.

In un questionario di questo tipo, quesiti a risposta multipla potrebbero limitare la scelta degli utenti e potrebbero quindi non corrispondere alle loro reali scelte.

D’altro canto è chiaro, che potrebbe essere problematico gestire quesiti a risposta aperta.

In questo caso ho preferito fare un mix di domande a risposta multipla e risposta aperta per raggiungere gli obiettivi stabiliti.

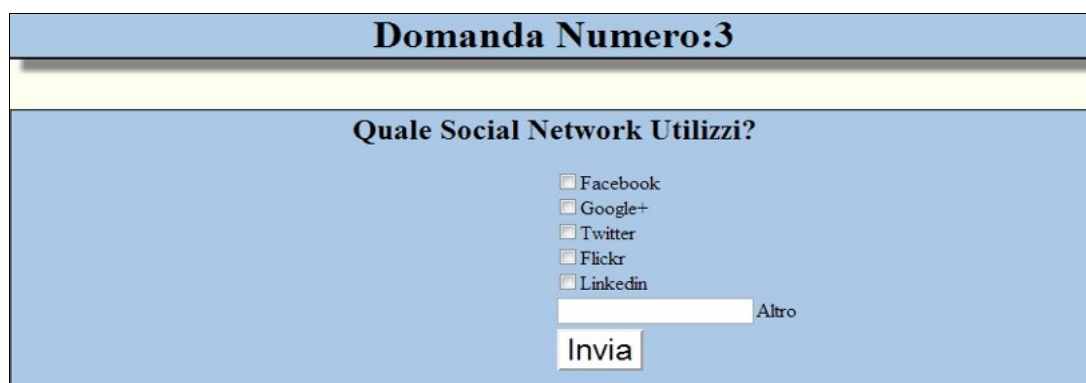
Questo perché talvolta è possibile che le opzioni offerte dal questionario non siano sufficienti.

Un esempio di questa occasione può essere offerto da una domanda di questo tipo:

“Quale Social Network utilizzi?”

In questo caso il sistema offre alcune scelte ma potrebbero non bastare.

L’esempio della domanda in questione sarà:



Domanda Numero:3

Quale Social Network Utilizzi?

Facebook

Google+

Twitter

Flickr

LinkedIn

Altro

Invia

Immagine 6. Esempio domanda risposta aperta e multipla

Nell'esempio sopra citato è chiaro che l'utente potrebbe essere iscritto ad un altro Social Network tra quelli che non sono offerti dalle Checkbox.

Quando l'utente ha scelto la sua risposta o le sue risposte, cliccando sul tasto "submit" invierà i dati al database che verranno memorizzati e utilizzati per calcolare le statistiche.

Le altre tabelle, ovvero quelle corrispondenti alla tabella delle domande(v. tab. 2) , alla tabella degli utenti(v. tab.7), alla tabella dell'accesso per l'amministratore(v. tab. 6), alla tabella delle opzioni di risposta(v. tab. 5) sono identiche a quelle del caso d'uso precedente.

4.3. Questionario di Autovalutazione

4.3.1. Definizione obiettivo

L'obiettivo di questo caso d'uso è quello di permettere a chiunque voglia, di misurare la sua conoscenza in un particolare argomento.

Il caso d'uso vorrebbe quindi dare un supporto ulteriore ai vari manuali che si possono trovare sul mercato, vorrei a questo proposito citare Alpha test che "è la prima e la più importante organizzazione in Italia che offre agli studenti tutti gli strumenti necessari per affrontare con successo i test selettivi. Opera a livello nazionale(..) e pubblica la collana TestUniversitari composta da oltre sessanta volumi tra manuali ed eserciziari".²⁰

Questa applicazione potrebbe quindi essere utile agli studenti che si vogliono preparare per un test di ingresso, vedendo attraverso le correzioni dove approfondire il proprio studio. In questo caso d'uso ho raccolto alcuni quesiti dei test di ingresso degli anni passati, in maniera da poter dare un esempio concreto del funzionamento, ma è chiaro che la solita applicazione è applicabile a qualunque test su qualunque argomento.

L'applicazione potrebbe essere inoltre utilizzata, apportando piccole modifiche, non solo per un'autovalutazione, ma anche per una verifica telematica.

Per questo obiettivo, basterà fare in modo che venga messo in linea per un certo tempo.

