

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Triennale in Informatica

**SISTEMA WEB BASED PER LA
SOMMINISTRAZIONE DI
QUESTIONARI ONLINE
PER LA REUMATOLOGIA**

Tesi di Laurea in Basi di Dati

Relatore:
Chiar.mo Prof.
Danilo Montesi

Presentata da:
Nicola Farina

Sessione Terza
Anno Accademico 2009 - 2010

*A Silvia,
Barbara,
Giorgio,
Stefano*

Introduzione

«Un questionario è uno strumento di ricerca composto di una serie di domande progettate per raccogliere elementi di conoscenza dalle persone che lo compilano»(wikipedia).

Il fondamento teorico dell'uso dei questionari è nella *item response theory* (IRT) che fa parte dell'ambito disciplinare della *psicometria*.

Nel campo della medicina i questionari di valutazione sono diventati oggi gli strumenti più utilizzati per valutare variabili quali la disabilità, il dolore, lo stato di salute o l'efficacia di una terapia [1]. A prima vista può sembrare azzardato che si cerchi di misurare in modo oggettivo quantità soggettive e invisibili (si dicono infatti anche variabili *latenti*) come il dolore. Tuttavia è stato dimostrato che, grazie all'uso di modelli matematici come il Rasch, si riescono ad ottenere dati significativi, dal punto di vista clinico, anche da queste fonti altrimenti poco o per nulla esposte alla misurazione scientifica. In medicina i questionari attraversano un vero e proprio processo di sviluppo, che parte dalla definizione dell'ambito di applicazione, la scrittura, il test, la verifica e infine la validazione.

A volte, per oggetti di studio molto importanti come il dolore, vengono sviluppati più tipi di questionari [2].

I questionari hanno il vantaggio di essere strumenti pratici e di facile utilizzo. Possono essere autocompilati, oppure utilizzati in interviste faccia a faccia o telefoniche. Prima con la diffusione dei personal computer e poi di internet tra un pubblico sempre più vasto, l'uso dei questionari si è spostato, dal semplice foglio di carta, verso forme informatizzate; i vantaggi sono quelli

tipici di questo genere di operazioni: l'immediata disponibilità dei dati, la standardizzazione, la velocità di trasferimento e di elaborazione.

Un ostacolo è senz'altro l'interfaccia: per quanto progettata per essere semplice, difficilmente è così intuitiva come un foglio di carta, soprattutto in caso di persone anziane o con deficit cognitivi.

È, questa, allo stesso tempo una sfida e un'opportunità per i progettisti di software e interfacce grafiche.

Inoltre questa situazione è probabilmente destinata a cambiare con la crescita delle nuove generazioni, abituate alle metafore informatiche fin quasi dalla culla.

In questa tesi presentiamo *Emma*, un sito web progettato per creare e somministrare questionari clinici nell'ambito della reumatologia.

Indice

Introduzione	i
1 Un sito web per questionari clinici	1
1.1 Il questionario PRO-CLARA	1
1.2 Requisiti del sistema e casi d'uso	4
1.2.1 Somministrare un questionario	5
1.2.2 Lettera al medico curante	6
1.2.3 Archivio dei risultati	6
2 La base dati	9
2.1 Schema concettuale	9
2.2 Schema logico	10
2.2.1 Nomenclatura	10
2.2.2 Principi generali di progettazione	11
2.2.3 Tabelle	12
2.2.4 Viste	18
3 Il sito Emma	23
3.1 Scelte tecnologiche	23
3.2 Nomenclatura	24
3.3 Principi generali di progettazione	24
3.4 Accessibilità e compatibilità	25
3.5 Sicurezza	25
3.6 Struttura interna	26

3.7	Struttura visibile	27
3.7.1	Il menù Tabelle	28
3.7.2	Il menù Questionari	29
3.7.3	Il menù Amministrazione	33
4	Listati dei programmi	41
4.1	Database	41
4.1.1	Creazione tabelle e viste	41
4.1.2	Stored procedure	50
4.1.3	Caricamento dati di prova	66
4.2	Sito web	82
4.2.1	Root directory	82
4.2.2	Gestione login	101
4.2.3	Amministrazione	105
4.2.4	Classi di sistema	116
4.2.5	Gestione questionari	188
4.2.6	Gestione tabelle	242
	Conclusioni	271
	A Il database MySQL	273
	B Il framework ASP.NET	275
	Bibliografia	279

Elenco delle figure

2.1	Schema concettuale	9
2.2	Schema logico	21
3.1	Mappa del sito	28
3.2	Pagina home	29
3.3	Pagina login	30
3.4	Pagina pazienti	31
3.5	Pagina struttura questionario	32
3.6	Pagina scegli questionario e paziente	33
3.7	Pagina domanda questionario	34
3.8	Pagina fine questionario	35
3.9	Pagina lettera	35
3.10	Pagina intestazione lettera	36
3.11	Pagina archivio	36
3.12	Pagina archivio raggruppato	37
3.13	Pagina grafico 2D	37
3.14	Pagina grafico 3D	38
3.15	Pagina archivio esporta	38
3.16	Pagina archivio Excel	39
3.17	Pagina gestione utenti	39
3.18	Pagina log di sistema	40

Elenco delle tabelle

1.1	Tabella coefficienti calcolo risultati PRO-CLARA	3
2.1	Tabelle utente e profilo	13
2.2	Tabelle stato civile, titolo di studio e paziente	15
2.3	Tabelle struttura questionario	16
2.4	Tabelle risultati questionario	17
2.5	Vista archivio	18
2.6	Vista testata questionari	19
2.7	Vista dettaglio questionari	19
2.8	Vista pazienti	20

Capitolo 1

Un sito web per questionari clinici

L'obiettivo del progetto è quello di mettere a disposizione un sistema per somministrare il questionario PRO-CLARA, sviluppato da un gruppo di reumatologi dell'università di Ancona guidati dal professore Fausto Salaffi. Si tratta di un questionario studiato per indagare lo stato di salute e di mobilità nei pazienti affetti da disturbi reumatologici (artriti ecc.). La forma del sito web è quella che garantisce la maggior flessibilità agli utenti, sia dal punto di vista della fruizione sia dell'impegno delle risorse hardware per ospitare l'applicazione. In tal modo, se gli utenti non dispongono di un proprio server, potranno usufruire del sistema da remoto. Per la consultazione l'unico vincolo richiesto è quello di disporre di un computer di qualunque marca o tipo dotato di un browser web. Il sito è anche consultabile da dispositivi palmari (smartphone, tavolette ecc.).

1.1 Il questionario PRO-CLARA

Il questionario PRO-CLARA (*Patient Reported Outcomes - CLinical ARthritis Activity*) è un cosiddetto indice composito di attività di malattia diviso in tre sezioni [3], ognuna dedicata a uno dei seguenti parametri clinici:

1. la disabilità funzionale, valutata mediante il *Recent Onset Arthritis Disability Activity questionnaire - ROAD*.
2. il dolore misurato con l'indice autovalutato *Rheumatoid Arthritis Disease Activity Index - RADAI*.
3. l'autovalutazione dello stato globale di salute.

La prima sezione, il questionario ROAD, è composta di 12 quesiti, concernenti la funzione degli arti superiori e inferiori, e le attività della vita quotidiana e lavorativa. Ogni quesito consente 5 possibili risposte, riguardo al grado di difficoltà che comporta l'azione richiesta:

0. nessuna difficoltà
1. lieve difficoltà
2. moderata difficoltà
3. marcata difficoltà
4. totale incapacità

La seconda sezione è composta di 16 domande relative al livello di dolore percepito alle articolazioni (mano, polso, gomito), 8 per la parte sinistra e 8 per la parte destra del corpo. Qui le risposte possibili sono 4:

0. assente
1. lieve
2. moderato
3. forte

La terza e ultima sezione contiene una singola domanda che chiede di specificare l'effetto che la malattia ha avuto, sul paziente, nell'ultima settimana. La risposta è un numero scelto nel range tra 0 e 10, dove 0 indica "nessun

effetto” e 10 “il peggior effetto”. Al questionario è associato un punteggio (globale e di ogni sezione); quello globale è compreso tra 0 (assenza di disabilità funzionale) e 10 (massima disabilità funzionale). Il punteggio si calcola, per le prime due sezioni, in questo modo: si calcola la somma delle risposte date, ottenendo così un numero tra 0 e 48; il risultato è quindi dato prendendo il valore, nella cella corrispondente, dalla tabella 1.1:

1: 0.2	17 : 3.5	33 : 6.9
2: 0.4	18 : 3.8	34 : 7.1
3: 0.6	19 : 4.0	35 : 7.3
4: 0.8	20 : 4.2	36 : 7.5
5: 1.0	21 : 4.4	37 : 7.7
6: 1.3	22 : 4.6	38 : 7.9
7: 1.5	23 : 4.8	39 : 8.1
8: 1.7	24 : 5.0	40 : 8.3
9: 1.9	25 : 5.2	41 : 8.5
10: 2.1	26 : 5.4	42 : 8.8
11: 2.3	27 : 5.6	43 : 9.0
12: 2.5	28 : 5.8	44 : 9.2
13: 2.7	29 : 6.0	45 : 9.4
14: 2.9	30 : 6.3	46 : 9.6
15: 3.1	31 : 6.5	47 : 9.8
16: 3.3	32 : 6.7	48 : 10

Tabella 1.1: Tabella coefficienti calcolo risultati PRO-CLARA

Per la terza sezione, invece, il risultato viene calcolato dividendo il numero, associato alla risposta data, per il coefficiente 3. Il risultato finale di tutto il questionario è dato dalla somma dei risultati delle tre sezioni.

1.2 Requisiti del sistema e casi d'uso

I requisiti identificati in fase di analisi sono i seguenti:

- Il sistema non dovrà essere aperto ma richiedere l'accesso tramite codice utente e password.
- Gli utenti potranno essere medici oppure infermieri. Medici o infermieri dovrebbero avere diritti di accesso differenziati, nel senso che alcune funzioni, per esempio la modifica delle domande del questionario, potrebbero essere riservate ai medici e invisibili agli altri. Per il momento non è stato specificato con esattezza quali funzioni dare agli uni o agli altri. È sufficiente che l'architettura dell'applicazione consenta di ottenere questo tipo di risultato.
- Il sito dovrebbe mettere a disposizione un archivio dei pazienti dove, per ogni paziente, dovrebbero essere memorizzati almeno i seguenti dati:
 - nome.
 - cognome.
 - data di nascita.
 - sesso.
 - codice fiscale.
 - titolo di studio.
 - stato civile.
 - nome, cognome e recapito mail del medico curante.
- Prevedere un meccanismo per far sì che ogni utente possa vedere e gestire solo i propri pazienti, senza vedere i pazienti gestiti dagli altri utenti.

Il requisito più importante è la possibilità di somministrare il questionario PRO-CLARA a un paziente, ma non si esclude di poter utilizzare il sistema anche per altri questionari con una struttura compatibile.

1.2.1 Somministrare un questionario

- Le domande, di cui è composto il questionario, devono essere mostrate all'utente una alla volta.
- Ogni domanda deve occupare l'intera schermata, mostrando i seguenti elementi:
 - il testo della domanda vera e propria.
 - un'immagine, associata alla domanda, che ha lo scopo di rendere più chiaro il contesto (questo tipo di patologie colpisce spesso persone anziane o con deficit cognitivi).
 - un controllo che permetta al paziente di scegliere tra le possibili risposte definite.
 - il numero progressivo della domanda all'interno del questionario.
 - eventualmente anche il nome del questionario e la sezione corrente.
 - un pulsante per passare alla domanda successiva: si intende che non deve essere possibile passare alla domanda successiva senza prima aver risposto a quella corrente.
- Le risposte date devono essere memorizzate, assieme ad una marca temporale e ai dati del paziente che partecipa al test.

Al momento si immagina che l'utente della procedura, medico o infermiere, abbia a disposizione una maschera in cui scegliere:

- il questionario da somministrare e
- il paziente e un pulsante, premendo il quale
- far partire il processo di compilazione delle risposte che saranno fornite dal paziente.

Si immagina quindi che il paziente sia seduto a fianco dell'utente del sistema (medico o infermiere), nel caso la fruizione avvenga su un personal

computer; nel caso di un dispositivo portatile, l'utente potrebbe predisporre il sistema e poi passare direttamente lo strumento al paziente che provvederà a compilare le risposte e poi a restituirlo.

Una possibile evoluzione potrebbe consistere nel dare la possibilità direttamente al paziente di connettersi e, in un certo senso, auto-somministrarsi un questionario. Questa funzione al momento è però stata tralasciata riservandola per una eventuale versione futura del sito.

1.2.2 Lettera al medico curante

Una volta completato il questionario deve essere costruito uno scheletro di lettera da inviare al medico curante del paziente:

- La lettera deve contenere il nome del paziente, la data e l'ora del test e i risultati ottenuti.
- L'utente deve poter completare la lettera, aggiungendo o modificando liberamente anche la parte preimpostata.
- La lettera, così modificata, dovrà poter essere salvata in archivio (per poter essere, eventualmente, richiamata in futuro) oppure inviata via mail al medico curante, oppure stampata in versione PDF.

1.2.3 Archivio dei risultati

I risultati dei questionari compilati devono essere mantenuti in un archivio facilmente consultabile. Dovrebbe inoltre essere possibile:

- scaricare i risultati in un file da elaborare in locale (in formato Excel, ad esempio).
- riaprire la lettera associata ad ogni test.
- visualizzare i risultati dei test in forma grafica.

- trovare in modo semplice i risultati dei test effettuati in un dato giorno oppure di un certo paziente.

Capitolo 2

La base dati

Vediamo come è strutturata la base dati dell'applicazione.

2.1 Schema concettuale

Consideriamo i concetti di *medico* e *infermiere*, incontrati nell'analisi dei requisiti, come specializzazioni di quello di *utente*. Il *questionario* contiene le *domande*. L'*utente* somministra il *questionario* al *paziente*, che, rispondendo alle domande, alimenta l'archivio dei risultati.

La figura 2.1 mostra il modello concettuale così ottenuto in forma grafica.

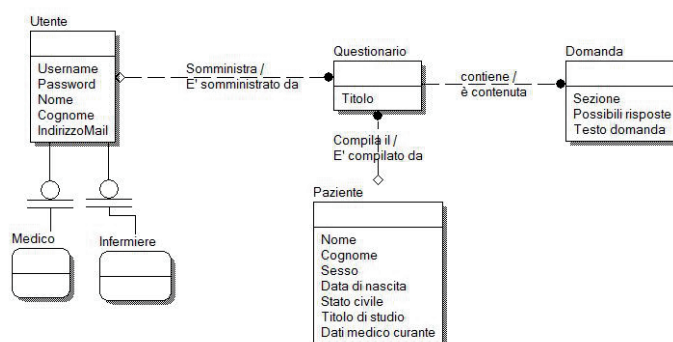


Figura 2.1: Schema concettuale

2.2 Schema logico

2.2.1 Nomenclatura

I nomi degli oggetti della base dati (tabelle, viste e campi) sono stati scelti seguendo una nomenclatura utilizzata dall'autore nella propria attività lavorativa.

Si tratta di regole assolutamente opinabili, non prive di ambiguità e senza nessuna pretesa di sistematicità o oggettività ma che, durante il lavoro di sviluppo di un sistema, si rivelano di una certa utilità.

Le regole sono le seguenti:

per i nomi della tabelle useremo il prefisso:

Z_ per le tabelle di “sistema”, quelle, cioè, non gestibili direttamente dagli utenti.

A_ per le tabelle “anagrafiche”, intese come tabelle contenenti dati di persone.

T_ per le tabelle cosiddette “di decodifica”: tipicamente elenchi composti da un codice e una descrizione.

M_ per le tabelle di “movimentazione” : tabelle popolate da processi periodici o comunque in qualche modo legati al tempo.

V_ per le viste.

Per i nomi dei campi useremo il prefisso:

DD_ per i campi di tipo stringa (o carattere).

NR_ per i campi numerici.

DT_ per i campi data.

TS_ per i campi timestamp.

CD_ per i campi contenenti un codice non tabellato (es. il codice fiscale di una persona).

C_[nome tabella] per i campi cosiddetti di *lookup*, quelli, cioè, collegati tramite foreign key con un'altra tabella; questa è forse la regola più utile: mentre si scrive il codice di un programma, quale che sia il linguaggio (Java, SQL o quel che si vuole), si può avere immediatamente l'idea del significato di un campo oppure, in senso inverso, si può ricostruire quale sarà il nome di un campo senza ricorrere alla documentazione o ai metadati. Ad esempio, nella tabella dei pazienti A_PAZIENTE, il campo che contiene il titolo di studio è collegato alla tabella dei titoli di studio, T_TITOLO, e si chiama quindi CT_TITOLO.

2.2.2 Principi generali di progettazione

- Tutte le tabelle hanno una chiave surrogata (la colonna ID) il cui valore è, nella maggior parte dei casi, generato in maniera automatica (auto_increment in MySQL). In alcuni casi l'ID è valorizzato manualmente perchè si tratta di un valore *sensibile*: è il caso, ad esempio, della tabella Z_NORMALIZZA (vedi tabella 2.4) dove l'ID rappresenta il valore ottenuto sommando le risposte al questionario.
- si utilizzano, ogni volta che è possibile, stored procedure al posto di funzioni scritte nel linguaggio client (C# in questo progetto).
- analogamente al punto precedente, tutte le volte che, leggendo i dati da una tabella, è necessaria una qualunque elaborazione (ad esempio la formattazione di un valore), si cercherà di realizzare questa elaborazione tramite una vista.

Gli ultimi due punti garantiscono una maggior modularità e flessibilità dell'applicazione: ad esempio è possibile, in molti casi (quando la modifica non riguarda l'interfaccia della stored procedure), modificare, correggere o

estendere un comportamento senza dover ricompilare e pubblicare l'intera applicazione. Purtroppo, da questo punto di vista, MySQL è un po' carente in quanto non supporta le stored procedure che ritornano un cursore.

Un altro vantaggio non trascurabile riguarda le prestazioni e la sicurezza : il codice lato database è spesso ottimizzato ma soprattutto risiede accanto ai dati (salvo casi particolari di database distribuiti): quindi non si genera traffico di rete che possa deteriorare le prestazioni o la sicurezza dei dati.

2.2.3 Tabelle

Utenti profili

La generalizzazione tra le entità *medico*, *infermiere* e *utente* è stata resa introducendo il concetto di *profilo*.

Esiste una tabella degli utenti (tabella Z_UTENTE), e ad ogni utente è associato un profilo (tabella Z_PROFILLO). Il profili definiti sono tre: oltre a *medico* e *infermiere*, previsti nell'analisi dei requisiti, si aggiunge quello di *amministratore*, a cui sono associate le funzioni di amministrazione dell'applicazione (in particolare la gestione degli utenti e la visualizzazione dei log di sistema).

Funzioni e menù

La tabella delle funzioni (Z_FUNZIONE) elenca tutte le voci di menù e le associa al profilo minimo necessario per accedervi. Si noti che sono elencate in questa tabella tutte le voci di menù, anche quelle non associate a maschere ma che hanno la semplice funzione di raggruppare voci comuni. Ad esempio, sotto la voce di menù "tabelle" sono raggruppate le maschere relative all'archivio pazienti, e non c'è nessuna maschera associata.

Per rappresentare questo collegamento tra i link alle maschere e la voce di menù che le raggruppa, si utilizza una struttura ad albero realizzata tramite una relazione ricorsiva della tabella con se stessa.

Tale relazione è realizzata con la coppia di campi ID e ID_PARENT, con

la convenzione che, per le voci “radice” ID_PARENT vale -1 mentre, per le altre, contiene il valore del campo ID per la voce padre.

Ad esempio la voce “Questionari”, che raggruppa le maschere relative ai questionari, ha ID uguale a 8 e ID_PARENT a -1. La voce “Somministra”, che fa parte di questo sotto-menù, ha ID_PARENT uguale a 8.

I campi delle tre tabelle Z_UTENTE, Z_PROFILO e Z_FUNZIONE sono illustrati in dettaglio nella tabella 2.1.

Tabella Z_PROFILO	
ID	chiave primaria
DD_DESCRIZIONE	descrizione (es. medico, infermiere)
Tabella Z_UTENTE	
ID	chiave primaria
DD_LOGIN	username per l'accesso al sistema
DD_PASSWORD	password di accesso
DD_NOME	nome dell'utente
DD_COGNOME	cognome
DD_MAIL	indirizzo mail
CZ_PROFILO	puntatore al profilo assegnato
Tabella Z_FUNZIONE	
ID	chiave primaria
DD_DESCRIZIONE	descrizione estesa della funzione
DD_DESCR_BREVE	descrizione breve
DD_URL	percorso della pagina web
CZ_PROFILO	puntatore al profilo minimo necessario per "vedere" la funzione
ID_PARENT	puntatore ricorsivo alla funzione "padre"

Tabella 2.1: Tabelle utente e profilo

Pazienti e tabelle collegate

La tabella dei pazienti (A_PAZIENTE) contiene i dati dei pazienti e ha, come tabelle collegate, quella dei titoli di studio (T_TITOLO) e dei codici di stato civile (T_STATO_CIVILE) (tabella 2.2).

È stato inserito anche un campo (CZ_UTENTE) che indica l'utente che ha inserito il record. Lo scopo di questo campo è quello di permettere, se sarà richiesto in futuro, di realizzare il filtro dei pazienti per utente descritto nei requisiti.

Questionari

Per quanto riguarda i questionari, possiamo distinguere una parte *statica*, nel senso che definisce la *struttura* del questionario, e una *dinamica* o di movimento che raccoglie le risposte ottenute.

La struttura dei questionari è realizzata mediante tre tabelle:

- la tabella A_QUEST contiene due descrizioni (una estesa e una breve) del questionario.
- A_QUEST_SEZIONE, dettaglio di A_QUEST, contiene le sezioni in cui è suddiviso il questionario (tre, come abbiamo visto, per il PROCLARA). Qui è anche memorizzato il numero di risposte possibili, che può essere diverso da sezione a sezione.
- infine la tabella A_QUEST_ITEM contiene il testo delle domande vere e proprie, il nome del file contenente l'immagine associata, e un campo contenente le risposte possibili separate dal carattere “|” (tabella 2.3).

Le risposte ai questionari sono memorizzate nelle tabelle M_QUEST (testata) e M_QUEST_ITEM (dettaglio).

In M_QUEST vengono salvati:

- il codice del paziente a cui si sta somministrando il questionario (CA_PAZIENTE).

Tabella T_TITOLO	
ID	chiave primaria
DD_DESCRIZIONE	descrizione del titolo di studio
Tabella T_STATO_CIVILE	
ID	chiave primaria
DD_DESCRIZIONE	descrizione (es. celibe, nubile)
Tabella A_PAZIENTE	
ID	chiave primaria
DD_COGNOME	cognome del paziente
DD_NOME	nome
CT_TITOLO	titolo di studio
CD_SESSO	sesso
DT_NASCITA	data di nascita
CT_STATO_CIVILE	stato civile
DD_MEDICO_NOME	nome del medico di famiglia
DD_MEDICO_COGNOME	cognome del medico di famiglia
DD_MEDICO_EMAIL	indirizzo mail del medico
DD_MEDICO_SALUTO	inizio della lettera al medico curante (es. "Cara dott.ssa")
DD_FOTO	nome del file contenente la foto del paziente. Opzionale
BL_NOTA	note libere legate al paziente
TS_NOW	marca temporale che indica data e ora dell'inserimento o modifica del record
CZ_UTENTE	puntatore all'utente che ha inserito il paziente

Tabella 2.2: Tabelle stato civile, titolo di studio e paziente

- i risultati delle sezioni del questionario e il risultato complessivo.
- la lettera per il medico curante (nel campo BL_LETTERA).

Tabella A_QUESTION	
ID	chiave primaria
DD_DESCRIZIONE	descrizione estesa
DD_DESCR_BREVE	descrizione breve
Tabella A_QUESTION_SEZIONE	
ID	chiave primaria
DD_DESCRIZIONE1	prima riga della descrizione
DD_DESCRIZIONE2	seconda riga della descrizione
CA_QUESTION	questionario a cui questa sezione appartiene
NR_NUM_SCELTE_POSSIBILI	numero di possibili risposte per questa sezione
Tabella A_QUESTION_ITEM	
ID	chiave primaria
CA_QUESTION	questionario a cui questa domanda appartiene
CA_QUESTION_SEZIONE	sezione a cui questa domanda appartiene
DD_DOMANDA	testo della domanda
DD_IMMAGINE	nome del file contenente l'immagine
DD_RISPOSTE	testo delle possibili risposte
NR_ORDER	posizione della domanda nel questionario

Tabella 2.3: Tabelle struttura questionario

La stored procedure `calcolaRisultato` calcola i risultati (delle sezioni e complessivo) del questionario, leggendo i coefficienti dalla tabella `Z_NORMALIZZA`. I risultati sono salvati sia in forma numerica che come stringa già formattata per essere mostrata dall'applicazione in vari contesti (intestazione di colonne, etichette descrittive ecc.).

Alla fine della somministrazione di un questionario viene anche costruito il testo di una lettera per il medico curante del paziente.

La lettera è composta di un'intestazione personalizzata per ogni utente/medico (letta dalla tabella `Z_UTENTE_TESTATA_LETTERA`) e di un testo contenente i dati anagrafici del paziente esaminato e i risultati del questionario; Il

calcolo della lettera viene effettuato dalla stored procedure `calcolaLettera`.
 La lettera precalcolata può essere modificata e salvata dall'utente/medico.
 La tabella 2.4 mostra il dettaglio di queste tabelle.

Tabella M_QUEST	
ID	chiave primaria
CA_QUEST	puntatore alla testata della struttura del questionario
CA_PAZIENTE	paziente a cui viene somministrato
DT_NOW	marca temporale della somministrazione
DD_RISULTATO	risultati in forma di stringa
NR_RISULTATO1	risultato prima sezione
NR_RISULTATO2	risultato seconda sezione
NR_RISULTATO3	risultato terza sezione
DD_RISULTATO	risultati in forma di stringa
CZ_UTENTE	utente che ha somministrato il questionario
BL_LETTERA	lettera
Tabella M_QUEST_ITEM	
ID	chiave primaria
CM_QUEST	puntatore al record di M_QUEST
CA_QUEST_ITEM	puntatore alla struttura della domanda
NR_SCELTA	risposta scelta dal paziente
Tabella Z_NORMALIZZA	
ID	chiave primaria e numero cella
NR_COEFF	coefficiente da utilizzare nel calcolo
Tabella Z_UTENTE_TESTATA_LETTERA	
ID	chiave primaria e numero cella
BL_TESTO	intestazione
CZ_UTENTE	utente proprietario della intestazione

Tabella 2.4: Tabelle risultati questionario

2.2.4 Viste

Il sistema utilizza alcune viste create ad hoc per estrarre i dati necessari alle varie pagine. Di seguito ne riportiamo il dettaglio.

La vista V_M_QUEST viene utilizzata nella pagina che mostra l'archivio dei risultati; è una join tra le tabelle M_QUEST, A_QUEST, A_PAZIENTE e Z_UTENTE. I campi che la compongono sono mostrati in dettaglio nella tabella 2.5.

Vista V_M_QUEST	
IdMQuest	id del record di M_QUEST
IdPaziente	id del record di A_PAZIENTE
IdAQuest	id del record di A_QUEST
Questionario	descrizione del questionario (da A_QUEST)
Data	data di somministrazione
Cognome	cognome del paziente
Nome	nome del paziente
Nr_Risultato1	risultato prima sezione
Nr_Risultato2	risultato seconda sezione
Nr_Risultato3	risultato terza sezione
Risultato	risultati in forma di stringa
Dd_utente	cognome e nome dell'utente che ha somministrato il questionario
Dd_utente_nome	nome dell'utente
Dd_utente_cognome	cognome dell'utente
Dd_medico_famiglia	cognome e nome del medico di famiglia
Dd_medico_email	indirizzo mail del medico
Dd_medico_nome	nome del medico
Dd_medico_cognome	cognome del medico

Tabella 2.5: Vista archivio

La vista V_A_QUEST viene utilizzata, nella pagina della somministra-

zione di un questionario, dal controllo che mostra l'elenco dei questionari definiti nel sistema. Ha, come fonte, un'unica tabella, A_QUEST, e l'unica operazione che fa è quella di concatenare la descrizione breve e quella estesa del questionario (tabella 2.6).

Vista V_A_QUEST	
Id	id del record di A_QUEST
Dd_descrizione	è uguale al corrispondente campo di A_QUEST
Dd_descr_breve	come sopra
Dd_descr_completa	concatena dd_descr_breve e dd_descrizione

Tabella 2.6: Vista testata questionari

Durante la somministrazione, per avere insieme i dettagli della domanda e della sezione d'appartenenza, viene utilizzata la vista V_A_QUEST_ITEM (tabella 2.7).

Vista V_A_QUEST_ITEM	
Id	id del record di A_QUEST_ITEM
Ca_quest	puntatore al record di testata del questionario (A_QUEST)
Ca_quest_sezione	puntatore alla sezione (A_QUEST_SEZIONE)
Dd_domanda	testo della domanda
Dd_immagine	immagine associata
Dd_risposte	risposte possibili
Nr_order	numero d'ordine della domanda
Dd_descrizione1	prima riga descrizione sezione
Dd_descrizione2	seconda riga descrizione sezione

Tabella 2.7: Vista dettaglio questionari

La vista V_A_PAZIENTE unisce i dati dell'anagrafica pazienti con le tabelle collegate del titolo di studio e dello stato civile (tabella 2.8).

Vista V_A_PAZIENTE	
Id	id del record di A_PAZIENTE
Dd_cognome	cognome del paziente
Dd_nome	nome
Ct_titolo	collegamento con la tabella dei titoli di studio
Cd_sesso	sesso
Dt_nascita	data di nascita
Ct_stato_civile	collegamento con la tabella dello stato civile
Dd_stato_civile	descrizione dello stato civile (es. "coniugato/a")
Ct_titolo	concatena dd_descr_breve e dd_descrizione
Dd_titolo	descrizione del titolo di studio
Cd_fiscale	codice fiscale
Dd_medico_cognome	cognome del medico di famiglia
Dd_medico_nome	nome del medico
Dd_medico_email	indirizzo mail del medico
Dd_foto	nome del file contenente la foto del paziente
Bl_nota	note associate

Tabella 2.8: Vista pazienti

La figura 2.2 mostra lo schema logico riassuntivo delle tabelle principali del sistema.

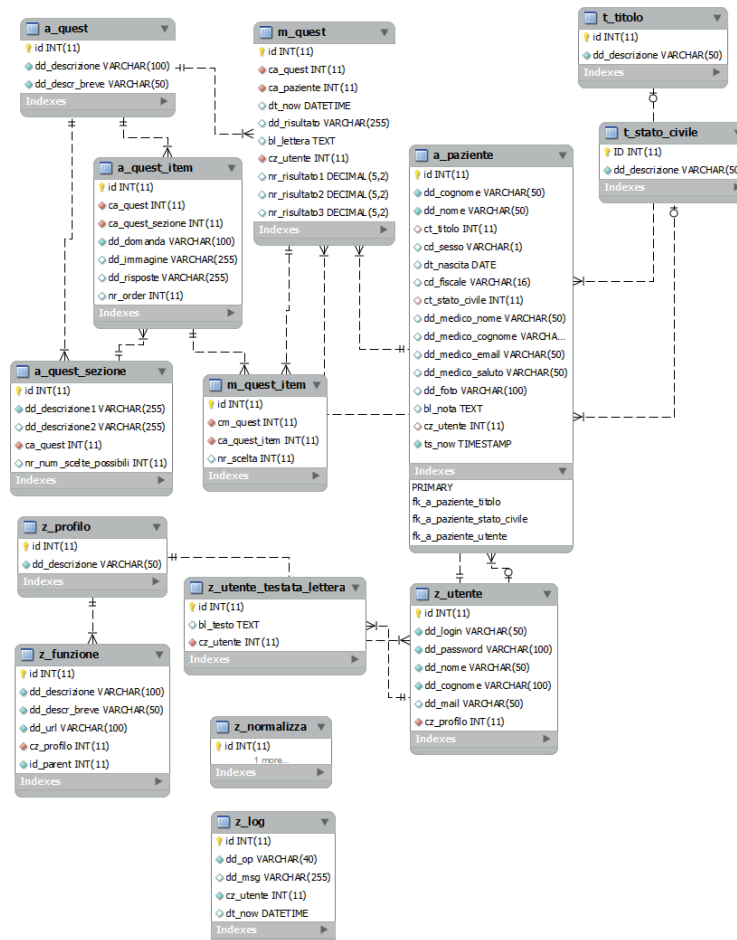


Figura 2.2: Schema logico

Capitolo 3

Il sito Emma

3.1 Scelte tecnologiche

La scelta delle tecnologie è stata guidata in parte dalle preferenze degli utenti (per quanto riguarda il database MySQL) e in parte dalle conoscenze dell'autore (il framework ASP.NET).

Gli strumenti utilizzati sono stati i seguenti:

- ambiente di sviluppo: Microsoft Visual Studio 2008 Express Edition (<http://www.microsoft.com/express/downloads>); gratuito.
- framework: ASP.NET versione 3.5 (www.asp.net); gratuito.
- linguaggio: C#.
- database: MySQL versione 5.1 (<http://www.mysql.com>); open source con licenza GPL.
- controlli dell'interfaccia utente: librerie DevExpress (www.devexpress.com): ASPxExperience e ASPxGridView suite; licenza commerciale: installata versione dimostrativa.
- per la generazione della lettera in formato pdf la libreria iTextSharp (<http://itextpdf.com>); open source con licenza GPL.

- per i grafici la libreria Microsoft Chart Control (<http://code.msdn.microsoft.com/mschart>); gratuito.

Il sito *Emma* risiede attualmente nel dominio www.kotik.it.

Il progetto è open source e i sorgenti sono sotto version control su Assembla all'indirizzo:

<https://www.assembla.com/spaces/Kotik>

3.2 Nomenclatura

- Le unit relative alle pagine del sito hanno tutte il prefisso `Fm`. La pagina dei pazienti, ad esempio, si chiama `FmPazienti`. Fanno eccezione a questa regola solo la pagina “home” (`Default.aspx`) e la pagina “about” (`About.aspx`).
- Le classi interne (quelle non associate, cioè, ad una pagina visibile) sono salvate in file con lo stesso nome della classe principale (es. `DALRuntime`).

3.3 Principi generali di progettazione

Durante lo sviluppo dell'applicazione si è cercato di rispettare questi principi:

- la classe `DALRuntime` (dove `DAL` sta per *Data Access Layer*) è il punto del programma dove è concentrata tutta la logica che si interfaccia con la base dati. Tutte le altre classi, quando devono accedere alla base dati, utilizzano i servizi messi qui a disposizione.
- consistenza dell'interfaccia utente: la stessa funzionalità, se presente in pagine diverse, deve essere rappresentata all'utente e controllata usando sempre gli stessi meccanismi e la stessa grafica; non deve mai accadere, in pratica, che, se in una pagina una certa operazione si effettua in un

determinato modo, in un'altra pagina i comandi non siano gli stessi, o non siano disposti nello stesso modo.

3.4 Accessibilità e compatibilità

I componenti DevExpress mettono a disposizione una serie di proprietà come:

- Testo alternativo.
- Opzioni per impostare contrasto e colore.
- Tasti di accesso veloce.

allo scopo di migliorare l'accessibilità delle applicazioni. Al momento, tuttavia, non è stata svolta nessuna attività a questo proposito.

Per quanto riguarda la compatibilità la libreria supporta i seguenti browser:

- Internet Explorer dalla versione 6.0 in poi.
- Google Chrome dalla 1.0.
- Netscape Navigator dalla 9.0.
- Opera dalla 9.0.
- Mozilla Firefox dalla 2.0.
- Apple Safari dalla 3.0.

I test sono stati condotti solamente con Firefox 3.6 e Safari 5.0.

3.5 Sicurezza

Per autenticare gli utenti l'applicazione usa una logica interna basata sulle tabelle Z_UTENTE e Z_PROFILO.

Per garantire che solo gli utenti autenticati possano accedere alle pagine si sfrutta il meccanismo, messo a disposizione dal framework ASP.NET, della cosiddetta *form based authentication*.

In pratica le pagine che si vuole mantenere riservate sono salvate in cartelle separate; in ogni cartella poi risiede un file di configurazione (`web.config`) in cui vengono indicati gli utenti e/o i ruoli che possono accedere alla cartella.

In questo modo, se un utente non è autenticato, non potrà accedere alle pagine, neanche fornendo al proprio browser il link completo della pagina.

Se l'applicazione dovesse avere un qualche concreto sviluppo futuro, la gestione delle password dovrebbe essere notevolmente rafforzata.

Solo per citare i punti più importanti bisognerà:

- cifrare la password che, al momento, è memorizzata in chiaro nella tabella `Z_UTENTE`.
- mettere a disposizione un meccanismo per consentire agli utenti di resettare la password in caso di dimenticanza, con tutta la gestione conseguente (invio via mail di un link di validazione ecc.).
- stabilire una qualche politica di rinnovo password.

3.6 Struttura interna

Alcune classi interne sono di particolare importanza per l'architettura dell'applicazione.

Sono le seguenti:

- `DALRuntime` è la classe, come abbiamo visto in precedenza, che rappresenta l'interfaccia con la base dati. Qui si trova, ad esempio, il metodo che ritorna la connessione alla base dati, e tutti i metodi di lettura e scrittura delle varie tabelle.

L'applicazione *Emma* sfrutta poi la possibilità, offerta dal framework ASP.NET, di personalizzare la logica con cui il sistema autentica gli uten-

ti: è il cosiddetto *Provider Model* di ASP.NET (si vedano alcuni cenni in Appendice B).

- **EmmaMembershipProvider**: contiene il *Membership Provider* dell'applicazione. Questa classe contiene i metodi per l'autenticazione degli utenti. In questo caso l'autenticazione è basata sulla tabella Z_UTENTE (tabella 2.1).
- **EmmaRoleProvider**: questo è il *Role Provider*, la classe, cioè, che stabilisce quali ruoli appartengono all'utente collegato. Dove il concetto di *ruolo* qui equivale a quello di *profilo* (tabella 2.1).
- **EmmaSitemapProvider**: questa classe costituisce il *Sitemap Provider* di *Emma*. Responsabilità di questa classe è costruire il menù dell'applicazione, leggendo la tabella Z_FUNZIONE (sempre in tabella 2.1).

