

UNIVERSITÀ DEGLI STUDI DI TRENTO

Facoltà di Scienze Matematiche, Fisiche e Naturali.



Corso di laurea triennale in Informatica

Elaborato finale

QUAESTIO: UN'APPLICAZIONE WEB PER IL MONITORAGGIO DELLA SODDISFAZIONE DEI CLIENTI.

Relatore:
Prof. Maurizio Marchese

Laureando:
Stefano Tranquillini

Correlatore:
Dott. Luigi Perna

Anno Accademico 2006/2007

SOMMARIO

1	Introduzione	1
1.1	Il questionario	1
1.2	Le richieste.....	2
1.3	Il nome Quaestio	3
2	Stato dell'arte	5
2.1	Introduzione	5
2.2	Java Enterprise Edition	6
2.2.1	Containers.....	8
2.2.2	Middle-tier	8
2.2.3	J2EE e J2SE.....	11
2.3	Il server: Tomcat	11
2.4	Il Database	12
2.4.1	Postgres.....	12
2.4.2	IBatis.....	14
2.5	Struts – MVC	15
2.5.1	Il design pattern MVC	16
2.5.2	I componenti di Struts	16
2.5.3	Struts e MVC	18
2.5.4	Vantaggi	19
2.6	Autenticazione e Autorizzazione.....	19
2.6.1	Realm.....	20
2.6.2	LDAP.....	20
2.7	Report, JaseperReport e Ireport	21
2.8	Ajax.....	22
3	La definizione dell'applicazione	25
3.1	Utenti	25
3.2	Requisiti	28
3.2.1	Funzionali	28
3.2.2	Informazioni	31
3.2.3	Requisiti non funzionali e Architetture	33
4	Introduzione sviluppo dell'applicazione	35
4.1	Introduzione	35
4.2	L'architettura	36

4.3	Le classi dell'applicazione.....	36
5	Il database.....	41
5.1	Questionario.....	41
5.2	Modulo.....	42
5.3	Domanda.....	42
5.4	Risposta.....	43
5.5	La gestione del multilingua.....	43
5.6	La gestione modulare.....	44
6	Il funzionamento dell'applicazione.....	47
6.1	Utente.....	47
6.1.1	Approfondimento funzionamento	49
6.2	Amministrazione.....	51
6.2.1	Approfondimento funzionamento	56
7	Interfaccia dell'applicazione	59
7.1	Amministratore	59
7.2	Utente.....	63
8	Manuale d'uso.....	65
8.1	Introduzione	65
8.2	Installazione e configurazione	65
8.3	Uso: Amministrazione	67
8.4	Uso: Utente	78
8.5	Aggiunta componenti.....	78
9	Conclusioni	81
10	Bibliografia	83

1 INTRODUZIONE

Questa tesi nasce dall'esigenza dell'azienda Enginsoft¹ di implementare un sistema di gestione Web dei questionari dei vari eventi organizzati. L'applicazione qui sviluppata ha l'intento di sostituire il classico questionario cartaceo con uno elettronico, da completare via web, così da ridurre la mole di lavoro sia nella fase realizzativa che nella raccolta ed analisi dei dati.

1.1 IL QUESTIONARIO

Dal dizionario²

que|stio|nà|rio

1 serie di domande o quesiti su determinati argomenti che viene proposta a una o più persone, spec. per valutare capacità, conoscenze, attitudini e aspirazioni personali o per raccogliere dati e opinioni da utilizzare in indagini di tipo statistico:

preparare un q. | serie di quesiti a scopo didattico: un q. di inglese, di matematica

2 foglio o modulo che riporta tali domande e quesiti: riempire un q., distribuire i questionari.



Figura 1 Struttura del questionario.

Il questionario, come noi tutti lo conosciamo, è un insieme di domande e risposte (aperte o meno) generalmente presentate su una carta.

Il passaggio dalla versione cartacea a quella web ha comportato la seguente strutturazione del questionario:

- Questionario
 - Modulo
 - Domande e risposte.

¹ © <http://www.enginsoft.it>

² Dizionario Demauro - <http://www.demauroparavia.it/89653>

Il Modulo è il componente che in un questionario cartaceo non esiste o non si nota; la sua struttura è caratterizzata da un titolo ed è, in sostanza, un raccoglitore di domande. Ogni modulo può essere utilizzato in uno o più questionari, questo permette di avere dei moduli “standard” da usare in vari questionari senza doverli riscrivere.

Le domande sono strutturate come un contenitore, ma in questo caso, di risposte. Le domande fanno riferimento al modulo che le contiene e non al questionario. Vale a dire che per inserire domande standard in un questionario si dovrà inserire il modulo che le contiene.

1.2 LE RICHIESTE.

L'applicazione doveva soddisfare alle richieste dell'azienda, in particolare:

- **Gestione modulare del questionario:** Il questionario deve essere diviso per moduli in modo da poter riutilizzare un singolo modulo in più questionari. Deve anche permettere l'utilizzo di un set di risposte standard così da minimizzare i tempi di costruzione.
- **Multilingua:** L'applicazione deve permettere la creazione di questionari in più lingue. Questo comporta di mantenere la stessa forma del questionario modificandone il contenuto.
- **Utilizzo del sistema esistente:** l'applicazione deve sfruttare il database esistente nell'azienda ed integrarsi con esso.

Le caratteristiche verranno esaminate in dettaglio più avanti.

La scelta di un'applicazione distribuita è dovuta a vari motivi, i principali sono:

- **Facilità di distribuzione e aggiornamento:** un'applicazione Web si trova interamente sul server, per cui la pubblicazione sul server coincide con la distribuzione e l'aggiornamento effettuato sul server è automaticamente reso disponibile a tutti gli utenti;
- **Accesso multiplatforma:** l'accesso all'applicazione è indipendente dall'hardware e dal sistema operativo utilizzato dagli utenti;
- **Riduzione del costo di gestione:** l'uso di Internet come infrastruttura per un'applicazione Web riduce notevolmente sia i costi di distribuzione che i costi di gestione dei clienti;

- **Scalabilità:** un'applicazione Web ben progettata può crescere insieme alle esigenze dell'azienda senza particolari problemi.

La scelta di Java come linguaggio è dovuta principalmente alla sua diffusione sia in ambiente universitario che lavorativo.

1.3 IL NOME QUAESTIO

In greco antico, la funzione di contrassegnare una domanda, espressa oggi col punto interrogativo, era demandata ad un punto e virgola ";". Nel corso dei secoli se ne perse l'utilizzo, e per tutta l'età antica non si utilizzarono segni particolari per esprimere l'intonazione interrogativa. Il punto interrogativo vero e proprio nacque nel Medioevo, all'epoca dei monaci copisti: essi solevano, per indicare le domande, scrivere alla fine delle frasi la sigla qo, che stava per quaestio (dal latino, domanda). Per evitare di confondere questa sigla con altre si cominciarono a scrivere le due lettere l'una sull'altra e a stilizzarle, mutando la Q in un ricciolo e la O in un punto ottenendo così il punto di domanda (?) come lo conosciamo.

Da qui la scelta di chiamare l'applicativo Quaestio – QO.(1)

2 STATO DELL'ARTE

2.1 INTRODUZIONE

Il World Wide Web (WWW) è una rete di risorse di informazioni basata sull'infrastruttura di Internet. E' un'architettura hardware e software che permette di accedere alle informazioni attraverso una rete alla quale l'hardware è collegato. Per fare questo, sfrutta principalmente tre componenti:

- Protocollo: HTTP.
- Ipertesto per la navigazione: HTML.
- Metodo di localizzazioni delle risorse: URL.

L'architettura software del web, invece, si distingue in due grandi classi:

- Client: un qualunque dispositivo che si connette alla rete e comunica attraverso essa con tutti gli altri dispositivi connessi.
- Server: ha il compito di ricevere le richieste del client e fornire le opportune risposte.

Un utente che visita una pagina Web in internet, non fa altro che collegarsi ad un computer sul quale risiede il file che rappresenta quella pagina e lo visualizza. Il client quindi si connette ad un server. Il software che permette al server di rispondere è chiamato Web Server e fornisce solamente gli strumenti per accettare e processare la richiesta ed inviare la risposta che è memorizzata su file.

Quando si ha la necessità di realizzare applicazioni più complesse si deve utilizzare uno strumento più evoluto qual è l'Application Server. L'Application Server permette di rispondere alle richieste attraverso programmi che risiedono e vengono eseguiti sul server.

Quindi la differenza tra Web e Application server sta nel fatto che il primo gestisce le richieste di pagine statiche mentre il secondo gestisce le richieste di pagine create dinamicamente.

Esistono numerose tecnologie che permettono l'implementazione di tutto questo; la più diffusa, che è stata utilizzata in questo progetto, è la J2EE. Per lo sviluppo si è utilizzato

un server Web e un Application server come Tomcat che può egregiamente svolgere entrambi i ruoli.

2.2 JAVA ENTERPRISE EDITION

La scelta del linguaggio di programmazione è caduta su Java, in particolare su J2EE o come viene chiamata ora: Java Enterprise Edition.

La tecnologia JEE o J2EE è diventata una tra le più importanti piattaforme di sviluppo di applicazioni aziendali, grazie soprattutto alla sua affidabilità, sicurezza e robustezza. Gran parte del suo successo è dovuto al fatto che questa tecnologia è sviluppata sul linguaggio object oriented Java, con tutti i vantaggi che ne derivano in fatto di funzionalità, facilità di programmazione e espansione.

Questa tecnologia è stata sviluppata con l'intento di creare delle applicazioni Enterprise che soddisfacessero le esigenze di affidabilità e robustezza che devono garantire le applicazioni distribuite aziendali.

J2EE al suo interno contiene molti servizi che poi andranno ad implementare i vari strati dell'applicazione:

- **Web Application:** si tratta delle tecnologie legate alla produzione di interfacce web dinamiche (ad esempio JSP XML, JSF.)
- **Enterprise Application:** le tecnologie legate alla logica di business cioè allo sviluppo vero e proprio come Enterprise JavaBean, Jndi, JavaMail.
- **Web Services:** le tecnologie utili allo sviluppo delle applicazioni orientate ai servizi.
- **Management and Security:** le tecnologie legate alla gestione del sistema Enterprise per realizzare lo scambio e l'accesso delle informazioni tra tutti i servizi distribuiti.

Le applicazioni J2EE possono essere divise in tre strati (chiamati anche layer o tier):

- **Client Tier:** Costituisce il front-end che viene eseguito sul client dell'applicazione. Inoltra le richieste al server per conto dell'utente e ne presenta a quest'ultimo i risultati. La J2EE supporta vari tipi di client che possono connettersi sia dall'interno del firewall che dall'esterno.

- **Middle Tier:** è il livello centrale e si occupa della logica del sistema. Si divide in due sotto tier: il WEB-tier che gestisce i servizi per i contenuti dinamici come le Servlet e JSP e il EJB-tier che gestisce la logica di business attraverso gli EJB.
- **EIS Tier:** si occupa della gestione e della comunicazione con i dati memorizzati, principalmente database.

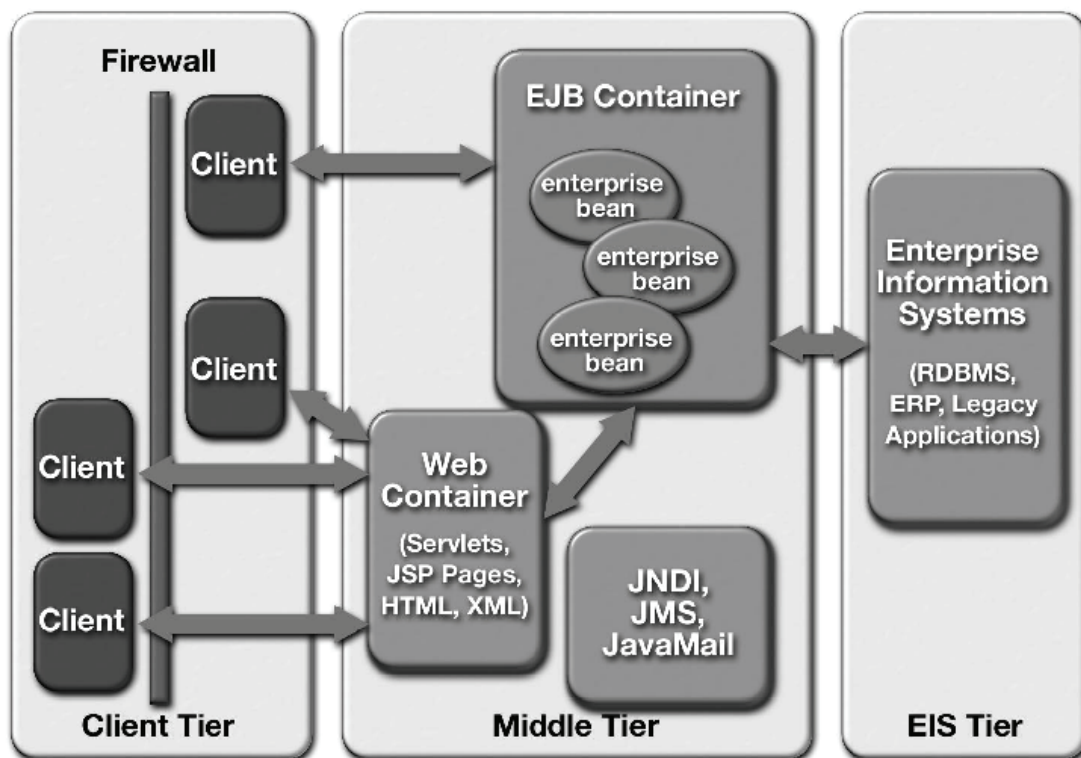


Figura 3 Architettura multilivello J2EE

La Figura 3 mostra i vari componenti e servizi che costituiscono l'ambiente J2EE. La presenza di un livello intermedio tra il client e i dati fornisce i tipici vantaggi di un'architettura multilivello. Le architetture client-server a due livelli, dove il server è costituito essenzialmente dalle sorgenti d'informazione, hanno dato prova della loro limitazione principalmente per la necessità di avere un'interfaccia grafica e una logica applicativa direttamente su ciascun client. Questo comporta un carico sulla macchina client, creando problemi derivanti sia dalle risorse limitate della macchina dei client, sia dall'onerosa manutenzione dell'applicazione: ogni cambiamento, anche piccolo, deve essere ripetuto su ogni client. Con un'architettura a tre livelli questi inconvenienti vengono superati grazie al middle tier che si occupa di gestire le richieste dei vari client,

di reperire le informazioni dall'EIS, rielaborare secondo la logica applicativa e in fine di fornirle al client eliminando così le problematiche dell'architettura cliente - server.

2.2.1 CONTAINERS

Prima di proseguire nella descrizione è essenziale introdurre un elemento chiave dell'architettura: i containers.

I containers sono delle interfacce standardizzate in grado di fornire specifici servizi ai vari componenti. Ci permettono di suddividere la logica applicativa dai servizi a basso livello che sono implementati dai container. Questi strumenti permettono al programmatore di disinteressarsi di come i container gestiscono i servizi per interessarsi esclusivamente del loro utilizzo all'interno del codice.

2.2.2 MIDDLE-TIER

Tra i tre tier quello più complesso è sicuramente il middle-tier. Al suo interno, infatti si sviluppa la logica dell'applicazione. Offre vari servizi come JNDI, JavaMail e in più si differenzia in WEB-tier e EJB-tier.

EJB-TIER

L'EJB-tier è la componente che gestisce la logica di business. Non è semplice definire cosa sia questa logica; in un certo qual modo si può dire che è un insieme di direttive per la gestione specifica della funzione che l'applicazione deve svolgere. All'interno di un contesto object-oriented si struttura in un insieme di componenti chiamati business-object. I business-object sono dotati di una struttura specifica e di un determinato comportamento. Si può allora definire la business-logic come l'insieme delle regole che servono ad identificare struttura e comportamento dei business-objects e i criteri di interazione con gli altri oggetti.

I requisiti più comuni dei business-object sono:

- Mantenere lo stato.
- Operare su dati condivisi.
- Partecipare alle transazioni.
- Servire più client.
- Fornire accesso remoto ai dati.
- Avere un controllo sugli accessi.

- Essere riusabili.

All'interno della J2EE i business-object sono rappresentati dai EJB.

All'interno dell'applicazione, gli EJB (Enterprise Java Bean) vengono presi in carica dall'EJB-container che gestisce il loro ciclo di vita e i servizi a basso livello. Questa separazione, che è presente anche nel WEB-tier, permette lo sviluppo di applicazioni altamente scalabili, flessibili e sicure.

WEB-TIER

Il web-tier si occupa della logica applicativa e della creazione dei contenuti dinamici.

Per questo la J2EE mette a disposizione vari strumenti tra i quali: Servlet e JSP che possono interagire tra loro in varie maniere e utilizzare altre componenti come gli EJB, JavaBean o altro.

SERVLET

Le Servlet sono degli oggetti Java, cioè dei pezzi di codice, che risiedono su un server e che offrono molteplici funzionalità.

Una Servlet ha la possibilità di accedere a tutte le risorse del sistema su cui risiede; questo permette l'implementazione di diverse funzioni che sfruttano, sia le caratteristiche del sistema, sia la potenza di un linguaggio ad oggetti come Java.

Esistono varie tipologie di Servlet a seconda del protocollo usato; per il nostro progetto vengono sviluppate solo le HttpServlet. Estendendo questa classe è possibile creare Servlet in grado di comunicare con i client e rispondere dinamicamente in base alle richieste.

L'idea che sta alla base del funzionamento della Servlet è di condividere un unico oggetto con tutti i clienti che ne richiedono i servizi. La condivisione di un unico oggetto permette, per esempio, di evitare le operazioni di creazione dell'oggetto stesso, della creazione di connessioni ai dati, e della loro successiva eliminazione alla fine del ciclo. In pratica, l'arrivo di richieste di un servizio si traduce nella creazione di un thread che concorrerà alla risorsa. Questo comporta un notevole aumento di performance dovuto alla diminuzione di utilizzo di risorse del sistema in quanto sono condivise da tutti gli utilizzatori.

Il funzionamento di una Servlet si può paragonare al modello client - server, dove ad una richiesta corrisponde una risposta adeguata. Questo paradigma viene sviluppato sfruttando `HttpServletRequest` e `HttpServletResponse`. La prima rappresenta la richiesta effettuata dal client del servizio, la seconda è la risposta che riceverà l'utente e che viene generata dalla Servlet.

JSP

La tecnologia Java Server Pages è una soluzione per realizzare pagine HTML dinamiche, lato server.

Le pagine JSP sono un'evoluzione delle Servlet Java e distinguono all'interno dello stesso file la rappresentazione dei dati dai contenuti. Si presenta come un documento HTML con al suo interno delle parti di codice Java sotto forma di tag. Si potranno quindi modificare le parti sviluppate in Java lasciando inalterata la struttura HTML o viceversa. Come per le Servlet, anche le JSP sono scritte nel linguaggio Java e quindi ne hanno tutti i vantaggi, in particolare: la robustezza, la portabilità e il fatto di essere multiplatform.

Il fatto di mescolare il codice Java all'HTML permette di modificare la presentazione dei dati in modo semplice e non invasivo: infatti, non è necessario conoscere il codice Java per modificare la forma della presentazione.

Una Java Server Page può essere richiamata principalmente in due maniere differenti. La richiesta può avvenire direttamente ad una pagina JSP, che grazie alle risorse lato server può elaborare i dati e generare l'output; in alternativa, la richiesta può essere fatta ad una Servlet che dopo un'elaborazione invierà i dati ad una JSP che avrà il compito di visualizzarli.

JSP vs SERVLET

La differenza tra una JSP e una Servlet è sottile ma non trascurabile, infatti, per come è strutturata la pagina JSP è composta sia di codice Java che codice HTML quindi è una specie di collante tra il layer presentation e Business logic. La Servlet, invece, pur presentando la possibilità di generare output è più orientata al controllo dell'applicazione cioè a far parte del layer Business logic.

JAVABEAN

Sono delle classi Java atte a contenere, o meglio incapsulare, i dati così da poterli trasportare. All'interno dell'applicazione sono state utilizzate sia per mappare i dati delle query, sia per scambiare i dati attraverso le pagine web con l'ausilio di Struts.

La peculiarità di queste classi è quella di avere degli attributi di diverso tipo e, per ogni attributo, avere un metodo setter e getter per accedervi a leggere o salvare i dati. Questo permette di disporre di un metodo di accesso standard ai dati che viene sfruttato anche dal framework Struts per riempire i Bean con i dati delle varie form.

2.2.3 J2EE E J2SE

La J2SE (Java 2 Standard Edition) è la versione Java che si usa normalmente per sviluppare le applicazioni. Le due piattaforme non sono mutuamente esclusive, anzi, la J2SE rappresenta la base su cui poggia la J2EE. Si può dire che la Standard Edition rappresenta l'ossatura su cui poggiano le componenti della versione Enterprise. Quindi per lo sviluppo di applicazioni J2EE è necessario avere una conoscenza della piattaforma J2SE.

2.3 IL SERVER: TOMCAT

Uno strumento essenziale per un'applicazione web è il web Server. Questo, deve rispondere alle richieste del client e quindi permettere di visualizzare le pagine Web. Il server web svolge principalmente tre compiti:

- Riconosce la richiesta.
- Cerca la pagina richiesta se presente.
- Invia la pagina al browser client che la visualizzerà.