Quello che il mio caso d'uso propone, dunque, sono una serie di quesiti suddivisi in tre sezioni, che riguardano: comprensione del testo, grammatica e conoscenze disciplinari, a cui lo studente o qualsiasi altro utente dovrà rispondere.

In questo caso, il sistema riceve le risposte dell'utente, le analizza raccogliendo i test che non hanno avuto esito positivo, cioè a cui l'utente non ha risposto correttamente in maniera da poterle mostrare alla fine del test.

²⁰ (6000 Quiz i quesiti delle prove di ammissione, lauree triennali professioni sanitarie, Seconda edizione, Alpha Test s.r.l)

Proprio alla fine dei test della sezione quindi, viene mostrato un calcolo in percentuale delle risposte corrette, una valutazione che va da insufficiente a ottimo ed un link da seguire dove verranno mostrate le risposte errate della sezione.

La pagina delle risposte errate ha, ovviamente, l'obiettivo di mostrare all'utente dove ha sbagliato e di poter apprendere dai propri errori. In questa sezione viene mostrato il numero del quesito, il numero del brano (nel caso in cui sia la sezione relativa alla comprensione del testo), cosa l'utente ha risposto, la domanda ed infine la correzione.

4.3.2. Definizione Database

Il database di questo caso d'uso differisce in alcune cose rispetto agli altri casi d'uso.

Questo per la necessità di dover dividere i test in più sezioni.

Nel caso della sezione "comprensione del testo", si deve inoltre specificare a quale brano fa riferimento il quesito per permettere all'utente di rileggere il test nel caso ne abbia bisogno.

Per la correzione degli errori ho inserito un campo di nome "Giusta" nel quale l'amministratore, durante l'inserimento dei quesiti, inserirà la risposta giusta, che sarà poi utile nella sezione dedicata alle correzioni.

Tabella Admin

Campo	Tipo	Indice	Auto_Increment
Id_test	INT	//	//
Test_Question	CLOB	//	//
Giusta	CHAR	//	//
Sezione	INT	//	//
Brano	INT	//	//
SottoCategoria	CHAR	//	//

Tab. 9 Tabella domande con aggiunta rispetto alle tabelle precedenti

PRIMARY KEY (id_test, Sezione)

I campi aggiunti quindi sono "Sezione", "Brano", "SottoCategoria" e "Giusta"

I primi due servono dunque per specificare a quale sezione la domanda appartiene e nel caso della comprensione del testo a quale brano.

Sono stati inseriti come interi in quanto la sezione viene memorizzata come un numero ed a ogni numero corrisponde una sezione differente, stesso discorso vale per il campo “Branò”.

Il campo “Giusta” come ho già anticipato ci è utile nella correzione, potendo così mostrare all’utente, una volta completato il test, la risposta che avrebbe dovuto dare.

Il campo “SottoCategoria” è utile per suddividere le sezioni in maniera che venga restituito un questionario omogeneo cioè con un numero di risposte uguali per tutte le sottosezioni.

La chiave primaria, questa volta è rappresentata dalla coppia “id_test, Sezione” in maniera da specificare che ci sia solo un “id_test” per una determinata sezione.

Anche nella tabella delle risposte, vengono aggiunti tre campi che corrispondono alla sezione, sotto categoria e all’eventuale brano, che faranno riferimento ai rispettivi campi “Sezione”, “Branò”, “SottoCategoria” della tabella **Admin**

La tabella delle risposte quindi si presenta così:

Tabella **RispostaUtente**:

Campo	Tipo	Indice	Auto_Increment
Id_testUtente	INT	FOREIGN KEY (reference Admin.id_test)	//
Answer	CLOB	//	//
NickNameRisposta	CHAR	//	//
SezioneRisposte	INT	FOREIGN KEY (reference Admin.Sezione)	//
SottoCategoriaRisposte	CHAR	FOREIGN KEY (reference Admin.Sottocategoria)	
BranòRisposte	INT	FOREIGN KEY (reference Admin.Branò)	//
Data	DATE	//	//

Tab. 10 Tabella delle Risposte con aggiunta rispetto alle precedenti

PRIMARY KEY(id_testUtente, NickNameRisposta, Answer)

Riguardo a questa tabella c'è poco da aggiungere in quanto è abbastanza chiaro il funzionamento.