3.7 Struttura visibile

La struttura generale del sito è mostrata nella figura 3.1.

La pagina principale (home) contiene un testo di spiegazione generale dello scopo del sito (figura 3.2). La corrispondente pagina nei sorgenti è la classica `Default.aspx`.

In alto a destra si trova il link “accesso” che porta alla pagina di login (figura 3.3).

Se non si effettua il login le uniche funzioni disponibili sono la pagina principale e una di “about” (`About.aspx`) che contiene informazioni tecniche sul sito.

Una volta effettuato l'accesso nel menù appaiono le funzioni configurate per il profilo assegnato all'utente.



Figura 3.1: Mappa del sito

3.7.1 Il menù Tabelle

Le pagine del menù tabelle si trovano tutte nella cartella “Tabelle” e sono le seguenti:

- la tabella dei titoli di studio (`FmTitoliDiStudio.aspx`).
- la tabella dello stato civile (`FmStatoCivile.aspx`).

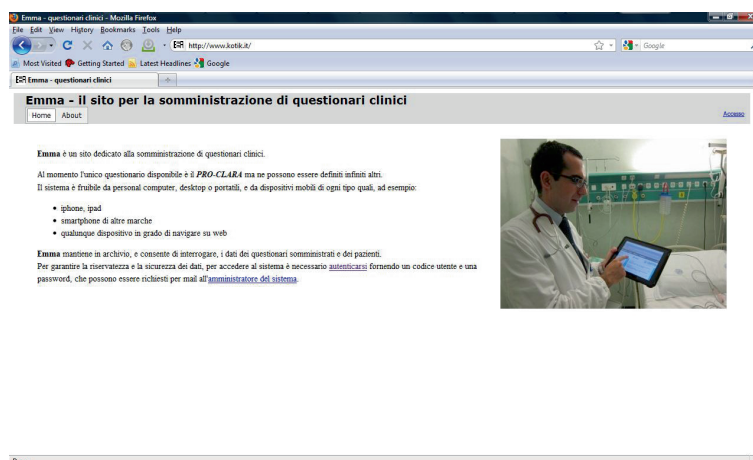


Figura 3.2: Pagina home

- l'archivio dei pazienti (FmPazienti.aspx).

Tutte queste pagine presentano i dati in una griglia che si può ordinare cliccando semplicemente sulla colonna desiderata.

Per modificare, inserire o cancellare è necessario cliccare sul link corrispondente nella prima colonna della griglia.

La figura 3.4 mostra come si presenta la maschera dei pazienti dopo aver premuto il link di modifica di un record.

3.7.2 Il menù Questionari

Nella voce di menù "Questionari" si trovano tutte le funzioni relative ai questionari.

La maschera "Struttura" consente di inserire nuovi questionari o modificare quelli già presenti.

Viene visualizzata una griglia gerarchica che consente di esplodere i vari livelli premendo su un link con l'immagine di un segno "più" (+). In questo modo si passa dalla testata del questionario (dove troviamo, sostanzialmente, il titolo) alle varie sezioni e, infine, alle domande vere e proprie (figura 3.5).

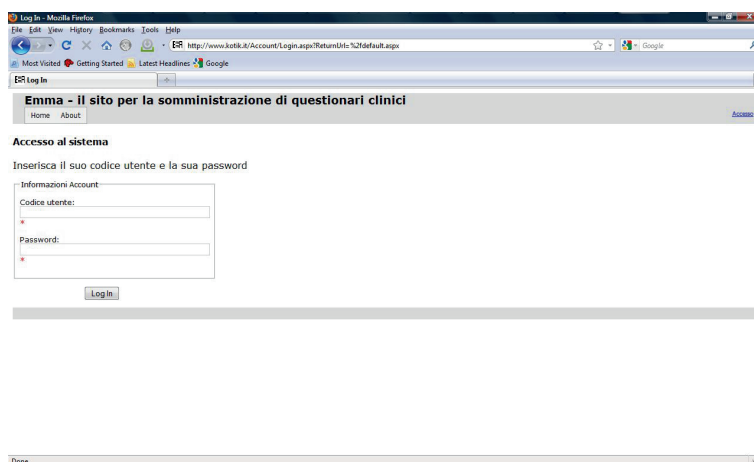


Figura 3.3: Pagina login

Per somministrare un questionario è necessario selezionare la voce “Somministra”. Al primo ingresso è necessario selezionare il questionario che si desidera somministrare (nel caso ne siano definiti più di uno) e il paziente. Fatta questa operazione, premendo il tasto “Avanti” inizia la somministrazione (figura 3.6).

La figura 3.7 mostra come si presenta una pagina con una domanda del questionario.

Nella parte alta della pagina sono riportati il titolo del questionario, il nome del paziente, il titolo della sezione.

In posizione centrale il testo della domanda, il numero progressivo della domanda all’interno del questionario e l’immagine associata (se caricata).

Al termine della compilazione viene calcolato il risultato (delle varie sezioni e complessivo) e costruito uno scheletro di lettera per il medico di famiglia del paziente.

Dopo l’ultima domanda viene mostrata una pagina conclusiva (figura 3.8) in cui viene mostrato il risultato calcolato del questionario e i due pulsanti:

- “Nuovo questionario” che ritorna alla pagina iniziale di scelta del questionario e del paziente.

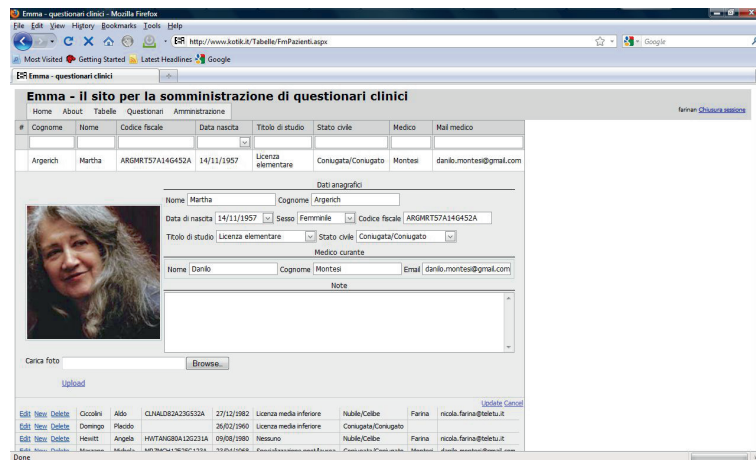


Figura 3.4: Pagina pazienti

- “Lettera” che porta alla pagina di gestione della lettera.

La pagina della lettera (figura 3.9) mostra la lettera in una *text area* che consente all’utente di apportare le modifiche necessarie al testo.

Appena sotto al testo ci sono quattro pulsanti con le seguenti funzioni:

- “Genera PDF” : stampa una copia della lettera in PDF e la invia al browser dell’utente.
- “Invia mail”: invia la lettera per email al medico curante del paziente (attivo solo se, in anagrafica pazienti, è presente l’indirizzo mail del medico).
- “Salva”: salva in archivio la lettera modificata.
- “Intestazione”: va alla pagina dove personalizzare l’intestazione della lettera. Quando la lettera viene inviata per email, oppure stampata in PDF, viene aggiunta un’intestazione, relativa all’utente che ha somministrato il questionario, contenente i riferimenti dell’utente. In questa pagina (figura 3.10) è possibile personalizzarla e salvarla (nella tabella Z_UTENTE_TESTATA_LETTERA tabella 2.4).

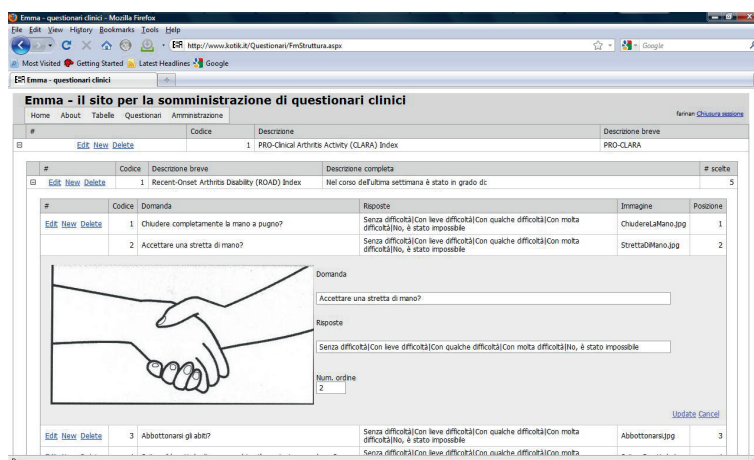


Figura 3.5: Pagina struttura questionario

I dati relativi ai questionari somministrati (le risposte, il risultato calcolato, la lettera per il medico curante) sono accessibili dalla voce “Archivio” (figura 3.11).

I risultati sono mostrati in una griglia che consente, ovviamente, la sola consultazione dei risultati, senza possibilità di modifica.

I risultati della griglia possono essere ordinati semplicemente cliccando sulle colonne.

Un'altra caratteristica interessante di questa griglia è la possibilità di raggruppare i dati della griglia trascinando semplicemente con il mouse l'intestazione di una colonna nell'apposito spazio nella parte superiore della griglia. Nella figura 3.12 si vede la griglia raggruppata per le colonne “cognome” e “data”.

Il raggruppamento si può annullare semplicemente trascinando di nuovo le colonne raggruppate dal pannello superiore di nuovo nella griglia.

Cliccando sul link “grafico” nella colonna con lo stesso nome viene prodotto un grafico dei risultati dei test per il paziente di quella riga. In ordinata sono riportati i tre risultati delle tre sezioni del test per quel paziente nell'ultimo anno (figura 3.13).

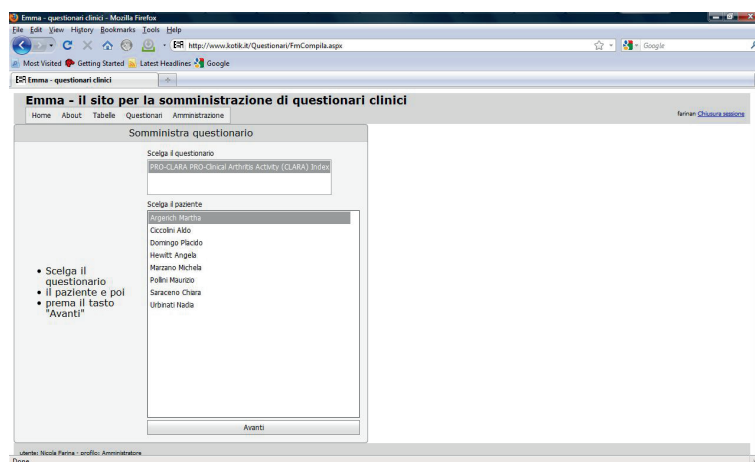


Figura 3.6: Pagina scegli questionario e paziente

Il grafico può anche essere visto in tre dimensioni cliccando sull’opzione “Abilita 3D” in basso a sinistra.

In caso di necessità di grafici (o elaborazioni) di altro tipo è possibile estrarre i dati della griglia per elaborarli con un software a propria scelta. Per compiere questa operazione è sufficiente cliccare con il tasto destro del mouse sulla griglia: apparirà un menù con un’unica voce: “Esporta in Excel” (figura 3.15).

Premendola viene generato, e inviato al browser, un file in formato Excel (figura 3.16).

3.7.3 Il menù Amministrazione

Questo menù è l’unico, per ora, a cui viene applicato il filtro in base al profilo richiesto nei requisiti. È infatti riservato al solo profilo “Amministratore”. Qui si trovano le pagine per:

- la gestione dell’archivio degli utenti dell’applicazione (`FmUtenti.aspx`).
- la visualizzazione del log di sistema (`FmLog.aspx`).

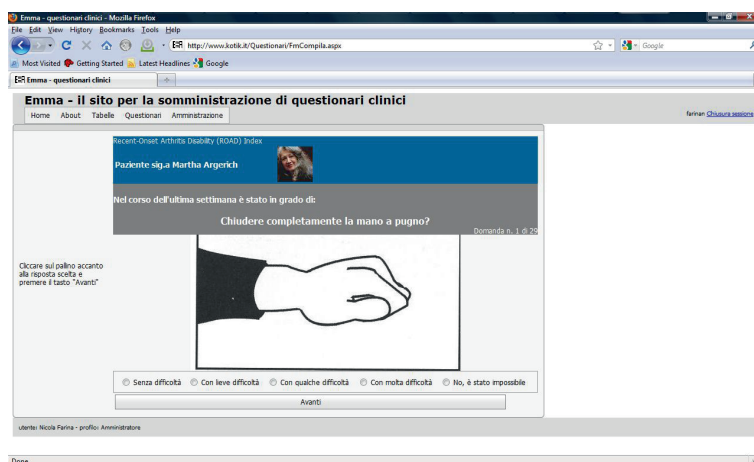


Figura 3.7: Pagina domanda questionario

La cartella è “Amministrazione”.

La maschera di gestione utenti mostra in una griglia gli utenti definiti nel sistema e consente di modificare, aggiungere, cancellare (figura 3.17).

La pagina “Log” è di sola consultazione e visualizza il log del sistema (tabella Z.LOG). Questa tabella è pensata per consentire all’amministratore di tenere sotto controllo il sistema e avere uno strumento per, da un lato, indagare eventuali problemi: dovrebbero essere scritti qui, infatti, i dettagli dei messaggi d’errore dovuti a eventuali eccezioni. Dall’altro dovrebbe funzionare anche come “audit” del sistema (per risalire a chi ha fatto cosa). Per ora contiene solo un messaggio che indica che un utente si è connesso al sistema (figura 3.18).

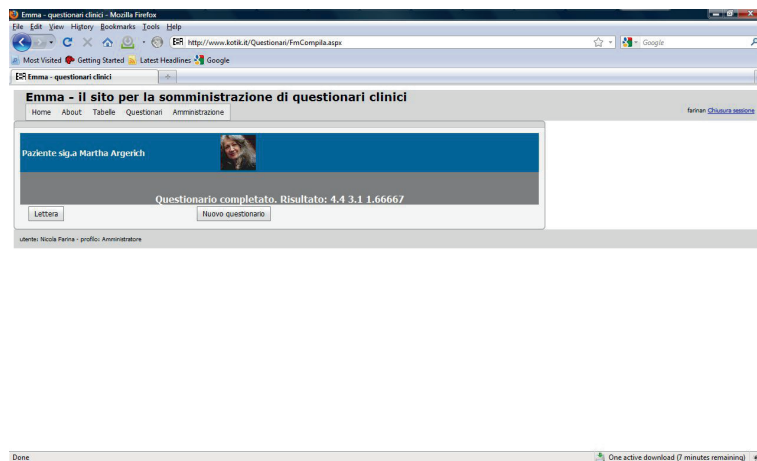


Figura 3.8: Pagina fine questionario

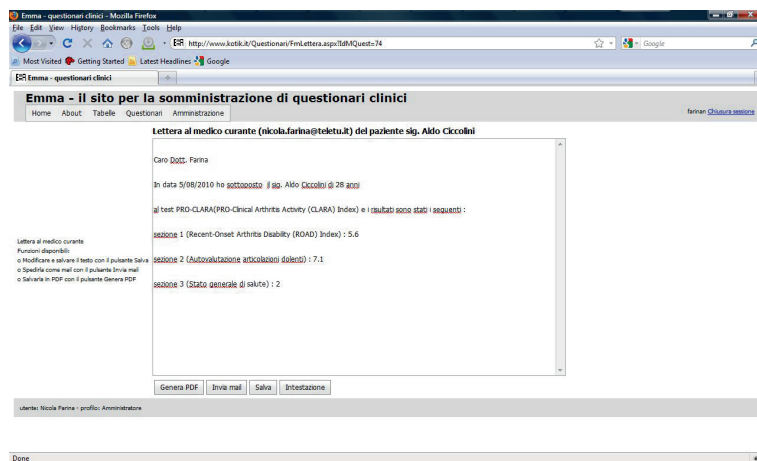


Figura 3.9: Pagina lettera

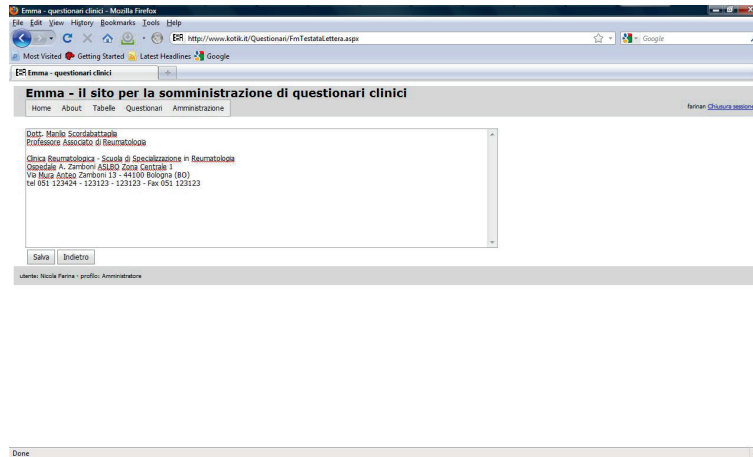


Figura 3.10: Pagina intestazione lettera

Cognome	Nome	Data	Risultato 1	Risultato 2	Risultato 3	Medico di famiglia	Somministrato da	Lettera	Grafico
Ciccolini	Alko	24/03/2010	4,60	5,60	2,90	Farina	Farina	Lettera	Grafico
Ciccolini	Alko	13/09/2010	5,40	4,80	0,90	Farina	Salaffi	Lettera	Grafico
Ciccolini	Alko	01/07/2010	6,90	5,40	1,30	Farina	Montesi	Lettera	Grafico
Ciccolini	Alko	31/03/2010	6,30	3,80	0,67	Farina	Farina	Lettera	Grafico
Ciccolini	Alko	15/01/2010	6,70	5,40	2,67	Farina	Salaffi	Lettera	Grafico
Ciccolini	Alko	11/12/2010	4,80	3,30	3,33	Farina	Montesi	Lettera	Grafico
Ciccolini	Alko	08/01/2010	3,30	3,80	2,67	Farina	Salaffi	Lettera	Grafico
Ciccolini	Alko	24/12/2010	4,40	4,80	0,90	Farina	Salaffi	Lettera	Grafico
Ciccolini	Alko	13/03/2010	5,40	5,00	2,67	Farina	Farina	Lettera	Grafico
Ciccolini	Alko	29/01/2010	5,20	3,80	1,33	Farina	Montesi	Lettera	Grafico
Ciccolini	Alko	07/11/2010	5,20	3,80	3,33	Farina	Montesi	Lettera	Grafico
Ciccolini	Alko	21/02/2010	5,60	5,20	0,33	Farina	Farina	Lettera	Grafico
Ciccolini	Alko	04/12/2010	6,70	4,00	0,67	Farina	Montesi	Lettera	Grafico
Ciccolini	Alko	26/12/2010	5,80	6,00	1,67	Farina	Farina	Lettera	Grafico
Ciccolini	Alko	19/01/2010	5,80	6,00	1,67	Farina	Farina	Lettera	Grafico
Ciccolini	Alko	06/01/2010	4,60	6,30	2,33	Farina	Farina	Lettera	Grafico
Ciccolini	Alko	26/08/2010	5,60	5,40	1,33	Farina	Farina	Lettera	Grafico
Ciccolini	Alko	27/10/2010	5,20	6,70	3,33	Farina	Salaffi	Lettera	Grafico
Ciccolini	Alko	24/01/2010	6,70	4,60	2,33	Farina	Salaffi	Lettera	Grafico
Hewitt	Angela	19/01/2010	5,40	6,90	1,67	Farina	Montesi	Lettera	Grafico

Figura 3.11: Pagina archivio

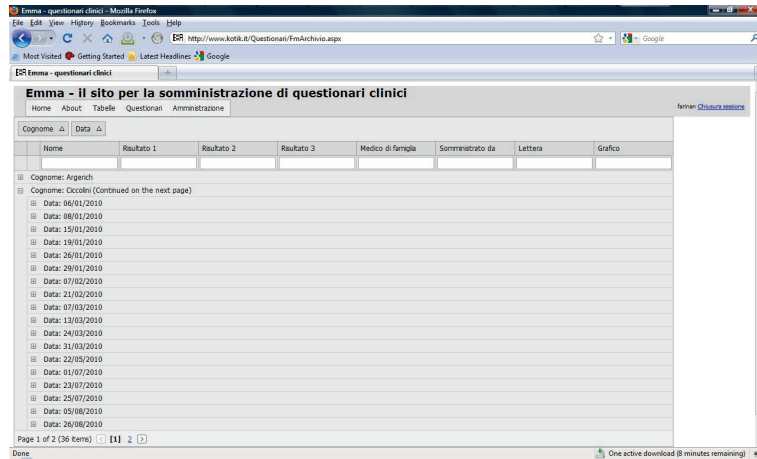


Figura 3.12: Pagina archivio raggruppato

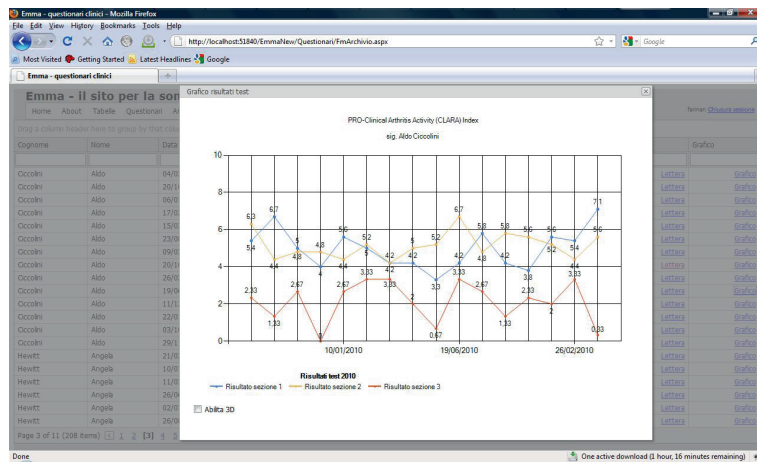


Figura 3.13: Pagina grafico 2D

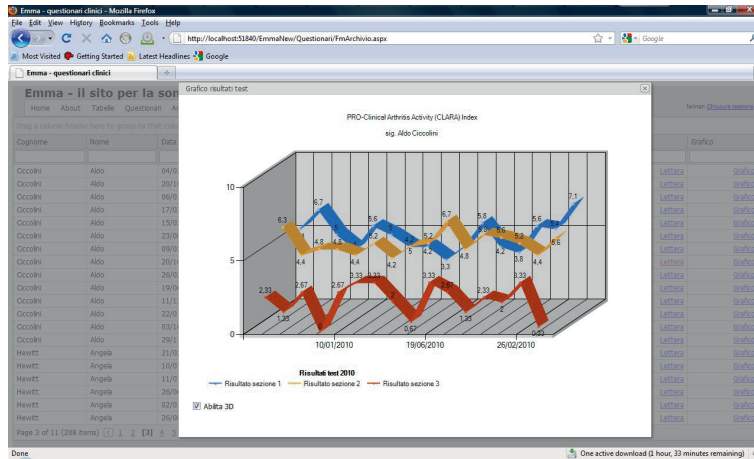


Figura 3.14: Pagina grafico 3D

The figure shows a data table titled "Emma - il sito per la somministrazione di questionari clinici". The table has columns for "Cognome", "Nome", "Data", "Risultato 1", "Risultato 2", "Risultato 3", "Medico di famiglia", "Somministrato da", "Lettera", and "Grafico". The data rows list patients like Argierich Martha with their respective test dates and results.

Cognome	Nome	Data	Risultato 1	Risultato 2	Risultato 3	Medico di famiglia	Somministrato da	Lettera	Grafico
Argierich	Martha	23/03/2010	5,20	5,00	2,90	Montesi	Salaffi	Lettera	Grafico
Argierich	Martha	16/09/2010	6,30	4,60	3,30	Montesi	Fanna	Lettera	Grafico
Argierich	Martha	24/11/2010	4,60	6,00	2,80	Montesi	Fanna	Lettera	Grafico
Argierich	Martha	21/02/2010	5,40	6,70	3,08	Montesi	Montesi	Lettera	Grafico
Argierich	Martha	24/12/2010	4,60	5,20	1,67	Montesi	Fanna	Lettera	Grafico
Argierich	Martha	13/01/2011				Montesi	Fanna	Lettera	Grafico
Argierich	Martha	15/12/2010				Montesi	Fanna	Lettera	Grafico
Argierich	Martha	08/09/2010	4,40	3,80	1,67	Montesi	Salaffi	Lettera	Grafico
Argierich	Martha	26/08/2010	3,30	5,80	3,08	Montesi	Montesi	Lettera	Grafico
Argierich	Martha	28/07/2010	4,80	7,30	0,33	Montesi	Fanna	Lettera	Grafico
Argierich	Martha	10/03/2010	5,40	4,60	2,67	Montesi	Salaffi	Lettera	Grafico
Argierich	Martha	02/03/2010	4,60	5,40	0,67	Montesi	Salaffi	Lettera	Grafico
Argierich	Martha	23/02/2010	5,00	5,40	1,00	Montesi	Fanna	Lettera	Grafico
Argierich	Martha	09/03/2010	4,00	5,40	2,33	Montesi	Montesi	Lettera	Grafico
Argierich	Martha	27/05/2010	5,80	5,60	0,67	Montesi	Fanna	Lettera	Grafico
Argierich	Martha	13/03/2010	5,80	4,60	1,33	Montesi	Montesi	Lettera	Grafico
Argierich	Martha	18/07/2010	6,50	3,50	2,33	Montesi	Fanna	Lettera	Grafico
Argierich	Martha	10/07/2010	3,60	3,80	1,33	Montesi	Fanna	Lettera	Grafico
Argierich	Martha	09/03/2010	6,50	4,00	0,33	Montesi	Montesi	Lettera	Grafico
Argierich	Martha	29/04/2010	4,20	5,00	0,00	Montesi	Montesi	Lettera	Grafico

Figura 3.15: Pagina archivio esporta

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Cognome	Nome	Data	Risultato 1	Risultato 2	Risultato 3	Medico di famiglia	Somministrato da	Lettera	Orificio						
2	Argersich	Martha	23/03/2010	5,2	5	2,5	Montesi	Salaffi	Lettera	Orificio						
3	Argersich	Martha	16/09/2010	6,3	4,6	3,3	Montesi	Farina	Lettera	Orificio						
4	Argersich	Martha	24/11/2010	4,6	6	2	Montesi	Farina	Lettera	Orificio						
5	Argersich	Martha	21/02/2010	5,4	6,7	7	Montesi	Montesi	Lettera	Orificio						
6	Argersich	Martha	24/12/2010	4,6	5,2	1,67	Montesi	Farina	Lettera	Orificio						
7	Argersich	Martha	04/09/2010	4,4	3,8	1,67	Montesi	Salaffi	Lettera	Orificio						
8	Argersich	Martha	26/08/2010	3,3	3,8	3	Montesi	Montesi	Lettera	Orificio						
9	Argersich	Martha	28/07/2010	4,8	7,3	0,33	Montesi	Farina	Lettera	Orificio						
10	Argersich	Martha	10/03/2010	5,4	4,6	2,67	Montesi	Salaffi	Lettera	Orificio						
11	Argersich	Martha	02/03/2010	4,6	5,4	0,67	Montesi	Salaffi	Lettera	Orificio						
12	Argersich	Martha	23/02/2010	5,4	5,4	0	Montesi	Farina	Lettera	Orificio						
13	Argersich	Martha	09/03/2010	4	5,4	2,33	Montesi	Montesi	Lettera	Orificio						
14	Argersich	Martha	27/05/2010	5,8	5,6	0,67	Montesi	Farina	Lettera	Orificio						
15	Argersich	Martha	18/03/2010	5,8	4,6	1,33	Montesi	Montesi	Lettera	Orificio						
16	Argersich	Martha	18/07/2010	6,5	3,5	2,33	Montesi	Farina	Lettera	Orificio						
17	Argersich	Martha	12/07/2010	5,6	3,8	1,33	Montesi	Farina	Lettera	Orificio						
18	Argersich	Martha	08/03/2010	6,5	4	0,33	Montesi	Montesi	Lettera	Orificio						
19	Argersich	Martha	29/04/2010	4,2	5	0	Montesi	Montesi	Lettera	Orificio						
20	Argersich	Martha	17/12/2010	5	5,2	1	Montesi	Farina	Lettera	Orificio						
21	Argersich	Martha	12/04/2010	4,4	5,6	2,67	Montesi	Montesi	Lettera	Orificio						
22	Argersich	Martha	02/10/2010	5,6	3,5	2,33	Montesi	Farina	Lettera	Orificio						
23	Argersich	Martha	04/09/2010	5	4,6	2,67	Montesi	Montesi	Lettera	Orificio						
24	Argersich	Martha	14/12/2010	4,6	6,9	1,33	Montesi	Farina	Lettera	Orificio						
25	Argersich	Martha	07/07/2010	5,8	3,5	0	Montesi	Montesi	Lettera	Orificio						
26	Argersich	Martha	06/07/2010	5,6	4,8	0,33	Montesi	Salaffi	Lettera	Orificio						
27	Argersich	Martha	22/05/2010	4,2	5	1	Montesi	Farina	Lettera	Orificio						

Figura 3.16: Pagina archivio Excel

The screenshot shows a web browser displaying the user management page for the 'Emma - questionari clinici' system. The page title is 'Emma - il sito per la somministrazione di questionari clinici'. The main content is a table listing users with columns for '#', 'Cognome', 'Nome', 'Codice utente', 'Indirizzo di posta', and 'Profilo'. The table contains the following data:

#	Cognome	Nome	Codice utente	Indirizzo di posta	Profilo
Edit New Delete	Farina	Nicola	farinan		Amministratore
Edit New Delete	Monosci	Dario	monosci		Medico
Edit New Delete	Farina	Stefano	farinas		Medico
Edit New Delete	Salaffi	Fausto	salaffi		Medico
Edit New Delete	Rossi	Mario	rossim		Operatore
Edit New Delete	Orsini	Umberto	orsini		Pubblico

At the bottom of the page, there is a status bar indicating 'One active download (86 minutes remaining)'.

Figura 3.17: Pagina gestione utenti

Emma - il sito per la somministrazione di questionari clinici

Home About Tabelle Questionari Amministrazione farisan Chiudi sessione

Drag a column header here to group by that column

Operazione	Messaggio	Utente	Data e ora
login	farisan	farisan	23/01/2011 16.09.59
login	farisan	farisan	19/01/2011 23.27.06
login	farisan	farisan	19/01/2011 22.52.26
login	farisan	farisan	19/01/2011 19.22.52
login	farisan	farisan	13/01/2011 19.09.25
login	farisan	farisan	13/01/2011 19.08.07
login	farisan	farisan	29/12/2010 22.30.12
login	farisan	farisan	27/12/2010 14.32.17
login	farisan	farisan	26/12/2010 10.58.05
login	farisan	farisan	24/12/2010 20.43.38

Page 1 of 3 (25 items) [1] 2 3

Utente: Nicola Farina - profilo: Amministratore

Done One active download (23 minutes remaining)

Figura 3.18: Pagina log di sistema

Capitolo 4

Listati dei programmi

4.1 Database

4.1.1 Creazione tabelle e viste

Listing 4.1: CreaTables.sql

```
drop schema Emma;
create schema Emma;
use Emma;

-----
-- Amministrazione e utenti
-----

create table z_profilo
(
  id int not null auto_increment ,
  dd_descrizione varchar(50) not null ,
  primary key(id)
);
create table z_utente
(
  id int not null auto_increment ,
```

```
    dd_login varchar(50) not null ,
    dd_password varchar(100) not null ,
    dd_nome varchar(50) not null ,
    dd_cognome varchar(100) not null ,
    dd_mail varchar(50) ,
    cz_profilo int not null ,
    primary key(id) ,
    foreign key(cz_profilo) references z_profilo(id)
);
create unique index IU_Z_UTENTE_0 on z_utente(dd_login)
;
create table z_funzione
(
    id int not null auto_increment ,
    dd_descrizione varchar(100) not null ,
    dd_descr_breve varchar(50) not null ,
    dd_url varchar(100) not null ,
    cz_profilo int not null ,
    id_parent int not null default -1 ,
    foreign key(cz_profilo) references z_profilo(id) ,
    primary key(id)
);
— Log
create table z_log
(
    id int not null auto_increment ,
    dd_op varchar(40) not null ,
    dd_msg varchar(255) ,
    cz_utente int not null ,
    dt_now datetime ,
    primary key(id)
```

```
);  
— join di z_log con z_utente  
create or replace view v_log as  
select zl.id, zl.dd_op, zl.dd_msg, zl.dt_now, zl.  
  cz_utente, zu.dd_login  
  from z_log zl, z_utente zu  
  where zl.cz_utente = zu.id;  
— join utenti e profilo corrispondente  
create or replace view v_utenti_profilo as  
select zu.id, zu.dd_login, zu.dd_mail, zu.dd_password,  
  zu.dd_nome, zu.dd_cognome, zu.cz_profilo, zp.  
  dd_descrizione dd_profilo  
  from z_utente zu, z_profilo zp  
  where zu.cz_profilo = zp.id;  
— Intestazione lettera personalizzata per utente  
create table z_utente_testata_lettera  
(  
  id int not null auto_increment,  
  bl_testo text,  
  cz_utente int not null,  
  primary key (id),  
  foreign key (cz_utente) references z_utente(id)  
);
```

— *Tabelle*

```
create table t_titolo  
(  
  id int not null auto_increment,  
  dd_descrizione varchar(50) not null,
```

```
    primary key(id)
);
create table t_stato_civile
(
  ID int not null auto_increment,
  dd_descrizione varchar(50) not null,
  primary key(id)
);
create table t_patologia
(
  ID int not null auto_increment,
  dd_descrizione varchar(50) not null,
  primary key(id)
);
```

— *tabella pazienti*

```
create table a_paziente
(
  id int not null auto_increment,
  dd_cognome varchar(50) not null,
  dd_nome varchar(50) not null,
  ct_titolo int,
  cd_sesso varchar(1),
  dt_nascita date,
  cd_fiscale varchar(16),
  ct_stato_civile int,
  dd_medico_nome varchar(50),
  dd_medico_cognome varchar(50),
  dd_medico_email varchar(50),
```



```
dd_medico_saluto varchar(50),
dd_foto varchar(100),
bl_nota text,
cz_utente int,
ts_now timestamp,
primary key(id)
);
alter table a_paziente add constraint
fk_a_paziente_titolo foreign key (ct_titolo)
references t_titolo(id);
alter table a_paziente add constraint
fk_a_paziente_stato_civile foreign key (
ct_stato_civile) references t_stato_civile(id);
alter table a_paziente add constraint
fk_a_paziente_utente foreign key (cz_utente)
references z_utente(id);

— vista pazienti
create or replace view v_a_paziente as
select
ap.id, ap.dd_cognome, ap.dd_nome, ap.cd_sesso,
date_format(ap.dt_nascita, '%d/%m/%Y') dt_nascita, ap
.cd_fiscale,
ap.ct_titolo, tt.dd_descrizione dd_titolo,
ap.ct_stato_civile, tsc.dd_descrizione
dd_stato_civile,
ap.dd_medico_nome, ap.dd_medico_cognome, ap.
dd_medico_email, dd_foto,
bl_nota
from
a_paziente ap
```

```
left outer join t_titolo tt on ap.ct_titolo = tt.id
left outer join t_stato_civile tsc on ap.
    ct_stato_civile = tsc.id;
```

— *Questionari - struttura*

```
create table a_quest
(
    id integer not null auto_increment ,
    dd_descrizione varchar(100) not null ,
    dd_descr_breve varchar(50) not null ,
    primary key(id)
);
create table a_quest_sezione
(
    id integer not null auto_increment ,
    dd_descrizione1 varchar(255) not null ,
    dd_descrizione2 varchar(255) ,
    ca_quest int not null ,
    nr_num_scelte_possibili int ,
    primary key(id)
);
alter table a_quest_sezione add constraint
    fk_a_quest_sezione_1 foreign key (ca_quest)
    references a_quest(id);
create table a_quest_item
(
    id integer not null auto_increment ,
    ca_quest integer not null ,
    ca_quest_sezione int not null ,
```

```
    dd_domanda varchar(100) not null ,
    dd_immagine varchar(255) ,
    dd_risposte varchar(255) ,
    nr_order integer ,
    primary key(id)
);
alter table a_quest_item add constraint
    fk_a_quest_item_1 foreign key (ca_quest) references
    a_quest(id);
alter table a_quest_item add constraint
    fk_a_quest_item_2 foreign key (ca_quest_sezione)
    references a_quest_sezione(id);
— testata del questionario compilato
create table m_quest
(
    id integer not null auto_increment ,
    ca_quest integer not null ,
    ca_paziente integer not null ,
    dt_now datetime ,
    dd_risultato varchar(255) ,
    bl_lettera text ,
    cz_utente int not null ,
    nr_risultato1 decimal(5,2) ,
    nr_risultato2 decimal(5,2) ,
    nr_risultato3 decimal(5,2) ,
    primary key(id)
);
alter table m_quest add constraint fk_m_quest_1 foreign
    key (ca_quest) references a_quest(id);
alter table m_quest add constraint fk_m_quest_2 foreign
    key (ca_paziente) references a_paziente(id);
```

```
alter table m_quest add constraint fk_m_quest_3 foreign
  key (cz_utente) references z_utente(id);
— dettaglio del questionario compilato
create table m_quest_item
(
  id integer not null auto_increment,
  cm_quest integer not null,
  ca_quest_item integer not null,
  nr_scelta integer,
  primary key(id)
);
alter table m_quest_item add constraint
  fk_m_quest_item_1 foreign key (cm_quest) references
  m_quest(id);
alter table m_quest_item add constraint
  fk_m_quest_item_2 foreign key (ca_quest_item)
  references a_quest_item(id);
— Tabella di coefficienti per calcolo risultati test
create table z_normalizza
(
  id integer not null,
  nr_coeff decimal(5,2),
  primary key (id)
);
— Viste varie su questionari
create or replace view v_a_quest as
select id, dd_descrizione, dd_descr_breve, concat(
  dd_descr_breve, '_', dd_descrizione)
  dd_descr_completa
  from a_quest;
create or replace view v_m_quest_item as
```

```
select
  mqi.id IdMQuestItem, aqi.id IdAQuestItem, mqi.
    cm_quest CmQuest,
  aqi.dd_domanda, aqi.dd_risposte, mqi.nr_scelta, aqi.
    nr_order
from
  m_quest_item mqi, a_quest_item aqi
where
  mqi.ca_quest_item = aqi.id
order by aqi.nr_order;
— Aggiunta decodifica di cz_utente in modo ordinare
  per utente
create or replace view v_m_quest as
select
  mq.id IdMQuest, mq.ca_paziente IdPaziente, aq.id
    IdAQuest,
  aq.dd_descrizione Questionario, date_format(mq.dt_now
    , '%d/%m/%Y') Data, ap.dd_cognome Cognome, ap.
    dd_nome Nome,
  mq.dd_risultato Risultato, mq.nr_risultato1, mq.
    nr_risultato2, mq.nr_risultato3,
  Concat(ap.dd_medico_cognome, '_ ', ap.dd_medico_nome)
    dd_medico_famiglia, ap.dd_medico_email,
  Concat(zu.dd_cognome, '_ ', zu.dd_nome) dd_utente, ap.
    dd_medico_cognome, ap.dd_medico_nome,
  zu.dd_cognome dd_utente_cognome, zu.dd_nome
    dd_utente_nome
from
  m_quest mq, a_quest aq, a_paziente ap, z_utente zu
where
  mq.ca_quest = aq.id and
```

```
mq.ca_paziente = ap.id and
mq.cz_utente = zu.id;
— Domande in join con sezioni
create or replace view v_a_quest_item as
select
  aqi.id ,
  aqi.ca_quest ,
  aqi.ca_quest_sezione ,
  aqi.dd_domanda ,
  aqi.dd_immagine ,
  aqi.dd_risposte ,
  aqi.nr_order ,
  aqs.dd_descrizione1 dd_sezione_descr1 ,
  aqs.dd_descrizione2 dd_sezione_descr2 ,
  aqs.nr_num_scelte_possibili
from
  a_quest_item aqi , a_quest_sezione aqs
where
  aqi.ca_quest_sezione = aqs.id;
```