Il Web Server non presenta nessuna logica di controllo e quindi ha la capacità di gestire unicamente pagine statiche. La nostra applicazione però, necessita di una funzionalità in più: il saper gestire e processare contenuti dinamici.

Lo strumento essenziale per fare tutto questo è l'Application Server, cioè un software residente su server che fornisce una serie di servizi; in questo caso, essendo l'applicazione in J2EE, mette a disposizione tutti i servizi della tecnologia Java Enterprise.

Esiste una gran quantità di Application Server, la scelta è ricaduta su Tomcat di Apache Software Foundation vista la semplicità di utilizzo, la stabilità e affidabilità nonché il fatto di essere gratuito. Oltre a questo non è da sottovalutare che dietro ai progetti della Apache Software Foundation c'è una grande comunità che lavora costantemente per migliorare i prodotti.

Ad essere precisi, Tomcat è un Servlet Container ed un JSP Engine, cioè un motore in grado di eseguire, lato server, applicazioni Web basate sulla tecnologia J2EE e costituite da componenti Servlet e da pagine JSP.

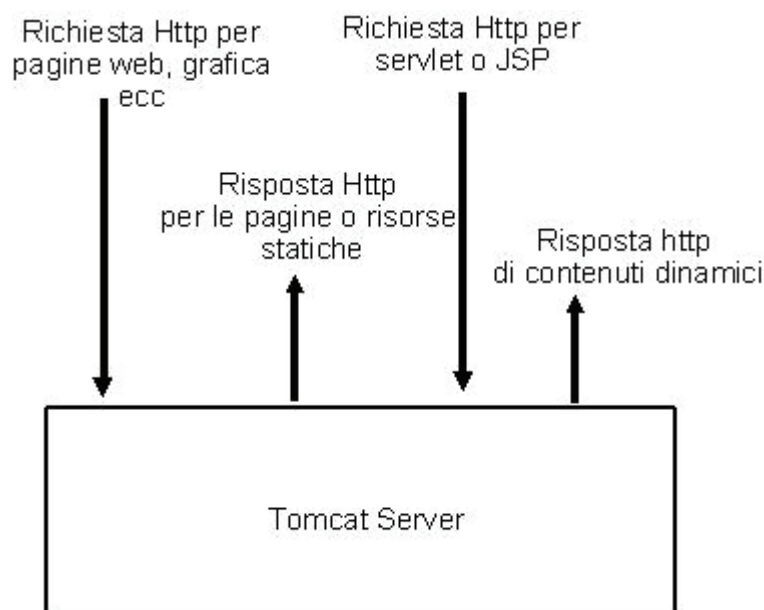


Figura 4 Tomcat in Standalone

Come si vede nella Figura 4 Tomcat nella versione Standalone, utilizzata nel progetto, funziona anche come web server. Non è necessario perciò avere un web server distaccato che gestisca le richieste.

2.4 IL DATABASE

2.4.1 POSTGRES

Un altro componente fondamentale all'interno delle applicazioni web è il database, cioè un archivio di dati. I dati contenuti nel database possono essere di diversa natura; nel nostro caso, riguardano principalmente gli utenti del sistema e i vari testi dei questionari e delle relative risposte. Per soddisfare alle richieste, il database dell'applicazione

doveva essere perfettamente integrato con quello già esistente in azienda, per questo è stato usato il DBMS (Data Base Management System) Postgres nella versione 8.

Un DBMS permette di:

- Fornire l'accesso dei dati
- Manipolare i dati tramite inserimenti, modifiche e cancellazioni
- Garantire la sicurezza dei dati
- Mantenere l'integrità dei dati
- Ottimizzare le performance

Postgres è un database relazionale, la cui struttura fondamentale è per l'appunto la relazione, cioè una tabella costituita da righe (tuple) e colonne (attributi). Gli elementi di queste relazioni vengono dette tuple e sono i valori dei vari attributi. Ogni tupla è identificata da una chiave (cioè un insieme di attributi) che deve avere la proprietà di essere univoca.

L'interazione tra un utente e la base di dati avviene secondo il modello client-server ("process-per-user"), coinvolgendo, per ogni connessione stabilita, tre processi:

- il postmaster: un processo daemon con funzione di supervisione;
- l'applicazione frontend dell'utente;
- un backend database server (per ogni connessione).

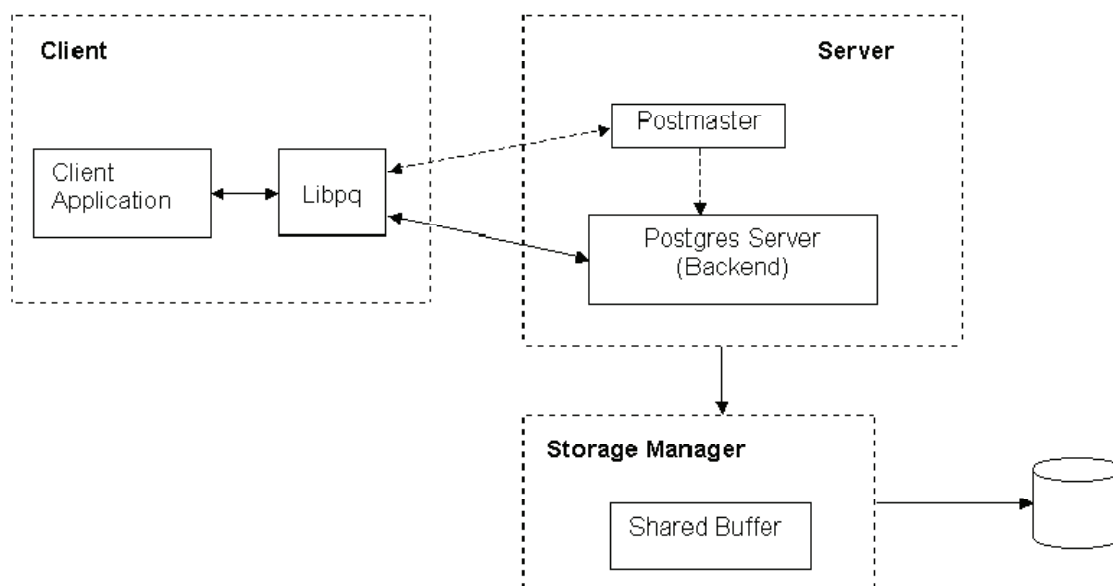


Figura 5 Architettura Postgres

Come si può vedere dalla Figura 5, le applicazioni front-end che desiderano accedere al database comunicano attraverso le librerie libpq. Libpq è responsabile della comunicazione tra il client e i processi; stabilisce una connessione con il postmaster e ottiene dal Postgres un server thread per eseguire le operazioni. Il server è composto da due sotto-sistemi: il postmaster e il Postgres server. Il primo è responsabile delle comunicazioni entranti dal client, esegue l'autenticazione e il controllo dell'accesso del client e ne permette la comunicazione con il server. Come detto la libreria invia le richieste di accesso al postmaster, che risponde attivando un nuovo processo server di backend e collegandolo al processo di frontend: da questo momento in poi frontend e backend continuano a comunicare senza ulteriori interventi del postmaster. Il postmaster, dopo aver portato a termine la fase iniziale di connessione tra frontend e nuovo backend, si mette in attesa di nuove richieste.

L'accesso a un database coinvolge quindi un'applicazione frontend, backend e postmaster ed è del tipo "un processo per utente".

2.4.2 IBATIS

Una volta creato il database, si ha la necessità di poterlo utilizzare all'interno dell'applicazione. Per interfacciarsi è stato deciso di usare il framework iBatis. Java mette a disposizione una serie di librerie (JDBC) per la gestione del proprio database. Purtroppo questo metodo non si adatta molto alla programmazione ad oggetti ma per questo ci viene d'aiuto iBatis che permette di mascherare lo stile relazionale del database con quello orientato agli oggetti del nostro linguaggio.

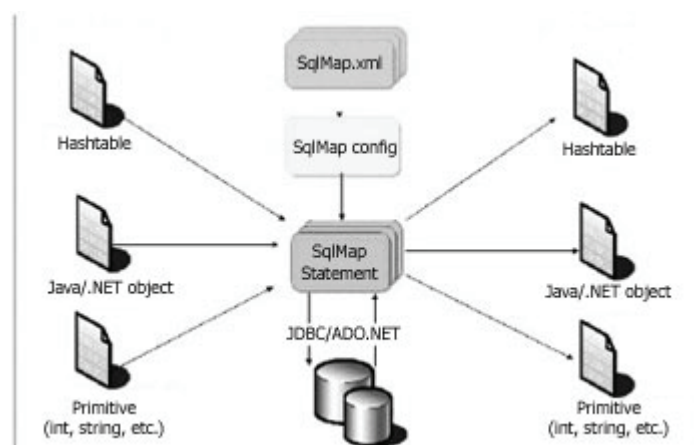


Figura 6 Architettura iBatis

Come si può vedere dalla Figura 6, iBatis Data Mapper Framework si avvale di uno o più descrittori XML per mappare il database. L'SqlMap.XML contiene la configurazione, in particolare la connessione al database e la mappatura delle query sql che verranno identificate da un id.

La creazione di un'istanza di SqlMap, chiamata generalmente SQLMapClient, mette a disposizione i vari metodi per l'accesso al database. L'oggetto di classe SQLMapClient è un oggetto di servizio thread-safe dalla lunga vita. Normalmente viene istanziato e configurato una sola volta per ogni ciclo di vita dell'applicazione. Questo lo rende un ottimo candidato per essere un membro statico di una classe. Attraverso i suoi metodi permette di interfacciarsi al database, in particolare offre dei metodi per eseguire le query che verranno richiamate attraverso il loro identificatore. Il risultato della query verrà fornito come un oggetto o come una lista di oggetti trasformando il database in un componente object-oriented. L'applicazione sviluppata è di tipo distribuito e, in questo contesto, i database devono essere gestiti in modo concorrente. Spesso può capitare che due client richiedano la stessa risorsa e quindi concorrere al suo utilizzo. Se entrambi tentano di leggere questo dato non ci sono grandi problemi, ben più complicato quando uno o più tentano di scrivere. Si deve quindi gestire la concorrenza dei client e la atomicità delle operazioni sul database in modo da salvaguardare l'integrità dei dati; iBatis fornisce questi strumenti permettendo di utilizzare le transazioni.

Il grande vantaggio di questo framework sta, per l'appunto, nel fatto che le query non sono scritte all'interno del codice sorgente ma in un file XML. Il programmatore potrà richiamare la query utilizzando i metodi della classe fornita da iBatis che mapperà i risultati all'interno di un Bean appositamente creato in precedenza. Questo permette di vedere il database come una classe di Java. Inoltre permette di modificare le query senza dover toccare il codice dal momento che sono inserite in un file XML letto dinamicamente. Per la connessione è stato usato JNDI (Java Naming and Directory Interface) che è un insieme di api che permette di rimappare i servizi associandoli a dei nomi; questo fornisce una maggiore portabilità dell'applicazione.

2.5 STRUTS – MVC

Struts è un progetto open source di Apache ed è l'implementazione Java lato server del design pattern MVC (Model View Controller.)

2.5.1 IL DESIGN PATTERN MVC

Le tre parti principali del design pattern MVC sono:

- **Model:** Rappresenta gli oggetti dati. Il modello è ciò che viene rappresentato e manipolato dall'utente.
- **View:** Serve come rappresentazione del modello, cioè dei dati contenuti nel model.
- **Controller:** Definisce il modo con cui l'interfaccia utente reagisce all'input dell'utente. Riceve i comandi dell'utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti.

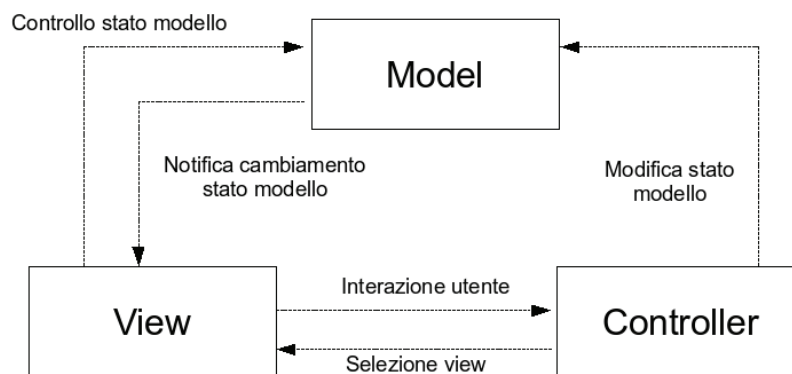


Figura 7 Schema MVC

L'utilizzo del pattern MVC offre molteplici vantaggi tra i quali la separazione tra i livelli di presentazione e di transazione; ciò permette di cambiare, ad esempio il view (vale a dire come i dati vengono mostrati) senza dover modificare il model e il controller. Un modello come questo rende più facile apportare eventuali modifiche al progetto in modo semplice e pulito.

2.5.2 I COMPONENTI DI STRUTS

I componenti base su cui si appoggia sono:

- **ActionServlet:** È la Servlet di controllo che gestisce tutte le richieste dell'applicazione. Estende la classe `javax.servlet.http.HttpServlet`.
- **Struts-config.XML:** Il file di configurazione dell'applicazione. In questo file XML si vanno a definire i vari elementi della nostra applicazione e le loro associazioni.

- **Action:** Le Action sono le classi alle quali le ActionServlet delega l'elaborazione della richiesta.
- **ActionMapping:** Contiene gli oggetti associati ad un'Action nello Struts-config come ad esempio gli ActionForward.
- **ActionForm:** Sono considerati dei veri contenitori di dati. Fanno riferimento ad uno specifico form e vengono popolati automaticamente dal framework con i dati contenuti nella request http. Questo fa sì che il popolamento del Bean con i dati dell'utente viene fatta in maniera automatica dal sistema.
- **ActionForward:** Contengono i path ai quali la Servlet di Struts inoltra il flusso in base alla logica dell'applicazione.
- **Custom-tags:** Sono tag particolari forniti dal framework Struts; vengono usati al posto dei comuni tag JSP e rendono il tutto più veloce e meno difficile.

L'ActionServlet è la Servlet di controllo di Struts. Gestisce le richieste del client e smista il flusso in base alla configurazione. E' la parte fondamentale di un'applicazione scritta in Struts. La configurazione viene inserita in un XML chiamato Struts-config.XML questo file viene letto in fase di inizializzazione dell'applicazione dall'ActionServlet e definisce le associazioni tra i vari elementi di Struts. Nello Struts-config.XML si trovano i path delle richieste http e le classi Action associate.

Gli ActionForm che sono collegate alle varie Action, vengono automaticamente popolati dal framework con i dati della richiesta associata. Poi tali dati, vengono passati all'Action.

Oltre a questo, la configurazione contiene l'associazione tra la Action e le ActionForward, cioè i path ai quali la ActionServlet ridirigerà il flusso applicativo al termine dell'elaborazione.

In figura è rappresentato il flusso elaborativo di Struts:

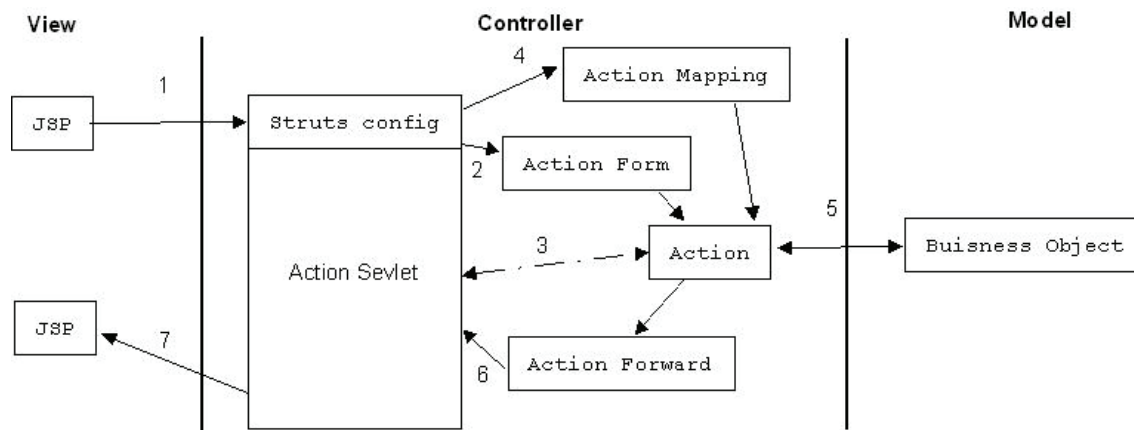


Figura 8 Flusso di elaborazione di Struts

Come si può vedere nella Figura 8, il client attraverso la pagina JSP invia una richiesta http (1), la quale viene ricevuta dalla Servlet di Struts che prevede di popolare l'ActionForm associato alla richiesta con i dati della request (2) e l'ActionMapping associata alla richiesta (4). L'ActionServlet delega l'elaborazione dei dati alla relativa Action (3) passando in input request e response http, actionForm e actionMapping precedentemente inizializzati. L'action si collega al Business Object che implementa la logica applicativa. Al termine dell'elaborazione restituisce all'ActionServlet un ActionForward (6) contenente il path della vista da fornire all'utente (che verrà ricercato nella configurazione di Struts). La Action esegue il forward alla vista specifica nell'ActionForward (7).

2.5.3 STRUTS E MVC

Si possono notare le peculiarità di Struts che si rifanno al pattern MVC:

- Esiste una sola Servlet di controllo centralizzata. Tutte le richieste sono mappate sull'ActionServlet nel web.XML dell'applicazione. Questo consente di avere un punto unico di gestione del flusso applicativo e quindi permette di implementare in modo univoco e centralizzato le funzioni (sicurezza, logging, filtri etc).
- I livelli logici dell'applicazione sono separati, le viste dell'applicazione non contengono al loro interno il riferimento al flusso dell'applicazione e la logica applicativa.
- Nel codice non è presente nessun riferimento a nomi di pagine JSP, le viste infatti, sono identificate da nomi logici definiti nella configurazione. Questo permette una modifica semplice del flusso.

- Tutta la configurazione dell'applicazione è scritta in un file XML il che consente di modificare le associazioni tra le richieste http e le classi ad essa associate in modo molto semplice.

2.5.4 VANTAGGI

- Modularità e Riutilizzabilità

I diversi ruoli dell'applicazione sono affidati a diversi componenti. Ciò consente di sviluppare un codice modulare più facilmente riutilizzabile. Questo permette modifiche veloci e senza gravare su tutta l'applicazione.

- Mantenibilità

L'applicazione è costituita da livelli distinti. Una modifica ad uno dei livelli non comporta modifiche degli altri livelli.

2.6 AUTENTICAZIONE E AUTORIZZAZIONE.

Bisogna fare una distinzione tra autenticazione e autorizzazione. La prima è il processo mediante il quale si verifica la corretta identità dell'utente, la seconda definisce a quali risorse ha accesso un determinato utente.

Spesso nelle applicazioni distribuite si ha l'esigenza di restringere l'accesso di una o più pagine solo ad alcuni utenti, come per esempio una parte di gestione accessibile al solo amministratore e non a tutti gli utenti.

Si possono individuare principalmente tre tipi di autenticazione:

- Autenticazione Base: All'utente viene mostrata una finestra di dialogo attraverso il browser. Questa finestra richiede l'inserimento dell'user name e password che verranno inviate in chiaro al server in maniera tale che sia possibile riconoscere le credenziali e autenticare l'utente. Questa tipologia è di facile implementazione ma poco sicura, dato che le informazioni sono inviate in chiaro.
- Autenticazione Digest: Utilizza lo stesso principio dell'autenticazione base con la differenza che le credenziali vengono inviate al server convertite in un formato tale da rappresentare una sorta di impronta digitale delle credenziali stesse. È più sicura del metodo precedente ma non è supportata da tutti i browser e non offre comunque un'elevata garanzia di sicurezza.
- Autenticazione Form-based: Utilizza un web form in sostituzione della finestra di dialogo. In questo caso le informazioni sono inviate in chiaro ma attraverso,

per esempio, connessioni SSL che possono crittografare i dati. Questo metodo è supportato dalla maggior parte dei browser e fornisce un'elevata sicurezza.

2.6.1 REALM

Il realm rappresenta il meccanismo per la gestione delle autenticazioni in Tomcat. Sfrutta un insieme di dati per gestire gli utenti e i vari ruoli all'interno del sistema. Il realm si divide in quattro tipologie:

- JDBCRealm: usa una base di dati relazionale.
- DataSourceRealm: usa una base di dati relazionale attraverso un collegamento JNDI.
- JNDIRealm: usa un collegamento JNDI, solitamente una directory LDAP.
- MemoryRealm: usa un file in memoria.

Di default Tomcat usa il MemoryRealm; nel progetto si è sfruttato il DataSourceRealm (DSR).

Il DSR è Form-based, si avvale di un database che contiene gli utenti e i relativi ruoli per gestire le credenziali. I ruoli si possono intendere come una classificazione degli utenti in gruppi: ogni ruolo rappresenterà delle determinate competenze e quindi aree accessibili o meno all'interno dell'applicazione. La gestione e il controllo dell'autenticità viene svolta in automatico da Tomcat per cui non è necessario implementare pagine di controlli che non è detto siano efficaci. Una volta definito il database e specificato la configurazione di Tomcat, si possono definire le autorizzazioni dei vari ruoli al fine di impedire, o meno, l'accesso a determinate aree dell'applicazione. Così facendo si ha un metodo per autenticare e autorizzare gli utenti che risulta semplice, preciso e perfettamente integrabile in un'applicazione web.