Come negli altri casi d'uso viene memorizzata la risposta dell'utente ma con la particolarità delle sezioni, sotto categorie e dei brani.

La chiave primaria viene rappresentata dai tre campi "Id_testUtente", "NickNameRisposta", "Answer" in modo da poter garantire che per un certo quesito ci sia una sola risposta di un determinato utente. Per quanto riguarda la tabella dell'utente(v. tab. 7) invece non c'è molto da dire dato che è uguale agli altri casi d'uso.

4.4. Indagine di mercato per apertura di una libreria.

4.4.1. Definizione Obiettivo.

Questo caso d'uso è stato creato per effettuare un'indagine di mercato.

L'applicazione creata è in grado di recuperare dati utili, dalle risposte degli utenti, per l'apertura di una libreria.

I dati utili in questione potrebbero essere ad esempio sapere un autore preferito degli utenti o il genere preferito o ancora il libro preferito.

L'indagine svolta, è una classica indagine di mercato, in cui vengono poste all'utente domande base e domande specifiche per il raggiungimento dell'obiettivo stabilito.

L'obiettivo finale quindi è di riuscire a capire le preferenze degli utenti in ambito letterario, oppure per verificare quale servizio aggiuntivo vorrebbero gli utenti, per orientare l'attività in quella direzione.

È ovvio, anche in questo caso, che l'applicazione è utilizzabile per qualsiasi tipo di attività.

4.4.2. Definizione Database

Il database non differisce da quello adottato per l'indagine sui Social Network.

Abbiamo sempre la tabella **Admin** contenente le domande(v. tab.2), la tabella **RispostaUtente** (v. tab. 8) contenente le risposte di ogni utente, la data della risposta (utile alle indagini mensili) e l'utente che ha risposto, la tabella degli **utenti** (v. tab.7), la tabella delle **opzioni di risposta** (v. tab. 5) e la tabella dell'accesso dell'**amministratore** (v. tab. 6).

5. Conclusioni

In questo lavoro si è cercato di progettare un sistema che possa essere utilizzato su più fronti, che possa essere utilizzato sia da aziende per effettuare le proprie indagini di mercato via web, sia da docenti che vorrebbero effettuare dei test telematici o offrire un servizio di preparazione ai test di ingresso agli studenti ed infine offrire una alternativa al questionario didattico cartaceo che ogni anno viene posto agli studenti.

Nei diversi casi d'uso mostrati e nei relativi database, si è potuto notare come con piccole variazioni il sistema cambia atteggiamento.

Gli obiettivi che i casi d'uso di indagine di Social Network e di mercato per una libreria pone sono obiettivi che possono essere raggiunti nel tempo, dopo che verranno messi in rete dovremo aspettare che un numero convincente di utenti effettui il questionario per avere un quadro più completo possibile.

Nel caso d'uso del test di autovalutazione invece l'obiettivo è raggiungibile sin da subito poiché non appena un utente si collega e completa i test ottiene i risultati e l'obiettivo è immediatamente raggiunto mentre nel questionario della didattica dobbiamo attendere solo che gli studenti effettuino il test.

Quindi a seconda dei vari casi d'uso ci sono tempistiche differenti per il raggiungimento degli obiettivi prestabiliti.

“In generale un campione troppo piccolo(..) espone ad errori(oltre il 10%) normalmente non accettabili, in qualche caso possono andare bene già circa 500 casi, normativamente vanno bene mille casi(...)”.²¹

Nonostante il sistema sia stato progettato per essere utilizzato per più casi d'uso, rimangono comunque alcune questioni di difficile risoluzione: in un questionario in cui abbiamo bisogno di una domanda a risposta aperta, rimane il problema dell'estrapolazione automatica da parte della macchina dei dati, in questo caso quindi ci dovrà essere anche una forte presenza umana nello studio complessivo del questionario.

La flessibilità del sistema è data, come ho già detto, da piccole modifiche che ho apportato al codice, sia codice CSS che codice PHP.

Pur essendo piccole modifiche, questo potrebbe ovviamente creare problemi ad utenti non esperti nel campo, dunque potrebbe essere utile potere, in futuro, sviluppare comunque una interfaccia utente che permetta a chiunque, anche a chi non ha nessuna conoscenza informatica, di apportare le modifiche che vuole.