4.1.2 Stored procedure

Listing 4.2: creaSP.sql

```
use Emma;

DROP PROCEDURE IF EXISTS CalcolaLettera;
DROP FUNCTION IF EXISTS CalcolaRisultato;
DROP FUNCTION IF EXISTS getRandomIdPaziente;
DROP FUNCTION IF EXISTS getRandomIdUtenteMedico;
DROP FUNCTION IF EXISTS insertMQuest;
DROP PROCEDURE IF EXISTS generaQuest;
DROP FUNCTION IF EXISTS getRandomDate;
```

```
DROP FUNCTION IF EXISTS getTestata;
DROP PROCEDURE IF EXISTS setTestata;
DROP FUNCTION IF EXISTS getLettera;
DROP PROCEDURE IF EXISTS setLettera;
DROP FUNCTION IF EXISTS getMailAddressMedico;
DROP FUNCTION IF EXISTS getNomeMedicoConSaluto;
DROP FUNCTION IF EXISTS getNomePazienteConSaluto;
DROP FUNCTION IF EXISTS getDescrSezione1;
DROP FUNCTION IF EXISTS getDescrSezione2;
DROP FUNCTION IF EXISTS getDescrQuest;
DROP FUNCTION IF EXISTS getDescrBreveQuest;
```

```
DELIMITER $$
```

```
CREATE FUNCTION getDescrQuest(idQuest int)
returns varchar(100)
DETERMINISTIC
begin
    declare sResult varchar(100);

    select aq.dd_descrizione
        into sResult
        from a_quest aq
        where aq.id = idQuest;

    return sResult;
end
$$

CREATE FUNCTION getDescrBreveQuest(idQuest int)
returns varchar(50)
```

```
DETERMINISTIC
begin
  declare sResult varchar(50);

  select aq.dd_descr_breve
    into sResult
    from a_quest aq
    where aq.id = idQuest;

  return sResult;
end
$$

CREATE FUNCTION getDescrSezione1(idSezione int)
returns varchar(255)
DETERMINISTIC
begin
  declare sResult varchar(255);

  select aqs.dd_descrizione1
    into sResult
    from a_quest_sezione aqs
    where aqs.id = idSezione;

  return sResult;
end
$$

CREATE FUNCTION getDescrSezione2(idSezione int)
returns varchar(255)
DETERMINISTIC
```



```
begin
  declare sResult varchar(255);

  select aqs.dd_descrizione2
    into sResult
    from a_quest_sezione aqs
    where aqs.id = idSezione;

  return sResult;
end
$$

CREATE FUNCTION getNomePazienteConSaluto(idMQuest int)
returns varchar(255)
DETERMINISTIC
begin
  declare sResult varchar(255);
  declare cdSesso varchar(1);
  declare ddNome varchar(50);
  declare ddCognome varchar(50);

  select ap.cd_sesso , ap.dd_cognome , ap.dd_nome
    into cdSesso , ddCognome , ddNome
    from a_paziente ap, m_quest mq
    where mq.id = idMQuest
      and mq.ca_paziente = ap.id;

  if cdSesso = 'F' then
    select concat('sig.a_', ddNome, ' ', ddCognome)
      into sResult;
  else
```

```
        select concat('sig.␣', ddNome, '␣', ddCognome)
            into sResult;
    end if;
    return sResult;
end
$$

CREATE FUNCTION getNomeMedicoConSaluto(idMQuest int ,
    iIncludiSaluto int)
returns varchar(255)
DETERMINISTIC
begin
    declare sResult varchar(255);

    if iIncludiSaluto = 1 then
        select concat(ap.dd_medico_saluto , '␣', ap.
            dd_medico_nome , '␣', ap.dd_medico_cognome)
            into sResult
        from a_paziente ap, m_quest mq
        where mq.id = idMQuest
            and mq.ca_paziente = ap.id;
    else
        select concat(ap.dd_medico_nome , '␣', ap.
            dd_medico_cognome)
            into sResult
        from a_paziente ap, m_quest mq
        where mq.id = idMQuest
            and mq.ca_paziente = ap.id;
    end if;
    return sResult;
end
```

```
$$
```

```
CREATE FUNCTION getMailAddressMedico(idMQuest int)  
returns varchar(50)  
DETERMINISTIC  
begin  
    declare sResult varchar(50);  
  
    select ap.dd_medico_email  
        into sResult  
    from a_paziente ap, m_quest mq  
    where mq.id = idMQuest  
        and mq.ca_paziente = ap.id;  
    return sResult;  
end  
$$
```

```
CREATE FUNCTION getLettera(idMQuest int)  
returns text  
DETERMINISTIC  
begin  
    declare blTesto text;  
  
    select bl_lettera  
        into blTesto  
    from m_quest  
    where id = idMQuest;  
  
    return blTesto;  
end  
$$
```

```
CREATE PROCEDURE setLettera(idMQuest int , blLettera
    text)
DETERMINISTIC
begin
    update m_quest
        set bl_lettera = blLettera
        where id = idMQuest;
end
$$
```

```
CREATE FUNCTION getTestata(idUtente int)
returns text
DETERMINISTIC
begin
    declare blTesto text;

    select bl_testo
        into blTesto
        from z_utente_testata_lettera
        where cz_utente = idUtente;

    return blTesto;
end
$$
```

```
CREATE PROCEDURE setTestata(idUtente int , blTesto text)
DETERMINISTIC
begin
    declare iCtr int;
```

```
— Verifica se esiste già il record per idUtente
select count(*)
  into iCtr
  from z_utente_testata_lettera
 where cz_utente = idUtente;

if iCtr = 0 then
  insert into z_utente_testata_lettera(cz_utente,
    bl_testo)
    values (idUtente, blTesto);
else
  update z_utente_testata_lettera
    set bl_testo = blTesto
    where cz_utente = idUtente;
end if;

end
$$
```

```
CREATE FUNCTION CalcolaRisultato (idMQuest int)
returns varchar(255)
DETERMINISTIC
begin
  declare rCoeff float;
  declare sResult varchar(255);
  declare iResult int;
  declare i int;
  set i = 1;
  set sResult = '';
  while i <= 3
  do
```

```
select sum(mqi.nr_scelta)
  into iResult
  from m_quest_item mqi, a_quest_item aqi
 where mqi.cm_quest = idMQuest
       and mqi.ca_quest_item = aqi.id
       and aqi.ca_quest_sezione = i;

— L'ultima sezione si divide per 3
if i = 3 then
  set rCoeff = iResult / 3;
else
select nr_coeff
  into rCoeff
  from z_normalizza
 where id = iResult;
end if;

set sResult = concat(sResult, ' ', cast(rCoeff as
  char(10)));

— aggiorna i tre risultati delle tre sezioni del
  PRO-CLARA
case i
  when 1 then
    update m_quest mq
      set mq.nr_risultato1 = rCoeff
      where mq.id = idMQuest;
  when 2 then
    update m_quest mq
      set mq.nr_risultato2 = rCoeff
      where mq.id = idMQuest;
```

```
        when 3 then
            update m_quest mq
                set mq.nr_risultato3 = rCoeff
                where mq.id = idMQuest;
        end case;
        set i = i + 1;
    end while;
    return sResult;
END
$$

CREATE PROCEDURE CalcolaLettera (idMQuest int, idUtente
    int)
DETERMINISTIC
begin
    declare tempLettera text;
    declare cdSesso varchar(1);
    declare ddNome varchar(50);
    declare ddCognome varchar(50);
    declare ddMedicoNome varchar(50);
    declare ddMedicoCognome varchar(50);
    declare ddMedicoSaluto varchar(50);
    declare iAnnoCorr int;
    declare iAnnoNascita int;
    declare nrRisultato1 float;
    declare nrRisultato2 float;
    declare nrRisultato3 float;
    declare dtTest date;
    declare idAQuest int; -- sarebbe una costante (1 il
        pro-clara)
```

```

set idAQuest = 1;

set tempLettera = '';

— Calcola etÃ e saluto (sig. o sig.a)
select extract(year from dt_nascita), year(curdate())
, cd_sesso, dd_nome, dd_cognome,
      ifNull(dd_medico_nome, ''), ifNull(
      dd_medico_cognome, ''), ifNull(
      dd_medico_saluto, '')
into iAnnoNascita, iAnnoCorr, cdSesso, ddNome,
      ddCognome,
      ddMedicoNome, ddMedicoCognome, ddMedicoSaluto
from a_paziente ap, m_quest mq
where mq.id = idMQuest
      and mq.ca_paziente = ap.id;

— Legge risultati test
select dt_now, nr_risultato1, nr_risultato2,
      nr_risultato3
into dtTest, nrRisultato1, nrRisultato2,
      nrRisultato3
from m_quest
where id = idMQuest;

set tempLettera = concat(tempLettera, '\n\r\n\r',
      trim(ddMedicoSaluto), '_ ', trim(ddMedicoCognome), '
\n\r\n\r');
set tempLettera = concat(tempLettera, 'In_data_',
      date_format(dtTest, '%e/%m/%Y'), '_ho_sottoposto_')
;

```



```

if cdSesso = 'M' then
  set tempLettera = concat(tempLettera, 'il sig. ');
else
  set tempLettera = concat(tempLettera, 'la sig. a ');
end if;
— Nome e cognome e et  del paziente
set tempLettera = concat(tempLettera, ddNome, ' ',
  ddCognome, ' di ', iAnnoCorr - iAnnoNascita, ' anni
  ');

— Risultati test
set tempLettera = concat(tempLettera, '\n\r\n\r',
  'al test ', getDescrBreveQuest(idAQuest), '( ',
  getDescrQuest(idAQuest), ') e i risultati sono
  stati i seguenti: ', '\n\r\n\r');

set tempLettera = concat(tempLettera, 'sezione 1(',
  getDescrSezione1(1), ') : ', cast(nrRisultato1 as
  char(10)), '\n\r\n\r');
set tempLettera = concat(tempLettera, 'sezione 2(',
  getDescrSezione1(2), ') : ', cast(nrRisultato2 as
  char(10)), '\n\r\n\r');
set tempLettera = concat(tempLettera, 'sezione 3(',
  getDescrSezione1(3), ') : ', cast(nrRisultato3 as
  char(10)), '\n\r\n\r');

— Salva lettera costruita in m_quest
update m_quest
  set bl_lettera = tempLettera

```

```
        where id = idMQuest;
end
$$

CREATE FUNCTION getRandomIdPaziente() returns int
begin
    declare idResult int;

    select id
        into idResult
        from a_paziente
        order by rand() limit 1;
    return idResult;
end
$$

CREATE FUNCTION getRandomIdUtenteMedico() returns int
begin
    declare idResult int;

    select id
        into idResult
        from z_utente
        where id <= 4 — prende solo utenti "medici"
        order by rand() limit 1;
    return idResult;
end
$$

— Crea nuovo record in M_QUEST e ritorna l'id generato
```

```
CREATE FUNCTION insertMQuest(caQuest int, caPaziente
  int, czUtente int, dtQuest datetime) returns int
begin
  declare idLast int;
  insert into m_quest(ca_quest, ca_paziente, dt_now,
    cz_utente)
    values (caQuest, caPaziente, dtQuest, czUtente);
  select last_insert_id()
    into idLast;
  return idLast;
end
$$
```

— *Costruisce una data partendo dall'1/1 dell'anno
corrente*

— *e aggiungendo un giorno random*

```
CREATE FUNCTION getRandomDate() returns datetime
begin
  declare iDd int;
  declare dtFirst datetime;
  declare dtResult datetime;

  set iDd = floor(rand()*365);
  select date_format(now(), '%Y-1-1')
    into dtFirst;

  select date_add(dtFirst, interval iDd day)
    into dtResult;
  return dtResult;
end
$$
```

```
— Genera iNumQuest questionari scegliendo un paziente
  e un utente random (e random anche le risposte)
CREATE PROCEDURE generaQuest(iNumQuest int, idAQuest
int)
begin
  declare i int;
  declare idPaziente int;
  declare idUtente int;
  declare idMQuest int;
  declare iSezione int;
  declare nrScelte int;
  declare ddRisultato varchar(255);

  set i = 1;

  while i <= iNumQuest
  do
    — Determina id paziente
    select getRandomIdPaziente() into idPaziente;
    — Id utente
    select getRandomIdUtenteMedico() into idUtente;
    — Genera testata questionario
    select insertMQuest(idAQuest, idPaziente, idUtente,
      getRandomDate()) into idMQuest;

    set iSezione = 1;

    — Loop sulle sezioni del questionario Clara per
      inserire le risposte
    while iSezione <= 3
```

```
do
  select nr_num_scelte_possibili
    into nrScelte
    from a_quest_sezione
  where id = iSezione;

  insert into m_quest_item(cm_quest, ca_quest_item,
    nr_scelta)
  select idMQuest, id, floor(rand() * nrScelte)
    from a_quest_item
  where ca_quest = idAQuest and ca_quest_sezione =
    iSezione;

  set iSezione = iSezione + 1;
end while;

— Calcola i risultati
select CalcolaRisultato(idMQuest)
  into ddRisultato;

update m_quest
set dd_risultato = ddRisultato
where id = idMQuest;

— Costruisce la lettera
call CalcolaLettera(idMQuest, idUtente);

— Passa al questionario successivo
set i = i + 1;
end while;
end
```

\$\$

DELIMITER ;

4.1.3 Caricamento dati di prova

Listing 4.3: insertData.sql

```
— profili
insert into z_profilo(id,dd_descrizione) values (1,'
  Pubblico');
insert into z_profilo(id,dd_descrizione) values (2,'
  Operatore');
insert into z_profilo(id,dd_descrizione) values (3,'
  Medico');
insert into z_profilo(id,dd_descrizione) values (4,'
  Amministratore');
— utenti
insert into z_utente(id, dd_login, dd_password, dd_nome
  , dd_cognome, cz_profilo)
values (1, 'farinan', 'farinan', 'Nicola', 'Farina', 4)
  ;
insert into z_utente(id, dd_login, dd_password, dd_nome
  , dd_cognome, cz_profilo)
values (2, 'montesid', 'montesid', 'Danilo', 'Montesi',
  3);
insert into z_utente(id, dd_login, dd_password, dd_nome
  , dd_cognome, cz_profilo)
values (3, 'farinas', 'farinas', 'Stefano', 'Farina',
  3);
insert into z_utente(id, dd_login, dd_password, dd_nome
  , dd_cognome, cz_profilo)
```

```
values (4, 'salaffif', 'salaffif', 'Fausto', 'Salaffi',
      3);
insert into z_utente(id, dd_login, dd_password, dd_nome
, dd_cognome, cz_profilo)
values (5, 'rossim', 'rossim', 'Mario', 'Rossi', 2);
insert into z_utente(id, dd_login, dd_password, dd_nome
, dd_cognome, cz_profilo)
values (6, 'guest', 'guest', 'Utente', 'Ospite', 1);
— testata lettera
insert into z_utente_testata_lettera(id, cz_utente,
  bl_testo)
values (1, 1, concat_ws('\n', 'Dott. Manlio
  Scordabattaglia\nProfessore Associato di
  Reumatologia\n',
'Clinica Reumatologica\n—Scuola di Specializzazione in
  Reumatologia',
'Ospedale A. Zamboni ASLBO Zona Centrale 1',
'Via Mura Anteo Zamboni 13\n—44100 Bologna (BO)',
'tel_051_123424_—_123123_—_123123_—_Fax_051_123123'
));
insert into z_utente_testata_lettera(id, cz_utente,
  bl_testo)
values (2, 2, concat_ws('\n', 'Dott. Danilo Montesi\n
  nProfessore Associato di Reumatologia\n',
'Clinica Reumatologica\n—Scuola di Specializzazione in
  Reumatologia',
'Ospedale A. Zamboni ASLBO Zona Centrale 1',
'Via Mura Anteo Zamboni 13\n—44100 Bologna (BO)',
'tel_051_123424_—_123123_—_123123_—_Fax_051_123123'
));
```

```

insert into z_utente_testata_lettera(id, cz_utente,
    bl_testo)
values (3, 3, concat_ws('\n', 'Dott. Stefano Farina\
    nProfessore Associato di Reumatologia\n',
    'Clinica Reumatologica\-\ Scuola di Specializzazione in
    Reumatologia',
    'Ospedale A. Zamboni ASLBO Zona Centrale 1',
    'Via Mura Anteo Zamboni 13\-\ 44100 Bologna (BO)',
    'tel 051 123424\-\ 123123\-\ 123123\-\ Fax 051 123123'
));

```

```

insert into z_utente_testata_lettera(id, cz_utente,
    bl_testo)
values (4, 4, concat_ws('\n', 'Dott. Fausto Salaffi\
    nProfessore Associato di Reumatologia\n',
    'Clinica Reumatologica\-\ Scuola di Specializzazione in
    Reumatologia',
    'Ospedale A. Murri ASUR Zona Territoriale 5',
    'Via dei Colli, 52\-\ 60035 Jesi (AN)',
    'tel 051 123424\-\ 123123\-\ 123123\-\ Fax 051 123123'
));

```

— *funzioni*

— *cz_profilo* Ã” il profilo minimo per cui la funzione
Ã” visibile

```

insert into z_funzione(id, dd_descrizione,
    dd_descr_breve, dd_url, cz_profilo, id_parent)
values (1, 'Root', 'Root', '', 1, -1);
insert into z_funzione(id, dd_descrizione,
    dd_descr_breve, dd_url, cz_profilo, id_parent)
values (2, 'Home', 'Home', '~/Default.aspx', 1, -1);

```



```
insert into z_funzione(id, dd_descrizione,
    dd_descr_breve, dd_url, cz_profilo, id_parent)
values (3, 'About', 'About', '~/About.aspx', 1, -1);
-- tabelle
insert into z_funzione(id, dd_descrizione,
    dd_descr_breve, dd_url, cz_profilo)
values (4, 'Tabelle', 'Tabelle', '', 3);
insert into z_funzione(id, dd_descrizione,
    dd_descr_breve, dd_url, cz_profilo, id_parent)
values (5, 'Pazienti', 'Pazienti', '~/Tabelle/
    FmPazienti.aspx', 3, 4);
insert into z_funzione(id, dd_descrizione,
    dd_descr_breve, dd_url, cz_profilo, id_parent)
values (6, 'Titoli_di_studio', 'TitoliDiStudio', '~/
    Tabelle/FmTitoliDiStudio.aspx', 2, 4);
insert into z_funzione(id, dd_descrizione,
    dd_descr_breve, dd_url, cz_profilo, id_parent)
values (7, 'Stato_civile', 'StatoCivile', '~/Tabelle/
    FmStatoCivile.aspx', 2, 4);
-- questionari
insert into z_funzione(id, dd_descrizione,
    dd_descr_breve, dd_url, cz_profilo)
values (8, 'Questionari', 'Questionari', '', 3);
insert into z_funzione(id, dd_descrizione,
    dd_descr_breve, dd_url, cz_profilo, id_parent)
values (9, 'Somministra', 'Somministra', '~/Questionari
    /FmCompila.aspx', 2, 8);
insert into z_funzione(id, dd_descrizione,
    dd_descr_breve, dd_url, cz_profilo, id_parent)
values (10, 'Archivio', 'Archivio', '~/Questionari/
    FmArchivio.aspx', 2, 8);
```

```
insert into z_funzione(id, dd_descrizione ,
    dd_descr_breve , dd_url , cz_profilo , id_parent)
values (11, 'Struttura', 'Struttura', '~/Questionari/
    FmStruttura.aspx', 3, 8);
— Amministrazione
insert into z_funzione(id, dd_descrizione ,
    dd_descr_breve , dd_url , cz_profilo)
values (12, 'Amministrazione', 'Amministrazione', '',
    4);
insert into z_funzione(id, dd_descrizione ,
    dd_descr_breve , dd_url , cz_profilo , id_parent)
values (13, 'Utenti', 'Utenti', '~/Amministrazione/
    FmUtenti.aspx', 4, 12);
— insert into z_funzione(id, dd_descrizione ,
    dd_descr_breve , dd_url , cz_profilo , id_parent)
— values (14, 'Profili', 'Profili', '~/Amministrazione
    /FmProfili.aspx', 4, 12);
insert into z_funzione(id, dd_descrizione ,
    dd_descr_breve , dd_url , cz_profilo , id_parent)
values (15, 'Log', 'Log', '~/Amministrazione/FmLog.aspx
    ', 4, 12);
— patologia
insert into t_patologia(id, dd_descrizione) values (1,
    'Artrite_reumatoide');
insert into t_patologia(id, dd_descrizione) values (2,
    'Fibromialgia');
insert into t_patologia(id, dd_descrizione) values (3,
    'Gonartrosi_sintomatica');
— stato civile
insert into t_stato_civile(id, dd_descrizione) values
    (1, 'Nubile/Celibe');
```

```
insert into t_stato_civile(id, dd_descrizione) values
  (2, 'Coniugata/Coniugato');
— titolo di studio
insert into t_titolo(id, dd_descrizione) values (1, '
  Nessuno');
insert into t_titolo(id, dd_descrizione) values (2, '
  Licenza_elementare');
insert into t_titolo(id, dd_descrizione) values (3, '
  Licenza_media_inferiore');
insert into t_titolo(id, dd_descrizione) values (4, '
  Licenza_media_superiore');
insert into t_titolo(id, dd_descrizione) values (5, '
  Laurea');
insert into t_titolo(id, dd_descrizione) values (6, '
  Specializzazione_post/laurea');
— pazienti
insert into a_paziente(id, cd_sesso, dd_cognome, dd_nome,
  ct_titolo, dt_nascita, cd_fiscale, ct_stato_civile,
  dd_medico_nome, dd_medico_cognome, dd_medico_email,
  dd_medico_saluto, dd_foto)
values (1, 'F', 'Marzano', 'Michela', '6', '1968-04-23T00
  :00:00', 'MRZMCH12F25G123A', '2',
  'Danilo', 'Montesi', 'danilo.montesi@gmail.com', 'Caro_
  Dott.', 'MichelaMarzano.jpeg');
insert into a_paziente(id, cd_sesso, dd_cognome, dd_nome,
  ct_titolo, dt_nascita, cd_fiscale, ct_stato_civile,
  dd_medico_nome, dd_medico_cognome, dd_medico_email,
  dd_medico_saluto, dd_foto)
values (2, 'M', 'Pollini', 'Maurizio', '3', '1956-09-24T00
  :00:00', 'PLLMRZ56F12G512A', '2',
```

```
'Danilo', 'Montesi', 'danilo.montesi@gmail.com', 'Caro_
Dott._', 'MaurizioPollini.jpeg');
insert into a_paziente(id, cd_sesso, dd_cognome, dd_nome,
    ct_titolo, dt_nascita, cd_fiscale, ct_stato_civile,
dd_medico_nome, dd_medico_cognome, dd_medico_email,
    dd_medico_saluto, dd_foto)
values (3, 'F', 'Argerich', 'Martha', '2', '1957-11-14T00
:00:00', 'ARGMRT57A14G452A', '2',
'Danilo', 'Montesi', 'danilo.montesi@gmail.com', 'Caro_
Dott._', 'MarthaArgerich.jpeg');
insert into a_paziente(id, cd_sesso, dd_cognome, dd_nome,
    ct_titolo, dt_nascita, cd_fiscale, ct_stato_civile,
dd_medico_nome, dd_medico_cognome, dd_medico_email,
    dd_medico_saluto, dd_foto)
values (4, 'M', 'Ciccolini', 'Aldo', '3', '1982-12-27T00
:00:00', 'CLNALD82A23G532A', '1',
'Stefano', 'Farina', 'nicola.farina@teletu.it', 'Caro_
Dott._', 'AldoCiccolini.jpeg');
insert into a_paziente(id, cd_sesso, dd_cognome, dd_nome,
    ct_titolo, dt_nascita, cd_fiscale, ct_stato_civile,
dd_medico_nome, dd_medico_cognome, dd_medico_email,
    dd_medico_saluto, dd_foto)
values (5, 'F', 'Saraceno', 'Chiara', '4', '1964-05-22T00
:00:00', 'SRCCHR64S23G532A', '1',
'Stefano', 'Farina', 's.farina@hcrema.it', 'Caro_Dott._
', 'ChiaraSaraceno.jpeg');
insert into a_paziente(id, cd_sesso, dd_cognome, dd_nome,
    ct_titolo, dt_nascita, cd_fiscale, ct_stato_civile,
dd_medico_nome, dd_medico_cognome, dd_medico_email,
    dd_medico_saluto, dd_foto)
```

```

values (6, 'F', 'Hewitt', 'Angela', '1', '1980-08-09T00
:00:00', 'HWTANG80A12G231A', '1',
'Stefano', 'Farina', 'nicola.farina@teletu.it', 'Caro_
Dott._', 'AngelaHewitt.jpeg');

```

```

insert into a_paziente(id, cd_sesso, dd_cognome, dd_nome,
ct_titolo, dt_nascita, cd_fiscale, ct_stato_civile,
dd_medico_nome, dd_medico_cognome, dd_medico_email,
dd_medico_saluto, dd_foto)

```

```

values (7, 'F', 'Urbinati', 'Nadia', '5', '1984-05-18T00
:00:00', 'URBNDI84A12G424D', '2',
'Stefano', 'Farina', 'nicola.farina@teletu.it', 'Caro_
Dott._', 'NadiaUrbinati.jpeg');

```

— *questionari - testata*

```

insert into a_quest(id, dd_descr_breve, dd_descrizione)

```

```

values (1, 'PRO-CLARA', 'PRO-Clinical_Arthritis_
Activity_(CLARA)_Index');

```

— *questionari - sezioni*

```

insert into a_quest_sezione(id, ca_quest,
dd_descrizione1, dd_descrizione2,
nr_num_scelte_possibili)

```

```

values (1, 1, 'Recent-Onset_Arthritis_Disability_(ROAD)
_Index', 'Nel_corso_dell''ultima_settimana_Ã" stato_
in_grado_di:', 5);

```

```

insert into a_quest_sezione(id, ca_quest,
dd_descrizione1, dd_descrizione2,
nr_num_scelte_possibili)

```

```

values (2, 1, 'Autovalutazione_articolazioni_dolenti',
'Indichi_l''entitÃ del_dolore_che_sente_attualmente_
nell''articolazione_indicata:', 4);

```

```

insert into a_quest_sezione(id, ca_quest,
    dd_descrizione1, dd_descrizione2,
    nr_num_scelte_possibili)
values (3, 1, 'Stato_generale_di_salute', 'Indichi_l''
    effetto_che_la_malattia_ha_avuto_sul_suo_stato_
    generale_di_salute_nel_corso_dell''ultima_settimana:_
    ', 11);
— questionari - domande
insert into a_quest_item(id, ca_quest, ca_quest_sezione
    , dd_domanda, dd_immagine, dd_risposte, nr_order)
values (1, 1, 1, 'Chiudere_completamente_la_mano_a_
    pugno?', 'ChiudereLaMano.jpg', 'Senza_difficoltÃ |Con_
    _lieve_difficoltÃ |Con_qualche_difficoltÃ |Con_molta_
    _difficoltÃ |No,Ã"stato_impossibile', 1);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
    , dd_domanda, dd_immagine, dd_risposte, nr_order)
values (2, 1, 1, 'Accettare_una_stretta_di_mano?', '
    StrettaDiMano.jpg', 'Senza_difficoltÃ |Con_lieve_
    _difficoltÃ |Con_qualche_difficoltÃ |Con_molta_
    _difficoltÃ |No,Ã"stato_impossibile', 2);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
    , dd_domanda, dd_immagine, dd_risposte, nr_order)
values (3, 1, 1, 'Abbottonarsi_gli_abiti?', '
    Abbottonarsi.jpg', 'Senza_difficoltÃ |Con_lieve_
    _difficoltÃ |Con_qualche_difficoltÃ |Con_molta_
    _difficoltÃ |No,Ã"stato_impossibile', 3);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
    , dd_domanda, dd_immagine, dd_risposte, nr_order)
values (4, 1, 1, 'Svitare_il_barattolo_di_un_coperchio_
    giÃ_aperto_in_precedenza?', 'SvitareBarattolo.jpg',
    'Senza_difficoltÃ |Con_lieve_difficoltÃ |Con_qualche_

```

```
difficoltÃ |Con_molta_difficoltÃ |No,Ã stato_
impossibile', 4);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (5, 1, 1, 'Raggiungere_e_afferrare_unoggetto_
del_peso_di_circa_2_kg_posto_sopra_la_Sua_testa?', '',
, 'Senza_difficoltÃ |Con_lieve_difficoltÃ |Con_
qualche_difficoltÃ |Con_molta_difficoltÃ |No,Ã_
stato_impossibile', 5);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (6, 1, 1, 'Stare_in_piedi_in_posizione_eretta?',
'', 'Senza_difficoltÃ |Con_lieve_difficoltÃ |Con_
qualche_difficoltÃ |Con_molta_difficoltÃ |No,Ã_
stato_impossibile', 6);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (7, 1, 1, 'Camminare_su_un_terreno_piano?', '',
, 'Senza_difficoltÃ |Con_lieve_difficoltÃ |Con_qualche_
difficoltÃ |Con_molta_difficoltÃ |No,Ã stato_
impossibile', 7);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (8, 1, 1, 'Salire_un_piano_di_scale?', '', '
Senza_difficoltÃ |Con_lieve_difficoltÃ |Con_qualche_
difficoltÃ |Con_molta_difficoltÃ |No,Ã stato_
impossibile', 8);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (9, 1, 1, 'Salire_e_scendere_dall''automobile?',
'', 'Senza_difficoltÃ |Con_lieve_difficoltÃ |Con_
```

```

qualche_difficoltÃ |Con_molta_difficoltÃ |No,Ã"
stato_impossibile', 9);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (10, 1, 1, 'Lavare_e_asciugare_tutto_il_corpo?',
'', 'Senza_difficoltÃ |Con_lieve_difficoltÃ |Con_
qualche_difficoltÃ |Con_molta_difficoltÃ |No,Ã"
stato_impossibile', 10);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (11, 1, 1, 'Fare_attivitÃ _vigorose_quali_
trasportare_oggetti_e_borse_pesanti?', '', 'Senza_
difficoltÃ |Con_lieve_difficoltÃ |Con_qualche_
difficoltÃ |Con_molta_difficoltÃ |No,Ã"stato_
impossibile', 11);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (12, 1, 1, 'Svolgere_un_lavoro_retribuito_o_
attivitÃ _domestiche?', '', 'Senza_difficoltÃ |Con_
lieve_difficoltÃ |Con_qualche_difficoltÃ |Con_molta_
difficoltÃ |No,Ã"stato_impossibile', 12);
— sezione 2 – parte sinistra
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (13, 1, 2, 'Mano_sinistra', '', 'Assente|Lieve|
Moderato|Forte', 13);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (14, 1, 2, 'Polso_sinistro', '', 'Assente|Lieve|
Moderato|Forte', 14);

```



```
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (15, 1, 2, 'Gomito_sinistro', '', 'Assente|Lieve
|Moderato|Forte', 15);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (16, 1, 2, 'Spalla_sinistra', '', 'Assente|Lieve
|Moderato|Forte', 16);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (17, 1, 2, 'Anca_sinistra', '', 'Assente|Lieve|
Moderato|Forte', 17);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (18, 1, 2, 'Ginocchio_sinistro', '', 'Assente|
Lieve|Moderato|Forte', 18);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (19, 1, 2, 'Caviglia_sinistra', '', 'Assente|
Lieve|Moderato|Forte', 19);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (20, 1, 2, 'Piede_sinistro', '', 'Assente|Lieve|
Moderato|Forte', 20);
— sezione 2 — parte destra
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (21, 1, 2, 'Mano_destra', '', 'Assente|Lieve|
Moderato|Forte', 21);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
```

```
values (22, 1, 2, 'Polso_destro', '', 'Assente|Lieve|
Moderato|Forte', 22);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (23, 1, 2, 'Gomito_destro', '', 'Assente|Lieve|
Moderato|Forte', 23);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (24, 1, 2, 'Spalla_destra', '', 'Assente|Lieve|
Moderato|Forte', 24);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (25, 1, 2, 'Anca_destra', '', 'Assente|Lieve|
Moderato|Forte', 25);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (26, 1, 2, 'Ginocchio_destro', '', 'Assente|
Lieve|Moderato|Forte', 26);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (27, 1, 2, 'Caviglia_destra', '', 'Assente|Lieve
|Moderato|Forte', 27);
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (28, 1, 2, 'Piede_destro', '', 'Assente|Lieve|
Moderato|Forte', 28);
— Sezione 3
insert into a_quest_item(id, ca_quest, ca_quest_sezione
, dd_domanda, dd_immagine, dd_risposte, nr_order)
values (29, 1, 3, '', '', '0|1|2|3|4|5|6|7|8|9|10', 29)
;
```

— *normalizza per risultati questionario*

```
insert into z_normalizza(id, nr_coeff)
values (1, 0.2);
insert into z_normalizza(id, nr_coeff)
values (2, 0.4);
insert into z_normalizza(id, nr_coeff)
values (3, 0.6);
insert into z_normalizza(id, nr_coeff)
values (4, 0.8);
insert into z_normalizza(id, nr_coeff)
values (5, 1.0);
insert into z_normalizza(id, nr_coeff)
values (6, 1.3);
insert into z_normalizza(id, nr_coeff)
values (7, 1.5);
insert into z_normalizza(id, nr_coeff)
values (8, 1.7);
insert into z_normalizza(id, nr_coeff)
values (9, 1.9);
insert into z_normalizza(id, nr_coeff)
values (10, 2.1);
insert into z_normalizza(id, nr_coeff)
values (11, 2.3);
insert into z_normalizza(id, nr_coeff)
values (12, 2.5);
insert into z_normalizza(id, nr_coeff)
values (13, 2.7);
insert into z_normalizza(id, nr_coeff)
values (14, 2.9);
insert into z_normalizza(id, nr_coeff)
values (15, 3.1);
```

```
insert into z_normalizza(id, nr_coeff)
values (16, 3.3);
insert into z_normalizza(id, nr_coeff)
values (17, 3.5);
insert into z_normalizza(id, nr_coeff)
values (18, 3.8);
insert into z_normalizza(id, nr_coeff)
values (19, 4.0);
insert into z_normalizza(id, nr_coeff)
values (20, 4.2);
insert into z_normalizza(id, nr_coeff)
values (21, 4.4);
insert into z_normalizza(id, nr_coeff)
values (22, 4.6);
insert into z_normalizza(id, nr_coeff)
values (23, 4.8);
insert into z_normalizza(id, nr_coeff)
values (24, 5.0);
insert into z_normalizza(id, nr_coeff)
values (25, 5.2);
insert into z_normalizza(id, nr_coeff)
values (26, 5.4);
insert into z_normalizza(id, nr_coeff)
values (27, 5.6);
insert into z_normalizza(id, nr_coeff)
values (28, 5.8);
insert into z_normalizza(id, nr_coeff)
values (29, 6.0);
insert into z_normalizza(id, nr_coeff)
values (30, 6.3);
insert into z_normalizza(id, nr_coeff)
```

```
values (31, 6.5);
insert into z_normalizza(id, nr_coeff)
values (32, 6.7);
insert into z_normalizza(id, nr_coeff)
values (33, 6.9);
insert into z_normalizza(id, nr_coeff)
values (34, 7.1);
insert into z_normalizza(id, nr_coeff)
values (35, 7.3);
insert into z_normalizza(id, nr_coeff)
values (36, 7.5);
insert into z_normalizza(id, nr_coeff)
values (37, 7.7);
insert into z_normalizza(id, nr_coeff)
values (38, 7.9);
insert into z_normalizza(id, nr_coeff)
values (39, 8.1);
insert into z_normalizza(id, nr_coeff)
values (40, 8.3);
insert into z_normalizza(id, nr_coeff)
values (41, 8.5);
insert into z_normalizza(id, nr_coeff)
values (42, 8.8);
insert into z_normalizza(id, nr_coeff)
values (43, 9.0);
insert into z_normalizza(id, nr_coeff)
values (44, 9.2);
insert into z_normalizza(id, nr_coeff)
values (45, 9.4);
insert into z_normalizza(id, nr_coeff)
values (46, 9.6);
```

```
insert into z_normalizza(id, nr_coeff)
values (47, 9.8);
insert into z_normalizza(id, nr_coeff)
values (48, 10);
```

4.2 Sito web

4.2.1 Root directory

Listing 4.4: web.config

```
<?xml version="1.0"?>
<!--
Note: As an alternative to hand editing this file
you can use the
web admin tool to configure settings for your
application. Use
the Website->Asp.Net Configuration option in Visual
Studio.
A full list of settings and comments can be found
in
machine.config.comments usually located in
\Windows\Microsoft.Net\Framework\v2.x\Config
-->
<configuration>
  <configSections>
    <sectionGroup name="system.web.extensions" type="
      System.Web.Configuration.
      SystemWebExtensionsSectionGroup, System.Web.
      Extensions, Version=3.5.0.0, Culture=neutral,
      PublicKeyToken=31BF3856AD364E35">
    <sectionGroup name="scripting" type="System.Web.
      Configuration.ScriptingSectionGroup, System.Web
```

```
. Extensions , _ Version = 3.5.0.0 , _ Culture = neutral , _
PublicKeyToken = 31BF3856AD364E35">
<section name="scriptResourceHandler" type="
System.Web.Configuration.
ScriptingScriptResourceHandlerSection , _ System
.Web.Extensions , _ Version = 3.5.0.0 , _ Culture =
neutral , _ PublicKeyToken = 31BF3856AD364E35"
requirePermission="false" allowDefinition="
MachineToApplication" />
<sectionGroup name="webServices" type="System.
Web.Configuration.
ScriptingWebServicesSectionGroup , _ System.Web.
Extensions , _ Version = 3.5.0.0 , _ Culture = neutral ,
_PublicKeyToken = 31BF3856AD364E35">
<section name="jsonSerialization" type="
System.Web.Configuration.
ScriptingJsonSerializationSection , _ System.
Web.Extensions , _ Version = 3.5.0.0 , _ Culture =
neutral , _ PublicKeyToken = 31BF3856AD364E35"
requirePermission="false" allowDefinition="
Everywhere" />
<section name="profileService" type="System.
Web.Configuration.
ScriptingProfileServiceSection , _ System.Web.
Extensions , _ Version = 3.5.0.0 , _ Culture =
neutral , _ PublicKeyToken = 31BF3856AD364E35"
requirePermission="false" allowDefinition="
MachineToApplication" />
<section name="authenticationService" type="
System.Web.Configuration.
ScriptingAuthenticationServiceSection , _
```

```

        System.Web.Extensions , Version=3.5.0.0 ,
        Culture=neutral , PublicKeyToken=31
        BF3856AD364E35" requirePermission="false"
        allowDefinition="MachineToApplication" />
<section name="roleService" type="System.Web.
    Configuration.ScriptingRoleServiceSection ,
    System.Web.Extensions , Version=3.5.0.0 ,
    Culture=neutral , PublicKeyToken=31
    BF3856AD364E35" requirePermission="false"
    allowDefinition="MachineToApplication" />
    </sectionGroup>
</sectionGroup>
</sectionGroup>
</configSections>
<appSettings>
    <add key="ChartImageHandler" value="storage=file;
        timeout=20;dir=c:\TempImageFiles\" />
</appSettings>
<connectionStrings>
    <!-- <add name="MySQLEmma" connectionString="
        Datasource=62.149.150.124;uid=Sql387638;Pwd=2095
        b2e8;Database=Sql387638_1;" /> -->
    <add name="MySQLEmma" connectionString="Datasource=
        localhost;uid=root;Pwd=nik900;Database=Emma;" />
</connectionStrings>
<system.web>
    <!--
        Set compilation debug="true" to insert
        debugging
        symbols into the compiled page. Because
        this

```


affects performance, set **this** value to **true** only during development.