2.6.2 LDAP

Tra le richieste fatte in fase di progettazione c'era quella di inserire, in futuro, l'autenticazione attraverso LDAP. LDAP (Lightweight Directory Access Protocol) è un protocollo standard per l'interrogazione e la modifica dei servizi di directory. A differenza dei database è ottimizzato per effettuare operazioni di ricerca ed accesso alle informazioni che all'interno del server sono organizzate in modo gerarchico. L'uso di LDAP ci permette di avere un database centralizzato e dedicato alla gestione degli utenti, quindi un unico login per tutti i servizi dell'azienda.

2.7 REPORT, JASPERREPORT E IREPORT

Spesso, come in questo caso, è necessario integrare, all'interno dell'applicazione, anche la funzionalità report. E' stato deciso di usare JasperReport perché è un'alternativa molto valida, non solo nell'ipotesi di realizzare ex novo il motore di creazione dei report, ma anche nei confronti di altri software commerciali. JasperReport mette a disposizione delle api per le funzionalità tipiche di un software per la creazione di report quali:

- Formattazione di ogni elemento di un testo (font, allineamento, spaziatura, colore, etc.)
- Definizione di parti standard predefinite (header, footer, sommario, etc.)
- Gestione di raggruppamenti
- Valutazione di espressioni
- Campi calcolati
- Gestione di subreport
- Inserimento di immagini
- Stampa
- Salvataggio dell'output su file di diverso formato

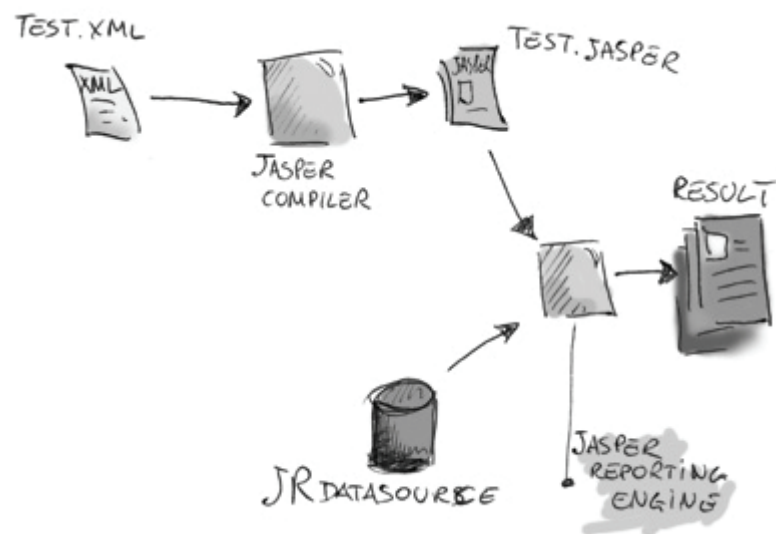


Figura 9 Come JasperReport lavora.

Come si può intuire dalla Figura 9, prima di potere utilizzare JasperReport deve essere creato il file XML contenente la struttura del report. Per creare questo file si è utilizzato iReport che è il tool visuale di JasperReport. Attraverso questo tool è possibile definire la struttura del report in ogni sua parte gestendone la forma e inserendo grafici o altri oggetti. L'utilizzo di un file XML esterno ci permette di modificare la forma dei report senza dover modificare il codice sorgente del progetto.

JasperReport usa questo file compilato per creare vari report, le api messe a disposizione non fanno altro che inserire i dati, che possono essere presi da Bean o da query, all'interno della struttura. Dopo aver completato la fase di inserimento, la struttura generata può essere salvata in diversi formati file: PDF (nel nostro caso), HTML, RTF ecc.

2.8 AJAX

Ajax, acronimo di Asynchronous JavaScript and XML, è una tecnica di sviluppo web per creare applicazioni web interattive. Ajax non è una nuova tecnologia, ma un concetto utilizzato principalmente da Google nei suoi applicativi. Questo concetto, come si può capire dall'acronimo, consiste nell'utilizzo asincrono di JavaScript attraverso l'interfacciamento con XML.

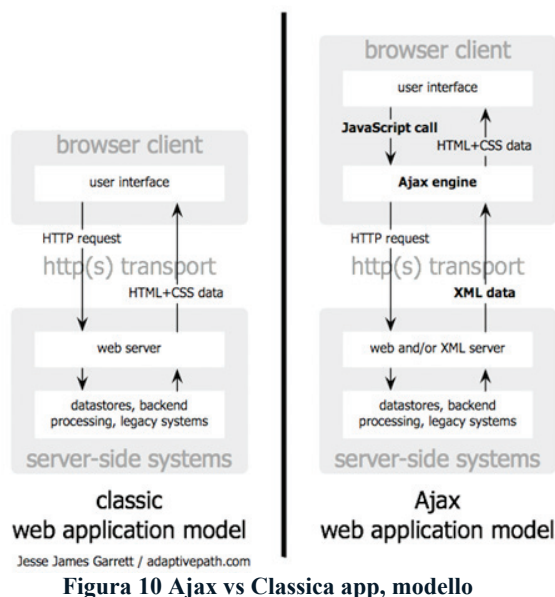


Figura 10 Ajax vs Classica app, modello

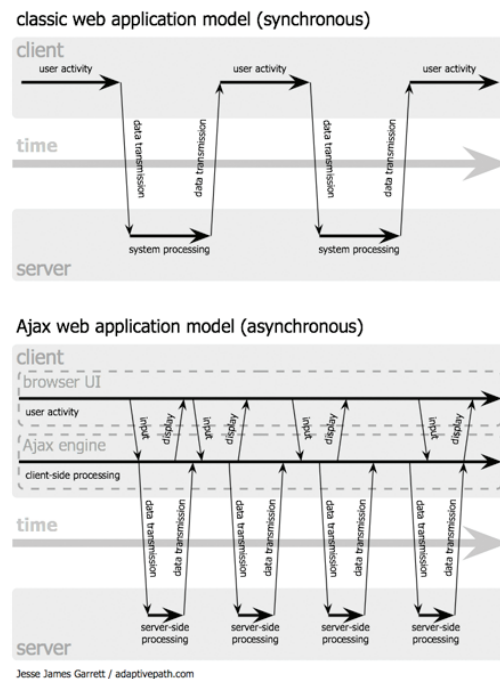


Figura 11 Ajax vs Classica app, funzionamento

Come si può vedere in Figura 10 il modello dell'applicazione Ajax si differenzia da quello di un'applicazione standard. L'applicazione Standard comunica in maniera diretta con il web server da cui poi riceve la risposta sotto forma di pagina web.

Nell'applicazione Ajax, la comunicazione avviene prima attraverso un Ajax engine che fa la richiesta al server web o XML e ne riceve la risposta, in generale in formato XML e attraverso l'Ajax engine la rielabora per presentarla all'utente sotto forma di html.

Un'altra differenza che si nota, in Figura 11, è come la richiesta venga gestita nei due casi. Nell'applicazione normale la richiesta è in modo sincrono:

1. L'utente compie un'azione.
2. I dati vengono trasferiti al server
3. Il server processa i dati
4. I dati elaborati vengono inviati al client.
5. Viene visualizzato l'output
6. L'utente può compiere un'altra azione

Tutte queste operazioni devono essere svolte in sequenza e ogni azione inizia solo dopo che la precedente è terminata.

Nell'applicativo Ajax la richiesta è asincrona:

1. L'utente compie un'azione.
2. L'input viene delegato all'Ajax engine.
 - a. Il controllo ritorna all'utente.
 - i. L'utente continua l'utilizzo dell'applicazione.
 - b. I dati sono passati al server che li processa.
 - i. Appena i dati sono stati elaborati vengono inviati all'Ajax engine.
 - ii. I dati sono visualizzati.

L'asincronicità delle azioni riguarda lo scorporamento dell'azione dal suo risultato, ad ogni azione non è necessario attendere la risposta del server per continuare, quindi si ha la possibilità di inviare un numero di richieste senza attendere il reale completamento di esse, le risposte verranno presentate appena possibile. L'attesa della risposta da parte del server normalmente è impercettibile e quindi si avrà un'applicazione che risponde in tempo reale alle nostre richieste aggiornando solo le parti essenziali.

Questo però può essere uno svantaggio, infatti, si può avere necessità di dover attendere la risposta per effettuare delle operazioni fondamentali, mentre l'utente potrebbe chiudere la pagina impedendo il completamento dell'operazione.

Un applicativo Ajax si sviluppa sfruttando XMLHttpRequest che permette di effettuare richieste ad un server Web in modo indipendente dal browser.

Il motivo del suo successo è dovuto principalmente ai livelli di funzionalità raggiunti delle applicazioni sviluppate con questa tecnologia e dalle innumerevoli applicazioni possibili, a volte anche esagerate. Nel suo utilizzo primeggia Google che n'è stato il pioniere principalmente con Gmail e Google Maps.

Ajax ha un numero elevato di applicazioni all'interno dei siti web. Viene sfruttato per creare delle interfacce interattive dove è possibile spostare e nascondere oggetti, aggiornare porzioni di pagina (come in Gmail) o ricavare in tempo reale dati utili all'utente (come nel caso dell'auto-completamento).

Per l'utilizzo di Ajax all'interno del progetto si è sfruttato la libreria di script.aculo.us che permette attraverso delle api un utilizzo semplice e immediato delle potenzialità di Ajax.

3 LA DEFINIZIONE DELL'APPLICAZIONE

In questa sezione viene presentata l'applicazione dal punto di vista dell'ingegneria del software e di tutti i suoi principali componenti.

3.1 UTENTI

Come qualsiasi altra applicazione distribuita e non, anche questa è accessibile da vari tipi di utenti (attori) che hanno ruoli diversi. Quindi, ogni attore si distingue per la competenza e le iterazioni che può avere con il sistema.

- **Corsista:** L'utente in questione usa il sistema per completare i questionari di corsi o manifestazioni alle quali ha partecipato. Viene identificato attraverso un codice alfanumerico univoco. Tale codice identifica la persona (che è presente nel database) nel caso di un questionario non anonimo. Nel caso si voglia mantenere l'anonimato viene creato un codice specifico associato ad un utente che è privo di generalità anagrafiche.

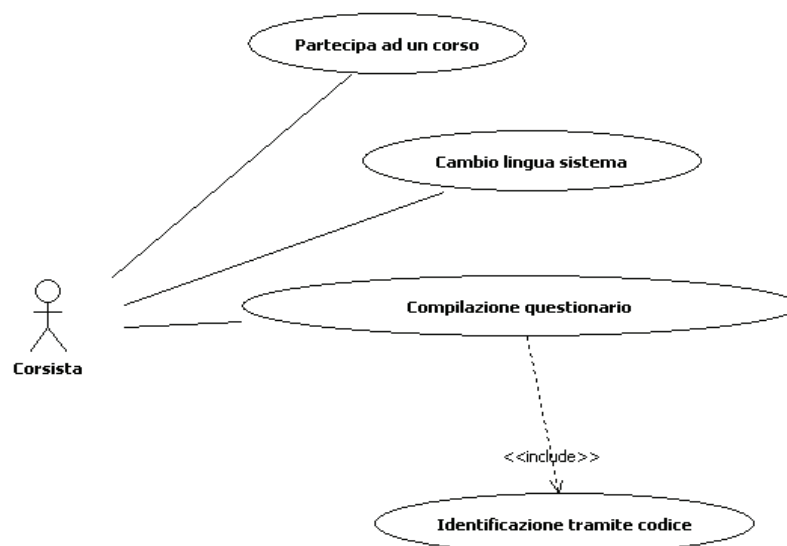


Figura 12 Use Case Corsista

- **Docente:** Il docente (identificato con un login e password) ha la possibilità di ricevere informazioni riguardo ai giudizi sui corsi che ha svolto e che lo

riguardano. In particolare ha la possibilità di visionare i vari report dei suoi corsi e di salvarli in un formato pdf.

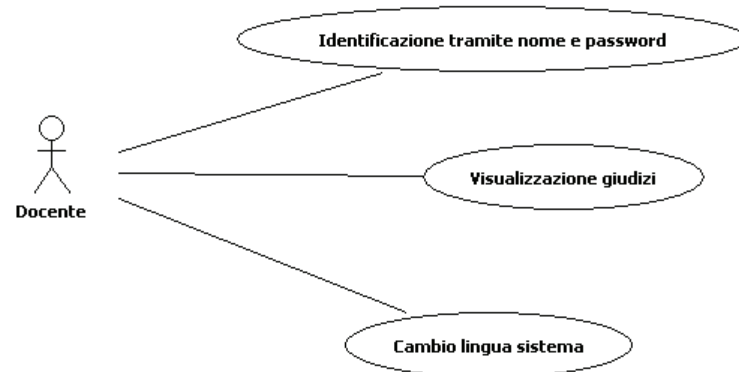


Figura 13 Use Case docente

- **Segretaria:** è l'attore che gestisce il collegamento tra un contatto e il relativo corso. Ha la possibilità di aggiungere le associazioni tra corso e contatto e quindi anche al relativo questionario.

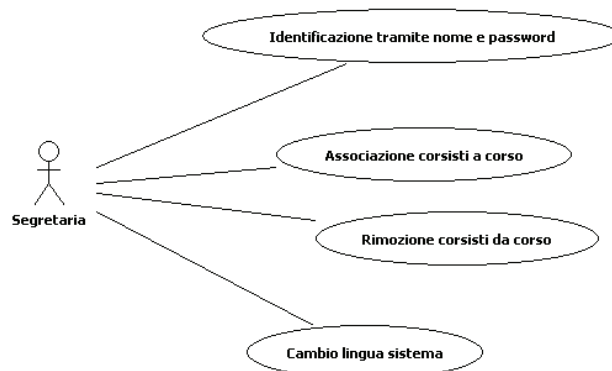


Figura 14 Use Case segretaria

- **Manager:** L'utente manager è un utente generico che ha la possibilità di creare report generici dei questionari.

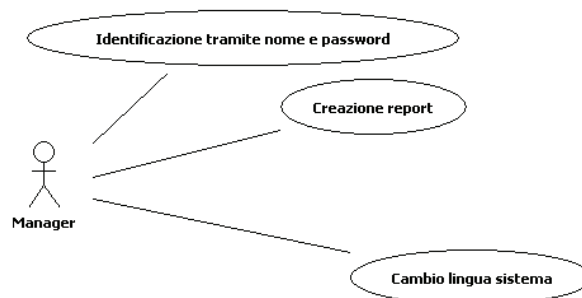


Figura 15 Use Case manager

- **Gestore corsi:** Il “gestore corsi” ha la possibilità di gestire i vari questionari, inserire domande con relative risposte, sia generiche sia specifiche per i vari questionari. Inoltre ha la possibilità di inserire corsi all’interno del sistema ed assegnare i relativi questionari.

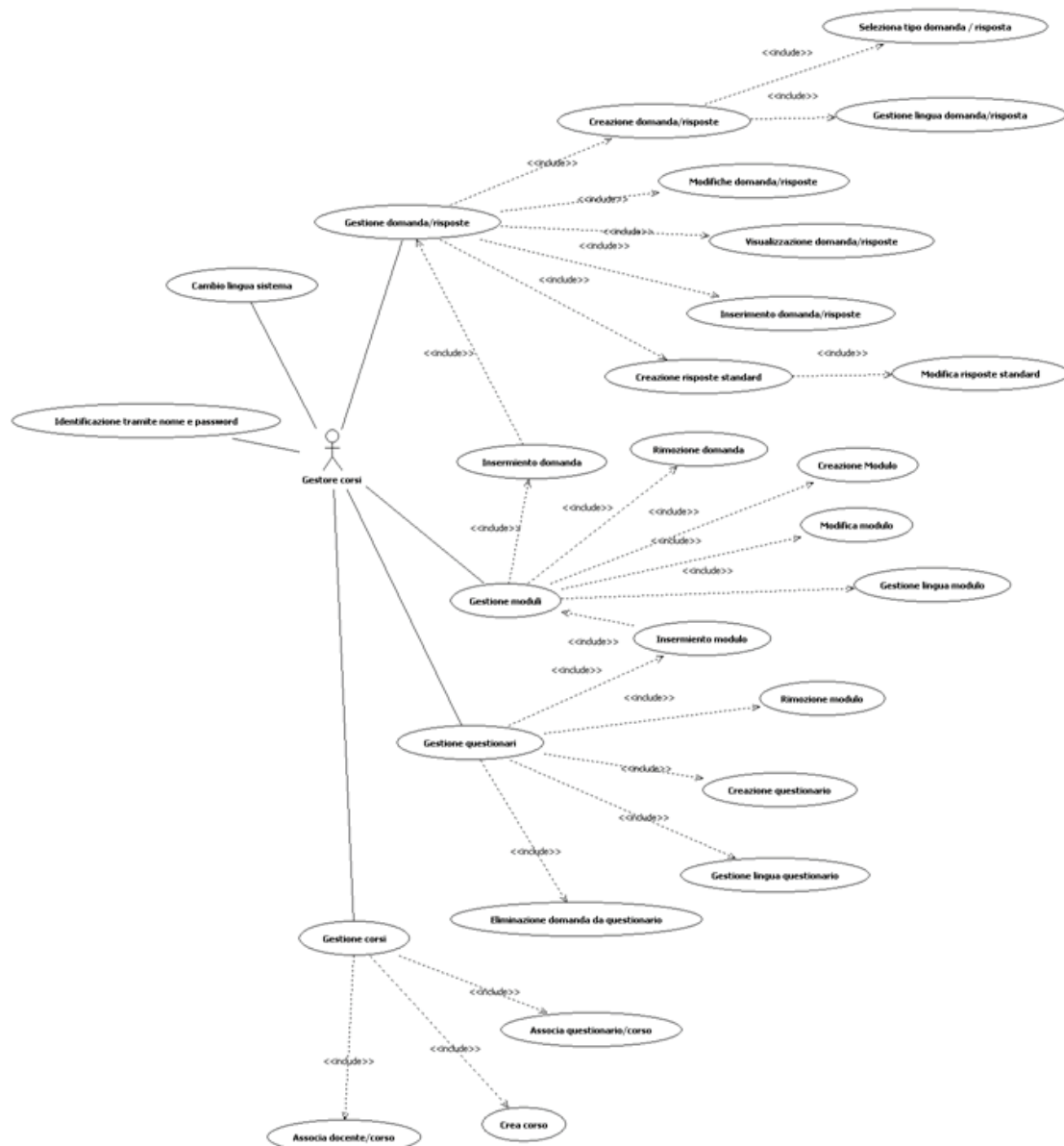


Figura 16 Use Case gestore corsi

- **Admin:** Ha la possibilità di creare utenti interni al sistema e di assegnare ruoli e compiti. Svolge anche tutte le attività degli altri utenti.

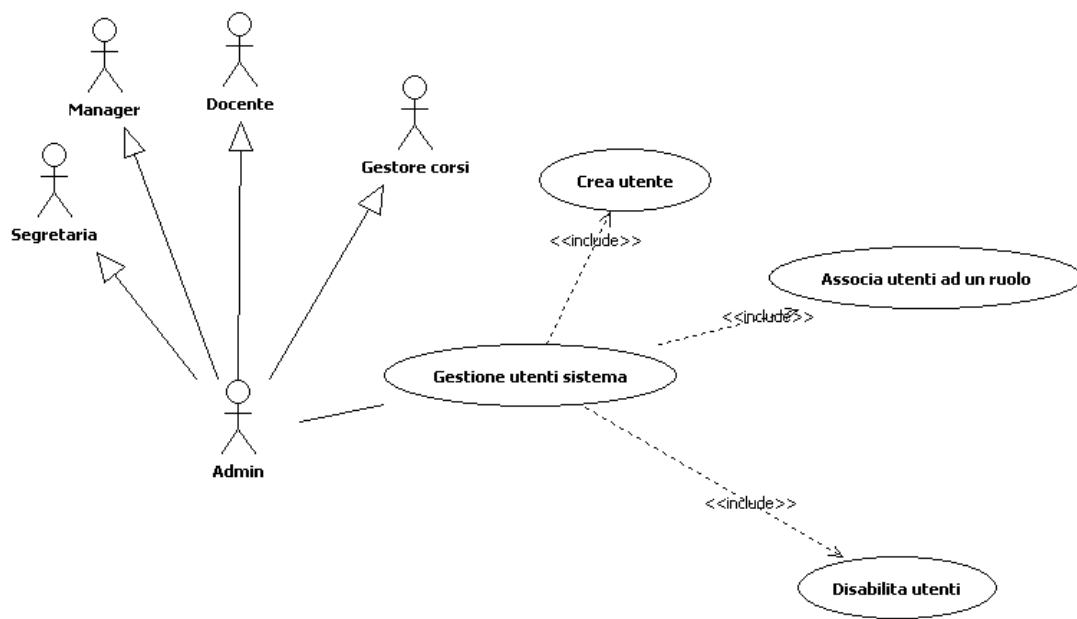


Figura 17 Use Case admin

L'admin, ha quindi tutte le caratteristiche degli altri utenti del sistema, escluso il corsista.

Per semplicità è stato implementato solo l'utente Admin e l'utente Corsista.

3.2 REQUISITI

3.2.1 FUNZIONALI

I requisiti funzionali sono quelli specificati dal richiedente del software. In pratica, sono le funzionalità che tale software dovrebbe avere.

MULTILINGUA

Il sistema deve dare la possibilità di essere usato da persone di diverse lingue. In particolare, deve permettere l'inserimento di questionari multilingua, cioè di questionari che cambiano nella forma (cioè la lingua delle domande presentate) ma non nel contenuto (il tipo di domanda e il suo significato). Quindi, ogni questionario può contenere il testo in più lingue.