21 (http://cirdis.stat.unipg.it/files//Statistica_vita_quotidiana/conferenza_brusati.pdf, p.3)

Inoltre come ho detto, sia per motivi di comodità che per motivi di tempo, è stata presa la decisione di inserire le domande dei vari questionari, tramite script, anche questo potrebbe creare problemi ad alcuni utenti, dunque anche in questo caso potrebbe comunque essere utile fornire all'utente inesperto o che lo ritiene necessario, un'interfaccia per l'inserimento delle domande.

Proprio per questo l'interfaccia, sia per la modifica “esteriore” che di codice, dovrebbe essere intuitiva e che riesca a semplificare il lavoro ai “non esperti”.

Sono sempre più presenti sistemi di questo tipo on-line, tuttavia quello che il mio sistema offre è un immediato calcolo delle statistiche : le aziende o i docenti potrebbero aver bisogno immediatamente dei dati estrapolati, senza una verifica ulteriore, inoltre è un codice open source, “termine inglese che significa Codice sorgente aperto, indica un software i cui autori (...)ne permettono, anzi ne favoriscono il libero studio e l'apporto di modifiche da parte di altri programmatori indipendenti”.²²

Infine il sistema offre sia una possibilità di scelta di come porre le domande a seconda della necessità dell'amministratore che può ritenere necessario dover stabilire che prima vengano poste le domande con meno risposte oppure ritenere opportuno che vengano poste inizialmente le domande con il maggior numero di risposte differenti, sia da l'opportunità agli utenti di non finire il questionario o il test e poterlo continuare in un secondo momento, senza perdere i dati fino ad allora registrati.

22 (Wikipedia, voce Open Source)

6. Bibliografia

Rampichini, Carla. *Introduzione alla costruzione di un questionario*,

Università di Firenze.

http://www.ds.unifi.it/concorsoscuole/materiale/intro_al_questionario.pdf

Castledine, Earle, e Craig Sharkie, 2011. *Jquery Guida completa*, Milano, Apogeo.

Lecky-Thompson, Ed , Steven D.Nowicki, Thomas Myer, 2009. *PHP6 Guida per lo sviluppatore*, Milano, Ulrico Hoepli Editore

Petrossi, Furio. *Progettiamo correttamente un TEST di verifica*

<http://www.exelearning.it/>

Kurose, F. James, Ross, W. Keith , 2008. *Reti calcolatori e internet un approccio Top Down*, Milano, Bruno Mondadori S.p.A.

Fortini, Marco , 2000. *Linee guida metodologiche per rilevazioni statistiche*,

<http://www.istat.it/it/files/2010/09/lineeguida.pdf>

Garetto, Maria, 2002. *Statistica lezione ed esercizi*,

http://www.dm.unito.it/quadernididattici/garetto/quaderno_statistica.pdf

MacIntyre, Peter, Brian Danchilla, Mladen Gogala, 2012. *PHP per professionisti*, Milano, Apogeo.

Stampato da New Press S.r.l , 2012. *6000 Quiz i quesiti delle prove di ammissione, lauree triennali professioni sanitarie*, Seconda edizione, Alpha Test s.r.l

Il processo di costruzione di un questionario,

<http://www.urp.it/Sezione.jsp?idSezione=922&idSezioneRif=62>

Come fare una ricerca di mercato

<http://www.formaper.it/sociale2/ricerca/ricerca.html>

Quali requisiti deve avere un sondaggio d'opinione scientifico?

http://cirdis.stat.unipg.it/files/Statistica_vita_quotidiana/conferenza_brusati.pdf

7. Appendice

7.1. Guida per l'amministratore

In questa ultima parte vorrei offrire una guida dettagliata, per l'inserimento dei dati, all'amministratore.

Come ho detto in precedenza è stato deciso di inserire nel database la porzione di codice HTML relativa alla domanda.

L'inserimento viene fatto tramite script per il motivo principale che dopo aver annotato manualmente le porzioni, si possono modificare o aggiungere domande a piacimento, con il semplice uso di un file, contenente varie Query per varie domande.

Effettivamente inizialmente potrebbe disorientare questo metodo, ma una volta capito il metodo di inserimento, è molto più comodo e rapido di una interfaccia grafica.

Si può partire specificando come sono fatti i form.

Ogni caso d'uso ha una sua particolarità strettamente legata all'obiettivo da raggiungere.