—>

```
<compilation debug="true">
  <assemblies>
    <add assembly="System.Core, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089" />
    <add assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
    <add assembly="System.Data.DataSetExtensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089" />
    <add assembly="System.Xml.Linq, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089" />
    <add assembly="DevExpress.Web.ASPxEditors.v9.3, Version=9.3.6.0, Culture=neutral, PublicKeyToken=B88D1754D700E49A" />
    <add assembly="System.Windows.Forms, Version=2.0.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089" />
    <add assembly="System.Design, Version=2.0.0.0, Culture=neutral, PublicKeyToken=B03F5F7F11D50A3A" />
    <add assembly="System.Web.DataVisualization, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
```

```

    <add assembly="DevExpress.Web.ASPxGridView.v9
      .3.Export, Version=9.3.6.0, Culture=neutral,
      PublicKeyToken=B88D1754D700E49A" />
    <add assembly="DevExpress.XtraPrinting.v9.3,
      Version=9.3.6.0, Culture=neutral,
      PublicKeyToken=B88D1754D700E49A" />
    <add assembly="DevExpress.XtraEditors.v9.3,
      Version=9.3.6.0, Culture=neutral,
      PublicKeyToken=B88D1754D700E49A" />
  </assemblies>
</compilation>
<!--
    The <authentication> section enables
    configuration
    of the security authentication mode used by
    ASP.NET to identify an incoming user.
  -->
<authentication mode="Forms">
  <forms loginUrl="~/Account/Login.aspx" timeout="
    2880" />
</authentication>
<membership defaultProvider="EmmaMembershipProvider
">
  <providers>
    <clear />
    <add name="EmmaMembershipProvider" type="
      EmmaMembershipProvider" connectionString="
      Datasource=localhost;uid=root;Pwd=nik900;
      Database=Emma;" />
  <!--      <add name="EmmaMembershipProvider"
    type="EmmaMembershipProvider"

```

```
        connectionString=" Datasource =62.149.150.124;
        uid=Sql387638;Pwd=2095b2e8;Database=
        Sql387638_1;" /> -->
    </providers>
</membership>
<siteMap>
    <providers>
        <add name=" EmmaSitemapProvider" type="
        EmmaSitemapProvider" securityTrimmingEnabled=
        " false" connectionString=" Datasource=
        localhost;uid=root;Pwd=nik900;Database=Emma;"
        />
        <!--          <add name=" EmmaSitemapProvider"
        type=" EmmaSitemapProvider"
        securityTrimmingEnabled=" false"
        connectionString=" Datasource =62.149.150.124;
        uid=Sql387638;Pwd=2095b2e8;Database=
        Sql387638_1;" /> -->
    </providers>
</siteMap>
<roleManager enabled=" true" defaultProvider="
    EmmaRoleProvider" cacheRolesInCookie=" true"
    cookieName=" .ASPXROLES" cookieTimeout=" 30"
    cookiePath=" /" cookieRequireSSL=" false"
    cookieSlidingExpiration=" true" cookieProtection="
    All">
    <providers>
        <add name=" EmmaRoleProvider" type="
        EmmaRoleProvider" connectionString="
        Datasource=localhost;uid=root;Pwd=nik900;
        Database=Emma;" applicationName="Emma" />
```

```

<!--      <add name="EmmaRoleProvider" type="
EmmaRoleProvider" connectionString="
Datasource=62.149.150.124;uid=Sql387638;Pwd
=2095b2e8;Database=Sql387638_1;"
applicationName="Emma"/> -->

```

```
</providers>
```

```
</roleManager>
```

```
<!--
```

The <customErrors> section enables configuration of what to **do if**/when an unhandled error occurs during the execution of a request. Specifically, it enables developers to configure html error pages to be displayed **in** place of a error stack trace.

```
-->
```

```

<customErrors mode="Off" defaultRedirect="
GenericErrorPage.htm">
  <error statusCode="403" redirect="NoAccess.htm"
  />
  <error statusCode="404" redirect="FileNotFound.
  htm" />
</customErrors>
<pages>
  <controls>
    <add tagPrefix="asp" namespace="System.Web.UI"
    assembly="System.Web.Extensions, _Version
    =3.5.0.0, _Culture=neutral, _PublicKeyToken=31

```

```
        BF3856AD364E35" />
    <add tagPrefix="asp" namespace="System.Web.UI.
        WebControls" assembly="System.Web.Extensions,
        _Version=3.5.0.0, _Culture=neutral, _
        PublicKeyToken=31BF3856AD364E35" />
    <add tagPrefix="asp" namespace="System.Web.UI.
        DataVisualization.Charting" assembly="System.
        Web.DataVisualization, _Version=3.5.0.0, _
        Culture=neutral, _PublicKeyToken=31
        bf3856ad364e35" />
</controls>
</pages>
<globalization culture="it-IT" uiCulture="it" />
<httpHandlers>
    <remove path="*.asmx" verb="*" />
    <add path="*.asmx" verb="*" type="System.Web.
        Script.Services.ScriptHandlerFactory, _System.
        Web.Extensions, _Version=3.5.0.0, _Culture=
        neutral, _PublicKeyToken=31BF3856AD364E35"
        validate="false" />
    <add path="*_AppService.axd" verb="*" type="
        System.Web.Script.Services.ScriptHandlerFactory
        , _System.Web.Extensions, _Version=3.5.0.0, _
        Culture=neutral, _PublicKeyToken=31
        BF3856AD364E35" validate="false" />
    <add path="ScriptResource.axd" verb="GET,HEAD"
        type="System.Web.Handlers.ScriptResourceHandler
        , _System.Web.Extensions, _Version=3.5.0.0, _
        Culture=neutral, _PublicKeyToken=31
        BF3856AD364E35" validate="false" />
```

```

<add path="ChartImg.axd" verb="GET,HEAD" type="
  System.Web.UI.DataVisualization.Charting.
  ChartHttpHandler , _System.Web.DataVisualization ,
  _Version=3.5.0.0 , _Culture=neutral , _
  PublicKeyToken=31bf3856ad364e35" validate="
  false" />
<add type="DevExpress.Web.ASPxUploadControl.
  ASPxUploadProgressHttpHandler , _DevExpress.Web.
  v9.3 , _Version=9.3.6.0 , _Culture=neutral , _
  PublicKeyToken=b88d1754d700e49a" verb="GET"
  path="ASPxUploadProgressHandlerPage.ashx"
  validate="false" />
</httpHandlers>
<httpModules>
  <add name="ScriptModule" type="System.Web.
    Handlers.ScriptModule , _System.Web.Extensions , _
    Version=3.5.0.0 , _Culture=neutral , _
    PublicKeyToken=31BF3856AD364E35" />
  <add type="DevExpress.Web.ASPxClasses.
    ASPxHttpHandlerModule , _DevExpress.Web.v9.3 , _
    Version=9.3.6.0 , _Culture=neutral , _
    PublicKeyToken=b88d1754d700e49a" name="
    ASPxHttpHandlerModule" />
</httpModules>
</system.web>
<system.codedom>
  <compilers>
    <compiler language="c#;cs;csharp" extension=".cs"
      warningLevel="4" type="Microsoft.CSharp.
      CSharpCodeProvider , _System , _Version=2.0.0.0 , _
      Culture=neutral , _PublicKeyToken=

```

```
        b77a5c561934e089">
        <providerOption name="CompilerVersion" value="
            v3.5" />
        <providerOption name="WarnAsError" value="false
            " />
    </compiler>
    <compiler language="vb;vbs;visualbasic;vbscript"
        extension=".vb" warningLevel="4" type="
        Microsoft.VisualBasic.VBCodeProvider, System,
        Version=2.0.0.0, Culture=neutral,
        PublicKeyToken=b77a5c561934e089">
        <providerOption name="CompilerVersion" value="
            v3.5" />
        <providerOption name="OptionInfer" value="true"
            />
        <providerOption name="WarnAsError" value="false
            " />
    </compiler>
</compilers>
</system.codedom>
<!--
    The system.webServer section is required for
    running ASP.NET AJAX under Internet
    Information Services 7.0. It is not necessary
    for previous version of IIS.
-->
<system.webServer>
    <validation validateIntegratedModeConfiguration="
        false" />
    <modules>
        <remove name="ScriptModule" />
```

```
<add name="ScriptModule" preCondition="
  managedHandler" type="System.Web.Handlers.
  ScriptModule , _System.Web.Extensions , _Version
  =3.5.0.0 , _Culture=neutral , _PublicKeyToken=31
  BF3856AD364E35" />
<add type="DevExpress.Web.ASPxClasses.
  ASPxHttpHandlerModule , _DevExpress.Web.v9.3 , _
  Version=9.3.6.0 , _Culture=neutral , _
  PublicKeyToken=b88d1754d700e49a" name="
  ASPxHttpHandlerModule" />
</modules>
<handlers>
  <remove name="WebServiceHandlerFactory-Integrated
    " />
  <remove name="ScriptHandlerFactory" />
  <remove name="ScriptHandlerFactoryAppServices" />
  <remove name="ScriptResource" />
  <remove name="ChartImageHandler" />
  <add name="ScriptHandlerFactory" verb="*" path="
    *.asmx" preCondition="integratedMode" type="
    System.Web.Script.Services.ScriptHandlerFactory
    , _System.Web.Extensions , _Version=3.5.0.0 , _
    Culture=neutral , _PublicKeyToken=31
    BF3856AD364E35" />
  <add name="ScriptHandlerFactoryAppServices" verb=
    "*" path="*_AppService.axd" preCondition="
    integratedMode" type="System.Web.Script.
    Services.ScriptHandlerFactory , _System.Web.
    Extensions , _Version=3.5.0.0 , _Culture=neutral , _
    PublicKeyToken=31BF3856AD364E35" />
```



```
<add name="ScriptResource" preCondition="
  integratedMode" verb="GET,HEAD" path="
  ScriptResource.axd" type="System.Web.Handlers.
  ScriptResourceHandler , System.Web.Extensions ,
  Version = 3.5.0.0 , Culture=neutral ,
  PublicKeyToken=31BF3856AD364E35" />
<add name="ChartImageHandler" preCondition="
  integratedMode" verb="GET,HEAD" path="ChartImg.
  axd" type="System.Web.UI.DataVisualization.
  Charting.ChartHttpHandler , System.Web.
  DataVisualization , Version = 3.5.0.0 , Culture=
  neutral , PublicKeyToken=31bf3856ad364e35" />
<add type="DevExpress.Web.ASPxUploadControl.
  ASPxUploadProgressHttpHandler , DevExpress.Web.
  v9.3 , Version = 9.3.6.0 , Culture=neutral ,
  PublicKeyToken=b88d1754d700e49a" verb="GET"
  path="ASPxUploadProgressHandlerPage.ashx" name=
  "ASPxUploadProgressHandler" preCondition="
  integratedMode" />
</handlers>
<security>
  <requestFiltering>
    <requestLimits maxAllowedContentLength="
      2097151000" />
  </requestFiltering>
</security>
</system.webServer>
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:
  asm.v1">
    <dependentAssembly>
```

```

    <assemblyIdentity name="System.Web.Extensions"
      publicKeyToken="31bf3856ad364e35" />
    <bindingRedirect oldVersion="1.0.0.0-1.1.0.0"
      newVersion="3.5.0.0" />
  </dependentAssembly>
  <dependentAssembly>
    <assemblyIdentity name="System.Web.Extensions.
      Design" publicKeyToken="31bf3856ad364e35" />
    <bindingRedirect oldVersion="1.0.0.0-1.1.0.0"
      newVersion="3.5.0.0" />
  </dependentAssembly>
</assemblyBinding>
</runtime>
<system.net>
  <mailSettings>
    <smtp>
      <network host="smtp.kotik.it" userName="
        software@kotik.it" password="pippo123" />
    </smtp>
  </mailSettings>
</system.net>
</configuration>

```

Listing 4.5: MasterEmma.master

```

i>_i<%@ Master Language="C#" AutoEventWireup="true"
  CodeFile="MasterEmma.master.cs" Inherits="MasterEmma"
  %>

<%@ Register Assembly="DevExpress.Web.v9.3, Version
  =9.3.6.0, Culture=neutral, PublicKeyToken=
  b88d1754d700e49a"

```

```

    Namespace="DevExpress.Web.ASPxSiteMapControl"
    TagPrefix="dx" %>

<%@ Register Assembly="DevExpress.Web.v9.3, Version
    =9.3.6.0, Culture=neutral, PublicKeyToken=
    b88d1754d700e49a"
    Namespace="DevExpress.Web.ASPxMenu" TagPrefix="dx" %>

<%@ Register assembly="DevExpress.Web.v9.3, Version
    =9.3.6.0, Culture=neutral, PublicKeyToken=
    b88d1754d700e49a" namespace="DevExpress.Web.ASPxPanel
    " tagprefix="dx" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
    Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/
    xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Emma - questionari clinici </title>
    <link href="MasterEmma.css" type="text/css" rel="
        stylesheet" />
    <asp:ContentPlaceHolder id="HeadContent" runat="
        server">
        </asp:ContentPlaceHolder>
    <script type="text/javascript">
        function ShowGraficoPopup(idMQuest) {
            popupGrafico.SetContentUrl('FmGrafico.aspx?
                IdMQuest=' + idMQuest);
            popupGrafico.Show();
        }

```

```

    </script>
</head>
<body class="oneColLiqCtrHdr">
    <form id="form1" runat="server">
        <div id="header">
            <h3>Emma – il sito per la somministrazione di
            questionari clinici <br /></h3>
            <div class="loginstatus">
                <asp:LoginName ID="LoginName" runat="
                server" />
                <asp:LoginStatus ID="LoginStatus" runat="
                server"
                LogoutPageUrl="~/Account/Login.aspx"
                LogoutAction="RedirectToLoginPage"
                onloggedout="LoginStatus_LoggedOut" />
            </div>
            <div id="PnlMenu" runat="server">
                <dx:ASPxSiteMapDataSource ID="smpMain"
                runat="server" SiteMapProvider="
                EmmaSitemapProvider" />
                <dx:ASPxMenu ID="mnuMain" runat="server"
                Orientation="Horizontal" DataSourceID="
                smpMain">
                    </dx:ASPxMenu>
            </div> <!-- end PnlMenu -->
        </div> <!-- end .header -->
        <div id="mainContent">
            <asp:ContentPlaceHolder id="MainContent"
            runat="server">
                </asp:ContentPlaceHolder>
        </div> <!-- end .content -->
    </form>

```

```

        <div id=" footer ">
            <p><asp:Label ID=" lblUtenteProfilo" runat="
                server" Text=""></asp:Label></p>
        </div> <!-- end . footer -->
    </form>
</body>
</html>

```

Listing 4.6: MasterEmma.master.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class MasterEmma : System.Web.UI.
    MasterPage
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // TODO: impostare questo parametro in un posto piu
        // "giusto"
        // DALRuntime.globalTheme = "Glass";
        // DevExUtility.MenuApplyTheme(mnuMain, DALRuntime.
        // globalTheme);
        // Aggiorna label in footer con utente e profilo
        if (DALRuntime.RecUtente != null)
            lblUtenteProfilo.Text = DALRuntime.RecUtente.
                getDescrizione();
    }
}

```

```

protected void LoginStatus_LoggedOut(object sender ,
    EventArgs e)
{
    // Forza logout
    Session.Abandon();
    // Redirect to home
    Response.Redirect("~/Default.aspx");
}
}

```

Listing 4.7: Default.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/
    MasterEmma.master" AutoEventWireup="true" CodeFile="
    Default.aspx.cs" Inherits="_Default" %>

<%@ Register Assembly="DevExpress.Web.ASPxEditors.v9.3,
    _Version=9.3.6.0, _Culture=neutral, _PublicKeyToken=
    b88d1754d700e49a"
    Namespace="DevExpress.Web.ASPxEditors" TagPrefix="dx"
    %>

<asp:Content ID="Content1" ContentPlaceHolderID="
    HeadContent" Runat="Server">
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="
    MainContent" Runat="Server">
<div style="height:_600px;_float:_left;" class="testo">
    <div style="float:_right;">
    <asp:Image ID="imgIPad" runat="server" ImageUrl="~/
        public/Immagini/MedicoConIPad.jpg" />
    </div>

```

```
<p>
<b>Emma</b> Ã un sito dedicato alla somministrazione
  di questionari clinici.
</p>
Al momento l'unico questionario disponibile Ã il <b><i>
  >PRO-CLARA</i></b> ma ne possono essere definiti
  infiniti altri.
<br />
  Il sistema Ã fruibile da personal computer, desktop
    o portatili, e da dispositivi mobili
  di ogni tipo quali, ad esempio:
<ul>
<li>iphone, ipad</li>
<li>smartphone di altre marche </li>
<li>qualunque dispositivo in grado di navigare su web</
  li>
</ul>
<p>
<b>Emma</b> mantiene in archivio, e consente di
  interrogare, i dati dei questionari somministrati
  e dei pazienti.
<br />
Per garantire la riservatezza e la sicurezza dei dati,
  per accedere al sistema Ã necessario
  <a href="Account/Login.aspx">autenticarsi</a> fornendo
  un codice utente e una password, che possono essere
  richiesti per mail all'<a href="mailto:software@kotik
  .it">amministratore del sistema</a>.
</p>
</div>
</asp:Content>
```

Listing 4.8: Default.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}

```

Listing 4.9: About.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/
    MasterEmma.master" AutoEventWireup="true" CodeFile="
    About.aspx.cs" Inherits="About" %>

<asp:Content ID="Content1" ContentPlaceHolderID="
    HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="
    MainContent" Runat="Server">
<div class="testo">
Emma Ã sviluppato con i seguenti strumenti:
<ul>
<li>Microsoft &copy; <a href="http://www.microsoft.com/
    express/downloads/#
    Visual_Studio_2008_Express_Downloads">Visual Studio
    Web Developer 2008 Express Edition</a></li>

```



```
<li>Framework <a href="http://www.asp.net/">ASP.NET 3.5  
    sp1</a></li>  
<li>Database <a href="http://dev.mysql.com/">MySQL  
    5.1</a></li>  
<li>Librerie <a href="http://devexpress.com/Products/  
    NET/Controls/ASP/">DevExpress ASP.NET Experience e  
    ASPxGridView and editors</a></li>  
<li>Libreria <a href="http://itextpdf.com/">iTextSharp  
    </a> per la generazione di file PDF</li>  
</ul>  
</div>  
</asp:Content>
```

Listing 4.10: About.aspx.cs

```
ï»¿using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
  
public partial class About : System.Web.UI.Page  
{  
    protected void Page_Load(object sender, EventArgs e  
    )  
    {  
  
    }  
}
```

4.2.2 Gestione login

Listing 4.11: Login.aspx

```

<%@ Page Title="Log_In" Language="C#" MasterPageFile
    = "~/MasterEmma.master" AutoEventWireup="true"
    CodeFile="Login.aspx.cs" Inherits="Account_Login"
    %>

<asp:Content ID="HeaderContent" runat="server"
    ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server"
    ContentPlaceHolderID="MainContent">
    <h4>
        Accesso al sistema
    </h4>
    <p>
        Inserisca il suo codice utente e la sua
        password
    </p>

    <asp:Login ID="LoginUser" runat="server"
        EnableViewState="false">
        <LayoutTemplate>
            <span class="failureNotification">
                <asp:Literal ID="FailureText" runat="
                    server"></asp:Literal>
            </span>
            <asp:ValidationSummary ID="
                LoginUserValidationSummary" runat="server
                " CssClass="failureNotification"
                ValidationGroup="
                    LoginUserValidationGroup"/>

```

```
<div class="accountInfo">
  <fieldset class="login">
    <legend>Informazioni Account</
    legend>
    <p>
      <asp:Label ID="UserNameLabel"
        runat="server"
        AssociatedControlID="UserName
        ">Codice utente:</asp:Label>
      <asp:TextBox ID="UserName"
        runat="server" CssClass="
        textEntry"></asp:TextBox>
      <asp:RequiredFieldValidator ID=
        "UserNameRequired" runat="
        server" ControlToValidate="
        UserName"
        CssClass="
        failureNotification"
        ErrorMessage="User_Name_
        is_required." ToolTip="
        Il_codice_utente_Ã_
        obbligatorio."
        ValidationGroup="
        LoginUserValidationGroup
        ">*</asp:
        RequiredFieldValidator>
    </p>
    <p>
      <asp:Label ID="PasswordLabel"
        runat="server"
        AssociatedControlID="Password
```

```

        ">Password:</asp:Label>
<asp:TextBox ID="Password"
    runat="server" CssClass="
    passwordEntry" TextMode="
    Password"></asp:TextBox>
<asp:RequiredFieldValidator ID=
    "PasswordRequired" runat="
    server" ControlToValidate="
    Password"
        CssClass="
        failureNotification"
        ErrorMessage="Password_
        is_required." ToolTip="
        Password_is_required."
        ValidationGroup="
        LoginUserValidationGroup
    ">*</asp:
        RequiredFieldValidator>
    </p>
</fieldset>
<p class="submitButton">
    <asp:Button ID="LoginButton" runat=
        "server" CommandName="Login" Text
        ="Log_In" ValidationGroup="
        LoginUserValidationGroup"/>
    </p>
</div>
</LayoutTemplate>
</asp:Login>
</asp:Content>

```

Listing 4.12: Login.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Account_Login : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // RegisterHyperLink.NavigateUrl = "Register.
        aspx?ReturnUrl=" + HttpUtility.UrlEncode(Request.
        QueryString["ReturnUrl"]);
    }
}

```

4.2.3 Amministrazione

Listing 4.13: FmLog.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/
    MasterEmma.master" AutoEventWireup="true" CodeFile="
    FmLog.aspx.cs" Inherits="Amministrazione.FmLog" %>

<%@ Register Assembly="DevExpress.Web.ASPxGridView.v9
    .3, _Version=9.3.6.0, _Culture=neutral, _PublicKeyToken=
    b88d1754d700e49a"
    Namespace="DevExpress.Web.ASPxGridView" TagPrefix="dx
    " %>

```

```
<asp:Content ID="Content1" ContentPlaceHolderID="
  HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="
  MainContent" Runat="Server">
  <dx:ASPxGridView ID="grdLog" runat="server"
    AutoGenerateColumns="False">
    <Columns>
      <dx:GridViewDataTextColumn Caption="Operazione"
        FieldName="dd_op"
        VisibleIndex="0">
      </dx:GridViewDataTextColumn>
      <dx:GridViewDataTextColumn Caption="Messaggio"
        FieldName="dd_msg"
        VisibleIndex="1">
      </dx:GridViewDataTextColumn>
      <dx:GridViewDataTextColumn Caption="Utente"
        FieldName="dd_login"
        VisibleIndex="2">
      </dx:GridViewDataTextColumn>
      <dx:GridViewDataTextColumn Caption="Data_e_ora"
        FieldName="dt_now"
        VisibleIndex="3">
      </dx:GridViewDataTextColumn>
    </Columns>
    <Settings ShowFilterRow="True" ShowGroupPanel="True"
      " />
  </dx:ASPxGridView>
</asp:Content>
```

Listing 4.14: FmLog.aspx.cs

```
i>using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Amministrazione_FmLog : System.Web
    .UI.Page
{
    protected void grdDataBind(bool setDatasource, bool
        doDatabind)
    {
        if (setDatasource)
            grdLog.DataSource = DALRuntime.getLog();
        if (doDatabind)
            grdLog.DataBind();
    }

    protected void Page_Init(object sender, EventArgs e)
    {
        grdDataBind(true, false);
    }

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack && !IsCallback)
            grdDataBind(false, true);
    }
}
```

Listing 4.15: FmUtenti.aspx

```

i>_i<%@ Page Title="" Language="C#" MasterPageFile="" ~ /
MasterEmma.master" AutoEventWireup="true" CodeFile=""
FmUtenti.aspx.cs" Inherits="Amministrazione_FmUtenti"
%>

<%@ Register Assembly="DevExpress.Web.ASPxGridView.v9
.3, _Version=9.3.6.0, _Culture=neutral, _PublicKeyToken=
b88d1754d700e49a"
Namespace="DevExpress.Web.ASPxGridView" TagPrefix="dx
" %>

<%@ Register assembly="DevExpress.Web.ASPxEditors.v9.3,
 _Version=9.3.6.0, _Culture=neutral, _PublicKeyToken=
b88d1754d700e49a" namespace="DevExpress.Web.
ASPxEditors" tagprefix="dx" %>

<asp:Content ID="Content1" ContentPlaceHolderID=""
HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID=""
MainContent" Runat="Server">
<dx:ASPxGridView ID="grdUtenti" runat="server"
AutoGenerateColumns="False"
onhtmlrowcreated="grdUtenti_HtmlRowCreated"
KeyFieldName="id"
onrowdeleting="grdUtenti_RowDeleting"
onrowinserting="grdUtenti_RowInserting"
onrowupdating="grdUtenti_RowUpdating">
<Templates>
<EditForm>

```



```
<table>
<tr>
<td><dx:ASPxlabel ID="ASPxlabel1" runat="server"
  Text="Codice_utente"></dx:ASPxlabel></td>
<td>
  <dx:ASPxTextBox ID="edtLogin" runat="server"
    Text='<%# Bind("dd_login") %>*></dx:
    ASPxTextBox>
</td>
<td><dx:ASPxlabel ID="ASPxlabel2" runat="server"
  Text="Password"></dx:ASPxlabel></td>
<td>
  <dx:ASPxTextBox ID="edtPassword" runat="server"
    Text='<%# Bind("dd_password") %>*></dx:
    ASPxTextBox>
</td>
</tr>
<tr>
<td><dx:ASPxlabel ID="ASPxlabel3" runat="server"
  Text="Nome"></dx:ASPxlabel></td>
<td>
  <dx:ASPxTextBox ID="edtNome" runat="server"
    Text='<%# Bind("dd_nome") %>*></dx:
    ASPxTextBox>
</td>
<td><dx:ASPxlabel ID="ASPxlabel4" runat="server"
  Text="Cognome"></dx:ASPxlabel></td>
<td>
  <dx:ASPxTextBox ID="edtCognome" runat="server"
    Text='<%# Bind("dd_cognome") %>*></dx:
    ASPxTextBox>
```

```

</td>
</tr>
<tr>
<td><dx:ASPxlabel ID="ASPxlabel5" runat="server"
  Text="Indirizzo_email"></dx:ASPxlabel></td>
<td>
  <dx:ASPxTextBox ID="edtMail" runat="server"
    Text='<%# Bind("dd_mail") %>'\></dx:
    ASPxTextBox>
</td>
<td><dx:ASPxlabel ID="ASPxlabel6" runat="server"
  Text="Profilo"></dx:ASPxlabel></td>
<td>
  <dx:ASPxComboBox ID="cbxProfilo" runat="server"
    TextField="dd_descrizione"
    ValueField="id" ValueType="System.Int32"
    Text='<%# Bind("cz_profilo") %>'\>
  </dx:ASPxComboBox>
</td>
</tr>
</table>
<div style="text-align:right;_padding:2px_2px_2px_
2px">
  <dx:ASPxGridViewTemplateReplacement ID="
  UpdateButton" ReplacementType="
  EditFormUpdateButton" runat="server"></dx:
  ASPxGridViewTemplateReplacement>
  <dx:ASPxGridViewTemplateReplacement ID="
  CancelButton" ReplacementType="
  EditFormCancelButton" runat="server"></dx:
  ASPxGridViewTemplateReplacement>

```

```
</div>
</EditForm>
</Templates>
<SettingsBehavior ConfirmDelete="True" />
<SettingsEditing Mode="EditFormAndDisplayRow" />
<SettingsText ConfirmDelete=" Confermi la
cancellazione?" />
<Columns>
  <dx:GridViewCommandColumn VisibleIndex="0">
    <EditButton Visible="True">
      </EditButton>
    <NewButton Visible="True">
      </NewButton>
    <DeleteButton Visible="True">
      </DeleteButton>
    </dx:GridViewCommandColumn>
  <dx:GridViewDataTextColumn Caption="Cognome"
    FieldName="dd_cognome"
    VisibleIndex="1">
    </dx:GridViewDataTextColumn>
  <dx:GridViewDataTextColumn Caption="Nome"
    FieldName="dd_nome" VisibleIndex="2">
    </dx:GridViewDataTextColumn>
  <dx:GridViewDataTextColumn Caption="Codice_utente
    " FieldName="dd_login"
    VisibleIndex="3">
    </dx:GridViewDataTextColumn>
  <dx:GridViewDataTextColumn Caption="Indirizzo_di_
    posta" FieldName="dd_mail"
    VisibleIndex="4">
    </dx:GridViewDataTextColumn>
```

```

<dx:GridViewDataTextColumn Caption=" Profilo "
    FileName=" dd_profilo "
    ToolTip=" Profilo " VisibleIndex=" 5">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption=" Password "
    FileName=" dd_password "
    Visible=" False " VisibleIndex=" 6">
</dx:GridViewDataTextColumn>
</Columns>
</dx:ASPxGridView>
</asp:Content>

```

Listing 4.16: FmUtenti.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using DevExpress.Web.ASPxEditors;
using DevExpress.Web.ASPxGridView;
using DevExpress.Web.Data;

public partial class Amministrazione_FmUtenti : System.
    Web.UI.Page
{
    protected void grdDataBind(bool setDatasource, bool
        doDatabind)
    {
        if (setDatasource)
        {
            DALRuntime.tblUtenti = DALRuntime.getUtenti();

```

```
        grdUtenti.DataSource = DALRuntime.tblUtenti;
    }
    if (doDatabind)
        grdUtenti.DataBind();
}

protected void Page_Init(object sender, EventArgs e)
{
    grdDataBind(true, false);
}

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack && !IsCallback)
        grdDataBind(false, true);
}

protected void grdUtenti_HtmlRowCreated(object sender
, ASPxGridViewTableRowEventArgs e)
{
    if (e.RowType == DevExpress.Web.ASPxGridView.
        GridViewRowType.EditForm)
    {
        ASPxGridView grdSender = (sender as ASPxGridView)
            ;
        System.Data.DataRow r = DALRuntime.tblUtenti.Rows
            .Find(e.KeyValue);
        // Data bind del combo del profilo
        ASPxComboBox cbxProfilo = (ASPxComboBox)grdSender
            .FindEditFormTemplateControl("cbxProfilo");
        if (cbxProfilo != null)
```

```
{
    cbxProfilo.DataSource = DALRuntime.getProfili()
    ;
    cbxProfilo.DataBindItems();
    if (r != null)
        cbxProfilo.Value = r["cz_profilo"];
}
}
```

```
protected void grdUtenti_RowDeleting(object sender,
    ASPxDataDeletingEventArgs e)
{
    int id = Convert.ToInt32(e.Keys[0].ToString());
    DALRuntime.deleteUtente(id);
    e.Cancel = true;
    grdDataBind(true, true);
}
```

```
protected void grdUtenti_RowInserting(object sender,
    ASPxDataInsertingEventArgs e)
{
    string ddLogin = e.NewValues["dd_login"].ToString()
    ;
    string ddNome = e.NewValues["dd_nome"].ToString();
    string ddCognome = e.NewValues["dd_cognome"].
    ToString();
    string ddPassword = e.NewValues["dd_password"].
    ToString();
    int czProfilo = getCzProfilo();
    string ddMail = e.NewValues["dd_mail"].ToString();
```

```
DALRuntime.insertUtente(ddLogin, ddPassword, ddNome
    , ddCognome, ddMail, czProfilo);
e.Cancel = true;
grdUtenti.CancelEdit();
grdDataBind(true, true);
}

private int getCzProfilo()
{
    ASPxComboBox cbxProfilo = (ASPxComboBox)grdUtenti.
        FindEditFormTemplateControl("cbxProfilo");
    int czProfilo = 0;
    if (cbxProfilo != null && cbxProfilo.Value != null)
        czProfilo = Convert.ToInt32(cbxProfilo.Value.
            ToString());
    return czProfilo;
}

protected void grdUtenti_RowUpdating(object sender,
    ASPxDataUpdatingEventArgs e)
{
    int id = Convert.ToInt32(e.Keys[0].ToString());
    string ddLogin = e.NewValues["dd_login"].ToString();
    ;
    string ddNome = e.NewValues["dd_nome"].ToString();
    string ddCognome = e.NewValues["dd_cognome"].
        ToString();
    string ddPassword = e.NewValues["dd_password"].
        ToString();
    int czProfilo = getCzProfilo();
    string ddMail = e.NewValues["dd_mail"].ToString();
```

```

        DALRuntime.updateUtente(id, ddLogin, ddPassword,
            ddNome, ddCognome, ddMail, czProfilo);
        e.Cancel = true;
        grdUtenti.CancelEdit();
        grdDataBind(true, true);
    }
}

```

4.2.4 Classi di sistema

Listing 4.17: DALRuntime.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.SessionState;
using MySql.Data.MySqlClient;
using System.Data;
using System.Configuration;

/// <summary>
/// Summary description for DALRuntime
/// </summary>
public class DALRuntime
{
    #region COSTANTI
    public static string fImageNotPresent = "
        ImmagineNonDisponibile.jpg";
    public static string fImagePazientiPath = "~/public/
        Immagini/Pazienti/";
    public static string fImageQuestionariPath = "~/
        public/Immagini/Questionari/";