GESTIONE MODULARE DEL QUESTIONARIO

L'applicazione deve permettere la creazione modulare di questionari, così da poter riutilizzare parti del questionario in altri questionari. Di conseguenza, deve

permettere di inserire i vari moduli o domande in modo indipendente dal questionario. In un questionario si potranno inserire moduli già esistenti o creati appositamente.

Si deve controllare la compatibilità delle lingue dei vari componenti che vengono utilizzati, cioè, che gli oggetti figli (gli oggetti che sono contenuti in altri oggetti) abbiano almeno le stesse lingue dell'oggetto padre (l'oggetto che contiene) (es: inserendo una domanda in un modulo con testo in italiano e inglese, la domanda dovrà avere il testo almeno in italiano e inglese).

Gestione moduli nel questionario.

Il sistema deve dare la possibilità di inserire, modificare o togliere moduli dal questionario.

Deve inoltre permettere la costruzione di nuovi moduli in varie lingue.

All'interno di un questionario è richiesta la possibilità di inserire moduli già esistenti. L'inserimento di tali moduli implica anche l'inserimento delle domande associate.

La modifica del modulo prevede la possibilità di modificare il testo inserito, cioè il titolo. È necessario informare l'utente che apporta una modifica nel caso che tale modifica influenzi più questionari ed eventualmente impedirla se uno dei questionari ha già avuto risposte.

Gestione domande nel modulo.

Il sistema deve far in modo che l'amministratore sia in grado di inserire domande esistenti cercandole o altrimenti creandone di nuove.

Creazione

Per ogni domanda deve essere possibile: inserire il testo in più lingue, selezionare il tipo di risposta (aperta, risposta singola, risposta multipla) ed eventualmente un numero minimo di risposte richieste.

Se il tipo di domanda comporta delle risposte predefinite si deve dare la possibilità all'amministratore di inserirle ex novo o di selezionarle da un set di risposte già esistenti.

Modifica

La modifica delle domande comporta la modifica sia del testo della domanda, sia del tipo, sia del numero minimo di risposte richieste.

Eliminazione

Si deve dare la possibilità di eliminare una domanda dal rispettivo modulo. Si deve informare l'utente se tale modifica comporterà delle modifiche ad altri questionari ed eventualmente negarla.

RISPOSTE STANDARD

E' richiesto che venga data la possibilità di utilizzare, e quindi inserire, un certo insieme di risposte standard. Di solito si tratta di giudizi: sufficiente, buono ottimo ecc.

ANONIMATO

Come succede per i questionari cartacei deve essere data la possibilità, se il questionario lo permette, di poter rispondere in maniera anonima. Questo comporta di utilizzare una strategia di immagazzinamento dei dati che non associ in alcun modo l'utente alle risposte date.

CREAZIONE REPORT

L'applicazione deve permettere di creare dei report in base al tipo di questionario. In oltre, i report devono essere flessibili e di facile modifica.

GESTIONE UTENTI

Il sistema deve permettere una corretta gestione degli utenti all'interno dell'applicazione. Per questo è richiesto che ogni utente sia identificato in un modo univoco. Per i corsisti l'identificazione avviene con un codice alfanumerico univoco, tale codice identifica l'utente stesso, se il questionario è non anonimo, altrimenti identifica un utente creato ad hoc, se il questionario è anonimo.

COMPLETAMENTO QUESTIONARIO.

L'applicazione deve permettere all'utente di inserire le risposte al questionario proposto.

Ad ogni utente sarà fornito un codice univoco.

3.2.2 INFORMAZIONI

QUESTIONARIO

Il questionario è l'oggetto principale del progetto. Un questionario è formato da un insieme di moduli che a sua volta contengono domande e risposte scritte nelle varie lingue. Nell'introduzione è spiegata la struttura del questionario.

QUESTIONARIO NON ANONIMO

Quando un utente risponde ad un questionario di questo tipo, il sistema, dopo aver identificato l'utente, lo assocerà alle risposte date. L'utente quindi non deve fornire i propri dati anagrafici poiché sono già presenti nel database. Una volta usato, il codice diventerà non valido e quindi non potrà essere riutilizzato. Il codice ha validità di un anno dopo questo tempo sarà inutilizzabile.

QUESTIONARIO ANONIMO

Quando un utente risponde ad un questionario di questo tipo, il sistema assocerà le risposte ad un utente (anonimo) creato ad hoc che sarà lo stesso per tutti i corsisti. La scelta dell'anonimato viene inserita per rendere più libero l'utente nel giudizio, questa necessità è dettata dal fatto che alcuni utenti, sapendo che i questionari sono associati alla loro persona, possono esprimere un giudizio non veritiero. Tale opzione è data all'amministratore in fase di creazione. Se viene scelto un questionario di tipo anonimo, quando si passerà all'inserimento degli utenti, verrà proposto semplicemente il numero di codici da generare e non a chi associarli.

MODULO

Il modulo è un raccoglitore di domande. Alla creazione del modulo sarà richiesto un testo per ogni lingua che si vuole inserire.

DOMANDA E RISPOSTA

Come il modulo, la domanda viene creata inserendo il testo del quesito nelle varie lingue; inoltre, viene scelto il tipo e il numero minimo di risposte da dare. Se il tipo di domanda lo richiede viene fornito anche il modulo per inserire le risposte creandole o ricercandole tra l'insieme di risposte standard create in precedenza. Il numero di

risposte nelle varie lingue deve combaciare ed essere maggiore del numero minimo richiesto.

CORSO

Il corso è un'entità marginale del progetto. Viene svolto da Enginsoft e serve, eventualmente, per collegare il questionario ai vari partecipanti. Si è deciso di staccare l'entità corso dal progetto in modo da poterlo riutilizzare anche in altri ambiti.

QUESTIONARIO/MODULO

Un modulo, per far parte di un questionario deve possedere almeno le stesse lingue del questionario, nulla vieta di averne in più (un modulo potrebbe avere più lingue per poterlo riutilizzare in altri questionari). Se le lingue del questionario e del modulo non sono compatibili non si potrà proseguire.

MODULO/DOMANDE

Una domanda, per far parte di un modulo, deve avere almeno le stesse lingue del modulo, anche qui nulla vieta di averne in più. In caso di lingue incompatibili non si potrà proseguire.

CORSO / QUESTIONARIO

Si deve dare la possibilità di associare un questionario al rispettivo corso, ma questa caratteristica non è essenziale, quindi un questionario non è sempre associato ad un corso. Questa scelta è dettata dalla possibilità di riutilizzare l'applicazione anche in altri contesti. Quindi è necessario dare la possibilità di creare questionari non legati necessariamente ad un corso. Se ad un questionario viene associato un corso, i partecipanti al corso verranno direttamente associati al questionario. Se tale questionario è anonimo verranno associati ad un numero di login anonimi pari al numero di partecipanti al corso.

CORSISTA / QUESTIONARI

Per poter rispondere al questionario il corsista deve aver partecipato al corso e quindi essere in possesso del codice.

DOCENTE / QUESTIONARI

Il docente ha accesso solo alle informazioni dei questionari dei corsi che ha tenuto.

MODIFICA/ELIMINAZIONE DOMANDA/MODULI DI PIÙ QUESTIONARI.

Una domanda (con relative risposte) o modulo può appartenere a più questionari. Perciò la modifica o l'eliminazione di tale domanda può portare a incongruenze di dati.

Pertanto la modifica è possibile per gli errori nella scrittura delle domande e risposte e può essere fatta solo sui moduli o domande alle quali ancora nessuno ha risposto.

L'amministratore sarà avvisto in caso in cui una modifica agisca su più di un questionario.

3.2.3 REQUISITI NON FUNZIONALI E ARCHITETTURALI

APPLICAZIONE DISTRIBUITA

L'applicazione deve essere distribuita dato che dovrà essere resa utilizzabile da utenti che non si trovano nello stesso posto.

SEMPLICITÀ

Il sistema deve essere intuitivo nell'utilizzo, sia nella fase di creazione che di completamento del questionario.

SICUREZZA

Il sistema deve essere sicuro e garantire la salvaguardia dei dati presenti nel database aziendale. Per una maggiore sicurezza viene introdotto l'utilizzo di realm e in una versione futura un server LDAP.

LINGUE

L'introduzione di nuove lingue, la modifica di quelle presenti, deve essere semplice e gravare in maniera minima sul sistema.

UTILIZZO DATABASE ESISTENTE

L'applicazione dovrà estendere il database esistente, deve perciò integrarsi con quello presente nell'azienda ed eventualmente ampliarlo inserendo tabelle ad hoc.

UTILIZZO FRAMEWORK

E' richiesto l'utilizzo dei seguenti framework per gestire il progetto in maniera standard e semplice:

- Struts per l'applicazione
- IBatis per la gestione del database
- JasperReport per i report
- Altri minori.

4 INTRODUZIONE SVILUPPO DELL'APPLICAZIONE

In questa sezione si parlerà delle scelte effettuate per lo sviluppo dell'applicazione. Verrà data un'idea generale del database dell'applicazione e una descrizione di come le varie funzionalità sono state create utilizzando i vari framework. All'interno di questa sezione si userà spesso il termine “oggetto” per indicare una parte del questionario cioè: questionario, modulo, domanda o risposta.

4.1 INTRODUZIONE

Prima di cominciare è opportuno precisare come è stata pensato e sviluppato il sistema della gestione della lingua. Gli oggetti hanno una dipendenza di lingua che è definita gerarchicamente tra gli oggetti contenuti e gli oggetti contenitori. Gli oggetti contenuti devono obbligatoriamente avere la lingua dell'oggetto che li contiene; per avere un'idea migliore di oggetto contenuto e oggetto contenitore vedere la Figura 2 a pagina 1. Si dà la possibilità, in fase di creazione di un oggetto, di inserire i testi in tutte le lingue disponibili nel sistema ma si rende obbligatorio l'inserimento del testo nelle lingue dell'oggetto che li contiene. Questo dà la possibilità di creare oggetti (come domande) con testi in varie lingue ed inserirli in questionari o moduli che hanno una sola lingua o un insieme di lingue minore di quello della domanda, come si può notare dalla Figura 18. Così facendo si possono generare domande compatibili con qualsiasi questionario.

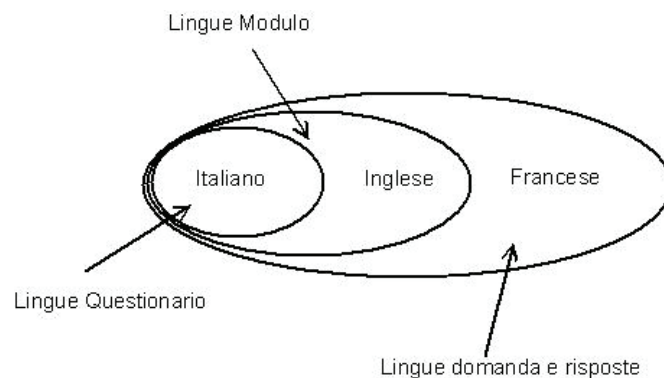


Figura 18 Vista insiemistica delle lingue degli oggetti di un questionario

In fase di presentazione del questionario all'utente finale verranno presentati solo le lingue del questionario; nel caso della Figura 18 verranno presentati solo i testi in italiano.

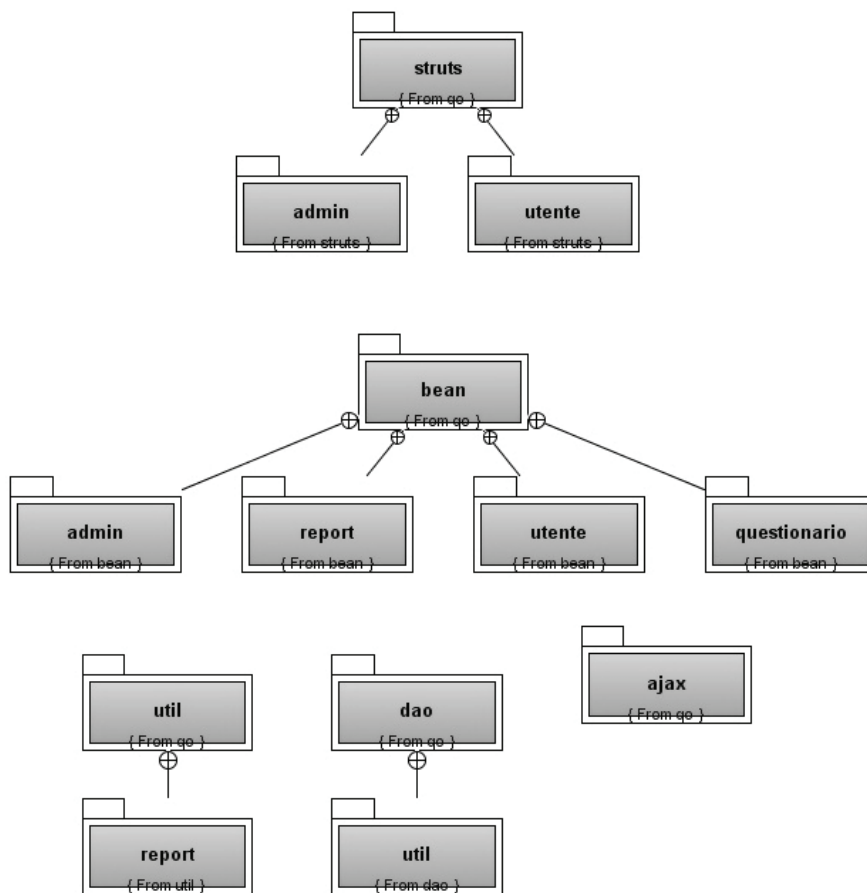


Figura 20 Diagramma package applicazione

Come si può notare dalla figura Figura 20, l'applicazione è costituita da vari package al fine di avere una gestione migliore delle classi dal punto di vista organizzativo.

All'interno del package Struts, abbiamo due sub-package, uno per l'amministratore e uno per l'utente. Queste due divisioni contengono le varie Action di Struts per quanto riguarda la gestione delle funzionalità amministrative e del utente semplice. Non sono state elencate tutte le classi in quanto sono un numero elevato e i diagrammi uml non sarebbero stati di grande interesse.

Nel package Bean si trovano i vari Bean usati nella raccolta e trasporto dei dati per: l'amministratore, l'utente, la creazione dei report e la gestione del questionario.

Come si può notare, dalle figure Figura 21 e Figura 22, la parte dell'amministrazione e dell'utente utilizzano i Bean del questionario. Questo per avere un'uniformità nella gestione dei dati, sia in fase di presentazione che in fase di inserimento nel database. La parte dell'amministrazione prevede molte più funzioni rispetto all'utente, da qui nascono i numerosi attributi presenti nel Bean. Nella parte dell'utente è stato aggiunto

un Bean che gestisce i record di domanda e risposta utilizzati per salvare le risposte. All'interno dei Bean principali è presente il metodo “validate” di Struts che permette di validare i dati prima di processarli, questa funzione verrà spiegata più avanti

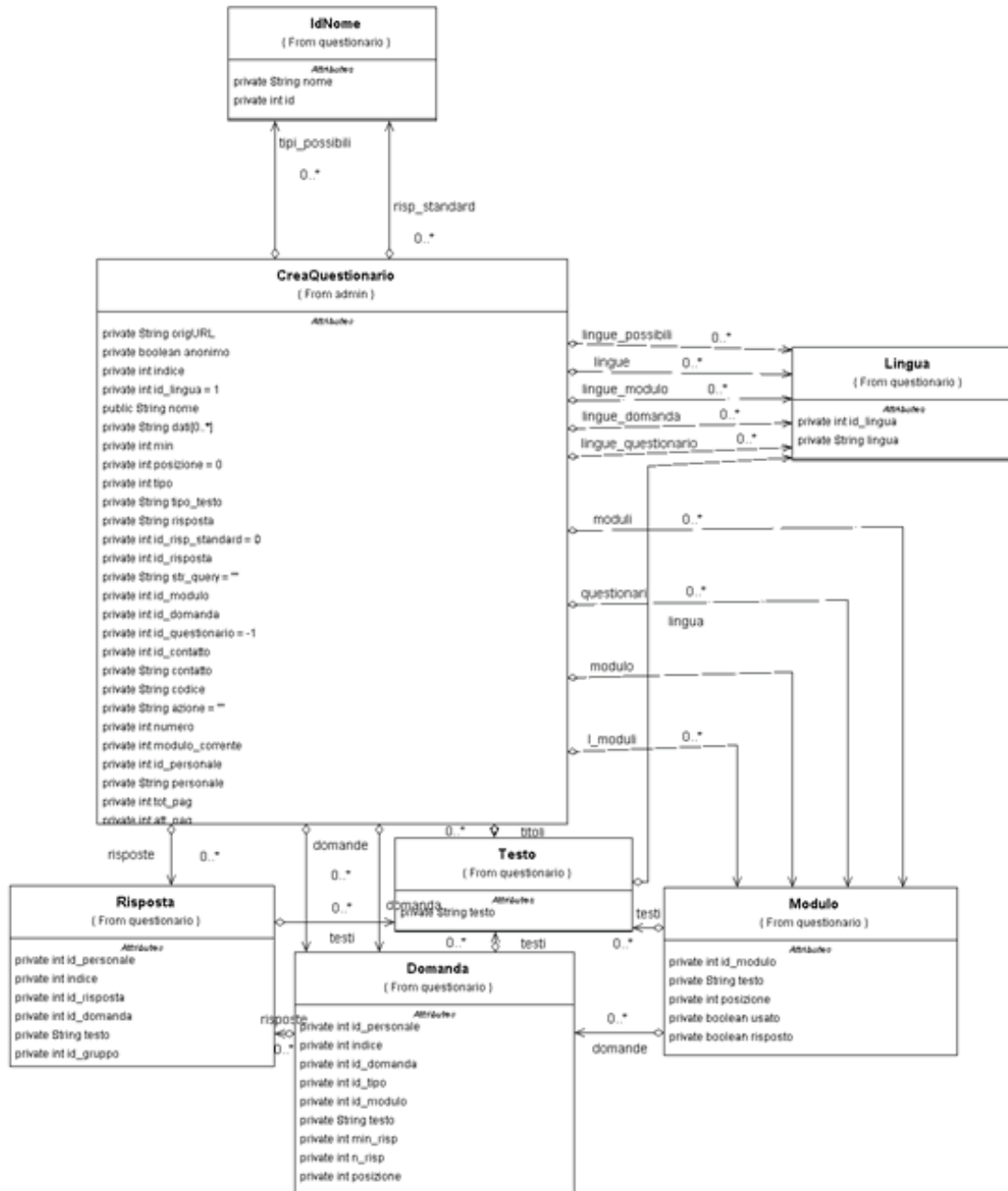


Figura 21 Bean amministratore e questionario

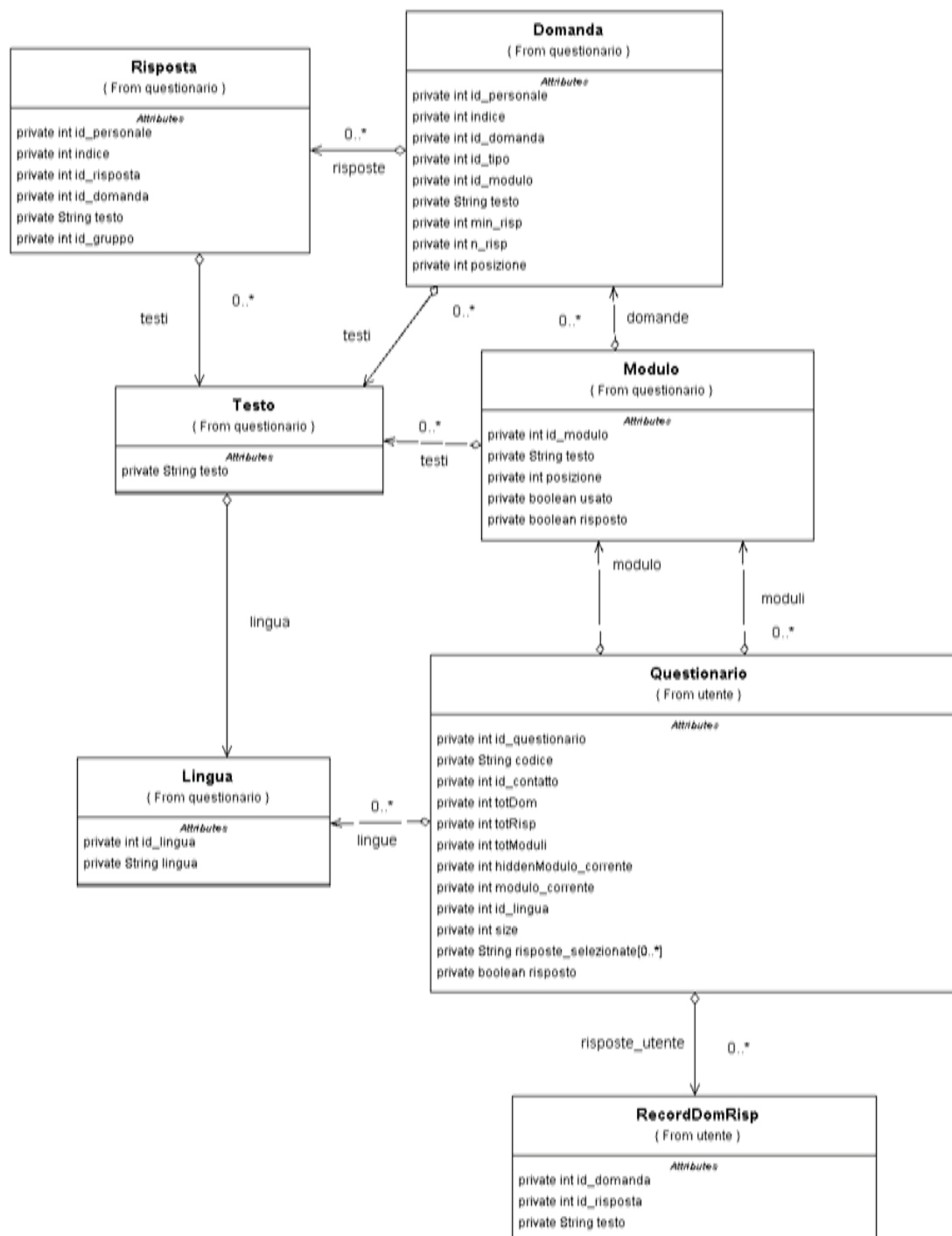


Figura 22 Bean Utente e questionario

All'interno del package dao sono presenti le classi di creazione del SqlMap di IBatis e delle classi create ad hoc per il passaggio e la ricezione di dati delle query. In util si trovano le classi che svolgono le funzioni di contorno dell'applicazione come la

generazione del codice. In questo package si trova anche la classe che contiene i parametri per la gestione dell'applicazione come: il numero di risultati da presentare e i vari identificatori del database delle lingue e utenti. Questa classe è stata introdotta per semplificare sia la creazione di nuove lingue che la gestione degli utenti e per rendere flessibile la struttura qualora si volessero utilizzare altri database. Nel package Ajax si trova la Servlet che genera l'XML per creare l'auto-complete della ricerca.