Nel caso del questionario della didattica l'obiettivo come è stato già detto è di permettere agli studenti di dare una votazione per i diversi servizi.

Prendendo per esempio la prima domanda:

```
$query = (  
    'INSERT INTO Admin (Test_Question)  
    VALUES ("

## 

<form id=ins method=POST action=rispondi.php >  
<input type=radio value=1 name=scelta />frequenza saltuaria <br/>  
<input type=radio value=2 name=scelta />frequenza <50% <br/>  
<input type=radio value=3 name=scelta />frequenza 50/75% <br/>  
<input type=radio value=4 name=scelta />frequenza >75% <br/>  
<input type=submit name=invia />  
</form>") '  
);
```

Come è possibile notare, l'inserimento della domanda avviene semplicemente tramite Query.

Specificherò ogni passo da seguire e cosa significa:

\$query è semplicemente un nome dato ad una domanda, ovviamente se il sistema ha più domande questo deve variare ad esempio inserendo \$query2, \$query3 ecc.

```
INSERT INTO Admin (Test_Question)
```

questa riga di codice permette di inserire nella tabella “**Admin**” e nel campo “**Test_Question**” il valore che sotto viene specificato.

Quindi in **VALUES** (“ ”) ci andrà la porzione di codice html che si vuole inserire nel Database.

In **<h2></h2>**ci andrà l'effettiva domanda, ovvero cosa vogliamo chiedere all'utente nell'esempio sopra:

```
<h2>Presenza dello studente alle lezioni</h2>
```

La parte che segue corrisponde alle risposte proposte dal sistema:

```
<form id=ins method=POST action=rispondi.php >
<input type=radio value=1 name=scelta />frequenza saltuaria <br/>
<input type=radio value=2 name=scelta />frequenza <50% <br/>
<input type=radio value=3 name=scelta />frequenza 50/75% <br/>
<input type=radio value=4 name=scelta />frequenza >75% <br/>
<input type=submit name=invia />
</form>
```

Le risposte proposte vengono registrate come **<form>** in maniera da poter inviare i risultati al database.

Al form devono essere assegnati tre attributi:

1. id =ins;
2. method=POST;
3. action=rispondi.php.

Per chi conosce html può sembrare strano non aver inserito le virgolette, ma inserendo questi dati tramite Query in SQLite, non c'è bisogno di scrivere ad esempio id=”ins”, ma bisogna scrivere **id=ins**, poiché inserendo le virgolette da errore.

L'attributo “**method**” serve per specificare in che modo i dati devono essere inviati, ci sono due metodi **GET** e **POST** , noi utilizzeremo il secondo per motivi di spazio dato che il metodo GET non può passare più di 255 caratteri, e le nostre domande potrebbero essere più lunghe.

L'attributo “**Action**” specifica dove, dopo aver inviato i dati, il form deve portare, nel mio sistema è importante che il form riporti esattamente nel punto in cui vengono proposte le domande, nel caso del questionario didattica le domande vengono mostrate nel file “**rispondi.php**” ed è qui che il **<form>** mi deve reindirizzare.

Sotto questa riga si possono notare i vari input che sono le effettive risposte come ad esempio:

```
<input type=radio value=1 name=scelta/>frequenza saltuaria<br>
```

Quello che c'è da capire è sostanzialmente poco:

1. **input type=radio** : come prima non bisogna inserire “radio” ma inserire senza virgolette
2. **radio** : questo obbliga l'utente a fare una scelta per domanda ovvero, essendo a risposta multipla l'utente può scegliere, in questo caso, una ed una sola opzione, a differenza della CheckBox che permette all'utente di fare più scelte;
3. **value=1** : come prima non bisogna inserire “1”. Qui inseriamo il vero valore che vogliamo registrare nel Database, questo è un valore arbitrario e starà all'amministratore decidere il valore dell'input. Come abbiamo detto prima, volendo inserire una votazione questo corrisponde ad un voto;
4. **name=scelta**: è molto importante non cambiare il valore dell'attributo name, in quanto il sistema è stato programmato per prendere i valori degli input con questo attributo ed è quindi fondamentale lasciarlo invariato;
5. **frequenza saltuaria**: questo è quello che verrà visto dall'utente al momento della domanda;
6. **
** : si chiude l'input e con il tag
 si permette all'input successivo di andare a capo e quindi essere visualizzato sotto dall'utente;
7. Ricapitolando quello che l'utente può modificare è Value=1, Frequenza saltuaria.