```



```
#endregion COSTANTI

#region PRIVATE
// numero domande
private const string spNumDomande = "spNumDomande";
// global theme
private const string spGlobalTheme = "globalTheme";
// client id txtLettera
private const string spLetteraClientId = "
    clientIdLettera";
// id profilo
private const string spIdProfilo = "idProfilo";
// id utente
private const string spIdUtente = "idUtente";
// tengo in sessione il numero della domanda corrente
    usato in FmCompila
private const string spIdDomanda = "idDomanda";
// tengo in sessione anche il dataset delle domande
private const string spTblDomande = "tblDomande";
// pazienti
private const string spTblPazienti = "tblPazienti";
// tabella utenti
private const string spTblUtenti = "tblUtenti";
// tblAQuestItem
private const string spTblAQuestItem = "tblAQuestItem
    ";
// Salvo in sessione anche la connessione al db
private const string spMainConnection = "
    MainConnection";
// id m_quest
private const string spIdMQuest = "idMQuest";
```

```
// id a_quest
private const string spIdAQuest = "idAQuest";
// id a_quest_sezione
private const string spIdAQuestSezione = "
    idAQuestSezione";
// rec paziente
private const string spRecPaziente = "recPaziente";
// rec utente
private const string spRecUtente = "recUtente";
// rec mquest
private const string spRecMQuest = "recMQuest";
#endregion PRIVATE

#region CONSTRUCTOR
public DALRuntime()
    {
        //
        // TODO: Add constructor logic here
        //
    }
#endregion CONSTRUCTOR

#region DB
///<summary>
///Ritorna la connessione principale al db
///</summary>
///<param name="connString"></param>
///<returns></returns>
public static MySqlConnection getConnection(string
    connString)
{
```

```
// Quando richiamato da SitemapProvider
if (HttpSession == null)
{
    MySqlConnection c = new MySqlConnection(
        connString != string.Empty ? connString :
        getConnectionString());
    c.Open();
    return c;
}
else
{
    if (MainConnection == null)
        MainConnection = new MySqlConnection(connString
            != string.Empty ? connString :
            getConnectionString());
    if (MainConnection.State == ConnectionState.
        Closed)
        MainConnection.Open();
    return MainConnection;
}
}

public static MySqlConnection getConnection()
{
    return getConnection(string.Empty);
}

public static MySqlCommand buildCommand(string sSQL)
{
    MySqlConnection conn = getConnection(
        getConnectionString());
```

```
        MySqlCommand cmd = new MySqlCommand(sSQL, conn);
        return cmd;
    }

    public static string getConnectionString()
    {
        string fConnString = ConfigurationManager.
            ConnectionStrings["MySQLEmma"].ToString();
        return fConnString;
    }
#endregion DB

#region SESSIONE
    private static HttpSessionState HttpSession
    {
        get { return HttpContext.Current.Session; }
    }

    public static MySqlConnection MainConnection
    {
        get { if (HttpSession[spMainConnection] == null)
            return null; else return (MySqlConnection)(
                HttpSession[spMainConnection]); }
        set { HttpSession[spMainConnection] = value; }
    }

    public static TRecPaziente RecPaziente
    {
        get { if (HttpSession[spRecPaziente] == null)
            return null; else return (TRecPaziente)(
                HttpSession[spRecPaziente]); }
    }
}
```

```
    set { HttpSession[spRecPaziente] = value; }
}

public static TRecUtente RecUtente
{
    get { if (HttpSession[spRecUtente] == null) return
        null; else return (TRecUtente)(HttpSession[
        spRecUtente]); }
    set { HttpSession[spRecUtente] = value; }
}

public static TRecMQuest RecMQuest
{
    get { if (HttpSession[spRecMQuest] == null) return
        null; else return (TRecMQuest)(HttpSession[
        spRecMQuest]); }
    set { HttpSession[spRecMQuest] = value; }
}

public static int IdProfilo
{
    get { if (HttpSession[spIdProfilo] == null) return
        1; else return (int)(HttpSession[spIdProfilo]); }
    set { HttpSession[spIdProfilo] = value; }
}

public static int IdUtente
{
    get { if (HttpSession[spIdUtente] == null) return
        1; else return (int)(HttpSession[spIdUtente]); }
    set { HttpSession[spIdUtente] = value; }
```

```
}
```

```
public static int IdAQuest
{
    get { if (HttpSession[spIdAQuest] == null) return
        1; else return (int)(HttpSession[spIdAQuest]); }
    set { HttpSession[spIdAQuest] = value; }
}
```

```
public static int IdAQuestSezione
{
    get { if (HttpSession[spIdAQuestSezione] == null)
        return 1; else return (int)(HttpSession[
        spIdAQuestSezione]); }
    set { HttpSession[spIdAQuestSezione] = value; }
}
```

```
public static int IdMQuest
{
    get { if (HttpSession[spIdMQuest] == null) return
        1; else return (int)(HttpSession[spIdMQuest]); }
    set { HttpSession[spIdMQuest] = value; }
}
```

```
public static int iDomanda
{
    get { if (HttpSession[spIdDomanda] == null) return
        0; else return (int)(HttpSession[spIdDomanda]); }
    set { HttpSession[spIdDomanda] = value; }
}
```

```
public static int iNumDomande
{
    get { if (HttpSession[spNumDomande] == null) return
        0; else return (int)(HttpSession[spNumDomande]); }
    set { HttpSession[spNumDomande] = value; }
}
```

```
public static bool isLastDomanda
{
    get { return iDomanda == tblDomande.Rows.Count - 1; }
}
```

```
public static DataTable tblDomande
{
    get { return (DataTable)(HttpSession[spTblDomande])
        ; }
    set { HttpSession[spTblDomande] = value; }
}
```

```
public static DataTable tblUtenti
{
    get { return (DataTable)(HttpSession[spTblUtenti]); }
    set { HttpSession[spTblUtenti] = value; }
}
```

```
public static DataTable tblPazienti
{
```

```
    get { return (DataTable)(HttpSession[spTblPazienti
        ])); }
    set { HttpSession[spTblPazienti] = value; }
}

public static DataTable tblAQuestItems
{
    get { return (DataTable)(HttpSession[
        spTblAQuestItem]); }
    set { HttpSession[spTblAQuestItem] = value; }
}

public static string letteraClientId
{
    get { return HttpSession[spLetteraClientId] == null
        ? string.Empty : HttpSession[spLetteraClientId].
        ToString(); }
    set { HttpSession[spLetteraClientId] = value; }
}

public static string globalTheme
{
    get { return HttpSession[spGlobalTheme] == null ?
        string.Empty : HttpSession[spGlobalTheme].
        ToString(); }
    set { HttpSession[spGlobalTheme] = value; }
}
#endregion SESSIONE

#region LETTERA
///<summary>
```



```
/// Esegue CalcolaLettera (IdMQuest, IdUtente)
/// </summary>
/// <param name="cmQuest"></param>
/// <param name="idUtente"></param>
public static void calcolaLettera(int cmQuest, int
    idUtente)
{
    MySqlCommand cmd = buildCommand(
        @"call _CalcolaLettera (@cmQuest, _@idUtente)");
    cmd.Parameters.AddWithValue("@cmQuest", cmQuest);
    cmd.Parameters.AddWithValue("@idUtente", idUtente);
    cmd.ExecuteNonQuery();
} /* calcolaLettera */

/// <summary>
/// Salva sLettera nel questionario idMQuest
/// </summary>
/// <param name="idMQuest"></param>
/// <param name="sLettera"></param>
public static void saveLettera(int idMQuest, string
    sLettera)
{
    MySqlCommand cmd = buildCommand("call _setLettera (
        @idMQuest, _@ddLettera);");
    cmd.Parameters.AddWithValue("@idMQuest", idMQuest);
    cmd.Parameters.AddWithValue("@ddLettera", sLettera)
        ;
    cmd.ExecuteNonQuery();
}

/// <summary>
```

```
/// Ritorna la testata associata all'utente in questo
    momento in sessione
/// </summary>
/// <returns></returns>
public static string getTestata()
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand("select _
        getTestata(@idUtente)", conn);
    cmd.Parameters.AddWithValue("@idUtente", IdUtente);

    object result = cmd.ExecuteScalar();
    if (result != null)
        return result.ToString();
    else
        return string.Empty;
} /* getTestata */

public static void setTestata(int idUtente, string
    sTestata)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand("call _
        setTestata(@idUtente, _@blTesto)", conn);
    cmd.Parameters.AddWithValue("@idUtente", IdUtente);
    cmd.Parameters.AddWithValue("@blTesto", sTestata);
    cmd.ExecuteNonQuery();
} /* setTestata */

/// <summary>
/// Ritorna la lettera del quest idMQuest
```

```
/// </summary>
/// <param name="idMQuest"></param>
/// <returns></returns>
public static string getLettera(int idMQuest)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand("select _
        getLettera(@idMQuest)", conn);
    cmd.Parameters.AddWithValue("@idMQuest", idMQuest);

    object result = cmd.ExecuteScalar();
    if (result != null)
        return result.ToString();
    else
        return string.Empty;
}

public static string getNomePazienteConSaluto(int
    idMQuest)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand("select _
        getNomePazienteConSaluto(@idMQuest)", conn);
    cmd.Parameters.AddWithValue("@idMQuest", idMQuest);

    object result = cmd.ExecuteScalar();
    if (result != null)
        return result.ToString();
    else
        return string.Empty;
} /* getNomePazienteConSaluto */
```

```

/// <summary>
/// Ritorna l'indirizzo mail del medico del paziente
/// associato a idMQuest
/// </summary>
/// <param name="idMQuest"></param>
/// <returns></returns>
public static string getMailAddressMedico(int
    idMQuest)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand("select _
        getMailAddressMedico(@idMQuest)", conn);
    cmd.Parameters.AddWithValue("@idMQuest", idMQuest);

    object result = cmd.ExecuteScalar();
    if (result != null)
        return result.ToString();
    else
        return string.Empty;
} /* getMailAddressMedico */

public static DataTable getMQuest()
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"select _IdPaziente , _IdMQuest , _IdAQuest , _
            Questionario , _Data , _Cognome , _Nome , _Risultato ,
            .....nr_risultato1 , _nr_risultato2 , _
            nr_risultato3 ,

```

```

....._dd_medico_cognome_as_dd_medico_famiglia ,
    _dd_medico_email ,
....._dd_utente_cognome_as_dd_utente
....._from_v_m_quest
....._order_by_Cognome" , conn);
    MySqlConnection adapter = new MySqlConnection(cmd
    );
    DataSet dataset = new DataSet();
    adapter.Fill(dataset);
    conn.Close();
    return dataset.Tables[0];
}
#endregion LETTERA

#region TABELLE
public static DataTable getTitoli()
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"select_id,_dd_descrizione_from_t_titolo" , conn)
        ;
    MySqlConnection adapter = new MySqlConnection(cmd
    );
    DataSet dataset = new DataSet();
    adapter.Fill(dataset);
    conn.Close();
    return dataset.Tables[0];
}

public static void updateTitolo(int id , string sDescr
)

```

```

    {
        MySqlConnection conn = getConnection();
        MySqlCommand cmd = new MySqlCommand(
            @"update_t_titolo
            .....set_dd_descrizione=@ddDescr
            .....where_id=@id", conn);
        cmd.Parameters.AddWithValue("@ddDescr", sDescr);
        cmd.Parameters.AddWithValue("@id", id);
        cmd.ExecuteNonQuery();
    } /* updateTitolo */

    public static void deleteTitolo(int id)
    {
        MySqlConnection conn = getConnection();
        MySqlCommand cmd = new MySqlCommand(
            @"delete_from_t_titolo
            .....where_id=@id", conn);
        cmd.Parameters.AddWithValue("@id", id);
        cmd.ExecuteNonQuery();
    } /* deleteTitolo */

    public static int insertTitolo(string sDescr)
    {
        MySqlConnection conn = getConnection();
        MySqlCommand cmd = new MySqlCommand(
            @"insert_into_t_titolo(dd_descrizione)
            .....values(@ddDescr)", conn);
        cmd.Parameters.AddWithValue("@ddDescr", sDescr);
        cmd.ExecuteNonQuery();
        MySqlCommand cmdLastId = new MySqlCommand("select _
            last_insert_id()", conn);
    }

```

```

        cmdLastId.ExecuteScalar();
        object result = cmdLastId.ExecuteScalar();
        if (result != null)
            return Convert.ToInt32(result.ToString());
        else
            return 0;
    } /* insertTitolo */

public static DataTable getStatoCivile()
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"select _id, _dd_descrizione _from _t_stato_civile",
        conn);
    MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
    );
    DataSet dataset = new DataSet();
    adapter.Fill(dataset);
    conn.Close();
    return dataset.Tables[0];
}

public static void deleteStatoCivile(int id)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"delete _from _t_stato_civile
        _____where _id=@id", conn);
    cmd.Parameters.AddWithValue("@id", id);
    cmd.ExecuteNonQuery();
} /* deleteStatoCivile */

```

```

public static void updateStatoCivile(int id, string
    sDescr)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"update_t_stato_civile
        .....set_dd_descrizione=@ddDescr
        .....where_id=@id", conn);
    cmd.Parameters.AddWithValue("@ddDescr", sDescr);
    cmd.Parameters.AddWithValue("@id", id);
    cmd.ExecuteNonQuery();
} /* updateStatoCivile */

public static int insertStatoCivile(string sDescr)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"insert_into_t_stato_civile(dd_descrizione)
        .....values_(@ddDescr)", conn);
    cmd.Parameters.AddWithValue("@ddDescr", sDescr);
    cmd.ExecuteNonQuery();
    MySqlCommand cmdLastId = new MySqlCommand("select _
        last_insert_id()", conn);
    cmdLastId.ExecuteScalar();
    object result = cmdLastId.ExecuteScalar();
    if (result != null)
        return Convert.ToInt32(result.ToString());
    else
        return 0;
} /* insertStatoCivile */

```



```

#endregion TABELLE

#region QUESTIONARI
public static void setRecMQuest(int idMQuest)
{
    if (DALRuntime.RecMQuest == null)
        DALRuntime.RecMQuest = new DALRuntime.TRecMQuest
            ();
    DALRuntime.RecMQuest.readMQuest(idMQuest);
}

public class TRecMQuest
{
    public int IdMQuest { get; set; }
    public string DdRisultato { get; set; }
    public int CaQuest { get; set; }
    public int CaPaziente { get; set; }
    public string DdQuestionario { get; set; }
    public DateTime Data { get; set; }

    public void readMQuest(int idMQuest)
    {
        this.IdMQuest = idMQuest;
        MySqlCommand cmd = buildCommand(
            @"select _IdPaziente, _IdAQuest, _Questionario, _Data,
                _Risultato
            from _v_m_quest
            where _IdMQuest = @idMQuest");
        cmd.Parameters.AddWithValue("@idMQuest", idMQuest
            );
        MySqlDataReader r = cmd.ExecuteReader();
    }
}

```

```
    if (r.Read())
    {
        this.DdRisultato = r["Risultato"].ToString();
        this.DdQuestionario = r["Questionario"].
            ToString();
        this.CaQuest = Convert.ToInt32(r["IdAQuest"].
            ToString());
        this.CaPaziente = Convert.ToInt32(r["IdPaziente
            "].ToString());
        this.Data = Convert.ToDateTime(r["Data"].
            ToString());
    }
    else
    {
        this.DdRisultato = string.Empty;
        this.DdQuestionario = string.Empty;
        this.CaQuest = 0;
        this.CaPaziente = 0;
        // TODO: cosa faccio con la data ??
    }
    r.Close();
}

/// <summary>
/// Inserisce un nuovo record in M_QUEST e ritorna l'
 id generato
/// </summary>
/// <param name="caQuest"></param>
/// <param name="caPaziente"></param>
/// <returns></returns>
```

```
public static int insertMQuest(int caQuest, int
    caPaziente, int czUtente)
{
    MySqlCommand cmdInsert = buildCommand(
        "select _insertMQuest(@caQuest, _@caPaziente, _
            @czUtente, _now())");
    cmdInsert.Parameters.AddWithValue("@caQuest",
        caQuest);
    cmdInsert.Parameters.AddWithValue("@caPaziente",
        caPaziente);
    cmdInsert.Parameters.AddWithValue("@czUtente",
        czUtente);
    cmdInsert.ExecuteScalar();
    object result = cmdInsert.ExecuteScalar();
    if (result != null)
        return Convert.ToInt32(result.ToString());
    else
        return 0;
}

public static void insertMQuestItem(int cmQuest, int
    caQuestItem, int iNrScelta)
{
    MySqlCommand cmdInsert = buildCommand(
        @"insert _into _m_quest_item (cm_quest, _
            ca_quest_item, _nr_scelta)
        values _(@cmQuest, _@caQuestItem, _@nrScelta)");
    cmdInsert.Parameters.AddWithValue("@cmQuest",
        cmQuest);
    cmdInsert.Parameters.AddWithValue("@caQuestItem",
        caQuestItem);
```

```

        cmdInsert.Parameters.AddWithValue("@nrScelta",
            iNrScelta);
        cmdInsert.ExecuteNonQuery();
    }

    /// <summary>
    /// Richiama CalcolaRisultato
    /// </summary>
    /// <param name="cmQuest"></param>
    /// <returns></returns>
    public static string calcolaRisultato(int cmQuest)
    {
        MySqlCommand cmd = buildCommand(
            @"select _CalcolaRisultato(@cmQuest)");
        cmd.Parameters.AddWithValue("@cmQuest", cmQuest);
        object result = cmd.ExecuteScalar();
        // TODO: chiudere la connessione ?
        if (result != null)
            return result.ToString();
        else
            return string.Empty;
    } /* calcolaRisultato */

    public static DataTable getQuestionari()
    {
        MySqlConnection conn = getConnection(
            getConnectionString());
        MySqlCommand cmd = new MySqlCommand("select *_from_
            v_a_quest", conn);
        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
            );
    }

```

```

        DataSet dataset = new DataSet();
        adapter.Fill(dataset);
        conn.Close();
        return dataset.Tables[0];
    }

    public static DataTable getMQuestItem(int CmQuest)
    {
        MySqlConnection conn = getConnection();
        MySqlCommand cmd = new MySqlCommand(
            @"select _IdMQuestItem , _IdAQuestItem , _CmQuest , _
                dd_domanda_Domanda ,
                dd_risposte_Risposte , _nr_scelta_Scelta , _
                nr_order
            from _v_m_quest_item
            where _CmQuest=_@CmQuest" , conn);
        cmd.Parameters.AddWithValue("@CmQuest" , CmQuest);
        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
            );
        DataSet dataset = new DataSet();
        adapter.Fill(dataset);
        conn.Close();
        return dataset.Tables[0];
    }

    public class TRecQuestItem
    {
        public int idAQuestItem { get; set; }
        public int caQuestSezione { get; set; }
        public int caQuest { get; set; }
        public int nrOrder { get; set; }
    }

```

```

public string ddDomanda { get; set; }
public string ddRisposte { get; set; }
public string ddImmagine { get; set; }

public void init(int idAQuestItem, int caQuest, int
    nrOrder,
    string ddDomanda, string ddRisposte, string
    ddImmagine, int caQuestSezione)
{
    this.idAQuestItem = idAQuestItem;
    this.caQuest = caQuest;
    this.nrOrder = nrOrder;
    this.ddDomanda = ddDomanda;
    this.ddRisposte = ddRisposte;
    this.ddImmagine = ddImmagine;
    this.caQuestSezione = caQuestSezione;
}

public void updateAQuestItem()
{
    MySqlCommand cmd = buildCommand(
        @"update _a_quest_item
        .....set _dd_domanda =_@ddDomanda, _dd_risposte =_
        _@ddRisposte ,
        .....dd_immagine =_@ddImmagine, _nr_order =_
        @nrOrder
        .....where _id =_@idAQuestItem");
    cmd.Parameters.AddWithValue("@ddDomanda", this.
        ddDomanda);
    cmd.Parameters.AddWithValue("@ddRisposte", this.
        ddRisposte);
}

```

```

cmd.Parameters.AddWithValue("@ddImmagine", this.
    ddImmagine);
cmd.Parameters.AddWithValue("@nrOrder", this.
    nrOrder);
cmd.Parameters.AddWithValue("@idAQuestItem", this
    .idAQuestItem);
cmd.ExecuteNonQuery();
} /* updateAQuestItem */

public void insertAQuestItem()
{
    MySqlCommand cmd = buildCommand(
        @"insert into a_quest_item (ca_quest, dd_domanda,
            dd_risposte, dd_immagine, nr_order,
            ca_quest_sezione)
.....values (@caQuest, @ddDomanda, @ddRisposte
, @ddImmagine, @nrOrder, @caQuestSezione)");
    cmd.Parameters.AddWithValue("@ddDomanda", this.
        ddDomanda);
    cmd.Parameters.AddWithValue("@ddRisposte", this.
        ddRisposte);
    cmd.Parameters.AddWithValue("@ddImmagine", this.
        ddImmagine);
    cmd.Parameters.AddWithValue("@nrOrder", this.
        nrOrder);
    cmd.Parameters.AddWithValue("@caQuest", this.
        caQuest);
    cmd.Parameters.AddWithValue("@caQuestSezione",
        this.caQuestSezione);
    cmd.ExecuteNonQuery();
} /* insertAQuestItem */

```

```

    /// <summary>
    /// Cancella da a_quest_item il record con id=
    /// idQuestItem
    /// </summary>
    /// <param name="idQuestItem"></param>
    public void deleteAQuestItem(int idQuestItem)
    {
        MySqlCommand cmd = buildCommand(
            @"delete _from_ a_quest_item
            .....where _id=@id");
        cmd.Parameters.AddWithValue("@id", idQuestItem);
        cmd.ExecuteNonQuery();
    } /* deleteAQuestItem */

}

    /// <summary>
    /// Ritorna tutti i questionari definiti
    /// </summary>
    /// <returns></returns>
    public static DataTable getAQuest()
    {
        MySqlConnection conn = getConnection();
        MySqlCommand cmd = new MySqlCommand(
            @"select _id, _dd_descrizione, _dd_descr_breve _from_
            a_quest", conn);
        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
        );
        DataSet dataset = new DataSet();
        adapter.Fill(dataset);
    }

```



```

        conn.Close();
        return dataset.Tables[0];
    }

    public static void updateAQuest(int id, string sDescr
        , string sDescrBreve)
    {
        MySqlConnection conn = getConnection();
        MySqlCommand cmd = new MySqlCommand(
            @"update_a_quest
            .....set_dd_descrizione=@ddDescr,
            .....dd_descr_breve=@ddDescrBreve
            .....where_id=@id", conn);
        cmd.Parameters.AddWithValue("@ddDescr", sDescr);
        cmd.Parameters.AddWithValue("@ddDescrBreve",
            sDescrBreve);
        cmd.Parameters.AddWithValue("@id", id);
        cmd.ExecuteNonQuery();
    } /* updateAQuest */

    public static int insertAQuest(string sDescr, string
        sDescrBreve)
    {
        MySqlConnection conn = getConnection();
        MySqlCommand cmd = new MySqlCommand(
            @"insert_into_a_quest(dd_descrizione ,
            dd_descr_breve)
            .....values_(@ddDescr ,
            @ddDescrBreve)", conn);
        cmd.Parameters.AddWithValue("@ddDescr", sDescr);

```

```

cmd.Parameters.AddWithValue("@ddDescrBreve",
    sDescrBreve);
cmd.ExecuteNonQuery();
MySQLCommand cmdLastId = new MySQLCommand("select _
    last_insert_id()", conn);
cmdLastId.ExecuteScalar();
object result = cmdLastId.ExecuteScalar();
if (result != null)
    return Convert.ToInt32(result.ToString());
else
    return 0;
} /* insertAQuest */

/// <summary>
/// Cancellata da a_quest il record con id=idQuest
/// </summary>
/// <param name="idQuest"></param>
public static void deleteAQuest(int idQuest)
{
    MySqlConnection conn = getConnection();
    MySQLCommand cmd = new MySQLCommand(
        @"delete from a_quest
        .....where id=@id", conn);
    cmd.Parameters.AddWithValue("@id", idQuest);
    cmd.ExecuteNonQuery();
} /* deleteAQuest */

/// <summary>
/// Ritorna le sezioni di DALRuntime.idAQuest
/// </summary>
/// <returns>Datatable delle sezioni</returns>

```

```

public static DataTable getAQuestSezione()
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"select id, dd_descrizione1, dd_descrizione2,
            nr_num_scelte_possibili
            from a_quest_sezione
            where ca_quest = @idAQuest", conn);
    cmd.Parameters.AddWithValue("@idAQuest", DALRuntime
        .IdAQuest);
    MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
        );
    DataSet dataset = new DataSet();
    adapter.Fill(dataset);
    conn.Close();
    // Forza id come pk
    dataset.Tables[0].PrimaryKey = new DataColumn[1] {
        dataset.Tables[0].Columns["id"] };
    return dataset.Tables[0];
} /* getAQuestSezione */

public static void updateAQuestSezione(int id, string
    sDescr1, string sDescr2, int caQuest, int
    nrNumSceltePossibili)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"update a_quest_sezione
            set dd_descrizione1 = @ddDescr1,
            dd_descrizione2 = @ddDescr2,
            ca_quest = @caQuest,

```

```

.....nr_num_scelte_possibili =
    @nrNumSceltePossibili
.....where_id=@id", conn);
    cmd.Parameters.AddWithValue("@ddDescr1", sDescr1);
    cmd.Parameters.AddWithValue("@ddDescr2", sDescr2);
    cmd.Parameters.AddWithValue("@caQuest", caQuest);
    cmd.Parameters.AddWithValue("@nrNumSceltePossibili"
        , nrNumSceltePossibili);
    cmd.Parameters.AddWithValue("@id", id);
    cmd.ExecuteNonQuery();
} /* updateAQuestSezione */

public static int insertAQuestSezione(string sDescr1,
    string sDescr2, int caQuest, int
    nrNumSceltePossibili)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"insert into a_quest_sezione (dd_descrizione1 ,
            dd_descrizione2 , ca_quest ,
            nr_num_scelte_possibili)
.....values (@ddDescr1 , @ddDescr1
, @caQuest , @nrNumSceltePossibili)", conn);
    cmd.Parameters.AddWithValue("@ddDescr1", sDescr1);
    cmd.Parameters.AddWithValue("@ddDescr2", sDescr2);
    cmd.Parameters.AddWithValue("@caQuest", caQuest);
    cmd.Parameters.AddWithValue("@nrNumSceltePossibili"
        , nrNumSceltePossibili);
    cmd.ExecuteNonQuery();
    MySqlCommand cmdLastId = new MySqlCommand("select
        last_insert_id()", conn);

```

```

    cmdLastId.ExecuteScalar();
    object result = cmdLastId.ExecuteScalar();
    if (result != null)
        return Convert.ToInt32(result.ToString());
    else
        return 0;
} /* insertAQuestSezione */

///summary
/// Cancella da a_quest_sezione il record con id=
    idQuestSezione
/// /summary
/// <param name="idQuestSezione"></param>
public static void deleteAQuestSezione(int
    idQuestSezione)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"delete from a_quest_sezione
-----where id=@id", conn);
    cmd.Parameters.AddWithValue("@id", idQuestSezione);
    cmd.ExecuteNonQuery();
} /* deleteAQuestSezione */

///summary
/// Ritorna il record di a_quest_item con id =
    idAQuestItem
/// /summary
/// <param name="idAQuestItem"></param>
/// <returns></returns>

```

```

public static DataTable getAQuestItem(int
    idAQuestItem)
{
    MySqlConnection conn = getConnection(
        getConnectionString());
    MySqlCommand cmd = new MySqlCommand(
        @"select_*
.....from_a_quest_item
.....where_id_=_@idAQuestItem", conn);
    cmd.Parameters.AddWithValue("@idAQuestItem",
        idAQuestItem);
    MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
    );
    DataSet dataset = new DataSet();
    adapter.Fill(dataset);
    conn.Close();
    return dataset.Tables[0];
} /* getAQuestItem */

/// <summary>
/// Ritorna a_quest_item per idAQuest e
 idAQuestSezione
/// </summary>
/// <param name="idAQuest"></param>
/// <returns></returns>
public static DataTable getAQuestItemsSezione(int
    idAQuest, int idAQuestSezione)
{
    MySqlConnection conn = getConnection(
        getConnectionString());
    MySqlCommand cmd = new MySqlCommand(

```

```

        @" select_*
        .....from_a_quest_item
        .....where_ca_quest_=_@caQuest
        .....and_ca_quest_sezione_=_@caQuestSezione" ,
    conn);
    cmd.Parameters.AddWithValue("@caQuest", idAQuest);
    cmd.Parameters.AddWithValue("@caQuestSezione",
        idAQuestSezione);
    MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
    );
    DataSet dataset = new DataSet();
    adapter.Fill(dataset);
    conn.Close();
    // Forza id come pk
    dataset.Tables[0].PrimaryKey = new DataColumn[1] {
        dataset.Tables[0].Columns["id"] };
    return dataset.Tables[0];
} /* getAQuestItemsSezione */

/// <summary>
/// Ritorna le domande del questionario idQuest
/// </summary>
/// <param name="idQuest"></param>
/// <returns></returns>
public static DataTable getQuestItems(int idQuest)
{
    MySqlConnection conn = getConnection(
        getConnectionString());
    MySqlCommand cmd = new MySqlCommand(
        @" select_*
        .....from_v_a_quest_item

```

```

.....where ca_quest = @caQuest
.....order by nr_order", conn);
    cmd.Parameters.AddWithValue("@caQuest", idQuest);
    MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
    );
    DataSet dataset = new DataSet();
    adapter.Fill(dataset);
    conn.Close();
    return dataset.Tables[0];
} /* getQuestItems */

/// <summary>
/// Ritorna il numero di domande per idQuest
/// </summary>
/// <param name="idQuest"></param>
/// <returns></returns>
public static int getNumDomande(int idQuest)
{
    MySqlCommand cmd = buildCommand(
        @"select count(*)
.....from a_quest_item
.....where ca_quest = @idQuest");
    cmd.Parameters.AddWithValue("@idQuest", idQuest);
    cmd.ExecuteScalar();
    object result = cmd.ExecuteScalar();
    if (result != null)
        return Convert.ToInt32(result.ToString());
    else
        return 0;
} /* getNumDomande */
#endregion QUESTIONARI

```



```

#region GRAFICO

/// <summary>
/// Ritorna DataTable di tutti i questionari del
/// paziente associato
/// al questionario idMQuest.
/// </summary>
/// <param name="idMQuest"></param>
/// <returns></returns>
public static DataTable getRisultatiPaziente(int
    idMQuest)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"select _Data, _nr_risultato1, _nr_risultato2, _
            nr_risultato3
            from _v_m_quest
            where _IdPaziente =_ ( _select _ca_paziente _from _
            m_quest _where _id =_ @idMQuest _)
            order _by _Data
            limit _20", conn);
    cmd.Parameters.AddWithValue("@idMQuest", idMQuest);
    MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
    );
    DataSet dataset = new DataSet();
    adapter.Fill(dataset);
    conn.Close();
    return dataset.Tables[0];
} /* getRisultatiPaziente */
#endregion GRAFICO

```

```

#region AMMINISTRAZIONE
/// <summary>
/// Inserisce un record in z_log
/// </summary>
/// <param name="ddOp"></param>
/// <param name="ddMsg"></param>
/// <param name="czUtente"></param>
public static void insertZLog(string ddOp, string
    ddMsg, int czUtente)
{
    MySqlCommand cmdInsert = buildCommand(
        @"insert into z_log (dd_op, dd_msg, cz_utente,
            dt_now)
        values (@ddOp, @ddMsg, @czUtente, now())");
    cmdInsert.Parameters.AddWithValue("@ddOp", ddOp);
    cmdInsert.Parameters.AddWithValue("@ddMsg", ddMsg);
    cmdInsert.Parameters.AddWithValue("@czUtente",
        czUtente);
    cmdInsert.ExecuteNonQuery();
} /* insertZLog */

public static DataTable getLog()
{
    MySqlConnection conn = getConnection(
        getConnectionString());
    MySqlCommand cmd = new MySqlCommand(
        @"select
        id, dd_op, dd_msg, cz_utente, dt_now,
        dd_login
        from v_log

```

```

.....order_by_id_desc", conn);
    MySqlConnection adapter = new MySqlConnection(cmd
    );
    DataSet dataset = new DataSet();
    adapter.Fill(dataset);
    conn.Close();
    // Forza id come pk
    dataset.Tables[0].PrimaryKey = new DataColumn[1] {
        dataset.Tables[0].Columns["id"] };
    return dataset.Tables[0];
} /* getLog */

public static void updateUtente(
    int id,
    string ddLogin, string ddPassword, string ddNome,
    string ddCognome,
    string ddMail, int czProfilo)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"update_z_utente
.....set_dd_login = @ddLogin,
.....dd_password = @ddPassword,
.....dd_nome = @ddNome,
.....dd_cognome = @ddCognome,
.....dd_mail = @ddMail,
.....cz_profilo = @czProfilo
.....where_id=@id", conn);
    cmd.Parameters.AddWithValue("@ddLogin", ddLogin);
    cmd.Parameters.AddWithValue("@ddPassword",
        ddPassword);

```

```

cmd.Parameters.AddWithValue("@ddNome" , ddNome);
cmd.Parameters.AddWithValue("@ddCognome" , ddCognome
);
cmd.Parameters.AddWithValue("@ddMail" , ddMail);
cmd.Parameters.AddWithValue("@czProfilo" , czProfilo
);
cmd.Parameters.AddWithValue("@id" , id);
cmd.ExecuteNonQuery();
} /* updateUtente */

public static void deleteUtente(int id)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"delete from z_utente
-----where id=@id" , conn);
    cmd.Parameters.AddWithValue("@id" , id);
    cmd.ExecuteNonQuery();
} /* deleteUtente */

public static int insertUtente(
    string ddLogin, string ddPassword, string ddNome,
    string ddCognome,
    string ddMail, int czProfilo)
{
    MySqlConnection conn = getConnection();
    MySqlCommand cmd = new MySqlCommand(
        @"insert into z_utente (dd_login , dd_password ,
        dd_nome , dd_cognome ,
-----dd_mail , cz_profilo)

```

```

        values (@ddLogin, @ddPassword, @ddNome,
        @ddCognome, @ddMail, @czProfilo)", conn);
        cmd.Parameters.AddWithValue("@ddLogin", ddLogin);
        cmd.Parameters.AddWithValue("@ddPassword",
            ddPassword);
        cmd.Parameters.AddWithValue("@ddNome", ddNome);
        cmd.Parameters.AddWithValue("@ddCognome", ddCognome
        );
        cmd.Parameters.AddWithValue("@ddMail", ddMail);
        cmd.Parameters.AddWithValue("@czProfilo", czProfilo
        );
        cmd.ExecuteNonQuery();
        MySqlCommand cmdLastId = new MySqlCommand("select
            last_insert_id()", conn);
        cmdLastId.ExecuteScalar();
        object result = cmdLastId.ExecuteScalar();
        if (result != null)
            return Convert.ToInt32(result.ToString());
        else
            return 0;
    } /* insertUtente */

    public static DataTable getUtenti()
    {
        MySqlConnection conn = getConnection(
            getConnectionString());
        MySqlCommand cmd = new MySqlCommand(
            @"select _id, _dd_login, _dd_nome, _dd_cognome,
            _dd_mail, _dd_profilo, _dd_password, _cz_profilo
        from v_utenti_profilo
        order by _id", conn);
    }

```

```

        MySqlConnection conn = getConnection(
            getConnectionString());
        MySqlCommand cmd = new MySqlCommand(
            @"select
.....id ,_dd_descrizione
.....from _z_profilo
.....order_by_id", conn);
        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
            );
        DataSet dataset = new DataSet();
        adapter.Fill(dataset);
        conn.Close();
        return dataset.Tables[0];
    } /* getUtenti */

    public static DataTable getProfili()
    {
        MySqlConnection conn = getConnection(
            getConnectionString());
        MySqlCommand cmd = new MySqlCommand(
            @"select
.....id ,_dd_descrizione
.....from _z_profilo
.....order_by_id", conn);
        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
            );
        DataSet dataset = new DataSet();
        adapter.Fill(dataset);
        conn.Close();
        return dataset.Tables[0];
    } /* getProfili */

#endregion AMMINISTRAZIONE

#region PAZIENTI

```

```
public static void setRecPaziente(int idPaziente)
{
    if (DALRuntime.RecPaziente == null)
        DALRuntime.RecPaziente = new DALRuntime.
            TRecPaziente();
    if (idPaziente > 0)
        DALRuntime.RecPaziente.readPaziente(idPaziente);
}

public static TRecPaziente getRecPaziente(int
    idPaziente)
{
    if (DALRuntime.RecPaziente == null)
        setRecPaziente(idPaziente);
    return DALRuntime.RecPaziente;
}

/// <summary>
/// Ritorna cognome e nome concatenato. Usata all '
/// inizio della compilazione dei questionari
/// </summary>
/// <returns></returns>
public static DataTable getPazienti()
{
    MySqlConnection conn = getConnection(
        getConnectionString());
    MySqlCommand cmd = new MySqlCommand("select _concat(
        dd_cognome, '_', _dd_nome) _CognomeNome, _id _from _
        a_paziente _order _by _dd_cognome", conn);
    MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
        );
```

```

    DataSet dataset = new DataSet();
    adapter.Fill(dataset);
    conn.Close();
    return dataset.Tables[0];
} /* getPazienti */

/// <summary>
/// Usata in FmPazienti
/// </summary>
/// <returns></returns>
public static DataTable getPazienti2()
{
    MySqlConnection conn = getConnection(
        getConnectionString());
    MySqlCommand cmd = new MySqlCommand(
        @"select
        .....id, _dd_cognome, _dd_nome, _ct_titolo, _cd_sesso,
        _dt_nascita,
        .....cd_fiscale, _ct_stato_civile, _dd_titolo, _
        dd_stato_civile,
        .....dd_medico_nome, _dd_medico_cognome, _
        dd_medico_email, _dd_foto,
        .....bl_nota
        .....from _v_a_paziente
        .....order _by _dd_cognome", conn);
    MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
    );
    DataSet dataset = new DataSet();
    adapter.Fill(dataset);
    conn.Close();
    // Forza id come pk

```



```

        dataset.Tables[0].PrimaryKey = new DataColumn[1] {
            dataset.Tables[0].Columns["id"] };
        return dataset.Tables[0];
    } /* getPazienti2 */

    /// <summary>
    /// Ritorna datatable vuota su a_pazienti. Usata per
    /// insert
    /// </summary>
    /// <returns></returns>
    public static DataTable getPazientiEmpty()
    {
        MySqlConnection conn = getConnection(
            getConnectionString());
        MySqlCommand cmd = new MySqlCommand(
            @"select
            .....id, _dd_cognome, _dd_nome, _ct_titolo, _cd_sesso,
            _cast(dt_nascita_as_date)_dt_nascita,
            .....cd_fiscale, _ct_stato_civile, _dd_titolo, _
            dd_stato_civile
            .....from _v_a_paziente
            .....where _0=1", conn);
        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
            );
        DataSet dataset = new DataSet();
        adapter.Fill(dataset);
        conn.Close();
        return dataset.Tables[0];
    } /* getPazientiEmpty */

    /// <summary>

```

```

    /// Apre a_paziente su id=idPaziente
    /// </summary>
    /// <param name="idPaziente"></param>
    /// <returns></returns>
    public static DataTable getPazienteById(int
        idPaziente)
    {
        MySqlConnection conn = getConnection(
            getConnectionString());
        MySqlCommand cmd = new MySqlCommand(
            @"select
            .....id ,_dd_cognome ,_dd_nome ,_ct_titolo ,_cd_sesso ,
            _dt_nascita ,
            .....cd_fiscale ,_ct_stato_civile ,_dd_titolo ,_
            dd_stato_civile
            .....from_v_a_paziente
            .....where_id=_@idPaziente", conn);
        cmd.Parameters.AddWithValue("@idPaziente",
            idPaziente);
        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
            );
        DataSet dataset = new DataSet();
        adapter.Fill(dataset);
        conn.Close();
        return dataset.Tables[0];
    } /* getPazienteById */

    /// <summary>
    /// Cancella il record idPaziente
    /// </summary>
    /// <param name="idPaziente"></param>