5 IL DATABASE

Il database è una delle parti fondamentali del progetto e la sua definizione è stata tutt'altro che semplice. Il database dell'applicazione utilizza parte di quello aziendale; in particolare le tabelle inerenti ai vari contatti, cioè i futuri utenti che risponderanno ai questionari, nonché il personale, vale a dire gli utenti che saranno il lato amministrativo dell'applicazione.

5.1 QUESTIONARIO

Il questionario è l'elemento principale dell'applicazione. È strutturato come un contenitore di oggetti; in particolare un contenitore di moduli.

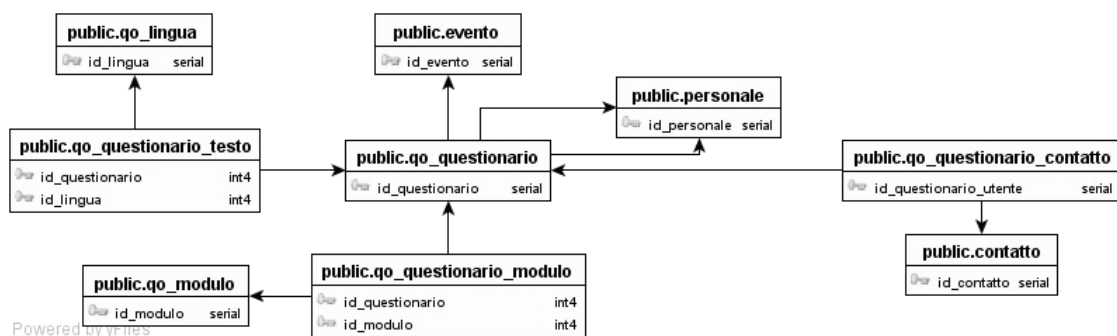


Figura 23 Questionario, schema tabelle

Come si vede nella Figura 23 il questionario presenta riferimenti ad altre tabelle. La prima peculiarità è l'indipendenza dell'oggetto "questionario" dal suo testo. Questo è stato fatto per la gestione del multilingua; il concetto verrà spiegato in modo più approfondito nella sezione: "5.5 - La gestione del multilingua".

La tabella "qo_questionario_contatto" rappresenta il collegamento che un utente ha con il questionario. In questa tabella è presente il codice alfanumerico che permette di identificare l'utente ed il questionario da presentargli; sono anche presenti: una data di scadenza e delle variabili di controllo che servono per la gestione degli accessi.

La tabella "qo_questionario_modulo" collega il modulo con il questionario. Il collegamento è del tipo "Molti-a-Molti" al fine di poter riutilizzare dei moduli: questa parte verrà spiegata in maniera più approfondita nella sezione "5.6 - La gestione modulare".

5.2 MODULO

Come detto nel paragrafo precedente, il questionario è un contenitore di moduli ed il modulo è un contenitore di domande.

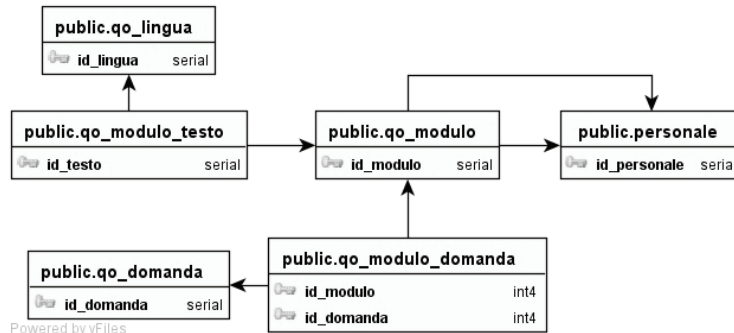


Figura 24 Modulo

La sua struttura è simile al questionario; oltre alla definizione del testo per la gestione multilingua (vedi sez 5.5 - La gestione del multilingua) al suo interno si trovano riferimenti alla tabella domanda che è del tipo “Multi-a-Molti” (per la spiegazione alla sezione 5.6 - La gestione modulare).

5.3 DOMANDA

La struttura è pressoché identica alle precedenti; in più è stato aggiunto un collegamento con la tabella “qo_tipo_domanda_risposta” che serve a specificare di che tipologia sia la domanda e quindi come deve essere presentata all’utente. Ovviamente, alla domanda sono collegate delle risposte. Nel caso di domande a risposta aperta verranno create delle nuove risposte con il testo dell’utente e poi collegate alla relativa domanda.

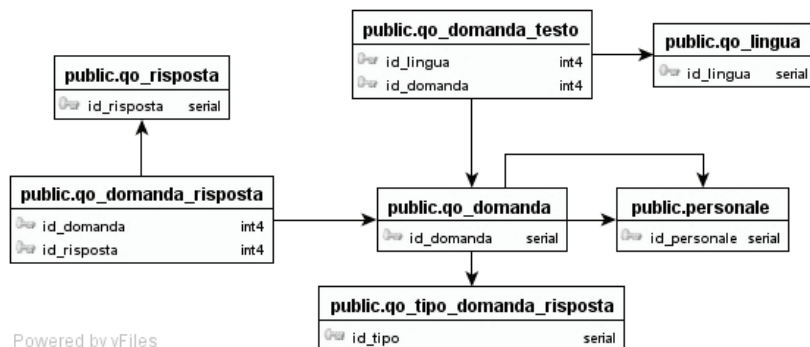


Figura 25 Domanda

5.4 RISPOSTA

È l'oggetto che sta alla base della gerarchia, la sua struttura è molto semplice. Presenta un testo gestito con la modalità del multilingua e un collegamento a "qo_gruppo_risposte_standard" che permette di raggruppare le risposte per aver dei set di risposte standard. Questa struttura è sviluppata con l'intento di ottimizzare la creazione dei questionari. Spesso accade che per determinate domande si può utilizzare sempre lo stesso set di risposte, come è il caso dei giudizi (es: sufficiente, buono e ottimo). In questi casi sarà possibile fornire all'amministratore, attraverso l'applicazione, un set di risposte già definite.

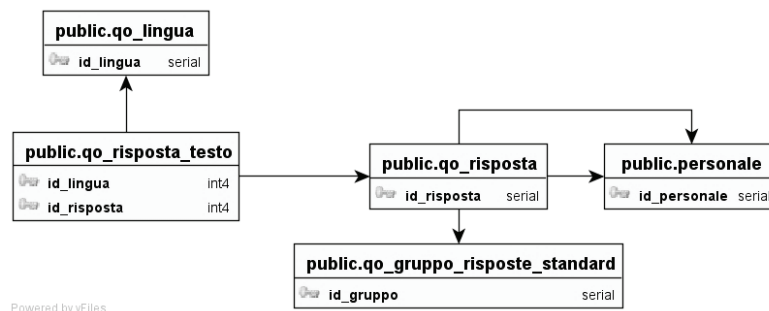


Figura 26 Risposta

5.5 LA GESTIONE DEL MULTILINGUA

Il sistema deve essere in grado di gestire dei questionari multilingua, vale a dire rendere disponibile la stessa forma del questionario cambiandone il contenuto, cioè la lingua del testo.

La struttura delle tabelle, come si vede nelle figure sopra, permette di attuare questa funzionalità. Ogni oggetto (questionario, modulo, domanda o risposta) è collegato ad una tabella che contiene il testo nelle varie lingue. Questo consente di creare una struttura del questionario collegando gli oggetti e al momento della presentazione proporre i testi nella lingua desiderata. Vi potrebbero essere delle problematiche in quanto non sempre sono disponibili i testi nelle varie lingue, per cui si prevedono dei controlli che saranno presentati più avanti.

5.6 LA GESTIONE MODULARE

Era stato anche richiesto che l'applicazione permettesse di riutilizzare parti delle componenti già esistenti.

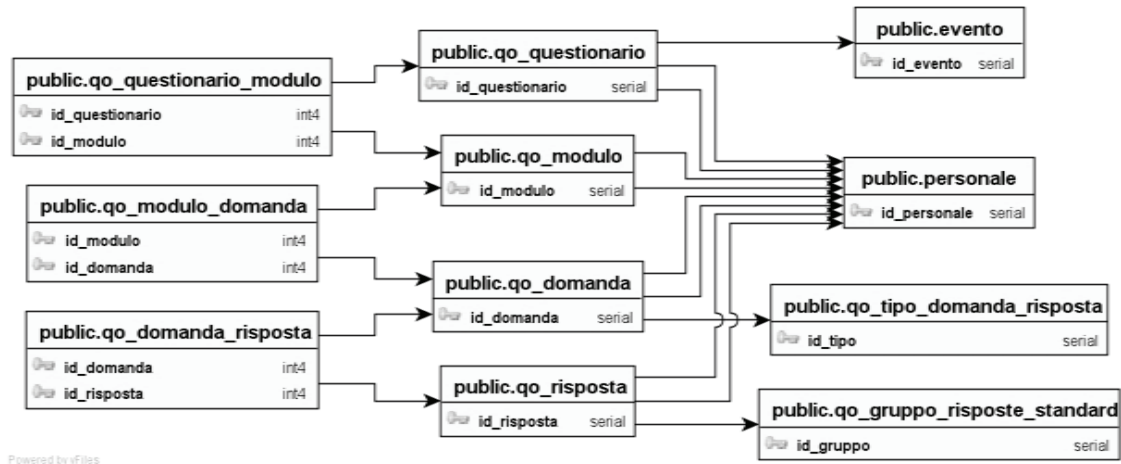


Figura 27 Collegamento questionario – modulo – domanda – risposta.

Come si vede dalla Figura 27 la struttura del database permette di soddisfare alla richiesta. I vari oggetti sono collegati tra loro attraverso una tabella esterna, cioè una relazione “Multi-a-Molti”. Permette così di collegare gli oggetti più volte; ad esempio è possibile far sì che modulo e relative domande e risposte collegate, facciano parte di più di un questionario. Questo può comportare dei problemi nella fase di collegamento e modifica dei vari oggetti coinvolti, per cui è necessario valutare le lingue dei due oggetti e tenere traccia di quanto invasiva sarà la modifica indotta. Questi controlli verranno approfonditi nella sezione specifica.

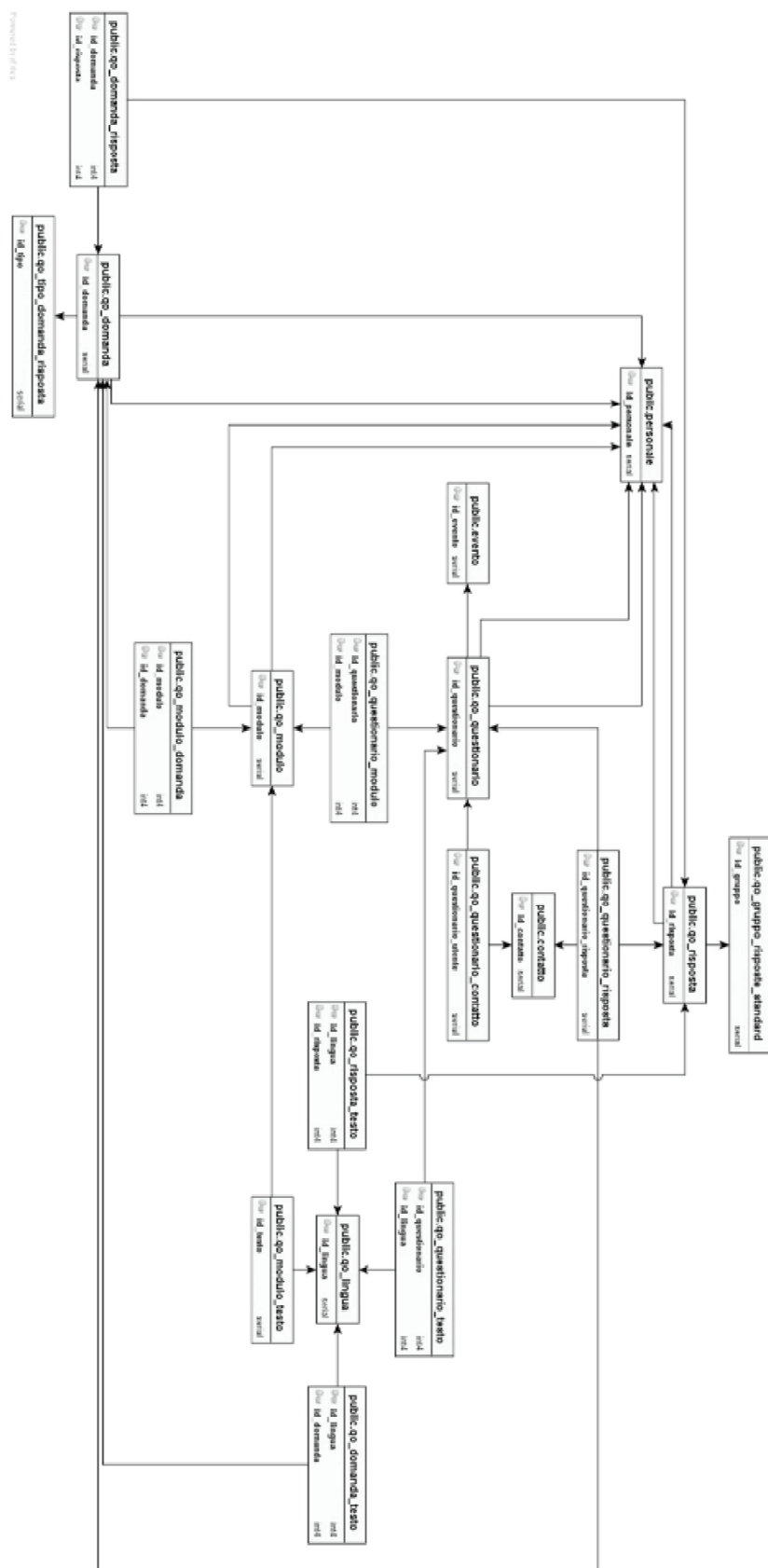


Figura 28 Schema completo del database dell'applicazione

6 IL FUNZIONAMENTO DELL'APPLICAZIONE.

6.1 UTENTE

L'utente all'interno dell'applicazione ha come unica funzionalità quella di rispondere al questionario.

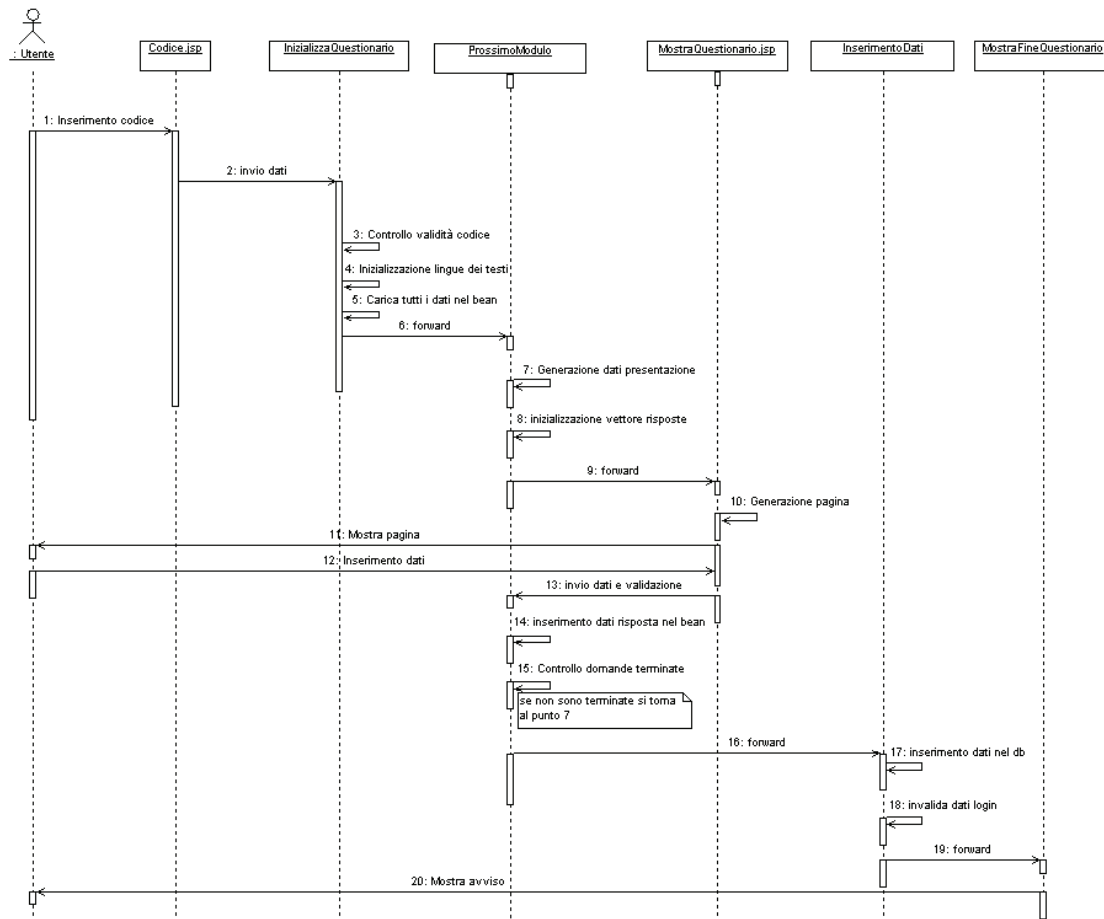


Figura 29 Sequence utente.

Nella Figura 29 è rappresentata la sequenza d'iterazioni tra utente e applicazione. Per prima cosa è richiesto l'inserimento del codice d'identificazione, che viene controllato cercando una corrispondenza nel database. Se l'esito è negativo, viene mostrato un messaggio di errore e l'utente non può proseguire; in caso di risposta positiva vengono presentati i testi del questionario utilizzando la lingua di sistema, se disponibile, in alternativa sarà utilizzata la prima lingua tra quelle presenti nel questionario. I testi del questionario e dei suoi oggetti sono caricati in apposite variabili all'interno del Bean così da diminuire le query con il database.

In seguito il Bean viene inviato alla classe di gestione del questionario che selezionerà i dati da presentare utilizzando un indice all'interno del server. All'interno di questa Servlet verrà selezionato il modulo corrente e le relative domande associate; il tutto viene inserito all'interno delle variabili di presentazione del Bean. Viene anche istanziato l'array di stringhe per utilizzare lo string properties (vedere sezione apposita) al fine di tenere traccia delle risposte dell'utente. Il Bean, appena modificato, verrà inviato alla pagina di presentazione.

La JSP di presentazione genera la propria struttura, ricavando le proprietà dal Bean, in base alla tipologia di domande (le domande cambiano struttura secondo la tipologia: si pensi alla diversità tra risposte aperte, per le quali bisogna inserire un testo e risposte chiuse, per le quali bisogna esprimere una scelta). L'utente ha ora la possibilità di rispondere al modulo corrente inviando le risposte e poi proseguire.

I dati inseriti dall'utente vengono inviati alla Servlet che gestisce il questionario; prima di essere inviati i dati vengono controllati. Il controllo viene effettuato per verificare che siano state fornite le risposte obbligatorie. In caso di controllo negativo verrà mostrato un errore e non si potrà proseguire fino all'inserimento delle risposte; se il controllo è positivo i dati vengono inviati alla classe di gestione.

All'interno della classe di gestione, prima di salvare i dati all'interno del Bean, viene controllato che l'utente stia rispondendo al modulo corrente, cioè che l'indice del modulo salvato all'interno del server sia uguale a quello inviato dall'utente, questo per impedire modifiche delle risposte già date. Nel caso di un controllo positivo i dati verranno inseriti all'interno del Bean e subito dopo verranno caricati i dati di presentazione del modulo successivo. Se l'utente ha risposto a tutti i moduli, il Bean verrà inviato alla Servlet che li inserirà nel database e invaliderà i dati di login quindi verrà mostrato all'utente il messaggio di fine questionario.