Dopo avere inserito gli input che si vuole, ovvero le risposte proposte è importante inserire il bottone di invio e la chiusura de form:

```
<input type=submit name=invia />
</form>") '
);
```

Una volta inserita e completata la Query come sopra ho spiegato, l'amministratore non deve far altro che andare in fondo alla pagina e inserire.

```
sqlite_query ($query , $database );
```

Con questo codice viene definitivamente inserita.

Ovviamente “**\$query**” è semplicemente una variabile e quindi cambia a seconda della domanda che si vuole inserire.

Nel file “**Inserimento.php**” potrebbe essere utile sapere cosa è la riga di codice:

```
include("database.php");
```

con questa riga avviene la connessione al database.

Include è una funzione PHP che serve per includere dei file a cui abbiamo lavorato, “i file sono inclusi in base al percorso determinato(..)”²³.

²³ (<http://php.net/manual/en/function.include.php>)

Nei casi d'uso che riguardano le indagini di mercato della libreria e sui Social Network o comunque in un qualsiasi caso in cui non servirà avere dei valori numerici per effettuare delle votazioni il procedimento è simile ma differisce in alcune parti.

Prendendo ad esempio una domanda dell'indagine sui Social Network.

```
$query3 = (
  'INSERT INTO Admin (Test_Question)

  VALUES ("

## 


```

Come avevo anticipato il principio è lo stesso le cose cambiano sono essenzialmente gli attributi value degli input, l'attributo type e name sempre degli input.

```
<input type=checkbox value=Facebook name=scelta[] />
```

L'attributo type viene modificato da radio a checkbox, in maniera che l'utente possa scegliere più opzioni, come ad esempio in questo caso un utente potrebbe utilizzare più di un Social Network e quindi potrebbe volere selezionare più di un'opzione.

Varia in questo caso anche l'attributo name che mentre nel caso precedente era:

scelta, adesso è **scelta[]**.

Come nel caso precedente le “ “ non bisogna inserirle, ma in questo caso sono importanti le parentesi quadre accanto all'attributo name : **scelta[]**.

Questo è importante perché permette al sistema di prendere, se selezionata, più di una risposta e inserirle nel database.

Ultima cosa da mettere in evidenza in questo caso d'uso è ad esempio un input fatto in questa maniera:

```
<input type=checkbox value=libri&nbsp;cartacei name=scelta[]/>Preferisco i
libri cartacei
```

Ho messo in evidenza un input di questo tipo perché come si può notare nell'attributo value appare una entità ** **: “le entità (in inglese entity) sono una codifica testuale usata per inserire alcuni caratteri speciali in maniera indipendente dalla tastiera e dal sistema operativo usato”²⁴.

²⁴ (Wikipedia, voce Entità markup)

Questa entità va utilizzata esclusivamente nell'attributo value quando il valore che vogliamo dare all'input contiene uno spazio, per ogni spazio, al posto di esso si dovrà inserire ** **; ad esempio:

```
<input type=checkbox value=si,ho&nbsp;utilizzato&nbsp;Ebook name=scelta[]/>
Si, ho utilizzato almeno un Ebook
<br/>
```

Nell'ultimo caso d'uso ovvero quello relativo al test di autovalutazione, cambia il database quindi cambia un leggermente l'inserimento.

L'inserimento di questo caso d'uso cambia a seconda della sezione in cui si dovrà inserire la domanda, se è di comprensione del testo si dovrà inserire la sezione e il brano corrispondente, mentre se non fa parte della comprensione del testo basterà inserire la sezione della domanda.

Prendendo questo esempio:

```
$queryBrano14 = (
  'INSERT INTO Admin (id_test, Test_Question, Giusta, Sezione, Brano)
VALUES ("4","<h2>Quale tra queste attività era, verosimilmente, quella
dell'autore?</h2>
<form class=ins method=POST action=DefaultComprensione.php?idBrano=1>
<input type=radio value=Sbagliata,Giurista name=scelta[]/>Giurista
<br/>
<input type=radio value=Sbagliata,Ingegnere name=scelta[]/>Ingegnere
<br/>
<input type=radio value=Giusta,Filologo name=scelta[]/>Filologo
<br/>
<input type=radio value=Sbagliata,Avvocato name=scelta[]/>Avvocato
<br/>
<input type=submit name=invia />
</form>',"Filologo", 1, 1)'
);
```

Come ho detto, bisogna inserire l'id del test, la domanda, la risposta giusta, la sezione e il brano.

```
INSERT INTO Admin (id_test, Test_Question, Giusta, Sezione, Brano)
```

Ogni campo deve essere inserito, importante è che si mantenga l'ordine di inserimento, nel caso precedente è importante che prima si inserisca il campo "id_test" poi "Test_Question" poi "Giusta" ecc.