```

```
/// <returns></returns>
public static void deletePazienteById(int idPaziente)
{
    MySqlConnection conn = getConnection(
        getConnectionString());
    MySqlCommand cmd = new MySqlCommand(
        @"delete
        .....from a_paziente
        .....where id = @idPaziente", conn);
    cmd.Parameters.AddWithValue("@idPaziente",
        idPaziente);
    cmd.ExecuteNonQuery();
    conn.Close();
} /* deletePazienteById */

public class TRecPaziente
{
    public int idPaziente { get; set; }
    public string Cognome { get; set; }
    public string Nome { get; set; }
    public DateTime DataNascita { get; set; }
    public string Sesso { get; set; }
    public string CdFiscale { get; set; }
    public int CtTitolo { get; set; }
    public int CtStatoCivile { get; set; }
    public string ddFoto { get; set; }
    public string MedicoCognome { get; set; }
    public string MedicoNome { get; set; }
    public string MedicoMail { get; set; }
    public string Note { get; set; }
```

```

public string getSaluto()
{
    if (Sesso == "F")
        return "sig.a_" + Nome + "_" + Cognome;
    else
        return "sig._" + Nome + "_" + Cognome;
}

/// <summary>
/// Si compila leggendo da a_paziente dove id=
/// idPaziente
/// </summary>
/// <param name="idPaziente"></param>
public void readPaziente(int idPaziente)
{
    this.idPaziente = idPaziente;
    MySqlCommand cmd = buildCommand(
        @"select _id , _dd_nome , _dd_cognome , _cd_sesso , _
            dt_nascita , _cd_fiscale , _ct_titolo ,
            .....ct_stato_civile , _dd_foto
            .....from _a_paziente
            .....where _id =_ @idPaziente");
    cmd.Parameters.AddWithValue("@idPaziente",
        idPaziente);
    MySqlDataReader r = cmd.ExecuteReader();
    if (r.Read())
    {
        this.Cognome = r["DD_COGNOME"].ToString();
        this.Nome = r["DDNOME"].ToString();
        this.Sesso = r["CD_SESSO"].ToString();
    }
}

```

```

        this.DataNascita = Convert.ToDateTime(r["
            DT_NASCITA"].ToString());
        this.CdFiscale = r["CD_FISCALE"].ToString();
        this.CtTitolo = Convert.ToInt32(r["CT_TITOLO"].
            ToString());
        this.CtStatoCivile = Convert.ToInt32(r["
            CT_STATO_CIVILE"].ToString());
        this.ddFoto = r["DD_FOTO"].ToString();
    }
    else
    {
        this.Cognome = string.Empty;
        this.Nome = string.Empty;
        this.Sesso = string.Empty;
        this.CdFiscale = string.Empty;
        this.CtTitolo = 0;
        this.CtStatoCivile = 0;
        // TODO: cosa faccio con la data di nascita ??
        this.ddFoto = string.Empty;
    }
    r.Close();
} /* readPaziente */

public void updatePaziente()
{
    MySqlCommand cmd = buildCommand(
        @"update _a_paziente
        .....set _dd_nome_=@ddNome, _dd_cognome_=@
        @ddCognome,
        .....cd_sesso_=@cdSesso, _dt_nascita_=@
        @dtNascita,

```

```
.....cd_fiscale = @cdFiscale , ct_titolo =
    @ctTitolo ,
.....ct_stato_civile = @tStatoCivile ,
.....dd_medico_nome = @ddMedicoNome ,
.....dd_medico_cognome = @ddMedicoCognome ,
.....dd_medico_email = @ddMedicoEmail ,
.....dd_foto = @ddFoto ,
.....bl_nota = @blNota
.....where_id = @idPaziente" );
    cmd.Parameters.AddWithValue( " @ddNome" , this.Nome)
        ;
    cmd.Parameters.AddWithValue( " @ddCognome" , this.
        Cognome) ;
    cmd.Parameters.AddWithValue( " @cdSesso" , this.
        Sesso) ;
    cmd.Parameters.AddWithValue( " @dtNascita" , this.
        DataNascita) ;
    cmd.Parameters.AddWithValue( " @cdFiscale" , this.
        CdFiscale) ;
    cmd.Parameters.AddWithValue( " @ctTitolo" , this.
        CtTitolo) ;
    cmd.Parameters.AddWithValue( " @tStatoCivile" , this
        . CtStatoCivile) ;
    cmd.Parameters.AddWithValue( " @idPaziente" , this.
        idPaziente) ;
    cmd.Parameters.AddWithValue( " @ddMedicoNome" , this
        . MedicoNome) ;
    cmd.Parameters.AddWithValue( " @ddMedicoCognome" ,
        this.MedicoCognome) ;
    cmd.Parameters.AddWithValue( " @ddMedicoEmail" ,
        this.MedicoMail) ;
```

```

        cmd.Parameters.AddWithValue("@ddFoto", this.
            ddFoto);
        cmd.Parameters.AddWithValue("@blNota", this.Note)
            ;
        cmd.ExecuteNonQuery();
    } /* updatePaziente */

    public void insertPaziente()
    {
        MySqlCommand cmd = buildCommand(
            @"insert into a_paziente
            .....(dd_nome, dd_cognome, cd_sesso, dt_nascita,
            cd_fiscale,
            ct_titolo, ct_stato_civile, dd_medico_nome,
            dd_medico_cognome, dd_medico_email, dd_foto,
            bl_nota, cz_utente)
            .....values (@ddNome, @ddCognome, @cdSesso,
            @dtNascita,
            @cdFiscale, @ctTitolo, @ctStatoCivile,
            @ddMedicoNome, @ddMedicoCognome,
            @ddMedicoEmail, @ddFoto,
            @blNota, @czUtente)");
        cmd.Parameters.AddWithValue("@ddNome", this.Nome)
            ;
        cmd.Parameters.AddWithValue("@ddCognome", this.
            Cognome);
        cmd.Parameters.AddWithValue("@cdSesso", this.
            Sesso);
        cmd.Parameters.AddWithValue("@dtNascita", this.
            DataNascita);
    }

```

```
cmd.Parameters.AddWithValue("@cdFiscale", this.
    CdFiscale);
cmd.Parameters.AddWithValue("@ctTitolo", this.
    CtTitolo);
cmd.Parameters.AddWithValue("@ctStatoCivile",
    this.CtStatoCivile);
cmd.Parameters.AddWithValue("@ddMedicoNome", this
    .MedicoNome);
cmd.Parameters.AddWithValue("@ddMedicoCognome",
    this.MedicoCognome);
cmd.Parameters.AddWithValue("@ddMedicoEmail",
    this.MedicoMail);
cmd.Parameters.AddWithValue("@ddFoto", this.
    ddFoto);
cmd.Parameters.AddWithValue("@blNota", this.Note)
    ;
cmd.Parameters.AddWithValue("@czUtente",
    DALRuntime.IdUtente);
cmd.ExecuteNonQuery();
} /* insertPaziente */

}
#endregion PAZIENTI

#region UTENTI
public class TRecUtente
{
    public int idUtente { get; set; }
    public int czProfilo { get; set; }
    string Nome { get; set; }
    string Cognome { get; set; }
```



```
string DdProfilo { get; set; }

public void init(int idUtente, int czProfilo,
    string Nome, string Cognome, string DdProfilo)
{
    this.czProfilo = czProfilo;
    this.idUtente = idUtente;
    this.Nome = Nome;
    this.Cognome = Cognome;
    this.DdProfilo = DdProfilo;
}

public string getDescrizione()
{
    return "utente:" + this.Nome + " " + this.
        Cognome + " - profilo:" + this.DdProfilo;
}
}

public static void setRecUtente(int idUtente, int
    czProfilo, string Nome, string Cognome, string
    DdProfilo)
{
    if (DALRuntime.RecUtente == null)
        DALRuntime.RecUtente = new DALRuntime.TRecUtente
            ();
    DALRuntime.RecUtente.init(idUtente, czProfilo, Nome
        , Cognome, DdProfilo);
    DALRuntime.IdUtente = RecUtente.idUtente;
    DALRuntime.IdProfilo = RecUtente.czProfilo;
}
}
```

```
#endregion

#region UTILITY
public static string getAbsolutePath(string
    sImagePath, string sFilename)
{
    string relAppPath = HttpContext.Current.Server.
        MapPath("~/");
    if (!relAppPath.EndsWith("/"))
        relAppPath += "/";
    return (sImagePath.Replace("~/", relAppPath) +
        sFilename).Replace('/', '\\');
}

public static string checkAndGetImagePath(string
    sImagePath, string sFoto)
{
    string sImage = sImagePath;
    // espande la tilde nel percorso fisico per
    // verificare l'esistenza del file
    string sPath = getAbsolutePath(sImagePath, sFoto);
    if (System.IO.File.Exists(sPath))
        sImage += sFoto;
    else
        sImage += fImageNotPresent;
    return sImage;
}
#endregion
}
```

Listing 4.18: EmmaMembershipProvider.cs

```
using System;
```

```
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using MySql.Data.MySqlClient;

/// <summary>
/// Summary description for EmmaMembershipProvider
/// </summary>
public class EmmaMembershipProvider :
    MembershipProvider
{
    private string fConnString;

    public override void Initialize(string name, System.
        Collections.Specialized.NameValueCollection config)
    {
        fConnString = config["connectionString"];
        base.Initialize(name, config);
    }

    private bool emmaIsValidLogon(string Login, string
        Passw)
    {
        MySqlCommand cmd = new MySqlCommand(
            @"select _id, _dd_password, _dd_nome, _dd_cognome, _
                cz_profilo, _dd_profilo
```

```

.....from_v_utenti_profilo
.....where_upper(dd_login) = upper(@ddLogon)",
    DALRuntime.getConnection());
cmd.Parameters.AddWithValue("@ddLogon", Login);
MySqlDataReader r = cmd.ExecuteReader();
bool userFound = false;
while (r.Read())
{
    userFound = true;
    // TODO: cifrare la password
    // check password
    userFound = string.Compare(r["dd_password"].
        ToString(), Passw, false) == 0;
    if (userFound)
    {
        // Salva in sessione IdUtente e IdProfilo
        int idUtente = Convert.ToInt32(r["id"]);
        int idProfilo = Convert.ToInt32(r["cz_profilo"
            ]);
        string nome = r["dd_nome"].ToString();
        string cognome = r["dd_cognome"].ToString();
        string ddProfilo = r["dd_profilo"].ToString();
        DALRuntime.setRecUtente(idUtente, idProfilo,
            nome, cognome, ddProfilo);
    }
}
r.Close();
// Log del login :)
if (userFound)
    DALRuntime.insertZLog("login", Login, DALRuntime.
        IdUtente);

```

```
    else
        DALRuntime.insertZLog("login_attempt", Login, 0);
    return userFound;
} /* emmaIsValidLogon */

public override bool ValidateUser(string username,
    string password)
{
    bool isGood = emmaIsValidLogon(username, password);
    return isGood;
}

    public EmmaMembershipProvider()
    {
        //
        // TODO: Add constructor logic here
        //
    }

public override string ApplicationName
{
    get
    {
        throw new Exception("The method or operation is not implemented.");
    }
    set
    {
        throw new Exception("The method or operation is not implemented.");
    }
}
```

```
}

public override bool ChangePassword(string username ,
    string oldPassword , string newPassword)
{
    throw new Exception("The_method_or_operation_is_not
        implemented.");
}

public override bool ChangePasswordQuestionAndAnswer(
    string username , string password , string
    newPasswordQuestion , string newPasswordAnswer)
{
    throw new Exception("The_method_or_operation_is_not
        implemented.");
}

public override MembershipUser CreateUser(string
    username , string password , string email , string
    passwordQuestion , string passwordAnswer , bool
    isApproved , object providerUserKey , out
    MembershipCreateStatus status)
{
    throw new Exception("The_method_or_operation_is_not
        implemented.");
}

public override bool DeleteUser(string username , bool
    deleteAllRelatedData)
{
```

```
        throw new Exception("The method or operation is not
            implemented.");
    }

    public override bool EnablePasswordReset
    {
        get { throw new Exception("The method or operation
            is not implemented."); }
    }

    public override bool EnablePasswordRetrieval
    {
        get { throw new Exception("The method or operation
            is not implemented."); }
    }

    public override MembershipUserCollection
        FindUsersByEmail(string emailToMatch, int pageIndex
        , int pageSize, out int totalRecords)
    {
        throw new Exception("The method or operation is not
            implemented.");
    }

    public override MembershipUserCollection
        FindUsersByName(string usernameToMatch, int
        pageIndex, int pageSize, out int totalRecords)
    {
        throw new Exception("The method or operation is not
            implemented.");
    }
}
```

```
public override MembershipUserCollection GetAllUsers(  
    int pageIndex, int pageSize, out int totalRecords)  
{  
    throw new Exception("The method or operation is not  
        implemented.");  
}
```

```
public override int GetNumberOfUsersOnline()  
{  
    throw new Exception("The method or operation is not  
        implemented.");  
}
```

```
public override string GetPassword(string username,  
    string answer)  
{  
    throw new Exception("The method or operation is not  
        implemented.");  
}
```

```
public override MembershipUser GetUser(string  
    username, bool userIsOnline)  
{  
    throw new Exception("The method or operation is not  
        implemented.");  
}
```

```
public override MembershipUser GetUser(object  
    providerUserKey, bool userIsOnline)  
{
```



```
        throw new Exception("The method or operation is not  
            implemented.");  
    }
```

```
public override string GetUserNameByEmail(string  
    email)  
{  
    throw new Exception("The method or operation is not  
        implemented.");  
}
```

```
public override int MaxInvalidPasswordAttempts  
{  
    get { throw new Exception("The method or operation  
        is not implemented."); }  
}
```

```
public override int  
    MinRequiredNonAlphanumericCharacters  
{  
    get { throw new Exception("The method or operation  
        is not implemented."); }  
}
```

```
public override int MinRequiredPasswordLength  
{  
    get { throw new Exception("The method or operation  
        is not implemented."); }  
}
```

```
public override int PasswordAttemptWindow
```

```
{  
    get { throw new Exception("The_method_or_operation_  
        is_not_implemented."); }  
}
```

```
public override MembershipPasswordFormat
```

```
    PasswordFormat
```

```
{  
    get { throw new Exception("The_method_or_operation_  
        is_not_implemented."); }  
}
```

```
public override string
```

```
    PasswordStrengthRegularExpression
```

```
{  
    get { throw new Exception("The_method_or_operation_  
        is_not_implemented."); }  
}
```

```
public override bool RequiresQuestionAndAnswer
```

```
{  
    get { throw new Exception("The_method_or_operation_  
        is_not_implemented."); }  
}
```

```
public override bool RequiresUniqueEmail
```

```
{  
    get { throw new Exception("The_method_or_operation_  
        is_not_implemented."); }  
}
```

```
public override string ResetPassword(string username,
    string answer)
{
    throw new Exception("The method or operation is not
        implemented.");
}

public override bool UnlockUser(string userName)
{
    throw new Exception("The method or operation is not
        implemented.");
}

public override void UpdateUser(MembershipUser user)
{
    throw new Exception("The method or operation is not
        implemented.");
}
}
```

Listing 4.19: EmmaSitemapProvider.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Security.Permissions;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Collections.Specialized;
```

```
using MySql.Data.MySqlClient;

/// <summary>
/// Summary description for EmmaSitemapProvider
/// </summary>
[AspNetHostingPermission(SecurityAction.Demand, Level =
    AspNetHostingPermissionLevel.Minimal)]
public class EmmaSitemapProvider :
    StaticSiteMapProvider
{
    private SiteMapNode fRootNode = null;
    private int fIdProfilo = 1; // default profilo
        pubblico
    private string fConnString;

        public EmmaSitemapProvider() {}

    public override void Initialize(string name, System.
        Collections.Specialized.NameValueCollection
        attributes)
    {
        fConnString = attributes["connectionString"];
        base.Initialize(name, attributes);
    }

    private int DoBuildSiteMap(
        SiteMapNode CurrRoot,
        DataTable TblMenu,
        int CurrRowMenu,
        int PrecLevel,
        int idProfilo)
```

```
{
    NameValueCollection Attributes = new
        NameValueCollection();
// Uncomment per usare window separata
//    Attributes.Add("target", "_blank");
    int i = CurrRowMenu;
    while (i < TblMenu.Rows.Count)
    {
        DataRow RowMenu = TblMenu.Rows[i];
        if (i < TblMenu.Rows.Count)
        {
            string DescrNodo = RowMenu["DD_DESCRIZIONE"].
                ToString();
            int IdNodo = Convert.ToInt32(RowMenu["ID"].
                ToString());
            string sUrl = RowMenu["DD_URL"].ToString();

            SiteMapNode NewNode = new SiteMapNode(this,
                IdNodo.ToString(), sUrl, DescrNodo);

            if (CurrRoot == null)
            {
                CurrRoot = NewNode;
                fRootNode = CurrRoot;
                AddNode(NewNode, null);
            }
            else
            {
                AddNode(NewNode, fRootNode);
                // Se Url è vuoto => è un parent => cerca
                figli
            }
        }
    }
}
```

```

if (sUrl == string.Empty)
{
    DataTable t = openSubMenu(idProfilo , IdNodo
        );
    for (int r = 0; r <= t.Rows.Count - 1; r++)
    {
        string DescrNodo2 = t.Rows[r][ "
            DD_DESCRIZIONE" ]. ToString ();
        string sIdNodo2 = t.Rows[r][ "ID" ].
            ToString ();
        string sUrl2 = t.Rows[r][ "DD_URL" ].
            ToString ();
        AddNode(new SiteMapNode(this , sIdNodo2 ,
            sUrl2 , DescrNodo2), NewNode);
    }
}
}
i++;
}
/*****
    DataRow RowMenu = TblMenu.Rows[i];

    string DescrNodo = RowMenu["DD_DESCRIZIONE
        "]. ToString ();
    decimal IdNodo = decimal.Parse(RowMenu["ID
        "]. ToString ());
    string Url = ""; // IdNodo.ToString(); //
        forza qualcosa nell'url altrimenti il
        nodo viene ignorato

```

```
SiteMapNode NewNode = new SiteMapNode(this ,
    IdNodo.ToString(), Url, DescrNodo);
if (CurrRoot == null)
{
    CurrRoot = NewNode;
    fRootNode = CurrRoot;
}
else
    AddNode(NewNode, CurrRoot);

i++;
// Se non siamo sull'ultimo record guarda
// se il prossimo livello è inferiore
if (i < TblMenu.Rows.Count)
{
    int NextLevel = int.Parse(TblMenu.Rows[i
        ]["LEVEL"].ToString());
    if (CurrLevel < NextLevel)
    {
        i = DoBuildSiteMap(NewNode, TblMenu, i,
            CurrLevel);
    }
    else
        // Se invece il nodo successivo ha
        // livello inferiore risali
        if (NextLevel < CurrLevel)
            return i;
}
}
*****/
}
return i;
```

```
    } /* DoBuildSiteMap */

protected override void Clear()
{
    lock (this)
    {
        fRootNode = null;
        base.Clear();
    }
}

private bool isProfiloChanged()
{
    if (DALRuntime.IdProfilo != 0)
        return fIdProfilo != DALRuntime.IdProfilo;
    else
        return true;
}

public override SiteMapNode RootNode
{
    get
    {
        SiteMapNode temp = null;
        temp = BuildSiteMap();
        return temp;
    }
}

private DataTable openMenu(int idProfilo)
{
```



```

        MySqlCommand cmd = new MySqlCommand(
            @"select *
            .....from z_funzione
            .....where cz_profilo <= @CzProfilo
            .....and id_parent < 0", DALRuntime.getConnection
        ());
        cmd.Parameters.AddWithValue("@CzProfilo", idProfilo
        );
        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
        );
        DataSet dataset = new DataSet();
        adapter.Fill(dataset);
        return dataset.Tables[0];
    }

```

```

private DataTable openSubMenu(int idProfilo, int
    idParent)
{
    MySqlCommand cmd = new MySqlCommand(
        @"select *
        .....from z_funzione
        .....where cz_profilo <= @CzProfilo
        .....and id_parent = @IdParent", DALRuntime.
        getConnection());
        cmd.Parameters.AddWithValue("@CzProfilo", idProfilo
        );
        cmd.Parameters.AddWithValue("@IdParent", idParent);
        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd
        );
        DataSet dataset = new DataSet();
        adapter.Fill(dataset);

```

```
        return dataset.Tables[0];
    }

    public override SiteMapNode BuildSiteMap()
    {
        lock (this)
        {
            // Costruisce sitemap se RootNode è null OPPURE
            // se è cambiato il profilo
            if (null == fRootNode || isProfiloChanged()) {

                fIdProfilo = DALRuntime.IdProfilo;

                // Start with a clean state
                Clear();

                DataTable TblMenu = openMenu(fIdProfilo);
                DoBuildSiteMap(fRootNode, TblMenu, 0, 0,
                    fIdProfilo);
            }
            return fRootNode;
        }
    }

    protected override SiteMapNode GetRootNodeCore()
    {
        throw new Exception("The method or operation is not
            implemented.");
    }
}
```

Listing 4.20: EmmaRoleProvider.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Configuration.Provider;
using System.Collections;
using System.Collections.Specialized;
using MySql.Data.MySqlClient;

/// <summary>
/// Summary description for EmmaRoleProvider
/// </summary>
public class EmmaRoleProvider : RoleProvider
{
    private string fConnString;

    public EmmaRoleProvider()
    {
        //
        // TODO: Add constructor logic here
        //
    }

    public override void Initialize(string name,
        NameValueCollection config)
```

```
{
    if (config == null)
        throw new ArgumentNullException("config");
    if (String.IsNullOrEmpty(name))
        name = "AccessRoleProvider";
    if (string.IsNullOrEmpty(config["description"]))
    {
        config.Remove("description");
        config.Add("description", "EmmaRoleProvider");
    }
    fConnString = config["connectionString"];
    base.Initialize(name, config);
}

public override void AddUsersToRoles(string []
    usernames, string [] roleNames)
{
    throw new Exception("The method or operation is not
        implemented.");
}

public override string ApplicationName
{
    get
    {
        throw new Exception("The method ApplicationName
            Get or operation is not implemented.");
    }
    set
    {

```

```
        throw new Exception("The method ApplicationName_
            set_or_operation_is_not_implemented.");
    }
}

public override void CreateRole(string roleName)
{
    throw new Exception("The method CreateRole_or_
        operation_is_not_implemented.");
}

public override bool DeleteRole(string roleName, bool
    throwOnPopulatedRole)
{
    throw new Exception("The method DeleteRole_or_
        operation_is_not_implemented.");
}

public override string[] FindUsersInRole(string
    roleName, string usernameToMatch)
{
    throw new Exception("The method FindUsersInRole_or_
        operation_is_not_implemented.");
}

private int getCountRoles()
{
    MySqlCommand cmd = new MySqlCommand("select count
        (*) from z_profilo", DALRuntime.getConnection());
    object result = cmd.ExecuteScalar();
    cmd.Connection.Close();
}
```

```
    if (result != null)
    {
        int r = Convert.ToInt32(result);
        return r;
    }
    else
        return -1;
}

/// <summary>
/// Ritorna il contenuto di z_profilo
/// </summary>
/// <returns></returns>
public override string [] GetAllRoles()
{
    int ctrRoles = getCountRoles();
    if (ctrRoles > 0)
    {
        string [] emmaRoles = new string [ctrRoles];
        MySqlCommand cmd = new MySqlCommand("select _
            dd_descrizione_from_z_profilo", DALRuntime.
            getConnection());
        int i = 0;
        MySqlDataReader r = cmd.ExecuteReader();
        while (r.Read())
            emmaRoles [i++] = r [0].ToString();
        r.Close();
        return emmaRoles;
    }
    else
        return new string [1] {""};
}
```

```
}

private string getProfiloUtente(string username)
{
    MySqlCommand cmd = new MySqlCommand(
        @"select z.dd_descrizione
        from z_profilo z
        where z.id = (
        select u.cz_profilo
        from z_utente u
        where u.dd_login = @ddLogon)", DALRuntime.
        getConnection());

    cmd.Parameters.AddWithValue("@ddLogon", username);
    object result = cmd.ExecuteScalar();
    cmd.Connection.Close();
    if (result != null)
        return result.ToString();
    else
        return string.Empty;
}

// Per ora gestisce solo un ruolo per username
public override string[] GetRolesForUser(string
    username)
{
    return new string[1] { getProfiloUtente(username)
        };
}
```

```
public override string [] GetUsersInRole(string
    roleName)
{
    throw new Exception("The method GetUsersInRole or
        operation is not implemented.");
}

public override bool IsUserInRole(string username,
    string roleName)
{
    string roleUser = getProfiloUtente(username);
    return string.Compare(roleName, roleUser, true) ==
        0;
}

public override void RemoveUsersFromRoles(string []
    usernames, string [] roleNames)
{
    throw new Exception("The method
        RemoveUsersFromRoles or operation is not
        implemented.");
}

public override bool RoleExists(string roleName)
{
    throw new Exception("The method RoleExists or
        operation is not implemented.");
}
}
```

4.2.5 Gestione questionari

Listing 4.21: FmArchivio.aspx

```
ï»¿<%@ Page Title="" Language="C#" MasterPageFile="~/
  MasterEmma.master" AutoEventWireup="true" CodeFile="
  FmArchivio.aspx.cs" Inherits="Tabelle_FmArchivio" %>

<%@ Register Assembly="DevExpress.Web.v9.3, Version
  =9.3.6.0, Culture=neutral, PublicKeyToken=
  b88d1754d700e49a"
  Namespace="DevExpress.Web.ASPxMenu" TagPrefix="dx" %>

<%@ Register Assembly="DevExpress.Web.ASPxGridView.v9
  .3.Export, Version=9.3.6.0, Culture=neutral,
  PublicKeyToken=b88d1754d700e49a"
  Namespace="DevExpress.Web.ASPxGridView.Export"
  TagPrefix="dx" %>

<%@ Register Assembly="DevExpress.Web.v9.3, Version
  =9.3.6.0, Culture=neutral, PublicKeyToken=
  b88d1754d700e49a"
  Namespace="DevExpress.Web.ASPxPopupControl" TagPrefix
  ="dx" %>

<%@ Register Assembly="DevExpress.Web.ASPxGridView.v9
  .3, Version=9.3.6.0, Culture=neutral, PublicKeyToken=
  b88d1754d700e49a"
  Namespace="DevExpress.Web.ASPxGridView" TagPrefix="dx
  " %>

<%@ Register assembly="DevExpress.Web.ASPxEditors.v9.3,
  Version=9.3.6.0, Culture=neutral, PublicKeyToken=
  b88d1754d700e49a" namespace="DevExpress.Web.
```

```
ASPxEitors" tagprefix="dx" %>

<asp:Content ID="Content1" ContentPlaceHolderID="
  HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="
  MainContent" Runat="Server">
  <dx:ASpGridView ID="grdMQuest" runat="server"
    AutoGenerateColumns="False">
    <SettingsPager PageSize="20">
    </SettingsPager>
    <Columns>
      <dx:GridViewDataTextColumn FieldName="IdMQuest"
        Visible="False"
        VisibleIndex="0">
      </dx:GridViewDataTextColumn>
      <dx:GridViewDataTextColumn FieldName="IdAQuest"
        Visible="False"
        VisibleIndex="0">
      </dx:GridViewDataTextColumn>
      <dx:GridViewDataTextColumn FieldName="IdPaziente"
        Visible="False"
        VisibleIndex="0">
      </dx:GridViewDataTextColumn>
      <dx:GridViewDataTextColumn Caption="Cognome"
        FieldName="Cognome"
        VisibleIndex="0">
      </dx:GridViewDataTextColumn>
      <dx:GridViewDataTextColumn Caption="Nome"
        FieldName="Nome" VisibleIndex="1">
      </dx:GridViewDataTextColumn>
```

```
<dx:GridViewDataDateColumn Caption="Data"
  FileName="Data" VisibleIndex="2">
  <PropertiesDateEdit DisplayFormatString="">
  </PropertiesDateEdit>
</dx:GridViewDataDateColumn>
<dx:GridViewDataTextColumn Caption="Risultato_1"
  FileName="nr_risultato1"
  VisibleIndex="3">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption="Risultato_2"
  FileName="nr_risultato2"
  VisibleIndex="4">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption="Risultato_3"
  FileName="nr_risultato3"
  VisibleIndex="5">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption="Medico_di_
  famiglia" FileName="dd_medico_famiglia"
  VisibleIndex="6">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption="Somministrato
  _da" FileName="dd_utente"
  VisibleIndex="7">
</dx:GridViewDataTextColumn>
<dx:GridViewDataHyperLinkColumn Caption="Lettera"
  VisibleIndex="8"
  FileName="IdMQuest">
<PropertiesHyperLinkEdit
  NavigateUrlFormatString="~/Questionari/
  FmLettera.aspx?IdMQuest={0}"
```

```

        Text=" Lettera">
    </PropertiesHyperLinkEdit>
</dx:GridViewDataHyperLinkColumn>
<dx:GridViewDataHyperLinkColumn Caption=" Grafico"
    VisibleIndex="9"
    FieldName=" IdMQuest">
    <PropertiesHyperLinkEdit
        NavigateUrlFormatString=" javascript :
        ShowGraficoPopup ( '{0}' );"
        Text=" Grafico">
    </PropertiesHyperLinkEdit>
</dx:GridViewDataHyperLinkColumn>
</Columns>
<Settings ShowFilterRow=" True"
    ShowGroupPanel=" True" />
</dx:ASPxGridView>
<dx:ASPxGridViewExporter ID=" ASPxGridViewExporter"
    runat=" server"
    FileName=" Archivio _questionari" GridViewID="
    grdMQuest">
</dx:ASPxGridViewExporter>
<dx:ASPxPopupMenu ID=" popMnuGridMQuest" runat=" server
"
    onitemclick=" popMnuGridMQuest_ItemClick"
    PopupElementID=" grdMQuest">
<Items>
    <dx:MenuItem Name=" mniEsportaInExcel" Text="
    Esporta _in _Excel">
    </dx:MenuItem>
</Items>
</dx:ASPxPopupMenu>

```

```

<dx:ASPxPopupControl ID="ASPxPopupControl1" runat="
server"
  ClientInstanceName="popupGrafico" Width="800px"
  Height="600px"
  AllowDragging="True" HeaderText="Grafico_risultati_
test" Modal="True"
  PopupHorizontalAlign="Center" AllowResize="True">
<ContentCollection>
<dx:PopupControlContentControl runat="server">
</dx:PopupControlContentControl>
</ContentCollection>
</dx:ASPxPopupControl>
</asp:Content>

```

Listing 4.22: FmArchivio.aspx.cs

```

i»j using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using DevExpress.Web.ASPxGridView;
using DevExpress.Web.ASPxMenu;

public partial class Tabelle_FmArchivio : System.Web.UI
  .Page
{
  protected void Page_Init(object sender, EventArgs e)
  {
    grdMQuest.DataSource = DALRuntime.getMQuest();
  }
}

```

```
/*
protected void grdMQuestBind()
{
    grdMQuest.DataSource = DALRuntime.getMQuest();
    grdMQuest.KeyFieldName = "IdMQuest";
    grdMQuest.DataBind();
}
*/

protected void Page_Load(object sender, EventArgs e)
{
    //DevExpressUtility.GridApplyTheme(grdMQuest, DALRuntime
        .globalTheme);
    if (!IsPostBack)
    {
        grdMQuest.DataSource = DALRuntime.getMQuest();
        grdMQuest.KeyFieldName = "IdMQuest";
        grdMQuest.DataBind();
        // grdMQuestBind();
    }
    // if (!IsPostBack)
    //     grdMQuestBind();
}

protected void Page_PreInit(object sender, EventArgs
    e)
{
    // this.Theme = "Glass";
}

/*
```

```

protected void grdMQuestItem_BeforePerformDataSelect(
    object sender, EventArgs e)
{
    ASPxGridView grdDetail = sender as ASPxGridView;
    int idMQuest = Convert.ToInt32(grdDetail.
        GetMasterRowKeyValue().ToString());
    grdDetail.DataSource = DALRuntime.getMQuestItem(
        idMQuest);
    //grdDetail.DataBind();
}
*/

protected void popMnuGridMQuest_ItemClick(object
    source, DevExpress.Web.ASPxMenu.MenuItemEventArgs e
)
{
    ASPxGridViewExporter.WriteXlsToResponse();
}
}

```

Listing 4.23: FmCompila.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/
    MasterEmma.master" AutoEventWireup="true" CodeFile="
    FmCompila.aspx.cs" Inherits="Tabelle_FmCompila" %>

<%@ Register Assembly="DevExpress.Web.v9.3, _Version
    =9.3.6.0, _Culture=neutral, _PublicKeyToken=
    b88d1754d700e49a"
    Namespace="DevExpress.Web.ASPxRoundPanel" TagPrefix="
    dx" %>

```

```

<%@ Register assembly="DevExpress.Web.ASPxEditors.v9.3,
    _Version=9.3.6.0, _Culture=neutral, _PublicKeyToken=
    b88d1754d700e49a" namespace="DevExpress.Web.
    ASPxEditors" tagprefix="dx" %>
<%@ Register assembly="DevExpress.Web.v9.3, _Version
    =9.3.6.0, _Culture=neutral, _PublicKeyToken=
    b88d1754d700e49a" namespace="DevExpress.Web.ASPxPanel
    " tagprefix="dx" %>

<asp:Content ID="Content1" ContentPlaceHolderID="
    HeadContent" Runat="Server">
    <style type="text/css">
        .style1
        {
            text-align: left;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="
    MainContent" Runat="Server">
    <dx:ASPxRoundPanel ID="pnlInizio" runat="server"
        Width="600px"
        Height="300px" style="margin-left: 0px"
        HeaderText="Somministra _questionario" HeaderStyle-
        HorizontalAlign="Center">
    <PanelCollection>
    <dx:PanelContent ID="pnlSceltaPaziente" runat="
        server">
    <table width="100%">
    <tr>
    <td rowspan="6">

```



```
<dx:ASPxPanel ID="ASPxPanel1" runat="server" Width=
    "200px" Height="100%">
<PanelCollection>
<dx:PanelContent>
<div>
    <ul>
        <li class="style1">Scelga il questionario </li>
        <li class="style1">il paziente e poi </li>
        <li class="style1">prema il tasto &quot;Avanti&
            quot; </li>
    </ul>
</div>
</dx:PanelContent>
</PanelCollection>
</dx:ASPxPanel>
</td>
<td>
<dx:ASPxLabel ID="lblQuestionario" runat="server"
    Text="Scelga il questionario">
</dx:ASPxLabel>
</td>
</tr>
<tr>
<td>
<dx:ASPxListBox ID="lstQuestionari" runat="server"
    TextField="dd_descr_completa"
    ValueField="id" Height="60px">
</dx:ASPxListBox>
</td>
</tr>
</tr>
```

```
<td>
<dx:ASPxlabel ID="lblAvviso1" runat="server" Text="
  " Visible="false" ForeColor="Red">
</dx:ASPxlabel>
</td>
</tr>
<tr>
<td>
<dx:ASPxlabel ID="ASPxlabel1" runat="server" Text="
  Scelga il paziente">
</dx:ASPxlabel>
</td>
</tr>
<tr>
<td>
<dx:ASPxListBox ID="lstPazienti" runat="server"
  TextField="CognomeNome"
  ValueField="id" Width="100%" Height="350px">
</dx:ASPxListBox>
</td>
</tr>
<tr>
<td align="center">
<dx:ASPxButton ID="btnAvanti1" runat="server" Text="
  "Avanti" Width="100%"
  OnClick="btnAvanti1_Click">
</dx:ASPxButton>
</td>
</tr>
</table>
</dx:PanelContent>
```

```

    </PanelCollection>
</dx:ASPxRoundPanel>
<dx:ASPxRoundPanel ID="pnlDomanda" runat="server"
    Width="900px" Visible="false">
    <HeaderTemplate>
    </HeaderTemplate>
<PanelCollection>
<dx:PanelContent>
<table width="100%" cellpadding="0px" cellspacing="0
    px">
<tr>
<td rowspan="11">
    <dx:ASPxLabel ID="lblIstruzioni" runat="server"
        Text="Cliccare sul pallino accanto alla risposta
            scelta e premere il tasto "Avanti""
        Width="150px">
    </dx:ASPxLabel>
</td>
<td style="background-color: #006699" align="left">
    <dx:ASPxLabel ID="lblIntesta1" runat="server"
        Text="LblIntesta1" ForeColor="White">
    </dx:ASPxLabel>
</td>
</tr>
<tr>
<td style="background-color: #006699" align="left">
<table width="50%">
<tr>
<td valign="middle" align="left">
    <dx:ASPxLabel ID="lblPaziente" runat="server" Text=
        "sig. a Martha Argerich"

```

```

        ForeColor="White" Font-Bold="true" Font-Size="14
        px">
    </dx:ASPxlabel>
</td>
<td align="left">
    <dx:ASPImage ID="imgPaziente" runat="server"
        ImageUrl="~/Immagini/Pazienti/MarthaArgerich.jpeg
        " Height="60px"
        Width="60px"></dx:ASPImage>
    </td>
</tr>
</table>
</td>
</tr>
<tr>
<td style="background-color: _Gray">&nbsp;</td>
</tr>
<tr>
<td style="background-color: _Gray">
    <dx:ASPxlabel ID="lblIntesta2" runat="server" Text=
        "" ForeColor="White"
        Font-Size="14px" Font-Bold="true">
    </dx:ASPxlabel>
    </td>
</tr>
<tr>
<td style="background-color: _Gray">&nbsp;</td>
</tr>
<tr>
<td style="background-color: _Gray" align="center">

```