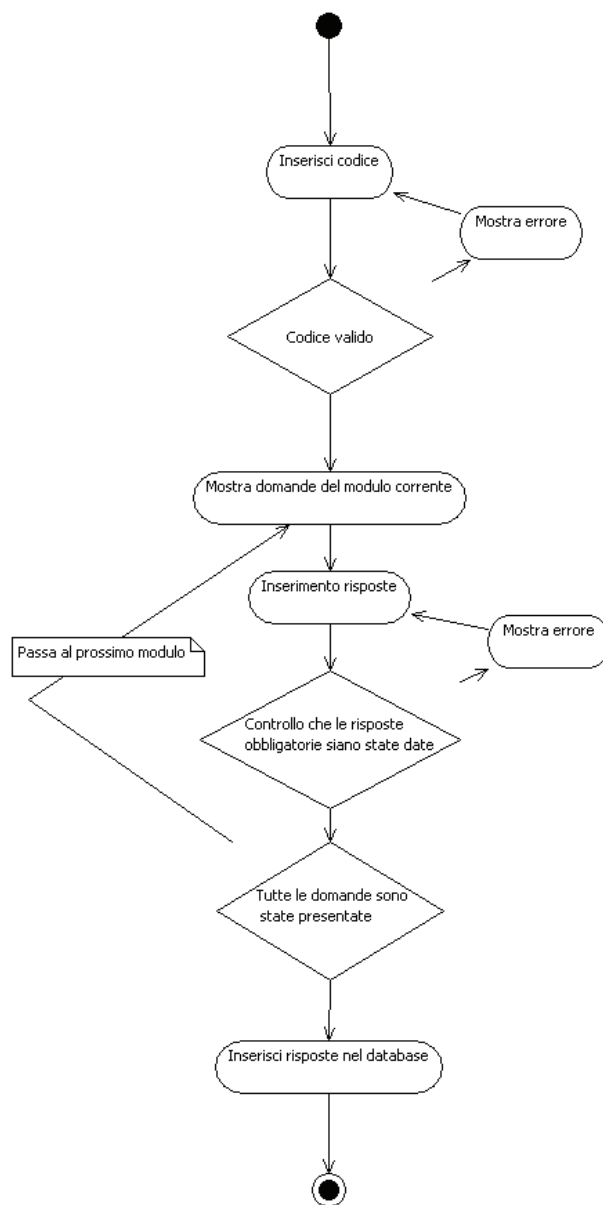


Figura 30 activity utente.

Nella Figura 30 è presente l'activity diagram semplificato.

6.1.1 APPROFONDIMENTO FUNZIONAMENTO RACCOLTA DELLE RISPOSTE

La parte più interessante del funzionamento lato utente è il metodo di raccolta delle risposte. Struts mappa automaticamente, attraverso i tag JSP, i campi del form con gli attributi del Bean; questa funzionalità non fa altro che popolare in automatico gli attributi del Bean con i dati inseriti nel form.

Nella prima versione dell'applicazione veniva mostrata una sola domanda per pagina, così da raccogliere le risposte in un attributo del Bean. Questa soluzione non era per nulla funzionale e si è quindi deciso di cercare di proporre più domande per pagina. Il problema della nuova versione era quello della raccolta dei dati in quanto non si riusciva a gestire il submit di un numero variabile di valori dal form al Bean. Dopo numerose ricerche si è riusciti a implementare questa funzionalità grazie all'utilizzo delle indexed properties che fa parte delle Bean Utilities. Questa funzionalità assieme a Struts permette di mappare un insieme di risposte in un array di stringhe di dimensione fissata a priori; nel nostro caso, però, l'array doveva essere della grandezza del numero di risposte possibili che variava da modulo a modulo. Il problema è stato superato istanziando, in fase di creazione dei dati di presentazione un array di dimensione uguale al numero di risposte possibili così da avere il Bean pronto per essere mappato da Struts.

Il funzionamento è molto semplice: l'applicazione attraverso il database ricava il numero di risposte possibili e istanzia il nuovo vettore all'interno del Bean. Il Bean viene inviato alla pagina JSP che permette all'utente di inserire le risposte. All'invio delle risposte struts mapperà i testi all'interno del vettore e nella Servlet di arrivo non resterà altro che processare i risultati all'interno del vettore.

Le diverse tipologie di risposte (aperta, singola e multipla) vengono rappresentate con una diversa stringa all'interno del vettore. Per le risposte singole e multiple verrà inviata una stringa contenente id della domanda e id della risposta. La differenza tra queste due tipologie è che la prima (risposta singola) occuperà una cella unica del vettore mentre la risposta multipla avrà la possibilità di occupare tante celle quante sono le risposte. Per la domanda di tipo aperto verrà inviato l'id della domanda e il testo della risposta. Una volta ricavati i vari id di domande-risposte e gli eventuali testi delle risposte, questi verranno messi in un vettore che a fine questionario sarà inserito nel database.

CONTROLLI

Un'altra funzionalità interessante è controllare che l'utente non modifichi le risposte già date. Poiché non è data la possibilità di tornare indietro attraverso l'applicazione, all'utente non resta che utilizzare il tasto back del browser. Per questo è stato inserito, sia lato server che lato utente, una variabile per controllare questa funzione. Lato server

viene salvato l'id del modulo corrente, mentre lato utente viene inserito nel form della pagina JSP un campo hidden che contiene l'id del modulo corrente. Quando i risultati vengono processati si controlla che questi due indici combacino, in caso di risposta negativa verrà mostrato un errore e si verrà rimandati al modulo salvato lato server.

VALIDATE

Struts mette a disposizione un metodo della classe ActionForm che ci permette di validare i campi del Bean. Tale metodo viene automaticamente richiamato dal framework quando i dati vengono inviati dalla pagina JSP alla Action. Questo metodo, per la parte dell'applicazione dell'utente, è stato usato per controllare che siano state date le risposte obbligatorie. Con dei semplici controlli sull'array delle risposte si verifica che siano presenti gli indici delle risposte obbligatorie; in caso contrario viene inviato un errore tramite ActionErrors che permette di mostrare degli avvisi in modo semplice e pulito.

6.2 AMMINISTRAZIONE

L'applicazione presenta un'interfaccia web che cerca di semplificare la gestione del questionario.

All'amministratore vengono fornite le seguenti funzionalità:

- Creazione questionario
- Gestione questionario
 - Modifica dati questionario
 - Aggiunta / modifica / eliminazione modulo
 - Aggiunta / modifica / eliminazione domanda e risposte
- Creazione risposte standard
- Creazione report dei questionari
- Gestione utenti
 - Aggiunta / Eliminazione collegamenti questionario-utente
 - Visualizzazione lista questionario-utente

Nella creazione dei questionari all'amministratore viene data la possibilità di creare da zero un nuovo questionario o di modificarne uno esistente come si può vedere nella Figura 31.

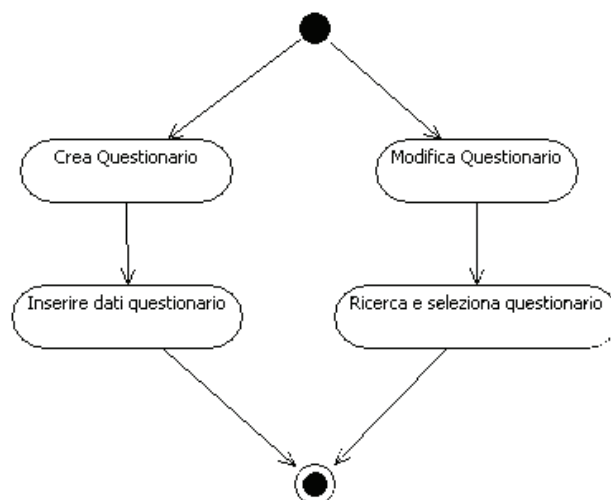


Figura 31 Activity creazione/gestione questionario

Una volta eseguita questa prima operazione verrà presentato il questionario in una pagina web. All'interno della pagina web saranno presenti dei pulsanti che permettono di modificarne la forma inserendo moduli e domande. Vengono date tre possibilità all'amministratore: aggiungere, togliere e modificare moduli (le due ultime solo se qualche modulo è già presente).

L'aggiunta di un nuovo modulo (Figura 32) presenta a sua volta due possibilità: crearne uno nuovo ad hoc o aggiungere un modulo esistente con le relative domande associate. Nel primo caso verrà presentata una pagina dove sarà possibile inserire i titoli del modulo. In questa pagina sarà possibile inserire il testo per tutte le lingue presenti nel sistema; verranno richiesti come obbligatori i testi delle lingue del questionario in modo che ci sia una corrispondenza tra lingue questionario e lingue modulo. Oltre a segnalare all'utente le lingue obbligatorie, il sistema provvederà anche a controllare che l'inserimento sia effettivamente fatto; in caso di controllo negativo verrà segnalato l'errore e non sarà possibile proseguire.

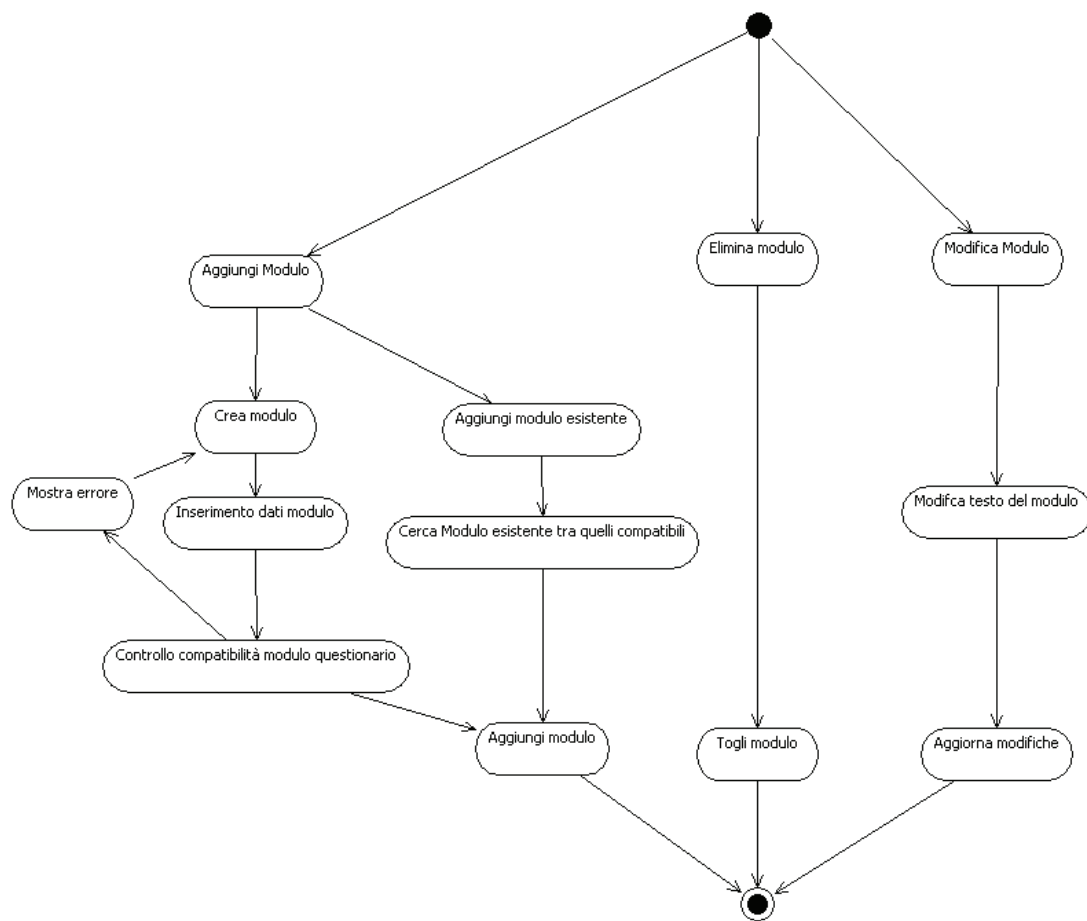


Figura 32 Activity gestione modulo/questionario

L'aggiunta di un modulo esistente, invece, demanda il controllo di compatibilità delle lingue alle query sul database. All'amministratore verranno presentati solo i moduli compatibili con il questionario che si sta gestendo.

La gestione dei moduli (Figura 33) è simile alla gestione del questionario, cioè permette di inserire, aggiungere e modificare le domande di ciascun modulo, ma deve gestire altre eventualità come la presenza del modulo in altri questionari e segnalare il fatto all'amministratore.

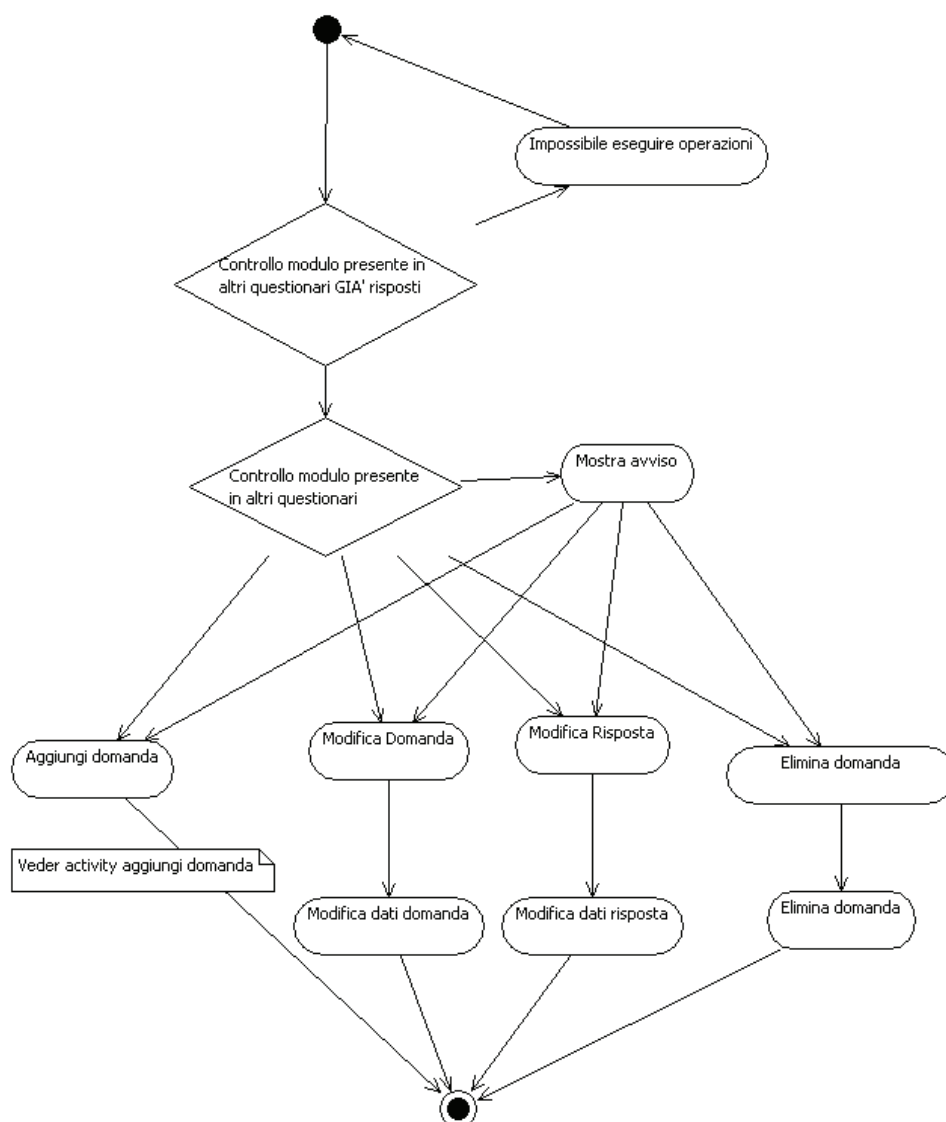


Figura 33 Activity gestione domanda/modulo

Prima di presentare le funzionalità all'amministratore viene controllato che il modulo preso in considerazione non sia presente in altri questionari già risposti. Nel caso di un risultato positivo verrà presentato un avviso e sarà impedita qualsiasi modifica o aggiunta in modo da salvaguardare la struttura del questionario e le relative risposte. In caso contrario verrà controllato se il modulo è presente in un altro questionario, in questo caso verrà presentato solo un avviso per informare che la modifica influenzerà più di un questionario ma non verrà impedita la modifica.



Figura 34 Activity gestione domanda

Questi controlli sono fatti attraverso le query sul database che impostano il valore di determinate variabili all'intero del Bean. In fase di costruzione della pagina JSP queste variabili vengono processate e vengono stampati i vari avvisi e pulsanti.

L'inserimento della domanda all'interno del modulo viene gestita in modo simile all'inserimento di un modulo nel questionario. Nella Figura 34 si possono notare le due differenti opportunità presentate all'amministratore cioè, la creazione di una nuova domanda o l'inserimento di una esistente. Per l'utilizzo di domande già esistenti verranno proposte solamente quelle compatibili con il modulo corrente, prelevandole attraverso delle query che controlleranno le lingue lato database. Per quanto riguarda la

creazione di una nuova domanda verrà presentata una pagina dove sarà possibile inserire i testi della domanda, il tipo delle risposte (aperta, singola e multipla) e il numero minimo di risposte. I testi della domanda dovranno essere compatibili con quelli del modulo; il sistema quindi avviserà e controllerà che siano inseriti i vari testi obbligatori. Il numero minimo di risposte cambia significato a seconda della tipologia della domanda; in una domanda aperta renderà la risposta obbligatoria e nella risposta multipla indicherà il numero minimo di risposte da dare. Successivamente, se il tipo di domanda non è aperto, verrà presentata una pagina dove sarà possibile inserire le risposte solamente per le lingue della domanda o, in alternativa, selezionare un set di risposte standard. Il sistema provvederà a controllare la compatibilità dei testi. Le altre funzionalità presenti per l'amministratore sono quelle della creazione delle risposte standard, con le quali viene data la possibilità di creare un set di risposte da utilizzare senza doverle inserire ogni volta. Viene data la possibilità di creare report per i quali si dovrà selezionare il questionario e il tipo di report da generare. Nel questionario anonimo non verrà data la possibilità di creare report con le risposte collegate ad ogni singolo utente. All'amministratore viene data la possibilità di gestire le associazioni tra contatti e questionari, cioè di decidere a quali contatti permettere di accedere al questionario. Questa funzionalità si traduce nella creazione di un codice che dovrà essere fornito all'utente. Nel caso di questionario anonimo non sarà presentato il collegamento diretto tra utente e questionario ma una casella di testo per indicare il numero di login anonimi da attivare.

6.2.1 APPROFONDIMENTO FUNZIONAMENTO RICERCA

Le varie funzionalità riservate all'amministratore comprendono una serie di ricerche degli oggetti. La ricerca del questionario, ad esempio, viene usata per:

- Ricerca questionario da modificare
- Ricerca questionario da cui ricavare report
- Ricerca questionario a cui aggiungere utenti
- Ricerca questionario a cui togliere utenti.

Anche le ricerche di modulo o di domanda vengono utilizzate per più di uno scopo. Al fine di ridurre il numero di classi, dato che le ricerche per scopi diversi differivano di poche righe, si è deciso di introdurre un metodo per poter utilizzare un'unica classe per ogni oggetto.

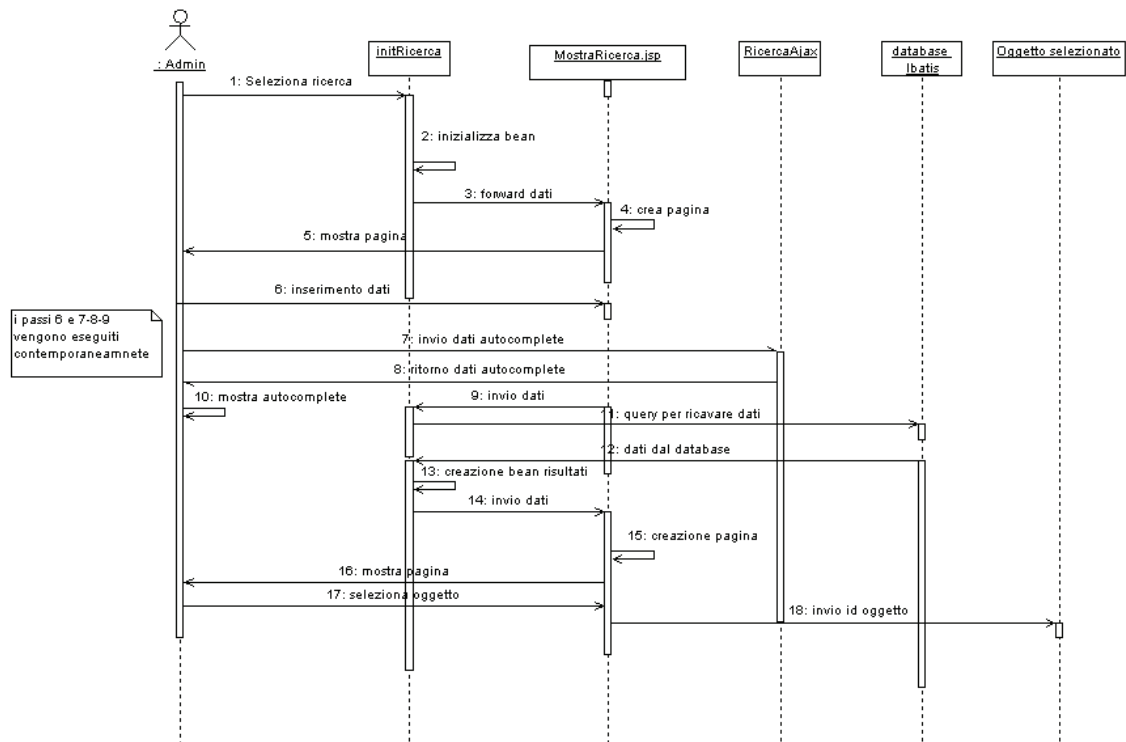


Figura 35 sequence della ricerca.