Nel caso precedente si inserisce l'id test con il valore 4, in maniera che questa sia la quarta domanda di quella sezione per quel brano.

Quello che cambia nel `<form>` è l'attributo "action" che nel caso della sezione della comprensione del brano è :

```
"DefaultComprensione.php?idBrano=1"
```

Importante è inserire "**?idBrano=1**" che corrisponde al numero del brano a cui si riferisce la domanda, se per esempio il brano di riferimento è il numero 2 basta cambiare "**?idBrano=1**" in "**?idBrano=2**".

Altra cosa che varia è l'attributo "value", quello che bisogna aggiungere è "**Giusta,**" o "**Sbagliata,**" mantenendo le maiuscole e la virgola vicina, attaccato alla risposta:

```
value=Sbagliata,Avvocato  
o  
value=Giusta,Filologo
```

il resto degli inserimenti cioè : "**Filologo, 1, 1**" sono riferiti alla risposta giusta, al numero della sezione che in questo caso è 1 e al numero del brano di riferimento.

Ogni sezione ha un intero che la definisce, ma questo non modifica il form in quanto nei casi differenti dalla comprensione la stampa delle domande viene gestita da una pagina unica dinamicamente.

```
<form class=ins method=POST action=Conoscenza.php >
```

Se ad esempio dovessimo aggiungere una sezione, si dovrebbe creare una pagina di stampa della nuova sezione che differisce dalla precedenti solamente nella Query della selezione degli Id e nella stampa del link per la pagina delle correzioni. Basta inserire nella pagina "**Rispondi.php**", nella lista dei link alle sezioni:

```
<a href="Conoscenza.php?sezione=3"></a>
```

dove **3** è il numero della nuova sezione che vogliamo stampare, mentre nel link delle verifiche :

```
<a href='Verifica.php?Sezione=3'>Verifica </a>
```

Il valore 3 è il valore della nuova sezione.

Ovviamente negli inserimenti del file "**Inserimento.php**" dovremo inserire una nuovo intero che identifichi la nuova sezione.

Cambia leggermente nel caso delle altre sezioni in quanto le domande non vengono poste una dopo l'altra ma tutte insieme in un form solo, quindi :

```
$queryGrammatica3 = (  
'INSERT INTO Admin (id_test,Test_Question, Giusta, Sezione, SottoCategoria)  
VALUES ("3",  
"<h2>Nella serie riportata di seguito, una sola parola rispetta  
l'ortografia italiana standard. Di quale parola si tratta?</h2>  
<input type=radio value=Sbagliata,perquotere name=scelta[3][]/>  
perquotere<br/>  
<input type=radio value=Sbagliata,accuarello name=scelta[3][]/>  
accuarello <br/>  
<input type=radio value=Giusta,propinquo name=scelta[3][]/>  
propinquo<br/>  
<input type=radio value=Sbagliata,cospigo name=scelta[3][]/>  
cospigo<br/>  
<input type=radio value=Sbagliata,oblicuo name=scelta[3][]/>  
oblicuo<br>", "propinquo", 3, "Grammatica")'  
);
```

Come si può notare in questo caso, che è il caso generale delle diverse sezioni differenti da quelle del brano, non viene inserito nel database la classica riga di codice per l'apertura e la chiusura del form e nemmeno il bottone di “**submit**”, perché questi vengono inseriti dal sistema al momento della creazione casuale dei test.

Le differenze importanti sono:

1. INSERT INTO Admin((...), **SottoCategoria**) in quanto bisogna inserire a quale sotto categoria si riferisce la domanda;
2. attributo “**name**” dell'input: **name scelta[x][i]** dove x sta per il numero del test. Nell'esempio precedente il test è il numero 3 della sezione numero 3 per la sotto categoria Grammatica.