```
<dx:ASPxlabel ID="lblDomanda" runat="server" Text="
    " ForeColor="White"
    Font-Size="16px" Font-Bold="true">
</dx:ASPxlabel>
</td>
</tr>
<tr>
<td style="background-color:_Gray" align="right">
    <dx:ASPxlabel ID="lblNumDomanda" runat="server"
        Text="lblNumDomanda"
        ForeColor="White" Font-Size="12px">
    </dx:ASPxlabel>
</td>
</tr>
<tr>
<td align="center">
    <dx:ASPxImage ID="imgDomanda" runat="server">
    </dx:ASPxImage>
</td>
</tr>
<tr>
<td>
    <dx:ASPxlabel ID="lblAvviso" runat="server"
        ForeColor="Red" Text="lblAvviso"
        Visible="False" Font-Size="14px" Font-Bold="
            true">
    </dx:ASPxlabel>
</td>
</tr>
<tr>
<td align="center">
```

```

        <dx:ASPxRadioButtonList ID="rblRisposta" runat=
            "server" Width="100%">
        </dx:ASPxRadioButtonList>
    </td>
</tr>
<tr>
    <td align="center">
        <table width="100%">
            <tr>
                <td>
                    <dx:ASPxButton ID="btnAvanti2" runat="server"
                        OnClick="btnAvanti2_Click"
                        Text="Avanti" ToolTip="Domanda_successiva"
                        Width="100%">
                    </dx:ASPxButton>
                </td>
                <td>
                    <dx:ASPxButton ID="btnLettera" runat="server"
                        Text="Lettera"
                        ToolTip="Vai_alla_lettera" Visible="false"
                        OnClick="btnLettera_Click"></dx:
                        ASPxButton>
                </td>
                <td>
                    <dx:ASPxButton ID="btnNuovo" runat="server"
                        Text="Nuovo_questionario"
                        ToolTip="Somministra_nuovo_questionario"
                        Visible="false"
                        OnClick="btnNuovo_Click"></dx:ASPxButton>
                </td>
            </tr>
        </table>
    </td>
</tr>

```

```
        </table>
    </td>
</tr>
</table>
</dx:PanelContent>
</PanelCollection>
</dx:ASPxRoundPanel>
<dx:ASPxPanel ID="pnlCompletato" runat="server" Width
    ="200px" Visible="false">
<PanelCollection>
<dx:PanelContent>
    <dx:ASPxLabel ID="lblCompletato" runat="server"
        Text="ASPxLabel">
    </dx:ASPxLabel>
    <dx:ASPxButton runat="server" ID="btnNuovoOld">
    </dx:ASPxButton>
    <dx:ASPxButton runat="server" ID="btnLetteraOld">
    </dx:ASPxButton>
</dx:PanelContent>
</PanelCollection>
</dx:ASPxPanel>
</asp:Content>
```

Listing 4.24: FmCompila.aspx.cs

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
public partial class Tabelle_FmCompila : System.Web.UI.
    Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
            resetData();
    }

    private void resetData()
    {
        DALRuntime.iDomanda = 0;
        DALRuntime.tblDomande = null;
        DALRuntime.IdMQuest = 0;
        DALRuntime.IdAQuest = 0;
        lstPazienti.DataSource = DALRuntime.getPazienti();
        lstPazienti.DataBind();
        lstQuestionari.DataSource = DALRuntime.
            getQuestionari();
        lstQuestionari.DataBind();
        DevExUtility.PanelApplyTheme(pnlInizio, DALRuntime.
            globalTheme);
        DevExUtility.PanelApplyTheme(pnlDomanda, DALRuntime
            .globalTheme);
        DevExUtility.PanelApplyTheme(pnlCompletato,
            DALRuntime.globalTheme);
        // Forza selezione primo questionario
        lstQuestionari.SelectedIndex = 0;
        showPanelCompletato(false);
    }
}
```



```
/// <summary>
/// Inizio questionario
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void btnAvanti1_Click(object sender,
    EventArgs e)
{
    if (lstPazienti.SelectedIndex < 0)
        setAvviso1("Selezionare un paziente!");
    else
    {
        // Salva in sessione id questionario scelto
        DALRuntime.IdAQuest = Convert.ToInt32(
            lstQuestionari.SelectedItem.Value.ToString());
        // Salva in sessione numero di domande del
            questionario idAQuest
        DALRuntime.iNumDomande = DALRuntime.getNumDomande
            (DALRuntime.IdAQuest);
        DALRuntime.setRecPaziente(Convert.ToInt32(
            lstPazienti.SelectedItem.Value.ToString()));
        // Salva in sessione elenco delle domande del
            questionario
        if (DALRuntime.iDomanda == 0)
            DALRuntime.tblDomande = DALRuntime.
                getQuestItems(DALRuntime.IdAQuest);
        // Crea testata in m_quest per il nuovo
            questionario che sta per essere compilato
        // e salva in sessione il nuovo id
        // TODO: se possibile non salvare questionario se
            non alla fine (si potrebbe tenere tutto in
```

```

        sessione)
DALRuntime.IdMQuest = DALRuntime.insertMQuest(
    DALRuntime.IdAQuest, DALRuntime.RecPaziente.
    idPaziente, DALRuntime.IdUtente);
// Prepara la prima domanda
fillQuestion(DALRuntime.iDomanda);
// Nasconde panel iniziale
pnlInizio.Visible = false;
// Mostra panel per le domande
pnlDomanda.Visible = true;
    }
} /* btnAvanti1_Click */

///

```

```

DALRuntime.insertMQuestItem (DALRuntime.IdMQuest ,
    getCurrIdAQuestItem (DALRuntime.iDomanda) ,
    rblRisposta.SelectedIndex);
// Passa alla domanda successiva
incDomanda();
if (DALRuntime.iDomanda < DALRuntime.tblDomande.
    Rows.Count)
{
    fillQuestion (DALRuntime.iDomanda);
    if (DALRuntime.iDomanda == DALRuntime.
        tblDomande.Rows.Count - 1)
        btnAvanti2.Text = "Fine";
}
else
{
    showPanelCompletato(true);
    string sRisultato = DALRuntime.calcolaRisultato
        (DALRuntime.IdMQuest);
    DALRuntime.calcolaLettera (DALRuntime.IdMQuest ,
        DALRuntime.IdUtente);
    lblDomanda.Text = "Questionario_completato._
        Risultato:_" + sRisultato;
}
}
} /* btnAvanti2_Click */

private string getFieldDomanda(string sFieldName , int
    iDomanda)
{
    return DALRuntime.tblDomande.Rows[iDomanda][
        sFieldName].ToString();
}

```

```

}

///<summary>
///Compila i controlli della pagina con i dati della
  domanda iDomand-esima
///</summary>
///<param name="iDomanda"></param>
private void fillQuestion(int iDomanda)
{
  // Compila dati paziente
  lblPaziente.Text = "Paziente_" + DALRuntime.
    RecPaziente.getSaluto();
  imgPaziente.ImageUrl = DALRuntime.
    fImagePazientiPath + DALRuntime.RecPaziente.
    ddFoto;
  pnlDomanda.HeaderText = lstQuestionari.SelectedItem
    .Text;
  // Dati sezione
  lblIntesta1.Text = getFieldDomanda("
    dd_sezione_descr1", iDomanda);
  lblIntesta2.Text = getFieldDomanda("
    dd_sezione_descr2", iDomanda);
  // Nasconde lblAvviso
  lblAvviso.Visible = false;
  // Imposta testo domanda
  lblDomanda.Text = DALRuntime.tblDomande.Rows[
    iDomanda][ "DDDOMANDA" ].ToString();
  lblNumDomanda.Text = "Domanda_n_" + (iDomanda + 1)
    .ToString() + "_di_" + DALRuntime.iNumDomande.
    ToString();
  // Imposta immagine associata alla domanda iDomanda

```

```
setImmagine(iDomanda);
string sRisposte = DALRuntime.tblDomande.Rows[
    iDomanda][ "DD_RISPOSTE" ].ToString();
string [] arrRisposte = sRisposte.Split( '|' );
rblRisposta.Items.Clear();
for (int i = 0; i < arrRisposte.Length; i++)
    rblRisposta.Items.Add(arrRisposte[i]);
// Forza stato di risposta non selezionata (
// altrimenti resta da domanda precedente)
rblRisposta.SelectedIndex = -1;
// Legge nr scelte della sezione corrente e
int iNumScelte = Convert.ToInt32(getFieldDomanda("
    nr_num_scelte_possibili", iDomanda));
// Imposta repeatcolumns di conseguenza
rblRisposta.RepeatColumns = iNumScelte;
}

private void setImmagine(int iDomanda)
{
    string sImmagineDb = DALRuntime.tblDomande.Rows[
        iDomanda][ "DD_IMMAGINE" ].ToString();
    if (sImmagineDb != string.Empty)
        imgDomanda.ImageUrl = DALRuntime.
            fImageQuestionariPath + sImmagineDb;
    else
        imgDomanda.ImageUrl = DALRuntime.
            fImageQuestionariPath + DALRuntime.
            fImageNotPresent;
} /* setImmagine */

protected void setAvviso1(string sMsg)
```

```
{
    lblAvviso1.Text = sMsg;
    lblAvviso1.Visible = sMsg != string.Empty;
}

private void showPanelCompletato(bool isFinish)
{
    // Valutare se il caso di modificare le istruzioni
    // per l'ultima pagina
    // (tasto lettera ecc..). Per ora metto
    // semplicemente invisibile
    lblIstruzioni.Visible = !isFinish;
    imgDomanda.Visible = !isFinish;
    rblRisposta.Visible = !isFinish;
    btnAvanti2.Visible = !isFinish;
    btnLettera.Visible = isFinish;
    btnNuovo.Visible = isFinish;
    lblNumDomanda.Visible = !isFinish;
    lblIntesta1.Visible = !isFinish;
    lblIntesta2.Visible = !isFinish;
    lblAvviso.Visible = !isFinish;
}

protected int getCurrIdAQuestItem(int iDomanda)
{
    return Convert.ToInt32(DALRuntime.tblDomande.Rows[
        iDomanda][ "ID" ].ToString());
}

private static void incDomanda()
{
```

```

        DALRuntime.iDomanda = DALRuntime.iDomanda + 1;
    }

    protected void btnLettera_Click(object sender,
        EventArgs e)
    {
        Response.Redirect(
            string.Format("~/Questionari/FmLettera.aspx?" +
                "IdMQuest={0}", DALRuntime.IdMQuest.ToString()));
    }

    protected void btnNuovo_Click(object sender,
        EventArgs e)
    {
        resetData();
        Response.Redirect("~/Questionari/FmCompila.aspx");
    }
}

```

Listing 4.25: FmGrafico.aspx

```

<%@ Page Language="C#" AutoEventWireup="true"
    CodeFile="FmGrafico.aspx.cs" Inherits="
    Questionari_FmGrafico" %>

<%@ Register Assembly="DevExpress.Web.ASPxEditors.v9.3,
    _Version=9.3.6.0, _Culture=neutral, _PublicKeyToken=
    b88d1754d700e49a"
    Namespace="DevExpress.Web.ASPxEditors" TagPrefix="dx"
    %>

<%@ Register Assembly="System.Web.DataVisualization, _
    Version=3.5.0.0, _Culture=neutral, _PublicKeyToken=31

```

```

        bf3856ad364e35"
        Namespace="System.Web.UI.DataVisualization.Charting"
        TagPrefix="asp" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title ></title >
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Chart ID="chartTest" runat="server" Height="
            500px" Width="750px">
            <Series >
                <asp:Series Name="Series1">
                    </asp:Series >
            </Series >
            <ChartAreas>
                <asp:ChartArea Name="ChartArea1">
                    </asp:ChartArea>
            </ChartAreas>
        </asp:Chart>
        <dx:ASPxCheckBox ID="chkShow3D" runat="server"
            AutoPostBack="True"
            Text="Abilita 3D">
        </dx:ASPxCheckBox>
    </div>

```



```
</form>
</body>
</html>
```

Listing 4.26: FmGrafico.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Questionari_FmGrafico : System.Web
    .UI.Page
{
    protected void doAddSerie(int iNum)
    {
        string sNomeSerie = " Risultato_§sezione_" + iNum.
            ToString();
        chartTest.Series.Add(sNomeSerie);
        chartTest.Series[sNomeSerie].XValueMember = "Data";
        // chartTest.Series[sNomeSerie].YValueType = System
            .Web.UI.DataVisualization.Charting.ChartValueType
            .Date;
        chartTest.Series[sNomeSerie].YValueMembers = "
            nr_risultato" + iNum.ToString();
        chartTest.Series[sNomeSerie].ChartType = System.Web
            .UI.DataVisualization.Charting.SeriesChartType.
            Line;
        chartTest.Series[sNomeSerie].IsValueShownAsLabel =
            true;
        switch (iNum)
```

```
{
    case 1:
        chartTest.Series[sNomeSerie].MarkerStyle =
            System.Web.UI.DataVisualization.Charting.
                MarkerStyle.Circle;
        break;
    case 2:
        chartTest.Series[sNomeSerie].MarkerStyle =
            System.Web.UI.DataVisualization.Charting.
                MarkerStyle.Cross;
        break;
    case 3:
        chartTest.Series[sNomeSerie].MarkerStyle =
            System.Web.UI.DataVisualization.Charting.
                MarkerStyle.Diamond;
        break;
    default:
        break;
}
}

protected void Page_Load(object sender, EventArgs e)
{
    // if (!IsPostBack) {
    //     Se Ãˆ stato passato IdMQuest come parametro
    //     ...
    if (Request.Params["IdMQuest"] != null && Request
        .Params["IdMQuest"].ToString() != string.Empty)
    {
        int idMQuest = Convert.ToInt32(Request.Params["
            IdMQuest"].ToString());
    }
}
```

```
DALRuntime.setRecMQuest(idMQuest);
DALRuntime.setRecPaziente(DALRuntime.RecMQuest.
    CaPaziente);
chartTest.Titles.Add(DALRuntime.RecMQuest.
    DdQuestionario);
chartTest.Titles.Add(DALRuntime.RecPaziente.
    getSaluto());
chartTest.ChartAreas[0].AxisX.MajorGrid.Enabled
    = true;
chartTest.ChartAreas[0].AxisX.MinorGrid.Enabled
    = true;
chartTest.ChartAreas[0].AxisY.Minimum = 0;
chartTest.ChartAreas[0].AxisY.Maximum = 10;
chartTest.Series.Clear();
doAddSerie(1);
doAddSerie(2);
doAddSerie(3);
chartTest.Legends.Clear();
chartTest.Legends.Add(" Grafico");
chartTest.Legends[" Grafico"].Title = " Risultati
    _test_2010";
chartTest.Legends[" Grafico"].Docking = System.
    Web.UI.DataVisualization.Charting.Docking.
    Bottom;
// abilita 3d
chartTest.ChartAreas[0].Area3DStyle.Enable3D =
    chkShow3D.Checked;

chartTest.DataSource = DALRuntime.
    getRisultatiPaziente(idMQuest);
chartTest.DataBind();
```

```

    }
  }
  // } // if !IsPostBack
}

```

Listing 4.27: FmLettera.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/
  MasterEmma.master" AutoEventWireup="true" CodeFile="
  FmLettera.aspx.cs" Inherits="Questionari_FmLettera"
  %>

<%@ Register Assembly="DevExpress.Web.ASPxEditors.v9.3,
  _Version=9.3.6.0, _Culture=neutral, _PublicKeyToken=
  b88d1754d700e49a"
  Namespace="DevExpress.Web.ASPxEditors" TagPrefix="dx"
  %>

<asp:Content ID="Content1" ContentPlaceHolderID="
  HeadContent" Runat="Server">
  <style type="text/css">
    #TextArea1
    {
      height: 233px;
    }
  </style>
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="
  MainContent" Runat="Server">

<table>
<tr>
<td rowspan="4">

```

```
<table>
<tr>
<td><dx:ASPxlabel ID="lblIstruz1" runat="server" Text="
  Lettera_al_medico_curante" Font-Size="10px"></dx:
  ASPxlabel></td>
</tr>
<tr>
<td><dx:ASPxlabel ID="lblIstruz2" runat="server" Text="
  Funzioni_disponibili:" Font-Size="10px"></dx:
  ASPxlabel></td>
</tr>
<tr>
<td><dx:ASPxlabel ID="lblIstruz3" runat="server" Text="
  o_Modificare_e_salvare_il_testo_con_il_pulsante_Salva
  " Font-Size="10px"></dx:ASPxlabel></td>
</tr>
<tr>
<td><dx:ASPxlabel ID="lblIstruz4" runat="server" Text="
  o_Spedirla_come_mail_con_il_pulsante_Invia_mail" Font
  -Size="10px"></dx:ASPxlabel></td>
</tr>
<tr>
<td><dx:ASPxlabel ID="lblIstruz5" runat="server" Text="
  o_Salvarla_in_PDF_con_il_pulsante_Genera_PDF" Font-
  Size="10px"></dx:ASPxlabel></td>
</tr>
</table>
</td>
<td>
  <dx:ASPxlabel ID="lblAvviso" runat="server"
```

```
        Text="eventuale_messaggio_di_errore" ForeColor="Red
        " Visible="False" Font-Bold="true" Font-Size="14
        px">
    </dx: ASPxLabel>
</td>
</tr>
<tr>
<td>
    <dx: ASPxLabel ID="lblLettera" runat="server" Text="
        Lettera_per_il_medico_curante_del_paziente:" Font-
        Bold="true" Font-Size="14px">
    </dx: ASPxLabel>
</td>
</tr>
<tr>
<td>
    <dx: ASPxMemo ID="memLettera" runat="server" Height="
        400px" Width="700px">
    </dx: ASPxMemo>
</td>
</tr>
<tr>
<td>
    <table>
    <tr>
    <td>
        <dx: ASPxButton ID="btnGeneraPDF" runat="server"
            Text="Genera_PDF" ToolTip="Genera_una_copia_in_
            pdf_della_lettera"
            onclick="btnGeneraPDF_Click">
        </dx: ASPxButton>
```

```
</td>
<td>
  <dx:ASPxBUTTON ID="btnSendMail" runat="server" Text
    ="Invia_mail" ToolTip="Spedisce_la_lettera_per_
    email_al_medico_curante"
    onclick="btnSendMail_Click">
  </dx:ASPxBUTTON>
</td>
<td>
  <dx:ASPxBUTTON ID="btnSalvaLettera" runat="server"
    Text="Salva" ToolTip="Salva_la_lettera_nel_
    database"
    onclick="btnSalvaLettera_Click">
  </dx:ASPxBUTTON>
</td>
<td>
  <dx:ASPxBUTTON ID="btnTestata" runat="server" Text=
    "Intestazione"
    ToolTip="Modifica_l'intestazione_della_lettera"
    onclick="btnTestata_Click"
    >
  </dx:ASPxBUTTON>
</td>
</tr>
</table>
</td>
</tr>
</table>
</asp:Content>
```

Listing 4.28: FmLettera.aspx.cs

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using iTextSharp.text;
using iTextSharp.text.pdf;
using System.Net.Mail;

public partial class Questionari_FmLettera : System.Web
    .UI.Page
{
    /// TODO: aggiungere controllo client side se lettera
        cambiata e non salvata

    protected void setAvviso(string sAvviso)
    {
        lblAvviso.Visible = sAvviso != string.Empty;
        lblAvviso.Text = sAvviso;
    }

    /// <summary>
    /// Costruisce e spedisce PDF con lettera corrente
    /// </summary>
    protected void buildPDF()
    {
        string fileName = "Visita_" + DALRuntime.
            getNomePazienteConSaluto(DALRuntime.IdMQuest) +
            ".pdf";
        fileName = fileName.Replace(' ', '_');
        Document pdfDoc = new Document();
    }
}
```



```
this.Response.Clear();
this.Response.AddHeader("content-disposition",
    string.Format("attachment;filename={0}", System.
        IO.Path.GetFileName(fileName)));
this.Response.ContentType = "application/pdf";

PdfWriter.GetInstance(pdfDoc, Response.OutputStream
    );
pdfDoc.Open();
// Aggiunge l'intestazione dell'utente
string sIntestazione = DALRuntime.getTestata();
// Definisce un font piu' piccolo
BaseFont bfTimes = BaseFont.CreateFont(BaseFont.
    TIMES_ROMAN, BaseFont.CP1252, true);
Font ftTimes10Italic = new Font(bfTimes, 10, Font.
    ITALIC);
Paragraph pIntestazione = new Paragraph(
    sIntestazione, ftTimes10Italic);
pdfDoc.Add(pIntestazione);
Font ftTimes10Normal = new Font(bfTimes, 10, Font.
    NORMAL);
string sLettera = memLettera.Text;
Paragraph pBody = new Paragraph(sLettera,
    ftTimes10Normal);
pdfDoc.Add(pBody);
pdfDoc.Close();
this.Response.Flush();
this.Response.End();
} /* buildPDF() */
```

```
/// <summary>
/// Spedisce la lettera corrente via mail al medico
/// di famiglia del paziente
/// </summary>
protected void sendMail()
{
    string sMailMedico = DALRuntime.
        getMailAddressMedico(DALRuntime.IdMQuest);
    string sSubject = "Visita_" + DALRuntime.
        getNomePazienteConSaluto(DALRuntime.IdMQuest);
    string sFrom = "emma@kotik.it";
    MailMessage mmMessage = new MailMessage(sFrom,
        sMailMedico);
    // Cerca di mettere l'intestazione dell'utente come
    // una specie di signature
    mmMessage.Body = memLettera.Text + "\n--\n" +
        DALRuntime.getTestata();
    mmMessage.Subject = sSubject;
    SmtplibClient smtpClient = new SmtplibClient();
    smtpClient.Send(mmMessage);
} /* sendMail() */

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        // Se Ã stato passato IdMQuest come parametro
        ...
        if (Request.Params["IdMQuest"] != null && Request.
            Params["IdMQuest"].ToString() != string.Empty)
        {
```

```
DALRuntime.IdMQuest = Convert.ToInt32(Request.
    Params["IdMQuest"].ToString());
// .. legge lettera associata a IdMquest e ...
string sLettera = DALRuntime.getLettera(
    DALRuntime.IdMQuest);
// .. la mette in memLettera
memLettera.Text = sLettera;

// Controlla che sia compilata la mail del
    medico
string sMailMedico = DALRuntime.
    getMailAddressMedico(DALRuntime.IdMQuest);

if (sMailMedico == string.Empty)
{
    btnSendMail.Enabled = false;
    setAvviso("Manca la mail del medico:
        impossibile spedire la lettera via mail");
}

// Compose lblLettera: "lettera al medico
    curante (mail) del paziente: sig ..."
lblLettera.Text = "Lettera al medico curante ("
    + sMailMedico + ") del paziente " +
    DALRuntime.getNomePazienteConSaluto(
    DALRuntime.IdMQuest);
}
}
}

///summary>
```

```
/// Salva il contenuto di memLettera in M_QUEST.  
/// </summary>  
/// <param name="sender"></param>  
/// <param name="e"></param>  
protected void btnSalvaLettera_Click(object sender ,  
    EventArgs e)  
{  
    DALRuntime.saveLettera(DALRuntime.IdMQuest ,  
        memLettera.Text);  
    setAvviso("Lettera_salvata");  
}  
  
protected void btnGeneraPDF_Click(object sender ,  
    EventArgs e)  
{  
    buildPDF();  
}  
  
protected void btnSendMail_Click(object sender ,  
    EventArgs e)  
{  
    try  
    {  
        sendMail();  
        setAvviso("Mail_spedita!");  
    }  
    catch (Exception ex)  
    {  
        setAvviso("Si_Ã_verificato_l'errore:_ " + ex.  
            Message);  
    }  
}
```

```

}

protected void btnTestata_Click(object sender,
    EventArgs e)
{
    Response.Redirect("~/Questionari/FmTestataLettera.
        asp");
}
}

```

Listing 4.29: FmTestataLettera.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/
    MasterEmma.master" AutoEventWireup="true" CodeFile="
    FmTestataLettera.aspx.cs" Inherits="
    Questionari_FmTestataLettera" %>

<%@ Register Assembly="DevExpress.Web.ASPxEditors.v9.3,
    _Version=9.3.6.0, _Culture=neutral, _PublicKeyToken=
    b88d1754d700e49a"
    Namespace="DevExpress.Web.ASPxEditors" TagPrefix="dx"
    %>

<asp:Content ID="Content1" ContentPlaceHolderID="
    HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="
    MainContent" Runat="Server">
<div style="padding-top: 20px; padding-left: 20px;">
    <dx:ASPMemo ID="memTestata" runat="server" Height="
        200px" Width="800px">
    </dx:ASPMemo>
</div>
<table>

```

```

<tr>
<td>
<dx:ASPxBUTTON ID="btnSalva" runat="server" Text="
    Salva" onclick="btnSalva_Click" ToolTip="Salva_l'
    intestazione">
</dx:ASPxBUTTON>
</td>
<td>
<dx:ASPxBUTTON ID="btnIndietro" runat="server" Text="
    Indietro" ToolTip="Torna_àlla_àlettera"
    onclick="btnIndietro_Click">
</dx:ASPxBUTTON>
</td>
</tr>
</table>
</div>
</asp:Content>

```

Listing 4.30: FmTestataLettera.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Questionari_FmTestataLettera :
    System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)

```

```

        {
            memTestata.Text = DALRuntime.getTestata();
        }
    }

    protected void btnSalva_Click(object sender,
        EventArgs e)
    {
        DALRuntime.setTestata(DALRuntime.IdUtente,
            memTestata.Text);
    }

    protected void btnIndietro_Click(object sender,
        EventArgs e)
    {
        Response.Redirect(string.Format("~/Questionari/
            FmLettera.aspx?IdMQuest={0}", DALRuntime.IdMQuest
            ));
    }
}

```

Listing 4.31: FmStruttura.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/
    MasterEmma.master" AutoEventWireup="true" CodeFile="
    FmStruttura.aspx.cs" Inherits="
    Questionari_FmStruttura" %>

<%@ Register Assembly="DevExpress.Web.ASPxGridView.v9
    .3, Version=9.3.6.0, Culture=neutral, PublicKeyToken=
    b88d1754d700e49a"
    Namespace="DevExpress.Web.ASPxGridView" TagPrefix="dx
    " %>

```

```

<%@ Register assembly="DevExpress.Web.ASPxEditors.v9.3,
    _Version=9.3.6.0, _Culture=neutral, _PublicKeyToken=
    b88d1754d700e49a" namespace="DevExpress.Web.
    ASPxEditors" tagprefix="dx" %>

<asp:Content ID="Content1" ContentPlaceHolderID="
    HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="
    MainContent" Runat="Server">
    <dx:ASPxGridView ID="grdAQuest" runat="server"
        AutoGenerateColumns="False"
        KeyFieldName="id" onrowdeleting="
            grdAQuest_RowDeleting"
        onrowinserting="grdAQuest_RowInserting"
        onrowupdating="grdAQuest_RowUpdating">
        <SettingsEditing Mode="EditFormAndDisplayRow" />
        <SettingsBehavior ConfirmDelete="True" />
        <SettingsText ConfirmDelete="Confermi la
            cancellazione?" />
    <Columns>
        <dx:GridViewCommandColumn VisibleIndex="0">
            <EditButton Visible="True">
            </EditButton>
            <NewButton Visible="True">
            </NewButton>
            <DeleteButton Visible="True">
            </DeleteButton>
        </dx:GridViewCommandColumn>

```



```
<dx:GridViewDataTextColumn Caption=" Codice"
  FileName=" id" VisibleIndex=" 1">
  <EditFormSettings Visible=" False" />
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption=" Descrizione"
  FileName=" dd_descrizione"
  VisibleIndex=" 2">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption=" Descrizione _
  breve"
  FileName=" dd_descr_breve" VisibleIndex=" 3">
</dx:GridViewDataTextColumn>
</Columns>
<Templates>
<DetailRow>
  <dx:ASPxGridView ID=" grdAQuestSezioni" runat="
  server"
  AutoGenerateColumns=" False"
  KeyFieldName=" id"
  ondatabinding=" grdAQuestSezioni_DataBinding"
  onrowdeleting=" grdAQuestSezioni_RowDeleting"
  onrowinserting=" grdAQuestSezioni_RowInserting"
  onrowupdating=" grdAQuestSezioni_RowUpdating">
  <SettingsEditing Mode=" EditFormAndDisplayRow"
  />
  <SettingsBehavior ConfirmDelete=" True" />
  <SettingsText ConfirmDelete=" Confermi_la_
  cancellazione_?" />
  <Templates>
  <DetailRow>
```

```

<dx:ASPxGridView ID="grdAQuestItem" runat="
server"
AutoGenerateColumns="False"
KeyFieldName="id"
ondatabinding="grdAQuestItem_DataBinding"
onrowdeleting="grdAQuestItem_RowDeleting"
onrowinserting="
grdAQuestItem_RowInserting"
onrowupdating="grdAQuestItem_RowUpdating"
onhtmlrowcreated="
grdAQuestItem_HtmlRowCreated" >
<SettingsEditing Mode="
EditFormAndDisplayRow" />
<SettingsDetail IsDetailGrid="True" />
<SettingsBehavior ConfirmDelete="True" />
<SettingsText ConfirmDelete="Confermi la
cancellazione?" />
<Templates>
<EditForm>
<div style="padding:4px_4px_3px_4px">
<table>
<tr>
<td rowspan="6">
<dx:ASPxImage ID="imgDomanda" runat="server"
">
</dx:ASPxImage>
</td>
<td>
<dx:ASPxLabel ID="ASPxLabel1" runat="server"
Text="Domanda">
</dx:ASPxLabel>

```

```
</td>
</tr>
<tr>
<td>
  <dx:ASPxTextBox ID="edtDdDomanda" runat="server"
    " Width="600px" Text='<%# Bind("dd_domanda")
    %>'>
  </dx:ASPxTextBox>
</td>
</tr>
<tr>
<td>
  <dx:ASPxLabel ID="ASPxLabel2" runat="server"
    Text="Risposte">
  </dx:ASPxLabel>
</td>
</tr>
<tr>
<td>
  <dx:ASPxTextBox ID="edtRisposte" runat="server"
    Width="600px" Text='<%# Bind("dd_risposte")
    %>'>
  </dx:ASPxTextBox>
</td>
</tr>
<tr>
<td>
  <dx:ASPxLabel ID="ASPxLabel3" runat="server"
    Text="Num. _ordine">
  </dx:ASPxLabel>
```

```

        <dx:ASPxTextBox ID="edtNrOrder" runat="server"
            Width="50px" Text='<%# Bind("nr_order") %>'>
        </dx:ASPxTextBox>
    </td>
</tr>
</table>
</div>
<div style="text-align:right; padding:2px 2px 2px 2px">
    <dx:ASPxGridViewTemplateReplacement ID="
        UpdateButton" ReplacementType="
        EditFormUpdateButton" runat="server"></dx:
        ASPxGridViewTemplateReplacement>
    <dx:ASPxGridViewTemplateReplacement ID="
        CancelButton" ReplacementType="
        EditFormCancelButton" runat="server"></dx:
        ASPxGridViewTemplateReplacement>
</div>
</EditForm>
</Templates>
    <Columns>
        <dx:GridViewCommandColumn VisibleIndex="
            0">
            <EditButton Visible="True">
            </EditButton>
            <NewButton Visible="True">
            </NewButton>
            <DeleteButton Visible="True">
            </DeleteButton>
        </dx:GridViewCommandColumn>

```

```
<dx:GridViewDataTextColumn Caption="
    Codice" FieldName="id"
    VisibleIndex="1" EditFormSettings-
    Visible="False">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption="
    Domanda" FieldName="dd_domanda"
    VisibleIndex="2">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption="
    Risposte" FieldName="dd_risposte"
    VisibleIndex="3">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption="
    Immagine" FieldName="dd_immagine"
    VisibleIndex="4">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption="
    Posizione"
    FieldName="nr_order" VisibleIndex="5"
    >
</dx:GridViewDataTextColumn>
</Columns>
</dx:ASPxGridView>
</DetailRow>
</Templates>
<Columns>
<dx:GridViewCommandColumn VisibleIndex="0">
    <EditButton Visible="True">
</EditButton>
    <NewButton Visible="True">
```

```
</NewButton>
<DeleteButton Visible="True">
</DeleteButton>
</dx:GridViewCommandColumn>
<dx:GridViewDataTextColumn Caption="Codice"
  FileName="id" VisibleIndex="1"
  EditFormSettings-Visible="False" >
  <EditFormSettings Visible="False" />
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption="
  Descrizione_breve" FileName="
  dd_descrizione1"
  VisibleIndex="2">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption="
  Descrizione_completa" FileName="
  dd_descrizione2"
  VisibleIndex="3">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption="#_scelte"
  FileName="nr_num_scelte_possibili"
  VisibleIndex="4">
</dx:GridViewDataTextColumn>
</Columns>
<SettingsDetail IsDetailGrid="True"
  ShowDetailRow="True" />
</dx:ASPxGridView>
</DetailRow>
</Templates>
<SettingsDetail ShowDetailRow="True" />
</dx:ASPxGridView>
```

```
</asp:Content>
```

Listing 4.32: FmStruttura.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using DevExpress.Web.ASPxGridView;
using DevExpress.Web.Data;
using DevExpress.Web.ASPxEditors;

public partial class Questionari_FmStruttura : System.
    Web.UI.Page
{
    protected void grdDataBind(bool setDatasource, bool
        doDatabind)
    {
        if (setDatasource)
            grdAQuest.DataSource = DALRuntime.getAQuest();
        if (doDatabind)
            grdAQuest.DataBind();
    }

    protected void Page_Init(object sender, EventArgs e)
    {
        grdDataBind(true, false);
    }

    protected void Page_Load(object sender, EventArgs e)
    {
```

```
        if (!IsPostBack && !IsCallback)
            grdDataBind(false , true);
    }

protected void grdAQuestSezioni_DataBinding(object
    sender , EventArgs e)
{
    ASPxGridView grdSezioni = (sender as ASPxGridView);
    DALRuntime.IdAQuest = Convert.ToInt32(grdSezioni.
        GetMasterRowKeyValue().ToString());
    grdSezioni.DataSource = DALRuntime.getAQuestSezione
        ();
}

protected void grdAQuest_RowDeleting(object sender ,
    ASPxDataDeletingEventArgs e)
{
    int id = Convert.ToInt32(e.Keys[0].ToString());
    DALRuntime.deleteAQuest(id);
    e.Cancel = true;
    grdDataBind(true , true);
}

protected void grdAQuest_RowUpdating(object sender ,
    ASPxDataUpdatingEventArgs e)
{
    int id = Convert.ToInt32(e.Keys[0].ToString());
    string ddDescr = e.NewValues["dd_descrizione"].
        ToString();
    string ddDescrBreve = e.NewValues["dd_descr_breve"]
        ].ToString();
```



```
DALRuntime.updateAQuest(id , ddDescr , ddDescrBreve);
e.Cancel = true;
grdAQuest.CancelEdit();
grdDataBind(true , true);
} /* grdAQuest_RowUpdating */

protected void grdAQuest_RowInserting(object sender ,
ASPxDATAInsertingEventArgs e)
{
string ddDescr = e.NewValues["dd_descrizione"].
ToString();
string ddDescrBreve = e.NewValues["dd_descr_breve"
].ToString();
DALRuntime.insertAQuest(ddDescr , ddDescrBreve);
e.Cancel = true;
grdAQuest.CancelEdit();
grdDataBind(true , true);
} /* grdAQuest_RowInserting */

#region A_QUEST_SEZIONE
protected void grdAQuestSezioni_RowDeleting(object
sender , ASPxDATADeletingEventArgs e)
{
int id = Convert.ToInt32(e.Keys[0].ToString());
DALRuntime.deleteAQuestSezione(id);
e.Cancel = true;
//   grdDataBind(true , true);
}

protected void grdAQuestSezioni_RowInserting(object
sender , ASPxDATAInsertingEventArgs e)
```

```
{
    ASPxGridView grdAQuestSezione = sender as
        ASPxGridView;
    string ddDescr1 = e.NewValues["dd_descrizione1"].
        ToString();
    string ddDescr2 = e.NewValues["dd_descrizione2"].
        ToString();
    int caQuest = DALRuntime.IdAQuest;
    int nrNumScelte = Convert.ToInt32(e.NewValues["
        nr_num_scelte_possibili"].ToString());
    DALRuntime.insertAQuestSezione(ddDescr1, ddDescr2,
        caQuest, nrNumScelte);
    e.Cancel = true;
    grdAQuestSezione.CancelEdit();
}
```

```
protected void grdAQuestSezioni_RowUpdating(object
    sender, ASPxDataUpdatingEventArgs e)
{
    int id = Convert.ToInt32(e.Keys[0].ToString());
    ASPxGridView grdAQuestSezione = sender as
        ASPxGridView;
    string ddDescr1 = e.NewValues["dd_descrizione1"].
        ToString();
    string ddDescr2 = e.NewValues["dd_descrizione2"].
        ToString();
    int caQuest = DALRuntime.IdAQuest;
    int nrNumScelte = Convert.ToInt32(e.NewValues["
        nr_num_scelte_possibili"].ToString());
    DALRuntime.updateAQuestSezione(id, ddDescr1,
        ddDescr2, caQuest, nrNumScelte);
}
```

```
e.Cancel = true;
grdAQuestSezione.CancelEdit();
}
#endregion A_QUEST_SEZIONE

#region A_QUEST_ITEM
protected void grdAQuestItem_DataBinding(object
sender, EventArgs e)
{
    ASPxGridView grdQuestItem = (sender as ASPxGridView
    );
    DALRuntime.IdAQuestSezione = Convert.ToInt32(
    grdQuestItem.GetMasterRowKeyValue().ToString());
    DALRuntime.tblAQuestItems = DALRuntime.
    getAQuestItemsSezione(DALRuntime.IdAQuest,
    DALRuntime.IdAQuestSezione);
    grdQuestItem.DataSource = DALRuntime.tblAQuestItems
    ;
}

protected void grdAQuestItem_RowDeleting(object
sender, ASPxDataDeletingEventArgs e)
{
    int id = Convert.ToInt32(e.Keys[0].ToString());
    DALRuntime.TRecQuestItem recQuestItem = new
    DALRuntime.TRecQuestItem();
    recQuestItem.deleteAQuestItem(id);
    e.Cancel = true;
}
```