Per capire il contesto della ricerca, ad esempio se si sta cercando il questionario per modificarlo o per aggiungere utenti, si è scelto di introdurre un parametro che viene passato nei link. Questo parametro viene poi utilizzato dalla classe che effettua il forward finale che viene richiamato dopo che l'utente ha effettuato una scelta tra gli oggetti presentati dalla ricerca. Questa classe opera un semplice forward trasmettendo come attributo nel forward il valore passato all'inizio. Tale parametro è presente nella configurazione dei forward di struts che così richiamerà la prossima azione.

Ogni ricerca si compone di tre classi più una di supporto per Ajax. Una prima classe ha il compito di reperire i dati dal database secondo le richieste dell'utente (cioè il testo digitato) e inserirli nei campi di presentazione del bean. La ricerca dei dati cambia secondo l'utilizzo che se ne farà. All'interno della classe, infatti, si controllerà il valore del parametro settato all'inizio e tale parametro sarà il discriminante per la query da

eseguire. Durante la digitazione della ricerca il sistema farà una chiamata asincrona attraverso l'XMLHttpRequest che permetterà di visionare in tempo reale i possibili risultati così da aiutare nella ricerca di oggetti; questa funzionalità viene comunemente chiamata auto-complete. Con la diffusione su larga scala di Ajax, questa funzionalità viene presentata in quasi tutte le applicazioni che necessitano di inserimento di testi. Una volta inseriti i criteri di ricerca, i dati vengono rimandati alla Servlet che reperisce i dati nel database e inizializza le variabili di presentazione paginando i risultati delle query. La paginazione dei dati consiste nel dividere i dati in gruppi di tot elementi, decisi a priori, in modo che in un'unica pagina non ci sia una lunga lista di dati ma vi siano più pagine con un numero fisso di dati. Una volta che l'amministratore seleziona l'oggetto e invia la richiesta i dati verranno inviati alla Servlet che provvederà al forward come spiegato in precedenza. Questa implementazione permette di riutilizzare la ricerca in più contesti senza dover implementare classi per ogni contesto. Così facendo si rende l'applicazione più estendibile e flessibile.

7 INTERFACCIA DELL'APPLICAZIONE

7.1 AMMINISTRATORE

QuaesTio
QUESTIONARI

Admin
CREAZIONE
crea questionario
crea risposte standard
COSTRUZIONE
costruzione
REPORT
report
UTENTI QUESTIONARIO
aggiungi
rimuovi
lista
LOGOUT
LOGOUT

lingua questionario | |

modelFrontier Partner Meeting

Valutazione del meeting

	ottimo	buono	sufficiente	insufficiente	gravemente insufficiente
Esposizione dei concetti	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Metodo di insegnamento	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Giudizio globale del meeting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

vai al modulo [Valutazione del meeting](#) | [Suggerimenti](#) |

Figura 36 Creazione del questionario

In Figura 36 si può vedere l'interfaccia che appare all'amministratore al momento della creazione del questionario; in questo caso un questionario multilingua (italiano e inglese) dal titolo modelFrontier Partner Meeting.

Si può subito notare che sono presenti due volte le bandierine delle lingue. La prima riga di bandierine, quelle senza alcuna scritta, serve a modificare la lingua di sistema (che attualmente sono italiano e inglese). Con queste si cambiano i testi statici della pagina come avviene, ad esempio, nella Figura 37. Le bandierine della seconda riga (vicino alla scritta "lingua del questionario"), servono per visualizzare il questionario nelle varie lingue in cui è stato scritto, (in questo caso italiano e inglese) senza cambiare anche la lingua di sistema.

Figura 37 Cambio lingua sistema, da italiano a inglese. La lingua del questionario rimane invariata.

Subito sotto alla gestione delle lingue del questionario compare il titolo del questionario affiancato dal pulsante per modificare il titolo . Segue il pulsante per aggiungere il modulo ; questo pulsante compare anche in fondo alla pagina. Il primo serve per aggiungere un modulo prima di quello corrente; il secondo permette di aggiungere un modulo dopo quello corrente.

Immediatamente sotto si può notare il titolo del modulo con accanto i pulsanti per modificare il modulo o eliminarlo .

Dentro il modulo compaiono le varie domande (nella Figura 37 sono domande a risposta singola). Prima e dopo ogni domanda compare il pulsante per aggiungere una domanda nella posizione del pulsante . Vicino al testo di ogni domanda compare il pulsante per modificarla e quello per toglierla dal modulo . Vicino ad ogni risposta è presente il pulsante per modificare il testo della risposta .

QUAESTIO
QUESTIONARI

lingua questionario | |

modelFrontier Partner Meeting

Evaluation of the meeting

	perfect	good	average	poor	bad
Explanation of concepts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Structure of the training	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Overall impression of the meeting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

vai al modulo [Evaluation of the meeting](#) | [Suggestion](#) |

Figura 38 Cambio lingua questionario, da italiano a inglese. La lingua del sistema rimane invariata.

A fondo pagina è presente il menù dei moduli del questionario che permette all'amministratore di visionare i vari moduli. Nel'esempio delle Figura 37 i moduli sono due: Valutazione del meeting e suggerimenti.

Nel secondo modulo (Figura 39) si può vedere come vengono mostrate le altre domande all'amministratore. In particolare si può notare la domanda di tipo aperto e una a risposta multipla.

QUAESTIO
QUESTIONARI

lingua questionario | |

modelFrontier Partner Meeting

Suggerimenti

Commenti generali

Desidero ricevere materiale informativo sulle applicazioni di mio interesse

☐ su supporto cartaceo

☐ per E-Mail

☐ per Telefono

Figura 39 il secondo modulo.

REPORT

All'amministratore è data anche la possibilità di generare report di tre tipologie come si vede nella Figura 40.

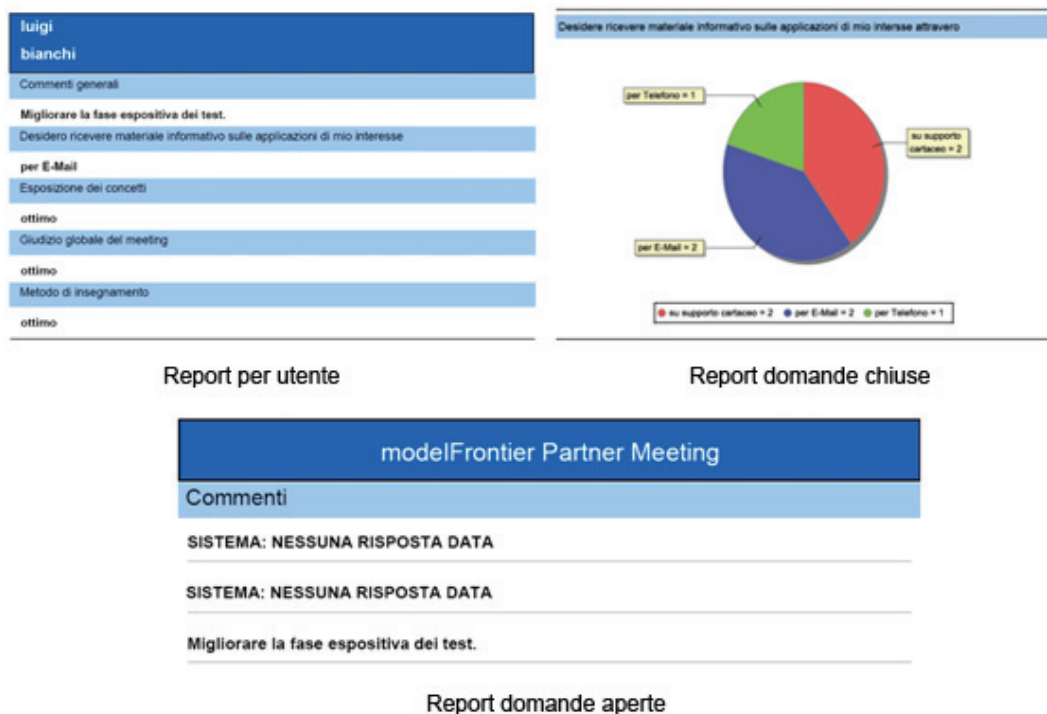


Figura 40 Le tre tipologie di report.

Il primo dei tre report mostra le risposte divise per utente (nell'esempio dell'immagine si mostra un singolo utente). Dopo il nome dell'utente appaiono (su sfondo azzurro) le domande, seguite dalle risposte (su sfondo bianco).

Il secondo report mostra un diagramma a torta delle risposte a domande chiuse. Viene presentato il testo della domanda e subito dopo il grafico in questione con relativa legenda ed etichette.

Il terzo report elenca le risposte aperte. Nell'esempio in figura si nota come due (in quanto la stringa compare due volte) utenti non abbiano risposto per cui compare il testo di sistema che avvisa l'amministratore.

Questi tre report vengono creati all'interno di altrettanti file pdf che l'utente può comodamente salvare attraverso il browser.

7.2 UTENTE

QUAESTIO
QUESTIONARI

(1/2)
50%

modelFrontier Partner Meeting

Valutazione del meeting

E' necessario rispondere a tutte le domande					
	ottimo	buono	sufficiente	insufficiente	gravemente insufficiente
Esposizione dei concetti	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Metodo di insegnamento	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Richiesta					
Giudizio globale del meeting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Rispondi

Figura 41 Risposta al questionario, primo modulo.

All'utente, dopo aver inserito il codice, verrà presentato il questionario. Come si può notare in Figura 41 il menù a sinistra diventerà una barra che indica la percentuale di completamento del questionario calcolata sui moduli completati. Nella figura in questione si può vedere come viene segnalato all'utente l'omessa risposta a una domanda obbligatoria, in questo caso l'utente non ha risposto alla domanda numero tre e quindi non può proseguire.

QUAESTIO
QUESTIONARI

(2/2)
100%

modelFrontier Partner Meeting

Suggerimenti

Commenti generali

Testo del commento dell'utente.
Può essere scritto su più righe.

Desidero ricevere materiale informativo sulle applicazioni di mio interesse

☒ su supporto cartaceo

☒ per E-Mail

☐ per Telefono

Rispondi

Figura 42 il secondo modulo del questionario.

Dopo avere risposto a tutte le domande del primo modulo si può proseguire e quindi rispondere al secondo modulo (Figura 42). Come si può notare in questo modulo compaiono gli altri due tipi di risposta: aperta e a scelta multipla. Nel caso di domanda con risposta aperta multi-riga sarà possibile inserire un testo su più righe; nel caso di una richiesta di risposta multipla sarà possibile selezionare più risposte.

Una volta risposto a tutte le domande verrà mostrato, all'utente, un messaggio per indicare il completamento e la fine delle sue attività.

Il codice utilizzato dall'utente sarà invalidato e quindi non sarà più possibile usarlo.

8 MANUALE D'USO

8.1 INTRODUZIONE

Questo documento nasce dalla necessità di avere una guida sull'utilizzo di Quaestio (QO), un'applicazione web per la gestione di questionari. A differenza di molte altre applicazioni di questo tipo, permette il multilingua e la gestione modulare del questionario.

8.2 INSTALLAZIONE E CONFIGURAZIONE

Per poter eseguire l'applicazione è necessario predisporre di una macchina con il server apache Tomcat ® (consigliata la versione 6) e un database Postgres. Come ambiente di sviluppo è stato usato NetBeans ®.

DATABASE

Per prima cosa è necessario creare un database in Postgres chiamato “quaestio” (consigliata codifica utf8) con username “Postgres” e password “Postgres”. In alternativa, se il database fosse già presente e con altre credenziali, modificare il config.XML del progetto (vedere la sezione modifica config.XML).

Usare i file sql forniti per creare le tabelle del progetto.

Usare i file sql forniti per l'insert dei dati essenziali (come le lingue del sistema, l'utente anonimo e i ruoli del sistema). Nel caso in cui ci fossero problemi con le insert, e non si avesse più una corrispondenza tra gli id nel database e quelli nella classe VariabiliSistema, modificare quest'ultima e ricompilare il progetto (maggiori informazioni nella sezione modica VariabiliSistema).

APPLICAZIONE

L'applicazione già compilata funziona solo con la configurazione spiegata in precedenza e il database creato con i file forniti.

In alternativa si devono modificare i file config.XML o la classe VariabiliSistema ed eventualmente ricompilare.

CONFIGURAZIONI

CONFIG.XML

Il file è presente nella cartella META-INF ed è così strutturato:

```
<?XML version="1.0" encoding="UTF-8"?>
<Context path="/qo">
  <Resource
    auth="Container"
    driverClassName="org.Postgresql.Driver"
    maxActive="8"
    maxIdle="4"
    name="jdbc/QO"
    type="javax.sql.DataSource"
    url="jdbc:Postgresql://localhost:5432/qo_db"
    username="Postgres"
    password="Postgres"
  />
  <Realm
    className="org.apache.catalina.realm.DataSourceRe
    alm"
    dataSourceName="jdbc/QO"
    debug="99"
    localDataSource="true"
    roleNameCol="role_name"
    userCredCol="user_pass"
    userNameCol="user_name"
    userRoleTable="user_roles"
    userTable="users" />
  </Context>
```

La prima sezione riguarda le credenziali di accesso al database.

Nel caso in cui le credenziali d'accesso siano diverse da "Postgres" con password "Postgres" modificare la riga username e password (evidenziate in neretto). Se necessario, modificare anche l'url del database.

La seconda sezione riguarda i dati di accesso realm (leggere la sezione Realm per come configurare gli utenti) ed in particolare dove risiedono i dati di login. Anche in questo caso modificare le righe in grassetto se necessario mettendo i riferimenti alle tabelle che contengono i dati di login.

VARIABILI SISTEMA

La classe VariabiliSistema contiene delle costanti che vengono usate all'interno dell'applicazione. Tali costanti riguardano gli id di alcune componenti all'interno del database. In dettaglio:

Tipo della risposta (id_tipo tabella qo_tipo):

```
public static final int SINGOLO = 1;
```

```
public static final int MULTILPO = 2;
public static final int APERTO_SINGOLO = 3;
public static final int APERTO_MULTIPLO = 4;
```

Lingua (id_lingua tabella qo_lingua) :

```
public static final int ITALIANO = 1;
public static final int INGLESE = 2;
```

Anonimo (id_utente tabella utenti già presente nel db).

```
public static final int ID_ANONIMO = 0;
```

Risultati per pagina

```
public static final int RIS_PER_PAG = 5;
```

E' **necessario** che ci sia corrispondenza tra questa classe e il database.

REALM

Il login, e i diritti d'accesso, vengono effettuati usando Realm di Tomcat ®. Il sistema, per ora, prevede un'unica tipologia di utente loggato con realm: Admin. L'utente loggato avrà l'accesso come amministratore, mentre l'utente che deve rispondere al questionario non necessita di login ma verrà identificato da un sistema creato ad hoc tramite un codice.

AGGIUNGERE UTENTI ADMIN

Per aggiungere un utente admin bisogna agire direttamente sul database. In particolare bisogna:

1. Nella tabella "users" inserire: username, password, e id (cioè l'id corrispondente all'utente nella tabella personale)
2. Nella tabella "user_roles" Inserire username (lo stesso del punto 1) e "Admin" come ruolo.

8.3 USO: AMMINISTRAZIONE

COSE DA SAPERE PRIMA DI INIZIARE

DI COSA È COMPOSTO UN QUESTIONARIO

Gli elementi di un questionario sono:

- Questionario
- Modulo
- Domanda
- Risposta (che appartiene alle domande)

Un questionario è composto da diversi elementi annidati tra loro. Ogni questionario può contenere da zero (vuoto) a più moduli, e ogni modulo appartiene a uno o più questionari. Le domande, invece, appartengono al modulo, ogni modulo ha da zero (vuoto) a più domande che possono essere di diversa tipologia.

Questa gerarchia impone dei vincoli sulle lingue usate all'interno di questi oggetti. Presi due oggetti tra Questionario, Modulo e Domanda, si ha che l'oggetto meno importante deve contenere **ALMENO** le stesse lingue di quello più importante.

ES: Questionario creato con la lingua italiana; il modulo deve avere almeno l'italiano nelle sue lingue.

COME SI SCELGONO LE LINGUE DEL QUESTIONARIO E DEI SUOI COMPONENTI

Il sistema permette la creazione di questionari, anche multilingua. La scelta della lingua per: questionario, modulo, domanda o risposta viene fatto semplicemente inserendo il testo nelle caselle proposte.

ES: Creazione questionario: Viene inserito il titolo in Italiano ma non in inglese, il questionario sarà solo in italiano. Mentre inserendo il titolo in italiano e inglese, il questionario avrà entrambe le lingue.

COME FUNZIONA LA RICERCA

Per facilitare la ricerca, si è introdotto l'autocomplete che permette di mostrare i risultati mentre l'utente sta ancora digitando così da semplificare le ricerche. Nel caso in cui si cerchi un componente con titolo in più lingue verrà mostrato con i titoli divisi, mentre la ricerca lo mostrerà come singolo risultato.

ES: un questionario ha come titolo "Evento 22/12" e "Event 12/22" quando si tenterà la ricerca digitando le lettere "ev" verranno mostrate su due righe diverse i titoli in questione. La successiva ricerca, però, mostrerà un unico risultato.

cerca

evento 22/12

event 12/22

	Italiano	Inglese
<input type="radio"/>	Evento 22/12	Event 12/22

Figura 43 Autocomplete e risultato della ricerca

La ricerca, limita i risultati in base alle richieste. Cercando una domanda di un modulo in italiano, verranno mostrati solo le domande che contengono almeno l'italiano tra le lingue.

Ricerca dell'utente:

La ricerca dell'utente è diversa dalla ricerca normale. Infatti, la stringa deve essere digitata nella forma: "nome – cognome" con "spazio trattino spazio" tra nome e cognome, nel caso in cui si voglia cercare sia in nome che in cognome, altrimenti si inserirà una stringa normale.

ES : Si vuole ricercare "Mario": basterà inserire "Mario" nella casella di testo. Analogo il caso in cui si ricerchi per cognome.

Se si vuole ricercare sia per nome che per cognome bisognerà inserire "Mario – Rossi".

Questo metodo, però, potrebbe non funzionare correttamente se il nome o il cognome contengono spazi.

LOGIN

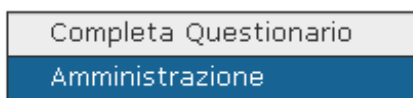


Figura 44 menu

Admin
CREAZIONE
crea questionario
crea risposte standard
COSTRUZIONE
costruzione
REPORT
report
UTENTI QUESTIONARIO
aggiungi
rimuovi
lista
LOGOUT
LOGOUT

Figura 45 Menu amministrazione

Per accedere alla sezione di amministrazione è necessario cliccare su: "Amministrazione" nel menù a sinistra. Una volta effettuata questa operazione verrà presentato il form di login dove inserire il proprio user name e password

MENU

La prima riga del menù mostra il nome utente attualmente in uso (nell'esempio "Admin").

Il menù presenta varie sezioni (v. Immagine) che sono analizzate di seguito.

CREAZIONE

CREA QUESTIONARIO

lingua	titolo
Italiano	<input type="text"/>
Inglese	<input type="text"/>
anonimo	sì <input type="radio"/> no <input checked="" type="radio"/>
<input type="button" value="Invia"/>	

La Creazione permette di creare un nuovo questionario.

Verrà richiesto di inserire un titolo per ogni lingua

Figura 46 creazione questionario

disponibile. L'inserimento del testo comporta l'uso della lingua all'interno del questionario (vedere sezione: cose da sapere).

Anonimo: Serve per scegliere se il questionario sarà anonimo o meno.

Una volta creato il questionario si verrà automaticamente rimandati alla sezione Costruzione

CREA RISPOSTA STANDARD

nome	<input type="text"/>
lingua	titolo
Italiano	<input type="text"/>
Inglese	<input type="text"/>
<input type="button" value="Invia"/>	

Generalmente vengono create come risposte standard quelle risposte che si usano più spesso all'interno dei questionari ad esempio: giudizio (insufficiente, buono, ottimo) o una scala numerica.

Figura 47 creazione risposte standard

Durante la creazione verrà richiesto un nome identificativo per questo insieme di risposte. Oltre a questo è necessario inserire le varie risposte nelle varie lingue, tenendo sempre conto del collegamento dati-lingua (vedere sezione: cose da sapere).

Le risposte verranno divise in base alla riga, cioè, ogni riga rappresenta una risposta e ovviamente le risposte, se inserite in più lingue, dovranno essere dello stesso numero

ES: in italiano inserisco 4 risposte, dovrò inserire 4 risposte anche per la lingua inglese.

Controlli del sistema: Viene controllato che il numero di risposte combaci.

Suggerimento: è consigliabile inserire le risposte standard per tutte le lingue, in modo da poterle utilizzare spesso.

COSTRUZIONE

Questa sezione permette di modificare un questionario già creato. Una volta cliccato sul collegamento verrà presentata la pagina di ricerca (vedere sezione: cose da sapere per maggiori informazioni sulla ricerca). **La ricerca mostrerà, tra i risultati, solo i questionari che non hanno ancora avuto alcuna risposta.** Così facendo si evitano i problemi di incongruenza dei dati nel caso in cui si tenti di modificare un questionario a cui qualcuno ha già risposto. Una volta selezionato il questionario, questo verrà mostrato in modo da poterlo modificare.

PULSANTI

Nell'interfaccia che compare verranno mostrati i pulsanti.

	Questionario	Modulo	Domanda	Risposta
Inserisci				
Elimina				
Modifica				

COME FUNZIONA

La pagina che verrà mostrata è del tipo WYSIWYG (What You See Is What You Get), cioè, quello che si vede è quello che si ottiene. In pratica, all'utente, il questionario verrà presentato esattamente come lo vede l'amministratore durante la creazione ad esclusione dei pulsanti, bandiere delle lingue e menù dei moduli). Se il questionario è vuoto (o è stato appena creato), all'amministratore verrà presentata la pagina seguente:



Figura 48 Questionario vuoto costruzione

questionario è bilingue) in modo da poter visualizzare il questionario in entrambe le lingue (Nota: questa modifica non cambia la lingua di sistema).

Inoltre, vicino al nome del questionario è presente un pulsante per modificare il questionario e uno per aggiungere un modulo.

Come si può notare, sono presenti due volte le bandiere delle lingue. La prima riga di bandiere modifica la lingua di sistema (cioè, verrà cambiata la lingua del menù ecc) mentre la seconda, modifica la lingua del questionario (dato che il

Nel caso in cui il questionario contenga dei dati (cioè, sono già stati inseriti moduli e domande) questi verranno mostrati nella pagina.

QUESTIONARIO: MODIFICA

Cliccando sul pulsante *modifica questionario* si accede alla modifica del questionario.

lingua	titolo
Italiano	<input type="text" value="Evento 22/12"/>
Inglese	<input type="text" value="Event 12/22"/>
anonimo	sì <input type="radio"/> no <input checked="" type="radio"/>
<input type="button" value="Invia"/>	

Con la modifica del questionario sarà possibile: modificare il titolo del questionario e le

Figura 49 Modifica questionario
impostazioni di anonimato del questionario.

Nota: Non è possibile eliminare o aggiungere lingue.

MODULO: AGGIUNTA

cerca modulo esistente

lingue obbligatorie	Italiano Inglese
lingua	titolo
Italiano	<input type="text"/>
Inglese	<input type="text"/>
<input type="button" value="Invia"/>	

Con il pulsante *aggiungi modulo* si aggiungono moduli.

È possibile creare un nuovo modulo o cercarlo in quelli esistenti.

Figura 50 Aggiunta modulo

Se si vuole inserire un modulo nuovo bisogna inserire **i titoli ALMENO per le lingue obbligatorie**. Nulla vieta di inserire anche i titoli per altre lingue. Così facendo però, le domande che verranno collegate al modulo dovranno contenere ALMENO tutte le lingue del modulo, come succede per questionario-modulo.

ES: Il questionario è in italiano; si vuole creare un modulo in italiano e inglese. Le nuove domande che verranno collegate al modulo dovranno essere (almeno) in italiano e inglese. All'interno di questo questionario (che è solo in italiano) verranno mostrate solo le domande in italiano.

Se si sceglie di usare un modulo già esistente (e di conseguenza tutte le domande che contiene) si deve cliccare sul collegamento “cerca modulo esistente” che rimanderà alla

pagina di ricerca dei moduli. Tale **ricerca, mostrerà solo i moduli compatibili con il questionario, cioè, quelli che contengono ALMENO le lingue del questionario.**

ES: nel caso di un questionario scritto in italiano, e avendo nel sistema le lingue italiano e inglese, la ricerca mostrerà tutti i moduli scritti in italiano e anche quelli scritti in italiano e inglese, ma non quelli scritti esclusivamente in inglese.

Inserito un modulo, sul fondo della pagina, comparirà la lista dei moduli presenti in modo tale da potersi spostare tra un modulo e l'altro

NB: Nel caso in cui si inseriscano moduli presenti in altri questionari a cui nessuno ha ancora risposto, verrà segnalato che la modifica del modulo (aggiunta/eliminazione domande ecc) influisce su più di un questionario. Se il modulo è presente in questionari che hanno già avuto risposte, allora, non sarà possibile modificarli.

Il modulo è presente in questionari GIA' RISPOSTI, quindi NON può essere modificato

Figura 51 Avviso: non è possibile modificare il modulo, quindi non si può aggiungere o togliere domande. (modulo in questionari già risposti)

Il modulo è presente in altri questionari, modificandolo si modificheranno anche gli altri questionari

Figura 52 Avviso: la modifica influenza più questionari (modulo in altri questionari non risposti)

CONTROLLI: Il sistema controllerà che le lingue del modulo e questionario siano compatibili.

MODULO: ELIMINAZIONE

Con il pulsante *elimina modulo* si toglie il modulo dal questionario e di conseguenza tutte le domande che contiene.

MODULO: MODIFICA

Con il pulsante *modifica modulo* si accede alla modifica del modulo. Qui è possibile modificare il titolo del modulo, ma non aggiungere o togliere lingue al modulo.

DOMANDA: AGGIUNTA

Anche qui, come nell'aggiunta dei moduli, è possibile o ricercare una domanda esistente o crearne una nuova.

Se si desidera inserirne una già esistente (con relative risposte) allora bisogna cliccare su "cerca domanda esistente". Così facendo, si verrà rimandati alla pagina di ricerca delle domande che funziona nello stesso modo della ricerca questionario-modulo per quanto riguarda la parte di compatibilità delle lingue.

cerca domanda esistente

tipo	<input type="text" value="singolo"/>
lingue obbligatorie:	Italiano Inglese
lingua	titolo
Italiano	<input type="text"/>
Inglese	<input type="text"/>
numero di risposte minime*	<input type="text" value="0"/>

Figura 53 Creazione domanda

Se invece si desidera inserire una domanda nuova, allora, si dovrà scegliere il tipo della domanda dal menù a

tendina. I tipi possibili sono quattro:

- Singolo
- Multiplo
- Aperta singola linea
- Aperta multi linea

Successivamente è necessario inserire il testo della domanda (almeno) per le lingue obbligatorie.

In oltre è possibile inserire il numero di risposte minime obbligatorie, in questo caso il significato del valore cambia a seconda del tipo della domanda.

- Singolo: il valore non ha significato, la risposta è obbligatoria ed al massimo se ne può dare una.
- Multipla: indica il numero minimo di risposte consentito.
- Aperta: se il valore è maggiore di 0 la risposta diventa obbligatoria.

Una volta completato il tutto, se la domanda è aperta, si verrà rimandati alla costruzione del questionario e la domanda appena creata sarà visibile all'interno del modulo. Se invece la domanda è di tipo singolo o multiplo si verrà mandati all'inserimento delle risposte.

Controlli: Lingue obbligatorie e lingue inserite

RISPOSTA

Se la domanda è di tipo "singola" o "multipla" si verrà rimandati all'inserimento delle risposte. Qui, verrà evidenziato in grassetto il tipo della domanda e il numero di risposte minime per ricordare questi dati a chi sta costruendo il questionario. Verrà poi mostrato un menù a tendina in cui si può selezionare un set di risposte standard. In questo caso, non serve completare le textarea sottostanti in quanto il loro contenuto non viene

considerato. L'elenco di risposte standard comprende solo le risposte standard

tipo	singolo
numero di risposte minime	0

risposte standard

Scegli risp standard ▼

inserire risposte per le lingue	
Italiano domanda	
<input type="text"/>	
Inglese question	
<input type="text"/>	
<input type="button" value="Invia"/>	

compatibili con le lingue della domanda.

Nel caso non si selezioni un set di risposte standard, sarà necessario riempire tutte le textarea presenti (sono mostrate solo le textarea per le lingue della domanda). La

Figura 54 Creazione delle risposte

risposta deve essere inserita su una singola linea.

ATTENZIONE: Ogni “a capo” viene considerato come una risposta diversa.

ES: inserendo:

Prova1

Prova2

Nella casella di testo di italiano, si avranno due risposte. Inserendo Prova1 Prova2 (sulla stessa linea) si avrà una risposta unica.

Controlli: Viene controllato l'inserimento di almeno una risposta per lingua. Inoltre viene controllato che il numero di risposte per le lingue combaci e che sia maggiore del numero minimo di risposte obbligatorie scelto in precedenza (nel caso di domanda a risposta multipla).

DOMANDA: ELIMINAZIONE

Con il pulsante *elimina domanda* si toglie la domanda dal modulo

DOMANDA: MODIFICA

Il pulsante *modifica domanda* dà l'opportunità di modificare la domanda. Con questa

tipo	singolo ▼
lingua	titolo
Italiano	domanda
Inglese	question
numero di risposte minime	0
<input type="button" value="Invia"/>	

Figura 55 modifica domanda

azione, oltre alla modifica della domanda, si può modificare anche il numero minimo di risposte e il tipo

della domanda.

Attenzione: è consentito modificare il tipo della domanda. Si può quindi passare dal tipo “aperta” a “singola” o “multipla”, così facendo però, verrà creata una domanda senza alcuna risposta possibile. Quindi inutilizzabile.

RISPOSTA: MODIFICA

Con il pulsante *modifica risposta* si può modificare il testo della risposta.

REPORT

La sezione report, consente di creare dei *pdf* riassuntivi dei dati del questionario. Una volta selezionata questa opzione si verrà rimandati alla ricerca del questionario di cui si vogliono estrarre i dati.

La ricerca, mostrerà solo i questionari che hanno delle risposte.

Una volta selezionato il questionario, verranno mostrati 3 tipi di report:

- **Riassuntivo per utente**
Questo report crea un elenco per utente di tutte le risposte date al questionario in forma testuale (cioè, verrà mostrato il testo di domanda e risposta).
Nota: Il report in questione è disponibile solo per questionario **NON** anonimo
- **Riassuntivo domande chiuse**
Questo report crea dei grafici a torta, con percentuali, di ogni domanda a risposta chiusa (singola o multipla)
- **Riassuntivo domande aperte**
Verrà creato un elenco di tutte le risposte date alle domande aperte del questionario.

NB: il report verrà creato usando la lingua di sistema usata dall'utente. Se l'utente usa una lingua non compatibile allora verrà presa la prima lingua disponibile.

ES: l'utente utilizza il sistema in inglese, ma il questionario è stato scritto in italiano, verrà creato un report con i testi in italiano.

I report sono creati usando JasperReport ®, è quindi possibile modificarne la struttura editando i file e riposizionando i file compilati nella cartella dei report.

UTENTI QUESTIONARIO

Questa sezione permette di collegare gli utenti al questionario. Il collegamento questionario-utente produrrà un codice che identifica in modo univoco l'utente e il questionario. Tale codice serve all'utente per effettuare un login nel sistema e rispondere al questionario.

AGGIUNGI

Questa azione permettere di aggiungere utenti al questionario. Come prima cosa bisogna ricercare il questionario che si desidera utilizzare. Una volta selezionato, si procederà all'inserimento dell'utente. Se il questionario è di tipo anonimo, si verrà rimandati ad una pagina che chiede quanti utenti anonimi creare. Se il questionario, invece, non è anonimo, si passerà alla ricerca dell'utente. La ricerca mostra gli utenti che non sono già collegati con il questionario (vedere sezione cose da sapere per come funziona la ricerca utente).

NB: Si può selezionare più di un utente. La selezione multipla funziona solo se gli utenti sono presenti nella stessa pagina.

ES: Si ricercano gli utenti Mario. A pagina uno è presente l'utente Mario Bianchi che viene contrassegnato. L'admin cambia pagina (pagina 2) dove è presente Mario Rossi, anche lui viene contrassegnato e si invia la richiesta per creare il codice. Sarà inserito solo Mario Rossi e non anche Mario Bianchi.

RIMUOVI

Questa azione permette di rimuovere un contatto se è stato erroneamente aggiunto. Il funzionamento è simile ad "aggiungi". Vengono mostrati solo i contatti che sono collegati con quel questionario.

LISTA

Mostra la lista dei contatti. Dopo aver selezionato il questionario tramite ricerca sarà stampata la lista a video. È possibile stampare la lista in pdf.

LOGOUT

Il logout dal sistema.

8.4 USO: UTENTE

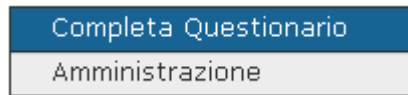


Figura 56 Menu

Inserire codice

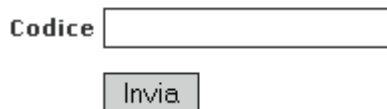
A form for entering a code. It features a label 'Codice' in bold, followed by a text input field. Below the input field is a button labeled 'Invia'.

Figura 57 Inserimento codice

L'utente, per poter completare il questionario, deve disporre del codice fornito dall'amministratore.

NB: La lingua di presentazione del questionario viene impostata in base a quella del sistema al momento dell'inserimento del codice. Una volta iniziato il questionario non è possibile

modificarle la lingua a meno di non ricominciare il questionario.

Inserito il codice, il sistema procederà ad una verifica e se positiva presenterà le domande.

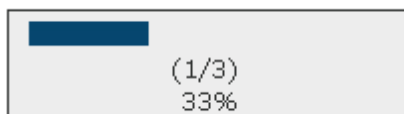


Figura 58 Il menu cambia durante il completamento

Il menu mostrerà la percentuale di completamento (che è calcolata in base ai moduli, cioè alle pagine del questionario).

Al termine del questionario verrà mostrato un avviso di fine questionario e il codice non sarà più utilizzabile.

NB: Non è prevista la modifica del questionario una volta completato. All'utente non è data la possibilità di confermare o meno il suo questionario, una volta completato verrà automaticamente inserito nel database.

Controlli: il sistema controlla che le risposte obbligatorie e il numero minimo di risposte sia rispettato. Inoltre impedisce di modificare le risposte date tornando indietro con il browser. L'unico modo di modificare le risposte è di rifare il questionario prima di completarlo. Le risposte vengono salvate solo al completamento del questionario.

8.5 AGGIUNTA COMPONENTI

In questa sezione viene spiegato come modificare gli elementi interni del sistema.

AGGIUNTA LINGUA

L'aggiunta di una lingua richiede una minima conoscenza dell'ambiente java e html.

Per aggiungere una lingua al sistema bisogna eseguire le seguenti operazioni:

- Modificare l'application.resource inserendo per il nuovo locate le stringhe tradotte.
- Inserire nel database l'id e nome della lingua (l'id deve essere sequenziale a quelli già presenti, senza lasciare buchi.)
- Modificare la classe VariabiliSistema inserendo una nuova costante con il nome della lingua.
- Creare un'immagine con la bandiera della nazione e rinominarla con nome uguale all'id nel database (se la lingua ha id 3 l'immagine sarà "3.gif") e inserirla nella cartella delle bandiere.
- Modificare la JSP /jps/utente/lingue.JSP aggiungendo un href per la nuova lingua copiando e modificando quelli già presenti.
- Modificare la JSP /jps/admin/lingue.JSP aggiungendo un href per la nuova lingua copiando e modificando quelli già presenti.
- Modificare la action i it.enginsoft.qo.struts.admin.CambiaLingua.java ed aggiungere un case nello switch copiando e modificando quelli già presenti.
- Modificare l'action i it.enginsoft.qo.struts.utente.CambiaLingua.java ed aggiungere un case nello switch copiando e modificando quelli già presenti.

Dopo aver eseguito queste operazioni ricompilare il progetto.

Non è possibile modificare ed aggiungere lingue a domande, questionario e moduli già creati. Per far questo bisogna crearne di nuovi.

9 CONCLUSIONI

Sentite le richieste e dopo aver verificato che le numerose applicazioni esistenti non soddisfacevano completamente l'azienda, si è deciso di implementare un'applicazione ad hoc.

Lo sviluppo dell'applicazione ha assorbito una discreta quantità di tempo ripartito in più periodi.

Il primo periodo è stato dedicato alla definizione dell'applicazione, cioè alla raccolta delle richieste, alla creazione degli use case e alla definizione del database. La costruzione della base di dati ha avuto, in corso di progetto, diverse modifiche. Per esempio: nella prima versione non era presente l'oggetto modulo e ogni questionario doveva essere necessariamente associato ad un evento. L'introduzione del modulo ha permesso di avere una struttura più flessibile; l'eliminazione del collegamento tra questionario ed evento permette di riutilizzare l'applicazione anche in altri contesti al di fuori della realtà dell'azienda.

Dopo la definizione del database e dei requisiti si è passati all'implementazione dell'applicazione. Inizialmente gran parte del tempo è stato speso per comprendere il funzionamento dei vari framework grazie anche all'aiuto di persone presenti in azienda che mi hanno aiutato a colmare queste lacune.

La prima versione dell'applicazione presentava molte limitazioni per quanto riguarda l'usabilità. Le risposte erano presentate una per pagina, questo comportava numerosi click e cambio di pagina per rispondere ad un questionario; anche la creazione dei questionari non era di semplice realizzazione. Non era presente, infatti, l'interfaccia grafica che mostrava il questionario all'amministratore esattamente come lo vede l'utente. I collegamenti di questionario, modulo e domande avvenivano solo tramite ricerche e il questionario creato si poteva vedere solo cliccando su un apposito link che generava la struttura, il che rendeva la costruzione molto laboriosa e poco efficiente. Solo ultimamente l'applicazione è stata integrata con i sistemi dell'azienda a causa del trasferimento dell'azienda e per problemi di comunicazione. Attualmente si stanno raccogliendo le prime impressioni e commenti per migliorarne l'efficienza e usabilità; si è alla ricerca di bug in modo da poterli correggere il prima possibile.

Nelle prossime versioni, molto probabilmente, sarà sostituito il metodo di autenticazione che passerà dal Realm ad un server LDAP con tutti i vantaggi che ne comporta. Saranno introdotte altre funzionalità come la possibilità di generare report in altri formati diversi dal file pdf, sempre grazie all'utilizzo del framework JasperReport. Con ogni probabilità sarà introdotta una funzionalità per poter aggiungere a questionari già creati dei testi in un'altra lingua. Verrà migliorata la parte di gestione dei login, sia lato amministrazione che lato utente. Per quanto riguarda l'amministrazione dell'applicazione saranno implementati i vari attori presenti negli usecase con relative funzionalità in modo da poter avere una migliore gestione dell'applicazione. Lato utente sarà migliorata la gestione di collegamento utente-questionario-evento probabilmente introducendo l'inserimento automatico degli utenti quando si associa un questionario ad un evento.

Lo sviluppo dell'applicazione web permette di proporre i questionari ad una vasta gamma di utenti. Sarà infatti possibile rispondere in qualsiasi momento e da qualsiasi posto connesso con un pc alla rete. Permette inoltre di avere un metodo centralizzato, semplice ed efficace di aggiornamento del software in modo centralizzato.

Questa applicazione può essere utilizzata in tutti quei contesti che hanno la necessità di raccogliere dei dati al fine di ricavare le impressioni dei propri utenti. Per com'è stata concepita copre il ramo dei questionari generati per la raccolta di giudizi non avendo lo scopo di giudicare il candidato.

La struttura dell'applicazione è abbastanza flessibile da poter essere integrata, con opportune modifiche, all'interno di quasi tutti i sistemi esistenti che rispettino le caratteristiche di hardware e software di base per questa applicazione.

10 BIBLIOGRAFIA

1. **Wikipedia.** Punto interrogativo. *Wikipedia, the free encyclopedia*. [Online] <http://it.wikipedia.org/>.
2. —. Lightweight Directory Access Protocol. *Wikipedia, the free encyclopedia*. [Online] http://it.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol.
3. —. AJAX. *Wikipedia, the free encyclopedia*. [Online] <http://it.wikipedia.org/wiki/AJAX>.
4. **Altese, Marco.** Uno sguardo d'insieme su J2EE. *Html.it*. [Online] <http://java.html.it/articoli/leggi/2021/uno-sguardo-dinsieme-su-j2ee/>.
5. —. Autenticazione nelle applicazioni web. *Html.it*. [Online] <http://java.html.it/articoli/leggi/2161/autenticazione-nelle-applicazioni-web/>.
6. **Zinato, Matteo.** Guida JSP. *Html.it*. [Online] <http://java.html.it/guide/leggi/23/guida-jsp/>.
7. **Pascuzzi, Simone.** Introduzione ad Apache Struts. *Html.it*. [Online] <http://java.html.it/articoli/leggi/2176/introduzione-ad-apache-struts/>.
8. **Congiusti, Pasquale.** Guida J2EE. *Html.it*. [Online] <http://java.html.it/guide/leggi/136/guida-j2ee/>.
9. **Carri, Antonio.** Guida Application server. *Html.it*. [Online] <http://server.html.it/guide/leggi/102/guida-application-server/>.
10. **Apache Software Foundation.** Struts. *Struts*. [Online] <http://struts.apache.org/>.
11. —. Ibatis. *Ibatis*. [Online] <http://ibatis.apache.org/>.
12. —. Apache Tomcat 6.0. *Apache Tomcat Documentation*. [Online] <http://tomcat.apache.org/tomcat-6.0-doc/>.
13. **JasperSoft Corporation.** JasperReports. *JasperReports*. [Online] <http://jasperforge.org/sf/projects/jasperreports>.