```
protected void grdAQuestItem_RowInserting(object
sender, ASPxDataInsertingEventArgs e)
{
    ASPxGridView grdAQuestItem = sender as ASPxGridView
    ;
    DALRuntime.TRecQuestItem recQuestItem = new
    DALRuntime.TRecQuestItem();
    recQuestItem.ddDomanda = e.NewValues["dd_domanda"].
    ToString();
    recQuestItem.ddRisposte = e.NewValues["dd_risposte"]
    .ToString();
    if (e.NewValues["dd_immagine"] != null)
        recQuestItem.ddImmagine = e.NewValues["
        dd_immagine"].ToString();
    if (e.NewValues["nr_order"] != null)
        recQuestItem.nrOrder = Convert.ToInt32(e.
        NewValues["nr_order"].ToString());
    recQuestItem.caQuestSezione = DALRuntime.
    IdAQuestSezione;
    recQuestItem.caQuest = DALRuntime.IdAQuest;
    recQuestItem.insertAQuestItem();
    e.Cancel = true;
    grdAQuestItem.CancelEdit();
}
```

```
protected void grdAQuestItem_RowUpdating(object
sender, ASPxDataUpdatingEventArgs e)
{
    ASPxGridView grdAQuestItem = sender as ASPxGridView
    ;
```

```
DALRuntime.TRecQuestItem recQuestItem = new
    DALRuntime.TRecQuestItem();
recQuestItem.idAQuestItem = Convert.ToInt32(e.Keys
    [0].ToString());
recQuestItem.ddDomanda = e.NewValues["dd_domanda"].
    ToString();
recQuestItem.ddRisposte = e.NewValues["dd_risposte"
    ].ToString();
if (e.NewValues["dd_immagine"] != null)
    recQuestItem.ddImmagine = e.NewValues["
    dd_immagine"].ToString();
if (e.NewValues["nr_order"] != null)
    recQuestItem.nrOrder = Convert.ToInt32(e.
    NewValues["nr_order"].ToString());
recQuestItem.caQuestSezione = DALRuntime.
    IdAQuestSezione;
recQuestItem.caQuest = DALRuntime.IdAQuest;
recQuestItem.updateAQuestItem();
e.Cancel = true;
grdAQuestItem.CancelEdit();
}
```

```
protected void grdAQuestItem_HtmlRowCreated(object
    sender, DevExpress.Web.ASPxGridView.
    ASPxGridViewTableRowEventArgs e)
{
    if (e.RowType == DevExpress.Web.ASPxGridView.
        GridViewRowType.EditForm)
    {
        ASPxGridView grdSender = (sender as ASPxGridView)
            ;
    }
}
```

```

ASPxImage img = (ASPxImage)grdSender.
    FindEditFormTemplateControl("imgDomanda");
System.Data.DataRow r = DALRuntime.tblAQuestItems
    .Rows.Find(e.KeyValue);
if (r != null)
{
    int idAQuestItem = Convert.ToInt32(r["id"].
        ToString());
    string sImage = DALRuntime.
        fImageQuestionariPath;
    if (r["dd.immagine"] == null || r["dd.immagine"]
        .ToString() == string.Empty)
        sImage += DALRuntime.fImageNotPresent;
    else
        sImage = DALRuntime.checkAndGetImagePath(
            DALRuntime.fImageQuestionariPath, r["
            dd.immagine"].ToString());
    img.ImageUrl = sImage;
}
}
}
#endregion A_QUEST_ITEM
}

```

4.2.6 Gestione tabelle

Listing 4.33: FmPazienti.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/
    MasterEmma.master" AutoEventWireup="true" CodeFile="
    FmPazienti.aspx.cs" Inherits="Tabelle_FmPazienti" %>

```

```
<%@ Register Assembly="DevExpress.Web.v9.3, Version=9.3.6.0, Culture=neutral, PublicKeyToken=b88d1754d700e49a"
    Namespace="DevExpress.Web.ASPxUploadControl"
    TagPrefix="dx" %>

<%@ Register Assembly="DevExpress.Web.ASPxGridView.v9.3, Version=9.3.6.0, Culture=neutral, PublicKeyToken=b88d1754d700e49a"
    Namespace="DevExpress.Web.ASPxGridView" TagPrefix="dx" %>

<%@ Register assembly="DevExpress.Web.ASPxEditors.v9.3, Version=9.3.6.0, Culture=neutral, PublicKeyToken=b88d1754d700e49a"
    namespace="DevExpress.Web.ASPxEditors" tagprefix="dx" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
    <dx:ASPxGridView ID="grdPazienti" runat="server"
        AutoGenerateColumns="False" ClientInstanceName="grdPazienti"
        KeyFieldName="ID" Width="800px" onhtmlrowcreated="grdPazienti_HtmlRowCreated"
        onrowupdating="grdPazienti_RowUpdating"
        onrowdeleting="grdPazienti_RowDeleting"
        onrowinserting="grdPazienti_RowInserting">
    <SettingsBehavior ConfirmDelete="True" />
```

```

<SettingsEditing Mode="EditFormAndDisplayRow" />
<SettingsText ConfirmDelete=" Confermi la
    cancellazione?" />
<Columns>
  <dx:GridViewCommandColumn VisibleIndex="0">
    <EditButton Visible="True">
      </EditButton>
    <NewButton Visible="True">
      </NewButton>
    <DeleteButton Visible="True">
      </DeleteButton>
    <ClearFilterButton Visible="True">
      </ClearFilterButton>
  </dx:GridViewCommandColumn>
  <dx:GridViewDataTextColumn Caption="Cognome"
    FieldName="dd_cognome"
    Name="Cognome" VisibleIndex="1">
  </dx:GridViewDataTextColumn>
  <dx:GridViewDataTextColumn Caption="Nome"
    FieldName="dd_nome" VisibleIndex="2">
  </dx:GridViewDataTextColumn>
  <dx:GridViewDataTextColumn Caption="Codice
    fiscale" FieldName="cd_fiscale"
    VisibleIndex="3">
  </dx:GridViewDataTextColumn>
  <dx:GridViewDataDateColumn Caption="Data
    nascita"
    FieldName="dt_nascita"
    VisibleIndex="4">
    <PropertiesDateEdit DisplayFormatString="d"
      DisplayFormatInEditMode="True">
    </PropertiesDateEdit>
  </dx:GridViewDataDateColumn>

```



```

</dx:GridViewDataDateColumn>
<dx:GridViewDataTextColumn Caption=" Titolo_di_
    studio" FieldName=" dd_titolo"
    VisibleIndex="5">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption=" Stato_civile"
    FieldName=" dd_stato_civile"
    VisibleIndex="6">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption=" Nome_medico"
    FieldName=" dd_medico_nome" Visible="false"
    VisibleIndex="7">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption=" Medico"
    FieldName=" dd_medico_cognome"
    VisibleIndex="8">
</dx:GridViewDataTextColumn>
<dx:GridViewDataTextColumn Caption=" Mail_medico"
    FieldName=" dd_medico_email"
    VisibleIndex="9">
</dx:GridViewDataTextColumn>
</Columns>
<Templates>
<EditForm>
<div style="padding:4px_4px_3px_4px">
    <table>
    <tr>
    <td>&nbsp;</td>
    <td colspan="4" align="center" style="border-
        bottom:_solid_1px_black">
Dati anagrafici

```

```

</td>
</tr>
<tr>
  <td rowspan="9">
    <div style="border: 1px solid #C2D4DA; padding: 2px;">
<% if (!grdPazienti.IsNewRowEditing)
  { %>
    <dx:ASPImage ID="imgPaziente" runat="server" Width="225px" Height="225px"></dx:ASPImage>
  <% } %>
    </div>
  </td>
<td>
<table>
  <tr>
    <td>
      Nome</td>
    <td>
      <dx:ASPTextBox ID="edFirst" runat="server"
        Text='<%# Bind("dd_nome") %>'
        Width="150px">
      </dx:ASPTextBox>
    </td>
    <td>
      Cognome</td>
    <td>
      <dx:ASPTextBox ID="edLast" runat="server"
        Text='<%# Bind("dd_cognome") %>'
        Width="150px">

```

```

                </dx:ASPTextBox>
            </td>
        </tr>
</table>
</td>
    </tr>

<tr>
<td>
    <table>
        <tr>
            <td style="white-space:nowrap">Data di nascita</
            td>
            <td style="width:_40px">
                <dx:ASPDateEdit runat="server" ID="edtDtNascita
                " Value='<%# Bind(" dt_nascita") %>' Width="100
                px">
            </dx:ASPDateEdit> </td>
            <td style="white-space:nowrap">Sesso</td>
        <td>
            <dx:ASPComboBox ID="cbxSesso" runat="server"
            ValueType="System.String" Text='<%# Bind("
            cd_sesso") %>' Width="100px">
            <Items>
                <dx:ListEditItem Text="Non specificato"
                Value="" />
                <dx:ListEditItem Text="Femminile" Value="F"
                />
                <dx:ListEditItem Text="Maschile" Value="M"
                />
            </Items>

```

```

</dx:ASPxComboBox>
</td>
<td style="white-space:nowrap">Codice fiscale </
td>
<td><dx:ASPxTextBox runat="server" ID="
edtCdFiscale" Text='<%# Bind("cd_fiscale") %>'
Width="150px">
</dx:ASPxTextBox> </td>
</tr>
</table>
</td>
</tr>

<tr>
<td colspan="4">
<table>
<tr>
<td style="white-space:nowrap">Titolo di
studio</td>
<td>
<dx:ASPxComboBox ID="cbxTitoloStudio" runat=
"server" TextField="dd_descrizione"
ValueField="id" ValueType="System.Int32"
Text='<%# Bind("ct_titolo") %>'>
</dx:ASPxComboBox>
</td>
<td style="white-space:nowrap">Stato civile </
td>
<td>
<dx:ASPxComboBox ID="cbxStatoCivile" runat="
server" TextField="dd_descrizione"

```

```

        ValueField="id" ValueType="System.Int32"
        Text='<%# Bind("ct_stato_civile") %>'
    </dx:ASPxComboBox>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<tr>
<td colspan="4" align="center" style="border-
    bottom: 1px solid black" >Medico curante</td>
</tr>
<tr>
    <td colspan="4">
        <div style="border: 1px solid #C2D4DA;
            padding: 2px;">
            <table>
                <tr>
                    <td>
                        Nome</td>
                    <td>
                        <dx:ASPxTextBox ID="edtNomeMedico"
                            runat="server"
                            Text='<%# Bind("dd_medico_nome")
                                %>' Width="150px">
                        </dx:ASPxTextBox>
                    </td>
                    <td>
                        Cognome</td>
                    <td>

```

```

        <dx:ASPxTextBox ID="edtCognomeMedico
            " runat="server"
            Text='<%# Bind("dd_medico_cognome"
                ) %>' Width="150px">
        </dx:ASPxTextBox>
    </td>
    <td>
        Email</td>
    <td>
        <dx:ASPxTextBox ID="edtMailMedico"
            runat="server"
            Text='<%# Bind("dd_medico_email")
                %>' Width="150px">
        </dx:ASPxTextBox>
    </td>
</tr>
</table>
</div>
</td>
</tr>
<tr>
    <td align="center" colspan="4" style="border-
        bottom: 1px solid black">
        Note
    </td>
</tr>
<tr>
    <td colspan="4">
        <dx:ASPxMemo ID="memNote" runat="server"
            Height="100px"
            Text='<%# Bind("bl_nota")%>' Width="100%">

```

```

        </dx:ASPxMemo>
    </td>
</tr>
</tr>
</table>
<% if (!grdPazienti.IsNewRowEditing)
    { %>
<table>
<tr>
    <td valign="top">Carica foto
    </td>
    <td>
        <dx:ASPxUploadControl ID="uploadImage" runat="
            server" ShowProgressPanel="True" Size="30"
            ShowUploadButton="True"
            onfileuploadcomplete="
                uploadImage_FileUploadComplete"
            ClientInstanceName="uploadImage">
            <ClientSideEvents FileUploadComplete="function
                (s,e){
.....grdPazienti.CancelEdit();
}" />
        </dx:ASPxUploadControl>
    </td>
</tr>
</table>
</table>
    <% } %>
</div>
<div style="text-align:right; padding:2px 2px 2px 2px">

```

```

        <dx: ASPxGridViewTemplateReplacement ID="
            UpdateButton" ReplacementType="
            EditFormUpdateButton" runat="server"></dx:
            ASPxGridViewTemplateReplacement>
        <dx: ASPxGridViewTemplateReplacement ID="
            CancelButton" ReplacementType="
            EditFormCancelButton" runat="server"></dx:
            ASPxGridViewTemplateReplacement>
    </div>
</EditForm>
</Templates>
<Settings ShowFilterRow="True" />
</dx: ASPxGridView>
</asp:Content>

```

Listing 4.34: FmPazienti.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using DevExpress.Web.ASPxEditors;
using DevExpress.Web.ASPxGridView;
using DevExpress.Web.Data;
using System.Collections.Specialized;
using DevExpress.Web.ASPxUploadControl;

// Finito l'upload chiude editform lato client (da js)
// In insert nascosto controllo di upload perch ©, se
// si fa insert di un nuovo record

```



```
// e, contemporaneamente, l'upload di un'immagine il
// post fallisce.
// Per ora se la grid Ã" in insert il controllo di
// upload viene nascosto.
// Upload quindi possibile solo in update

public partial class Tabelle_FmPazienti : System.Web.UI
    .Page
{
    private object getPropValue(OrderedDictionary
        NewValues, OrderedDictionary OldValues, string
        propName)
    {
        return NewValues[propName] != null ? NewValues[
            propName] : OldValues[propName] != null ?
            OldValues[propName] : null;
    }

    private string getPropValueAsString(OrderedDictionary
        NewValues, OrderedDictionary OldValues, string
        propName)
    {
        object obj = getPropValue(NewValues, OldValues,
            propName);
        return obj == null ? string.Empty : obj.ToString();
    }

    /// <summary>
    /// Compila e ritorna recPaziente leggendo da
    /// NewValues o, se null, da OldValues
    /// </summary>
```

```
///<param name="NewValues"></param>
///<param name="OldValues"></param>
///<returns></returns>
private DALRuntime.TRecPaziente getRecPaziente(
    OrderedDictionary NewValues, OrderedDictionary
    OldValues)
{
    DALRuntime.TRecPaziente recPaziente = DALRuntime.
        getRecPaziente(0);
    recPaziente.Cognome = getPropValueAsString(
        NewValues, OldValues, "dd_cognome");
    recPaziente.Nome = getPropValueAsString(NewValues,
        OldValues, "dd_nome");
    recPaziente.CdFiscale = getPropValueAsString(
        NewValues, OldValues, "cd_fiscale");
    System.Globalization.DateTimeFormatInfo dtfi = new
        System.Globalization.DateTimeFormatInfo();
    dtfi.ShortDatePattern = "MM/dd/yyyy";
    string sDtNascita = getPropValueAsString(NewValues,
        OldValues, "dt_nascita");
    if (sDtNascita != string.Empty)
        recPaziente.DataNascita = Convert.ToDateTime(
            sDtNascita, dtfi);
    recPaziente.Sesso = getPropValueAsString(NewValues,
        OldValues, "cd_sesso");
    recPaziente.MedicoCognome = getPropValueAsString(
        NewValues, OldValues, "dd_medico_cognome");
    recPaziente.MedicoNome = getPropValueAsString(
        NewValues, OldValues, "dd_medico_nome");
    recPaziente.MedicoMail = getPropValueAsString(
        NewValues, OldValues, "dd_medico_email");
}
```

```
    recPaziente.Note = getPropValueAsString(NewValues,
        OldValues, "bl_nota");
    return recPaziente;
} /* getRecPaziente */

/// <summary>
/// Legge valori di ct_stato_civile e ct_titolo dai
/// controlli (combobox) e li
/// scrive in recPaziente
/// </summary>
/// <param name="recPaziente"></param>
/// <param name="grdSender"></param>
private void readComboBox(DALRuntime.TRecPaziente
    recPaziente, ASPxGridView grdSender)
{
    ASPxComboBox cbxStatoCivile = (ASPxComboBox)
        grdSender.FindEditFormTemplateControl("
            cbxStatoCivile");
    if (cbxStatoCivile != null && cbxStatoCivile.Value
        != null)
        recPaziente.CtStatoCivile = Convert.ToInt32(
            cbxStatoCivile.Value.ToString());
    ASPxComboBox cbxTitolo = (ASPxComboBox)grdSender.
        FindEditFormTemplateControl("cbxTitoloStudio");
    if (cbxTitolo != null && cbxTitolo.Value != null)
        recPaziente.CtTitolo = Convert.ToInt32(cbxTitolo.
            Value.ToString());
    ASPxComboBox cbxSesso = (ASPxComboBox)grdSender.
        FindEditFormTemplateControl("cbxSesso");
    if (cbxSesso != null && cbxSesso.Value != null)
        recPaziente.Sesso = cbxSesso.Value.ToString();
}
```

```
    } /* readTitoloEStatoCivile */

    protected void grdDataBind(bool setDatasource , bool
        doDatabind)
    {
        if (setDatasource)
        {
            setTblPazienti();
            grdPazienti.DataSource = DALRuntime.tblPazienti;
        }
        if (doDatabind)
        {
            grdPazienti.KeyFieldName = "id";
            grdPazienti.DataBind();
        }
    }

    /*
    protected override void InitializeCulture()
    {
        base.InitializeCulture();
        System.Globalization.CultureInfo culture = new
            System.Globalization.CultureInfo(System.Threading
                .Thread.CurrentThread.CurrentCulture.LCID);
        culture.DateTimeFormat.ShortDatePattern = "dd/MM/
            yyyy";
        System.Threading.Thread.CurrentThread.
            CurrentCulture = culture;
    }
    */
```

```
protected void Page_Init(object sender, EventArgs e)
{
    grdDataBind(true, false);
}

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack && !IsCallback)
    {
        // Reset recPaziente
        DALRuntime.RecPaziente = null;
        grdDataBind(false, true);
    }
}

private static void setTblPazienti()
{
    DALRuntime.tblPazienti = DALRuntime.getPazienti2();
}

protected void grdPazienti_HtmlRowCreated(object
sender, DevExpress.Web.ASPxGridView.
ASPxGridViewTableRowEventArgs e)
{
    if (e.RowType == DevExpress.Web.ASPxGridView.
GridViewRowType.EditForm)
    {
        ASPxGridView grdSender = (sender as ASPxGridView)
;
    }
}
```

```
ASPxImage img = (ASPxImage)grdSender.  
    FindEditFormTemplateControl("imgPaziente");  
System.Data.DataRow r = DALRuntime.tblPazienti.  
    Rows.Find(e.KeyValue);  
if (r != null)  
{  
    int idPaziente = Convert.ToInt32(r["id"].  
        ToString());  
    DALRuntime.setRecPaziente(idPaziente);  
    string sImage = DALRuntime.fImagePazientiPath;  
    if (r["dd_foto"] == null || r["dd_foto"].  
        ToString() == string.Empty)  
        sImage += DALRuntime.fImageNotPresent;  
    else  
        sImage = DALRuntime.checkAndGetImagePath(  
            DALRuntime.fImagePazientiPath, r["dd_foto"]  
                .ToString());  
    img.ImageUrl = sImage;  
}  
  
// Data bind del combo dello stato civile ...  
ASPxComboBox cbxStatoCivile = (ASPxComboBox)  
    grdSender.FindEditFormTemplateControl("cbxStatoCivile");  
if (cbxStatoCivile != null)  
{  
    cbxStatoCivile.DataSource = DALRuntime.  
        getStatoCivile();  
    cbxStatoCivile.DataBindItems();  
    if (r != null)  
        cbxStatoCivile.Value = r["ct_stato_civile"];
```

```
}

// .. e di quello dei titoli di studio ...
ASPxComboBox cbxTitolo = (ASPxComboBox)grdSender.
    FindEditFormTemplateControl("cbxTitoloStudio");
if (cbxTitolo != null)
{
    cbxTitolo.DataSource = DALRuntime.getTitoli();
    cbxTitolo.DataBindItems();
    if (r != null)
        cbxTitolo.Value = r["ct_titolo"];
}

// .. e del sesso
ASPxComboBox cbxSesso = (ASPxComboBox)grdSender.
    FindEditFormTemplateControl("cbxSesso");
if (cbxSesso != null && r != null)
    cbxSesso.Value = r["cd_sesso"];
// Databind delle note
ASPxMemo memNote = (ASPxMemo)grdSender.
    FindEditFormTemplateControl("memNote");
if (memNote != null && r != null)
    memNote.Text = r["bl_nota"].ToString();

/* non c'Ã© la property TempFile ...
ASPxUploadControl uploadImage = (
    ASPxUploadControl)grdSender.
    FindEditFormTemplateControl("uploadImage");
if (uploadImage != null)
    ;
*/
```

```
    }  
  }  
  
  protected void grdPazienti_RowUpdating(object sender ,  
    ASPxDataUpdatingEventArgs e)  
  {  
    int id = Convert.ToInt32(e.Keys[0].ToString());  
    DALRuntime.TRecPaziente recPaziente =  
      getRecPaziente(e.NewValues, e.OldValues);  
    ASPxGridView grdSender = (sender as ASPxGridView);  
    // Legge i valori di ct_titolo e ct_stato_civile e  
      li scrive in recPaziente  
    readComboBox(recPaziente, grdSender);  
    // Imposta id  
    recPaziente.idPaziente = id;  
    recPaziente.updatePaziente();  
    e.Cancel = true;  
    endEditAndRefresh();  
  }  
  
  private void endEditAndRefresh()  
  {  
    grdPazienti.CancelEdit();  
    grdDataBind(true, true);  
  }  
  
  protected void grdPazienti_RowInserting(object sender  
    , ASPxDataInsertingEventArgs e)  
  {  
    DALRuntime.TRecPaziente recPaziente =  
      getRecPaziente(e.NewValues, null);
```



```
ASPxGridView grdSender = (sender as ASPxGridView);  
// Legge i valori di ct_titolo e ct_stato_civile e  
// li scrive in recPaziente  
readComboBox(recPaziente, grdSender);  
recPaziente.insertPaziente();  
e.Cancel = true;  
endEditAndRefresh();  
}
```

```
protected void grdPazienti_RowDeleting(object sender,  
    ASPxDataDeletingEventArgs e)  
{  
    int id = Convert.ToInt32(e.Keys[0].ToString());  
    DALRuntime.deletePazienteById(id);  
    e.Cancel = true;  
    grdDataBind(true, true);  
}
```

```
protected void uploadImage_FileUploadComplete(object  
    sender, DevExpress.Web.ASPxUploadControl.  
    FileUploadCompleteEventArgs e)  
{  
    if (e.IsValid)  
    {  
        string sFilename = e.UploadedFile.FileName;  
        string sPath = DALRuntime.getAbsolutePath(  
            DALRuntime.fImagePazientiPath, sFilename);  
        e.UploadedFile.SaveAs(sPath);  
        e.CallbackData = sFilename;  
        // Salva in sessione.recPaziente il nome file.  
        // SarÃ  poi salvato assieme agli altri campi
```

```

        DALRuntime.TRecPaziente recPaziente = DALRuntime.
            getRecPaziente(0);
        recPaziente.ddFoto = sFilename;
        recPaziente.updatePaziente();
        endEditAndRefresh();
    }
}
}

```

Listing 4.35: FmStatoCivile.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/
    MasterEmma.master" AutoEventWireup="true" CodeFile="
    FmStatoCivile.aspx.cs" Inherits="
    Tabelle_FmStatoCivile" %>

<%@ Register Assembly="DevExpress.Web.ASPxGridView.v9
    .3, _Version=9.3.6.0, _Culture=neutral, _PublicKeyToken=
    b88d1754d700e49a"
    Namespace="DevExpress.Web.ASPxGridView" TagPrefix="dx
    " %>

<%@ Register assembly="DevExpress.Web.ASPxEditors.v9.3,
    _Version=9.3.6.0, _Culture=neutral, _PublicKeyToken=
    b88d1754d700e49a" namespace="DevExpress.Web.
    ASPxEditors" tagprefix="dx" %>

<asp:Content ID="Content1" ContentPlaceHolderID="
    HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="
    MainContent" Runat="Server">

```

```

<dx:ASPxGridView ID="grdStatoCivile" runat="server"
  AutoGenerateColumns="False"
  KeyFieldName="id" onrowdeleting="
    grdStatoCivile_RowDeleting"
  onrowinserting="grdStatoCivile_RowInserting"
  onrowupdating="grdStatoCivile_RowUpdating">
  <SettingsBehavior ConfirmDelete="True" />
  <SettingsEditing Mode="EditForm" />
  <SettingsText ConfirmDelete="Confermi la
    cancellazione?" />
  <Columns>
    <dx:GridViewCommandColumn VisibleIndex="0">
      <EditButton Visible="True">
      </EditButton>
      <NewButton Visible="True">
      </NewButton>
      <DeleteButton Visible="True">
      </DeleteButton>
    </dx:GridViewCommandColumn>
    <dx:GridViewDataTextColumn Caption="Codice"
      FieldName="id" VisibleIndex="1" ReadOnly="
      true">
      <EditFormSettings Visible="False" />
    </dx:GridViewDataTextColumn>
    <dx:GridViewDataTextColumn Caption="
      Descrizione" FieldName="dd_descrizione"
      VisibleIndex="2">
    </dx:GridViewDataTextColumn>
  </Columns>
  <Paddings PaddingBottom="150px" PaddingLeft="250
    px" PaddingRight="250px"

```

```

                PaddingTop=" 50px" />
        </dx:ASPxGridView>
</asp:Content>

```

Listing 4.36: FmStatoCivile.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using DevExpress.Web.Data;

public partial class Tabelle_FmStatoCivile : System.Web
    .UI.Page
{
    protected void grdDataBind(bool setDatasource, bool
        doDatabind)
    {
        if (setDatasource)
            grdStatoCivile.DataSource = DALRuntime.
                getStatoCivile();
        if (doDatabind)
            grdStatoCivile.DataBind();
    }

    protected void Page_Init(object sender, EventArgs e)
    {
        grdDataBind(true, false);
    }

    protected void Page_Load(object sender, EventArgs e)

```

```
{
    if (!IsPostBack && !IsCallback)
        grdDataBind(false , true);
}

protected void grdStatoCivile_RowDeleting(object
    sender , ASPxDataDeletingEventArgs e)
{
    int id = Convert.ToInt32(e.Keys[0].ToString());
    DALRuntime.deleteStatoCivile(id);
    e.Cancel = true;
    grdDataBind(true , true);
}

protected void grdStatoCivile_RowInserting(object
    sender , ASPxDataInsertingEventArgs e)
{
    string ddDescr = e.NewValues["dd_descrizione"].
        ToString();
    DALRuntime.insertStatoCivile(ddDescr);
    e.Cancel = true;
    grdStatoCivile.CancelEdit();
    grdDataBind(true , true);
}

protected void grdStatoCivile_RowUpdating(object
    sender , ASPxDataUpdatingEventArgs e)
{
    int id = Convert.ToInt32(e.Keys[0].ToString());
    string ddDescr = e.NewValues["dd_descrizione"].
        ToString();
```

```

        DALRuntime.updateStatoCivile(id, ddDescr);
        e.Cancel = true;
        grdStatoCivile.CancelEdit();
        grdDataBind(true, true);
    }
}

```

Listing 4.37: FmTitoliDiStudio.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/
MasterEmma.master" AutoEventWireup="true" CodeFile="
FmTitoliDiStudio.aspx.cs" Inherits="
Tabelle_FmTitoliDiStudio" %>

<%@ Register Assembly="DevExpress.Web.ASPxGridView.v9
.3, Version=9.3.6.0, Culture=neutral, PublicKeyToken=
b88d1754d700e49a"
Namespace="DevExpress.Web.ASPxGridView" TagPrefix="dx
" %>

<asp:Content ID="Content1" ContentPlaceHolderID="
HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="
MainContent" Runat="Server">
    <dx:ASPxGridView ID="grdTitoliDiStudio" runat="
server" KeyFieldName="id"
    AutoGenerateColumns="False" onrowdeleting="
grdTitoliDiStudio_RowDeleting"
    onrowinserting="grdTitoliDiStudio_RowInserting"
    onrowupdating="grdTitoliDiStudio_RowUpdating">
        <SettingsBehavior ConfirmDelete="True" />
        <SettingsEditing Mode="EditForm" />

```

```

<SettingsText ConfirmDelete=" Confermi la
    cancellazione?" />
<Columns>
    <dx:GridViewCommandColumn VisibleIndex="0">
        <EditButton Visible="True">
            </EditButton>
        <NewButton Visible="True">
            </NewButton>
        <DeleteButton Visible="True">
            </DeleteButton>
        </dx:GridViewCommandColumn>
    <dx:GridViewDataTextColumn Caption=" Codice"
        FieldName=" id" VisibleIndex="1">
        <EditFormSettings Visible="False" />
    </dx:GridViewDataTextColumn>
    <dx:GridViewDataTextColumn Caption="
        Descrizione" FieldName=" dd_descrizione"
        VisibleIndex="2">
    </dx:GridViewDataTextColumn>
</Columns>
<Paddings PaddingBottom=" 150px" PaddingLeft=" 250
    px" PaddingRight=" 250px"
    PaddingTop=" 50px" />
</dx:ASPxGridView>
</asp:Content>

```

Listing 4.38: FmTitoliDiStudio.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;

```

```
using System.Web.UI.WebControls;
using DevExpress.Web.Data;

public partial class Tabelle_FmTitoliDiStudio : System.
    Web.UI.Page
{
    protected void grdDataBind(bool setDatasource, bool
        doDatabind)
    {
        if (setDatasource)
            grdTitoliDiStudio.DataSource = DALRuntime.
                getTitoli();
        if (doDatabind)
            grdTitoliDiStudio.DataBind();
    }

    protected void Page_Init(object sender, EventArgs e)
    {
        grdDataBind(true, false);
    }

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack && !IsCallback)
            grdDataBind(false, true);
    }

    protected void grdTitoliDiStudio_RowDeleting(object
        sender, ASPxDataDeletingEventArgs e)
    {
        int id = Convert.ToInt32(e.Keys[0].ToString());
    }
}
```



```
DALRuntime.deleteTitolo(id);
e.Cancel = true;
grdDataBind(true, true);
}

protected void grdTitoliDiStudio_RowInserting(object
sender, ASPxDataInsertingEventArgs e)
{
    string ddDescr = e.NewValues["dd_descrizione"].
        ToString();
    DALRuntime.insertTitolo(ddDescr);
    e.Cancel = true;
    grdTitoliDiStudio.CancelEdit();
    grdDataBind(true, true);
}

protected void grdTitoliDiStudio_RowUpdating(object
sender, ASPxDataUpdatingEventArgs e)
{
    int id = Convert.ToInt32(e.Keys[0].ToString());
    string ddDescr = e.NewValues["dd_descrizione"].
        ToString();
    DALRuntime.updateTitolo(id, ddDescr);
    e.Cancel = true;
    grdTitoliDiStudio.CancelEdit();
    grdDataBind(true, true);
}
}
```


Conclusioni

La produzione di software per la somministrazione di questionari in campo medico potrebbe rivelarsi in futuro una nicchia di mercato interessante. I questionari, in medicina, sono sempre più utilizzati, e spesso i singoli medici o gli studi associati non dispongono di grandi mezzi informatici. Inoltre la diffusione sempre maggiore di dispositivi portatili (smartphone, tablet), in grado di navigare in rete, potrebbe diventare un ulteriore stimolo all'informaticizzazione di questi importanti strumenti clinici. Una possibile evoluzione futuro di *Emma* potrebbe essere proprio una sua versione ottimizzata per smartphone.

Appendice A

Il database MySQL

MySQL è un database relazionale tra i più popolari. È robusto, veloce, affidabile e, essendo scritto in C e C++, è presente in moltissime piattaforme (da MacOS a Windows, da Netware a Solaris e Linux).

Oltre a queste qualità, un'altra ragione del suo successo è senz'altro il suo essere open source. Spesso questo è stato confuso col fatto di essere gratuito. In realtà la licenza di MySQL è di tipo GPL, ed è gratuita solo se utilizzata all'interno di un software anch'esso open source; in caso contrario è a pagamento (<http://www.mysql.com/about/legal>).

Di recente MySQL è stato acquisito da Oracle che, nonostante qualche iniziale perplessità del mondo legato a MySQL, ha confermato che continuerà a mantenere lo stesso modello di *licensing*.

MySQL è molto utilizzato in ambito web, tanto da essere uno dei componenti del famoso acronimo LAMP (Linux Apache MySQL PHP).

Per questo progetto è stata utilizzata la versione 5.1 per windows, con l'aggiunta della libreria per .NET Connector/Net 6.3.6, reperibile all'indirizzo: <http://dev.mysql.com/downloads/connector/net>.

MySQL supporta viste, stored procedure e un buon sottoinsieme dello standard ANSI SQL 99.

Una caratteristica molto particolare di questo database è la possibilità di selezionare differenti *storage engine*. Lo storage engine è lo strato responsabile

di tutte le operazioni a livello fisico del database. Nel tempo, per MySQL, ne sono stati sviluppati diversi, ognuno con diverse caratteristiche.

Alcuni, ad esempio, supportano le transazioni multi-statement, oppure gli indici BTREE; alcuni sono più adatti per un uso totalmente in memoria, altri per un'applicazione web.

Attualmente lo storage engine di default è InnoDB, che supporta, tra le altre cose, le transazioni multi-statement, gli indici BTREE e le foreign key [4].

Per quanto strano possa sembrare, l'engine MyISAM, che era lo standard fino a poco tempo fa, non supporta nè l'una nè l'altra di queste due caratteristiche che qualunque sviluppatore con un minimo di esperienza in altri DBMS definirebbe fondamentali.

Tuttavia certamente questa possibilità di scelta lascia una grande flessibilità ai progettisti. Lo storage engine può essere specificato anche a livello di singola tabella (nello stesso database possono così convivere engine differenti) e, nelle ultime versioni, addirittura essere selezionato runtime.

Appendice B

Il framework ASP.NET

ASP.NET è la parte dedicata allo sviluppo di applicazioni web del framework Microsoft .NET.

È probabilmente la parte che ha avuto più successo del framework che invece, dal canto suo, ha nel tempo visto un sostanziale ridimensionamento delle ambizioni iniziali di soppiantare completamente Java come piattaforma di sviluppo.

Il progetto di questa tesi utilizza ASP.NET nella modalità classica, nota ufficialmente come “web forms”:

in quest’ambito un’applicazione web è vista come una collezione di pagine (le web form, appunto) che sono contenute in file con l’estensione .aspx. Ognuno di questi file ne ha poi associato un altro con lo stesso nome ma estensione legata al linguaggio di programmazione scelto (ad esempio, in questa tesi, .cs per C#).

Il file .cs contiene la parte di codice, mentre la parte di “presentazione” (html, css ecc..) va nel .aspx.

Questo è il cosiddetto modello *Code-behind*, che è pensato per forzare il più possibile la separazione tra presentazione e codice di un’applicazione web.

Uno degli aspetti più interessanti di ASP.NET è il pattern *provider model*. Si tratta di un meccanismo che consente agli sviluppatori di sfruttare caratteristiche base, messe a disposizione dal sistema, adattandole però alle proprie

esigenze.

Ad esempio ci sono tutta una serie di componenti e servizi legati all'autenticazione degli utenti (il controllo in cui immettere codice utente e password, l'etichetta che mostra automaticamente il nome dell'utente collegato oppure il link per accedere ecc.). Sono tutti elementi già pronti che evitano agli sviluppatori di reinventare l'acqua calda in ogni nuovo progetto. La flessibilità è data dal fatto che tutti questi componenti sono basati su una definizione di interfaccia di cui è possibile definire una classe derivata che viene installata e diventa a tutti gli effetti parte del sistema.

Per *Emma* sono stati creati tre “provider” modificati:

- un *membership* provider. La classe base si chiama `MembershipProvider`, il derivato `EmmaMembershipProvider`.

I metodi che sono stati ridefiniti sono i seguenti:

- `void Initialize()` è il costruttore.
- `bool ValidateUser(string username, string password)` è il cuore del sistema: è il metodo che controlla se username e password sono corretti e, in tal caso, ritorna il valore booleano “true”.

- un *role* provider. In questo caso la classe base si chiama `RoleProvider` e l'implementazione in *Emma* `EmmaRoleProvider`. I metodi ridefiniti sono:

- `void Initialize()` costruttore.
- `string[] GetAllRoles()` ritorna tutti i ruoli definiti. In questo caso ritorna il contenuto della tabella `Z_PROFILO`.
- `string[] GetRolesForUser(string username)` ritorna i ruoli concessi all'utente “username”. In *Emma* il disegno consente un solo ruolo (o profilo) per utente.
- `bool IsUserInRole(string username, string roleName)` ritorna true se “username” possiede il ruolo “roleName”. Questo metodo è importante all'interno della *form based authentication*. Una

volta specificato che una certa cartella dell'applicazione ammette solo determinati ruoli, questo metodo viene richiamato dal framework per sapere se l'utente corrente appartiene ai ruoli ammessi oppure no.

- un *sitemap* provider. Questa classe ha il compito di fornire la struttura di navigazione del sito, che può essere visualizzata da diversi controlli (di tipo albero o menù), tutti alimentati da questa classe. In questo caso la classe `EmmaSitemapProvider` deriva da `StaticSitemapProvider`. In questo caso esiste anche un'altra classe base, `XMLSitemapProvider` che si può utilizzare quando la mappa di navigazione del sito è memorizzata in un file in formato XML. I metodi ridefiniti in *Emma* sono:

- `void Initialize()` il solito costruttore.
- `void Clear()` azzera la struttura.
- `SiteMapNode RootNode` ritorna il nodo radice.
- `SiteMapNode BuildSiteMap()` costruisce la mappa vera e propria. In questo c'è la logica che, in base al profilo dell'utente corrente, va a ricostruire le funzioni assegnate dalla tabella Z_FUNZIONE.

Bibliografia

- [1] Penta M, Carlyne A, Decruynaere C, *Analisi di Rasch e questionari di misura. Applicazioni in medicina e scienze sociali*, Milano, Springer, 2005
- [2] <http://healthskills.wordpress.com/2008/09/08/questionnaire-validation-a-brief-guide-for-readers-of-the-research-literature/>
- [3] Salaffi F, *Attività di Malattia e remissione nell'Artrite Reumatoide. Gli indici compositi*, Fidenza, Mattioli, 2010
- [4] http://dev.mysql.com/tech-resources/articles/storage-engine/part_3.html

Ringraziamenti

Ringrazio tutti i colleghi, gli amici e i semplici contatti internet che mi hanno insegnato così tanto e a cui ho restituito così poco.